

Ping and its various uses

Objective:

Learn how to use ping and its different parameters.

Purpose:

Ping is a simple and useful network-based utility which can be used to identify if a host is alive or dead. Technically, we can call it an echo reply. By “alive”, I mean that the host is active, and by “dead”, that the host is in shutdown mode. Anything which has a network card can be a host: computers, servers, switches, websites, smartphones, IOT devices, etc.

It is often useful when setting up some new infrastructure to use ping to test if your infrastructure can correctly reach the network.

Tool:

Kali Linux or Windows

Topology:

You can use Kali Linux for this lab.

Walkthrough:

Task 1:

Ping works on both Kali linux and Windows. For this lab, we will be demonstrating ping on Kali Linux VM machine. To begin, open a terminal window. Then, type the following:

```
ping google.com
```

```
(olalekan@kali)-[~]
$ ping google.com
PING google.com (142.250.200.110) 56(84) bytes of data:
64 bytes from mad41s13-in-f14.1e100.net (142.250.200.110): icmp_seq=1 ttl=112
time=277 ms
64 bytes from mad41s13-in-f14.1e100.net (142.250.200.110): icmp_seq=2 ttl=112
time=268 ms
64 bytes from mad41s13-in-f14.1e100.net (142.250.200.110): icmp_seq=3 ttl=112
time=286 ms
64 bytes from mad41s13-in-f14.1e100.net (142.250.200.110): icmp_seq=4 ttl=112
time=246 ms
64 bytes from mad41s13-in-f14.1e100.net (142.250.200.110): icmp_seq=5 ttl=112
time=286 ms
64 bytes from mad41s13-in-f14.1e100.net (142.250.200.110): icmp_seq=6 ttl=112
time=284 ms
64 bytes from mad41s13-in-f14.1e100.net (142.250.200.110): icmp_seq=7 ttl=112
time=285 ms
64 bytes from mad41s13-in-f14.1e100.net (142.250.200.110): icmp_seq=8 ttl=112
time=283 ms
^C
— google.com ping statistics —
8 packets transmitted, 8 received, 0% packet loss, time 7003ms
rtt min/avg/max/mdev = 246.424/276.947/286.186/12.905 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.

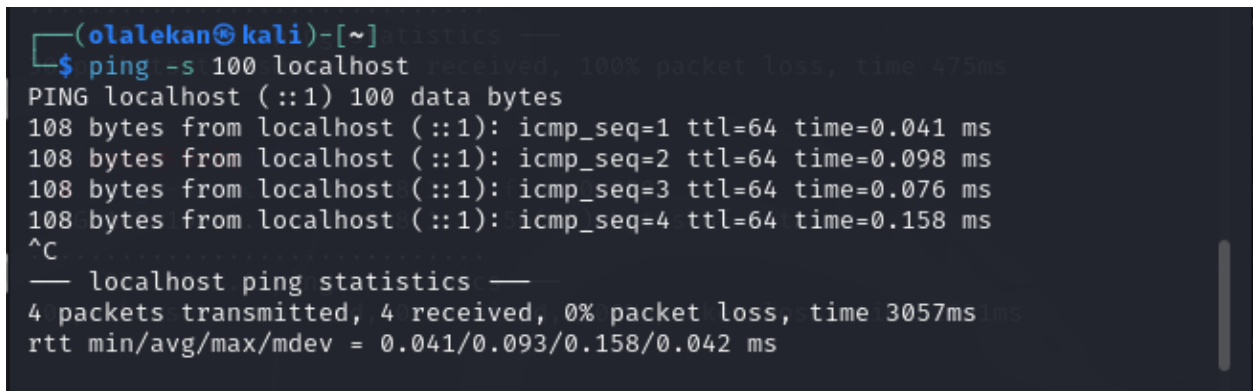
9) RTT (Round-trip time) metrics of those ping packages. RTT is the duration in milliseconds it takes for a network request to go from a starting point to a target and back again to the starting point.

Task 2:

We can set the packet size using the following commands:

ping -s 100 localhost

ping -s

A terminal window screenshot from a Kali Linux machine. The prompt is (olalekan@kali)-[~]. The user enters the command \$ ping -s 100 localhost. The output shows a 100% packet loss with a time of 475ms. The user then enters ^C to stop the command. The output shows the ping statistics: 4 packets transmitted, 4 received, 0% packet loss, time 3057ms. The RTT min/avg/max/mdev is 0.041/0.093/0.158/0.042 ms.

```
(olalekan@kali)-[~]
$ ping -s 100 localhost
PING localhost (::1) 100 data bytes
108 bytes from localhost (::1): icmp_seq=1 ttl=64 time=0.041 ms
108 bytes from localhost (::1): icmp_seq=2 ttl=64 time=0.098 ms
108 bytes from localhost (::1): icmp_seq=3 ttl=64 time=0.076 ms
108 bytes from localhost (::1): icmp_seq=4 ttl=64 time=0.158 ms
^C
— localhost ping statistics —
4 packets transmitted, 4 received, 0% packet loss, time 3057ms
rtt min/avg/max/mdev = 0.041/0.093/0.158/0.042 ms
```

100 google.com

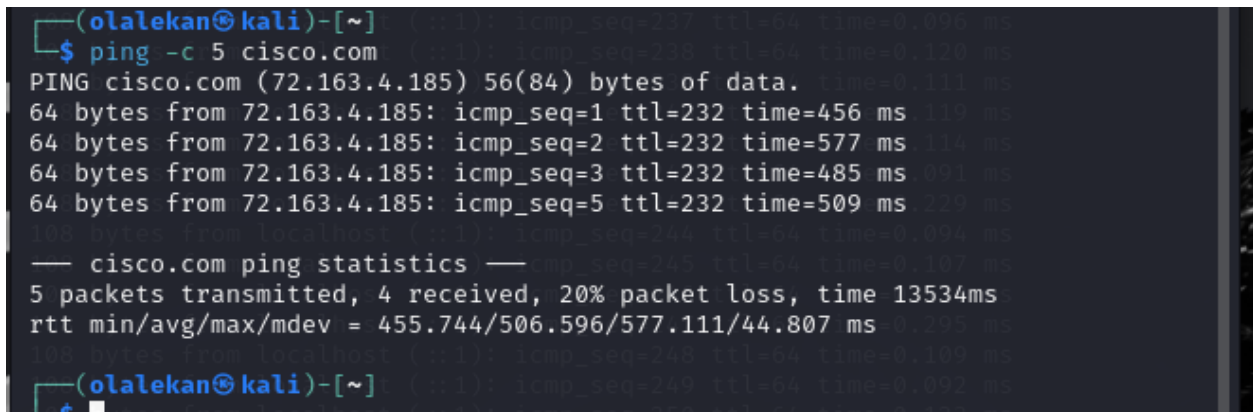
Some targets respond to ping packets as expected (1), some of them just drop (2).

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.

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
```

A terminal window screenshot showing a ping command being executed. The prompt is (olalekan@kali)-[~]. The command is \$ ping -c 5 cisco.com. The output shows five successful ping requests to cisco.com (72.163.4.185) with varying response times. Below the individual pings, a summary line reads: — cisco.com ping statistics —. The statistics show 5 packets transmitted, 4 received, and a 20% packet loss. The total time is 13534ms. The round-trip time (rtt) statistics are: min/avg/max/mdev = 455.744/506.596/577.111/44.807 ms. The terminal also shows subsequent ping requests to localhost with very low response times.

```
(olalekan@kali)-[~]  
$ ping -c 5 cisco.com  
PING cisco.com (72.163.4.185) 56(84) bytes of data. time=0.111 ms  
64 bytes from 72.163.4.185: icmp_seq=1 ttl=232 time=456 ms 119 ms  
64 bytes from 72.163.4.185: icmp_seq=2 ttl=232 time=577 ms 114 ms  
64 bytes from 72.163.4.185: icmp_seq=3 ttl=232 time=485 ms 101 ms  
64 bytes from 72.163.4.185: icmp_seq=5 ttl=232 time=509 ms 120 ms  
108 bytes from localhost: icmp_seq=244 ttl=64 time=0.094 ms  
— cisco.com ping statistics —  
5 packets transmitted, 4 received, 20% packet loss, time 13534ms  
rtt min/avg/max/mdev = 455.744/506.596/577.111/44.807 ms 0.295 ms  
108 bytes from localhost: icmp_seq=245 ttl=64 time=0.109 ms  
(olalekan@kali)-[~]
```

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
```

```
olalekan@kali: ~  
File Actions Edit View Help  
olalekan@kali: ~ x olalekan@kali: ~ x olaleka...kali: ~ x olaleka...kali: ~ x  
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data: 0.000 ms  
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.039 ms .167 ms  
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.091 ms .100 ms  
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.188 ms .113 ms  
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.179 ms .091 ms  
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.210 ms .078 ms  
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.094 ms .211 ms  
64 bytes from 127.0.0.1: icmp_seq=7 ttl=64 time=0.098 ms .186 ms  
64 bytes from 127.0.0.1: icmp_seq=8 ttl=64 time=0.148 ms .058 ms  
64 bytes from 127.0.0.1: icmp_seq=9 ttl=64 time=0.099 ms .110 ms  
64 bytes from 127.0.0.1: icmp_seq=10 ttl=64 time=0.101 ms .577 ms  
64 bytes from 127.0.0.1: icmp_seq=11 ttl=64 time=0.115 ms .092 ms  
64 bytes from 127.0.0.1: icmp_seq=12 ttl=64 time=0.177 ms .086 ms  
64 bytes from 127.0.0.1: icmp_seq=13 ttl=64 time=0.159 ms .092 ms  
64 bytes from 127.0.0.1: icmp_seq=14 ttl=64 time=0.098 ms .111 ms  
64 bytes from 127.0.0.1: icmp_seq=15 ttl=64 time=0.112 ms .162 ms  
64 bytes from 127.0.0.1: icmp_seq=16 ttl=64 time=0.096 ms .284 ms  
64 bytes from 127.0.0.1: icmp_seq=17 ttl=64 time=0.094 ms .105 ms  
64 bytes from 127.0.0.1: icmp_seq=18 ttl=64 time=0.110 ms .104 ms  
64 bytes from 127.0.0.1: icmp_seq=19 ttl=64 time=0.113 ms .111 ms  
64 bytes from 127.0.0.1: icmp_seq=20 ttl=64 time=0.108 ms .087 ms  
64 bytes from localhost: icmp_seq=21 ttl=64 time=0.110 ms  
— 127.0.0.1 ping statistics —  
20 packets transmitted, 20 received, 0% packet loss, time 3938ms  
rtt min/avg/max/mdev = 0.039/0.121/0.210/0.040 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. Unprivileged 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
```

```
olalekan@kali: ~  
File Actions Edit View Help  
olalekan@kali: ~ x olalekan@kali: ~ x olaleka...kali: ~ x olaleka...kali: ~ x  
64 bytes from 127.0.0.1: icmp_seq=17 ttl=64 time=0.021 ms 288 ms  
64 bytes from 127.0.0.1: icmp_seq=18 ttl=64 time=0.020 ms 101 ms  
64 bytes from 127.0.0.1: icmp_seq=19 ttl=64 time=0.020 ms 885 ms  
64 bytes from 127.0.0.1: icmp_seq=20 ttl=64 time=0.063 ms 891 ms  
64 bytes from localhost (127.0.0.1): icmp_seq=744 ttl=64 time=0.109 ms  
— 127.0.0.1 ping statistics —  
20 packets transmitted, 20 received, 0% packet loss, time 40ms  
rtt min/avg/max/mdev = 0.017/0.032/0.063/0.011 ms time=0.069 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
```

```
(root@kali)-[~]  
# ping -4n -c30 192.168.1.1 -f 100.net (192.168.1.1): icmp_seq=7 ttl=11  
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 475ms  
rtt min/avg/max/mdev = 251.027/295.918/321.727/27.202 ms  
  
(root@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.data.  
..... 100.net (192.168.1.1): icmp_seq=1 ttl=11  
— 192.168.1.1 ping statistics —  
30 packets transmitted, 0 received, 100% packet loss, time 1621ms seq=2 ttl=11  
time=169 ms  
100 bytes from 192.168.1.1: icmp_seq=2 ttl=11  
100 bytes from 192.168.1.1: icmp_seq=3 ttl=11
```