Dante Session Module Documentation

Inferno Nettverk A/S
Oslo Research Park
Gaustadalleen 21
NO-0349 Oslo
Norway

Date: 2010/09/07 13:18:50

1 Description

The session module makes it possible to set an upper limit on the number of simultaneous connections users will be allowed to create. The *Dante* server distributes the available sessions on a first come, first served basis.

This module can be used to e.g., limit the number of sessions going to certain servers. When the session limit has been reached, future clients will not be able to create new sessions until some of the existing sessions have ended.

The *Dante* server first accepts any connection in order determine the IP-address and other credentials of the client, and then immediately closes connections for which a matching session-limit has been reached.

Clients may re-try again later, and if one of the old clients has finished in the meantime, the new client will be granted access.

2 Syntax

The syntax is as follows:

`maxsessions: <sessions>`

`sessions` is the maximum number of sessions that can be active at any time.

3 Semantics

The `maxsessions` statement is typically used as a part of the *socks-rules*, but it can also be used in *client-rules*. See *sockd.conf(5)* for more information about the different rule types.

4 Examples

This section gives some examples on how the module can be used.

4.1 Limiting the number of negotiating clients

The rule below shows how to limit to ten the number of clients on the 10.0.0.0/24 network that can be in the SOCKS protocol negotiation phase at any time.

```
client-pass {
   from: 10.0.0.0/24 to: 0.0.0.0/0
   maxsessions: 10
}
```

Note that since this is a client-rule, it only limits the clients while they are doing SOCKS protocol negotiation. It enforces no limitations on the number of clients active after the negotiation has completed.

4.2 Limiting the number of clients using the web

The next rule shows how one can limit the number of concurrent clients accessing the web to 10.

```
pass {
   from: 0.0.0.0/0 to: 0.0.0.0/0 port http
   maxsessions: 100
}
```

Since this is a socks-rule, it takes action when the clients have finished SOCKS protocol negotiation with the *Dante* server.

The following example shows how one could limit a particular user, in this example, the user ”monica”, to one FTP-session at a time.

```
# first, limit monica to one FTP session
pass {
   from: 0.0.0.0/0 to: 0.0.0.0/0 port ftp
   maxsessions: 1
   user: monica
}

# then a rule with no limits for all other users.  Remember,
# first match is what is used for matching rules.
pass {
   from: 0.0.0.0/0 to: 0.0.0.0/0 port ftp
}
```

5 Special notes

Sending the *Dante* server a `SIGHUP` signal forces a reload of the configuration file. It should be noted that this does not affect current sessions.

That is, a reload of the configuration file does not let sessions created before the reload affect sessions created after the reload.

Changing e.g., a *pass* statement to a *block* statement, does not terminate the session of any existing client.

This means that after a reload of the configuration file, the session counter for new sessions will be reset, and will only apply to new connections. The old sessions will remain until they finish, meaning that the total number of sessions might temporarily exceed the session limit until all the old sessions have ended.