FUN WITH DHCP

The DHCP protocol is very useful to system administrators. It allows us to setup a single network configuration file, and then have the clients on the network come and fetch it. But sometimes we have problems. What about when an unauthorized person plugs their machine into our network and they also just get the information ? Or what about when we want certain people/devices to have a certain IP address? Well, fear not, the Internet Software Consortium DHCP -the de-facto standard on linux boxes- can help.

A Basic Setup
DHCP runs off of one configuration file, */etc/dhcpd.conf*. A basic setup would look like (of course all IP addresses and MAC addresses are entirely fictional)..

```
ddns-update-style ad-hoc;
authoritative;
option netbios-name-servers 10.0.0.10;
option domain-name-servers 10.0.0.11;
option subnet-mask 255.255.255.0;
option routers 10.0.0.1;

subnet 10.0.0.0 netmask 255.255.255.0 {
      one-lease-per-client;
     max-lease-time 28800;
     range 10.0.0.100 10.0.0.200;
     authoritative;
    }
```

Simple. The first couple of lines give the global options for DHCP (the nameservers, the gateways/routers, etc). The *subnet* section gives the range of addresses the server gives out, how long a lease lasts and says that a client can only have one IP. Most networks rarely get more complicated then this. But what if you want certain devices to have specific addresses...

Static IP
Now lets change our setup to make sure that certain machines get certain IP addresses. The first thing you will need is those machine's network card MAC addresses. Then change your configuration to look like..

```
ddns-update-style ad-hoc;
authoritative;
option netbios-name-servers 10.0.0.10;
option domain-name-servers 10.0.0.11;
option subnet-mask 255.255.255.0;
option routers 10.0.0.1;

subnet 10.0.0.0 netmask 255.255.255.0 {
      one-lease-per-client;
     max-lease-time 28800;
     range 10.0.0.100 10.0.0.200;
     authoritative;
    }

host temp-1 { hardware ethernet 11:11:11:11:11:11; fixed-address 10.0.0.101;
}
host temp-2 { hardware ethernet 22:22:22:22:22:22; fixed-address 10.0.0.102;
}
host temp-3 { hardware ethernet 33:33:33:33:33:33; fixed-address 10.0.0.103;
}
```

As you can see, the last three lines specify the machine name, the MAC address and the then the IP address it must get. This is great for network attached devices. But what if you want to break up your network users into different groups...

Pooling DHCP clients
Sometimes you want certain groups of users to get certain settings, like different lease time or address ranges, or even if you only want certain MAC addresses to get an IP from the DHCP server. Change your configuration to

look like..

```
ddns-update-style ad-hoc;
authoritative;
option netbios-name-servers 10.0.0.10;
option domain-name-servers 10.0.0.11;
option subnet-mask 255.255.255.0;
option routers 10.0.0.1;

subnet 10.0.0.0 netmask 255.255.255.0 {
      authoritative;
      pool {
              host trusted1 { hardware ethernet 44:44:44:44:44:44; }
              host trusted2 { hardware ethernet 55:55:55:55:55:55; }
              host trusted3 { hardware ethernet 66:66:66:66:66:66; }
              one-lease-per-client true;
              max-lease-time 86400;
              range 10.0.0.100 10.0.0.105;
              deny unknown-clients;
              }
      pool {
              allow unknown-clients;
              one-lease-per-client true;
              max-lease-time 1000;
              range 10.0.0.106 10.0.0.200;
              }
    }

host temp-1 { hardware ethernet 11:11:11:11:11:11; fixed-address 10.0.0.101;
}
host temp-2 { hardware ethernet 22:22:22:22:22:22; fixed-address 10.0.0.102;
}
host temp-3 { hardware ethernet 33:33:33:33:33:33; fixed-address 10.0.0.103;
}
```

Now there are some major changes here. First off, you will notice that the *subnet* section has been divided up into two *pool* sections, each with their own options and address ranges. The first pool section has hosts with specified MAC address and a *deny unknown-clients* statement, this means that only the specified MAC addresses in this *pool* statement can get an address from it's range and use the options specified within it. The second *pool* section has no specified MAC addresses and has an *allow unknown-clients* statement, this means that anyone connecting to the network will use this section's address range and options. What you can do is use different IP classes to separate the different groups. Or to even lock it down totally, once you have all your trusted MAC addresses, just remove the second *pool* section, then any unknown MAC address will not be served by the DHCP server.

Final Words
This is one of those cases where you can make things more secure but you will spend time making sure that it is up to date. Depending on the size of your network this could be either a big or small job. Also be aware that it is possible to spoof MAC addresses in multiple operating systems, so this is not some silver bullet. This is another step in your overall security plan. As always, have fun and learn