PERMANENT SSH TUNNELS

If you have gone through any of the article listings on this site, you will know I am a fan of *ssh* and all the tricks you can pull with it. One of those is the setup of a VPN-like tunnel. This is very useful in many circumstances - but I am not going to rehash all of that now, rather I am going to talk about a handy little utility that allows you to setup the tunnel so that it automatically reconnects in case of it dropping. *Autossh* is that utility and you can find it here. As I always say, all good software is small, simple and does what it says it will with a minimum of fuss, and autossh is another fine example.

Needed
You need a linux machine, the ssh server and the download *autossh* source code. Unextract the code, change in the extracted folder and then do the normal 3-step dance of..

```
./configure && make && make install
```

Running
Ok, *autossh* is installed now you want to use it, lets take a look at a typical command line..

```
/usr/local/bin/autossh -M 0 -vv -f -N -R 2266:127.0.0.1:8080 root@192.168.0.12
```

lets break that down into the sections..

`/usr/local/bin/autossh`	the actual autossh binary
`-M`	turns monitoring off and only restarts is ssh restarts or exits
`-vv`	verbose
`-f`	drop into background
`-N -R 2266:127.0.0.1:8080 root@192.168.0.12`	normal ssh options for forwarding local port 8080 to 192.168.0.12 port 2266

If you are using ssh tunnels for a more permanent solution you could put that command line into your startup script (something like *rc.local*).

Final Words
Like I said- small, easy and very useful. Tunneling through ssh is never going to replace a proper VPN setup, but it can be used for smaller projects or needs and having *autossh* keeping things alive means one less thing to worry about. As always, fun fun and learn.