Internship Report File

Mohammad Nishar Ansari

Cyber security Intern

fping

fping is a small command line tool to send ICMP (Internet Control Message Protocol) echo request to network hosts, like ping, but much higher performing when pinging multiple hosts. fping totally differs from ping in that you can define any number of hosts on the command line or specify a file with the list of IP addresses or hosts to ping.

For example, using fping, we can specify the complete network range (192.168.0.1/24). It will send Fping request to host and move to another target host in a round-robin fashion. Unlike ping, Fping is meant for basically scripting.

Unlike the standard ping command, fping can send ICMP echo requests to multiple hosts in parallel, making it useful for checking the status of multiple hosts at once. It's often used in network monitoring, troubleshooting, and automation scripts.

Flags:-

-h or --help: Displays a help message showing all available options.

-v or --version: Displays the version of fping.

-a: Show targets that are alive.

-u: Show targets that are unreachable.

-e: Show elapsed times of responses.

-q: Quiet mode. Only display errors.

-g: Generate target list from a given IP address range.

-s: Print summary statistics.

-c <count>: Specify the number of ICMP echo requests to send to each target (default is one).

-t <timeout>: Set the timeout for waiting for a response (in milliseconds).

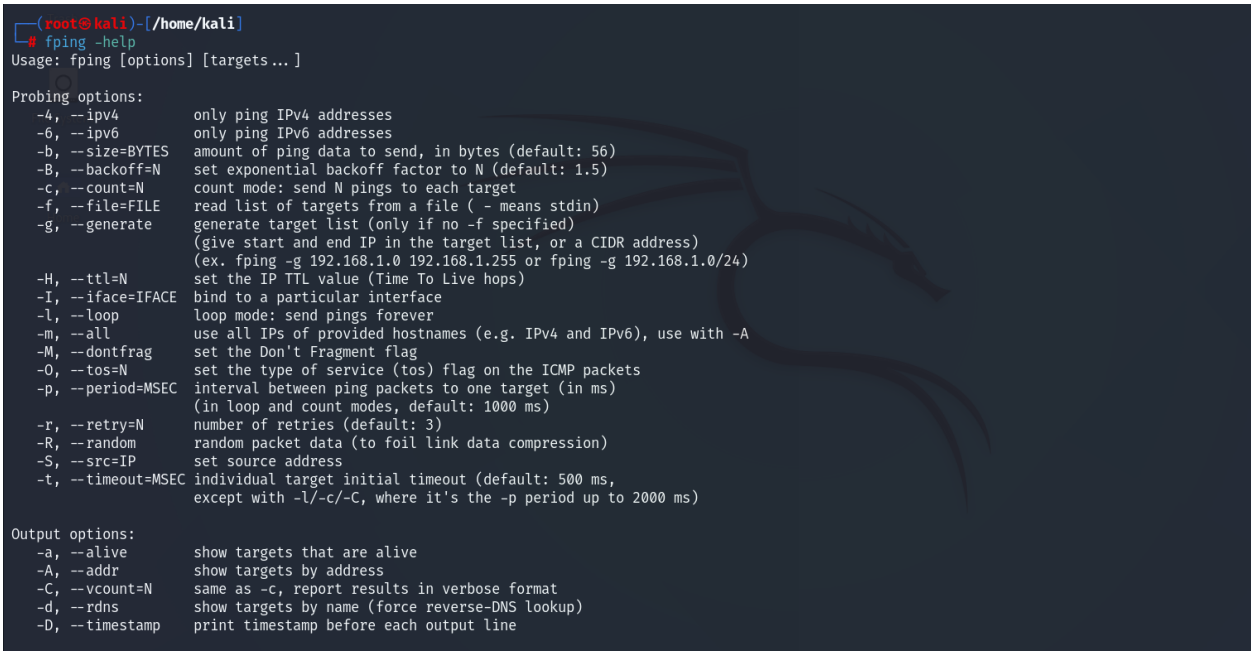
-i <interval>: Set the interval between sending ICMP echo requests (in milliseconds).

-r <retry>: Set the number of retries for a failed request (default is 3).

-l: Continuous loop mode. Send ICMP echo requests continuously to targets.

Usage: -

1. To have a insight of fping manual page type fping along with -help.



```
┌──(root㉿kali)-[/home/kali]
└─# fping -help
Usage: fping [options] [targets...]

Probing options:
   -4, --ipv4         only ping IPv4 addresses
   -6, --ipv6         only ping IPv6 addresses
   -b, --size=BYTES   amount of ping data to send, in bytes (default: 56)
   -B, --backoff=N    set exponential backoff factor to N (default: 1.5)
   -c, --count=N      count mode: send N pings to each target
   -f, --file=FILE    read list of targets from a file ( - means stdin)
   -g, --generate     generate target list (only if no -f specified)
                      (give start and end IP in the target list, or a CIDR address)
                      (ex. fping -g 192.168.1.0 192.168.1.255 or fping -g 192.168.1.0/24)
   -H, --ttl=N        set the IP TTL value (Time To Live hops)
   -I, --iface=IFACE  bind to a particular interface
   -l, --loop         loop mode: send pings forever
   -m, --all          use all IPs of provided hostnames (e.g. IPv4 and IPv6), use with -A
   -M, --dontfrag     set the Don't Fragment flag
   -O, --tos=N        set the type of service (tos) flag on the ICMP packets
   -p, --period=MSEC  interval between ping packets to one target (in ms)
                      (in loop and count modes, default: 1000 ms)
   -r, --retry=N      number of retries (default: 3)
   -R, --random       random packet data (to foil link data compression)
   -S, --src=IP       set source address
   -t, --timeout=MSEC individual target initial timeout (default: 500 ms,
                      except with -l/-c/-C, where it's the -p period up to 2000 ms)

Output options:
   -a, --alive        show targets that are alive
   -A, --addr         show targets by address
   -C, --vcount=N     same as -c, report results in verbose format
   -d, --rdns         show targets by name (force reverse-DNS lookup)
   -D, --timestamp    print timestamp before each output line
```

2. To check the target is alive or not use fping command along with the target domain.

```
┌──(root㉿kali)-[/home/kali]
└─# fping google.com
google.com is alive
```

3. Here we use -l to run the ping command in a loop until the user interrupts it.

```
┌──(root㉿kali)-[/home/kali]
└─# fping -l google.com
google.com : [0], 64 bytes, 15.4 ms (15.4 avg, 0% loss)
google.com : [1], 64 bytes, 25.9 ms (20.7 avg, 0% loss)
google.com : [2], 64 bytes, 14.1 ms (18.5 avg, 0% loss)
google.com : [3], 64 bytes, 22.4 ms (19.5 avg, 0% loss)
google.com : [4], 64 bytes, 15.6 ms (18.7 avg, 0% loss)
google.com : [5], 64 bytes, 20.5 ms (19.0 avg, 0% loss)
google.com : [6], 64 bytes, 24.4 ms (19.8 avg, 0% loss)
google.com : [7], timed out (19.8 avg, 12% loss)
google.com : [8], 64 bytes, 14.3 ms (19.1 avg, 11% loss)
google.com : [9], 64 bytes, 19.3 ms (19.1 avg, 10% loss)
google.com : [10], 64 bytes, 16.8 ms (18.9 avg, 9% loss)
google.com : [11], 64 bytes, 18.0 ms (18.8 avg, 8% loss)
google.com : [12], 64 bytes, 18.5 ms (18.8 avg, 7% loss)
google.com : [13], 64 bytes, 21.4 ms (19.0 avg, 7% loss)
google.com : [14], 64 bytes, 18.1 ms (18.9 avg, 6% loss)
^C
google.com : xmt/rcv/%loss = 15/14/6%, min/avg/max = 14.1/18.9/25.9
```

4. Here we use -p to define the timestamp of the ping which is 10 milliseconds in our case and to demonstrate the timestamp we use -d so that prints the timestamp of each packet sent.

```
┌──(root㉿kali)-[/home/kali]
└─# fping -D -l -p 10  google.com
[1716582016.48443] google.com : [0], timed out (NaN avg, 100% loss)
[1716582016.49801] google.com : [1], timed out (NaN avg, 100% loss)
[1716582016.51122] google.com : [2], timed out (NaN avg, 100% loss)
[1716582016.52361] google.com : [3], timed out (NaN avg, 100% loss)
[1716582016.53414] google.com : [4], timed out (NaN avg, 100% loss)
[1716582016.54475] google.com : [5], timed out (NaN avg, 100% loss)
[1716582016.55516] google.com : [6], timed out (NaN avg, 100% loss)
[1716582016.56563] google.com : [7], timed out (NaN avg, 100% loss)
[1716582016.57951] google.com : [8], timed out (NaN avg, 100% loss)
[1716582016.59157] google.com : [9], timed out (NaN avg, 100% loss)
[1716582016.60350] google.com : [10], timed out (NaN avg, 100% loss)
[1716582016.61498] google.com : [11], timed out (NaN avg, 100% loss)
[1716582016.62540] google.com : [12], timed out (NaN avg, 100% loss)
[1716582016.63588] google.com : [13], timed out (NaN avg, 100% loss)
[1716582016.64667] google.com : [14], timed out (NaN avg, 100% loss)
[1716582016.65750] google.com : [15], timed out (NaN avg, 100% loss)
[1716582016.66943] google.com : [16], timed out (NaN avg, 100% loss)
[1716582016.68035] google.com : [17], timed out (NaN avg, 100% loss)
[1716582016.69134] google.com : [18], timed out (NaN avg, 100% loss)
[1716582016.70152] google.com : [19], timed out (NaN avg, 100% loss)
[1716582016.71260] google.com : [20], timed out (NaN avg, 100% loss)
[1716582016.72287] google.com : [21], timed out (NaN avg, 100% loss)
[1716582016.73608] google.com : [22], timed out (NaN avg, 100% loss)
[1716582016.75142] google.com : [23], timed out (NaN avg, 100% loss)
[1716582016.76233] google.com : [24], timed out (NaN avg, 100% loss)
[1716582016.77470] google.com : [25], timed out (NaN avg, 100% loss)
[1716582016.78486] google.com : [26], timed out (NaN avg, 100% loss)
[1716582016.79504] google.com : [27], timed out (NaN avg, 100% loss)
[1716582016.80519] google.com : [28], timed out (NaN avg, 100% loss)
[1716582016.81599] google.com : [29], timed out (NaN avg, 100% loss)
[1716582016.82671] google.com : [30], timed out (NaN avg, 100% loss)
```

5. In order to check the host within the subnet we will use -g command.