NAME

sc_radargun — scamper driver to run radargun on a list of candidate aliases.

SYNOPSIS

sc_radargun [**-?D**] [**-a** *infile*] [**-f** *fudge*] [**-o** *outfile*] [**-O** *options*] [**-p** *port*] [**-P** *pps*] [**-q** *attempts*] [**-r** *wait-round*] [**-R** *round-count*] [**-t** *logfile*] [**-U** *unix*]

sc_radargun [**-d** *dump*] data-file

DESCRIPTION

The **sc_radargun** utility provides the ability to connect to a running scamper(1) instance and have a set of IPv4 addresses for aliases using the Radargun technique. For all addresses in the file, **sc_radargun** establishes which probe methods (UDP, TCP-ack, ICMP-echo) solicit an incrementing IP-ID value, and then uses the Radargun technique on addresses where a probe method is able to obtain an incrementing IP-ID for the addresses. The output is written to a warts file. The options are as follows:

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

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

-a *infile*
specifies the name of the input file which consists of a list of IPv4 addresses. The file can either contain sets to test, one set per line, or simply one set, one address per line.

-d *dump*
specifies the dump ID to use to analyze the collected data. Currently, only dumpid 1 is valid, which dumps candidate aliases.

-f *fudge*
specifies the fudge to use when inferring if a device is deriving IP-ID values from a counter. By default, responses the maximum difference between two samples must be no larger than 5000. The fudge value also impacts alias inference. If a value of zero is used, the IP-ID samples must simply be in order.

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

-O *options*
allows the behavior of **sc_radargun** to be further tailored. The current choices for this option are:
- **nobs:** do not consider if IP-ID values might be byte-swapped in the header
- **noreserved:** do not probe reserved IP addresses.
- **rows:** the addresses in the input file are supplied in rows, and the radargun measurements should consider each set in turn.
- **nobudget:** do not consider if the radargun measurement can complete in the round time give the packets-per-second rate specified.
- **tc:** when dumping candidate aliases, report the transitive closure, rather than pairs in isolation.

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

-P *pps*
specifies the packets-per-second rate that scamper is running at. The PPS value is used to infer if the radargun measurement can fit in scamper's probe budget.

-q *attempts*
specifies the number of probe packets to use to when inferring if an IP address assigns IP-ID values from a counter.

-r *wait-round*
specifies the length of time, in seconds, each round should aim to complete in. By default, 30 seconds.

-R *round-count*
specifies the number of rounds to pursue in radargun. By default, 30 rounds.

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

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

EXAMPLES

sc_radargun requires a scamper(1) instance listening on a port for commands in order to collect data, at 20 packets per second:

```
scamper -P 31337 -p 20
```

will start a scamper(1) instance listening on port 31337 on the loopback interface. To use **sc_radargun** to infer which addresses might be aliases, listed in a file named set-1.txt

```
192.0.2.2
192.0.32.10
192.0.30.64
192.0.31.8
```

the following command will test these IP addresses for aliases using ICMP, UDP, and TCP probes (as appropriate) using the radargun technique with 10 rounds, each round taking 4 seconds:

```
sc_radargun -a set-1.txt -o set-1.warts -p 20 -r 4 -R 10
```

To use **sc_radargun** to infer which addresses might be aliases, listed in a file named set-2.txt organized as sets of candidate aliases to test:

```
192.0.2.2 192.0.32.10 192.0.30.64 192.0.31.8
192.0.2.3 192.0.32.11 192.0.30.65 192.0.31.9
```

the following command will test these organized sets IP addresses for aliases:

```
sc_radargun -a set-2.txt -o set-2.warts -p 20 -O rows
```

To use data previously collected with **sc_radargun** and stored in set-2.warts, to infer likely aliases, reported in pairs:

```
sc_radargun -d 1 set-2.warts
```

SEE ALSO

A. Bender, R. Sherwood, and N. Spring, *Fixing Ally's growing pains with velocity modeling*. scamper(1), sc_ally(1), sc_wartsdump(1), sc_warts2json(1)

AUTHORS

sc_radargun was written by Matthew Luckie <mjl@luckie.org.nz>, but the original implementation was by Bender et al.