NAME

sc_speedtrap — scamper driver to resolve aliases for a set of IPv6 interfaces.

SYNOPSIS

sc_speedtrap [**-?Iv**] [**-a** *addressfile*] [**-A** *aliasfile*] [**-l** *logfile*] [**-o** *outfile*] [**-p** *port*] [**-R** *unix-remote*] [**-s** *stop*] [**-S** *skipfile*] [**-U** *unix-local*]

sc_speedtrap [**-d** *dump*] [*file ...*]

DESCRIPTION

The **sc_speedtrap** utility provides the ability to connect to a running scamper(1) instance and use it to collect data for alias resolution of a set of IPv6 addresses using the "speedtrap" technique. **sc_speedtrap** induces each address to send fragmented ICMP echo replies, with the goal of obtaining an incrementing Identifier (ID) field in the fragmentation header. If two addresses are aliases, they will return ICMP echo replies with a monotonically increasing value in the ID field because the ID field is implemented as a counter shared amongst all interfaces. **sc_speedtrap** implements a scalable algorithm to quickly determine which addresses are aliases. Further information about the algorithm is found in the "see also" section. The supported options to **sc_speedtrap** are as follows:

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

-v prints the version of **sc_speedtrap** and exits.

-a *addressfile*
specifies the name of the input file which consists of a sequence of IPv6 addresses to resolve for aliases, one address per line.

-A *aliasfile*
specifies the name of an output file which will receive pairs of aliases, one address-pair per line.

-d *dump*
specifies the number identifying an analysis task to conduct. Valid dump numbers are 1-3. See the examples section.

-I specifies that the addressfile contains only interfaces known to send fragmentation headers containing incrementing values.

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

-o *outfile*
specifies the name of the output file to be written. The output file will use the warts 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. The unix-remote parameter can either be a unix domain socket for a single remote scamper(1) instance, or be a sc_remoted(1) mux socket with the name of the remote VP encoded after a trailing slash.

-s *stop*
specifies the step at which **sc_speedtrap** should halt. The available steps are "classify", "descend", "overlap", "descend2", "candidates", and "ally".

-S *skipfile*
specifies the name of an input file which contains known aliases that do not need to be resolved, one address-pair per line.

-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 IPv6 addresses contained in a file named addressfile.txt and a scamper process listening on port 31337 configured to probe at 30 packets per second started as follows:

```
scamper -P 31337 -p 30
```

the following command will resolve the addresses for aliases, store the raw measurements in outfile1.warts, and record the interface-pairs that are aliases in aliases.txt:

```
sc_speedtrap -p 31337 -a addressfile.txt -o outfile1.warts -A
aliases.txt
```

Given a `sc_remoted`(1) process listening on a unix domain socket named /path/to/socket, and a remote vantage point named 'foo' connected to the controller, probe the addresses with the remote vantage point using:

```
sc_speedtrap -R /path/to/socket/foo -a addressfile.txt -o
outfile2.warts
```

The next example is useful when inferring aliases from multiple vantage points. Given the output of aliases.txt from a previous measurement, the following will resolve the addressfile for aliases, skipping those in aliases.txt, and appending the new aliases to aliases.txt:

```
sc_speedtrap -p 31337 -a addressfile.txt -o outfile3.warts -A
aliases.txt -S aliases.txt
```

To obtain a transitive closure of routers from an input warts file:

```
sc_speedtrap -d 1 outfile1.warts
```

To obtain a list of the interfaces probed and their IPID behaviour:

```
sc_speedtrap -d 2 outfile1.warts
```

To obtain statistics of how many probes are sent in each stage, and how long the stage takes:

```
sc_speedtrap -d 3 outfile1.warts
```

SEE ALSO

M. Luckie, R. Beverly, W. Brinkmeyer, and k. claffy, *Speedtrap: Internet-scale IPv6 Alias Resolution*, Proc. ACM/SIGCOMM Internet Measurement Conference 2013. `scamper`(1), `sc_ally`(1), `sc_ipiddump`(1), `sc_remoted`(1), `sc_wartsdump`(1), `sc_warts2text`(1), `sc_warts2json`(1)

AUTHORS

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