

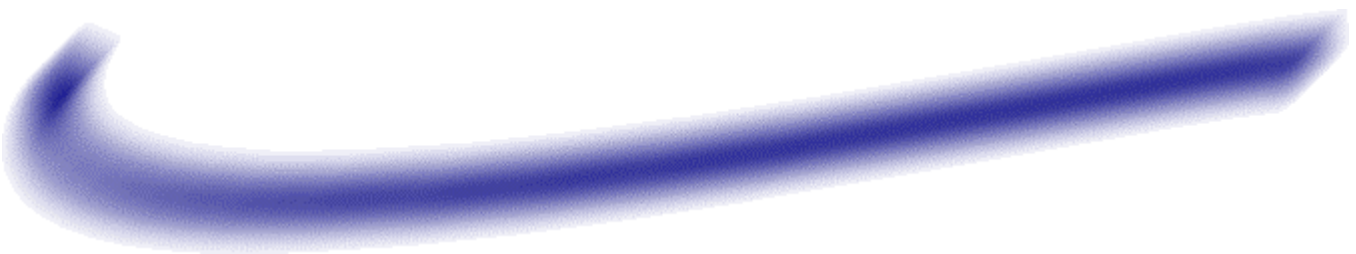


NetReg

”An automated DHCP Registration System”

Installing NetReg v1.3rc2 HOWTO

Instructions by
Patrick M. Jaques
e-mail: pjaques@comcast.net



Last Modified on June 17, 2004

Table of Contents

Chapter 1. Introduction	3
1.1. What is NetReg?	3
1.2. How Does NetReg Work?	3
1.3. Why use NetReg?	4
1.4. Potential pitfalls to NetReg	4
1.5. Recommendations	4
Chapter 2. NetReg Requirements	5
2.1. Pre-requirements	5
2.2. Download required software for NetReg	5
2.2.1 Download the following software distributions	5
2.2.2. Download the following perl modules from http://search.cpan.org	5
Chapter 3. Unpack Software Distributions	6
Chapter 4. Install Software Distributions	8
4.1. Install ISC DHCPd	8
4.2. Install Apache 2.0.49 SSL aware	8
4.2.1. Build OpenSSL.....	8
4.2.2. Fix Apache 2.x compile problem with Red Hat 9.0 (only)	8
4.2.3. Build Apache 2.0.49	8
4.2.4. Create SSL Certificates	8
4.3. Install Perl Modules	9
4.3.1. Libnet.....	9
4.3.2. Mail::POP3Client	9
4.3.3. Net::IMAP::Simple.....	9
4.3.4. Authen::SASL	9
4.3.5. Convert::ASN1	9
4.3.6. IO::Socket::SSL.....	10
4.3.7. MIME::Base64	10
4.3.8. XML::NamespaceSupport.....	10
4.3.9. XML::SAX.....	10
4.3.10. Net::SSLeay.....	10
4.3.11. Net::LDAP.....	10
Chapter 5. Configure Software	11
5.1. Configure NetReg	11
5.1.2. Rename index.html to register.html	11

5.1.3. Create a new index.html	12
5.1.4. Configure how NetReg will authenticate your users.....	12
5.1.5. Customize the Default paths that NetReg uses.....	13
5.1.6. Customize register.cgi	13
5.1.7. Configure subnets managed by NetReg	13
5.1.8. Configure the DHCP lease path in dhcpdctl.....	14
5.2. Configure Apache 2.0.49 w/SSL Support.....	14
5.2.1. Create user/group for Apache/DHCPd & NetReg.....	14
5.2.2. Add ErrorDocument lines to httpd.conf	14
5.2.3. Configure Apache httpd.conf restrictions.....	14
5.2.4. Configure ssl.conf in /usr/local/apache/conf	15
5.2.5. Create a .htaccess file	15
5.2.6. Create a user Authentication File	16
5.2.7. Change ownership/access rights on Apache's directories.....	16
5.2.8. Create a Startup script for Apache's httpd daemon.....	16
5.2.9. Create the proper Symbolic Links for Apache	18
5.3. Configuring ISC DHCPd	18
5.3.1. Modify the dhcpd.conf.	18
5.3.2. Copy your dhcpd.conf file.	20
5.3.3. Change ownership/access rights on DHCP files	20
5.3.4. Create a DHCP lease and tmp file.....	20
5.3.5. Create a startup script for the DHCPd daemon	20
5.3.6. Create the proper Symbolic Links for ISC DHCPd	22
5.4. Configuring DNS Bind	23
5.4.1. Configure named.conf	23
5.4.2. Configure the DNS Bind cache file.....	23
Chapter 6. Scheduling NetReg Update Script	24
6.1. Scheduling the refresh-dhcpdconf script.....	24
6.2. Fix refresh-dhcpdconf script.	25
Chapter 7. Troubleshooting	26
Chapter 8. Acknowledgements.....	29

Chapter 1. Introduction

This documentation is meant to serve as a step by step guide in downloading, compiling, installing and configuring a Network Registration system called NetReg on a Red Hat Linux system. The instructions reflect how to install **NetReg v1.3rc2** and the **CIDR kit** update running with **Apache web server version 2.0.49**.

Although this documentation is intended for Red Hat 6.0/7.0/8.0/9.0 systems, it should work fine on other Linux distributions. Please keep in mind that some of the script examples may not work on other Linux distributions without modifications. While the documentation provides a solid step by step instructional guide on installing NetReg, it does assume that the installer has a good understanding of Linux, DHCP, DNS Bind, Apache web server 2.x and TCP/IP networking.

1.1. What is NetReg?

NetReg is an automated network registration system that requires client computers that use DHCP to register their hardware (MAC) address before they can gain full network access.

1.2. How Does NetReg Work?

When a client computer running a standard DHCP client is connected to a network, it sends out a DHCP request. When a NetReg DHCP server receives this request, a lookup of the client's MAC address occurs. If the client's computer has previously registered its hardware address, it receives fully functional TCP/IP information (IP address, subnet mask, gateway address, WINS addresses, DNS server addresses), otherwise it receives an IP address within a restrictive network with no access to the internet.

How is this accomplished? The NetReg DHCP server is configured with two address pools per subnet. One pool of addresses is assigned to unregistered clients and the other pool is assigned to registered clients. The TCP/IP information passed to unregistered clients has a non-routable IP address or an IP address that is blocked on your firewall, and a bogus DNS server. The bogus DNS server is designed so it resolves all names back to a Network Registration Web Server. When a user starts a web browser, the web browser connects to the NetReg server and redirects all URLs to the NetReg Registration Page. From the Registration Page, the user reads and agrees to your "Acceptable Network Usage Policy", then enters a username/password that is authenticated against a POP/FTP/IMAP or LDAP server. If the user authenticates successfully, the computer gets registered. After the client reboots, the computer will have full access to the network and the Internet.

1.3. Why use NetReg?

DHCP is a standard protocol that automates the process of configuring network hosts by allowing hosts to obtain IP addresses and configuration parameters through the network; DHCP eliminates the need for manual configuration of hosts and manual assignments of IP addresses by network administrators. The problem with DHCP is that there is no security. Any computer can obtain access to your network through a DHCP server and for the most part is essentially anonymous. This makes tracking down malicious users more difficult.

Once a user registers their computer through NetReg's system, it links a user to a computer's hardware (MAC) address. This adds accountability to people's actions while they are connected to your network.

1.4. Potential pitfalls to NetReg.

Some clients may learn through others your network configuration and manually configure their computer so they can bypass the NetReg system.

- ❑ Manually assigning DNS servers
- ❑ Manually assigning IP Addresses
- ❑ Manually assigning a Default gateways

So, what can you do?

- ❑ Look at your switches/router's bridge and/or IP ARP tables and compare them to NetReg's registered hardware (MAC) addresses. This will tell you if you have any rogue users that are statically assigning IP addresses and bypassing your NetReg System. You can accomplish this by scheduling a perl script that uses "Net::SNMP" to periodically check your switches/routers.
- ❑ After identifying any rogue MAC addresses from your switch/router that were not registered by NetReg, your perl script can e-mail this information to your Network Administrator or another account. You may also be able to create a rule on your firewall that bans them from the Internet.

1.5. Recommendations

- ❑ Configure NetReg so it uses HTTP connections that run over SSL.
- ❑ Force SSL connections on your NetReg Registration web server. I added a "SSLRequireSSL" directive in my **.htaccess** file. For convenience, I configured NetReg so it redirected index.html to register.html using a SSL connection.

Chapter 2. NetReg Requirements

2.1. Pre-requirements

- ❑ Dedicated PC (Pentium 200MHz, 4GB HD, 32MB RAM, 10/100 Ethernet Adapter)
- ❑ Red Hat Linux 6.0/7.0/8.0/9.0, and other Linux platforms
- ❑ Developer tools - gcc, cc, make utilities
- ❑ DNS Bind 8/9
- ❑ Perl 5 +
- ❑ SSH (recommendation)
- ❑ An FTP, POP, IMAP, or LDAP server for NetReg to authenticate against
- ❑ Do not install Apache or ISC DHCPd as part of your Linux OS installation or as RPMs later. You will manually compile and install these services later.

2.2. Download required software for NetReg

2.2.1 Download the following software distributions

- ❑ Download [netreg-1.3rc2.tar.gz](http://www.netreg.org/netreg-1.3rc2.tar.gz) from <http://www.netreg.org>
- ❑ Download [netreg-cidr.tar.gz](http://www.netreg.org/contrib/netreg-cidr.tar.gz) from <http://www.netreg.org/contrib>
- ❑ Download [dhcp-latest.tar.gz](http://www.isc.org/dhcp-latest.tar.gz) from <http://www.isc.org>
- ❑ Download [httpd-2.0.49.tar.gz](http://www.apache.org/httpd-2.0.49.tar.gz) from <http://www.apache.org/>
- ❑ Download [openssl-0.9.7d.tar.gz](http://www.openssl.org/openssl-0.9.7d.tar.gz) from <http://www.openssl.org>

2.2.2. Download the following perl modules from <http://search.cpan.org>

- ❑ Download [libnet-1.18.tar.gz](http://search.cpan.org/libnet-1.18.tar.gz) (libnet)
- ❑ Download [POP3Client-2.13.tar.gz](http://search.cpan.org/POP3Client-2.13.tar.gz) (Mail::POP3Client)
- ❑ Download [Net-IMAP-Simple-0.93.tar.gz](http://search.cpan.org/Net-IMAP-Simple-0.93.tar.gz) (Net::IMAP::Simple)
- ❑ Download [Authen-SASL-2.06.tar.gz](http://search.cpan.org/Authen-SASL-2.06.tar.gz) (Authen::SASL)
- ❑ Download [Convert-ASN1-0.18.tar.gz](http://search.cpan.org/Convert-ASN1-0.18.tar.gz) (Convert::ASN1)
- ❑ Download [IO-Socket-SSL-0.95.tar.gz](http://search.cpan.org/IO-Socket-SSL-0.95.tar.gz) (IO::Socket::SSL)
- ❑ Download [Net-SSLeay.pm-1.25.tar.gz](http://search.cpan.org/Net-SSLeay.pm-1.25.tar.gz) (Net::SSLeay)
- ❑ Download [XML-NamespaceSupport-1.08.tar.gz](http://search.cpan.org/XML-NamespaceSupport-1.08.tar.gz) (XML::NamespaceSupport)
- ❑ Download [XML-SAX-0.12.tar.gz](http://search.cpan.org/XML-SAX-0.12.tar.gz) (XML::SAX)
- ❑ Download [MIME-Base64-3.00.tar.gz](http://search.cpan.org/MIME-Base64-3.00.tar.gz) (MIME::Base64)
- ❑ Download [perl-ldap-0.31.tar.gz](http://search.cpan.org/perl-ldap-0.31.tar.gz) (Net::LDAP)

Chapter 3. Unpack Software Distributions

Unpack the following software distributions in /usr/local/src.

- ❑ **Uncompress *httpd-2.0.49.tar.gz***
cd /usr/local/src
tar xvzf httpd-2.0.49.tar.gz
- ❑ **Uncompress *dhcp-latest.tar.gz***
cd /usr/local/src
tar xvzf dhcp-latest.tar.gz
- ❑ **Uncompress *openssl-0.9.7d.tar.gz***
cd /usr/local/src
tar xvzf openssl-0.9.7d.tar.gz
- ❑ **Uncompress *libnet-1.18.tar.gz***
cd /usr/local/src
tar xvzf libnet-1.18.tar.gz
- ❑ **Uncompress *POP3Client-2.13.tar.gz***
cd /usr/local/src
tar xvzf POP3Client-2.13.tar.gz
- ❑ **Uncompress *Net-IMAP-Simple-0.93.tar.gz***
cd /usr/local/src
tar xvzf Net-IMAP-Simple-0.93.tar.gz
- ❑ **Uncompress *Authen-SASL-2.06.tar.gz***
cd /usr/local/src
tar xvzf Authen-SASL-2.06.tar.gz
- ❑ **Uncompress *Convert-ASN1-0.18.tar.gz***
cd /usr/local/src
tar xvzf Convert-ASN1-0.18.tar.gz
- ❑ **Uncompress *IO-Socket-SSL-0.95.tar.gz***
cd /usr/local/src
tar xvzf IO-Socket-SSL-0.95.tar.gz
- ❑ **Uncompress *MIME-Base64-3.00.tar.gz***
cd /usr/local/src
tar xvzf MIME-Base64-3.00.tar.gz
- ❑ **Uncompress *XML-SAX-0.12.tar.gz***
cd /usr/local/src
tar xvzf XML-SAX-0.12.tar.gz

❑ **Uncompress Net_SSLeay.pm-1.25.tar.gz**

```
cd /usr/local/src  
tar xvzf Net_SSLeay.pm-1.25.tar.gz
```

❑ **Uncompress perl-ldap-0.31.tar.gz**

```
cd /usr/local/src  
tar xvzf perl-ldap-0.31.tar.gz
```

❑ **Uncompress netreg-1.3rc2.tar.gz**

```
cd /usr/local/src  
tar xvzf netreg-1.3rc2.tar.gz
```

❑ **Uncompress netreg-cidr.tar.gz**

```
cd /usr/local/src/netreg-1.3rc2  
tar xvzf ../netreg-cidr.tar.gz
```

Chapter 4. Install Software Distributions

4.1. Install ISC DHCPd

```
cd /usr/local/src/dhcp-3.0pl2
./configure
make
make install
```

4.2. Install Apache 2.0.49 SSL aware

4.2.1. Build OpenSSL

```
cd /usr/local/src/openssl-0.9.7d
./config
make
make install
```

4.2.2. Fix Apache 2.x compile problem with Red Hat 9.0 (only)

For some strange reason, Red Hat 9.0 decided to move the include files for Kerberos from /usr/include to /usr/kerberos/include. You will need to create a symlink for the following files to avoid getting the error “**krb5.h no such file or directory**” when compiling Apache 2.x.

```
# ln -s /usr/kerberos/include/com_err.h /usr/include
# ln -s /usr/kerberos/include/profile.h /usr/include
# ln -s /usr/kerberos/include/krb5.h /usr/include
```

4.2.3. Build Apache 2.0.49

```
# cd /usr/local/src/httpd-2.0.49
# ./configure --prefix=/usr/local/apache --enable-ssl --with-mpm-prefork --enable-rewrite
--enable-so --enable-cgi
# make
# make install
```

4.2.4. Create SSL Certificates

```
# openssl genrsa -out server.key 1024
# openssl req -new -key server.key -out server.csr
# openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
```

Change permission on certificate files

```
# chmod 400 server.*
```

Create SSL certificate directories

```
# mkdir /usr/local/apache/conf/ssl.key
# mkdir /usr/local/apache/conf/ssl.crt
```

Copy certificate files to Apache conf SSL directories

```
# cp server.key /usr/local/apache/conf/ssl.key  
# cp server.crt /usr/local/apache/conf/ssl.crt
```

4.3. Install Perl Modules

4.3.1. Libnet

```
cd /usr/local/src/libnet-1.18  
perl Makefile.PL  
make  
make test  
make install
```

4.3.2. Mail::POP3Client

```
cd /usr/local/src/POP3Client-2.13  
perl Makefile.PL  
make  
make test  
make install
```

4.3.3. Net::IMAP::Simple

```
cd /usr/local/src/Net-IMAP-Simple-0.93  
perl Makefile.PL  
make  
make test  
make install
```

4.3.4. Authen::SASL

```
cd /usr/local/src/Authen-SASL-2.06  
perl Makefile.PL  
make  
make test  
make install
```

4.3.5. Convert::ASN1

```
cd /usr/local/src/Convert-ASN1-0.18  
perl Makefile.PL  
make  
make test  
make install
```

4.3.6. IO::Socket::SSL

```
cd /usr/local/src/IO-Socket-SSL-0.95
perl Makefile.PL
make
make test
make install
```

4.3.7. MIME::Base64

```
cd /usr/local/src/MIME-Base64-3.00
perl Makefile.PL
make
make test
make install
```

4.3.8. XML::NamespaceSupport

```
cd /usr/local/src/XML-NamespaceSupport-1.08
perl Makefile.PL
make
make test
make install
```

4.3.9. XML::SAX

```
cd /usr/local/src/XML-SAX-0.12
perl Makefile.PL
make
make test
make install
```

4.3.10. Net::SSLeay

```
cd /usr/local/src/Net_SSLeay.pm-1.25
perl Makefile.PL
make
make test
make install
```

4.3.11. Net::LDAP

```
cd /usr/local/src/perl-ldap-0.31
perl Makefile.PL
make
make test
make install
```

Chapter 5. Configure Software

5.1. Configure NetReg

- Copy files that were previously extracted from the *netreg-1.3rc2.tar.gz* archive to the appropriate directory.

```
# cd /usr/local/src/netreg-1.3rc2
# cp -r usr/local/apache/htdocs/ /usr/local/apache
# cp -r usr/local/apache/htdocs/gfx /usr/local/apache/htdocs
# cp -r usr/local/bin/ /usr/local
# cp -r usr/sbin/ /usr
```

- Copy files that were previously extracted from the *netreg-cidr.tar.gz* archive to the appropriate directory.

```
# cd /usr/local/src/netreg-1.3rc2
# cp -r usr/local/apache2/cgi-bin/ /usr/local/apache
# cp -r etc/ /
```

- Rename subnet.dat.example to subnet.dat

```
# mv /etc/netreg/subnet.dat.example /etc/netreg/subnet.dat
```

The following are a list of files copied:

- index.html – gets copied to /usr/local/apache/htdocs
- register.cgi – gets copied to /usr/local/apache/cgi-bin
- variables.pl – gets copied to /usr/local/apache/cgi-bin
- subnet.pl – gets copied to /usr/local/apache/cgi-bin
- admin.cgi – gets copied to /usr/local/apache/cgi-bin/admin
- subnet.dat – gets copied to /etc/netreg
- *.gif – gets copied to /usr/local/apache/htdocs/gfx
- *.jpg – gets copied to /usr/local/apache/htdocs/gfx
- refresh-dhcpdconf – gets copied to /usr/local/bin
- db.root – gets copied to /etc
- named.conf – gets copied to /etc
- dhcpd.conf – gets copied to /etc/dhcpd
- dhcpdctl - gets copied to /usr/sbin

5.1.2. Rename index.html to register.html

```
cd /usr/local/apache/htdocs
mv index.html register.html
```

5.1.3. Create a new index.html

So it automatically redirects you to **register.html** using a secure connection. Change the below yellow highlight so it reflects your NetReg server's host name or IP Address.

```
<html>
<head>
  <title> Online Network Registration</title>
  <meta http-equiv="pragma" content="no-cache">
  <meta http-equiv="refresh" content="0; url=https://192.168.0.33/register.html">
</head>
<body>
  Redirecting... please wait. If you are not redirected automatically, then click
  <a href="/register.html"> here</a>.
</body>
</html>
```

5.1.4. Configure how NetReg will authenticate your users

Edit **Variables.pl** in **/usr/local/apache/cgi-bin**.

- If you want NetReg to authenticate against a POP server:
 - ✓ Change **\$AUTH_METHOD** to "**POP**".
 - ✓ Change **\$POPSERVER** to the hostname or IP Address of your mail server.
- If you want NetReg to authenticate against a FTP server:
 - ✓ Change **\$AUTH_METHOD** to "**FTP**".
 - ✓ Change **\$FTPSERVER** to the hostname or IP Address of your FTP server.
- If you want NetReg to authenticate against a IMAP server:
 - ✓ Change **\$AUTH_METHOD** to "**IMAP**".
 - ✓ Change **\$IMAPSERVER** to the hostname or IP Address of your IMAP server.
- If you want NetReg to authenticate against a LDAP server:
 - ✓ Change **\$AUTH_METHOD** to "**LDAP**".
 - ✓ Change **@LDAP_SERVERS** to the hostname or IP Address of your LDAP servers. Example: **@LDAP_SERVERS = ("143.44.65.6", "143.44.65.18");**
 - ✓ Change **\$LDAP_BASE** to your LDAP user base. For example: **"ou=users,dc=homedomain,dc=cc";**
 - ✓ Change **\$LDAP_AUTH_ATTR** to the user attribute, which is either "**uid**" or "**cn**". To authenticate against Microsoft's Active Directory, change **\$LDAP_AUTH_ATTR** to "cn".

- ✓ Change `$LDAP_USE_ADS` to 1 if you wish to use Microsoft's Active Directory Server as your authentication source.
- ✓ Change `$LDAP_ADS_DOMAIN` to your domain. For example:
`$LDAP_ADS_DOMAIN = "homedomain.cc";`

The above examples show you how to authenticate against Microsoft's Active Directory. To authenticate against other LDAP entities, there are other LDAP variables that you may need to change in `variables.pl`.

5.1.5. Customize the Default paths that NetReg uses

Edit `Variables.pl` in `/usr/local/apache/cgi-bin`. Verify that the following variables are set to the following values. These are the default filename and paths for ISC DHCPd and NetReg 1.3rc2 w/CIDR Update:

```
$LEASESPATH      = "/var/state/dhcp";
$LEASESFILE      = "dhcpd.leases";
$DHCPDCONF_PATH  = "/etc/dhcpd";
$DHCPDCONF_FILE  = "dhcpd.conf";
$SUBNETFILE      = "/etc/netreg/subnet.dat";
```

5.1.6. Customize register.cgi

Edit `register.cgi` in `/usr/local/apache/cgi-bin`. Locate subroutine called 'sub error' and change the HelpDesk message and number.

```
print "Please try to register again. If the problem persists, call the ";
print "HelpDesk at x6570.<P>";
```

5.1.7. Configure subnets managed by NetReg

Edit `subnet.dat` in `/etc/netreg` so it reflects all the subnets on your network that utilize DHCP. The format of `subnet.dat` is:

- Subnet/mask length for registered clients
- Location of subnet (description)
- Total number of leases from the subnet for registered clients
- Subnet/mask length for unregistered clients if different, otherwise blank.

Suppose your network consisted of 3 VLANS using a different subnet for registered and unregistered DHCP clients. Registered clients receive a `10.x.x.x` IP address and unregistered clients receive a `172.x.x.x` IP address. Below is what `subnet.dat` would resemble:

```
10.1.0.0/16: Dorms Blds: 5100: 172.16.0.0/16:
10.2.0.0/16: Classrooms: 5100: 172.17.0.0/16:
10.3.0.0/16: Admin Blds: 5100: 172.18.0.0/16:
```

5.1.8. Configure the DHCP lease path in dhcpdctl

Edit NetReg's script called **dhcpdctl** in `/usr/sbin` so the DHCP lease path reflects where ISC DHCP server stores and updates leases.

Change the DHCPD variable to:

```
DHCPD='/usr/sbin/dhcpd -cf /etc/dhcpd/dhcpd.conf -lf /var/state/dhcp/dhcpd.leases -q'
```

5.2. Configure Apache 2.0.49 w/SSL Support

5.2.1. Create user/group for Apache/DHCPd & NetReg

- Create user & group called **netreg** with no logon abilities.
- Edit Apache's `httpd.conf` file and change the following lines:

From:

```
User nobody  
Group nobody
```

To:

```
User netreg  
Group netreg
```

5.2.2. Add ErrorDocument lines to httpd.conf

```
ErrorDocument 403 /  
ErrorDocument 404 /
```

5.2.3. Configure Apache httpd.conf restrictions

Configure Apache's `httpd.conf` file so your registration server's web directories are properly restricted. The below example restrict access to the registration server so only clients from a 192.168.0.0 network can access it. (see the below yellow highlights).

-----Example: portion of httpd.conf-----

```
DocumentRoot "/usr/local/apache/htdocs"  
<Directory />  
    Options FollowSymLinks  
    AllowOverride None  
    Order deny,allow  
    deny from all  
    allow from 192.168.0 127.0.0.1  
</Directory>  
<Directory "/usr/local/apache/htdocs">  
    Options Indexes FollowSymLinks MultiViews  
    AllowOverride None  
    Order deny,allow  
    deny from all  
    allow from 192.168.0 127.0.0.1
```

```

</Directory>
AccessFileName .htaccess
<Files ~ "^\.ht">
    Order allow,deny
    Deny from all
    Satisfy All
</Files>
Alias /icons/ "/usr/local/apache/icons/"
<Directory "/usr/local/apache/icons">
    Options Indexes MultiViews
    AllowOverride None
    Order deny,allow
    deny from all
    allow from 192.168.0 127.0.0.1
</Directory>
ScriptAlias /cgi-bin/ "/usr/local/apache/cgi-bin/"
<Directory "/usr/local/apache/cgi-bin">
    AllowOverride AuthConfig
    Options None
    Order deny,allow
    deny from all
    allow from 192.168.0 127.0.0.1
</Directory>

```

5.2.4. Configure ssl.conf in /usr/local/apache/conf

- ❑ Change “**ServerName**” from www.example.com:443 to your host.domain name.
For example: ServerName netreg.homedomain.cc:443
- ❑ Change “**ServerAdmin**” to an e-mail address, where you want problems with the server to be e-mailed. For example: ServerAdmin root@netreg.homedomain.cc

5.2.5. Create a .htaccess file

Place this file (**.htaccess**) in your **/usr/local/apache/cgi-bin/admin** to protect NetReg’s administrative script.

-----Example: .htaccess file-----

```

AuthUserFile /usr/local/apache/conf/htusers
AuthName "NetReg Administration"
AuthType Basic
AuthAuthoritative on
SSLRequireSSL
require valid-user

```

5.2.6. Create a user Authentication File

Create a user authentication file called 'htusers' and place it in /usr/local/apache/conf.

Ex. `/usr/local/apache/bin/htpasswd -c /usr/local/apache/conf/htusers netreg`

The `-c` argument tells **htpasswd** to create a new users file for Apache in the specified directory. When you run this command, you will be prompted to enter a password for 'netreg', and then confirm it. Other users can be added to the existing file in the same way without the `-c` argument.

FYI: To allow a directory to be restricted within a .htaccess file, you first need to ensure that the httpd.conf allows user authentication to be set up in a .htaccess file. This is controlled by the AuthConfig override. The **httpd.conf** file should include **AllowOverride AuthConfig** under the **<Directory "/usr/local/apache/cgi-bin">** directive.

5.2.7. Change ownership/access rights on Apache's directories

`chown -R root:netreg /usr/local/apache`

5.2.8. Create a Startup script for Apache's httpd daemon

Since you are manually compiling and installing Apache from scratch (not using a RPM), you will need to create a startup script called '**httpd**' to start Apache when you boot your server. Place the startup script '**httpd**' in the **/etc/rc.d/init.d** directory. After creating a startup script for Apache, change the file permissions of your startup script so the Owner has read/write/execute access, Group has read/execute access and Other has read/execute access.

Example: `chmod 755 /etc/rc.d/init.d/httpd`

Make sure you start Apache's http daemon with SSL support by using either:

`/usr/local/apache/bin/apachectl startssl` (or)
`/usr/local/apache/bin/httpd -DSSL`

-----Example: Apache Script -- (Put in /etc/rc.d/init.d)-----

```
#!/bin/bash
# httpd - init file for Apache
#
# description: Apache Web Server protocol (HTTP) Daemon
#
# processname: /usr/local/apache/bin/httpd
# config: /usr/local/apache/conf/httpd.conf
# pidfile: /var/run/httpd
#
# source function library
. /etc/init.d/functions
#
OPTIONS="startssl"
RETVAL=0
```

```

prog="httpd"

start() {
    echo -n $"Starting $prog: "
    daemon /usr/local/apache/bin/apachectl $OPTIONS
    RETVAL=$?
    echo
    touch /var/lock/subsys/httpd
    return $RETVAL
}

stop() {
    echo -n $"Stopping $prog: "
    killproc /usr/local/apache/bin/httpd
    RETVAL=$?
    echo
    rm -f /var/lock/subsys/httpd
    return $RETVAL
}

reload(){
    echo -n $"Reloading $prog: "
    killproc /usr/local/apache/bin/httpd -HUP
    RETVAL=$?
    echo
    return $RETVAL
}

restart(){
    stop
    start
}

condrestart(){
    [ -e /var/lock/subsys/httpd ] && restart
    return 0
}

case "$1" in
    start)
        start
        ;;
    stop)
        stop
        ;;
    restart)
        restart
        ;;
    reload)
        reload
        ;;
)

```

```

condrestart)
    condrestart
    ;;
status)
    status httpd
    RETVAL=$?
    ;;
*)
    echo $"Usage: $0 {start|stop|status|restart|condrestart|reload}"
    RETVAL=1
esac
exit $RETVAL

```

5.2.9. Create the proper Symbolic Links for Apache

You will need to create symbolic links to the 'httpd' startup script in the run-level directories (rc0.d, rc1.d, rc2.d, rc3.d, rc4.d, rc5.d, rc6.d) in order for Apache to be automatically started and shutdown properly.

Create the proper Symbolic Links

```

cd /etc/rc.d/init.d
ln -s /etc/rc.d/init.d/httpd /etc/rc.d/rc0.d/K37httpd
ln -s /etc/rc.d/init.d/httpd /etc/rc.d/rc1.d/K37httpd
ln -s /etc/rc.d/init.d/httpd /etc/rc.d/rc2.d/S45httpd
ln -s /etc/rc.d/init.d/httpd /etc/rc.d/rc3.d/S45httpd
ln -s /etc/rc.d/init.d/httpd /etc/rc.d/rc4.d/S45httpd
ln -s /etc/rc.d/init.d/httpd /etc/rc.d/rc5.d/S45httpd
ln -s /etc/rc.d/init.d/httpd /etc/rc.d/rc6.d/K37httpd

```

5.3. Configuring ISC DHCPd

5.3.1. Modify the dhcpd.conf.

Modify the **dhcpd.conf** file in **/etc/netreg** so it reflects your network infrastructure. The below example dhcpd.conf file provides DHCP support to one network, which is 192.168.0.0, although it could support many networks/subnets and shared networks.

-----Example dhcpd.conf—(single network)-----

```

# dhcpd.conf
# Configuration file for ISC dhcpd
# Option definitions common to all supported networks...
option domain-name "homedomain.cc";
option netbios-node-type 8;
server-identifier netreg.homedomain.cc;
max-lease-time 120;
default-lease-time 120;

```

```

allow bootp;
allow booting;
ddns-update-style none;
ddns-updates off;
omapi-port 7911;

# Define subnet pools for known/unknown clients
subnet 192.168.0.0 netmask 255.255.255.0 {
    authoritative;
    option subnet-mask 255.255.255.0;
    option broadcast-address 192.168.0.255;
    pool {
        range 192.168.0.100 192.168.0.150;
        option domain-name-servers 192.168.0.33;
        default-lease-time 120;
        max-lease-time 120;
        allow unknown clients;
    }
    pool {
        range 192.168.0.151 192.168.0.200;
        option routers 192.168.0.254;
        option domain-name-servers 24.218.0.228, 24.218.0.229;
        default-lease-time 1209600;
        max-lease-time 1209600;
        deny unknown clients;
    }
}

```

Note: A shared-network directive in the dhcpd.conf file allows ISC DHCP server to support multiple networks to on the same wire. This of course means that your router or Layer 3 switch is configured with overlapping VLANS.

-----Example portion of dhcpd.conf—(shared-network)-----

```

shared-network campus {
    subnet 192.168.1.0 netmask 255.255.255.0 {
        option broadcast-address 192.168.1.255;
        option routers 192.168.1.254;
        range 192.168.1.50 192.168.1.200;
    }
    subnet 192.168.2.0 netmask 255.255.255.0 {
        option broadcast-address 192.168.2.255;
        option routers 192.168.2.254;
        range 192.168.2.50 192.168.2.240;
    }
}

```

5.3.2. Copy your dhcpd.conf file.

In order for NetReg's system to update the dhcp registrations, you will need to make copies of dhcpd.conf.

```
# cd /etc/dhcpd
# cp -f dhcpd.conf dhcpd.conf.new
# cp -f dhcpd.conf dhcpd.conf.bak
```

When a client registers their computer's hardware address through NetReg's registration web server, it puts the hardware address in 'dhcpd.conf.new'. NetReg's scripts check every minute whether 'dhcpd.conf.new' has been updated and if so, it stops the dhcpd service, copies 'dhcpd.conf.new' to 'dhcpd.conf' and then restarts the dhcpd service.

5.3.3. Change ownership/access rights on DHCP files

Change the ownership/access rights to ISC DHCP configuration files in **/etc/dhcpd**.

```
# chown -R root:netreg /etc/dhcpd
# chmod 664 /etc/dhcpd/*
```

5.3.4. Create a DHCP lease and tmp file

Create a lease file and tmp file in **/var/state/dhcp**.

```
# touch /var/state/dhcp/dhcpd.leases
# touch /var/state/dhcp/dhcpd.leases.tmp
```

5.3.5. Create a startup script for the DHCPd daemon

Since you are manually compiling and installing ISC's dhcpd from scratch (not using a RPM), you will need to create a startup script called '**dhcpd**' to start ISC's dhcpd daemon when you boot your server. Place the startup script '**dhcpd**' in the **/etc/rc.d/init.d** directory. After creating a startup script for dhcpd, change the file permissions of your startup script so the Owner has read/write/execute access, Group has read/execute access and Other has read/execute access.

Example: `chmod 755 /etc/rc.d/init.d/dhcpd`

-----Example Script to Start dhcpd-----

```
#!/bin/bash
# dhcpd - init file for ISC DHCPd
#
# description: Internet Software Consortium DHCP Server Daemon

# source function library
. /etc/init.d/functions

# source networking functions
. /etc/sysconfig/network
```

```
# Check that networking is up.  
[ ${NETWORKING} = "no" ] && exit 0
```

```
# Check that ISC DHCPD files exist  
[ -f /usr/sbin/dhcpd ] || exit 0  
[ -f /etc/dhcpd/dhcpd.conf ] || exit 0  
[ -f /var/state/dhcp/dhcpd.leases ] || exit 0
```

```
OPTIONS="-cf /etc/dhcpd/dhcpd.conf -lf /var/state/dhcp/dhcpd.leases"
```

```
RETVAL=0
```

```
prog="dhcpd"
```

```
start() {  
    echo -n "$Starting $prog: "  
    daemon /usr/sbin/dhcpd $OPTIONS  
    RETVAL=$?  
    echo  
    touch /var/lock/subsys/dhcpd  
    return $RETVAL  
}  
stop() {  
    echo -n "$Stopping $prog: "  
    killproc /usr/sbin/dhcpd  
    RETVAL=$?  
    echo  
    rm -f /var/lock/subsys/dhcpd  
    return $RETVAL  
}  
reload(){  
    echo -n "$Reloading $prog: "  
    killproc /usr/sbin/dhcpd -HUP  
    RETVAL=$?  
    echo  
    return $RETVAL  
}  
restart(){  
    stop  
    start  
}  
condrestart(){  
    [ -e /var/lock/subsys/dhcpd ] && restart  
    return 0  
}  
case "$1" in  
    start)
```

```

        start
        ;;
stop)
    stop
    ;;
restart)
    restart
    ;;
reload)
    reload
    ;;
condrestart)
    condrestart
    ;;
status)
    status dhcpd
    RETVAL=$?
    ;;
*)
    echo $"Usage: $0 {start|stop|status|restart|condrestart|reload}"
    RETVAL=1
esac
exit $RETVAL

```

5.3.6. Create the proper Symbolic Links for ISC DHCPd

You will need to create symbolic links to the 'dhcpd' startup script in the run-level directories (rc0.d, rc1.d, rc2.d, rc3.d, rc4.d, rc5.d, rc6.d) in order for ISC DHCPd to be automatically started and shutdown properly.

Create the proper Symbolic Links:

```

cd /etc/rc.d/init.d
ln -s /etc/rc.d/init.d/dhcpd /etc/rc.d/rc0.d/K38dhcpd
ln -s /etc/rc.d/init.d/dhcpd /etc/rc.d/rc1.d/K38dhcpd
ln -s /etc/rc.d/init.d/dhcpd /etc/rc.d/rc2.d/S47dhcpd
ln -s /etc/rc.d/init.d/dhcpd /etc/rc.d/rc3.d/S47dhcpd
ln -s /etc/rc.d/init.d/dhcpd /etc/rc.d/rc4.d/S47dhcpd
ln -s /etc/rc.d/init.d/dhcpd /etc/rc.d/rc5.d/S47dhcpd

```

5.4. Configuring DNS Bind

5.4.1. Configure named.conf

Edit **named.conf** in **/etc**. Change the server directive so it reflects your NetReg server's hostname or IP address.

-----Example named.conf-----

```
server 192.168.0.33 {
    bogus yes;
};
options {
    directory "/etc/";
    recursion no;
    fetch-glue no;
};
zone "." in {
    type master;
    file "db.root";
}
```

5.4.2. Configure the DNS Bind cache file

Edit **/etc/db.root**.

- Add **\$TTL 3600**
- Change hostname and domain in **SOA** and **NS** lines.
- Change IP address and hostname in **A** lines.

-----Example db.root-----

```
$TTL 3600
. IN SOA netreg.homedomain.cc. root.netreg.homedomain.cc. (
    2          ; serial
    10800     ; refresh
    3600      ; retry
    604800    ; expire
    86400     ; default_ttl
)
      IN NS netreg.homedomain.cc.
netreg 86400 IN A 192.168.0.33
*.      86400 IN A 192.168.0.33
```

Chapter 6. Scheduling NetReg Update Script

6.1. Scheduling the refresh-dhcpdconf script.

Schedule the **refresh-dhcpdconf** script so it runs every minute. Since the DHCP server only reads its conf file when it is started, you will need a cron job that checks to see if the server needs to re-read its configuration. The refresh-dhcpdconf script helps do this but cron will need to run this script every minute or two. We have cron running the script every minute as root like so in /var/spool/cron/root. FYI: Don't edit /var/spool/cron/root directly, use crontab -e.

```
0-59/1 * * * * /usr/local/bin/refresh-dhcpdconf
```

Within a minute, refresh-dhcpdconf will check if the dhcpd.conf.new has been updated. If dhcpd.conf.new is newer than dhcpd.conf, it updates dhcpd.conf and restart the DHCP server.

- You have now successfully built a Network Registration server utilizing NetReg v1.3 rc2 and CIDR update. Please reboot your server or manually start the '**dhcpd**' and '**httpd**' services.

```
/etc/rc.d/init.d/dhcpd start    or    service dhcpd start  
/etc/rc.d/init.d/httpd start   or    service httpd start
```

6.2. Fix refresh-dhcpdconf script.

Some users have reported problems that the **refresh-dhcpdconf** script does not work on **Red Hat 9.0 Linux** systems. It appears that the process ID of dhcpd in the 'ps aux' command is not matching the process ID in /var/run/dhcpd.pid file. Below is a fix for that situation. See the yellow highlighted lines. The red highlight is the added code to fix the problem:

```
-----Example: refresh-dhcpdconf -----
#!/bin/bash
# refresh-dhcpdconf for Netreg
# Belongs at /usr/local/bin/refresh-dhcpdconf

if [ /etc/dhcpd/dhcpd.conf.new -nt /etc/dhcpd/dhcpd.conf ]; then
    echo "dhcpd.conf.new is newer than dhcpd.conf...Copying dhcpd.conf
to dhcpd.conf.bak"
    cp /etc/dhcpd/dhcpd.conf /etc/dhcpd/dhcpd.conf.bak
    echo "Copying dhcpd.conf.new to dhcpd.conf"
    cp /etc/dhcpd/dhcpd.conf.new /etc/dhcpd/dhcpd.conf
    echo "Reloading the server..."
    /usr/sbin/dhcpdctl stop
    sleep 1
    /usr/sbin/dhcpdctl start
    if [ "`ps aux|grep dhcpd|grep -v grep|grep -v refresh-dhcpdconf|awk
    '{print $2}'`" != "`cat /var/run/dhcpd.pid`" ]; then
        echo "Server start failed...copying .conf to .conf.bad"
        cp /etc/dhcpd/dhcpd.conf /etc/dhcpd/dhcpd.conf.bad
        echo "Copying dhcpd.conf.bak to dhcpd.conf"
        cp /etc/dhcpd/dhcpd.conf.bak /etc/dhcpd/dhcpd.conf
        echo "Trying to start the server..."
        /usr/sbin/dhcpdctl start
        if [ "`ps aux|grep dhcpd|grep -v grep|grep -v refresh-dhcpdconf|awk
        '{print $2}'`" != "`cat /var/run/dhcpd.pid`" ]; then
            echo "Fatal Error: Server could not start."
        fi
    fi
fi
```

Chapter 7. Troubleshooting

Below are just a few examples of some common problems that an inexperienced Linux user may encounter when installing NetReg.

The first rule of thumb in troubleshooting NetReg problems is to make sure that TCP/IP is working properly between your NetReg server and your client networks. If you can't ping your NetReg server from your client networks, then NetReg's core components (DNS, DHCP, WWW) will not be accessible to your clients.

Problem:

Clients are able to successfully register with NetReg, but do not receive a registered address after rebooting. Why?

Reason:

First, make sure your DHCP server is running by typing `'ps aux | dhcpd'`. Next, make sure your cron job for the `refresh-dhcpdconf` script is scheduled to run every minute. Providing that this is not your problem, it sounds like Apache web server doesn't have the correct permissions to modify `dhcpd.conf.new`. By default, Apache is configured to run using the 'nobody' account. You will need to create an account, then change Apache's `httpd.conf` file so it runs under this account, and then change the file permissions on `dhcpd.conf.new`, `dhcpd.conf`, `dhcpd.conf.bak` and `dhcpd.conf.bad` so the account Apache is running under has read/write permissions to these files. This is covered in the NetReg-HowTo guide.

Problem:

When I try to compile ISC DHCP server, it fails to compile with the following error message "`cc: Command not found`" and/or "`make: Comand not found`".

Reason:

When you installed and configured your Linux system, you didn't install the packages that contain the 'make', 'cc' and 'gcc' utilities. If you are running Red Hat Linux, you need to install the Developer Tools. If you are running a different Linux distribution, then you will need to figure out what package contains the 'make', 'cc' and 'gcc' utilities and install it.

Problem:

I followed the instructions in the NetReg-HowTo guide on how to install Apache web server. The problem is that clients are getting the wrong web page. Clients should be getting NetReg's registration page, but instead are seeing the default Apache installation web page, why?

Reason:

When you initially installed your Linux server, you must have installed Apache web server as part of the install. You will need to uninstall Apache's web server packages or RPMs and manually compile and install Apache as per the NetReg-HowTo instructions.

Problem:

Unregistered clients are able to get an IP address from the DHCP server, but are not able to browse the NetReg's web server's web pages.

Reason:

First, make sure your web server is running. You can do this by typing the command `'ps aux | grep httpd'`. If you don't see `'httpd'` as a process, then your web server is not running. Try starting Apache manually. If Apache's web server is running, then you might have a firewall enabled that is blocking incoming HTTP/HTTPS, DNS, or DHCP requests. You can check this by typing `'iptables --list'`. Adjust your firewall rules, so it permits incoming HTTP/HTTPS requests.

Problem:

DNS is not working. Performing lookups using either `'dig'` or `'nslookup'`, produces the following error *"connection timed out; no servers could be reached"*.

Reason:

DNS is not running on the server. To confirm that DNS is not running on your server, issue the command `'ps aux | grep named'`. If you don't see `'/usr/sbin/named'` as a process, then DNS is not running. Try starting DNS manually by issuing `'/etc/rc.d/init.d/named start'`. If the `'named'` service starts, then all you probably need to do is create symlinks to your named startup script in your runlevel directories (rc0.d, rc1.d, rc2.d, rc3.d, rc4.d, rc5.d, rc6.d). If DNS still does not start, then check your log files for more information. Also, check your `named.conf` and `db.root` files for errors. Another possibility is that you are running bind in chroot configuration that is not properly set up. Please check the location of your chroot bind configuration files, usually `/var/named/chroot`.

Problem:

I just set up my NetReg server (NetReg version 1.3rc2 with the latest CIDR update). Unregistered clients are correctly redirected to my NetReg registration page, but are unable to successfully register their computer. They get the following error message when they try to register:

Permission Denied -- You are not in my allowable ip range xx.xx.xx.xx.

Solution:

Usually, this points to a problem in either your `subnet.dat` or `dhcpd.conf.new` file.

- ❑ Make sure your subnets declared in your ***dhcpd.conf*** and ***subnet.dat*** files accurately reflects your network topology.
- ❑ Make sure the account Apache webserver runs under has read/write access to ***dhcpd.conf.new***, ***dhcpd.conf***, and ***dhcpd.conf.bak*** files.
- ❑ Make sure the subnets you declare in `subnet.dat` do not contain any leading spaces. Each line must begin with a digit or `#` for a comment line.

Problem:

I installed a self-signed certificate generated from **OpenSSL**, however, I still get a message saying: 'the name on the security certificate is not valid or does not match the name of the site'. Why?

Solution:

The SSL certificate you generated should specify a Fully Qualified Domain Name (FQDN) for your Netreg server like netreg.yourdomain.edu. The SSL Certificate can only be used on this FQDN and nothing else - otherwise a name mismatch occurs.

Problem:

I tried generating another self-signed certificate because my previous certificate expired, but after doing so the certificate still contains the old information. Why? I am running Apache 1.3.x web server.

Solution:

Try deleting the contents of the directories that contain your certificate authority info and re-issue your certificate. The SSL directories should be located in '/usr/local/apache/conf/'. To find the path to your certificate authority, you will need to look at your httpd.conf or ssl.conf file, delete the contents of these directories and then issue the following commands for Apache 1.3:

```
# cd /usr/local/src/apache_1.3.29
# make
# make certificate TYPE=custom
# make install
```

I recommend that you do not make the expiration date for your certificate too short. One year is recommended (365). FYI: Apache 2.x web server does not use the 'make certificate' command, so follow command to generate another self-signed certificate:

```
# openssl genrsa -out server.key 1024
# openssl req -new -key server.key -out server.csr
# openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
```

Change permission on certificate files

```
# chmod 400 server.*
```

Create SSL certificate directories

```
# mkdir /usr/local/apache/conf/ssl.key
# mkdir /usr/local/apache/conf/ssl.crt
```

Copy certificate files to Apache conf SSL directories

```
# cp server.key /usr/local/apache/conf/ssl.key
# cp server.crt /usr/local/apache/conf/ssl.crt
```

Weighing whether you want to use a self-signed certificate is a security decision you will need to decide for yourself. Some may feel it is fine for Intranet web servers and others may decide to only use them in lab or testing purposes. That being said, self-signed certificates should never be used on any web server that is accessible on the Internet.

Chapter 8. Acknowledgements

Many thanks go to the developers and contributors of the NetReg project. Their dedication has helped many network administrators take solace in knowing who are using our networks.

Programming:

Peter Valian valianp@southwestern.edu

Documentation:

Todd K Watson tkw@southwestern.edu

Peter Valian valianp@southwestern.edu

Code Contributors:

Allie M. Tate allie@lsu.edu

Bobby Clark bobby.clark@eku.edu

Robert Lowe robert.h.lowe@lawrence.edu