SOME QUICK SAFETY TIPS

If you are anything like me you are always on the lookout for just another way to be a little bit more paranoid. No? Well maybe it's just me then. Regardless, Here are three quick and simple things you can do make your network just that little bit more draconian..

OpenDNS
OpenDNS (see here) is a setup that has a very large DNS server setup. They allow anyone to use their DNS servers for free. They do filtering on some results, and even allow you to block certain classes of sites. All for free, and fairly fast as well. So how does one go about using this service? Very simple, take the following two ip addresses..

```
208.67.222.222
208.67.220.220
```

And use them to in your *resolv.conf,* or network settings, or to replace your *forwarder* addresses, whatever. Replace your old dns server addresses with these, and then ... actually thats all. Now thats what I call an easy setup.

RBN Blocking
RBN stands for Russian Business Network. There was a very good paper done (see here - yes I know the site is in french, but use google to translate or something and the pdf is in english), and simply put it is a Russian ISP pretty much dedicated to serving up all the web nastiness people seem able to do - porn, spam, malware, phishing, bots, etc. A bad place. One of the protections is to block the whole area. Think of it has walling of the neighborhood and firebombing it. Anyway, the paper has some cisco type acls you could use, here are my iptables adaptations..

```
echo ">rbn checks "
$IPT -N RBN_CHK
for RBN in 81.95.144.0/12 194.146.204.0/14 195.114.8.0/15 80.70.224.0/12 81.84.16.0/12
193.238.36.0/14 193.93.232.0/14 195.64.162.0/15 195.114.8.0/15 195.114.16.0/15
85.249.20.0/14 85.249.128.0/12
 do
  $IPT -A RBN_CHK -s $RBN -j DROP
  $IPT -A RBN_CHK -d $RBN -j DROP
 done
$IPT -A RBN_CHK -j RETURN
$IPT -A INPUT -j RBN_CHK
$IPT -A FORWARD -j RBN_CHK
```

Dump that near the top of your ruleset, near after the stateful packet inspection ruleset. If you want to, put so log rules in so you can see who is trying to get to the RBN or trying to come from the RBN. So far, I have had no false positives.

BLEEDING EDGE Firewall Rules
The bleeding edge (see here) has a sub-project where they take reports from a couple of places (like *dshield*) and compile them into firewall rules. This allows you to periodically get these files and add them into your ruleset. They have a bunch of different formats, but of course I wanted iptables. My little "*go and get*" script looks something like..

```
/usr/bin/wget -q -N -nd http://www.bleedingthreats.net/fwrules/bleeding-edge-IPTABLES-
ALL.rules
```

Short and sweet! Once you have got it down you can make any changes you want or just run it. Once again, I have had no hassles with using these rules.

Final Words

Three easy ways to keep a few more bad guys out of the network, these type of tweaks are always useful and can actually be very helpful, especially for a large organisation. So, go play and -as always- have fun and learn.