

Lab 12 – Ping and its various uses

Lab Objective:

Using ping and its different parameters.

Lab 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.

Lab Tool:

Kali Linux or Windows

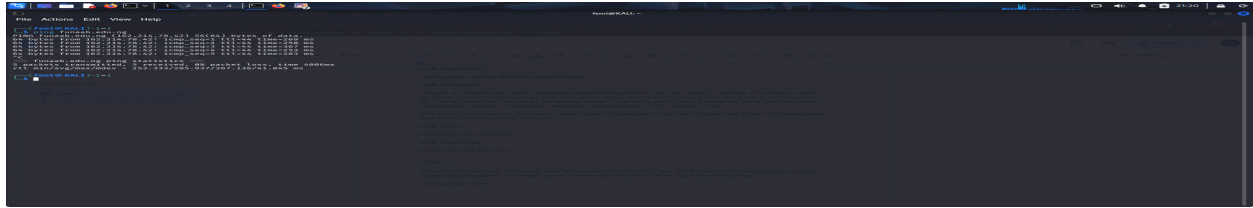
Lab Topology:

Kali Linux for this lab.

STEP 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 funaab.edu.ng
```



```
ping -n google.com
Pinging google.com [64.60.64.64] with 32 bytes of data:
64 bytes from google.com: icmp_seq=1 ttl=64 time=1.23 ms
64 bytes from google.com: icmp_seq=2 ttl=64 time=1.15 ms
64 bytes from google.com: icmp_seq=3 ttl=64 time=1.18 ms
64 bytes from google.com: icmp_seq=4 ttl=64 time=1.21 ms
64 bytes from google.com: icmp_seq=5 ttl=64 time=1.19 ms
64 bytes from google.com: icmp_seq=6 ttl=64 time=1.22 ms
64 bytes from google.com: icmp_seq=7 ttl=64 time=1.17 ms

Statistics: 7 packets: 100% success, 0% loss, average time=1.19 ms, stdev=0.03 ms
```

The ping command will continue to send ICMP packages to the destined 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

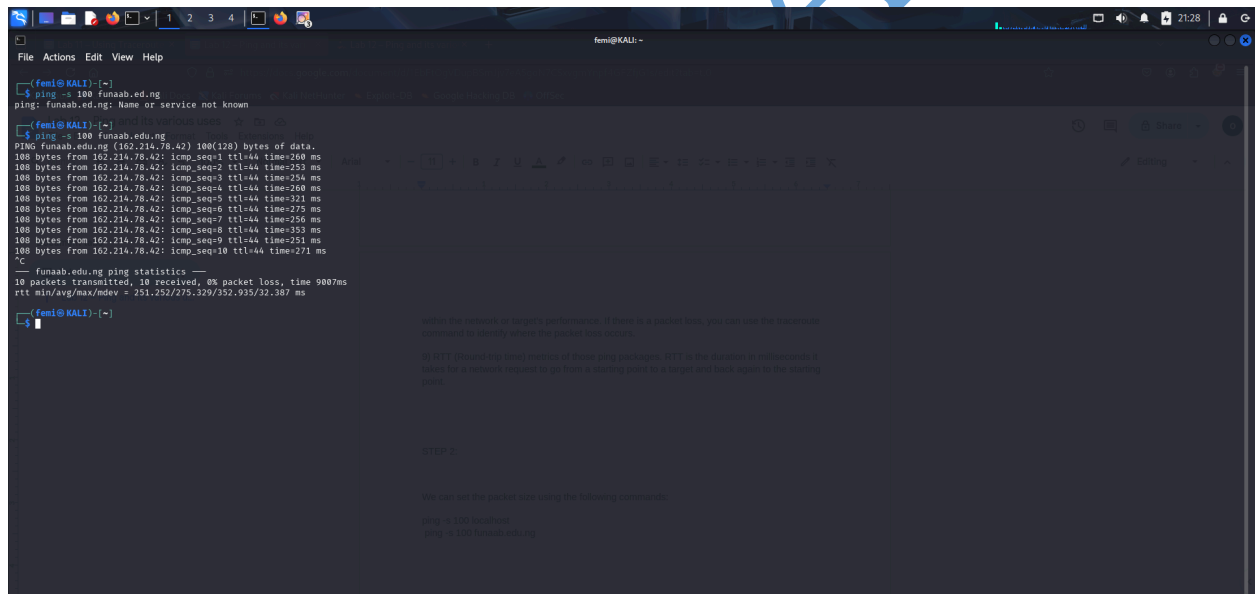
within the network or target's performance. If there is a packet loss, you can use the traceroute command to identify where the packet loss occurs.

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.

STEP 2:

We can set the packet size using the following commands:

```
ping -s 100 localhost
ping -s 100 funaab.edu.ng
```



```
femi@KALI:~$ ping -s 100 funaab.edu.ng
ping: funaab.edu.ng: Name or service not known

femi@KALI:~$ ping -s 100 funaab.edu.ng
PING funaab.edu.ng (162.214.78.42) 100(128) bytes of data:
100 bytes from 162.214.78.42: icmp_seq=1 ttl=44 time=260 ms
100 bytes from 162.214.78.42: icmp_seq=2 ttl=44 time=253 ms
100 bytes from 162.214.78.42: icmp_seq=3 ttl=44 time=254 ms
100 bytes from 162.214.78.42: icmp_seq=4 ttl=44 time=260 ms
100 bytes from 162.214.78.42: icmp_seq=5 ttl=44 time=321 ms
100 bytes from 162.214.78.42: icmp_seq=6 ttl=44 time=275 ms
100 bytes from 162.214.78.42: icmp_seq=7 ttl=44 time=256 ms
100 bytes from 162.214.78.42: icmp_seq=8 ttl=44 time=353 ms
100 bytes from 162.214.78.42: icmp_seq=9 ttl=44 time=251 ms
100 bytes from 162.214.78.42: icmp_seq=10 ttl=44 time=271 ms
^C
--- funaab.edu.ng ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9007ms
rtt min/avg/max/mdev = 251.252/275.329/352.935/32.387 ms

femi@KALI:~$
```

STEP 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 10 funaab.edu.ng
```

```
femi@KALI:~$ ping -c 10 funaab.edu.ng
ping: funaab.edu.ng: Name or service not known

femi@KALI:~$ ping -c 10 funaab.edu.ng
PING funaab.edu.ng (162.214.78.42) 100(128) bytes of data:
100 bytes from 162.214.78.42: icmp_seq=1 ttl=44 time=240 ms
100 bytes from 162.214.78.42: icmp_seq=2 ttl=44 time=253 ms
100 bytes from 162.214.78.42: icmp_seq=3 ttl=44 time=254 ms
100 bytes from 162.214.78.42: icmp_seq=4 ttl=44 time=240 ms
100 bytes from 162.214.78.42: icmp_seq=5 ttl=44 time=321 ms
100 bytes from 162.214.78.42: icmp_seq=6 ttl=44 time=275 ms
100 bytes from 162.214.78.42: icmp_seq=7 ttl=44 time=256 ms
100 bytes from 162.214.78.42: icmp_seq=8 ttl=44 time=353 ms
100 bytes from 162.214.78.42: icmp_seq=9 ttl=44 time=251 ms
100 bytes from 162.214.78.42: icmp_seq=10 ttl=44 time=271 ms
^C
--- funaab.edu.ng ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9007ms
rtt min/avg/max/mdev = 251.252/275.329/352.935/32.387 ms

femi@KALI:~$ ping -c 10 funaab.edu.ng
PING funaab.edu.ng (162.214.78.42) 56(84) bytes of data:
64 bytes from 162.214.78.42: icmp_seq=1 ttl=44 time=278 ms
64 bytes from 162.214.78.42: icmp_seq=2 ttl=44 time=253 ms
64 bytes from 162.214.78.42: icmp_seq=3 ttl=44 time=251 ms
64 bytes from 162.214.78.42: icmp_seq=4 ttl=44 time=254 ms
64 bytes from 162.214.78.42: icmp_seq=5 ttl=44 time=252 ms
64 bytes from 162.214.78.42: icmp_seq=6 ttl=44 time=230 ms
64 bytes from 162.214.78.42: icmp_seq=7 ttl=44 time=263 ms
64 bytes from 162.214.78.42: icmp_seq=8 ttl=44 time=254 ms
64 bytes from 162.214.78.42: icmp_seq=9 ttl=44 time=230 ms
64 bytes from 162.214.78.42: icmp_seq=10 ttl=44 time=251 ms
^C
--- funaab.edu.ng ping statistics ---
10 packets transmitted, 10 received, 0% packet loss, time 9017ms
rtt min/avg/max/mdev = 249.958/261.011/304.140/16.471 ms

femi@KALI:~$
```

STEP 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

```
femi@KALI:~$ ping -n 127.0.0.1
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=0.084 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.043 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.028 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.024 ms
64 bytes from 127.0.0.1: icmp_seq=5 ttl=64 time=0.028 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.025 ms
— 127.0.0.1 ping statistics —
6 packets transmitted, 6 received, 0% packet loss, time 5124ms
rtt min/avg/max/mdev = 0.024/0.038/0.084/0.021 ms

femi@KALI:~$ ping -n 127.0.0.1 -c 3
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=0.029 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.026 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.033 ms
— 127.0.0.1 ping statistics —
3 packets transmitted, 3 received, 0% packet loss, time 2846ms
rtt min/avg/max/mdev = 0.026/0.029/0.033/0.003 ms

femi@KALI:~$ ping -n 127.0.0.1 -c 3
ping: 127.0.0.1: Address family for hostname not supported

femi@KALI:~$ ping -n localhost -c 3
PING localhost (::1) 56 data bytes
64 bytes from localhost (::1): icmp_seq=1 ttl=64 time=0.085 ms
64 bytes from localhost (::1): icmp_seq=2 ttl=64 time=0.036 ms
64 bytes from localhost (::1): icmp_seq=3 ttl=64 time=0.044 ms
— localhost ping statistics —
3 packets transmitted, 3 received, 0% packet loss, time 2839ms
rtt min/avg/max/mdev = 0.036/0.055/0.085/0.021 ms

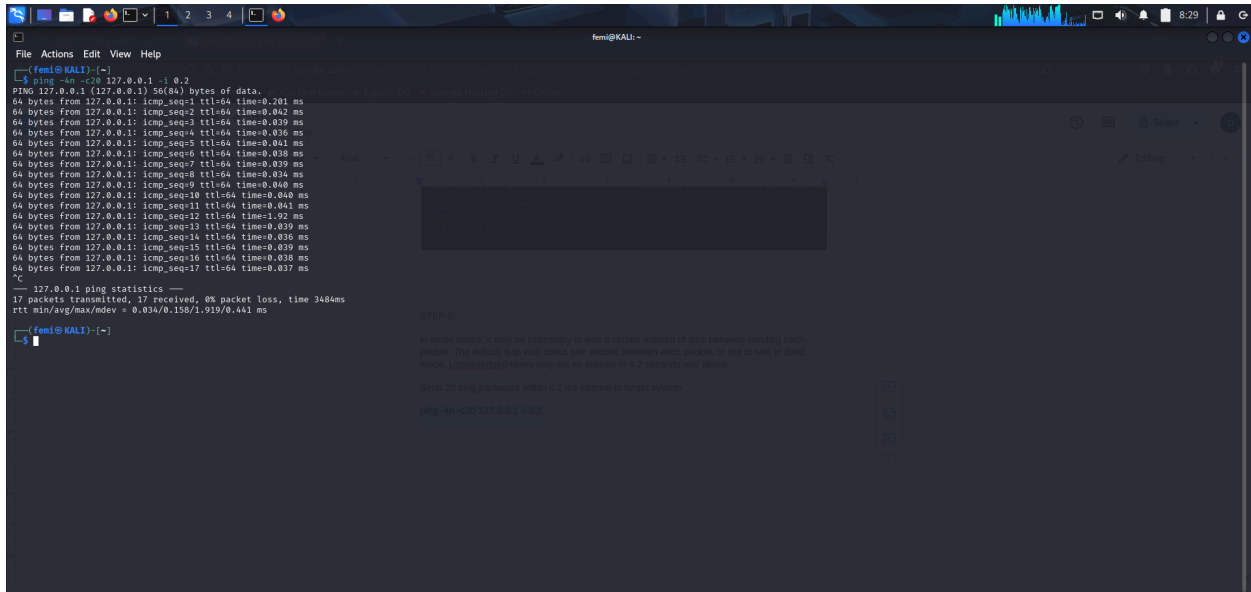
femi@KALI:~$ ping -n don't localhost -c 5
PING localhost (127.0.0.1) 56(84) bytes of data:
64 bytes from localhost (127.0.0.1): icmp_seq=1 ttl=64 time=0.019 ms
64 bytes from localhost (127.0.0.1): icmp_seq=2 ttl=64 time=0.026 ms
64 bytes from localhost (127.0.0.1): icmp_seq=3 ttl=64 time=0.034 ms
64 bytes from localhost (127.0.0.1): icmp_seq=4 ttl=64 time=0.027 ms
64 bytes from localhost (127.0.0.1): icmp_seq=5 ttl=64 time=0.037 ms
— localhost ping statistics —
```

STEP 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
```



```
(femi@KALI)-[~]
$ ping -n -c30 127.0.0.1 -i 0.2
PING 127.0.0.1 (127.0.0.1) 56(64) bytes of data:
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.201 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.042 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.039 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=0.041 ms
64 bytes from 127.0.0.1: icmp_seq=6 ttl=64 time=0.038 ms
64 bytes from 127.0.0.1: icmp_seq=7 ttl=64 time=0.039 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=0.040 ms
64 bytes from 127.0.0.1: icmp_seq=10 ttl=64 time=0.040 ms
64 bytes from 127.0.0.1: icmp_seq=11 ttl=64 time=0.041 ms
64 bytes from 127.0.0.1: icmp_seq=12 ttl=64 time=1.192 ms
64 bytes from 127.0.0.1: icmp_seq=13 ttl=64 time=0.039 ms
64 bytes from 127.0.0.1: icmp_seq=14 ttl=64 time=0.036 ms
64 bytes from 127.0.0.1: icmp_seq=15 ttl=64 time=0.039 ms
64 bytes from 127.0.0.1: icmp_seq=16 ttl=64 time=0.035 ms
64 bytes from 127.0.0.1: icmp_seq=17 ttl=64 time=0.037 ms
^C
--- 127.0.0.1 ping statistics ---
17 packets transmitted, 17 received, 0% packet loss, time 3484ms
rtt min/avg/max/ndev = 0.034/0.156/1.919/0.441 ms
(femi@KALI)-[~]
$
```

STEP 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
```

In this flood test, packet loss is 73% in test number (1), while in test number (2), when the packet is sent with 50 ms delay, the loss is around 30%.

This feature can be used to slow down the target system's network or to measure end-to-end network performance. It can also be used to create artificial loads. For this reason, it is necessary to be careful when using it on systems in production.

