

# Building DHCP Automatically

Eric Seigne

## Introduction

The articles aim is to present:

- a static DHCP server (in order to be able to detect new guests on the network such as people connecting their laptop and trying to play pirate's games)
- a minimal loss of time.

The server used for this article is Paul Vixie/ISC DHCPd.

## Building a blank DHCP

Our first step is to create a blank configuration file which will be able to accept all clients on the network.

I am using here the network configuration of the Slash Party #2 for which I tried to configure a proper Linux server.

```
<file dhcpd.conf>
default-lease-time 86400;
max-lease-time 604800;
get-lease-hostnames true;
option subnet-mask 255.255.255.0;
option domain-name "slach2-100.party";
option domain-name-servers 192.168.12.1;
option interface-mtu 1500;

subnet 192.168.12.0 netmask 255.255.255.0
  # default gateway
  option routers 192.168.12.1;
  option broadcast-address 192.168.12.255;
  range 192.168.12.50 192.168.12.200;

</file>
```

Once your DHCP server is running (using `dhcp start` or a similar command according to your Linux distribution), you can start client stations on the network. They should have been configured to obtain automatically an IP address.

As soon as clients ask for an address from the DHCP server, a bloc such as this one will be added to the file `dhcpd.leases`:

```
<bloc attributed to a dhcp client>
lease 192.168.12.58
  starts 2 1999/08/24 06:28:48;
  ends 3 1999/08/25 06:28:48;
  hardware ethernet 00:10:5a:2e:56:a7;
  uid 01:00:10:5a:2e:56:a7;
  client-hostname "KLUSTER";

</bloc>
```

## Building DHCP Automatically

Eric Seigne

So, once all the clients have obtained an IP address from the server, the dhcpd.leases file will look like this:

```
<file dhcpd.leases>
lease 192.168.12.58
    starts 2 1999/08/24 06:28:48;
    ends 3 1999/08/25 06:28:48;
    hardware ethernet 00:10:5a:2e:56:a7;
    uid 01:00:10:5a:2e:56:a7;
    client-hostname "KLUSTER";

lease 192.168.12.53
    starts 2 1999/08/24 05:42:22;
    ends 3 1999/08/25 05:42:22;
    hardware ethernet 00:80:ad:97:e1:76;
    uid 01:00:80:ad:97:e1:76;
    client-hostname "ceddz";

lease 192.168.12.54
    starts 2 1999/08/24 03:07:26;
    ends 3 1999/08/25 03:07:26;
    hardware ethernet 00:80:ad:97:e1:7d;
    uid 01:00:80:ad:97:e1:7d;
    client-hostname "SDS";

lease 192.168.12.67
    starts 2 1999/08/24 02:52:19;
    ends 3 1999/08/25 02:52:19;
    hardware ethernet 00:50:04:45:e1:65;
    uid 01:00:50:04:45:e1:65;
    client-hostname "HOMER";

lease 192.168.12.64
    starts 2 1999/08/24 01:26:05;
    ends 3 1999/08/25 01:26:05;
    hardware ethernet 00:80:ad:97:e2:1c;
    uid 01:00:80:ad:97:e2:1c;
    client-hostname "chAwArmA";

lease 192.168.12.59
    starts 2 1999/08/24 01:14:06;
    ends 3 1999/08/25 01:14:06;
    hardware ethernet 00:00:21:2c:30:e7;
    uid 01:00:00:21:2c:30:e7;
    client-hostname "WOOKIEE";

</file>
```

### Securing the DHCP Configuration

Next, it is necessary to convert our "open" DHCP to a static and more secure one. This is done using the now proper dhcp.lease we just created and converting it to what I call a static dhcp.

# Building DHCP Automatically

Eric Seigne

What are the differences between a static and an open DHCP? As far as I am concerned, an open DHCP allows any computer connected to the network to obtain an IP address and usable network parameters. This is a big security hole, any unauthorized pirate could physically connect himself to the network and obtain good network parameters :( To counter such an attack, I use in the static DHCP. Each IP address is only given to clients with corresponding Mac of the associated Ethernet adapter. It is therefore easier to detect an intrusion.

```
<file dhcpd.conf>
default-lease-time 86400;
max-lease-time 604800;
get-lease-hostnames true;
option subnet-mask 255.255.255.0;
option domain-name "slach2-100.party";
option domain-name-servers 192.168.12.1;
option lpr-servers 192.168.12.1;
option interface-mtu 1500;

subnet 192.168.12.0 netmask 255.255.255.0
    # default gateway
    option routers 192.168.12.1;
    option broadcast-address 192.168.12.255;
    # Those not in the dhcp
    # will get ip between .10 et .50
    range 192.168.12.10 192.168.12.50;

host
    hardware ethernet 00:10:5a:2e:56:a7;
    fixed-address "kluster.slach2-100.party";

host
    hardware ethernet 00:80:ad:97:e1:76;
    fixed-address "ceddz.slach2-100.party";

host
    hardware ethernet 00:80:ad:97:e1:7d;
    fixed-address "sds.slach2-100.party";

host
    hardware ethernet 00:40:95:49:0b:a5;
    fixed-address "saigneur.slach2-100.party";

host
    hardware ethernet 00:50:04:45:e1:65;
    fixed-address "homer.slach2-100.party";

</file>
```

**WARNING:** If you do not have a DNS server running, the file dhcp.conf must use IP addresses and not machine names.

## Building DHCP Automatically

Eric Seigne

```
<extracted from dhcpd.conf whithout dns>
host
    hardware ethernet 00:40:95:49:0b:a5;
    fixed-address "192.168.12.57";

host
    hardware ethernet 00:50:04:45:e1:65;
    fixed-address "192.168.12.67";

</extract>
```

I wrote a small Perl script which converts the file dhcpd.leases into a static dhcp configuration file (see end of article).

### Automatic Construction of the DNS

Why stop along the way when we could configure the DNS server at the same time?

The same Perl script, with option `-dns`, will build files named `.$domaine.ajouter` and named `.$domaine.rev.ajouter` that you should check before adding them to your own dns reverse dns files.

Moreover, it is necessary to fill the `named.conf` file with parameters from your own domain. For instance:

```
<add to file named.conf>
zone "slach2-100.party"
    type master;
    file "named.slach2-100";
;

zone "12.168.192.in-addr.arpa"
    type master;
    file "named.slach2-100.rev";
;
</add>
```

Congratulate yourself, you now have a working dhcp and dns configuration.

### Network Neighborhood

Available for the same price (even better when it is free :), the configuration of "Network Neighborhood" from window machines. Here is the story with a small drawing and explanations: At Slach 2 (A demo party), I installed two networks, one was a 10 Mbits and the other a 100 Mbits (for privileged users at that time...). The major problem was: both networks could not "see" each other though the "Network Neighborhood" of windows.

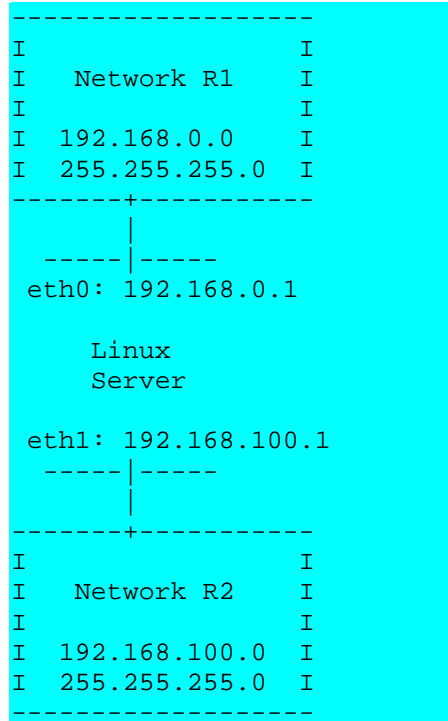
The solution is WINS. WINS allow to declare, on the network, a server which holds a list of links between IPs and "name of several domains". It is a sort of master for networks R1 and R2.

It is therefore necessary to configure a WINS server and clients so that they use it and create an IP gateway between these two networks.

## Building DHCP Automatically

Eric Seigne

Wins is well supported by Samba :)  
Schematic:



dhcpd.conf is modified in order to auto-configure clients using our wins server :). Without dhcp, it would have been necessary to configure each client individually in order to activate their access to the wins server!

**WARNING: DO NOT FORGET TO ADD THE FOLLOWING LINES TO YOUR DHCPD.CONF IN ORDER FOR YOUR WINS SERVER TO BE USED BY CLIENTS**

```
<file dhcpd.conf>  
[...]  
option routers 192.168.0.1;  
option netbios-name-servers 192.168.0.1;  
option netbios-dd-server 192.168.0.1;  
option netbios-node-type 8;  
[...]  
</file>
```

To have a working wins server, I have a working Samba, configured as follows:

```
<file smb.conf>  
; validated with samba 2.0.5  
[global]  
  workgroup = rycks.com  
  server string = Linux Box  
  comment = Linux Box
```

## Building DHCP Automatically

Eric Seigne

```
netbios name = pantoufle
volume = pantoufle
guest only = yes

guest account = nobody
log file = /var/log/samba-log.%m
max log size = 50

share modes = yes
security = share
socket options = TCP_NODELAY
os level = 33

; Configured as network master
domain master = yes
local master = yes
preferred master = yes

; Activate wins support
wins support = yes
wins proxy = yes

; FTP sharing
[ftp]
path = /home/ftp/pub
public = yes
printable = no
guest ok = yes

</file>
```

It is necessary to restart the servers, dhcp, samba and dns if modifications have been made. Restart also dhcp clients so that they take into account the new configuration. Watch for groups on the "other" network in the "network neighborhood".

Given the usual delay windows needs to bring new machines into the "network neighborhood" I suggest you look for a computer using its name. Try to find a computer on the network you are on, then one from the "other" network.

If you can "see" machines from the outside but cannot access them, you need to configure the gateway so that it acts as a real gateway between the two networks.

For more informations about this topic, take a look at ipchains. Usually it is necessary to use something similar to:

```
#Reset chains
ipchains -F
#Activate masquerading
#to check according to default etc.
ipchains -A forward -i eth0 -j MASQ
```

# Building DHCP Automatically

Eric Seigne

## Bugs and Limitations

Be careful, if the dhcp server is used in two areas at the same time, there will only be one dns file on the output from the program... It will then be necessary to sort the file manually.

I hope this document will be useful. Please feel free to contact me for comments or questions.

```
#!/usr/bin/perl
#/******
# * File   : dhcp-auto.pl
# * Author: Eric SEIGNE
# *       : mailto:erics@rycks.com
# *       : http://www.rycks.com/erics/linux/
# * Date   : 26/03/2000
# *
# * Description:
# * -----
# * Convertisseur automatique de dhcpd.leases vers un dhcp fixe
# *
# * 27/03/2000 Modification, si on ne fait pas de dns, il faut mettre
# * les ip dans le dhcp fixe.
# * Sortie des fichiers dns classés par ordre croissant d'ip
# *
# * Attention, si il y a plusieurs réseaux ip dans le fichier leases
# * seul un fichier de dns ne sera créé. Vous devrez ensuite le répartir
# * entre vos différents fichiers dns.
# *
# /******/

use Getopt::Long; #pour le getoptions
use DB_File;

$choix = "";
$dns = 0;
GetOptions ("f:s",\ $fichier,"d:s",\ $domaine,"h",\ $help,"dns",\ $dns);
if($help){
    print "\n\n";
    print "dhcp-auto\t\t\t\t\tUn utilitaire rycks.com\n-----\n";
    print "\t\t\t\t\tterics@rycks.com\n\n";
    print "Constitution du dhcpd.conf à partir du fichier dhcpd.leases\n";
    print "Options:\n";
    print "\t-f [fichier] donne le dhcpd.leases à utiliser\n";
    print "\t-d [domaine] donne le nom de domaine dans lequel on se situe\n";
    print "\t-dns si vous voulez creer les fichiers dns\n";
    print "\t ils porteront les noms named.\$domaine et
named.\$domaine.rev\n";
    print "\t-h affiche cet écran d'aide\n\n";
    exit;
}

#si on a pas de parametre
while ($fichier eq "") {
    print STDERR "Ou est votre fichier dhcpd.leases ?:";
    $fichier = <STDIN>;
}
```

# Building DHCP Automatically

Eric Seigne

```
    chomp $fichier;
}
#si on constitue un dns et qu'on ne sait pas dans quel domaine on se trouve
while ($dns && $domaine eq "") {
    print STDERR "Quel est le nom de votre domaine ?:";
    $domaine = <STDIN>;
    chomp $domaine;
}

#on ouvre le fichier en lecture
open(IN,"<$fichier");

#lecture du fichier
my $ip = "";
my $hostname = "";
my $mac = "";
my $pos = 0;

#on ouvre la table de hash
my %h;

while (<IN>) {
#    print $_;
    #on cherche l'ip
    my ($t, $lease) = split("lease ",$_);
    if($lease){
        ($ip,$t) = split(" ",$lease);
    }

    #puis la hardware ethernet
    my ($t, $hard) = split("hardware ethernet ",$_);
    if($hard){
        ($mac,$t) = split(";", $hard);
    }

    #et enfin, le nom de la machine
    my ($t, $nom) = split("client-hostname \",$_);
    if($nom){
        ($hostname,$t) = split("\\"", $nom);
    }

    if($ip && $hostname && $mac){
        #on met les noms en minuscules
        $hostname = lc $hostname;
        #print "$mac/$ip/$hostname\n";
        #on ajoute tout ca dans la table de hash
        #pour eviter d'avoir des doublons
        $h{"$mac"} = "$ip/$hostname";
        $ip = $mac = $hostname = "";
    }
}

#on referme le fichier
Revised November 15, 2009
```

# Building DHCP Automatically

Eric Seigne

```
close(IN);

#On affiche ce qu'il faut rajouter à la fin du dhcpd.conf
open(OUT,">dhcpd.conf.ajouter");
while (($mac, $v) = each %h){
    my ($ip, $hostname) = split("\/", $v);
    #si on ne fait pas de dns, il faut les adresses IP
    #des machines
    if($dns){
        print OUT "host {
            hardware ethernet $mac;
            fixed-address \"$hostname.$domaine\";\n}\n\n";
    }
    else{
        print OUT "host {
            hardware ethernet $mac;
            fixed-address \"$ip\";\n}\n\n";
    }
}
print "\n\n\t\t\tRESULTATS:\n\n\t\t\t-----\n\n";
print "1 - Ajoutez dans votre fichier dhcpd.conf les lignes qui\n";
print "   se trouvent dans le fichier dhcpd.conf.ajouter\n\n";
close(OUT);

if($dns){
    #on classe le tableau par ordre croissant des ip
    #pour facilité la lecture du fichier de dns
    tie(%classe, 'DB_File', undef, O_RDWR|O_CREAT, 0, $DB_BTREE);
    while (($mac, $v) = each %h){
        my ($ip, $hostname) = split("\/", $v);
        $classe{"$ip"} = "$hostname";
    }

    #on ouvre le fichier de dns classique
    open(OUT,">named.$domaine.ajouter");
    print OUT "; Constitution automatique du dns à partir du dhcpd.leases\n";
    print OUT "; Eric Seigne <erics@rycks.com>\n";
    print OUT "; merci de controler ce fichier et d'ajouter son contenu à
votre dns\n\n";
    while (($ip, $hostname) = each %classe){
        # on envoie ce qu'il faut dans le fichier de dns
        # et les tabulations correctes pour éviter des
        # non alignements
        if(length($hostname)<8){
            print OUT "$hostname\t\t\tIN\tA\t$ip\n";
        }
        else{
            print OUT "$hostname\tIN\tA\t$ip\n";
        }
    }
}
#on ferme ce fichier
close(OUT);
```

# Building DHCP Automatically

Eric Seigne

```
#on passe au reverse dns
open(OUT,">named.$domaine.rev.ajouter");
print OUT "; Constitution automatique du dns inversé à partir du
dhcpd.leases\n";
print OUT "; Eric Seigne <erics\@rycks.com>\n";
print OUT "; merci de controler ce fichier et d'ajouter son contenu à
votre dns inversé\n";
while (($ip, $hostname) = each %classe){
    print OUT "$ip\t\tIN\tPTR\t$hostname.\n";
}
#on ferme ce fichier
close(OUT);

#les commentaires
print "2 - Pensez à ajoutez dans votre fichier named.conf les lignes
qui\n";
print "    correspondent à vos domaines\n\n";
print "3 - Ajoutez dans votre fichier named.$domaine les lignes qui\n";
print "    se trouvent dans le fichier named.$domaine.ajouter\n\n";
print "4 - Ajoutez dans votre fichier named.$domaine.rev les lignes
qui\n";
print "    se trouvent dans le fichier named.$domaine.rev.ajouter\n\n";
}
#on "libere" h ...
untie %h;
untie %classe;
#au revoir
print "Pensez ensuite à relancer les serveurs dhcp et dns si nécessaire !\n";
```