

KMBS Lame & Lousy Linux Guide: Linux dialout-on-demand PPP + IP Masquerading

Khairulmizam Samsudin, kmbs at eng.upm.edu.my

v0.2, 17 August 2002

Abstract

This guide attempt to provide basic guidance and a practical example to set up your Linux box as a NAT server for a small LAN with PPP on demand for internet access.

Contents

1 Introduction	2
1.1 Foreword	2
1.2 Feedback	2
1.3 Disclaimer	2
1.4 Internet Resources	2
1.5 Copyright, License and all that stuff	3
2 Overview	3
2.1 Aims	3
2.2 Requirements	4
3 The LAN	4
3.1 Steps to configure LAN	4
4 ISP Dialup	5
5 Connecting to the WORLD	6
6 Troubleshooting Tips	6
7 Misc	7
7.1 Author	7
7.2 Changes	7
7.3 To Do	7

8 Appendix	7
8.1 /etc/dhcpd.conf	7
8.2 /etc/ppp/options	7
8.3 /etc/ppp/call-isp	8
8.4 /etc/rc.d/rc.firewall	8
8.5 /etc/rc.d/rc.firewall.stop	11

1 Introduction

1.1 Foreword

As a new Linux user I just found out that it is very confusing to setup a decent NAT server to host a local LAN with a PPP on demand connection to the internet. There are several informative and concise documents lying around the net, so I decided to summarise these documents in a single "Lame & Lousy Linux Guide" as a starting point for new users alike.

1.2 Feedback

Feedback is very welcome for this guide. PLEASE REPORT ANY INACCURACIES!!! Please post additions, comments and criticism to 'kmb at eng.upm.edu.my'

1.3 Disclaimer

No liability for the contents of this guide can be accepted. A am not responsible for any damages incurred due to actions taken based on this document. I am not a Linux Guru, nor do I pretend to be one ;^).

Also bear in mind this is just a 'Lame' guide directly reproduce on better and more informative guide or HOWTO found on the net. The configuration & example given is 'Lousy' enough just to demonstrate a working practical example and can be further improved!!!. That allows us to make things simpler and leave the complex issues to the official HOWTOs. I am also lazy enough to reproduce detail information and urge you to go through the reading list.

1.4 Internet Resources

1. The Linux Net HOWTO <http://www.linuxdoc.org/HOWTO/Net-HOWTO/index.html>
2. The Linux DHCP mini-HOWTO <http://www.linuxdoc.org/HOWTO/mini/DHCP/index.html>
3. The Linux Modem HOWTO <http://www.linuxdoc.org/HOWTO/Modem-HOWTO.html>
4. The Linux PPP HOWTO <http://www.linuxdoc.org/HOWTO/PPP-HOWTO/index.html>

5. The Linux IP Masquerade HOWTO <http://www.linuxdoc.org/HOWTO/IP-Masquerade-HOWTO/index.html>
6. The Linux Firewall and Proxy Server HOWTO <http://www.linuxdoc.org/HOWTO/Firewall-HOWTO.html>
7. Iptables Tutorial <http://people.unix-fu.org/andreasson/iptables-tutorial/iptables-tutorial.html>
8. Linux PPP Dial-on-demand and IP Masquerade, A Hands-On HOWTO <http://handsonhowto.com/dodip.html>
9. Controlling Dial-on-Demand with Firewall Rules <http://sites.inka.de/sites/bigred/misc/dod.html>
10. FAQ for isdn4linux: dod: Unwanted dialout on demand <http://www.isdn4linux.de/faq/i4lfaq-16.html>

1.5 Copyright, License and all that stuff

This document is copyrighted (c) 2002 Khairulmizam Samsudin and distributed under the terms of the GNU General Public License.

Permission is granted to make and distribute copies of this manual provided the copyright notice and this permission notice are preserved on all copies.

Permission is granted to copy and distribute modified versions of this manual under the conditions for verbatim copying, provided that the derived work is distributed under the terms of a permission notice identical to this one. Translations fall under the category of "modified versions."

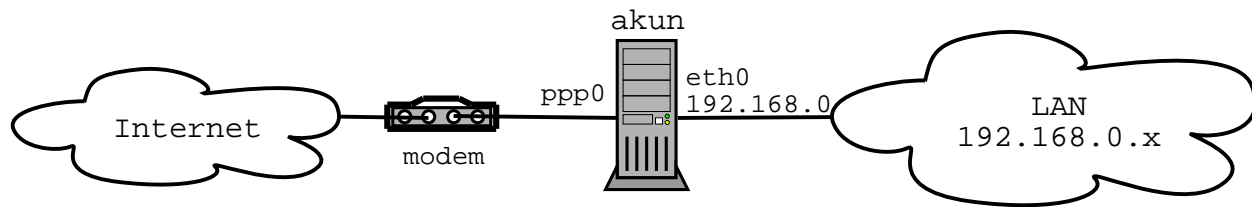
Warranty: None

2 Overview

2.1 Aims

This document will provide a guide for people who wish to setup up a low cost internet facility for their own LAN using PPP dialup account to an ISP. The Linux NAT server will also act as a DHCP server for the LAN.

For this reason, this document does NOT intended for setting up an ISDN or leased line connection.



2.2 Requirements

I am using Redhat 7.2, so these instructions are heavily geared towards that setup.

- You are running a 2.4.x Linux kernel. Please consult other documentation or HOWTO on installing and configuring a Linux system.

Note: Make sure you disable the firewall during the Redhat installation setup.

- dhcpd (version 2.0 or newer)
- Iptables
- A dialup account
- A lot of time :)

3 The LAN

First we'll configure the internal LAN. As you can see from the figure above, all my client machines are in the 192.168.0.x netblock where x is a value between 0 and 255. A DHCP server will eliminate the need to manually setup the network settings for each new PC that connects to your LAN. Most people only need the DHCP client daemon to obtain their network information (IP address, DNS server, etc.) from the DHCP server. (See Section 1.4).

NOTE: All networking interfaces (eth0 and ppp0), routing and daemons (dhcpd and pppd) are configured directly from my scripts. Apart from being lazy, I have warn you, didn't I? ;) these step will ensure that this guide is applicable to all Linux distribution. (Section 8)

3.1 Steps to configure LAN

Step #1 Configure DHCP server

Depending on your distribution, you can easily install and enable a DHCP daemon. Now you will need to configure `dhcpcd` by creating or editing `/etc/dhcpcd.conf`. You can find a simple `dhcpcd` configuration in Section 8. Please consult the man pages for an explanation of the file content. You will need to restart your `dhcpcd` daemon. For Redhat users `/etc/rc.d/init.d/dhcpcd restart` will do.

NOTE: Update `/etc/resolv.conf` so that the Linux box can lookup machines by name. The information can be obtain from your ISP. The file looks like this:

```
# /etc/resolv.conf
nameserver 123.123.123.123
nameserver 123.123.123.124
options attempts:10
```

Step #2 Configure DHCP client

Please refer to Section 1.4 for information on configuring a Linux client. For Window\$ make sure that the client obtain an IP address automatically and disable WINS resolution. You can also add the IP address of the server (in this example 192.168.0.1) as the default gateway and disable the DNS option.

Now your client should be able to obtain their network information automatically from your DHCP server. You can confirm that by noting the DHCP acknowledgement from the system logs. If you have any problem see Section 6 for suggestion and tips.

4 ISP Dialup

Next will configure our NAT server connection through the internet with the modem. Your ISP account information will be handy right now ;)

Step #1 Test modem configuration

For Redhat, the easiest way to test your hardware and ISP configuration is by using `wvdial`. You need to create and modify your ISP account information in the `wvdial.conf` file:

```
akun# wvdialconf > /etc/wvdial.conf
akun# vi /etc/wvdial.conf
akun# wvdial
```

NOTE: This testing procedure is not related to `dhcpcd`. If you have any problem please consult the related document. (See Section 1.4 or Section 6).

Step #2 Dialling on demand (dod)

To automate the ISP dialup and you need to pass optional parameter to `dhcpcd`. My options file (`/etc/ppp/options`) and an additional chat script (`/etc/ppp/call-isp`) is included in Section 8.

NOTE: The ISP account login and password has been specified in the chat script and the options file include additional *connect* and *disconnect* option that will be discussed in the next section.

5 Connecting to the WORLD

Step #1 Confirm iptables and related modules is enabled

Usually iptables is enabled by default each distribution kernel. You can make sure by running the command `ls /proc/net/` and make sure that the entry `ip_tables_names` exist.

All the related modules for my setup are `ip_tables`, `iptables_filter`, `ip_conntrack`, `ip_conntrack_ftp`, `ip_conntrack_irc`, `iptables_nat` and `ip_nat_ftp`. Run the command `ls /lib/modules/$(uname -r)/kernel/net/ipv4/netfilter/` and make sure all the modules exist. If it doesn't, you will need to compile a new kernel. Instruction on how to do this can be found in Linux-Kernel-HOWTO.

Step #2 Create rc.firewall

Create the file `/etc/rc.d/rc.firewall` and make it executable by running the command `chmod 700 /etc/rc.d/rc.firewall`. A sample rc.firewall file is included in Section 8.4.

The rc.firewall file include a modification to prevent dod disaster with a new chain, here named `dod`. The idea is to only accepts certain packet to trigger the dialout. After the PPP devices go up or down, the first rule is replaced with the connect and disconnect options of `/etc/ppp/options` file. (See Section 4). For this example I also drop HTTP and DNS lookup packets as a conscious policy. We don't want an application (anti-virus, winamp, etc) automatic updates to trigger the dialout.

NOTE: This is only a simple firewall ruleset to enable IP Masquerading. Please consult other documents to implement a stricter and secure Linux box. (See Section 1.4)

Step #3 Enable IP Masquerading.

If you plan on having IP Masquerading run after each reboot of your Linux machine, add it to your Linux startup scripts with the command: `echo "/etc/rc.d/rc.firewall" > > /etc/rc.local`. A sample script to stop all the NAT server daemon is included in Section 8.4.

6 Troubleshooting Tips

- User the primary system log files to troubleshoot any problem. I prefer to open another terminal to inspect the message from the Linux system.

```
akun# tail -f /var/log/messages
```

- groups.google.com is your friend. Try reading the related post from the achieve.
- Turn on kernel logging of matching packets in `rc.firewall` to inspect the ruleset.

7 Misc

7.1 Author

The author and maintainer of this guide is Khairulmizam Samsudin (kmbs at eng.upm.edu.my). Please send me any comments, additions and correction to improve this guide. You can take a look at my homepage at the URL: <http://eng.upm.edu.my/~kmbs/>

7.2 Changes

17/08/02 Added license and stuff and a new "Misc" section.

7.3 To Do

- Use default Redhat scripts (eg. /etc/sysconfig/iptables , etc..)

8 Appendix

8.1 /etc/dhcpd.conf

```
default-lease-time 600;
max-lease-time 7200;
option subnet-mask 255.255.255.0;
option broadcast-address 192.168.0.255;
option routers 192.168.0.1;
option domain-name-servers 161.x.x.x;
option domain-name "akun.box.my";

subnet 192.168.0.0 netmask 255.255.255.0 {
    range 192.168.0.10 192.168.0.50;
}
```

8.2 /etc/ppp/options

```
demand
debug
idle 60
ipcp-accept-remote
ipcp-accept-local
lock
noauth
defaultroute
modem crtscts /dev/ttyS1 57600
connect '/sbin/iptables -R FORWARD 1 -i eth0 -o ppp0 -j ACCEPT
&& /usr/sbin/chat -v -f /etc/ppp/call-isp'
disconnect '/sbin/iptables -R FORWARD 1 -o eth0 -j dod'
```

8.3 /etc/ppp/call-isp

```
TIMEOUT      5
ABORT        '\nBUSY\r'
ABORT        '\nNO ANSWER\r'
ABORT        '\nRINGING\r\n\r\nRINGING\r'
' '          \rAT
'OK-+++c-OK' ATH0
TIMEOUT      50
OK           ATDT9-1313
CONNECT      ''
ogin:-ogin:  user1
assword:     user1passwd
```

8.4 /etc/rc.d/rc.firewall

```
#!/bin/sh
#
#Configuring eth0 interface & starting daemons
/sbin/ifconfig eth0 192.168.0.1 netmask 255.255.255.0 up
/sbin/ifconfig lo 127.0.0.1 netmask 255.0.0.0 up
/usr/sbin/pppd
/etc/rc.d/init.d/dhcpd start
#Added for Window$ dhcp client
/sbin/route add -host 255.255.255.255 dev eth0
#
kmbsFWVER=0.1
#
#Modified from "Initial SIMPLE IP Masquerade test for 2.4.x
#kernels using IPTABLES" version 0. 63, from Linux IP Masquerade
HOWTO
#
echo -e "\n\nLoading KMBS LLLG rc.firewall version $kmbsFWVER..\n"

# The location of the 'iptables' program
IPTABLES=/sbin/iptables
#IPTABLES=/usr/local/sbin/iptables

#Setting the EXTERNAL and INTERNAL interfaces for the network
EXTIF="ppp0"
INTIF="eth0"
echo " External Interface: $EXTIF"
echo " Internal Interface: $INTIF"

echo -en " loading modules: "
```

```

# Need to verify that all modules have all required dependencies
echo " - Verifying that all kernel modules are ok"
/sbin/depmod -a

# Load the main body of the IPTABLES module - "iptables"
echo -en "ip_tables, "
/sbin/insmod ip_tables

#Load the IPTABLES filtering module - "iptables_filter"
# - Loaded automatically when filter policies are activated

#Load the stateful connection tracking framework - "ip_conntrack"
echo -en "ip_conntrack, "
/sbin/insmod ip_conntrack

#Load the FTP tracking mechanism for full FTP tracking
# Enabled by default – insert a "#" on the next line to deactivate
echo -en "ip_conntrack_ftp, "
/sbin/insmod ip_conntrack_ftp

#Load the IRC tracking mechanism for full IRC tracking
# Enabled by default – insert a "#" on the next line to deactivate
echo -en "ip_conntrack_irc, "
/sbin/insmod ip_conntrack_irc

#Load the general IPTABLES NAT code - "iptables_nat"
echo -en "iptables_nat, "
/sbin/insmod iptable_nat

#Loads the FTP NAT functionality into the core IPTABLES code
# Required to support non-PASV FTP.
# Enabled by default – insert a "#" on the next line to deactivate
echo -en "ip_nat_ftp, "
/sbin/insmod ip_nat_ftp

echo ". Done loading modules."

#CRITICAL: Enable IP forwarding since it is disabled by default
since
echo " enabling IP forwarding.."
echo "1" > /proc/sys/net/ipv4/ip_forward

# Dynamic IP users:
#
echo " enabling DynamicAddr.."
echo "1" > /proc/sys/net/ipv4/ip_dynaddr

```

```

# Enable simple IP forwarding and Masquerading
# NOTE: In IPTABLES speak, IP Masquerading is a form of Source-
NAT or SNAT.
# NOTE #2: The following is an example for an internal LAN address
in the
#      192.168.0.x network with a 255.255.255.0 or a "24" bit subnet
mask
#      connecting to the Internet on external interface "ppp0". This
#      example will MASQ internal traffic out to the Internet but not
#      allow non-initiated traffic into your internal network.

#Clearing previous configuration
# Unless specified, the defaults for INPUT and OUTPUT is ACCEPT
# The default for FORWARD is DROP
echo " clearing any existing rules and setting default policy.."
$IPTABLES -F INPUT ACCEPT
$IPTABLES -F INPUT
$IPTABLES -F OUTPUT ACCEPT
$IPTABLES -F OUTPUT
$IPTABLES -F FORWARD DROP
$IPTABLES -F FORWARD
$IPTABLES -t nat -F

#added by KMBS
# note: When the device is down, packets are routed to a
#       special "Dial on demand" (dod) chain
# note2: The new chain "dod" handle dialout permission, only
#         specified packets can trigger a dialout
$IPTABLES -N dod
$IPTABLES -A dod -j LOG
$IPTABLES -A dod -p icmp -i icmp-type ping -j ACCEPT
$IPTABLES -A dod -p tcp -dport telnet -j ACCEPT
$IPTABLES -A dod -p tcp -dport ssh -j ACCEPT
#$IPTABLES -A dod -p tcp -dport http -j ACCEPT
$IPTABLES -A dod -p tcp -dport smtp -j ACCEPT
$IPTABLES -A dod -p tcp -dport nntp -j ACCEPT
$IPTABLES -A dod -p tcp -dport pop3 -j ACCEPT
$IPTABLES -A dod -p tcp -dport imap2 -j ACCEPT
$IPTABLES -A dod -p tcp -dport imap3 -j ACCEPT
#$IPTABLES -A dod -p tcp -dport domain -j ACCEPT
#$IPTABLES -A dod -p udp -dport domain -j ACCEPT
#DEBUG: Log other stray packets
#$IPTABLES -A dod -j LOG
$IPTABLES -A dod -j DROP

```

```

echo " FWD: Allow all connections OUT and only existing and re-
lated ones IN"
#added by KMBS
#note: First create a dummy rule to be used when ppp0
# interface is down. There for rest of the rule will
# be travelled
#note2: When the interface is up (ip-up), replace the rule with
# $IPTABLES -R FORWARD 1 -i $INTIF -o $EXTIF -j ACCEPT
#note3: When the interface is down (ip-down), replace
# the rule with

$IPTABLES -A FORWARD -o $EXTIF -j dod
$IPTABLES -A FORWARD -i $EXTIF -o $INTIF -m state --state ES-
TABLISHED,RELATED -j ACCEPT
$IPTABLES -A FORWARD -j LOG

echo " Enabling SNAT (MASQUERADE) functionality on $EXTIF"
$IPTABLES -t nat -A POSTROUTING -o $EXTIF -j MASQUER-
ADE

echo -e "\nDone.\n"

```

8.5 /etc/rc.d/rc.firewall.stop

```

#!/bin/bash

echo "Shutting down DHCP server"
/etc/rc.d/init.d/dhcpd stop
echo "Shutting down PPP interface"
kill -9 `cat /var/run/ppp0.pid`
echo "Shutting down eth0 interface"
/sbin/ifconfig eth0 down
echo "Flushing all firewall rules & chain"
/sbin/iptables -F
/sbin/iptables -X dod

```