## 4.5. Enable IP forwarding and Masquerading

Doing the above might not be enough to make the Ubuntu machine a real router which does NAT (Network Address Translation) and IP Forwarding. The following script configures the Kernel IPTable and IP forwarding. You will have to configure at least the script's 2 variables; the 1st is the external network interface; the 2nd is the internal network interface.

* EXTIF="eth0"  
  INTIF="eth1"

The script was originally from a [Ubuntu router guide forum article](http://ubuntuforums.org/showthread.php?t=119787) which has 2 internal network interfaces. What's showing below uses only 1 internal network interface. You will have to modify the script manually or use the script in the [Ubuntu router guide forum article](http://ubuntuforums.org/showthread.php?t=119787) if you need to configure 2 internal network interfaces.

* echo -e "\n\nLoading simple rc.firewall-iptables version $FWVER..\n"  
  DEPMOD=/sbin/depmod  
  MODPROBE=/sbin/modprobe  
    
  EXTIF="eth0"  
  INTIF="eth1"  
  #INTIF2="eth0"  
  echo " External Interface: $EXTIF"  
  echo " Internal Interface: $INTIF"  
    
  #======================================================================  
  #== No editing beyond this line is required for initial MASQ testing ==   
  echo -en " loading modules: "  
  echo " - Verifying that all kernel modules are ok"  
  $DEPMOD -a  
  echo "----------------------------------------------------------------------"  
  echo -en "ip\_tables, "  
  $MODPROBE ip\_tables  
  echo -en "nf\_conntrack, "   
  $MODPROBE nf\_conntrack  
  echo -en "nf\_conntrack\_ftp, "   
  $MODPROBE nf\_conntrack\_ftp  
  echo -en "nf\_conntrack\_irc, "   
  $MODPROBE nf\_conntrack\_irc  
  echo -en "iptable\_nat, "  
  $MODPROBE iptable\_nat  
  echo -en "nf\_nat\_ftp, "  
  $MODPROBE nf\_nat\_ftp  
  echo "----------------------------------------------------------------------"  
  echo -e " Done loading modules.\n"  
  echo " Enabling forwarding.."  
  echo "1" > /proc/sys/net/ipv4/ip\_forward  
  echo " Enabling DynamicAddr.."  
  echo "1" > /proc/sys/net/ipv4/ip\_dynaddr   
  echo " Clearing any existing rules and setting default policy.."  
    
  iptables-restore <<-EOF  
  \*nat  
  -A POSTROUTING -o "$EXTIF" -j MASQUERADE  
  COMMIT  
  \*filter  
  :INPUT ACCEPT [0:0]  
  :FORWARD DROP [0:0]  
  :OUTPUT ACCEPT [0:0]  
  -A FORWARD -i "$EXTIF" -o "$INTIF" -m conntrack --ctstate ESTABLISHED,RELATED -j ACCEPT   
  -A FORWARD -i "$INTIF" -o "$EXTIF" -j ACCEPT  
  -A FORWARD -j LOG  
  COMMIT  
  EOF  
    
  echo -e "\nrc.firewall-iptables v$FWVER done.\n"

After configuring the 2 variables, save the script below as nat.sh and make it executable by doing

* chmod a+x nat.sh

Now, test the script by running as root

* sudo sh nat.sh

Investigate the messages from the console output to see if any error happened. If everything looks fine, use another host in the internal network to test if it can access the external network (presumably the Internet). A quick way to test is pinging [Google public DNS](http://code.google.com/speed/public-dns/) from the console.

* ping -c 3 -W 10 8.8.8.8

If ping responds, make our new script bootable so we don't have to run the script every time we restart.

* sudo cp nat.sh /etc/init.d/  
  sudo ln -s /etc/init.d/nat.sh /etc/rc2.d/S95masquradescript

As a final test, restart your computer and test to see if you still have the same functionality. If so then congratulations! If not then make sure you followed the above correctly so the script is bootable.