

## Dante\_Socks\_Server

When working on a private LAN with no direct internet access available, squid is normally used a proxy server for internet and ftp traffic. SSH or scp (or other traffic) however, will require a SOCKS server in order to reach the internet.

Dante socks server provides just that.

## Contents

- [1 How to install dante socks server on debian \(ubuntu is similar\):](#)
- [2 To configure Dante socks server:](#)
- [3 Test Dante Socks Server](#)
- [4 Other SOCKS Servers](#)

## How to install dante socks server on debian (ubuntu is similar):

```
apt-get install dante-server
```

This installs dante onto your debian server. By default it will not start (it will try to start - but will fail saying "no internal interfaces are configured etc...").

## To configure Dante socks server:

```
vi /etc/danted.conf
```

By default - most required lines will be uncommented. The following lines need to be inserted/ uncommented in danted.conf

```
#logoutput: stderr
logoutput: syslog
#the above line will send any logs to /var/log/syslog instead to a terminal

internal: eth0 port = 1080
internal: 127.0.0.1 port = 1080

external: eth0

method: username none
#the above puts no username or password. Access will instead be controlled via client ip address/range
#if there is no username or password - then danted socks server needs to run as nobody, i.e.

#method: pam
# if you choose to use pam instead - a valid username&password as required for sshing to the socks server
#Note: not sure if proxy login details are sent in clear text.

#user.privileged: proxy
user.notprivileged: nobody

client pass {
    from: 136.201.251.21/0 port 1-65535 to: 0.0.0.0/0
    # 136.201.251.21/0 = specific ip address.
```

## Dante\_Socks\_Server

```
}  
  
client pass {  
    from: 127.0.0.0/8 port 1-65535 to: 0.0.0.0/0  
}  
  
client block {  
    from: 0.0.0.0/0 to: 0.0.0.0/0  
    log: connect error  
}  
  
#Finally block other traffic  
block {  
    from: 0.0.0.0/0 to: 127.0.0.0/8  
    log: connect error  
}  
  
pass {  
    from: 136.201.251.21/0 to: 0.0.0.0/0  
    protocol: tcp udp  
}  
  
pass {  
    from: 127.0.0.0/8 to: 0.0.0.0/0  
    protocol: tcp udp  
}  
  
block {  
    from: 0.0.0.0/0 to: 0.0.0.0/0  
    log: connect error  
}
```

Once the config is complete. Start/Restart dante socks server:

```
/etc/init.d/danted start
```

If there is a problem with the config - it will tell you immediately on trying to start the danted daemon.

## Test Dante Socks Server

```
netstat -n -a  
#check to see if server is listening on 1080
```

Make sure the firewall is open. Add appropriate rule as on [Iptables Firewall](#). Test also with winscp or putty.

## Other SOCKS Servers

Note that putty itself can provide a SOCKS 5 server!! (But putty needs to be able to connect to an external server/computer firstly!)

**To configure socks to act as a socks v5 server:**

```
Open Putty, Go to CONNECTION -> SSH -> TUNNELS  
In the Source Port - put 1080
```

To configure Dante socks server:

## Dante\_Socks\_Server

In the Destination Port - put 1080

Click on the **Dynamic** radio button for "Dynamic Port Forwarding"

Connect to an server with external access. Telnet localhost 1080 and it should connect.

Also note - SOCKS v5 can be setup using ssh on the command line.

```
ssh user@server.com -D 1080  
# -D is for Dynamic Port Forwarding.
```

---

More info on Dante's Config at:

[http://trekweb.com/~jasonb/articles/dante\\_tunnel.shtml](http://trekweb.com/~jasonb/articles/dante_tunnel.shtml)