Dante Bandwidth 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 *Bandwidth* module gives control over how much bandwidth the clients of a *Dante* SOCKS serve can consume.

The module can be used to limit bandwidth to non-work related web/FTP sites, or to prevent FTP-related traffic from impacting too much on interactive telnet/ssh traffic.

It can also be used to give more bandwidth to certain clients or for traffic to certain sites.

When combined with the *Dante bind extension*, the module can be used to provide bandwidth control for network servers (like e.g., web servers) that do not have support for bandwidth control.

2 Syntax

The syntax of the `bandwidth` statement is as follows:

`bandwidth: <bytes>`

`bytes` is the maximum bandwidth to use per second, measured in bytes.

3 Semantics

The `bandwidth` statement can be used in both the Dante client-rules and socks-rules. See *sockd.conf(5)* for more information about the different rule types.

Note that a bandwidth limitation set in a client-rule *is inherited by any* socks-rule *also matching the client.*

The maximum allowed `bandwidth` set for a rule will be shared by all clients matching that rule. The *Dante* server will attempt to distribute the bandwidth to the matching clients in a least-recently used fashion, trying to let all clients get a fair share.

Note that for UDP, as with TCP, the bandwidth setting is based on the rule matching the control-connection, not on each individual UDP packet.

4 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 or limits placed on them.

Changing e.g., a *pass* statement to a *block* statement, does not terminate the session of any existing client. Likewise, a reload of the configuration file does not let sessions created before the reload affect sessions created after the reload.

This means that after a reload of the configuration file, the bandwidth counter for new sessions will be reset, and will only apply to new sessions. The old sessions will remain until they finish normally. The amount of bandwidth used might at this point be higher than otherwise expected, until all the old existing sessions have ended.

5 Examples

This section shows several examples of how the *bandwidth* module can be used.

5.1 Limiting web/http bandwidth

The below rule shows how to limit the bandwidth used for web traffic, by the clients on the 10.0.0.0/24 network, to a total of 10240 bytes (10 KiloBytes/second).

```
pass {
   from: 10.0.0.0/24 to: 0.0.0.0/0 port = http
   command: connect
   bandwidth: 102400
}
```

5.2 Increasing web/http bandwidth

The next rule, if placed before other bandwidth-limiting rules, shows how one can increase the bandwidth used for web traffic from the clients on the 10.0.0.0/24 network to a specified host.

In this case, the clients will be able to use 1024000 bytes (one MegaByte/second), when getting data from the host *work.example.com*.

```
pass {
   from: 10.0.0.0/24 to: work.example.com port = http
   command: connect
   bandwidth: 1024000
}
```

5.3 Limiting FTP bandwidth

The next rule shows how one can limit the bandwidth used for FTP data transfers for the clients on the 10.0.0.0/24 network to a total of 10240 bytes (10 KiloBytes/second).

This only works for *active* FTP, since for *passive* FTP there are no fixed port numbers.

```
pass {
   from: 0.0.0.0/0 port = ftp-data to: 10.0.0.0/24
   command: bindreply
   bandwidth: 10240
}
```

5.4 Limiting bandwidth provided by internal servers to the outside

The next rule shows how one could use the *Dante bind extension* together with the *Bandwidth* module to limit the amount of data provided by a internal server, in this case, a web server called *our-webserver.example.com*, to a total of 10240 bytes, or 10 KiloBytes/second.

This requires the webserver to be socksified and the *bind extension* to be enabled on both the socksified client and on the *Dante* server.

```
pass {
   from: 0.0.0.0/0 to: our-webserver.example.com port = http
   command: bindreply
   bandwidth: 10240
}
```