SSHHOP

It is no secret I am a great fan of ssh, and how we can use it to do all sorts of things securely. So when I find something that I think makes it even a little better, I am always glad. I do a lot of work through gateway servers and servers in different sites, and it can sometimes get a little bothersome always re-entering each command as you bounce from server to server. Thats where sshhop comes in. This is a patch to normal ssh put out by some MIT chaps (thanks guys) which really helps with the one narrow but bothersome problem.

Head to the site (here) to get it, and once you have it do the normal dance - *configure*, *make*. One gotcha though, on my system the *make* failed and it seems this could be a common problem. But no worries, edit the *sshhop.c* file, look for this section..

```
/* External variables used for command line option parsing in getopt. */
extern int optind;
extern int optreset;
```

And make it look like this..

```
/* External variables used for command line option parsing in getopt. */
extern int optind;
extern int optreset;
extern char *optarg;
```

That should solve the make problem. Now sharp-eyed readers will note that there was no *make install* step. You see after the *make* command you have the *sshhop_test* binary. You need to copy that into your relevant directory - I just put it in the same folder as my other ssh binaries. Now for the fun part.

To use this is actually simple. This is just a modified normal ssh binary with a new option - the *-H* option. Each *-H* option you specify is a hop through a ssh server. For example, I am at point A and need to get to point C. To get to point C I need to login to point B and then from here login to point C. Now I could always manually do this, or..

```
# sshhop -H bob@192.168.6.106 -H root@192.168.4.105
[21799]
[21799] login for bob@192.168.6.106
[21799] bob@192.168.6.106's password:
[21801]
[21801] login for root@192.168.4.105
[21801] root@192.168.4.105's password:
Last login: from 192.168.6.106
[root@localhost ~]#
```

I think that is very cool. And it even tidies up nicely when you finish up. When you are finished and you close the terminal..

```
[root@localhost ~]# logout

Connection to localhost closed.
process 21801 exited (host 192.168.2.104)
process 21799 exited (host 192.168.2.108)
```

It closes down each hop, until you are back where you started. Linux is full of useful commands that do one thing and do it well. I think this little tool fits that mentality quite well. Give it a whirl and see what you think. Have fun and learn.