

**NAME**

traffic\_replay - Samba traffic generation tool.

**SYNOPSIS**

```
traffic_replay [-F, --fixed-password <test-password>] [-T, --packets-per-second <number>]
  [-S, --scale-traffic <scale by factor>] [-r, --replay-rate <scale by factor>] [-D, --duration <seconds>]
  [--traffic-summary <output file>] [-I, --instance-id <id>] [-K, --prefer-kerberos]
  [-B, --badpassword-frequency <frequency>] [--dns-rate <rate>] [-t, --timing-data <file>]
  [--random-seed <seed>] [-U, --username user] [--password <password>]
  [-W --workgroup <workgroup>] [--realm <realm>] [-s, --config-file <file>]
  [-k, --kerberos <kerberos>] [--ipaddress <address>] [-P, --machine-pass] [--option <option>]
  [-d, --debuglevel <debug level>] [--conversation-persistence <0-1>] [--latency-timeout <seconds>]
  [--stop-on-any-error] {summary-file} {dns-hostname}
```

```
traffic_replay [-G, --generate-users-only] [-F, --fixed-password <test-password>]
  [-n, --number-of-users <total users>] [--number-of-groups <total groups>]
  [--average-groups-per-user <average number>] [--group-memberships <total memberships>]
  [--max-members <group size>] {dns-hostname}
```

```
traffic_replay {-c|--clean-up} {dns-hostname}
```

```
traffic_replay [-h, --help] [-V, --version]
```

**DESCRIPTION**

This tool is part of the **samba(7)** suite.

This tool generates traffic in order to measure the performance of a Samba DC, and to test how well Samba will scale as a network increases in size. It can simulate multiple different hosts making multiple different types of requests to a DC.

This tool is intended to run against a dedicated test DC (rather than a live DC that is handling real network traffic).

Note that a side-effect of running this tool is that user accounts will be created on the DC, in order to test various Samba operations. As creating accounts can be very time-consuming, these users will remain on the DC by default. To remove these accounts, use the `--clean-up` option.

**OPTIONS**

```
-h|--help
```

Print a summary of command line options.

#### summary-file

File containing the network traffic to replay. This should be a traffic-model (generated by traffic\_learner). Based on this file, this tool will generate 'conversations' which represent Samba activity between a network host and the DC.

#### dns-hostname

The full DNS hostname of the DC that's being tested. The Samba activity in the summary-file will be replicated and directed at this DC. It's recommended that you use a dedicated DC for testing and don't try to run this tool against a DC that's processing live network traffic.

#### -F|--fixed-password <test-password>

Test users are created when this tool is run, so that actual Samba activity, such as authorizing users, can be mimicked. This option specifies the password that will be used for any test users that are created.

Note that any users created by this tool will remain on the DC until you run the --clean-up option. Therefore, the fixed-password option needs to be the same each time the tool is run, otherwise the test users won't authenticate correctly.

#### random-seed

A number to seed the random number generator with. When traffic is generated from a model-file, use this option to keep the traffic consistent across multiple test runs. This allows you to compare the performance of Samba between different releases.

#### Traffic Model Options

When the summary-file is a traffic-model (produced by traffic\_learner), use these options to alter the traffic that gets generated.

#### -D|--duration <seconds>

Specifies the approximate duration in seconds to generate traffic for. The default is 60 seconds.

#### -T|--packets-per-second <number>

Generate this many packets per second, regardless of the traffic rate of the sample on which the model was based. This cannot be used with -S.

#### -S|--scale-traffic <factor>

Increases the number of conversations by this factor, relative to the original traffic sample on which the model was based. This option won't affect the rate at which packets get sent (which is still based on the traffic model), but it will mean more conversations get replayed.

It cannot be combined with **-T**, which sets the traffic rate in a different way.

**-r|--replay-rate <factor>**

Replays the traffic faster by this factor. This option won't affect the number of packets sent, but it will squeeze them into fewer conversations, which may reduce resource usage.

**--traffic-summary <output-file>**

Instead of replaying a traffic-model, this option generates a traffic-summary file based on what traffic would be sent. Using a traffic-model allows you to scale the packet rate and number of packets sent. However, using a traffic-model introduces some randomness into the traffic generation. So running the same traffic\_replay command multiple times using a model file may result in some differences in the actual traffic sent. However, running the same traffic\_replay command multiple times with a traffic-summary file will always result in the same traffic being sent.

For taking performance measurements over several test runs, it's recommended to use this option and replay the traffic from a traffic-summary file, or to use the **--random-seed** option.

**--stop-on-any-error**

Any client error causes the whole run to stop.

**--conversation-persistence <0-1>**

Conversation termination (as decided by the model) is re-interpreted as a long pause with this probability.

**--latency-timeout <seconds>**

Wait this long at the end of the run for outstanding reply packets. The number of conversations that have not finished at the end of the timeout is a failure metric.

**--generate-users-only**

Add extra user/groups on the DC to increase the DB size. By default, this tool automatically creates test users that map to the traffic conversations being generated. This option allows extra users to be created on top of this. Note that these extra users may not actually be used for traffic generation - the traffic generation is still based on the number of conversations from the model/summary file.

Generating a large number of users can take a long time, so this option allows this to be done only once.

Note that the users created will remain on the DC until the tool is run with the **--clean-up** option.

This means that it is best to only assign group memberships once, i.e. run `--clean-up` before assigning a different allocation of group memberships.

`-n|--number-of-users <total-users>`

Specifies the total number of test users to create (excluding any machine accounts required for the traffic). Note that these extra users simply populate the DC's DB - the actual user traffic generated is still based on the summary-file.

`--number-of-groups <total-groups>`

Creates the specified number of groups, for assigning the test users to. Note that users are not automatically assigned to groups - use either `--average-groups-per-user` or `--group-memberships` to do this.

`--average-groups-per-user <average-groups>`

Randomly assigns the test users to the test groups created. The group memberships are distributed so that the overall average groups that a user is member of matches this number. Some users will belong to more groups and some users will belong to fewer groups. This option is incompatible with the `--group-membership` option.

`--group-memberships <total-memberships>`

Randomly assigns the test users to the test groups created. The group memberships are distributed so that the total groups that a user is member of, across all users, matches this number. For example, with 100 users and 10 groups, `--group-memberships=300` would assign a user to 3 groups on average. Some users will belong to more groups and some users will belong to fewer groups, but the total of all member linked attributes would be 300. This option is incompatible with the `--average-groups-per-user` option.

`--max-members <group size>`

Limit the largest group to this size, even if the other group options would have it otherwise.

`--clean-up`

Cleans up any users and groups that were created by previously running this tool. It is recommended you always clean up after running the tool.

`-I|--instance-id <id>`

Use this option to run multiple instances of the tool on the same DC at the same time. This adds a prefix to the test users generated to keep them separate on the DC.

`-K|--prefer-kerberos`

Use Kerberos to authenticate the test users.

**-B|--badpassword-frequency** <frequency>

Use this option to simulate users trying to authenticate with an incorrect password.

**--dns-rate** <rate>

Increase the rate at which DNS packets get sent.

**-t|--timing-data** <file>

This writes extra timing data to the file specified. This is mostly used for reporting options, such as generating graphs.

### Samba Common Options

**-d|--debuglevel=level**

*level* is an integer from 0 to 10. The default value if this parameter is not specified is 1.

The higher this value, the more detail will be logged to the log files about the activities of the server. At level 0, only critical errors and serious warnings will be logged. Level 1 is a reasonable level for day-to-day running - it generates a small amount of information about operations carried out.

Levels above 1 will generate considerable amounts of log data, and should only be used when investigating a problem. Levels above 3 are designed for use only by developers and generate HUGE amounts of log data, most of which is extremely cryptic.

Note that specifying this parameter here will override the **log level** parameter in the smb.conf file.

**-s|--configfile=<configuration file>**

The file specified contains the configuration details required by the server. The information in this file includes server-specific information such as what printcap file to use, as well as descriptions of all the services that the server is to provide. See smb.conf for more information. The default configuration file name is determined at compile time.

**--option=<name>=<value>**

Set the **smb.conf(5)** option "<name>" to value "<value>" from the command line. This overrides compiled-in defaults and options read from the configuration file.

**--realm=REALM**

Set the realm name

**-V|--version**

Prints the program version number.

### Credential Options

**--simple-bind-dn=DN**

DN to use for a simple bind

**--password=PASSWORD**

Password

**-U USERNAME|--username=USERNAME**

Username

**-W WORKGROUP|--workgroup=WORKGROUP**

Workgroup

**-k|--kerberos**

Try to authenticate with kerberos. Only useful in an Active Directory environment.

**--ipaddress=IPADDRESS**

IP address of the server

**-P|--machine-pass**

Use stored machine account password.

## OPERATIONS

### Generating a traffic-summary file

To use this tool, you need either a traffic-summary file or a traffic-model file. To generate either of these files, you will need a packet capture of actual Samba activity on your network.

Use Wireshark to take a packet capture on your network of the traffic you want to generate. For example, if you want to simulate lots of users logging on, then take a capture at 8:30am when users are logging in.

Next, you need to convert your packet capture into a traffic summary file, using `traffic_summary.pl`.

Basically this removes any sensitive information from the capture and summarizes what type of packet was sent and when.

Refer to the `traffic_summary.pl --help` help for more details, but the basic command will look something like:

```
tshark -r capture.pcapng -T pdml | traffic_summary.pl > traffic-summary.txt
```

### Replaying a traffic-summary file

Once you have a traffic-summary file, you can use it to generate traffic. The `traffic_replay` tool gets passed the traffic-summary file, along with the full DNS hostname of the DC being tested. You also need to provide some user credentials, and possibly the Samba realm and workgroup (although the realm and workgroup may be determined automatically, for example from the `/etc/smb.conf` file, if one is present). E.g.

```
traffic_replay traffic-summary.txt my-dc.samdom.example.com -UAdmin%password -W samdom --realm=samdom.example.com --fixed-password=blahblah123!
```

This simply regenerates Samba activity seen in the traffic summary. The traffic is grouped into 'conversations' between a host and the DC. A user and machine account is created on the DC for each conversation, in order to allow logon and other operations to succeed. The script generates the same types of packets as those seen in the summary.

Creating users can be quite a time-consuming process, especially if a lot of conversations are being generated. To save time, the test users remain on the DC by default. You will need to run the `--clean-up` option to remove them, once you have finished generating traffic. Because the same test users are used across multiple runs of the tool, a consistent password for these users needs to be used - this is specified by the `--fixed-password` option.

The benefit of this tool over simply using `tcprelay` is that the traffic generated is independent of any specific network. No setup is needed beforehand on the test DC. The traffic no longer contains sensitive details, so the traffic summary could be potentially shared with other Samba developers.

However, replaying a traffic-summary directly is somewhat limited in what you can actually do. A more flexible approach is to generate the traffic using a model file.

### Generating a traffic-model file

To create a traffic-model file, simply pass the traffic-summary file to the `traffic_learner` script. E.g.

```
traffic_learner traffic-summary.txt -o traffic-model.txt
```

This generates a model of the Samba activity in your network. This model-file can now be used to generate traffic.

### Replaying the traffic-model file

Packet generation using a traffic-model file uses the same command as a traffic-summary file, e.g.

```
traffic_replay traffic-model.txt my-dc.samdom.example.com -UAdmin%password
```

By default, this will generate 60 seconds worth of traffic based on the model. You can specify longer using the `--duration` parameter.

The traffic generated is an approximation of what was seen in the network capture. The traffic generation involves some randomness, so running the same command multiple times may result in slightly different traffic being generated (although you can avoid this, by specifying the `--random-seed` option).

As well as changing how long the model runs for, you can also change how many conversations get generated and how fast the traffic gets replayed. To roughly double the number of conversations that get replayed, use `--scale-traffic=2` or to approximately halve the number use `--scale-traffic=0.5`. To approximately double how quickly the conversations get replayed, use `--replay-rate=2`, or to halve this use `--replay-rate=0.5`

For example, to generate approximately 10 times the amount of traffic seen over a two-minute period (based on the network capture), use:

```
traffic_replay traffic-model.txt my-dc.samdom.example.com -UAdmin%password  
--fixed-password=blahblah123! --scale-traffic=10 --duration=120
```

### Scaling the number of users

The performance of a Samba DC running a small subset of test users will be different to a fully-populated Samba DC running in a network. As the number of users increases, the size of the DB increases, and a very large DB will perform worse than a smaller DB.

To increase the size of the Samba DB, this tool can also create extra users and groups. These extra users are basically 'filler' for the DB. They won't actually be used to generate traffic, but they may slow down authentication of the test users.

For example, to populate the DB with an extra 5000 users (note this will take a while), use the command:



```
traffic_replay my-dc.samdom.example.com -UAdmin%password --generate-users-only
--fixed-password=blahblah123! --number-of-users=5000
```

You can also create groups and assign users to groups. The users can be randomly assigned to groups - this includes any extra users created as well as the users that map to conversations. Use either `--average-groups-per-user` or `--group-memberships` to specify how many group memberships should be assigned to the test users.

For example, to assign the users in the replayed conversations into 10 groups on average, use a command like:

```
traffic_replay traffic-model.txt my-dc.samdom.example.com -UAdmin%password
--fixed-password=blahblah123! --generate-users-only --number-of-groups=25
--average-groups-per-user=10
```

The users created by the test will have names like STGU-0-xyz. The groups generated have names like STGG-0-xyz.

## VERSION

This man page is complete for version 4.13.17 of the Samba suite.

## SEE ALSO

**traffic\_learner**(7).

## AUTHOR

The original Samba software and related utilities were created by Andrew Tridgell. Samba is now developed by the Samba Team as an Open Source project similar to the way the Linux kernel is developed.

The `traffic_replay` tool was developed by the Samba team at Catalyst IT Ltd.

The `traffic_replay` manpage was written by Tim Beale.