

NAME

sc_prefixprober — scamper driver to probe addresses in specified prefixes

SYNOPSIS

```
sc_prefixprober [-?D] [-a in-file] [-c command] [-d duration] [-l limit]
                [-L list-attr] [-m move-dir] [-o out-file] [-O options]
                [-p port] [-R unix-remote] [-t log-file] [-U unix-local]
                [-x dnp-file]
```

DESCRIPTION

The **sc_prefixprober** utility provides the ability to connect to a running **scamper(1)** instance and use it to probe addresses in specified prefixes.

sc_prefixprober can probe both the first and a randomly-selected address in each prefix. When **sc_prefixprober** is instructed to probe both, **sc_prefixprober** will probe the addresses in a single prefix serially, and back-to-back so that measurements within a single prefix occur close in time.

The supplied prefixes can be nested. If a /24 prefix is contained in a less-specific /23, **sc_prefixprober** will probe addresses in both the specified /24, and the remaining /24 contained in the /23.

The command line options for **sc_prefixprober** are as follows:

- ? prints a list of command line options and a synopsis of each.
- D causes **sc_prefixprober** to detach and become a daemon.
- a *in-file*
 specifies the name of the input file which consists of IPv4 and IPv6 prefixes, one per line.
- c *command*
 specifies the command to use with each address. Valid commands are ping, trace, and tracerb. By default, **sc_prefixprober** uses trace.
- d *duration*
 specifies the total duration allowed for all measurements. **sc_prefixprober** will space probing of individual prefixes out over the total duration specified. By default, **sc_prefixprober** probes prefixes as fast as allowed by **scamper(1)**.
- 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_prefixprober** uses to embed a counter value that increments with each new output file. If the user uses the move option, **sc_prefixprober** moves the file when it closes the file.
- L *list-attr*
 allows **sc_prefixprober** to override **scamper(1)** default values for list and cycle objects. The current choices for this option are:
 - id=%u specify a 32-bit unsigned integer for the list id.
 - name=%s specify a string for the list's name attribute.
 - descr=%s specify a string for the list's description attribute.
 - monitor=%s specify a string for the list's monitor attribute.
 - cycle-id=%u specify a 32-bit unsigned integer for the cycle id.
- m *move-dir*
 specifies the name of the directory to move completed files to. By default, **sc_prefixprober** leaves completed files in place.

- o** *out-file*
specifies the prefix of the name of the output file to be written. The output file will use the `warts(5)` format, and can be compressed with `gz`, `bz2`, or `xz` at collection time if the specified out-file has the equivalent extension, or the output type was explicitly specified with **-O**
- O** *options*
allows the behavior of `sc_prefixprober` to be further tailored. The current choices for this option are:
 - first**: probe first address in prefix.
 - random**: probe random address in prefix.
 - noshuffle**: do not shuffle probe order.
 - warts.gz**: compress warts output using `gzip` compression.
 - warts.bz2**: compress warts output using `bzip2` compression.
 - warts.xz**: compress warts output using `xz` compression.
- p** *port*
specifies the port on the local host where `scamper(1)` is accepting control socket connections.
- R** *unix-remote*
specifies the unix domain socket on the local host where a remote `scamper(1)` instance is accepting commands.
- t** *log-file*
specifies the name of a file to log output from `sc_prefixprober` generated at run time.
- U** *unix-local*
specifies the unix domain socket on the local host where a local `scamper(1)` instance is accepting commands.
- x** *dnf-file*
specifies a file containing prefixes whose addresses must not be probed.

EXAMPLES

Given a set of prefixes in a file named `infile.txt`:

```
192.0.30.0/23
192.0.30.0/24
192.0.2.0/24
```

and a `scamper(1)` instance listening on port 31337, then both the first and a randomly selected address within each prefix can be tracerouted using `ICMP-paris` as follows:

```
sc_prefixprober -c 'trace -P icmp-paris' -a infile.txt -o
outfile.warts -p 31337 -O random -O first -L name=foo
```

In this scenario, `sc_prefixprober` may probe 192.0.30.1, 192.0.30.69, 192.0.31.1, 192.0.31.169, 192.0.2.1, and 192.0.2.233. These are addresses in the two specified /24s, and a /24 contained in the less-specific /23 that was not covered by a more-specific /24. The output `warts(5)` file will have the list's name recorded as `foo`.

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_prefixprober -c 'ping' -a infile.txt -o outfile_%04u.warts.gz -p
31337 -O first -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)`, `warts(5)`

AUTHORS

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