NAME

sc_tbitpmtud — scamper driver to test systems for responsiveness to ICMP packet too big messages

SYNOPSIS

sc_tbitpmtud [**-r**] [**-a** *address-file*] [**-c** *completed-file*] [**-l** *limit-per-file*] [**-m** *maximum-transmission-unit*] [**-o** *output-file*] [**-O** *options*] [**-p** *scamper-port*] [**-t** *log-file*] [**-w** *window*]

sc_tbitpmtud [**-d** *dump-id*] [**-A** *ip2as-file*] [**-m** *maximum-transmission-unit*] [*file ...*]

DESCRIPTION

The **sc_tbitpmtud** utility provides the ability to connect to a running scamper(1) instance and use that instance to test end systems for their ability to perform Path MTU Discovery, with the output written to a file in warts format. **sc_tbitpmtud** first tests a given system for responsiveness to ICMP echo packets, and then tests the given system's TCP stack response to ICMP packet too big messages.

The options are as follows:

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

-a *address-file*
specifies the name of the input file which consists of a sequence of systems to test, one system per line.

-A *ip2as-file*
specifies the name of a file which consists of a mapping of prefixes to ASes, one prefix per line.

-c *completed-file*
specifies the name of a file to record IP addresses that have been tested.

-d *dump-id*
specifies the dump ID to use to analyze the collected data. Currently, ID values 1 (mssresults) and 2 (asnresults) are valid, which report PMTUD behaviour according to the server's MSS or the server's origin ASN.

-l *limit-per-file*
specifies the number of tbit objects to record per warts file, before opening a new file and placing new objects.

-m *maximum transmission unit*
specifies the pseudo maximum transmission unit to use. The available choices are 0, 256, 576, 1280, 1480. If 0 is chosen, **sc_tbitpmtud** will test each website with all available MTU choices in decreasing size. The default MTU value tested is 1280.

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

-O *options*
allows the behavior of **sc_tbitpmtud** to be further tailored. The current choices for this option are:
- **gz:** compress the warts output using gzip compression.
- **warts.gz:** compress the warts output using gzip compression.
- **bz2:** compress the warts output using bzip2 compression.
- **warts.bz2:** compress the warts output using bzip2 compression.

– **xz:** compress the warts output using xz compression.
– **warts.xz:** compress the warts output using xz compression.

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

-r shuffle the order in which websites are tested.

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

-w *window-size*
specifies the maximum number of tests to conduct in parallel. The window size value depends on the value of the -F parameter passed to the `scamper`(1) instance.

EXAMPLES

Use of this driver requires a `scamper`(1) instance listening on a port for commands with a configured firewall. The following invocation uses `ipfw`(8) firewall rules 1 to 100, with a corresponding window size of 100, and an unrestricted packets per second rate, as follows:

```
scamper -P 31337 -F ipfw:1-100 -w 100 -p 0
```

To test a set of web servers specified in a file named webservers.txt and formatted as follows:

```
1,example.com 5063 192.0.2.1 http://www.example.com/
1,example.com 5063 2001:DB8::1 http://www.example.com/
1,example.com 5063 2001:DB8::2 https://www.example.com/
```

the following command will test all servers for responsiveness to ICMP packet too big messages and record raw data into webservers_00.warts, webservers_01.warts, etc:

```
sc_tbitpmtud -a webservers.txt -p 31337 -o webservers
```

The webservers.txt file is required to be formatted as above. The format is: numeric ID to pass to tbit, a label for the webserver, the size of the object to be fetched, the IP address to contact, and the URL to use.

To characterize PMTUD behaviour according to the server's advertised MSS value:

```
sc_tbitpmtud -d mssresults webservers_*.warts
```

Given files with IPv4 prefixes in prefix2as4.txt and IPv6 prefixes in prefix2as6.txt formatted as follows:

```
2001:DB8::    48   64496
2001:DB8:1::  48   64497
192.0.2.0     24   64498
```

the following command will characterize PMTUD behaviour according to the origin ASN of the server:

```
sc_tbitpmtud -d asnresults -A prefix2as4.txt -A prefix2as6.txt
webservers_*.warts
```

SEE ALSO

`scamper`(1), `sc_wartsdump`(1), `sc_warts2json`(1), `warts`(5),

M. Luckie and B. Stasiewicz, *Measuring Path MTU Discovery Behaviour*, Proc. ACM/SIGCOMM Internet Measurement Conference 2010.

A. Medina, M. Allman, and S. Floyd, *Measuring Interactions between Transport Protocols and Middleboxes*, Proc. ACM/SIGCOMM Internet Measurement Conference 2004.

AUTHORS

sc_tbitpmtud was written by Matthew Luckie <mjl@luckie.org.nz>. Ben Stasiewicz contributed an initial implementation of the Path MTU Discovery TBIT test to scamper, building on the work of Medina et al.