NAME

sc_pinger — scamper driver to run ping with different probe methods on a list of addresses.

SYNOPSIS

sc_pinger [**-?D**] [**-a** *infile*] [**-c** *probe-count*] [**-l** *limit*] [**-m** *method*]
[**-M** *move-dir*] [**-o** *outfile*] [**-p** *port*] [**-R** *unix-remote*] [**-t** *logfile*]
[**-U** *unix-socket*]

DESCRIPTION

The **sc_pinger** utility provides the ability to connect to a running scamper(1) instance and run ping on a set of IPv4 and IPv6 addresses. For each address in the file, **sc_pinger** will try ICMP, UDP, and TCP-ack probe methods to solicit responses from the address. **sc_pinger** will not try all methods if one method obtains responses. The output of **sc_pinger** is written to a warts(5) file, which can then be processed to extract details of responses. The options are as follows:

-? prints a list of command line options and a synopsis of each.

-D causes **sc_pinger** to detach and become a daemon.

-a *infile*
specifies the name of the input file which consists of a sequence of IPv4 and IPv6 addresses, one per line.

-c *probe-count*
specifies the number of probes to send for each method. **sc_pinger** accepts two formats: a single integer that specifies the number of probes (and responses) desired; or, two integers, separated by /, that specify the number of responses desired and maximum number of probes to send. By default, **sc_pinger** seeks three responses from up to five probes.

-l *limit*
specifies the number of objects to write to an output file, before closing it and opening the next file. The output file must contain a %u format specifier, which **sc_pinger** uses to embed a counter value that increments with each new output file. If the user uses the move option, **sc_pinger** moves the file when it closes the file.

-m *method*
specifies a single probe method to try. The available probe methods are the same as scamper's ping implementation, listed in scamper(1) manual page. By default, **sc_pinger** uses ICMP-echo, UDP-dport, and TCP-ack-sport to destination port 80.

-M *move-dir*
specifies the name of the directory to move completed files to. By default, **sc_pinger** leaves completed files in place.

-o *outfile*
specifies the name of the output file to be written. The output file will use the warts(5) format.

-p *port*
specifies the port on the local host where scamper(1) is accepting control socket connections.

-R *unix-remote*
specifies the name of a unix domain socket on the local host where a remote scamper(1) instance is accepting commands.

-t *logfile*
specifies the name of a file to log output from **sc_pinger** generated at run time.

-U *unix-local*
specifies the name of a unix domain socket on the local host where a local scamper(1) instance is accepting commands.

EXAMPLES

Given a set of IPv4 and IPv6 address sets in a file named infile.txt:

```
192.0.2.1
192.0.32.10
192.0.31.60
2001:db8::1
```

and a scamper(1) daemon listening on port 31337, then these addresses can be probed using

```
sc_pinger -a infile.txt -o outfile.warts -p 31337
```

To send 4 probes, and stop after receiving two responses:

```
sc_pinger -a infile.txt -o outfile.warts -p 31337 -c 2/4
```

To use ICMP-echo and TCP-syn probes to destination port 443

```
sc_pinger -a infile.txt -o outfile.warts -p 31337 -m icmp-echo -m
'tcp-syn -d 443'
```

The following command writes a series of gzip-compressed warts(5) files, each of which have up to 1000 objects in them, with names such as outfile_0000.warts.gz, outfile_0001.warts.gz, moving them to the finished directory:

```
sc_pinger -a infile.txt -o outfile_%04u.warts.gz -p 31337 -l 1000 -m
finished
```

A user can concatenate these files into a final bzip2-compressed warts(5) file with sc_wartscat(1):

```
sc_wartscat -o outfile_final.warts.bz2 outfile_0000.warts.gz
outfile_0001.warts.gz
```

SEE ALSO

scamper(1), sc_wartscat(1), sc_wartsdump(1), sc_warts2json(1), sc_warts2text(1)

AUTHORS

sc_pinger was written by Matthew Luckie <mjl@luckie.org.nz>.