

The Perfect SpamSnake - Ubuntu 8.04 LTS

By Rocky (Contact Author) (Forums)

Published: 2008-05-01 18:45

The Perfect SpamSnake - Ubuntu 8.04 LTS Postfix w/Bayesian Filtering and Anti-Backscatter (Relay Recipients), Apache, Mysql, Bind, MailScanner (Spamassassin, ClamAV, Pyzor, Razor, DCC-Client), MailWatch, SPF Checks, FuzzyOcr, PDF/XLS/Phishing Sanesecurity Signatures, Postfix-GLD (Greylisting Optional), Logwatch Statistical Reporting (Optional), Outgoing Disclaimer with alterMIME (Optional), FireHOL (Iptables Firewall)

Version 2.0

Author: Mohammed Alli

This tutorial shows how to set up an Ubuntu Hardy Heron (8.04 LTS) based server as a spamfilter in Gateway mode. In the end, you will have a SpamSnake Gateway which will relay clean emails to your MTA. You will also be able to view your incoming queue, train your SpamSnake and carry out a few more advanced operations via MailWatch.

I cannot offer any guarantees that this will work for you, the same way it(TM)s working for me.

I will use the following software:

- Web Server: Apache 2.2 with PHP 5.2.4 and Ruby
- Database Server: MySQL 5.0
- Mail Server: Postfix
- DNS Server: BIND9
- PHP: PHP5
- MailScanner: MailScanner v4.68.8
- MailWatch: MailWatch v1.0.4

Credit goes to the guys at HowToForge and the developers of MailScanner and MailWatch.

1 Requirements

To install such a system you will need the following:

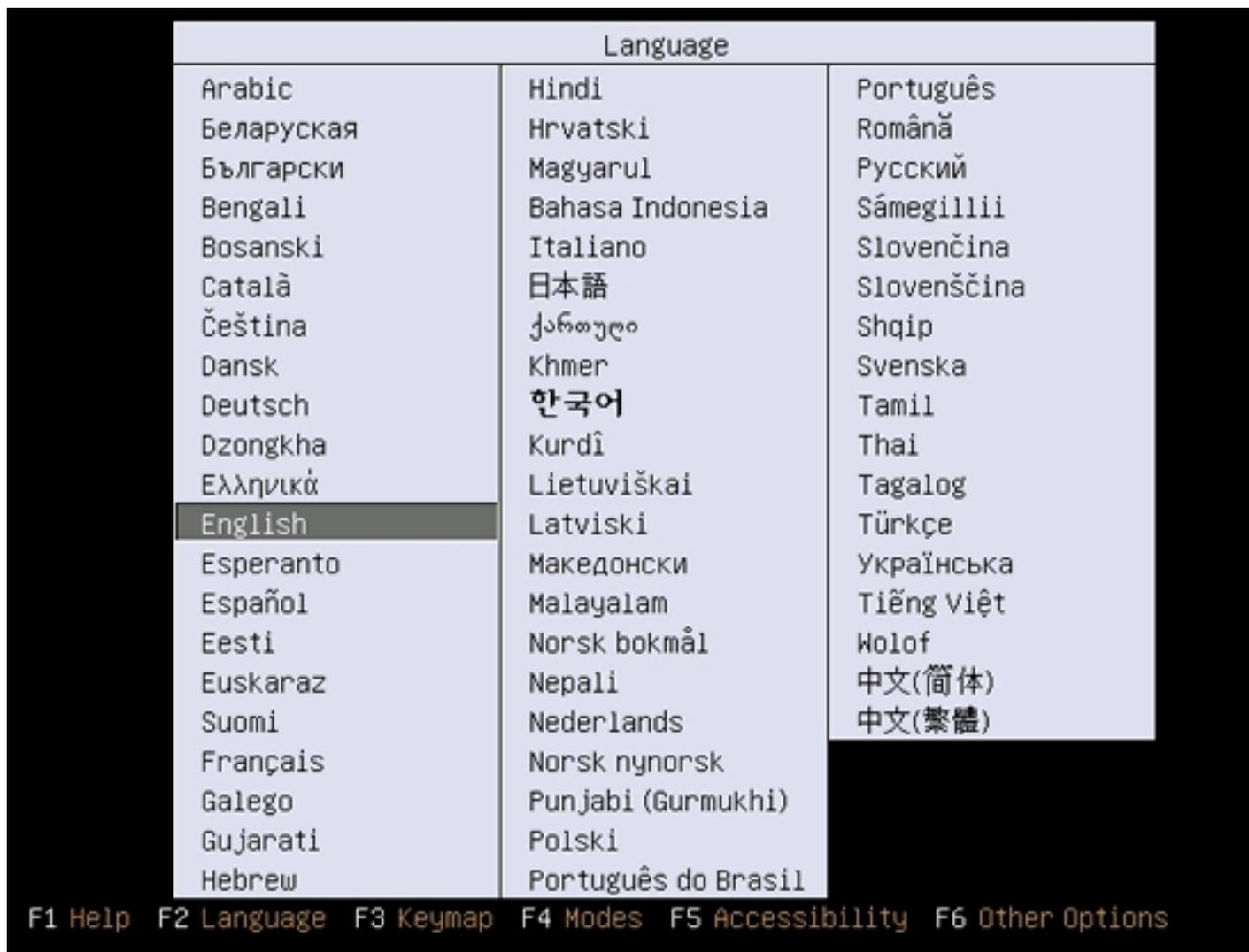
- The Ubuntu 8.04 LTS server CD, available here: <ftp://releases.ubuntu.com/releases/hardy/ubuntu-8.04-server-i386.iso>
- A fast internet connection.

1.1 Preliminary Note

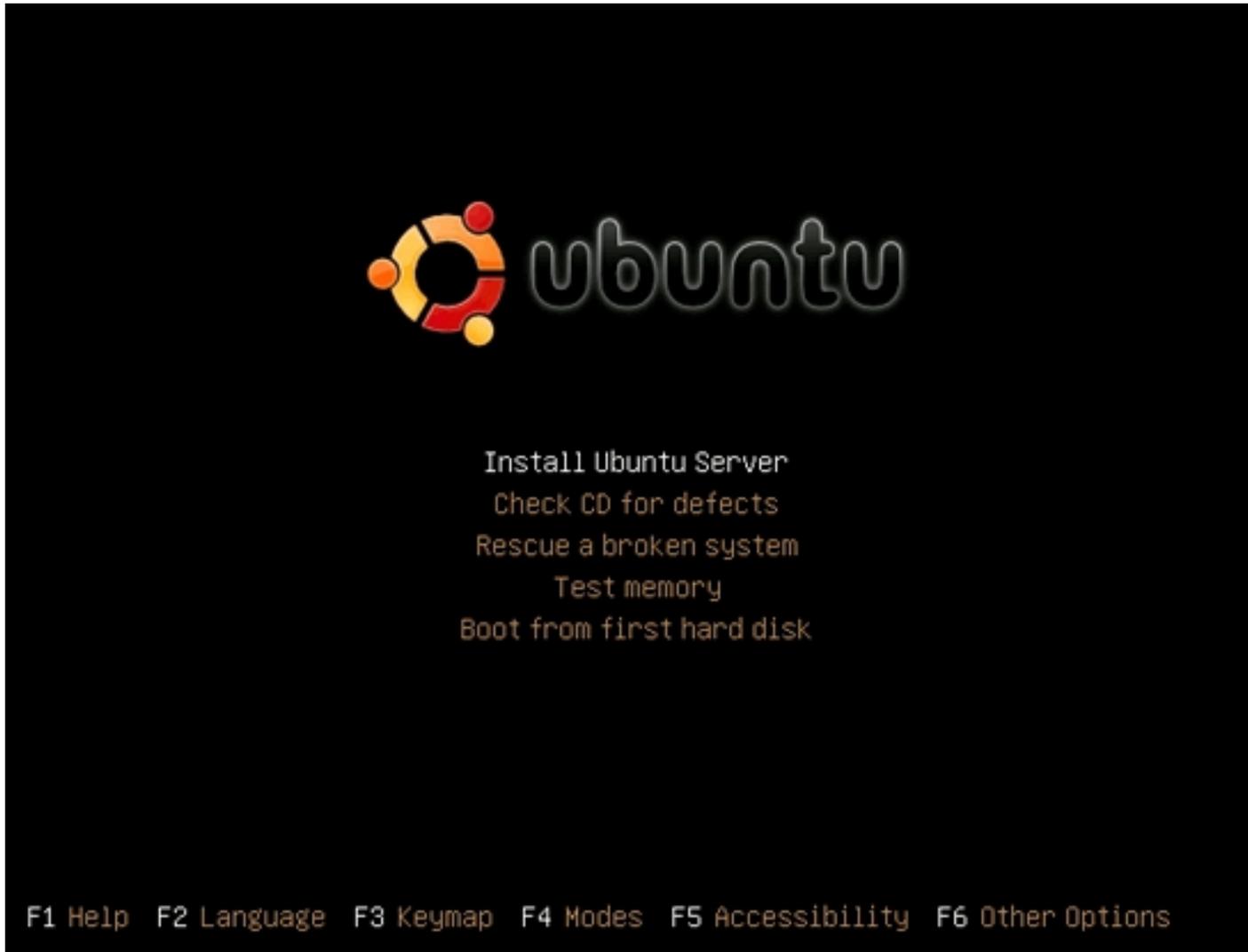
In this tutorial I use the hostname *server1.example.com* with the IP address *192.168.0.100* and the gateway *192.168.0.1*. These settings might differ for you, so you have to replace them where appropriate.

1.2 The Base System

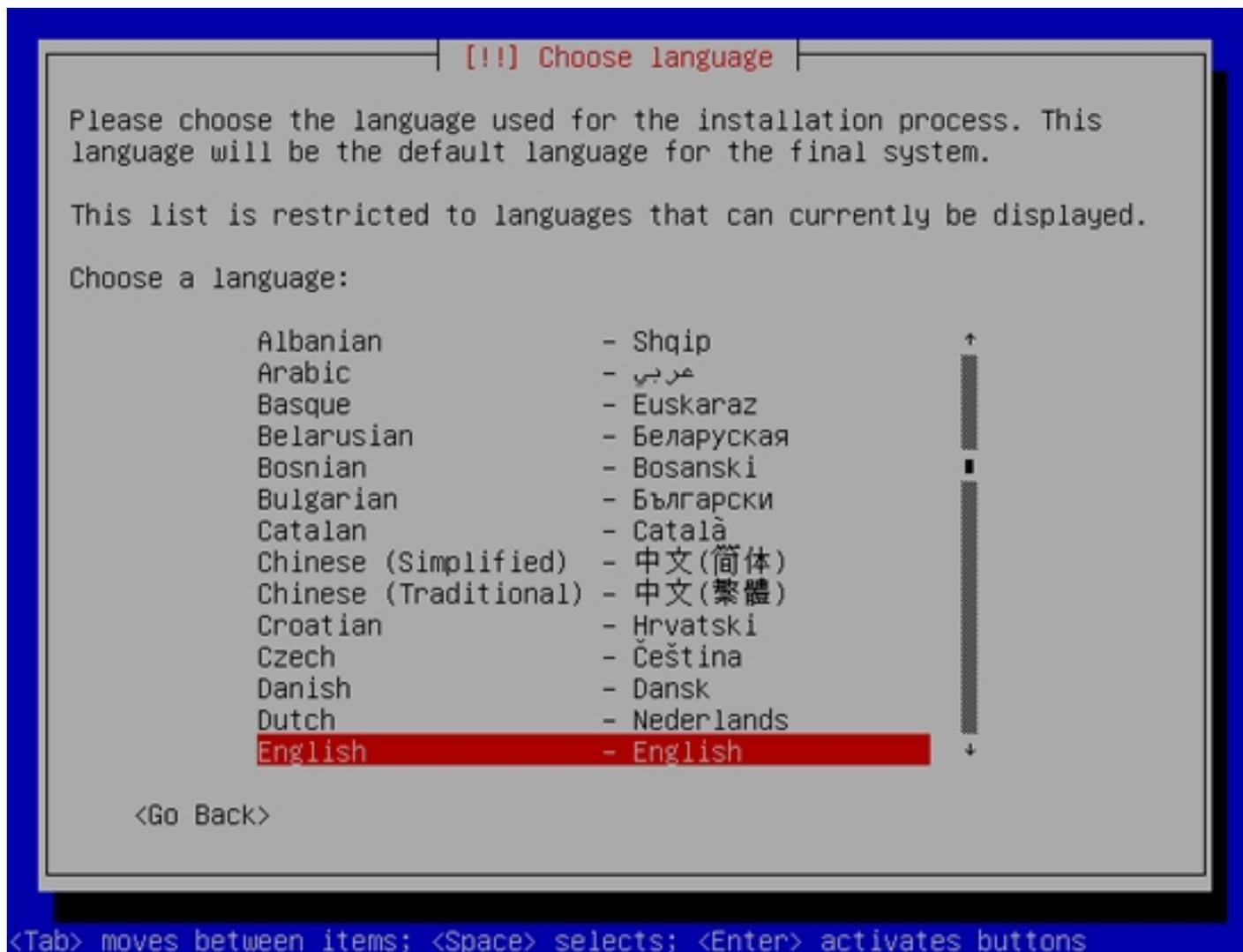
1. Insert your Ubuntu install CD into your system and boot from it. Select your language:



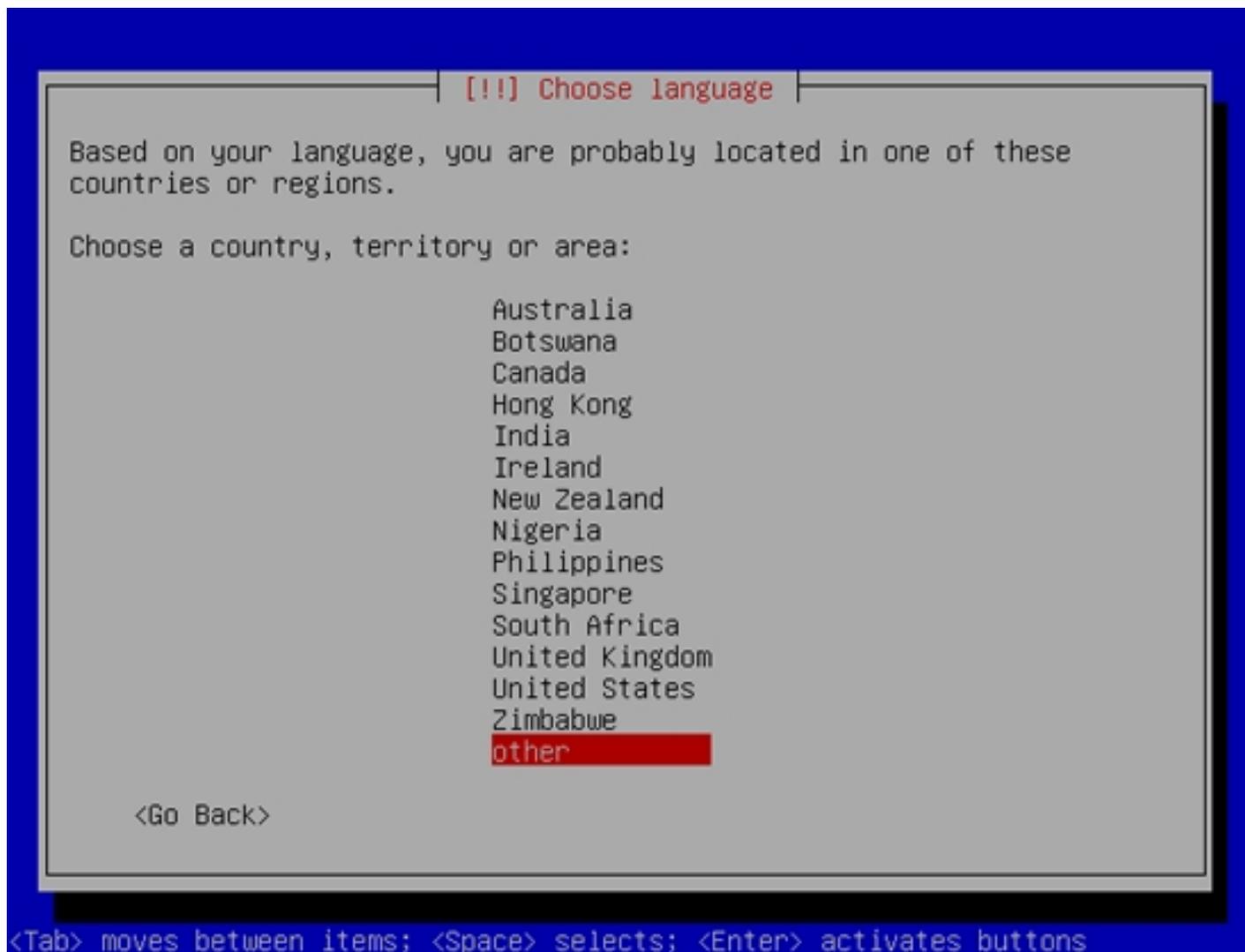
2. Select Install to the hard disk:



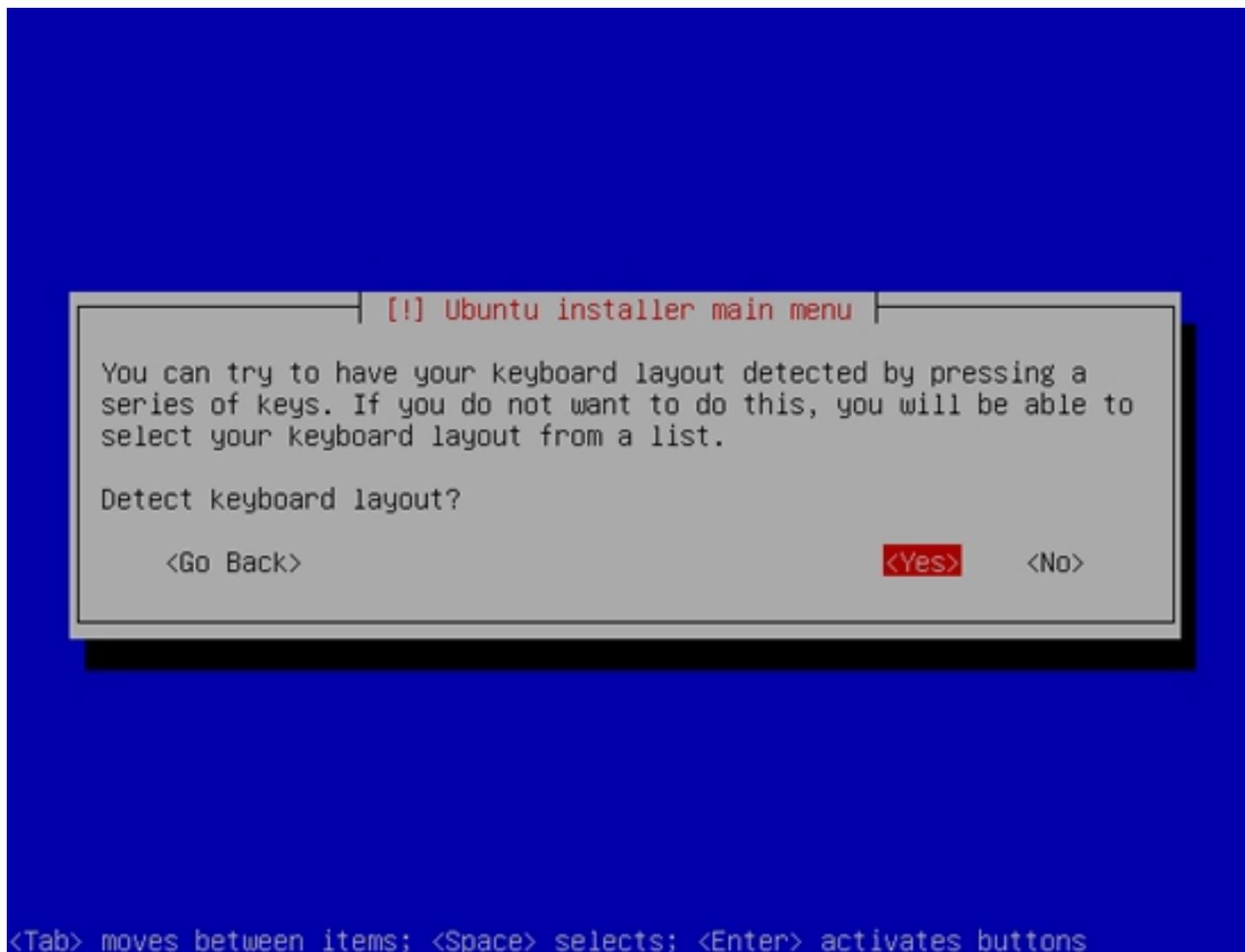
3. The installation starts, choose your language again:



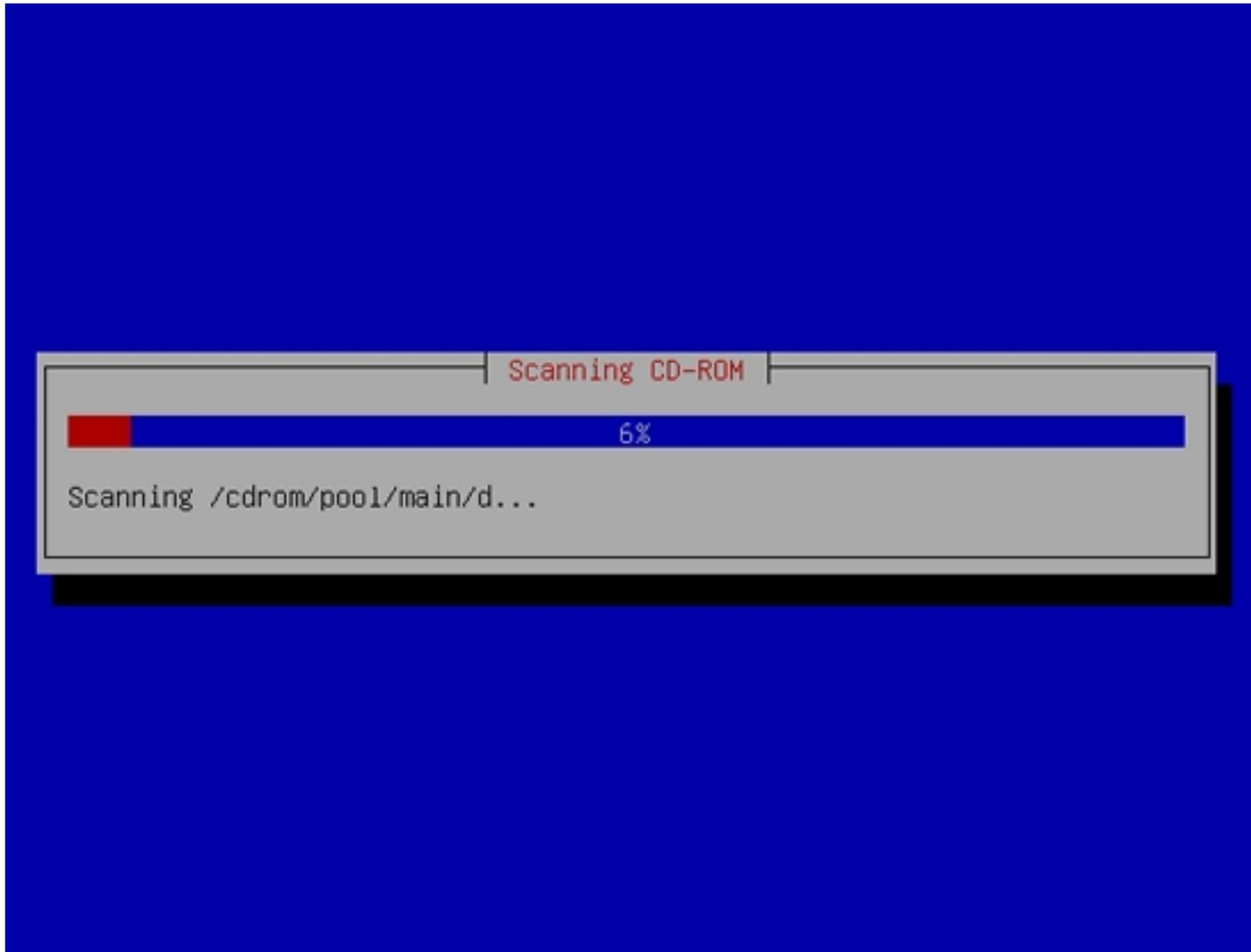
4. Then select your location:

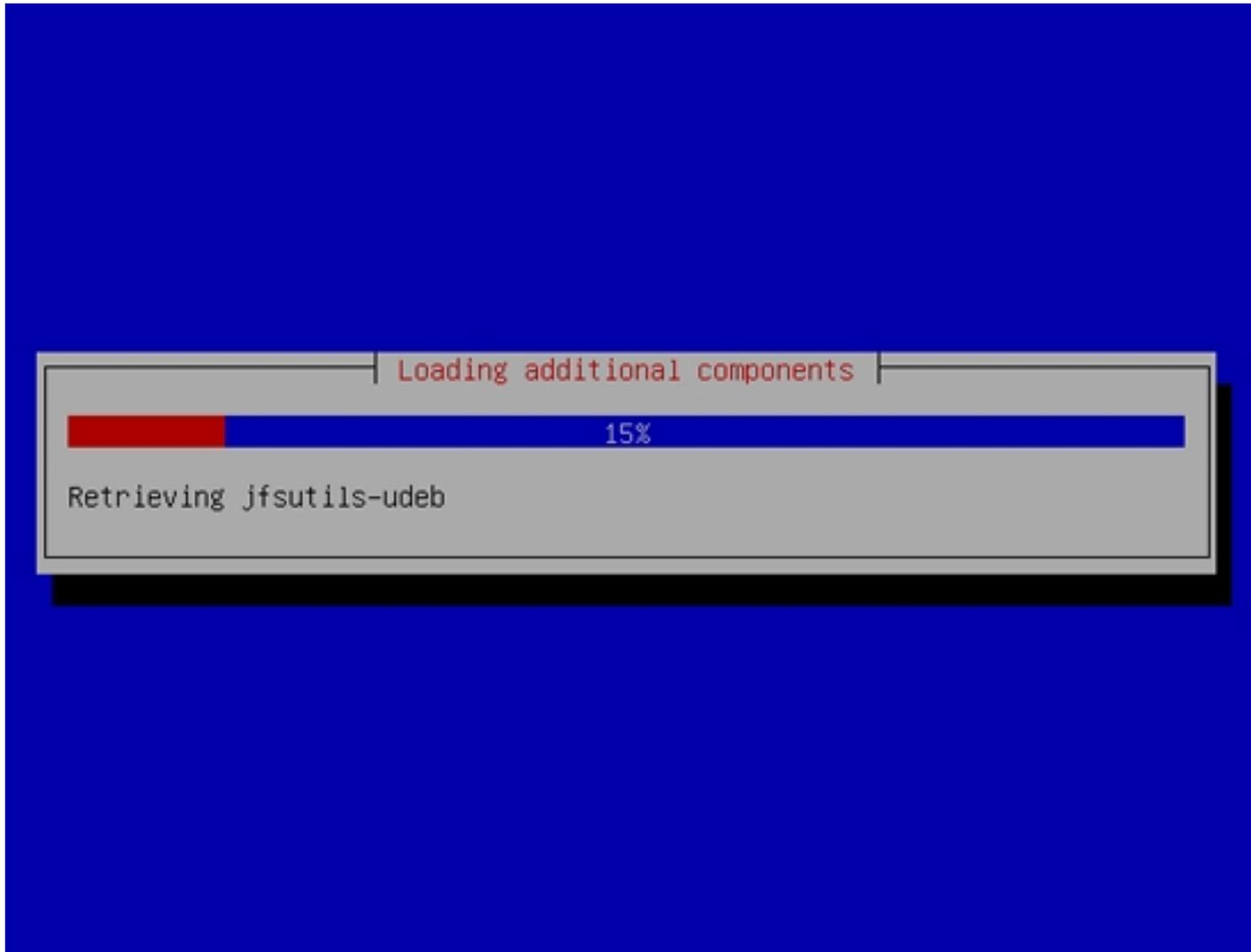


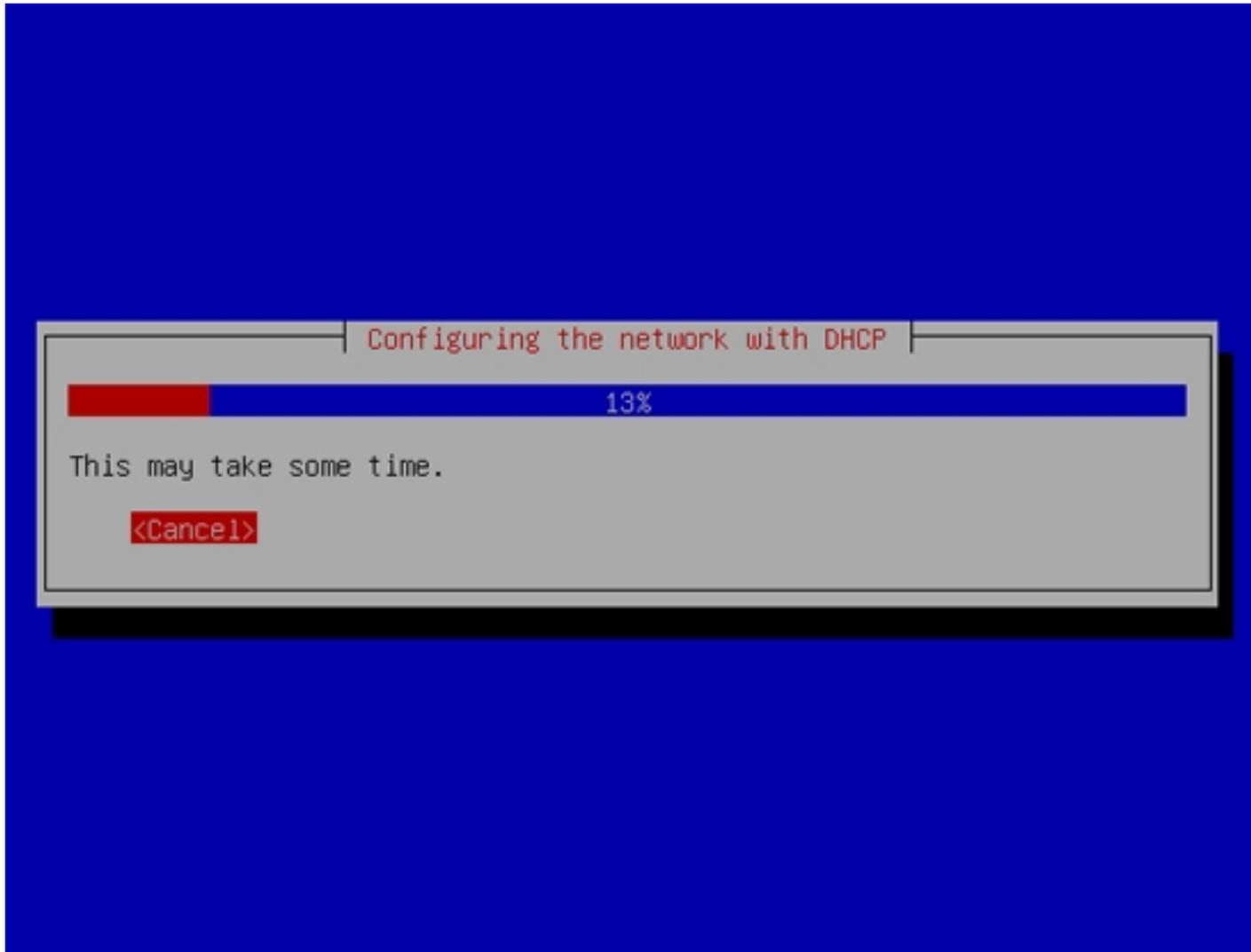
Choose a keyboard layout (you will be asked to press a few keys, and the installer will try to detect your keyboard layout based on the keys you pressed):



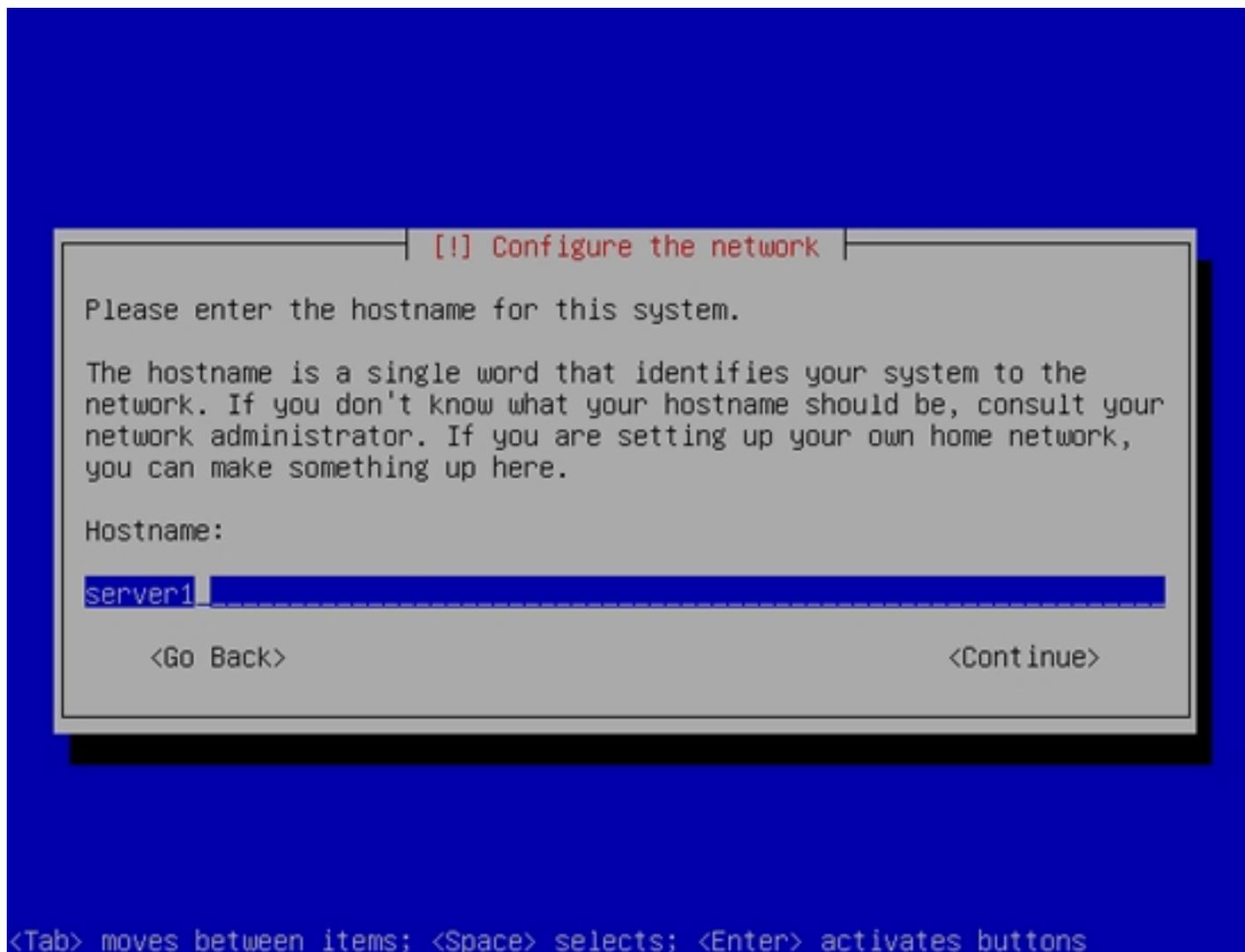
5. The installer checks the installation CD, your hardware, and configures the network with DHCP if there is a DHCP server in the network:



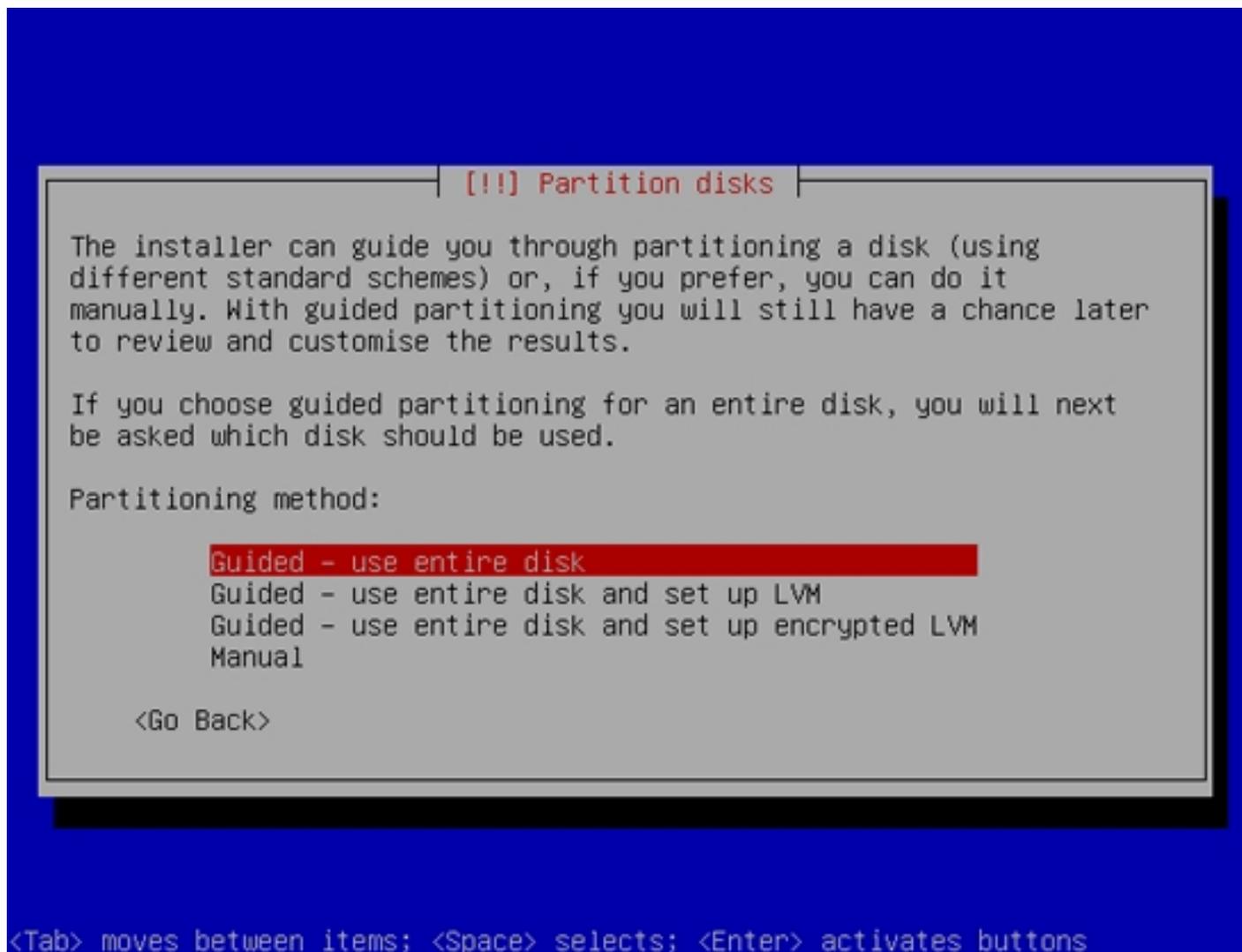




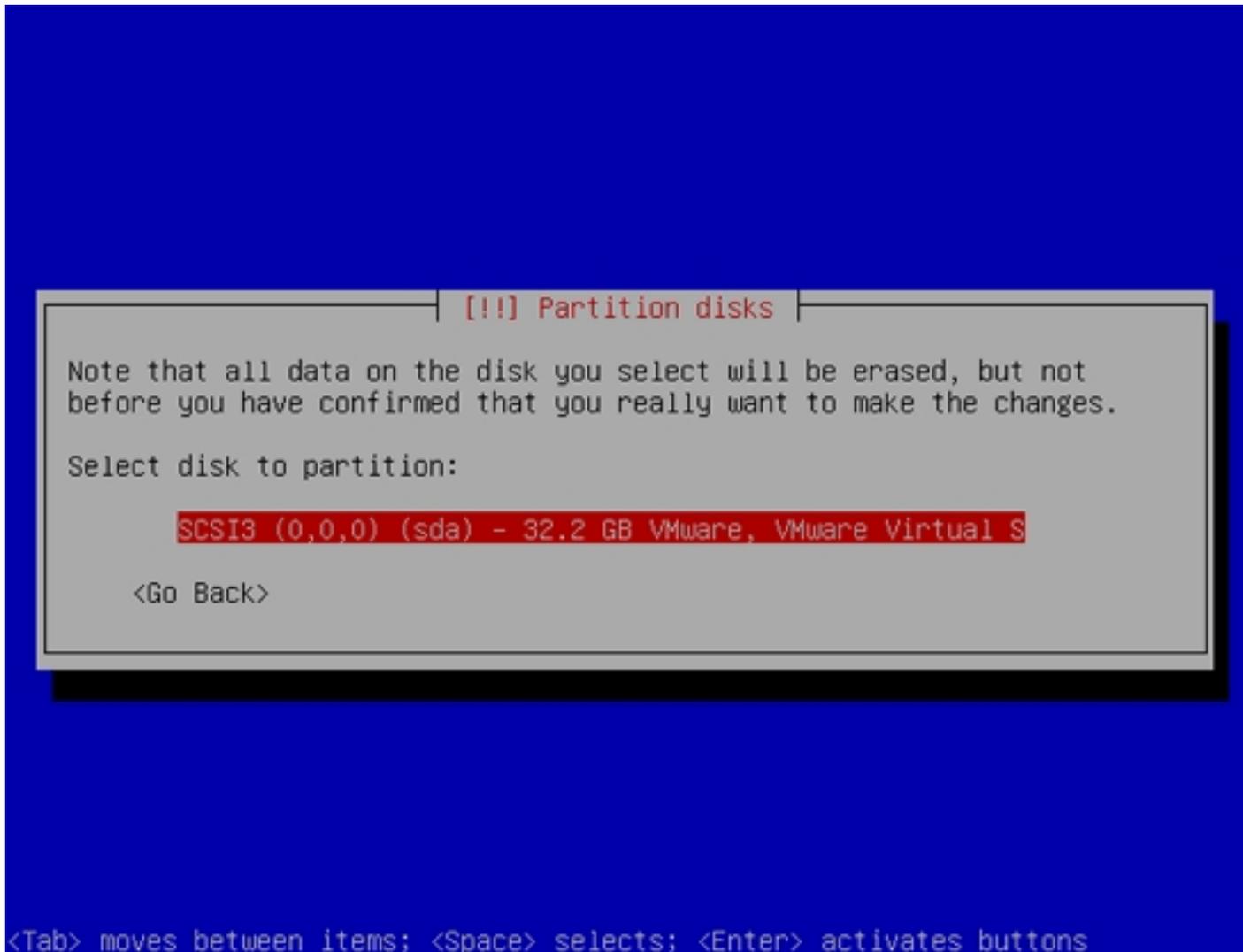
Enter the hostname. In this example, my system is called `server1.example.com`, so I enter `server1`:



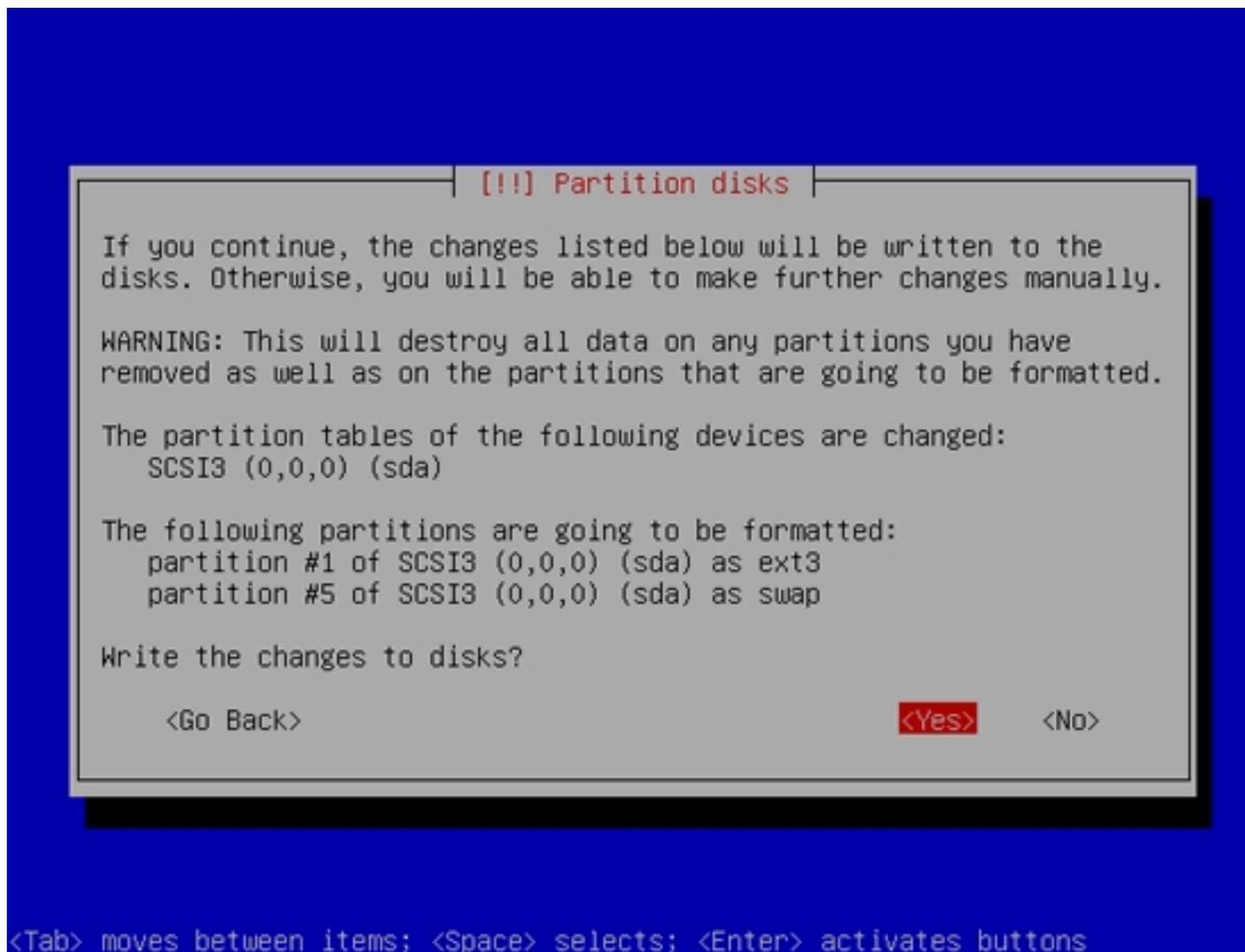
6. Now you have to partition your hard disk. For simplicity's sake I will create one big partition (with the mount point /) and a little swap partition so I select *Guided - use entire disk* (of course, the partitioning is totally up to you - if you like, you can create more than just one big partition, and you can also use LVM):



Select the disk that you want to partition:



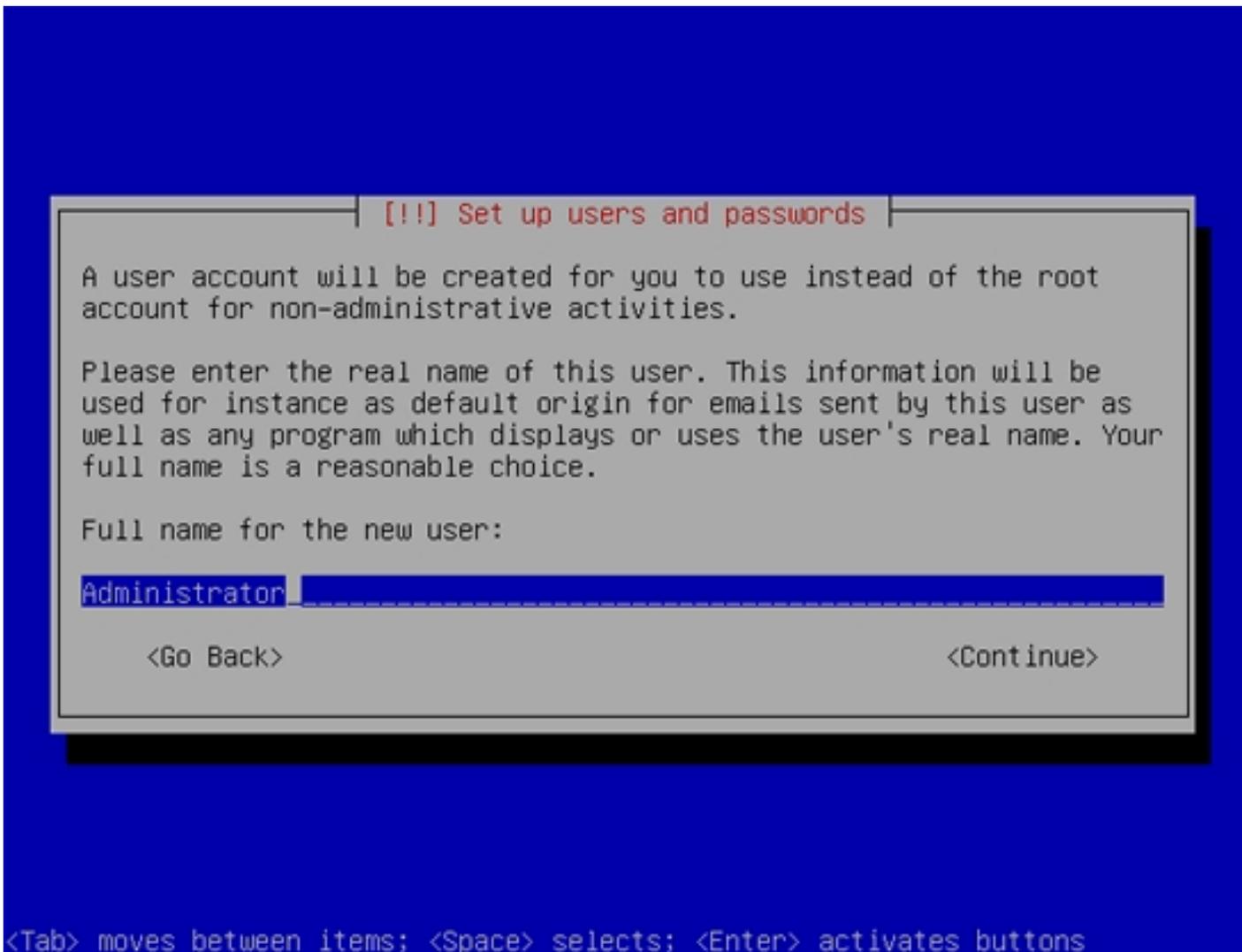
When you're finished, hit *Yes* when you're asked *Write the changes to disks?:*

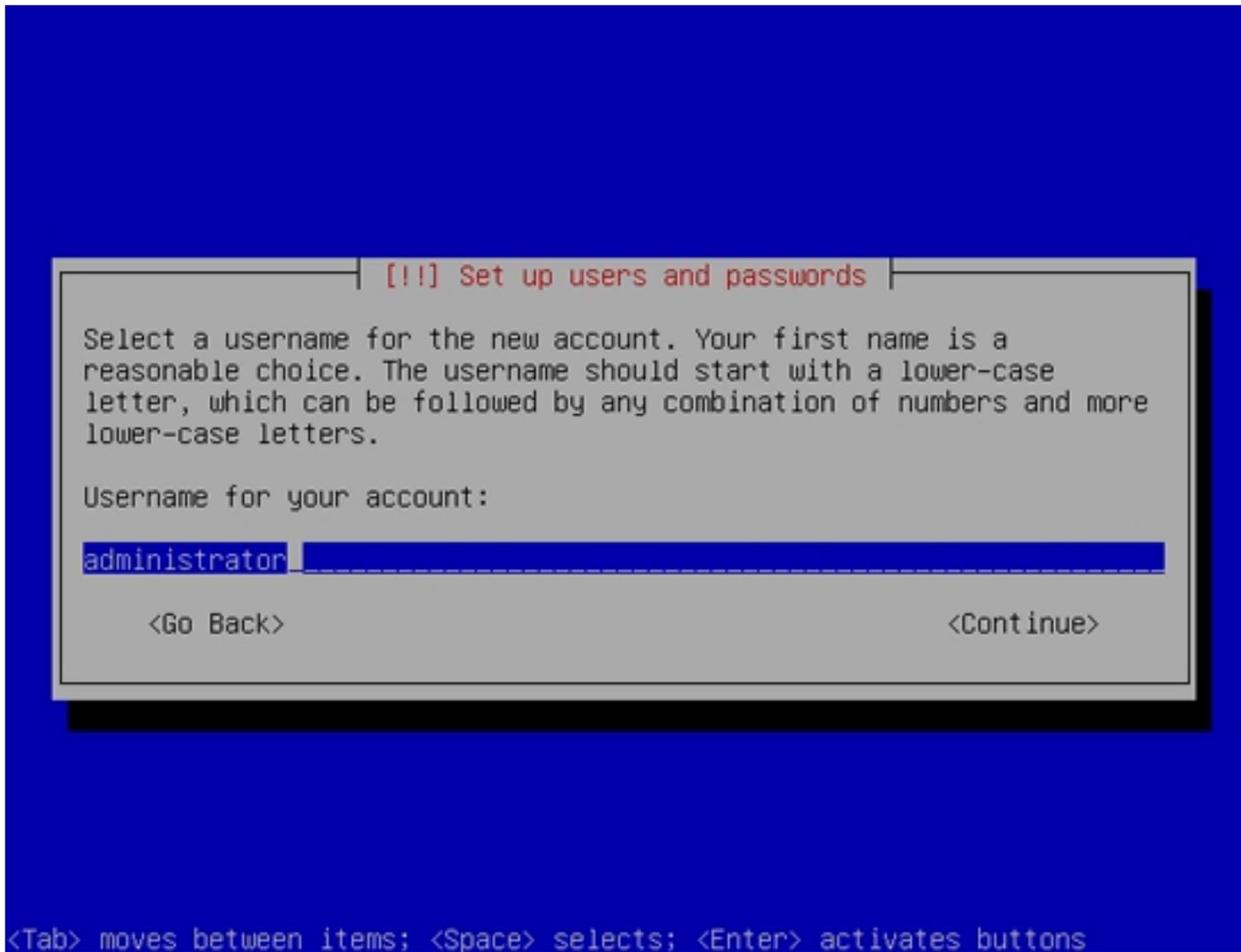


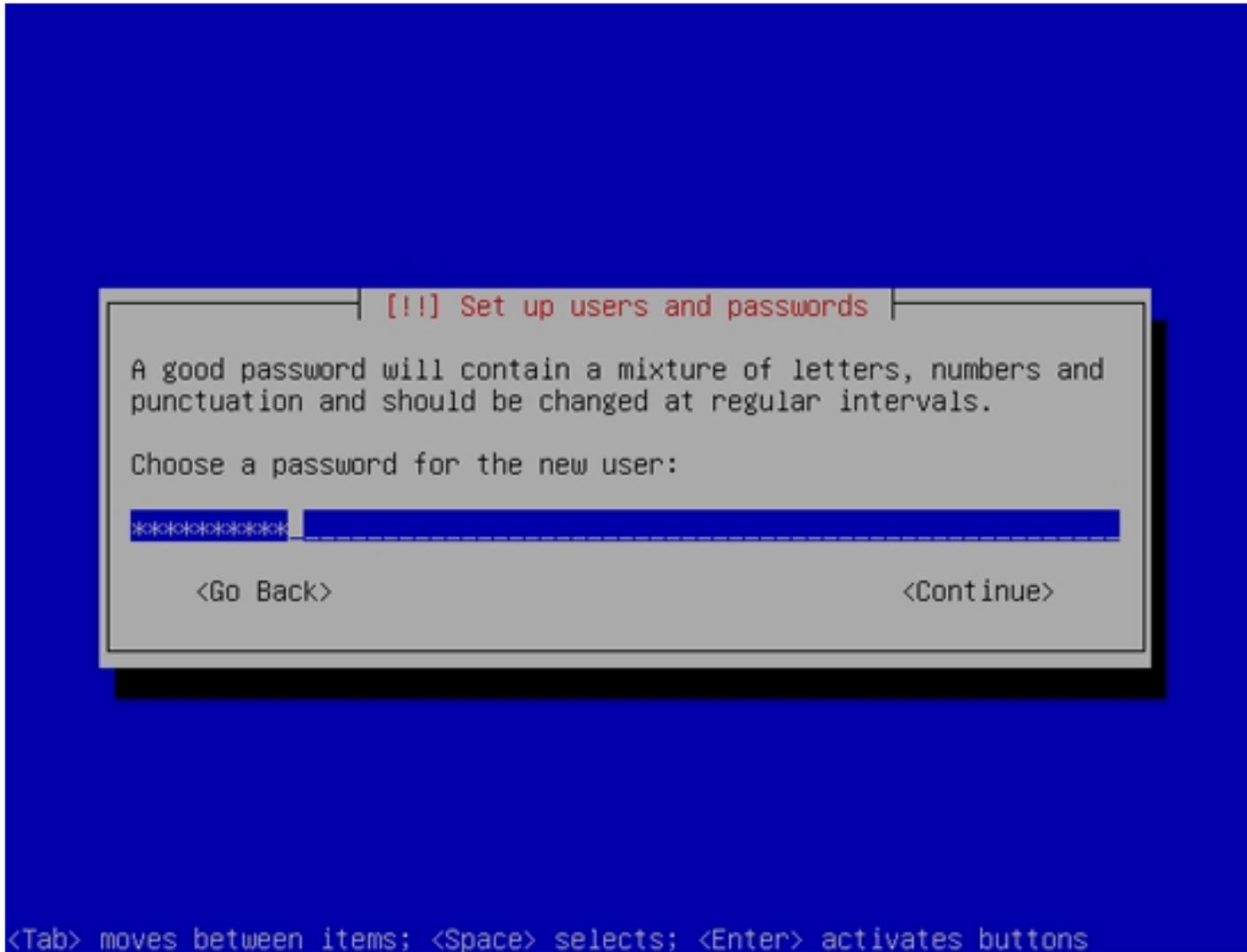
Afterwards, your new partitions are being created and formatted.

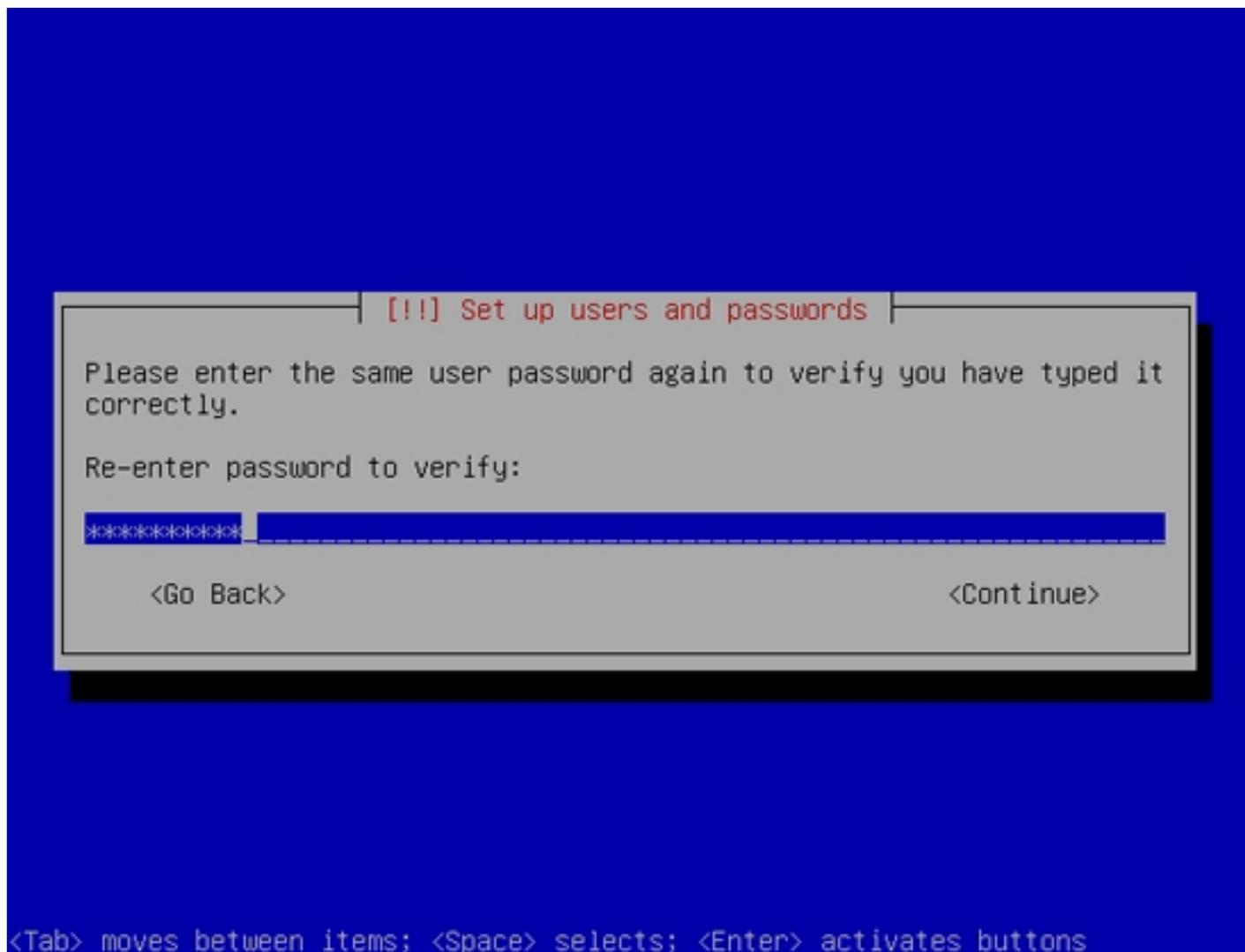
7. Create a user, for example the user *Administrator* with the user name *administrator* (don't use the user name *admin* as it is a reserved name on

Ubuntu 8.04):

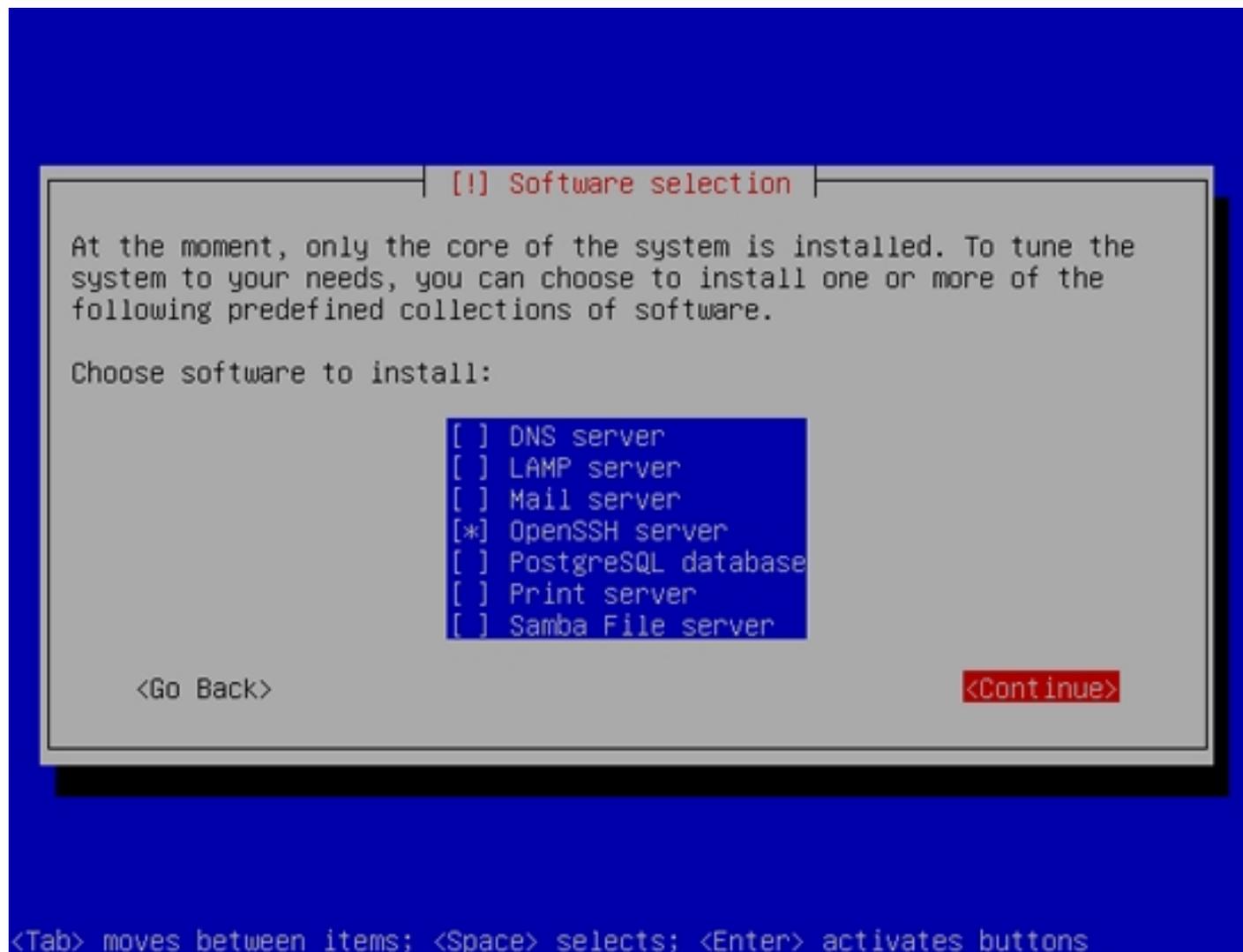




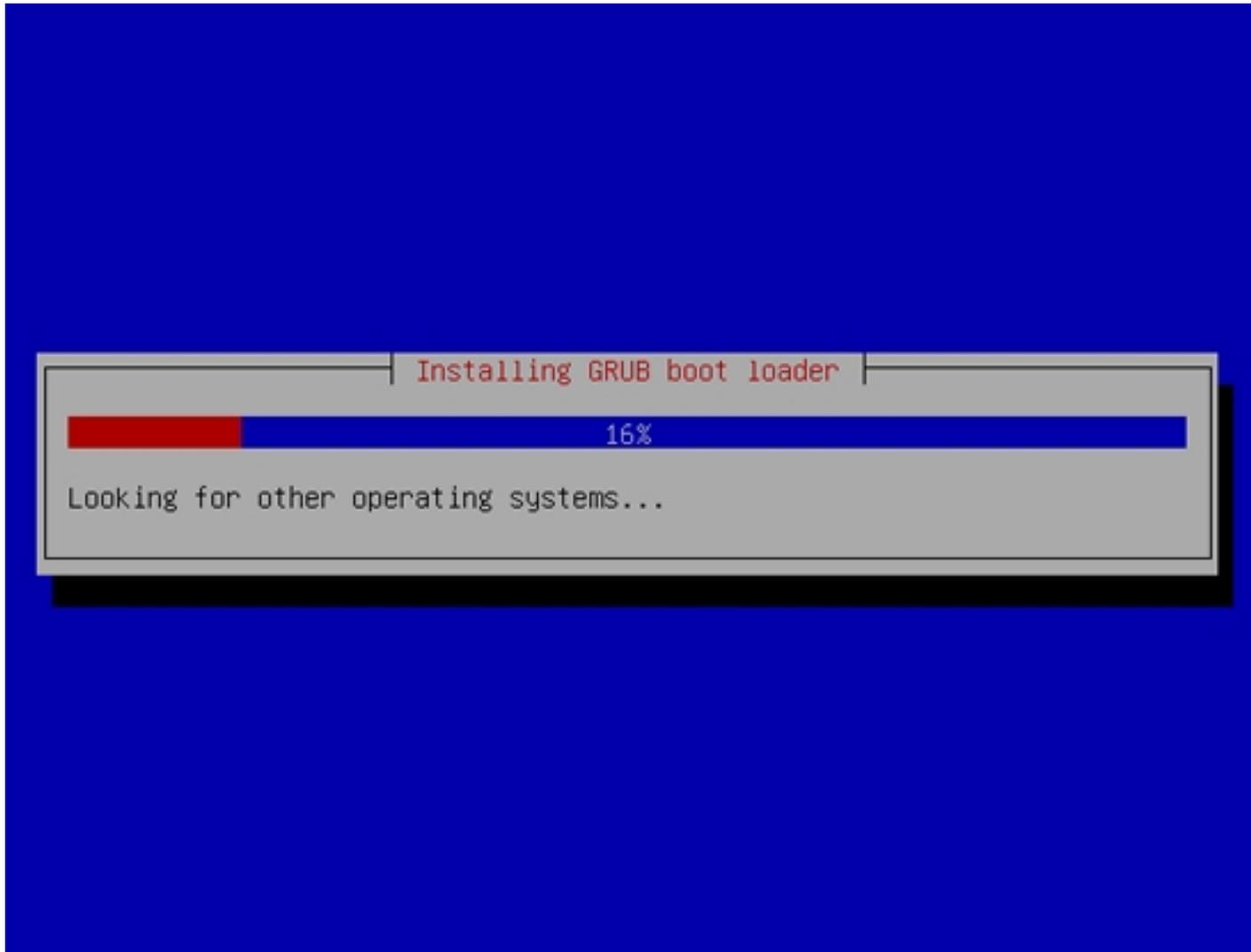




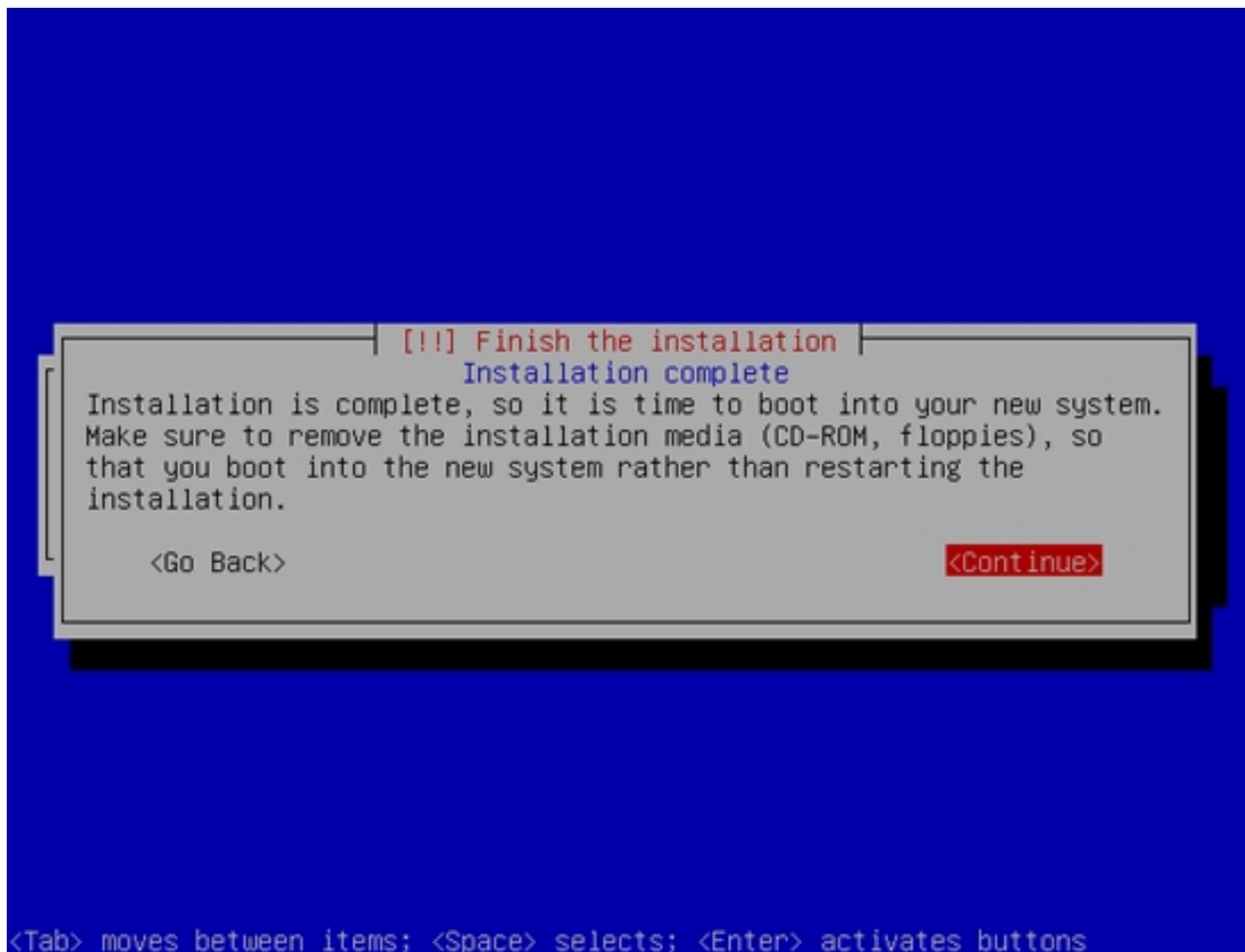
8. The only item I select here is OpenSSH server so that I can immediately connect to the system with an SSH client such as PuTTY after the installation has finished:



9. The GRUB boot loader gets installed:



10. The base system installation is now finished. Remove the installation CD from the CD drive and hit *Continue* to reboot the system:



1.3 Enable The root Account

After the reboot you can log in with your previously created username (e.g. *administrator*). Because we must run all the steps from this tutorial as root

user, we must enable the root account now. Run

```
sudo passwd root
```

and give root a password. Afterwards we become root by running

```
su
```

1.4 Install vim-full (Optional)

I'll use vi as my text editor in this tutorial. The default vi program has some strange behaviour on Ubuntu and Debian; to fix this, we install vim-full:

```
apt-get install vim-full
```

(You don't have to do this if you use a different text editor such as joe or nano.)

1.5 Configure The Network

Because the Ubuntu installer has configured our system to get its network settings via DHCP, we have to change that now because a server should have a static IP address. Edit `/etc/network/interfaces` and adjust it to your needs (in this example setup I will use the IP address `192.168.0.100`):

```
vi /etc/network/interfaces
```

```
# This file describes the network interfaces available on your system
# and how to activate them. For more information, see interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback
```

```
# The primary network interface
auto eth0
iface eth0 inet static
    address 192.168.0.100
    netmask 255.255.255.0
    network 192.168.0.0
    broadcast 192.168.0.255
    gateway 192.168.0.1
```

Then restart your network:

```
/etc/init.d/networking restart
```

Then edit `/etc/hosts`. Make it look like this:

```
vi /etc/hosts
```

```
127.0.0.1    localhost.localdomain localhost
192.168.0.100 server1.example.com  server1

# The following lines are desirable for IPv6 capable hosts
::1        ip6-localhost ip6-loopback
fe00::0    ip6-localnet
ff00::0    ip6-mcastprefix
ff02::1    ip6-allnodes
ff02::2    ip6-allrouters
ff02::3    ip6-allhosts
```

Now run

```
echo server1.example.com > /etc/hostname  
/etc/init.d/hostname.sh start
```

Afterwards, run

```
hostname  
hostname -f
```

Both should show *server1.example.com* now.

1.6 Edit `/etc/apt/sources.list` And Update Your Linux Installation

Edit `/etc/apt/sources.list`. Comment out or remove the installation CD from the file and make sure that the universe and multiverse repositories are enabled. It should look like this:

```
vi /etc/apt/sources.list
```

```
#  
# deb cdrom:[Ubuntu-Server 8.04 _Hardy Heron_ - Release i386 (20080423.2)]/ hardy main restricted  
  
#deb cdrom:[Ubuntu-Server 8.04 _Hardy Heron_ - Release i386 (20080423.2)]/ hardy main restricted  
# See http://help.ubuntu.com/community/UpgradeNotes for how to upgrade to  
# newer versions of the distribution.  
  
deb http://de.archive.ubuntu.com/ubuntu/ hardy main restricted  
deb-src http://de.archive.ubuntu.com/ubuntu/ hardy main restricted  
  
## Major bug fix updates produced after the final release of the
```

```
## distribution.
deb http://de.archive.ubuntu.com/ubuntu/ hardy-updates main restricted
deb-src http://de.archive.ubuntu.com/ubuntu/ hardy-updates main restricted

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## universe WILL NOT receive any review or updates from the Ubuntu security
## team.
deb http://de.archive.ubuntu.com/ubuntu/ hardy universe
deb-src http://de.archive.ubuntu.com/ubuntu/ hardy universe
deb http://de.archive.ubuntu.com/ubuntu/ hardy-updates universe
deb-src http://de.archive.ubuntu.com/ubuntu/ hardy-updates universe

## N.B. software from this repository is ENTIRELY UNSUPPORTED by the Ubuntu
## team, and may not be under a free licence. Please satisfy yourself as to
## your rights to use the software. Also, please note that software in
## multiverse WILL NOT receive any review or updates from the Ubuntu
## security team.
deb http://de.archive.ubuntu.com/ubuntu/ hardy multiverse
deb-src http://de.archive.ubuntu.com/ubuntu/ hardy multiverse
deb http://de.archive.ubuntu.com/ubuntu/ hardy-updates multiverse
deb-src http://de.archive.ubuntu.com/ubuntu/ hardy-updates multiverse

## Uncomment the following two lines to add software from the 'backports'
## repository.
## N.B. software from this repository may not have been tested as
## extensively as that contained in the main release, although it includes
## newer versions of some applications which may provide useful features.
## Also, please note that software in backports WILL NOT receive any review
## or updates from the Ubuntu security team.
# deb http://de.archive.ubuntu.com/ubuntu/ hardy-backports main restricted universe multiverse
# deb-src http://de.archive.ubuntu.com/ubuntu/ hardy-backports main restricted universe multiverse
```

```
## Uncomment the following two lines to add software from Canonical's
## 'partner' repository. This software is not part of Ubuntu, but is
## offered by Canonical and the respective vendors as a service to Ubuntu
## users.
# deb http://archive.canonical.com/ubuntu hardy partner
# deb-src http://archive.canonical.com/ubuntu hardy partner

deb http://security.ubuntu.com/ubuntu hardy-security main restricted
deb-src http://security.ubuntu.com/ubuntu hardy-security main restricted
deb http://security.ubuntu.com/ubuntu hardy-security universe
deb-src http://security.ubuntu.com/ubuntu hardy-security universe
deb http://security.ubuntu.com/ubuntu hardy-security multiverse
deb-src http://security.ubuntu.com/ubuntu hardy-security multiverse
```

Then run

```
apt-get update
```

to update the apt package database and

```
apt-get upgrade
```

to install the latest updates (if there are any).

1.7 Change The Default Shell

`/bin/sh` is a symlink to `/bin/dash`, however we need `/bin/bash`, not `/bin/dash`. Therefore we do this:

```
ln -sf /bin/bash /bin/sh
```

1.8 Disable AppArmor

AppArmor is a security extension (similar to SELinux) that should provide extended security. In my opinion you don't need it to configure a secure system, and it usually causes more problems than advantages (think of it after you have done a week of trouble-shooting because some service wasn't working as expected, and then you find out that everything was ok, only AppArmor was causing the problem).

We can disable it like this:

```
/etc/init.d/apparmor stop
update-rc.d -f apparmor remove
```

1.9 Install Some Software

Now we install a few packages that are needed later on. Run

```
apt-get install binutils cpp fetchmail flex gcc libarchive-zip-perl libc6-dev libcompress-zlib-perl libdb4.3-dev libpcre3 libpopt-dev lynx m4
make ncftp nmap openssl perl perl-modules unzip zip zlib1g-dev autoconf automake1.9 libtool bison autotools-dev g++ build-essential dpkg-dev
db4.3-util vim bzip2 perl-doc libwww-perl libdbi-perl libconvert-binhex-perl libmail-spf-query-perl rblcheck libnet-ident-perl tnef pax
libberkeleydb-perl unzoo arj lzop nomarch arc zoo libdb-file-lock-perl
```

(This command must go into one line!)

1.10 Install Unarj

```
cd /usr/src
wget http://http.us.debian.org/debian/pool/main/a/arj/unarj_3.10.21-2_all.deb
dpkg -i unarj_3.10.21-2_all.deb
```

1.11 Install Perl Modules(Pre-requisites)

Can be installed via `perl -MCPAN` or Webmin. I find that doing this through Webmin is better.

```
perl -MCPAN -e shell
install Module::Build
install Mail::SPF (Needed for SPF Checking)
install NetAddr::IP (Needed for SPF Checking)
install MLDBM::Sync this should also install MLDBM (Needed for MailWatch)
```

```
apt-get install libdbd-mysql-perl libapache-dbi-perl (Needed for MailWatch)
```

1.12 Webmin

```
apt-get install libauthen-pam-perl libio-pty-perl libmd5-perl libnet-ssleay-perl
```

Download latest webmin using the following command:

```
wget http://internap.dl.sourceforge.net/sourceforge/webadmin/webmin_1.410_all.deb
```

Now we have `webmin_1.410_all.deb` package; you need to install it using the following command:

```
dpkg -i webmin_1.410_all.deb
```

If your server complains that there is some library it does not find, just run the following command

```
apt-get install -f
```

You should now be able to login to Webmin at the URL `https://localhost:10000/`

1.13 Remove Programs

Now we also need to remove some programs, hopefully you don't need PCMCIA or printer support. This server will not need dial-up support either. You will not necessarily have all of these programs installed.

Uninstall the following software (all one line):

```
apt-get remove pcmciautils ubuntu-minimal pppoeconf ppp pppconfig
```

1.14 Cleaning up services

Some services might still linger even after uninstalling the daemons. First we need to backup inet.d:

```
cp -R /etc/init.d /etc/init.d.backup
```

Now we can stop all of the services that might be running which we don't need:

```
/etc/init.d/pcmciautils stop  
update-rc.d -f pcmciautils remove
```

Disable all of the services we stopped:

```
update-inetd --disable time
```

```
update-inetd --disable daytime
```

```
update-inetd --disable echo
```

```
update-inetd --disable chargen
```

```
update-inetd --disable ident
```

```
update-inetd --disable discard
```

The last one may ask you a question regarding "multiple entries", answer yes (y).

Check that we got everything:

```
lsof -i | grep LISTEN
```

The only daemon you should see at this point is `*:ssh` and `miniserv`. You may have to run this again:

```
update-inetd --disable discard
```

If there are other programs shown, try rebooting and test again.

2 DNS Server

Run

```
apt-get install bind9
```

For security reasons we want to run BIND chrooted so we have to do the following steps:

```
/etc/init.d/bind9 stop
```

Edit the file `/etc/default/bind9` so that the daemon will run as the unprivileged user `bind`, chrooted to `/var/lib/named`. Modify the line: `OPTIONS="-u bind"` so that it reads `OPTIONS="-u bind -t /var/lib/named"`:

```
vi /etc/default/bind9
```

```
OPTIONS="-u bind -t /var/lib/named"
```

Create the necessary directories under `/var/lib`:

```
mkdir -p /var/lib/named/etc  
mkdir /var/lib/named/dev  
mkdir -p /var/lib/named/var/cache/bind  
mkdir -p /var/lib/named/var/run/bind/run
```

Then move the config directory from `/etc` to `/var/lib/named/etc`:

```
mv /etc/bind /var/lib/named/etc
```

Create a symlink to the new config directory from the old location (to avoid problems when `bind` gets updated in the future):

```
ln -s /var/lib/named/etc/bind /etc/bind
```

Make null and random devices, and fix permissions of the directories:

```
mknod /var/lib/named/dev/null c 1 3  
mknod /var/lib/named/dev/random c 1 8  
chmod 666 /var/lib/named/dev/null /var/lib/named/dev/random  
chown -R bind:bind /var/lib/named/var/*
```

```
chown -R bind:bind /var/lib/named/etc/bind
```

We need to modify `/etc/default/syslogd` so that we can still get important messages logged to the system logs. Modify the line: `SYSLOGD=""` so that it reads `SYSLOGD="-a /var/lib/named/dev/log"`:

```
vi /etc/default/syslogd
```

```
SYSLOGD="-a /var/lib/named/dev/log"
```

Restart the logging daemon:

```
/etc/init.d/syslogd restart
```

Start up BIND, and check `/var/log/syslog` for errors:

```
/etc/init.d/bind9 start
```

3 MySQL

In order to install MySQL, we run

```
apt-get install mysql-server mysql-client libmysqlclient15-dev
```

You will be asked to provide a password for the MySQL root user - this password is valid for the user `root@localhost` as well as `root@server1.example.com`, so we don't have to specify a MySQL root password manually later on (as was the case with previous Ubuntu versions):

New password for the MySQL "root" user: [<-- yourrootsqlpassword](#)

We want MySQL to listen on all interfaces, not just localhost, therefore we edit `/etc/mysql/my.cnf` and comment out the line `bind-address = 127.0.0.1`:

```
vi /etc/mysql/my.cnf
```

```
#bind-address =127.0.0.1
```

Then we restart MySQL:

```
/etc/init.d/mysql restart
```

Now check that networking is enabled. Run

```
netstat -tap | grep mysql
```

The output should look like this:

```
tcp 0 0 *:mysql *.* LISTEN 5286/mysqld
```

4 Apache with PHP5 and Ruby

Now we install Apache:

```
apt-get install apache2 apache2-doc apache2-mpm-prefork apache2-utils libexpat1 ssl-cert
```

Next we install PHP5 and Ruby (both as Apache modules):

```
apt-get install libapache2-mod-php5 libapache2-mod-ruby php5 php5-common php5-curl php5-dev php5-gd php5-idn php-pear php5-imagick php5-imagick php5-imagick php5-imagick
```

```
php5-json php5-mcrypt php5-memcache php5-mhash php5-ming php5-mysql php5-pspell php5-recode php5-snmp php5-sqlite php5-tidy php5-xmlrpc php5-xml
php5-sqlite php5-tidy php5-xmlrpc php5-xml
```

You will be asked the following question:

Continue installing libc-client without Maildir support? <-- Yes

Next we edit `/etc/apache2/mods-available/dir.conf` and change the following:

```
vi /etc/apache2/mods-available/dir.conf
```

```
DirectoryIndex index.html index.htm index.shtml index.cgi index.php index.php3 index.pl index.xhtml
```

Now we have to enable some Apache modules (SSL, rewrite, suexec, and include):

```
a2enmod ssl
a2enmod rewrite
a2enmod suexec
a2enmod include
```

Reload the Apache configuration:

```
/etc/init.d/apache2 force-reload
```

4.1 Fix for Imagick

Because of a bug that causes the following error, the below must be done as a workaround:

PHP Warning: PHP Startup: Unable to load dynamic library '/usr/lib/php5/20060613/imagick.so' - libWand.so.9: cannot open shared object file: No such file or directory in Unknown on line 0

```
apt-get remove php5-imagick
```

```
apt-get install libmagick9-dev
```

```
pecl install imagick
```

Edit `/etc/php5/apache2/php.ini` and add the following:

```
vi /etc/php5/apache2/php.ini
```

```
extension=imagick.so
```

```
/etc/init.d/apache2 restart
```

5 Synchronize the System Clock

It is a good idea to synchronize the system clock with an NTP (network time protocol) server over the internet. Simply run

```
apt-get install ntp ntpdate
```

and your system time will always be in sync.

6 Setting up Postfix

```
apt-get install postfix postfix-pcre postfix-mysql postfix-ldap cabextract lha unrar razor pyzor spamassassin
```

You will be asked two questions. Answer as follows:

General type of mail configuration: <-- [Internet Site](#)

System mail name: <-- [server1.example.com](#)

Stop Postfix:

```
postfix stop
```

6.1 Edit master.cf

BTW watch for the two Postfix configuration files, both located in the `/etc/postfix` folder. More than one admin has gotten confused between `master.cf` and `main.cf`!

First back up the current master.cf:

```
cp /etc/postfix/master.cf /etc/postfix/master.cf-orig
```

Edit master.cf:

```
vi /etc/postfix/master.cf
```

We need to add two items below the pickup service type. The pickup service "picks up" local mail (local meaning "on this machine") and delivers it. This is a way to bypass content filtering for mail generated by this machine.

Add this just below the 'pickup' service type:

```
-o content_filter=  
-o receive_override_options=no_header_body_checks
```

It should look like this when you are done:

```
pickup fifo n - - 60 1 pickup  
-o content_filter=  
-o receive_override_options=no_header_body_checks
```

6.2 Edit main.cf

First we need to backup the main.cf file.

```
cp /etc/postfix/main.cf /etc/postfix/main.cf-orig
```

6.2.1 alias_maps

We simply need to make a correction to the default setting here:

```
postconf -e "alias_maps = hash:/etc/aliases"
```

Create the aliases file:

```
newaliases
```

Since our system will be configured not to store any local mails, this will be ignored.

6.2.2 myorigin

The domain name that mail created on this machine appears to come from. For example, if cron sends mail to "mnight@secretgovagency.gov" it will appear to come from "root@example.com".

```
postconf -e "myorigin = example.com"
```

Obviously, in the above, and all the following commands, replace my example parameters, like "example.com", with your own specific values.

6.2.3 myhostname

The fully-qualified domain name (FQDN) of the machine running the Postfix system.

```
postconf -e "myhostname = server1.example.com"
```

6.2.4 mynetworks

These are the machines I trust, and will relay mail for, to any destination. If you will be dealing with multiple internal mail servers, and/or want to allow several machines and/or subnets to relay through this server (careful!), just add them to this parameter in CIDR format and separate the networks like this:

```
postconf -e "mynetworks = 127.0.0.0/8, 192.168.0.0/24"
```

The `127.0.0.0/8` is there to allow the local server to send, you need to at least put this one in.

6.2.4.1 outbound trusted relay IP

If you'd like your SpamSnake to handle outgoing emails as well, be sure to add your local network to the list e.g. `192.168.0.0/24 172.16.0.0/16`. If your mailserver is `172.16.5.20` and you only want to trust only that IP, add `172.16.5.20/32`. You just have to setup your mailserver to relay (smarthost) to your SpamSnake.

6.2.5 message_size_limit

Maximum size email that Postfix will let in the "front door".

```
postconf -e "message_size_limit = 10485760"
```

The above allows email up to 10MB, the value is in bytes (10*1024*1024). Mail larger than this may possibly get bypassed by the anti-virus scanner (ClamAV). You could increase this if you also configure ClamAV to scan files larger than 10MB. If you allow messages larger than 10MB, keep an eye on RAM.

6.2.6 local_transport

Return an error message for local delivery attempts.

```
postconf -e "local_transport = error:No local mail delivery"
```

6.2.7 mydestination

An empty mydestination tells Postfix this machine is not the final destination.

```
postconf -e "mydestination = "
```

6.2.8 local_recipient_maps

An empty local_recipient_maps tells Postfix there are no local mailboxes.

```
postconf -e "local_recipient_maps = "
```

6.2.9 virtual_alias_maps

Our spamfilter must be able to receive mail for postmaster@yourIP. Reportedly, some things actually expect this ability to exist. We will also allow mail to

abuse@yourIP. Since we do not allow local mail delivery, mail addressed to our spamfilter's IP address will get rejected with an error message. Setting up `virtual_alias_maps` allows email to these two accounts to be forwarded to an inside address. Make sure your Exchange server is set up to receive messages addressed to "root", "postmaster" and "abuse".

Set up a reference to the virtual file:

```
postconf -e "virtual_alias_maps = hash:/etc/postfix/virtual"
```

Then edit the virtual file:

```
vi /etc/postfix/virtual
```

Add these lines to the top of the virtual file:

```
postmaster postmaster@example.com  
abuse abuse@example.com  
root root@example.com
```

Save and exit the file, then create the binary file that Postfix will use:

```
postmap /etc/postfix/virtual
```

6.2.10 relay_recipient_maps

We are going to build a table of every single user in every single domain that we accept mail for.

Set up a reference to a file we will create to store the data:

```
postconf -e "relay_recipient_maps = hash:/etc/postfix/relay_recipients"
```

Then edit `relay_recipients`:

```
vi /etc/postfix/relay_recipients
```

For the moment, we are going to accept mail for all users in our domain(s) so enter each domain you accept mail for in the following format:

```
@example.com OK  
@example2.com OK
```

Then create the binary file that Postfix will use:

```
postmap /etc/postfix/relay_recipients
```

The entries above are temporary. They are wildcards that allow mail to your domains. You **MUST** remove the entries above at some point in the near future and replace them with every single one of your valid recipients' email addresses. When you are ready to enter each user individually in the `relay_recipients` file, you would first remove (or comment out) the data above that allows mail to all users in the domain, and then list each user individually in the form:

```
admin@example.com OK  
admin@example2.com OK
```

6.2.11 transport_maps

Tells Postfix where to look for a transport file. We use the transport file to tell Postfix where to forward valid mail for our domain(s). Setting up transport is similar to setting up `relay_recipients`.

Create a reference to it in `main.cf`:

```
postconf -e "transport_maps = hash:/etc/postfix/transport"
```

Then edit transport:

```
vi /etc/postfix/transport
```

Add 1 new line for each domain for which you will be handling mail, similar to the example below. The IP address is that of whatever server is the final destination of messages addressed to our domain(s) (our Exchange server). It does not matter where you place these items in the file, but I like to put them at the top.

```
example.com smtp:[192.168.0.x]  
example2.com smtp:[192.168.0.x]
```

Include the brackets on these lines!. You can also use FQDN hostname instead of an IP address (i.e. `smtp:[exchange1.example.com]`).

Now to create the binary file Postfix will use:

```
postmap /etc/postfix/transport
```

6.2.12 relay_domains

What destination domains (and subdomains thereof) this system will relay mail for.

```
postconf -e "relay_domains = hash:/etc/postfix/relay_domains"
```

Edit relay_domains:

```
vi /etc/postfix/relay_domains
```

Add 1 new line for each domain for which you will be handling mail, similar to the example below:

```
example.com OK
example2.com OK
```

This file currently has a very similar format to *relay_recipients* do not mistake the two. This file cannot have '@' in front of the domain name. Just thought I'd mention it, some very smart people have been known to have done this...

Then create the binary file Postfix will use:

```
postmap /etc/postfix/relay_domains
```

6.3 Postfix Anti-Spam Settings

6.3.1 smtpd_helo_required

Make any connecting mail server do a proper smtp "handshake" and announce its name. Internet RFCs require this, so we do too.

```
postconf -e "smtpd_helo_required = yes"
```

I also changed the `smtpd_banner` to "\$myhostname ESMTP \$mail_name SpamSnake".

Preface: Postfix' restriction stages are as follows, and are processed in the following order:

```
smtpd_client_restrictions
smtpd_helo_restrictions
smtpd_sender_restrictions
smtpd_recipient_restrictions
smtpd_data_restrictions
```

We are only going to place entries in the last three restriction stages. Restriction stages are processed in this order regardless of the order listed in `main.cf`.

6.3.2 smtpd_sender_restrictions

This restriction stage restricts what sender addresses this system accepts in MAIL FROM: commands (the envelope sender). We will place three tests (restrictions) in this restriction stage.

6.3.4 check_sender_access (Optional)

Here we ask Postfix to compare the envelope sender to entries in an `/etc/postfix/sender_access` database and act upon those entries if a match is found. We also define what action is taken there (OK, DUNNO, REJECT etc.) on a sender by sender basis. If the sender is not listed in the file, the test evaluates to DUNNO, and the next test is performed.

6.3.5 reject_non_fqdn_sender

Reject when the envelope sender mail address is not in the proper format.

6.3.6 reject_unknown_sender_domain

Reject when the envelope sender's domain part of the mail address has no DNS "A" or "MX" record at all. On occasion, you will see in a report that someone you wish to receive mail from has been rejected by this setting. One possible cause of this is when legitimate senders deliberately use bogus domain names so you will not reply to them. This is where the sender access list comes in handy. You can give them an OK there, and this test will be bypassed.

Now to implement these three restrictions:

```
postconf      -e      "smtpd_sender_restrictions      =      check_sender_access      hash:/etc/postfix/sender_access,      reject_non_fqdn_sender,      reject_unknown_sender_domain"
```

6.3.7 smtpd_recipient_restrictions

The access restrictions that the Postfix SMTP server applies in the context of the RCPT TO: command. This refers to the "envelope recipient" which is what the client gave in the "RCPT TO:" line during the SMTP session, not the header "To:" line. Let's look at those specific restrictions (tests) we place in `smtpd_recipient_restrictions`:

6.3.8 permit_mynetworks

Allows machines listed in "mynetworks" to skip the rest of the tests in this restriction stage (permit = OK). In other words, it exits this stage and is tested in the next stage (smtpd_data_restrictions). Because permit_mynetworks is placed in front of reject_unauth_destination, this means machines in \$mynetworks are allowed to relay mail to any domain. Without this, we would only be able to send mail to our own domain(s). If the IP address of the sender is not listed in \$mynetworks, the test evaluates to "DUNNO" and continues on to the next test (reject_unauth_destination).

6.3.9 reject_unauth_destination & reject_unknown_recipient_domain

This, along with permit_mynetworks is used for relay control. This setting, in essence, means that mail bound for any domain that we have not configured our machine to accept mail for will be rejected. In our case Postfix will use the relay_domains setting (or table) that we configured earlier to determine what domains those are. If the domain is listed in relay_domains, this test evaluates to "DUNNO" and the session is allowed to go on to the next test (if any).

6.3.10 reject_unauth_pipelining

Rejects bulk mailers that attempt to use pipelining to speed delivery, without checking if it is supported first (non-RFC, common among spammers).

Now to implement these three restrictions:

```
postconf -e "smtpd_recipient_restrictions = reject_non_fqdn_sender, reject_unknown_sender_domain, reject_non_fqdn_recipient, reject_unknown_recipient_domain, permit_mynetworks, reject_unauth_destination, reject_unauth_pipelining, reject_invalid_helo_hostname, reject_non_fqdn_helo_hostname, reject_rbl_client zen.spamhaus.org"
```

6.3.11 smtpd_data_restrictions

Optional access restrictions that the Postfix SMTP server applies in the context of the SMTP DATA: command. Like smtpd_recipient_restrictions, this is a restriction stage.

6.3.12 reject_unauth_pipelining

I repeat this setting in smtpd_data_restrictions as it is not always effective when placed in smtpd_recipient_restrictions. I include it in smtpd_recipient_restrictions as I like to place it prior to any policy servers. Note that there are only a couple of restrictions that make good use of smtpd_data_restrictions.

```
postconf -e "smtpd_data_restrictions = reject_unauth_pipelineing"
```

6.3.13 /etc/postfix/sender_access

We referenced this file in `smtpd_sender_restrictions`. We use this file to check the sender right at the front door. In this file, we'll list certain senders/domains/IPaddress ranges for special handling. Below are bogus examples, create your own as you see fit. Please read `/etc/postfix/sender_access` for more information. Although you could use this file for various purposes, considering the way we have set this up in `smtpd_sender_restrictions`, I suggest using it to either blacklist senders, or allow certain senders to bypass the remaining tests in `smtpd_sender_restrictions`.

```
vi /etc/postfix/sender_access
```

```
#Example sender access map file
makeabuck@mlm.tld 550 No MLM thanks
allspam.tld 550 Spam is not accepted here
badguy.net REJECT
justaspamminfool@allspamallthetime.com REJECT
newsletter-favorite-lug.org OK
my-really-1337-test-domain.com OK
```

Since this is a hash table, you need to postmap it as usual:

```
postmap /etc/postfix/sender_access
```

6.3.14 Final Look at the Postfix Install

Review changes:

```
less /etc/postfix/main.cf
```

Check the contents of the file for errors and repair if needed. Fire up Postfix:

```
postfix start
```

Check that Postfix responds:

```
telnet 127.0.0.1 25
```

You should see:

```
220 [yourFQDNhere] ESMTP Postfix (Ubuntu)
```

Hit [enter] a few times; then type *quit* to exit.

If it does not reply in this manner, open another terminal window and stop Postfix:

```
postfix stop
```

Make sure you ran *newaliases* and all the *postmap* commands above. Check all the settings in *main.cf* and *master.cf*. Any time you make changes to *master.cf* or *main.cf* or to data tables, most (not all) of the time, it is required that you to reload Postfix with:

```
postfix reload
```

7 Pyzor, Razor, DCC, SpamAssassin and MailScanner Configuration 7.1 Install MailScanner

Install MailScanner Dependencies by doing the following:

```
apt-get install libconvert-tnef-perl libdbd-sqlite3-perl libfilesys-df-perl libmailtools-perl libmime-tools-perl libmime-perl libnet-cidr-perl  
libsys-syslog-perl libio-stringy-perl libfile-temp-perl
```

Install MailScanner from the Debian .deb Source:

```
wget http://debian.intergenia.de/debian/pool/main/m/mailscanner/mailscanner_4.68.8-1_all.deb

dpkg -i mailscanner_4.68.8-1_all.deb
```

7.2 Pyzor Configuration

We need to change some permissions on pyzor first:

```
chmod -R a+rX /usr/share/doc/pyzor /usr/bin/pyzor /usr/bin/pyzord

chmod -R a+rxX /usr/share/python-support/pyzor
```

Here we supply the IP address of the Pyzor server to Pyzor. This will create the server's IP address in a servers file therein. Then it will test the connection. If you are behind a firewall, open port 24441/udp in and out to your server. While you're at it also open up 6277/udp for DCC, 2703/tcp for Razor and 783/tcp for SpamAssassin:

```
pyzor --homedir /var/lib/MailScanner discover

pyzor ping
```

7.3 Razor Configuration

Create the .razor configuration:

```
cd

rm /etc/razor/razor-agent.conf
```

```
mkdir /var/lib/MailScanner/.razor

razor-admin -home=/var/lib/MailScanner/.razor -create

razor-admin -home=/var/lib/MailScanner/.razor -discover

razor-admin -home=/var/lib/MailScanner/.razor -register

chown -R postfix:www-data /var/lib/MailScanner

chmod -R ug+rx /var/lib/MailScanner
```

Make the following changes to `/var/lib/MailScanner/.razor/razor-agent.conf`:

```
vi /var/lib/MailScanner/.razor/razor-agent.conf
```

Change `debuglevel = 3` to `debuglevel = 0` (yes zero not "o"). This will prevent Razor from filling up your drive with debug information. Those two lines should look like this when done:

```
debuglevel = 0
razorhome = /var/lib/MailScanner/.razor/
```

7.4 DCC Setup and Configuration

Install DCC from .deb source:

```
wget http://launchpadlibrarian.net/11564361/dcc-server_1.3.42-5_i386.deb

wget http://launchpadlibrarian.net/11564359/dcc-common_1.3.42-5_i386.deb
```

```
dpkg -i dcc-common_1.3.42-5_i386.deb
```

```
dpkg -i dcc-server_1.3.42-5_i386.deb
```

We are not running a DCC server, so we don't need to waste time checking ourselves.

Once the installation is done run:

```
cdcc "delete 127.0.0.1"
```

```
cdcc "delete 127.0.0.1 Greylist"
```

Test our installation with:

```
cdcc info
```

You should get 'requests ok' from the servers.

8 Configuring MailScanner and ClamAV8.1 Stop Postfix:

```
postfix stop
```

Install the packages:

```
apt-get install clamav clamav-daemon
```

Update ClamAV virus defenitions:

```
freshclam
```

Once that is done, we need to make a directory for SpamAssassin in the spool and give postfix permissions to it, if you run `sa-learn --force` as root, bayes database that is stored in these directories will change to `root:root` and `spamassassin` will error looking at the db. Just keep an eye on the `mail.log` and you'll remember to change the permissions back. Also disable the MailScanner default configs:

```
mkdir /var/spool/MailScanner/spamassassin
```

Backup your `MailScanner.conf` file:

```
cp /etc/MailScanner/MailScanner.conf /etc/MailScanner/MailScanner.conf.back
```

Edit `MailScanner.conf`:

```
vi /etc/MailScanner/MailScanner.conf
```

Change the following parameters in `MailScanner.conf`:

```
%org-name% = ORGNAME
%org-long-name% = ORGFULLNAME
%web-site% = ORGWEBSITE
Run As User = postfix
Run As Group = postfix
Incoming Queue Dir = /var/spool/postfix/hold
Outgoing Queue Dir = /var/spool/postfix/incoming
MTA = postfix
Virus Scanners = clamav
Spam Subject Text = ***SPAM***
Send Notices = no
Spam List = spamcop.net SBL+XBL
Required SpamAssassin Score = 6
High SpamAssassin Score = 10
```

```
Spam Actions = deliver
High Scoring Spam Actions = delete
Rebuild Bayes Every = 0
Wait During Bayes Rebuild = no
SpamAssassin User State Dir = /var/spool/MailScanner/spamassassin
```

The first 9 lines are basically required in order for everything to work, the rest are recommended.

8.2 header_checks & body_checks

Let's go ahead and put this in main.cf. header_checks is required because it allows us to hold all incoming email in order for MailScanner to do its thing:

```
postconf -e "header_checks = regexp:/etc/postfix/header_checks"
```

Edit header_checks:

```
vi /etc/postfix/header_checks
```

Add this line to the header_checks file, without it MailScanner will not work:

```
/^Received:/ HOLD
```

8.3 Fix to Disable Permission Checks on MailScanner Directories

Comment out the lines that check directory permissions on `/var/*` in `/etc/rc2.d/S20mailscanner`.

In the file `/etc/default/mailscanner`, make sure this parameter is at 1:

```
vi /etc/default/mailscanner
```

```
run_mailscanner=1
```

8.4 MailScanner Webmin Plugin (Optional)

Login to Webmin, <https://localhost:10000>, and install the MailScanner module for webmin found at <http://internap.dl.sourceforge.net/sourceforge/msfrontend/webmin-module-1.1-4.wbm>. After this is done, you'll have to enter the following into your mailscanner module to get it to work:

Full path to MailScanner program `/etc/init.d/mailscanner`

Full path and filename of MailScanner config file `/etc/MailScanner/MailScanner.conf`

Full path to the MailScanner bin directory `/usr/sbin`

Full path and filename for the MailScanner pid file `/var/run/MailScanner/MailScanner.pid`

Command to start MailScanner `/etc/init.d/mailscanner start`

Command to stop MailScanner `/etc/init.d/mailscanner stop`

8.5 You can now start the system

```
/etc/init.d/mailscanner start
```

```
/etc/init.d/postfix start
```

Check your logs for errors:

```
tail -f /var/log/mail.log
```

9 MailWatch Installation Instructions

This setup assumes you are using Apache v2.x and not Apache v1.x.

9.1 Before Starting

Make sure that MailScanner is working before you continue with the MailWatch install!

Notes for Ubuntu:

You must have a working MailScanner set-up and running copies of MySQL, Apache, and PHP. You must also have the Perl DBD-MySQL package installed for the Perl portions of MailScanner to utilize the MySQL database.

The default php.ini set should have the following set correctly, you may want to check this:

- short_open_tag = On
- safe_mode = Off
- register_globals = Off
- magic_quotes_gpc = On
- magic_quotes_runtime = Off
- session.auto_start = 0

These will be commented out you must remove the "#" to activate them:

- extension=mysql.so
- extension=gd.so

9.2 Installation

All commands below should be run as root.

9.3 Download the latest MailWatch release

```
wget http://downloads.sourceforge.net/mailwatch/mailwatch-1.0.4.tar.gz?modtime=1178902008&big_mirror=0
tar xzvf mailwatch-1.0.4.tar.gz
cd mailwatch-1.0.4
```

9.4 Create the database

```
mysql -p < create.sql
```

NOTE: you will need to modify the above as necessary for your system if you have a root password for your MySQL database (recommended!) - Debian will ask for one.

9.5 Create a MySQL user and password & Set-up MailScanner for SQL logging

```
mysql -p
mysql> GRANT ALL ON mailscanner.* TO mailwatch@localhost IDENTIFIED BY 'password';
```

Remember the password! You need the single quotes ' to surround your password.

9.6 Edit and copy MailWatch.pm

Edit MailWatch.pm and change the \$db_user and \$db_pass values accordingly and move MailWatch.pm.

```
mv MailWatch.pm /etc/MailScanner/CustomFunctions/
```

9.7 Create a MailWatch Web User

```
mysql mailscanner -u mailwatch -p
```

Enter password: *****

```
mysql> INSERT INTO users VALUES ('username',md5('password'),'mailscanner','A','0','0','0','0','0');
```

9.8 Install & Configure MailWatch

From within the unpacked mailwatch directory move the directory called 'mailscanner' to the web server's root.

```
mv mailscanner/ /var/www/
cd /var/www/mailscanner
```

Make a temp directory:

```
mkdir temp
chgrp www-data temp
chmod g+w temp
```

Check the permissions of `/var/www/mailscanner/images` and `/var/www/images/cache` - they should be `ug+rwX` and owned by `root` and in the same group as the web server user.

```
chown root:www-data images
chmod ug+rwX images
chown root:www-data images/cache
chmod ug+rwX images/cache
```

Create `conf.php` by copying `conf.php.example` and edit the values to suit, you will need to set `DB_USER` and `DB_PASS` to the MySQL user and password that you created earlier.

Change these values as shown below:

```
# define(DB_USER, 'mailwatch');
# define(DB_PASS, 'password');
# define(MAILWATCH_HOME, '/var/www/mailscanner');
# define(MS_LIB_DIR, '/usr/share/MailScanner/');
# define(QUARANTINE_USE_FLAG, true);
```

9.9 Set-up MailScanner

Next edit `/etc/MailScanner/MailScanner.conf`.

```
vi /etc/MailScanner/MailScanner.conf
```

You need to make sure that the following options are set:

- Quarantine User = root

- Quarantine Group = www-data
- Quarantine Permissions = 0660
- Quarantine Whole Message = yes
- Always Looked Up Last = &MailWatchLogging

And check these as well:

- Quarantine Whole Message As Queue Files = no
- Detailed Spam Report = yes
- Include Scores In SpamAssassin Report = yes

Spam Actions, High Scoring Spam Actions and No Spam Actions should also have 'store' as one of the keywords if you want to quarantine those items for bayes learning or viewing from within MailWatch.

9.10 Integrate SQL Blacklist/Whitelist (optional)

If you would like to manage the MailScanner whitelist and blacklist from within the MailWatch web interface perform the following steps.

1. Edit the MySQL connection values within the CreateList subroutine of SQLBlackWhiteList.pm to match the values you entered previous into MailWatch.pm. Both files should contain the same values. (Look for the following lines in SQLBlackWhiteList.pm and enter your own data.)

```
my($db_user) = 'mailwatch';  
my($db_pass) = 'password';
```

2. Copy SQLBlackWhiteList.pm to /etc/MailScanner/CustomFunctions/.

3. Edit MailScanner.conf and set:

- Is Definitely Not Spam = &SQLWhitelist
- Is Definitely Spam = &SQLBlacklist

9.11 Fix to allow MailWatch to work with Postfix Inbound/Outbound Queue

Download the patch from <http://www.gbnetwork.co.uk/mailscanner/postfixmail.tar.gz>

```
cd /usr/src
wget http://www.gbnetwork.co.uk/mailscanner/files/postfixmail.tar.gz
tar xvfz postfixmail.tar.gz
cd postfixmail
cp postfix* /var/www/mailscanner
patch /var/www/mailscanner/functions.php functions.php.diff
```

9.12 SpamAssassin

First we need to disable the default SpamAssassin configuration file:

```
mv /etc/spamassassin/local.cf /etc/spamassassin/local.cf.disabled
```

Now let's backup the SpamAssassin configuration file in MailScanner then edit:

```
cp /etc/MailScanner/spam.assassin.prefs.conf /etc/MailScanner/spam.assassin.prefs.conf.back
```

Add pyzor and razor paths:

```
vi /etc/MailScanner/spam.assassin.prefs.conf
```

Add these lines to the top of spam.assassin.prefs.conf:

```
pyzor_options --homedir /var/lib/MailScanner/
razor_config /var/lib/MailScanner/.razor/razor-agent.conf
```

9.13 Move the Bayesian Databases and set-up permissions (skip this if you don't use bayes)

Edit /etc/MailScanner/spam.assassin.prefs.conf and set:

```
vi /etc/MailScanner/spam.assassin.prefs.conf
```

```
bayes_path /etc/MailScanner/bayes/bayes
bayes_file_mode 0660
```

Look for these lines and change them accordingly:

```
bayes_ignore_header X-YOURDOMAIN-COM-MailScanner
bayes_ignore_header X-YOURDOMAIN-COM-MailScanner-SpamCheck
bayes_ignore_header X-YOURDOMAIN-COM-MailScanner-SpamScore
bayes_ignore_header X-YOURDOMAIN-COM-MailScanner-Information
```

"YOURDOMAIN-COM" should be replaced with whatever you used for "%org-name%" in the MailScanner.conf file. Leave the "X-" in place. This is the same orname used in the MailScanner.conf above.

Create the 'new' bayes directory, make the directory owned by the same group as the web server user and make the directory setgid:

```
mkdir /etc/MailScanner/bayes
chown -R root:www-data /etc/MailScanner/bayes
chmod -R ug+rw /etc/MailScanner/bayes
chmod g+s /etc/MailScanner/bayes
```

Copy the existing bayes databases and set the permissions (**Note:** This part can be skipped if bayes was not previously enabled because the bayes directory would not have been created):

```
cp /var/lib/MailScanner/bayes_* /etc/MailScanner/bayes
chown root:www-data /etc/MailScanner/bayes/bayes_*
chmod g+rw /etc/MailScanner/bayes/bayes_*
```

Make sure that "bayes_auto_expire 0" is not commented out in spam.assassin.prefs.conf:

```
bayes_auto_expire 0
```

Edit the SpamAssassin v310.pre to enable Razor and DCC:

```
vi /etc/spamassassin/v310.pre
```

Uncomment the following lines:

```
loadplugin Mail::SpamAssassin::Plugin::DCC
loadplugin Mail::SpamAssassin::Plugin::Razor2
```

If you want then you can test SpamAssassin to make sure that it is using the new databases correctly:

```
spamassassin -D -p /etc/MailScanner/spam.assassin.prefs.conf --lint
```

and you should see something like:

```
debug: using "/etc/MailScanner/spam.assassin.prefs.conf" for user prefs file
debug: bayes: 28821 tie-ing to DB file R/O /etc/MailScanner/bayes/bayes_toks
debug: bayes: 28821 tie-ing to DB file R/O /etc/MailScanner/bayes/bayes_seen
debug: bayes: found bayes db version 2
debug: Score set 3 chosen.
```

9.13.1 SpamAssassin Bayes Database to SQL Conversion

Pre-requisites

a. You'll need the perl-DBI and perl-DBD-MySQL modules installed.

Assumptions and Variables:

SpamAssassin Bayes Database Name: sa_bayes

SpamAssassin Bayes Database UserName: sa_user

SpamAssassin Bayes Database Password: sa_password

Create the MySQL database:

First of all, create a database on the server where you intend on storing the bayesian information.

```
mysql -u root -p
```

```
mysql> create database sa_bayes;  
mysql> GRANT ALL ON sa_bayes.* TO sa_user@localhost IDENTIFIED BY 'sa_password';  
mysql> flush privileges;
```

Locate the bayes_mysql.sql file:

```
find / -name bayes_mysql.sql  
mysql -u sa_user -p sa_bayes < /path/to/bayes_mysql.sql
```

Backup your current bayes database:

```
sa-learn -p /etc/MailScanner/spam.assassin.prefs.conf --backup > sa_bayes_backup.txt
```

Warning: The next command can completely wipe out your bayes database!

```
sa-learn -p /path/to/spam.assassin.prefs.conf --clear #(entirely optional, incase you want to rollback)
```

Make some changes to your spam.assassin.prefs.conf:

```
bayes_store_module Mail::SpamAssassin::BayesStore::SQL  
bayes_sql_dsn DBI:mysql:sa_bayes:localhost  
bayes_sql_username sa_user  
bayes_sql_password sa_password  
bayes_sql_override_username root
```

and comment out the following lines:

```
#bayes_path /etc/MailScanner/bayes/bayes
#bayes_file_mode 0660
```

Populate the Bayes SQL database.

Now for recovering the bayes_dbm to bayes_sql.

```
sa-learn -p /etc/MailScanner/spam.assassin.prefs.conf --restore sa_bayes_backup.txt
```

This process may take some time depending on the size of your bayes database.

Also add this to your crontab:

```
crontab -e
```

```
30 01 * * * /path/to/sa-learn --force-expire --sync -p /etc/MailScanner/spam.assassin.prefs.conf
```

9.14 Bring it all Together

Now that we have everything in there, set the correct permissions:

```
chown -R postfix:www-data /var/spool/MailScanner
chown -R postfix:www-data /var/lib/MailScanner
chown -R postfix:www-data /var/run/MailScanner
chown -R postfix:www-data /var/lock/subsys/MailScanner
chown -R postfix:www-data /var/spool/postfix/hold
chmod -R ug+rwx /var/spool/postfix/hold
```

Finally make sure you restart MailScanner.

```
/etc/init.d/mailscanner restart
```

Test out the setup:

```
spamassassin -x -D -p /etc/MailScanner/spam.assassin.prefs.conf --lint
```

Check for lines like:

```
debug: bayes: Database connection established
debug: bayes: found bayes db version 3
debug: bayes: Using userid: 2
```

and some more like

```
debug: bayes: tok_get_all: Token Count: 20
debug: bayes token 'somewhat' ? 0.978
debug: bayes: score = 0.845189622547555
```

You should see lines come up with DCC, Pyzor and Razor that say loading plugin and hopefully no errors.

Finishing up this part we need to add cron jobs that will clean/update, you probably saw the message about this after the MailScanner install script finished.

First edit conf.php and set 'QUARANTINE_DAYS_TO_KEEP' in conf.php and change the following line in db_clean.

```
#!/usr/bin/php -qn
```

to

```
#!/usr/bin/php -q
```

Install quarantine clean up script:

```
cp /usr/src/mailwatch-1.0.4/tools/quarantine_maint.php /usr/bin/quarantine_maint.php
```

```
cp /usr/src/mailwatch-1.0.4/tools/db_clean.php /usr/bin/db_clean.php
chmod +x /usr/bin/quarantine_maint.php
chmod +x /usr/bin/db_clean.php
```

Run

```
crontab -e
```

and add the following:

```
15 10 * * 2 /usr/bin/quarantine_maint.php -clean &> /dev/null
58 23 * * * /usr/bin/db_clean.php &> /dev/null
```

Disable the mailscaanner installed cron script `/etc/cron.daily/clean.quarantine` (**Note:** Do this only if the `clean.quarantine` script exists).

```
$disabled = 1;
```

9.15 Reboot

reboot

Check your `mail.log` again:

```
tail -f /var/log/mail.log
```

At this point you should have a functional spamfilter and should see something like:

```
Jun 13 12:18:23 hoshi MailScanner[26388]: MailScanner E-Mail Virus Scanner version 4.20-3 starting...
Jun 13 12:18:24 hoshi MailScanner[26388]: Config: calling custom init function MailWatchLogging
Jun 13 12:18:24 hoshi MailScanner[26388]: Initialising database connection
Jun 13 12:18:24 hoshi MailScanner[26388]: Finished initialising database connection
```

Congratulations - you now have MailScanner logging to MySQL.

9.16 Test the MailWatch interface

Point your browser to `http://<hostname>/mailscanner/` - you should be prompted for a username and password - enter the details of the MailWatch web user that you created earlier, and you should see a list of the last 50 messages processed by MailScanner.

If you're not able to see the mails, then you may have to set the following permissions:

```
chgrp -R www-data /var/spool/MailScanner
```

You may have to create the following to prevent an error in a lint test:

```
mkdir /var/www/.spamassassin
```

9.17 Fix for Ubuntu 8.04 (kept removing directories upon reboot)

Edit `/etc/rc.local` and add the following before the exit line:

```
mkdir /var/run/MailScanner
mkdir /var/lock/subsys
mkdir /var/lock/subsys/MailScanner
chown -R postfix:www-data /var/run/MailScanner
chown -R postfix:www-data /var/lock/subsys/MailScanner
/etc/init.d/postfix restart
/etc/init.d/mailscanner restart
```

9.18 Update the SpamAssassin Rules table

MailWatch keeps a list of all the SpamAssassin rules and descriptions which are displayed on the 'Message Detail' page - to show the descriptions, you need to run the updater every time you add new rules or upgrade SpamAssassin. Click on the 'Tools/Links' menu and select 'Update SpamAssassin Rule Descriptions' and click 'Run Now'.

9.19 Update the GeoIP database

Change /var/www/mailscanner/geoip_update.php:

```
vi /var/www/mailscanner/geoip_update.php
```

```
dbquery("LOAD DATA INFILE
```

to

```
dbquery("LOAD DATA LOCAL INFILE
```

Make sure you have `allow_url_fopen = On` in your `php.ini` set.

Click on the 'Tools/Links' menu and select 'Update GeoIP database' and click 'Run Now'.

9.20 Setup the Mail Queue watcher (optional)

You can get MailWatch to watch and display your sendmail or exim queue directories - all you need to do is copy `mailq.php` (from the root of the mailwatch tarball - not from the mailscanner directory - they are different!) to `/usr/local/bin` and set-up a cron-job to run it.

Edit `mailq.php` first to change the `require` line to point to the location of `functions.php`, then:

```
cp mailq.php /usr/local/bin
crontab -e
```

```
0-59 * * * * /usr/local/bin/mailq.php
```

Note: `mailq.php` re-creates all entries on each run, so for busy sites you will probably want to change this to run every 5 minutes or greater.

9.21 Setup the Sendmail Relay Log watcher (optional)

You can get MailWatch to watch your sendmail logs and store all message relay information which is then displayed on the 'Message Detail' page which helps debugging and makes it easy for a Helpdesk to actually see where a message was delivered to by the MTA and what the response back was (e.g. the remote queue id etc.).

```
cp tools/sendmail_relay.php /usr/local/bin
nohup /usr/local/bin/sendmail_relay.php 2>&1 > /dev/null &
```

9.22 Fix to allow wildcards in Whitelist/Blacklist

Add the following to the bottom of the return 1 section in your SQLBlackWhiteList.pm:

```
return 1 if $BlackWhite->{$to}{'*@'.$fromdomain};
return 1 if $BlackWhite->{$to}{'*.*'.$fromdomain};
return 1 if $BlackWhite->{$todomain}{'*@'.$fromdomain};
return 1 if $BlackWhite->{$todomain}{'*.*'.$fromdomain};
return 1 if $BlackWhite->{'default'}{'*@'.$fromdomain};
return 1 if $BlackWhite->{'default'}{'*.*'.$fromdomain};
```

9.23 Fix for the Reporting Function in Message Operations

Change the following in /var/www/mailscanner/do_message_ops.php file:

```
vi /var/www/mailscanner/do_message_ops.php
```

```
$id = $Regs[1];
```

to

```
$id = str_replace("_", ".", $Regs[1]);
```

9.24 Fix to Allow Quarantine Release of Messages

Change the following in `/var/www/mailscanner/conf.php`:

```
define(QUARANTINE_FROM_ADDR, 'postmaster@domain.tld');
```

*You need to put the full email address or this will not work.

Also make sure the following string is set to true:

```
define(QUARANTINE_USE_FLAG, true);
```

If you'd like the message to be released in it's original form and not as an attachment, set the following line to true:

```
define(QUARANTINE_USE_SENDMAIL, true);
```

9.24.1 Dangerous Content:

Open `/etc/MailScanner/MailScanner.conf` and change the following:

```
Dangerous Content Scanning = yes
```

To

```
Dangerous Content Scanning = %rules-dir%/content.scanning.rules
```

Create `/etc/MailScanner/rules/content.scanning.rules` and add the following:

```
From: 127.0.0.1 no
```

```
FromOrTo: default yes
```

9.24.2 Filename and Filetype Release:

Modify /etc/MailScanner/MailScanner.conf and set the following:

Filename Rules = %etc-dir%/filename.rules

Filetype Rules = %etc-dir%/filetype.rules

Then create the following files as shown in /etc/MailScanner:

/etc/MailScanner/filename.rules:

From: 127.0.0.1 /etc/MailScanner/filename.rules.allowall.conf

FromOrTo: default /etc/MailScanner/filename.rules.conf

/etc/MailScanner/filetype.rules:

From: 127.0.0.1 /etc/MailScanner/filetype.rules.allowall.conf

FromOrTo: default /etc/MailScanner/filetype.rules.conf/etc/MailScanner/filename.rules.allowall.conf:

allow .* - -

/etc/MailScanner/filetype.rules.allowall.conf:

allow .* - -

9.24.3 Releasing Spam Messages

To allow MailWatch to release Spam messages without them being processed again, add 127.0.0.1 as a whitelist item in MailWatch/List interface. Make sure to restart MailScanner after configuring these options. Below is what my entry looks like.

127.0.0.1defaultDelete

9.25 Fix to Allow Multiple Release of Messages in Message Operations

Edit /var/www/mailscanner/do_message_ops.php and make the following changes:

```
case 'F':
```

```
$type='forget';
```

```
break;
```

```
case 'R':
```

```
$type='release';
```

```
break;
```

```
default:
```

```
continue;
```

```
break;
```

Then, find the following section and change it to look like this:

```
$itemnum = array($num);
```

```
if ($type == 'release'){
```

```
    if($quarantined = quarantine_list_items($id,RPC_ONLY)) {
```

```
        $to = $quarantined[0]['to'];
```

```
    }
```

```
    echo "<tr><td><a href='detail.php?id=$id'>$id</a></td><td>$type</td><td>" . quarantine_release($quarantined, $itemnum, $to, RPC_ONLY) .
"</td></tr>\n";

    } else {

    echo "<tr><td><a href='detail.php?id=$id'>$id</a></td><td>$type</td><td>" . quarantine_learn($items, $itemnum, $type, RPC_ONLY) .
"</td></tr>\n";

    }

}

}

echo "</TABLE>\n";

}

}

}

}

echo " </TD>\n";
```

Next we edit the /var/www/mailscanner/functions.php file and change:

```
$fieldname[$f] = "Ops<br>S H F";
```

To

```
$fieldname[$f] = "Ops<br>S H F R";
```

Next change:

```
array_unshift($row, "<INPUT NAME='OPT-REPLACEME' TYPE=RADIO VALUE='S'> <INPUT NAME='OPT-REPLACEME' TYPE=RADIO VALUE='H'> <INPUT NAME='OPT-REPLACEME' TYPE=RADIO VALUE='F'>");
```

To:

```
array_unshift($row, "<INPUT NAME='OPT-REPLACEME' TYPE=RADIO VALUE='S'> <INPUT NAME='OPT-REPLACEME' TYPE=RADIO VALUE='H'> <INPUT NAME='OPT-REPLACEME' TYPE=RADIO VALUE='F'> <INPUT NAME='OPT-REPLACEME' TYPE=RADIO VALUE='R'> ");
```

Next find the block with the javascript function to handle radio buttons. Add a third value like so:

```
echo "function SetRadios(p) {n";
```

```
echo " var val;n";
```

```
echo " if (p == 'S') {n";
```

```
echo " val = 0;n";
```

```
echo " } else if (p == 'H') {n";
```

```
echo " val = 1;n";
```

```
echo " } else if (p == 'F') {n";
```

```
echo " val = 2;n";
```

```
echo " } else if (p == 'R') {n";
```

```
echo " val = 3;n";
```

```
echo " } else if (p == 'C') {n";  
echo " ClearRadios();n";
```

Now, add the text for the radios:

```
echo " <a href='javascript:SetRadios('S')'>S</a>";  
echo " <a href='javascript:SetRadios('H')'>H</a>";  
echo " <a href='javascript:SetRadios('F')'>F</a>";  
echo " <a href='javascript:SetRadios('R')'>R</a>";
```

Finally, change:

```
echo "<P><b>S</b> = Spam <b>H</b> = Ham <b>F</b> = Forgetn";
```

To:

```
echo "<P><b>S</b> = Spam <b>H</b> = Ham <b>F</b> = Forget <b>R</b> = Releasen";
```

9.26 Fix to Allow Correct ClamAV Status

Change the following in `/var/www/mailscanner/clamav_status.php` file:

```
<?passthru(get_virus_conf('clamav')." -V | awk -f ./clamav.awk");?>
```

to

```
<?passthru("â€˜/usr/sbin/clamd â€˜v | awk â€˜f ./clamav.awkâ€™");?>
```

10 Install and Configure SPF

The `postfix-policyd-spf-perl` package depends on the `Mail::SPF` and the `NetAddr::IP` Perl modules.

We need to download `postfix-policyd-spf-perl` from <http://www.openspf.org/Software> to the `/usr/src/` directory and install it to the `/usr/lib/postfix/` directory like this:

```
cd /usr/src
wget http://www.openspf.org/blobs/postfix-policyd-spf-perl-2.005.tar.gz
tar xvfz postfix-policyd-spf-perl-2.005.tar.gz
cd postfix-policyd-spf-perl-2.005
cp postfix-policyd-spf-perl /usr/lib/postfix/policyd-spf-perl
```

Then we edit `/etc/postfix/master.cf` and add the following stanza at the end:

```
vi /etc/postfix/master.cf
```

```
policy unix - n n - - spawn
user=nobody argv=/usr/bin/perl /usr/lib/postfix/policyd-spf-perl
```

(The leading spaces before *user=nobody* are important so that Postfix knows that this line belongs to the previous one!)

Then open */etc/postfix/main.cf* and search for the *smtpd_recipient_restrictions* directive. You should have *reject_unauth_destination* in that directive, and right after *reject_unauth_destination* you add *check_policy_service unix:private/policy* like this:

```
vi /etc/postfix/main.cf
```

```
[...]
smtpd_recipient_restrictions = permit_sasl_authenticated,permit_mynetworks,reject_unauth_destination,check_policy_service unix:private/policy
[...]
```

or like this:

```
[...]
smtpd_recipient_restrictions =
[...
reject_unauth_destination
check_policy_service unix:private/policy
[...]
```

It is important that you specify *check_policy_service* AFTER *reject_unauth_destination* or else your system can become an open relay!

Then restart Postfix:

```
/etc/init.d/postfix restart
```

That's it already.

11 Install and Configure FuzzyOcr

```
apt-get install netpbm gifsicle libungif-bin gocr ocrad libstring-approx-perl libmldbm-sync-perl imagemagick tesseract-ocr
```

Download and install the latest FuzzyOCR devel version from <http://fuzzyocr.own-hero.net/wiki/Downloads>:

```
cd /usr/src/  
wget http://users.own-hero.net/~decoder/fuzzyocr/fuzzyocr-3.5.1-devel.tar.gz
```

Unpack FuzzyOCR and move all *FuzzyOcr** files and the FuzzyOcr directory (they are all in the *FuzzyOcr-3.5.1/* directory) to */etc/mail/spamassassin*;

```
tar xvfz fuzzyocr-3.5.1-devel.tar.gz  
cd FuzzyOcr-3.5.1/  
mv FuzzyOcr* /etc/mail/spamassassin/  
wget http://www.gbnetwork.co.uk/mailscanner/FuzzyOcr.words -O /etc/mail/spamassassin/FuzzyOcr.words
```

We will be storing the image hashes in a mysql database to improve on performance such that images that we have already scanned do not get scanned again as OCR is a resource intense activity.

11.1 Create MySQL Database

The sql script creates the database and tables and adds a user *fuzzyocr* with the password *fuzzyocr*:

```
mysql -p < /etc/mail/spamassassin/FuzzyOcr.mysql
```

Change the password:

```
mysqladmin -u fuzzyocr -p fuzzyocr newpassword
```

11.2 MailWatch Fix

Do the following to prevent an error in MailWatch:

```
vi /etc/mail/spamassassin/FuzzyOcr.pm
```

Change 'use POSIX;' to 'use POSIX qw(SIGTERM);'

11.3 FuzzyOcr Configuration

FuzzyOCR's configuration file is /etc/mail/spamassassin/FuzzyOcr.cf. In that file almost everything is commented out. We open that file now and make some modifications:

```
vi /etc/mail/spamassassin/FuzzyOcr.cf
```

Put the following line into it to define the location of FuzzyOCR's spam words file:

```
focr_global_wordlist /etc/mail/spamassassin/FuzzyOcr.words
```

`/etc/mail/spamassassin/FuzzyOcr.words` is a predefined word list that comes with FuzzyOCR. You can adjust it to your needs.

Next change:

```
# Include additional scanner/preprocessor commands here:  
#
```

```
focr_bin_helper pnmnorm, pnminvert, pamthreshold, ppmtopgm, pamtopnm  
focr_bin_helper tesseract
```

to

```
# Include additional scanner/preprocessor commands here:  
#  
focr_bin_helper pnmnorm, pnminvert, convert, ppmtopgm, tesseract
```

Finally add/enable the following lines:

```
# Search path for locating helper applications  
focr_path_bin /usr/local/netpbm/bin:/usr/local/bin:/usr/bin  
focr_preprocessor_file /etc/mail/spamassassin/FuzzyOcr.preps  
focr_scanset_file /etc/mail/spamassassin/FuzzyOcr.scansets  
focr_digest_db /etc/mail/spamassassin/FuzzyOcr.hashdb  
focr_db_hash /etc/mail/spamassassin/FuzzyOcr.db  
focr_db_safe /etc/mail/spamassassin/FuzzyOcr.safe.db  
focr_minimal_scanset 1  
focr_autosort_scanset 1  
focr_enable_image_hashing 3  
focr_logfile /var/log/FuzzyOcr.log  
#Mysql Connection#  
focr_mysql_db FuzzyOcr  
focr_mysql_hash Hash  
focr_mysql_safe Safe  
focr_mysql_user fuzzyocr  
focr_mysql_pass password  
focr_mysql_host localhost  
focr_mysql_port 3306  
focr_mysql_socket /var/run/mysqld/mysqld.sock
```

This is what the FuzzyOCR developers say about image hashing:

"The Image hashing database feature allows the plugin to store a vector of image features to a database, so it knows this image when it arrives a second time (and therefore does not need to scan it again). The special thing about this function is that it also recognizes the image again if it was changed slightly (which is done by spammers). "

11.4 Test FuzzyOCR

```
cd /usr/src/FuzzyOcr-3.5.1/samples
spamassassin --debug FuzzyOcr < ocr-animated.eml > /dev/null
```

You see the following:

```
[14808] info: FuzzyOcr: Found Score <9.000> for Exact Image Hash
[14808] info: FuzzyOcr: Matched [1] time(s). Prev match: 16 sec. ago
[14808] info: FuzzyOcr: Message is SPAM. Words found:
[14808] info: FuzzyOcr: "price" in 1 lines
[14808] info: FuzzyOcr: "company" in 1 lines
[14808] info: FuzzyOcr: "alert" in 1 lines
[14808] info: FuzzyOcr: "news" in 1 lines
[14808] info: FuzzyOcr: (6 word occurrences found)
[14808] dbg: FuzzyOcr: Remove DIR: /tmp/.spamassassin14808JZSvHBtmp
[14808] dbg: FuzzyOcr: Processed in 0.104555 sec.
```

12 Apply Relay Recipients

The following directions are meant for people using Microsoft Exchange 2000 or Microsoft Exchange 2003.

This page describes how to configure your mail gateway to periodically get a list of valid recipient email addresses from your Exchange system. By doing this, you can configure your server to automatically reject any email addressed to invalid addresses. This will reduce the load on your exchange server, since it no longer has to process non-delivery reports, and it will reduce the load on your postfix server since it won't have to perform spam and virus scanning on the message.

12.1 Install Dependencies

Install the perl module Net::LDAP:

```
perl -MCPAN -e shell
install Net::LDAP
```

12.2 Create the Get Email Address Script

Create and edit the script:

```
vi /usr/bin/getadsmtpl
```

Copy and paste the code below into this new file.

```
#!/usr/bin/perl -T -w
# This script will pull all users' SMTP addresses from your Active Directory
# (including primary and secondary email addresses) and list them in the
# format "user@example.com OK" which Postfix uses with relay_recipient_maps.
# Be sure to double-check the path to perl above.
# This requires Net::LDAP to be installed. To install Net::LDAP, at a shell
# type "perl -MCPAN -e shell" and then "install Net::LDAP"
use Net::LDAP;
use Net::LDAP::Control::Paged;
use Net::LDAP::Constant ( "LDAP_CONTROL_PAGED" );
# Enter the path/file for the output
$VALID = "/etc/postfix/relay_recipients";
open VALID, ">$VALID" or die "CANNOT OPEN $VALID $!";
# Enter the FQDN of your Active Directory domain controllers below
$dc1="domaincontroller1.example.com";
$dc2="domaincontroller2.example.com";
```

```
# Enter the LDAP container for your userbase.
# The syntax is CN=Users,dc=example,dc=com
# This can be found by installing the Windows 2000 Support Tools
# then running ADSI Edit.
# In ADSI Edit, expand the "Domain NC [domaincontroller1.example.com]" &
# you will see, for example, DC=example,DC=com (this is your base).
# The Users Container will be specified in the right pane as
# CN=Users depending on your schema (this is your container).
# You can double-check this by clicking "Properties" of your user
# folder in ADSI Edit and examining the "Path" value, such as:
# LDAP://domaincontroller1.example.com/CN=Users,DC=example,DC=com
# which would be $hqbase="cn=Users,dc=example,dc=com"
# Note: You can also use just $hqbase="dc=example,dc=com"
$hqbase="cn=Users,dc=example,dc=com";
# Enter the username & password for a valid user in your Active Directory
# with username in the form cn=username,cn=Users,dc=example,dc=com
# Make sure the user's password does not expire. Note that this user
# does not require any special privileges.
# You can double-check this by clicking "Properties" of your user in
# ADSI Edit and examining the "Path" value, such as:
# LDAP://domaincontroller1.example.com/CN=user,CN=Users,DC=example,DC=com
# which would be $user="cn=user,cn=Users,dc=example,dc=com"
# Note: You can also use the UPN login: "user@example.com"
$user="cn=user,cn=Users,dc=example,dc=com";
$password="password";
# Connecting to Active Directory domain controllers
$noldapservers=0;
$ldap = Net::LDAP->new($dc1) or
    $noldapservers=1;
if ($noldapservers == 1) {
    $ldap = Net::LDAP->new($dc2) or
        die "Error connecting to specified domain controllers $@ \n";
}
```

```

$mesg = $ldap->bind ( dn => $user,
                    password => $passwd);
if ( $mesg->code() {
    die ("error:", $mesg->error_text(),"\n");
}
# How many LDAP query results to grab for each paged round
# Set to under 1000 for Active Directory
$page = Net::LDAP::Control::Paged->new( size => 990 );
@args = ( base => $hqbase,
# Play around with this to grab objects such as Contacts, Public Folders, etc.
# A minimal filter for just users with email would be:
# filter => "((&(sAMAccountName=*)(mail=*))"
    filter => "( (& (mailnickname=*) (| (&(objectCategory=person)
        (objectClass=user)!(homeMDB=*))!(msExchHomeServerName=*))
        (&(objectCategory=person)(objectClass=user)(homeMDB=*)
        (msExchHomeServerName=*))(&(objectCategory=person)(objectClass=contact))
        (objectCategory=group)(objectCategory=publicFolder) ))",
    control => [ $page ],
    attrs => "proxyAddresses",
);
my $cookie;
while(1) {
    # Perform search
    my $mesg = $ldap->search( @args );
# Filtering results for proxyAddresses attributes
foreach my $entry ( $mesg->entries ) {
    my $name = $entry->get_value( "cn" );
    # LDAP Attributes are multi-valued, so we have to print each one.
    foreach my $mail ( $entry->get_value( "proxyAddresses" ) ) {
        # Test if the Line starts with one of the following lines:
        # proxyAddresses: [smtp]SMTP];
        # and also discard this starting string, so that $mail is only the
        # address without any other characters...

```

```
if ( $mail =~ s/^(smtp|SMTP)://gs ) {
    print VALID $mail." OK\n";
}
}
}
# Only continue on LDAP_SUCCESS
$mesg->code and last;
# Get cookie from paged control
my($resp) = $mesg->control( LDAP_CONTROL_PAGED ) or last;
$cookie = $resp->cookie or last;
# Set cookie in paged control
$page->cookie($cookie);
}
if ($cookie) {
    # We had an abnormal exit, so let the server know we do not want any more
    $page->cookie($cookie);
    $page->size(0);
    $ldap->search( @args );
    # Also would be a good idea to die unhappily and inform OP at this point
    die("LDAP query unsuccessful");
}
# Add additional restrictions, users, etc. to the output file below.
#print VALID "user\@domain1.com OK\n";
#print VALID "user\@domain2.com 550 User unknown.\n";
#print VALID "domain3.com 550 User does not exist.\n";
close VALID;
```

Next set the permissions on the file to allow it to be executed:

```
chmod 500 /usr/bin/getadsmtp.pl
```

Edit the file to customize it for your specific domain. Since the file is read only, you will need to use :w! to save the file in vi.

1. Set `$dc1` and `$dc2` to the fully qualified domain names or IP addresses of 2 of your domain controllers.
2. Set `$hqbase` equal to the LDAP path to the container or organizational unit which holds the email accounts for which you wish to get the email addresses.
3. Set `$user` and `$passwd` to indicate which user account should be used to access this information. This account only needs to be a member of the domain, so it would be a good idea to setup an account specifically for this.

12.3 Run the Script

Try running the script. If it works correctly, it will create `/etc/postfix/relay_recipients` Note that if your postfix server is separated from your active directory controllers by a firewall, you will need to open TCP port 389 from the postfix server to the ADCs. At this point, you can update your `/etc/postfix/main.cf` to `relay_recipient_maps`. You will also have to postmap the file to create the database.

```
getadsmtp.pl
```

At this point, you may want to edit `/etc/postfix/relay_recipients` and edit out any unwanted email addresses as this script imports everything.

12.4 Create the Table

```
postmap /etc/postfix/relay_recipients
```

Finally, you may want to set up a cron job to periodically update and build the `/etc/postfix/relay_recipients.db` file. You can set up a script called `/usr/bin/update-relay-recipients.sh`: (Optional)

```
vi /usr/bin/update-relay-recipients.sh
```

```
#!/bin/sh
```

```
/usr/bin/getadsmtp.pl
cd /etc/postfix
postmap relay_recipients
```

Don't forget to make sure the following is in your `/etc/postfix/main.cf` file:

```
relay_recipient_maps = hash:/etc/postfix/relay_recipients
```

Make the script executable:

```
chmod +x /usr/bin/update-relay-recipients.sh
```

Run crontab to add this script to the scheduled jobs:

```
crontab -e
```

Now add the following lines to the bottom of the file. Note that this cron job will run every day at 2:30 AM to update the database file. You may want to run yours more frequently or not depending on how often you add new email users to your system.

```
# synchronize relay_recipients with Active Directory addresses
30 2 * * * /usr/bin/update-relay-recipients.sh
```

13 Filtering PDF, XLS and Phishing Spam with ClamAV (Sanesecurity Signatures)

There is currently a lot of spam where the spam "information" is attached as .pdf or .xls files, sometimes also hidden inside a .zip file. While these spam mails are not easy to catch with e.g. SpamAssassin or a Bayes filter, the ClamAV virus scanner can catch them easily when it is fed with the correct signatures as ClamAV is built to scan mail attachments.

13.1 Create a Folder and Download the Script

Create a folder for sanesecurity and download and give the script the proper permission.

```
apt-get install curl
```

```
mkdir /usr/src/sanesecurity
cd /usr/src/sanesecurity
wget http://www.sanesecurity.co.uk/clamav/ss-msrbl.txt
mv ss-msrbl.txt /usr/bin/ss_update.sh
chmod +x /usr/bin/ss_update.sh
```

Edit `ss_update.sh` and change the following variables to match your installation:

```
clam_sigs="/var/lib/clamav"
```

The variable `clamav_sigs` contains the path to the directory where your ClamAV signatures are stored.

```
clam_user="clamav"
```

Now we run the update script to check if the download works:

```
./ss_update.sh
```

The result should look similar to this:

```
=====
SaneSecurity SCAM Database Update
```

```
=====  
% Total % Received % Xferd Average Speed Time Time Time Current  
Dload Upload Total Spent Left Speed  
100 116k 100 116k 0 0 65448 0 0:00:01 0:00:01 --:--:-- 139k
```

```
=====  
SaneSecurity PHISH Database Update  
=====  
% Total % Received % Xferd Average Speed Time Time Time Current  
Dload Upload Total Spent Left Speed  
100 179k 100 179k 0 0 216k 0 --:--:-- --:--:-- --:--:-- 216k
```

```
=====  
MSRBL SPAM Database Update  
=====  
Number of files: 1  
Number of files transferred: 1  
Total file size: 228436 bytes  
Total transferred file size: 228436 bytes  
Literal data: 228436 bytes  
Matched data: 0 bytes  
File list size: 33  
File list generation time: 0.001 seconds  
File list transfer time: 0.000 seconds  
Total bytes sent: 101  
Total bytes received: 228579  
sent 101 bytes received 228579 bytes 26903.53 bytes/sec  
total size is 228436 speedup is 1.00
```

```
=====  
MSRBL IMAGE Database Update  
=====  
Number of files: 1
```

```
Number of files transferred: 1
Total file size: 550503 bytes
Total transferred file size: 550503 bytes
Literal data: 550503 bytes
Matched data: 0 bytes
File list size: 35
File list generation time: 0.001 seconds
File list transfer time: 0.000 seconds
Total bytes sent: 103
Total bytes received: 550688
sent 103 bytes received 550688 bytes 157368.86 bytes/sec
total size is 550503 speedup is 1.00
```

Now we add the script to the root crontab to be run once a day:

```
crontab -e
```

Add the following line at the end of the root crontab:

```
00 04 * * * /usr/bin/ss_update.sh &> /dev/null
```

14 GreyListing with Postfix-gld14.1 Installing Postfix-gld

```
apt-get install postfix-gld
```

Create MySQL Database:

```
mysql -u root -p
```

```
mysql> create database gld_db
mysql> GRANT ALL ON gld_db.* TO gld_user@localhost IDENTIFIED BY '~gld_password';
mysql> flush privileges;
```

Import `tables.mysql`:

```
mysql -u gld_user -p gld_db < /usr/share/gld/tables.mysql
```

Import `table-whitelist.sql`:

```
mysql -u gld_user -p gld_db < /usr/share/gld/table-whitelist.sql
```

You will have to enable it by configuring that in the `/etc/default/gld`:

```
vi /etc/default/gld
```

```
#/etc/default/gld
ENABLED=1
```

14.2 Configuration

Edit `/etc/gld.conf` according to your needs. I'm using the following settings:

```
vi /etc/gld.conf
```

```
# Config file for gld
```

```
# TCP Port gld should listen to (default is 2525)
#
PORT=2525
# Shall we bind only to loopback ? (0=No,1=Yes) (default is 1)
LOOPBACKONLY=1
# The list of networks allowed to connect to us (default is everybody)
CLIENTS=127.0.0.1/32
# The user used to run gld (default value is no user change)
USER=postfix-gld
# The group used to run gld (default value is no group change)
GROUP=postfix-gld
# Maximum simultaneous connexions (default is 100)
MAXCON=100
# How many seconds we should wait before accepting a mail that is in greylist (default is 60)
MINTIME=60
# Shall we use lightgrey option ? (0=No,1=Yes) (default is 0)
# The lightgrey option, mask the last octet of IP addresses
# and thus we greylist only C classes (/24) instead of individual IPs.
LIGHTGREY=0
# Shall we use the mxgrey algorithm ? (0=No,>0=Yes) (default is 0)
# the mxgrey algorithm is a variation of the greylist algorithm.
# When this is enabled, we allow all incoming mails from an IP address
# whatever source/destination email as long as this IP has been greylisted
# at least X time and succeeded the mail resend .
#
# Example:
# The IP 1.2.3.4 sends an email from src@domain.com to user@yourdomain.com
# We greylist this mail as this IP is not yet in database and send a 450 SMTP code
# After some time, the IP re-send the mail from src@domain.com to user@yourdomain.com
# We update the db.
# Some time after the ip 1.2.3.4 sends an email from john@domain.com to fred@yourdomain.com
# We will accept this mail without any greylisting, as this ip already succeeded a greylist test
# and thus seems to be a valid smtp server and not a spammer .
```

```
#
# The advantage of this method, is that it reduce the re-send time due to greylisting to
# x mail per server instead of one mail per destination .
#
# The value you provide in MXGREY is the minimum number of succesful greylists
# before accepting all mails from this MX. higher the number is, harder is to get in.
#
# This algortihm replace the old LIGHTGREYDOMAIN which was available prior version 1.6
#
MXGREY=1
# Shall we use the whitelist table ? (0=No,1=Yes) (default is 1)
# If set to yes, then the table 'whitelist' is looked up
# each time postfix request the server
# if the email/domain/ip is in the whitelist, then the response
# will be 'dunno' .
# In the whitelist table, you can set the following values:
# an email: ie john@foo.tld
# a domain: ie @bar.tld
# an IP : ie 1.2.3.4
# a subnet: ie 1.2.3
#
WHITELIST=1
# Shall we use a DNS based whitelist ? (default is no)
# To activate it, the line must be uncommented
# and the value set to the domain of the DNS whitelist.
# for example, if DNSWL is set to toto.com and we get a mail from ip a.b.c.d
# then gld will DNS lookup d.c.b.a.toto.com
# and if found allow the ip without greylisting it.
#DNSWL=toto.com
# Shall we send a 'dunno' in case of error (mysql down,...) (0=No,1=Yes) (default is 1)
# Normaly, if an error occur, the server is supposed to close the connection
# and thus postfix will return a 450 Server configuration error
# if this parameter is set to 1, then the server will return 'dunno'
```

```
# and thus let postfix decide the fate of the mail.
ERRACCEPT=1
# Shall we log to the syslog (0=No,1=Yes) (default is 1)
SYSLOG=1
# If we use syslog, which facility shall we use (default is mail)
# it can only be one of the following facilities:
# daemon mail local0 local1 local2 local3 local4 local5 local6 local7
FACILITY=mail
# The Message that we display in case of reject (default is "Greylisted")
# If you want another SMTP return code than the default 450, just put it at
# the beginning of the message, ie: 451 You have been greylisted by gld ...
# If you don't provide any SMTP code, the default 450 will be used by postfix
# WARNING: if you set a custom smtp code make sure it's a 4XX code.
# if you don't provide a 4XX code, gld will ignore it and send the default 450.
# Be also warned that if you set a custom code, gld will not use defer_if_permit anymore
# but direct supplied code to postfix .
MESSAGE=Service temporarily unavailable, please try later
# Training mode activated ? (0=No,1=Yes) (default is 0)
# If activated, gld will do all the work but will always reply dunno to postfix
# and thus, will never greylist any mail.
# This feature is useful for testing gld performances without greylisting any mail
TRAINING=0
# SQL INFOS (defaults are localhost,myuser,mypasswd,mydb)
#
SQLHOST=localhost
SQLUSER=gld_user
SQLPASSWD=gld_password
SQLDB=gld_db
```

Edit `/etc/postfix/main.cf` and add the following to `smtpd_recipient_restrictions`:

```
vi /etc/postfix/main.cf
```

```
check_policy_service inet:127.0.0.1:2525
```

Do a

```
tail -f /var/log/mail.log
```

and check your log for the following:

```
Apr 28 09:07:03 server1 gld: Greylist activated for recipient=<xxx@xxx.com> sender=<xxx@xxx.com> ip=<xxx.xxx.xxx.xxx>
```

You can set up a cron job to keep your database clean. Below is what I'm using in crontab -e running at midnight.

```
@daily /usr/bin/mysql -ugld_user -pgld_password -e 'USE gld_db; DELETE FROM greylist WHERE n > 0;' &> /dev/null
```

15 Logwatch Statistical Reporting (Optional)

Logwatch is a customizable log analysis system. Logwatch parses through your system's logs for a given period of time and creates a report analyzing areas that you specify, in as much detail as you require.

We will be using Logwatch to give us daily reports for mailscanner. This is a way for us to see how effective mailscanner really is.

Install Logwatch:

```
apt-get install logwatch
```

Edit the `/usr/share/logwatch/default.conf/logwatch.conf` and set the options:

```
vi /usr/share/logwatch/default.conf/logwatch
```

```
Mail To = youremailaddress
Service = mailscanner
```

Test Logwatch:

```
logwatch
```

It should generate a log file and email it to the email you specified.

16 Automatically Add A Disclaimer To Outgoing Emails With alterMIME (Optional)

This tutorial shows how to install and use alterMIME. alterMIME is a tool that can automatically add a disclaimer to emails. In this article I will explain how to install it as a Postfix filter on Ubuntu.

16.1 Installing alterMIME

alterMIME can be installed as follows:

```
apt-get install altermime
```

Next we create the user *filter* with the home directory */var/spool/filter* - alterMIME will be run as that user:

```
useradd -r -c "Postfix Filters" -d /var/spool/filter filter
mkdir /var/spool/filter
chown filter:filter /var/spool/filter
chmod 750 /var/spool/filter
```

Afterwards we create the script */etc/postfix/disclaimer* which executes alterMIME. Ubuntu's alterMIME package comes with a sample script that we can simply copy to */etc/postfix/disclaimer*:

```
cp /usr/share/doc/altermime/examples/postfix_filter.sh /etc/postfix/disclaimer
chgrp filter /etc/postfix/disclaimer
chmod 750 /etc/postfix/disclaimer
```

Now the problem with this script is that it doesn't distinguish between incoming and outgoing emails - it simply adds a disclaimer to all mails. Typically you want disclaimers only for outgoing emails, and even then not for all sender addresses. Therefore I've modified the `/etc/postfix/disclaimer` script a little bit - we'll come to that in a minute.

Right now, we create the file `/etc/postfix/disclaimer_addresses` which holds all sender email addresses (one per line) for which alterMIME should add a disclaimer:

```
vi /etc/postfix/disclaimer_addresses
```

```
user1@example.com
user2@example.org
user3@example.net
```

Now we open `/etc/postfix/disclaimer` and modify it as follows (I have marked the parts that I've changed):

```
vi /etc/postfix/disclaimer
```

```
#!/bin/sh
# Localize these.
INSPECT_DIR=/var/spool/filter
SENDMAIL=/usr/sbin/sendmail
##### Changed From Original Script #####
DISCLAIMER_ADDRESSES=/etc/postfix/disclaimer_addresses
##### Changed From Original Script END #####
# Exit codes from <sysexits.h>
```

```

EX_TEMPFAIL=75
EX_UNAVAILABLE=69
# Clean up when done or when aborting.
trap "rm -f in.$$" 0 1 2 3 15
# Start processing.
cd $INSPECT_DIR || { echo $INSPECT_DIR does not exist; exit
$EX_TEMPFAIL; }
cat >in.$$ || { echo Cannot save mail to file; exit $EX_TEMPFAIL; }
##### Changed From Original Script #####
# obtain From address
from_address=`grep -m 1 "From:" in.$$ | cut -d "<" -f 2 | cut -d ">" -f 1`
if [ `grep -wi ^${from_address}$ ${DISCLAIMER_ADDRESSES}` ]; then
  /usr/bin/altermime --input=in.$$ \
    --disclaimer=/etc/postfix/disclaimer.txt \
    --disclaimer-html=/etc/postfix/disclaimer.txt \
    --xheader="X-Copyrighted-Material: Please visit http://www.company.com/privacy.htm" || \
    { echo Message content rejected; exit $EX_UNAVAILABLE; }
fi
##### Changed From Original Script END #####
$SENDMAIL "$@" <in.$$
exit $?

```

Next we need the text file `/etc/postfix/disclaimer.txt` which holds our disclaimer text. Ubuntu's alterMIME package comes with a sample text that we can use for now (of course, you can modify it if you like):

```
cp /usr/share/doc/altermime/examples/disclaimer.txt /etc/postfix/disclaimer.txt
```

Finally we have to tell Postfix that it should use the `/etc/postfix/disclaimer` script to add disclaimers to outgoing emails. Open `/etc/postfix/master.cf` and add `-o content_filter=dfilt:` to the `smtp` line:

```
vi /etc/postfix/master.cf
```

```
#
# Postfix master process configuration file. For details on the format
# of the file, see the master(5) manual page (command: "man 5 master").
#
# =====
# service type private unpriv chroot wakeup maxproc command + args
#          (yes) (yes) (yes) (never) (100)
# =====
smtp inet n - - - smtpd
  -o content_filter=dfilt:
[...]
```

At the end of the same file, add the following two lines:

```
[...]
dfilt unix - n n - - pipe
  flags=Rq user=filter argv=/etc/postfix/disclaimer -f ${sender} -- ${recipient}
```

Restart Postfix afterwards:

```
/etc/init.d/postfix restart
```

That's it! Now a disclaimer should be added to outgoing emails sent from the addresses listed in `/etc/postfix/disclaimer_addresses`.

17 Firewalling the SpamSnake with FireholIntroduction

Firehol is a stateful iptables packet filtering firewall configurator. It is abstracted, extensible, easy and powerful. It can handle any kind of firewall, but most importantly, it gives you the means to configure it, the same way you think of it.

17.1 Install Firehol

Install firehol by doing:

```
apt-get install firehol
```

17.2 Firehol Settings

Edit `/etc/default/firehol` and change the following:

```
vi /etc/default/firehol
```

```
START_FIREHOL=NO
```

to

```
START_FIREHOL=YES
```

Edit `/etc/firehol/firehol.conf` and add the following:

```
vi /etc/firehol/firehol.conf
```

```
version 5
# Accept all client traffic on any interface
interface any internet
    protection strong
    server "icmp ping ICMP ssh http https telnet samba webmin dns dcc echo smtp" accept
    client all accept
```

Be sure to comment out the default configuration before applying these settings. This filters all incoming connections that are not related to the above services.

If you want to be less polite, you can drop them by adding the following after 'protection strong': *policy drop*

17.3 Updating RESERVED_IPS list

```
cd /usr/src
wget http://firehol.sf.net/firehol.tar.gz
tar xzvf firehol.tar.gz
cd firehol
mv get-iana.sh /usr/bin/get-iana.sh
chmod +x /usr/bin/get-iana.sh
```

Run the script to update RESERVED_IPS:

```
get-iana.sh
```

Make sure to select 'yes' when asked if you would like to save RESERVED_IPS to */etc/firehol/RESERVED_IPS*.

Start Firehol:

```
/etc/init.d/firehol start
```

Now your server is set up to only accept connections for the services you allowed.

Congratulations!

You should now have a complete working SpamSnake.

Here are some Mailwatch screenshots:



Jump to message:

Color Codes	
Bad Content/Infected	Red
Spam	Red
High Spam	Red
MCP	Light Blue
High MCP	Dark Blue
Whitelisted	Green
Blacklisted	Black
Clean	White

Status	
MailScanner:	YES 5 children
Postfix:	YES 9 proc(s)
Lead Average:	0.12 0.12 0.09
Mail Queues	
Inbound:	0
Outbound:	0

Today's Totals	
Processed:	414 5Mb
Clean:	186 44.9%
Viruses:	0 0.0%
Top Virus:	None
Blocked files:	0 0.0%
Others:	0 0.0%
Spam:	8 1.9%
High Scoring Spam:	220 53.1%
MCP:	0 0.0%
High Scoring MCP:	0 0.0%

- Recent Messages
- Lists
- Quarantine
- Reports
- Tools/Links
- Logout

Last 50 Messages (Refreshing every 30 seconds)							
#	Date/Time	From	To	Subject	Size	SA Score	Status
[]	04/30/08 10:43:36	dvsportomotoringm@sportomotoring.com	mailto:...	Get your free 2400\$ welcome bo...	1.8kb	18.00	Spam
[]	04/30/08 10:43:17	kconley@ ...	mailto:...	PROMEDIX Sean Ahern 5-6-08	119.4kb	0.00	W/L
[]	04/30/08 10:42:51	600firestone2@cox.net	mailto:...	Limited quantities, '08 Collec...	2.9kb	27.46	Spam
[]	04/30/08 10:42:40	qb-billpay-errors@ ...	mailto:...	Paid Invoice from Audio Visual...	81.3kb	-6.60	Clean
[]	04/30/08 10:42:07	ebecker@ ...	mailto:...	Read: RE:	2.2kb	-0.43	Clean
[]	04/30/08 10:41:46	ret@www.broadlinesupply.com	mailto:...	Satellite TV On Your Laptop or...	3.9kb	20.55	Spam
[]	04/30/08 10:40:30	terretir_1968@bluecube.com	mailto:...	Say no to all diseases!	2.1kb	16.69	Spam
[]	04/30/08 10:39:20	bounce-21531524-13948969@tigeronline.com	mailto:...	\$299 Laptop...22" LCD \$19...	34.5kb	4.69	Spam

Displaying page 1 of 9 - Records 1 to 50 of 406
 << < 1 2 3 4 5 6 7 8 9 > >>

Folder: 04/30/2008							
#	Date/Time (A/D)	From (A/D)	To (A/D)	Subject (A/D)	Size (A/D)	SA Score (A/D)	Status
[]	04/30/08 10:38:32	znhkay@bellsouth.com	znhkay@bellsouth.com	Order for 5/9/08 at 12:30pm Te...	116.7Kb	0.00	W/L
[]	04/30/08 10:38:09	sp@bellsouth.com	sp@bellsouth.com	Someone has a crush on you!	2.9Kb	8.46	Spam
[]	04/30/08 10:37:52	re@bellsouth.com	re@bellsouth.com	Re: Ubuntu - sendmail problem	4.3Kb	-3.60	Clean
[]	04/30/08 10:37:36	sp@bellsouth.com	sp@bellsouth.com	I am amazed by how fast I grew...	2.1Kb	24.63	Spam
[]	04/30/08 10:37:28	friendster@friendster.com	friendster@friendster.com	Friendster Friend Update	34Kb	0.00	Spam B/L
[]	04/30/08 10:35:35	sp@bellsouth.com	sp@bellsouth.com	\$59.95 Price for Viagra 50mg x...	1.6Kb	37.10	Spam
[]	04/30/08 10:35:27	sp@bellsouth.com	sp@bellsouth.com	Hi Maintain your peak	1.4Kb	30.47	Spam
[]	04/30/08 10:34:52	sp@bellsouth.com	sp@bellsouth.com	Your Nintendo Wii is ready to...	2.6Kb	17.75	Spam

Add to Whitelist/Blacklist

From:

To:

List: Whitelist Blacklist

Action:

Whitelist for spamsnake			Blacklist for spamsnake		
From	To	Action	From	To	Action
*@hotmail.com	default	Delete	*@friendster.com	default	Delete
*@aol.com	default	Delete	*@myspace.com	default	Delete
www-data@server1.ubuntu.com	default	Delete	*@mail.friendster.com	default	Delete
onlineorders@amazon.com	default	Delete			
*@yahoo.com	default	Delete			
*@gmail.com	default	Delete			
*@hotmail.com	default	Delete			
*@comcast.net	default	Delete			
apple@comcast.net	default	Delete			
*@aol.com	default	Delete			
*@comcast.net	default	Delete			