SuSE 10

Mail Scanning

Gateway Build

Guide For Beginners

Written By: Stephen Carter
stephen@retnet.co.uk

Last Modified 17. Feb. 2006

Table of Contents

Due credit

Thank you to everyone who has e-mailed corrections and ideas to help make this revision happen.

Special thanks to:

Andy Rosulek, Computer Network Coordinator at Northeast Iowa Community College, USA

Arthur Bezuidenhout, IT Professional at Colorado Department of Human Service, USA

Koltogyan Sergey, System Administrator at NPP AMI, Donetsk, Ukraine

Your efforts and patience are greatly appreciated.

Overview

This is a guide to introduce people to one method of building a reliable, free and flexible mail scanning gateway. The reason I created it is although there are many README and INSTALL documents and other bits around on the Internet today, I couldn't find a single consolidated guide for configuring everything I wanted in an e-mail scanning gateway for SuSE.

As a result I've created my own, based on SuSE 10. I've put it all together in an easy to follow step-by-step format to help even a close to near computer novice setup a simple e-mail scanning gateway solution.

If you come across something that doesn't work as expected, is just plain wrong, could be better in any way or just want to comment in general, please feel free to e-mail me your feedback.

The e-mail gateway server I describe here is designed to sit between your Internet connection and your existing e-mail server, be that GroupWise, Exchange, Notes, postfix or whatever else is out there... as long as it runs an SMTP daemon this gateway will work.

Software used in this guide includes:

SuSE 10 *Base OS* *http://www.novell.com/linux*	Amavisd-new v2.2.0 (*downgraded from the SuSE installed version)* *The 'glue' between Postfix and spam/virus scanners* *http://www.ijs.si/software/amavisd/*	DCC v1.3.20 *(although it's probably a newer version by now, that shouldn't matter)* *Anti-spam plug-in to SpamAssassin* *http://www.rhyolite.com/anti-spam/dcc/*
Postfix v2.2.5 *Mail Transfer Agent* *http://www.postfix.org*	SpamAssassin v3.0.4 *Anti-spam scanner* *http://spamassassin.apache.org*	Razor2 v2.72 *Anti-spam plug-in to SpamAssassin* *http://razor.sourceforge.net/*
MySQL v4.1.13 *Database Server* *http://www.mysql.com*	ClamAV v0.86.2 *Anti-virus scanner* *http://www.ClamAV.net*	Maia Mailguard v1.0.0 *Web front-end management for spam & virus'* *http://www.maiamailguard.com*
Apache v2.0.54 *Web Server* *http://httpd.apache.org*		

How much e-mail your solution will cope with will depend on many assumptions such as how long an acceptable processing time per e-mail is for you, the average size of e-mails that pass through your system and how often the system is managed by someone to clear out spam / non-spam (clean) e-mail confirmations.

Typically this gateway should be able to cope with around 30,000+ untrusted e-mails per day (trusted e-mails aren't scanned for spam but again this can be easily changed).

System Requirements

To follow this guide you should have available:

SuSE 10 Evaluation DVD

I will be using the SuSE 10.0 Evaluation DVD, which may differ slightly to the 5 CD's or other DVD versions.

You can download and burn the SuSE Linux 10.0 Evaluation version DVD ISO from
http://www.novell.com/products/suselinux/downloads/suse_linux/index.html

An existing e-mail server

This is an e-mail gateway scanning solution only, so you will need an e-mail server for this gateway to pass e-mail onto it's final destination. The SMTP interface on your e-mail server will also need to accept connections from the gateway server, so if you have configured any relay restrictions on your current e-mail server, you'll need to add the IP address of the server as an exception.

A Pentium class PC

If you're just setting this up for training or testing, just about anything will do, including a VMWare session but make sure it has a least 256Mb RAM available. For a production system I would suggest at least 1Gb RAM, to ensure that nothing gets swapped out to disk. If you are forced to run with less than 512 Mb, you can reduce the number of scanning threads available which is explained later in the amavisd-new section, to help reduce resource requirements on the server.

You will need plenty of disk space depending on the number of users it will be supporting. By default the database will hold up to 30 days worth of e-mail (configurable in Maia Mailguard), queued e-mail plus logs. For a typical 1000 user company you could be looking at potentially 50Gb +. Faster hard disks will also make a big difference with scanning times as an e-mail travels through the gateway.

The processor is hit quite hard so the faster the better, but realistically for less than 30,000 processed e-mails per day, a 3 Ghz AMD or Intel processor will be fine.

Internet Access

You will need to download software from the Internet, and some software being installed will also need to access the Internet in order to be configured.

I only describe a directly connected box, so if you need to go via a proxy server, you'll need to configure that as well. If you don't know how to do this, I will point it out during the installation.

Internet Firewall Modifications

Some software described here needs access through your Internet firewall. If you cannot open some ports, just don't install those services.

Service	*Port*	*TCP/UDP*	*Direction*	*Description*
smtp	25	tcp	both	Send & receive e-mail
http	80	tcp	out	Download gateway components and SuSE updates when required.
razor2	2703	tcp	out	Confirmed spam e-mail razor2 network check
dcc	6277	udp	out	Mass e-mail DCC network check

Installation Summary

Firstly, I will lead you through it all so you shouldn't worry about not knowing how to enable services, edit configuration files or compile software.

This guide starts with a default OS installation along with additional packages that SuSE can provide for a complete setup. I have chosen to use wherever possible SuSE packages for simplicity.

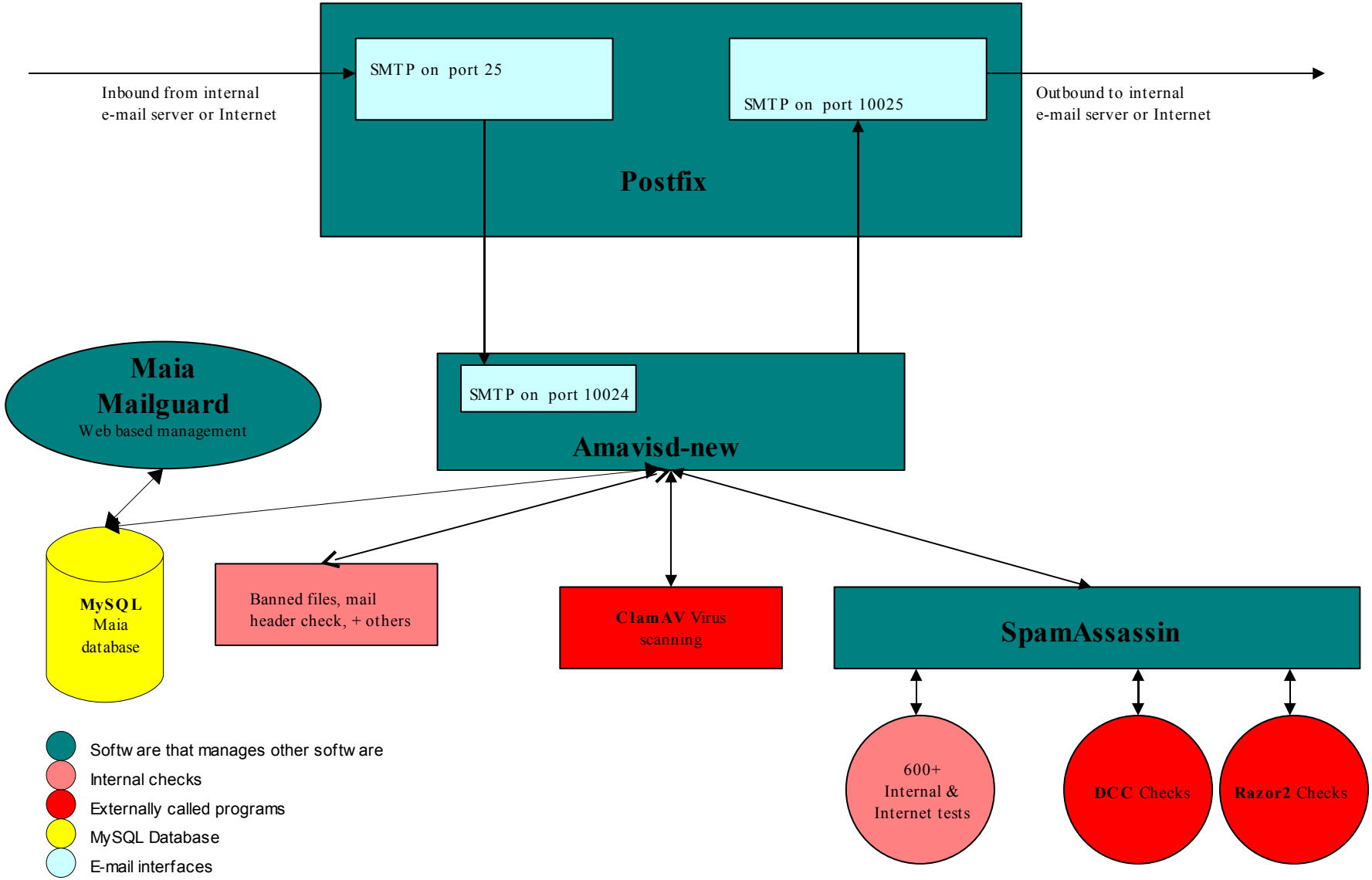
After installing the OS, you will configure all components before testing any of it. This is due of the dependencies each component has with the other. For example after configuring amavisd-new it won't initially be able to connect to the Maia Mailgaurd database as that isn't setup until later.

It will take about a day the very first time you install this if you're new to SuSE / Linux but afterwards it should only take a couple of hours for subsequent installs.

If you think you can handle it, keep reading and good luck.

How it all works

A picture tells a thousand words, so instead of boring you to death with that much text, here it is as a single graphic.



The base SuSE 10 installation

Note this install assumes a clean hard disk. You can of course ask the installer to overwrite whatever else you have on the disk, but that isn't covered here.

Insert the DVD into your mail scanning gateway box and turn it on.

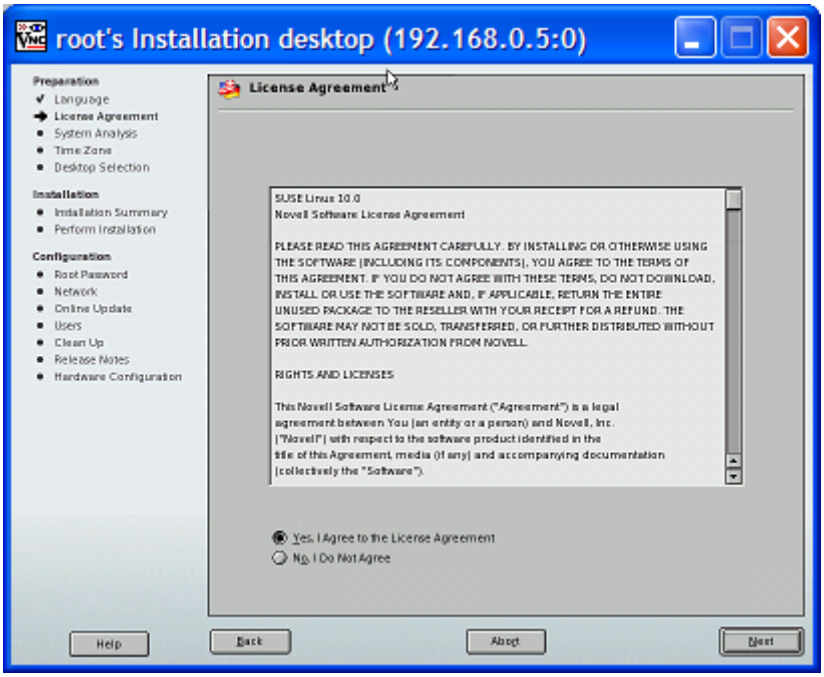
At the main menu, select **Installation**

If at any time through the installation process you find you computer hangs, try going through the install process again, selecting **Installation – ACPI Disabled** and if you still have problems try the **Installation – Safe Settings** option.

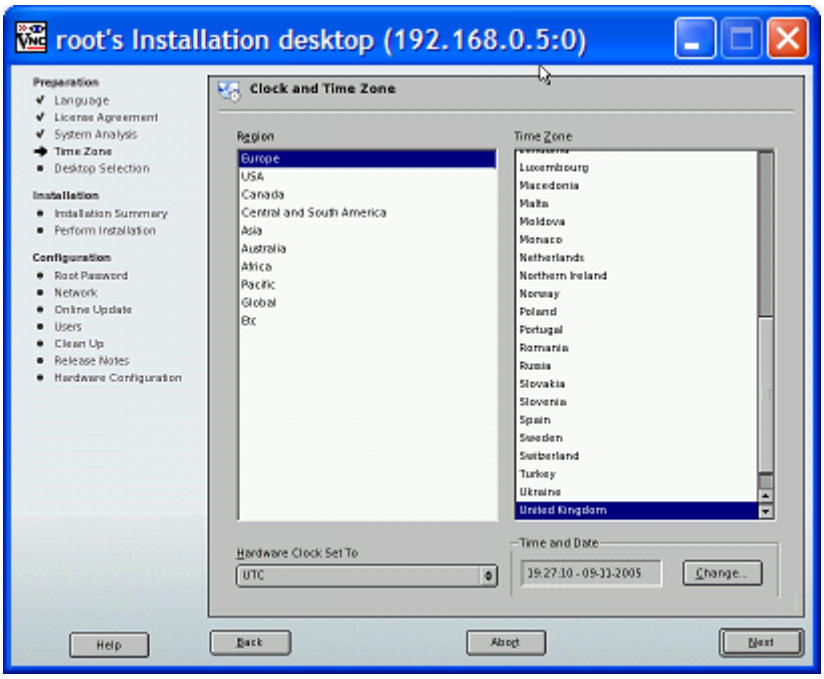
Select your language then click **Next**

If you're installing from CD's you'll be prompted to perform a **Media Check**. That is, you can check the CD's you've burned to ensure the CD/DVD reader can successfully read all the files on the disk. If you want to be safe, click the **Start Check** button which will scan CD1, then eject and insert each subsequent CD to check the rest.

If you like a little risk, click **Next** without checking the media first.



At the **License Agreement** screen, have a read, then if you agree, which I hope you will, select **Yes** then click **Next**



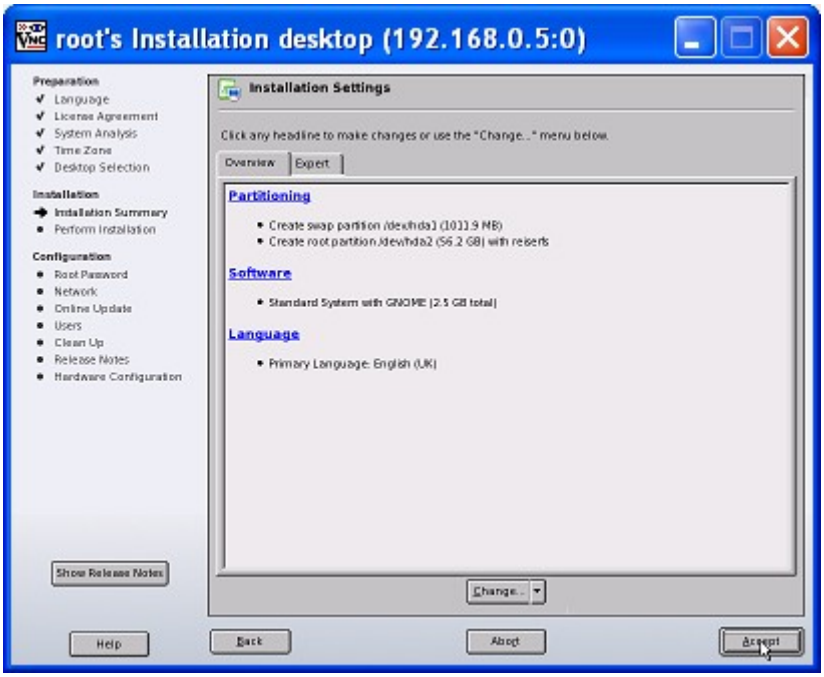
At the **Clock and Time Zone** screen, check that the selected time zone information is correct (obviously change it if required) then click **Next**

At the **Desktop Selection** screen, select the **GNOME** option then click **Next**. This is the setup I will describe, so if you don't know any better or simply want to stay on the track with this guide, select this option. You can choose Other to have a much leaner installation but you'll be on your own to figure out all the individual package selections you'll need to select.

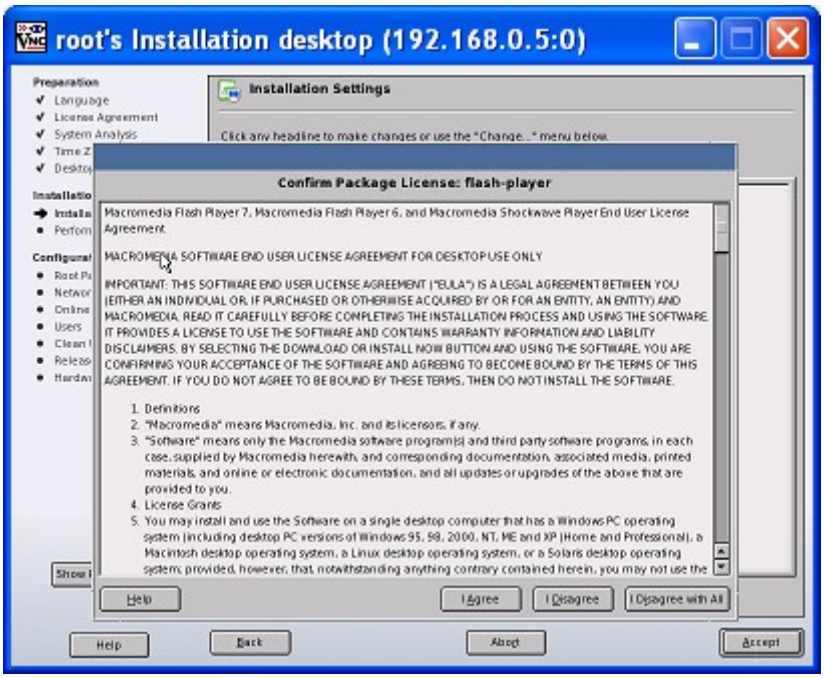
Note: Gnome is simply my personal preference. I also include notes on using some Gnome specific software but you are obviously free to install any GUI you like.

You'll be installing as much software as possible from YaST, making it easier to manage and update, but unfortunately not everything is available this way and some software you will be installing later on from Internet downloads.

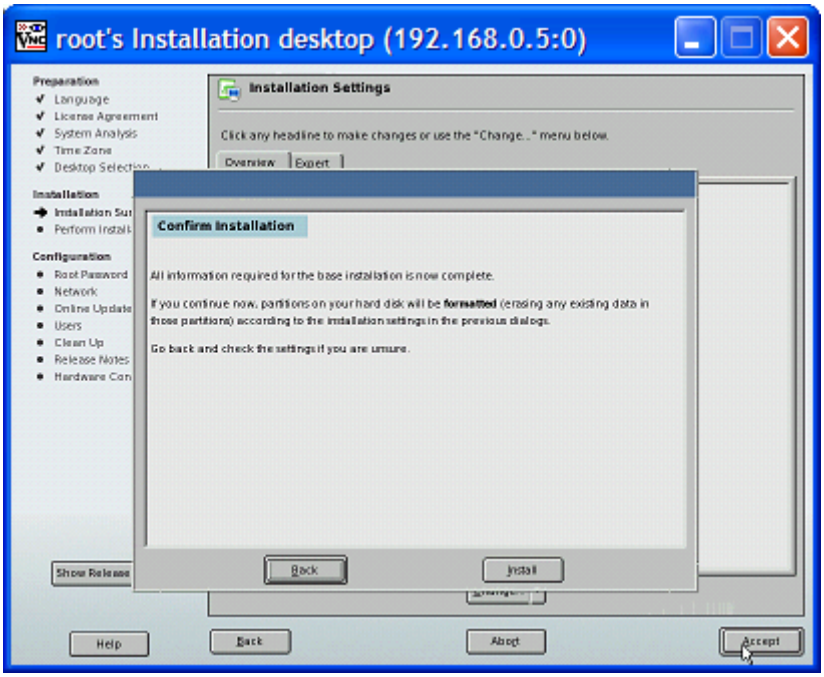
Technically you don't need a graphical system at all, but because the Maia Mailguard web front-end is being installed to configure and manage this system day-to-day, I think it's a good idea to be able to test the system locally, without the requirement of another computer. If you don't like that idea, then install it anyway and call it a test box, so you can get familiar with how all the components work, then re-install the system again to your liking.



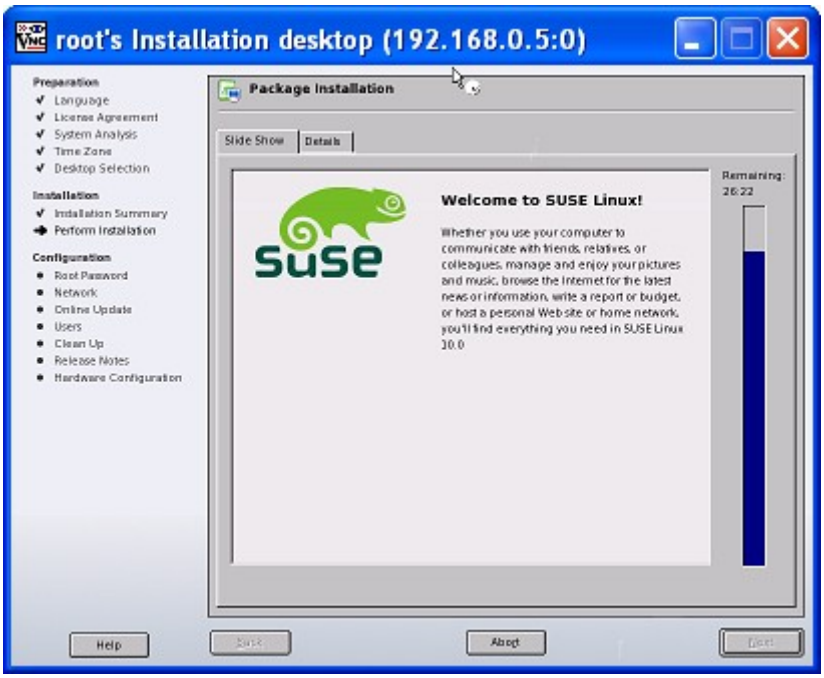
At the **Installation Settings** screen, just click **Accept**.



Read any license agreements, and choose the appropriate option – generally that would be the **I Agree** button.

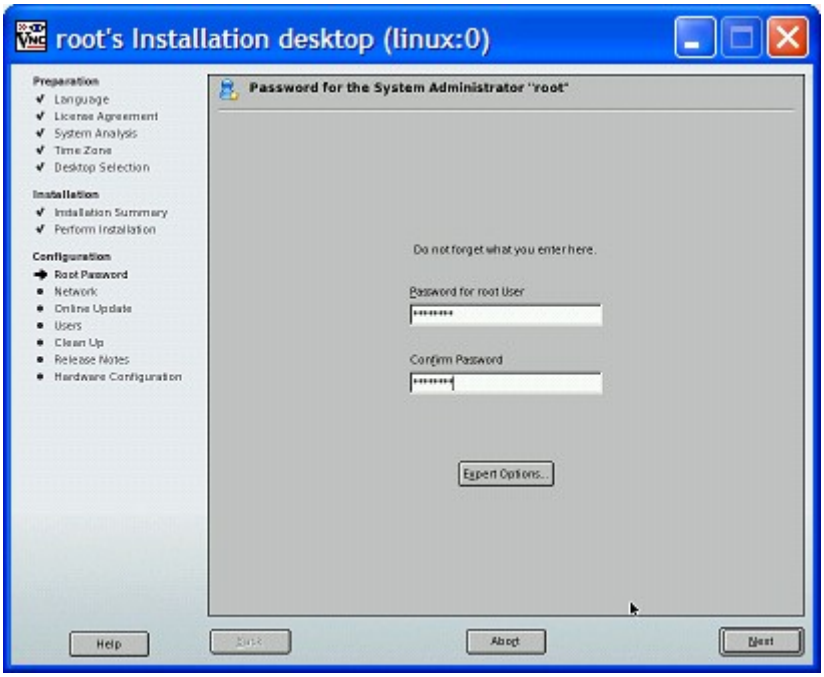


Click **Install** to confirm the choices and now it happily goes off to perform the install.

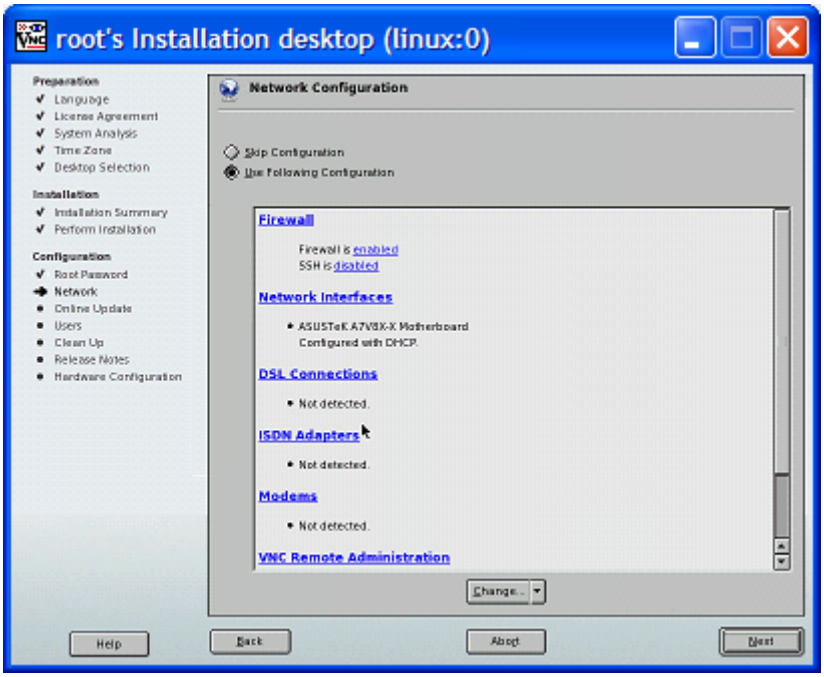


Grab a coffee at this point and relax.....

A nice tip is that after about a minute it will tell you approximately how long the DVD will take to finish installing the packages.

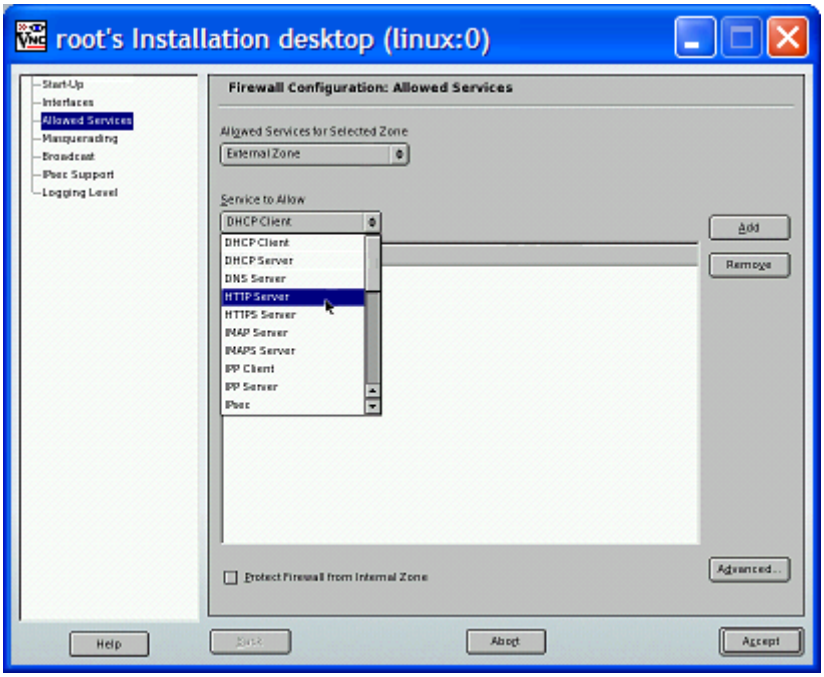


At the **System Administrator** password screen, enter a password then click **Next**.



At the **Network Configuration** screen, you need to change both the firewall and network card settings. Start by clicking the **Firewall** link

The firewall settings must be changed to allow e-mail from remote hosts, access to the web based management and remote administration.

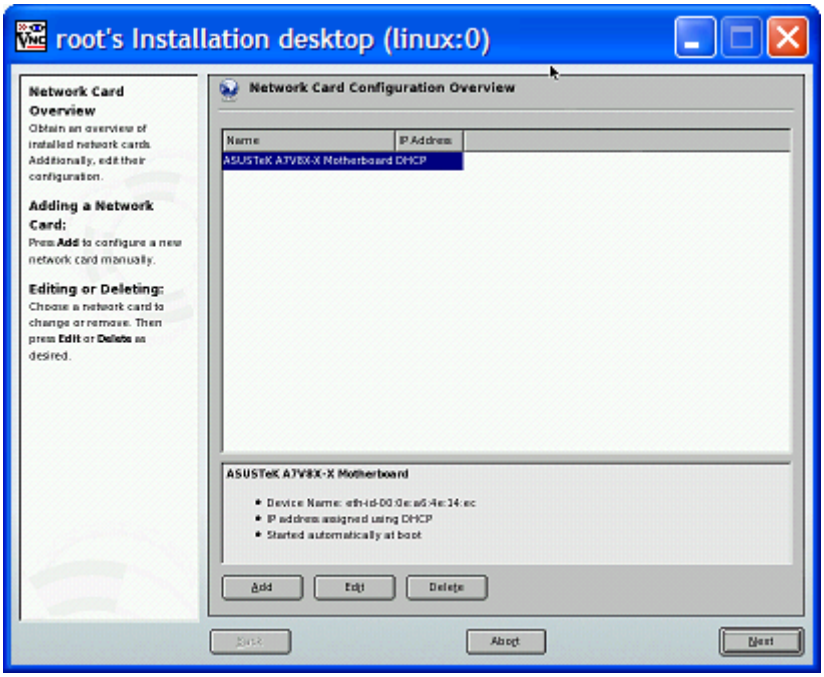


Highlight **Allowed Services** on the left, then from the **Services to Allow** drop down box, select **HTTP Server** then click the **Add** button. Repeat the process for the **Mail Server** and **SSH** services as well.

HTTP Server will allow you to log into Maia Mailguard from any PC.
Mail Server will allow incoming SMTP traffic.
SSH will allow remote login for administration of the server.

Click **Accept** to return to the previous menu.

Now click the **Network Interfaces** link.

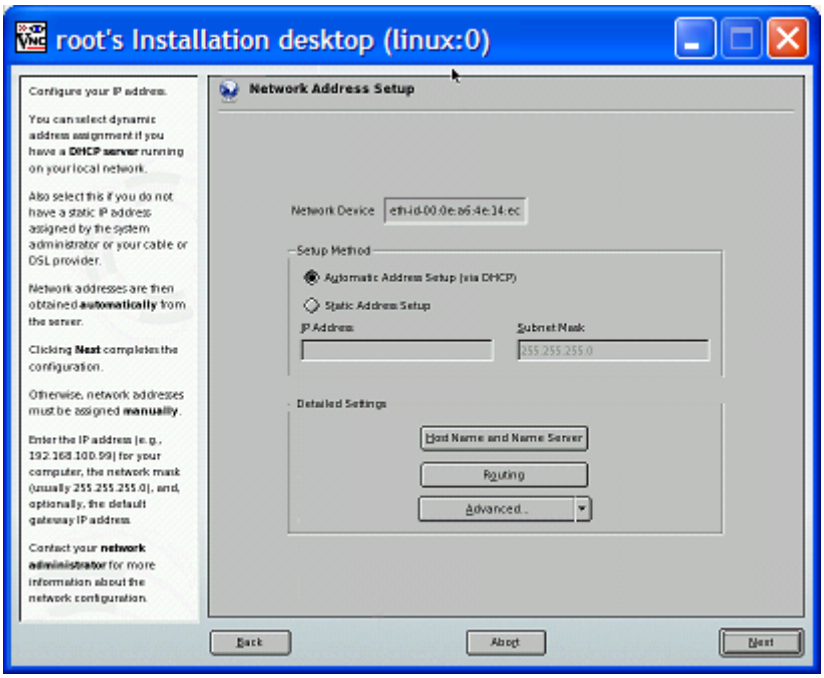


SuSE should have already detected and configured the network card for DHCP and should be listed in the main window section. If not, then click the **Add** button to try and add it manually. You will need to know what type of chipset your network card is using to install it. If you run into trouble, skip out of this guide

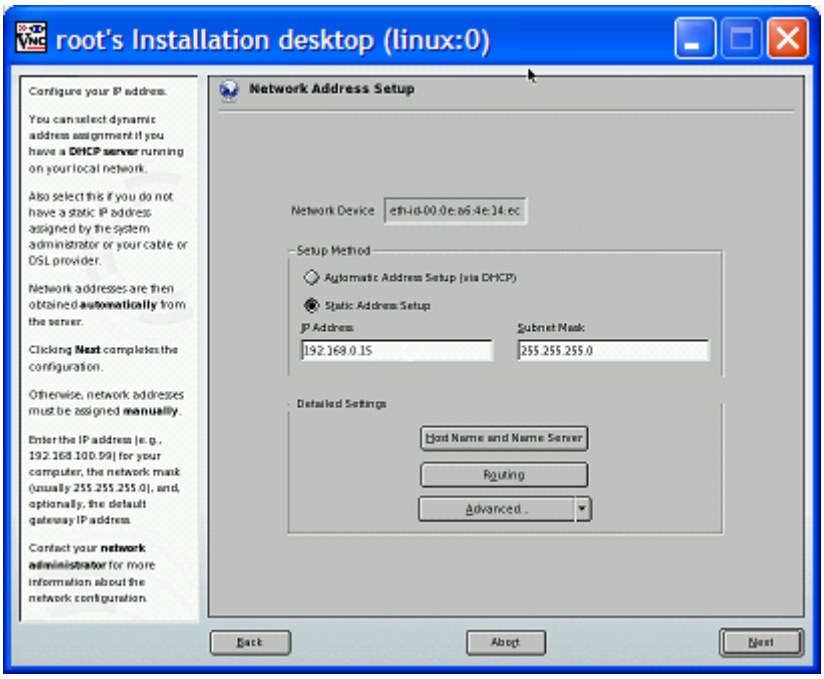
and pop over to an on-line SuSE forum such as http://support.novell.com/forums/ for some help. Once that's taken care of, come back here.

This computer is going to need a static IP address and should also be named something other than 'linux' which it is by default, so to change these settings click the **Edit** button at the bottom of the screen.

At the next screen, select **Edit** to edit your network card.

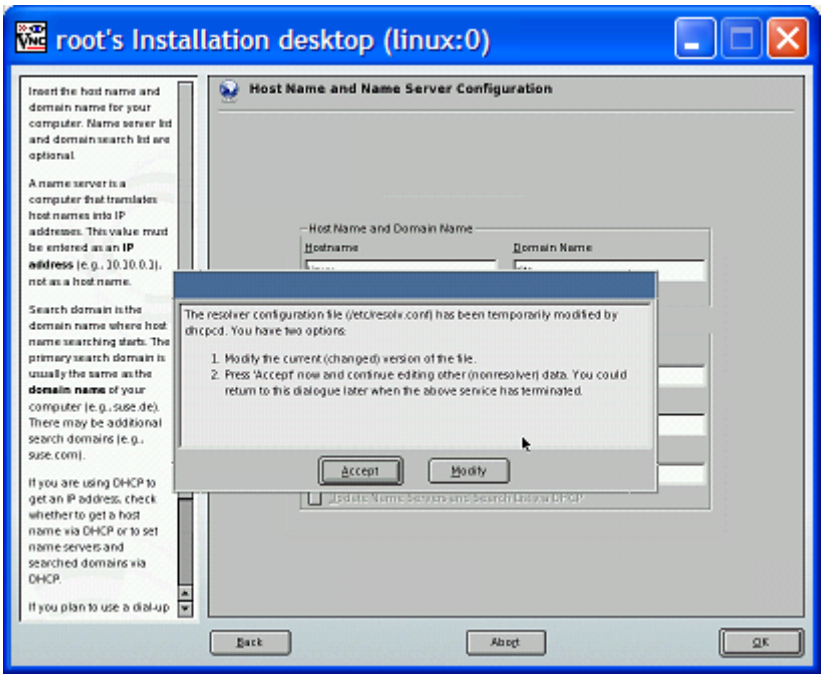


Typically mail gateway servers are configured a with static IP address, so that you can use port forwarding to forward all incoming SMTP traffic to it, or address mapping to map a static external address to the internal address of your gateway. There are so many ways to configure this, but I'm going to show you just one. It's simple and works.



Select **Static Address Setup** and configure the server with an internal network IP address and network

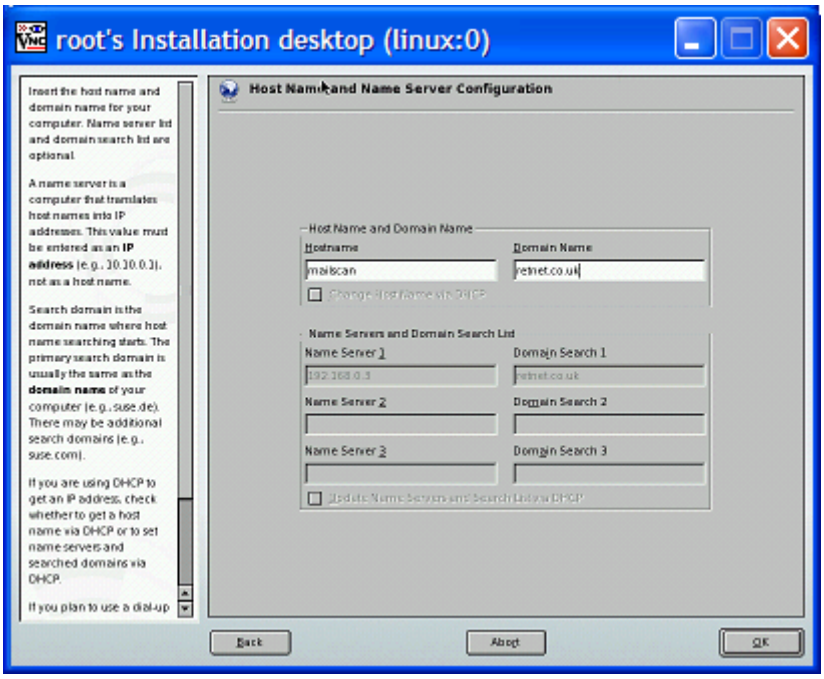
mask, such as 192.168.0.15, 255.255.255.0.



Now select **Host Name and Name Server**.

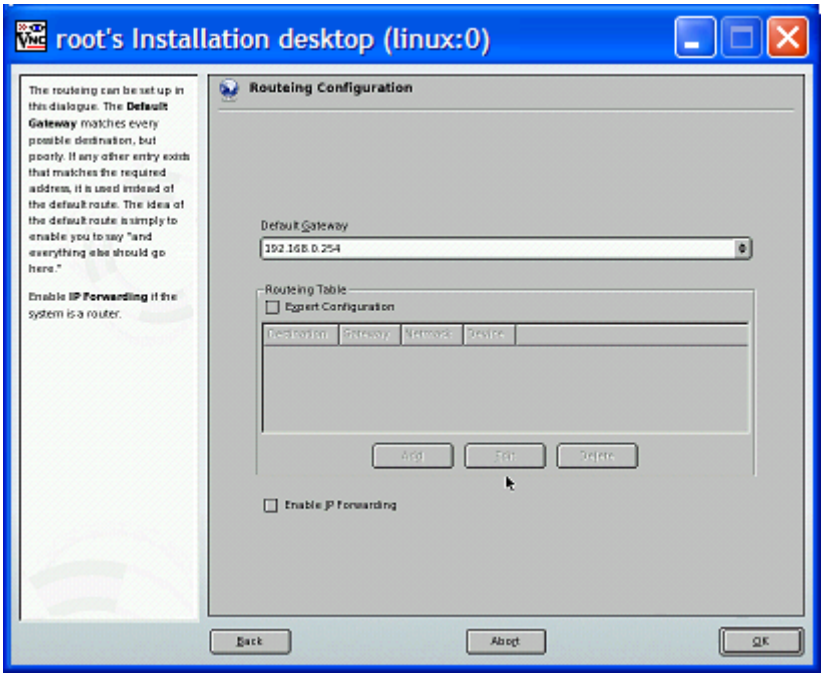
You may be prompted with a message saying the DNS settings have been modified, which will be true only if you're installing this over a remote VNC connection, as I am, and those settings were changed by DHCP when the computer got an address earlier.

You should click the **Modify** button, and confirm all DNS and routing configuration is okay.



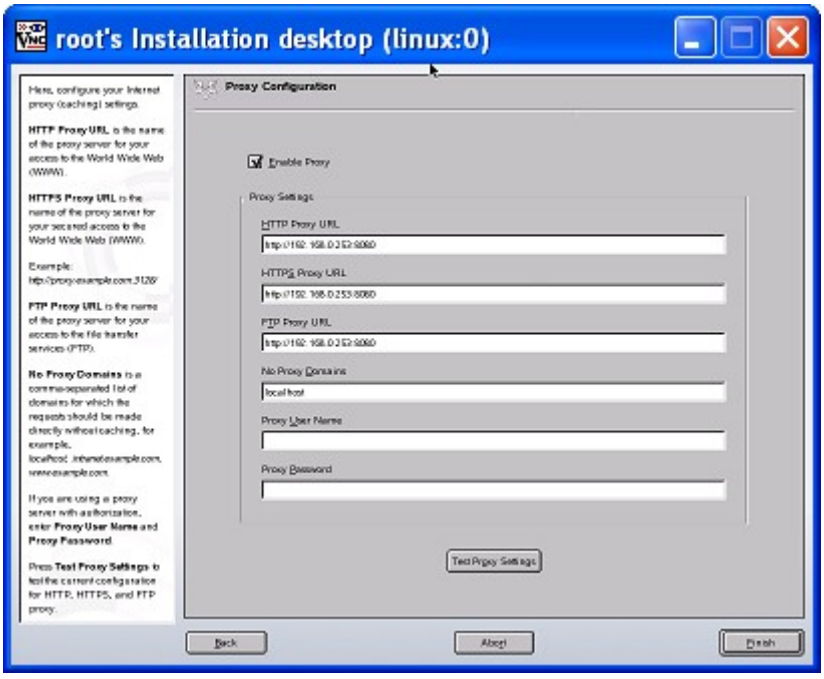
Now fill out the server's Host Name and local domain name then click **OK**.
If you did not see this warning, you will also have to fill in at least one name server (DNS server) and the

default domain name of the box (such as retnet.co.uk). After you've filled in all the information, click **OK**.



Select the **Routing** button, fill in the default gateway address of your network then click **OK**. Change it as required.

Click **Next** twice to take you back to the **Network Configuration** screen.



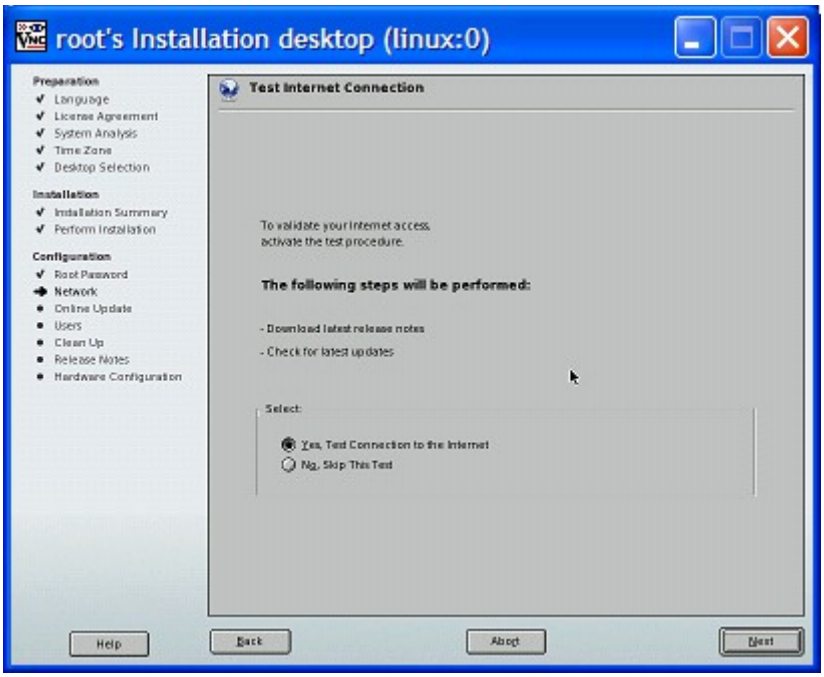
If you go through a proxy server, scroll down the list of options and click the **Proxy** link, select the **Enable Proxy** option, then fill out the proxy details as required.

The syntax for the URL is http://ServerOrIP:port so in the case above, it looks like

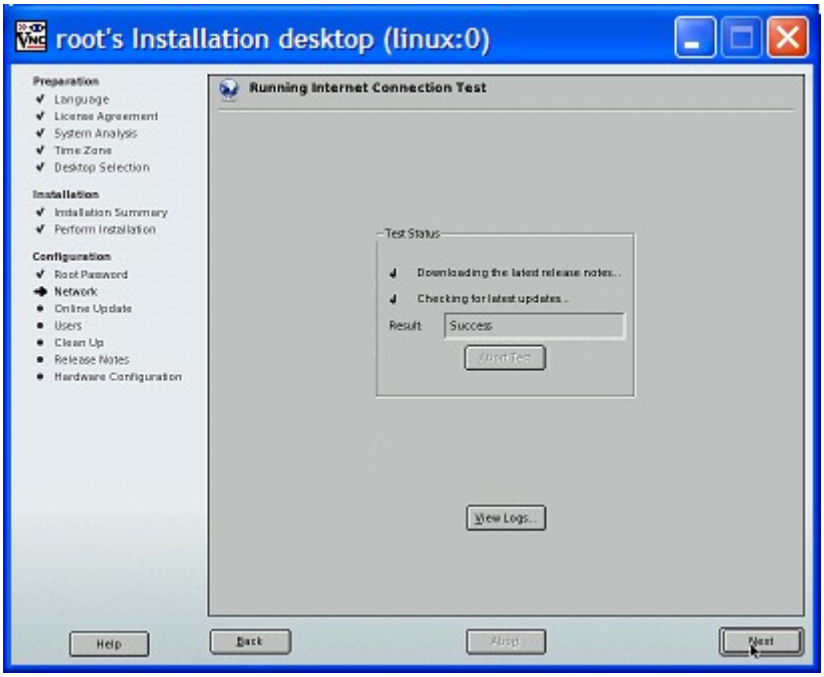
http://192.168.0.253:8080

Once you have finished your changes here, click **Finish**. You will be presented with a note from YaST regarding software that is compatible with this proxy setting. Once you have read the notice click **OK**.

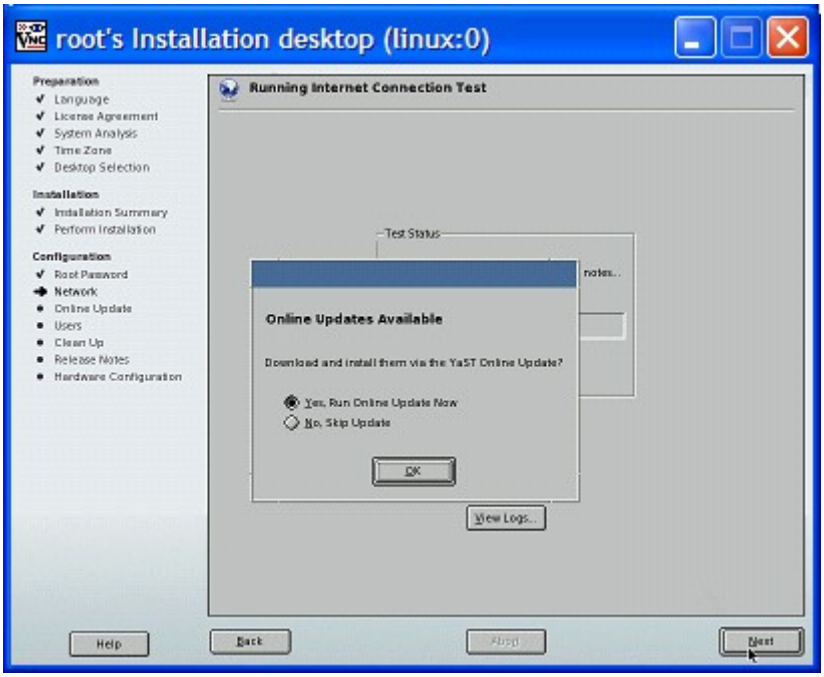
That is all for the network screen, so click **Next**.



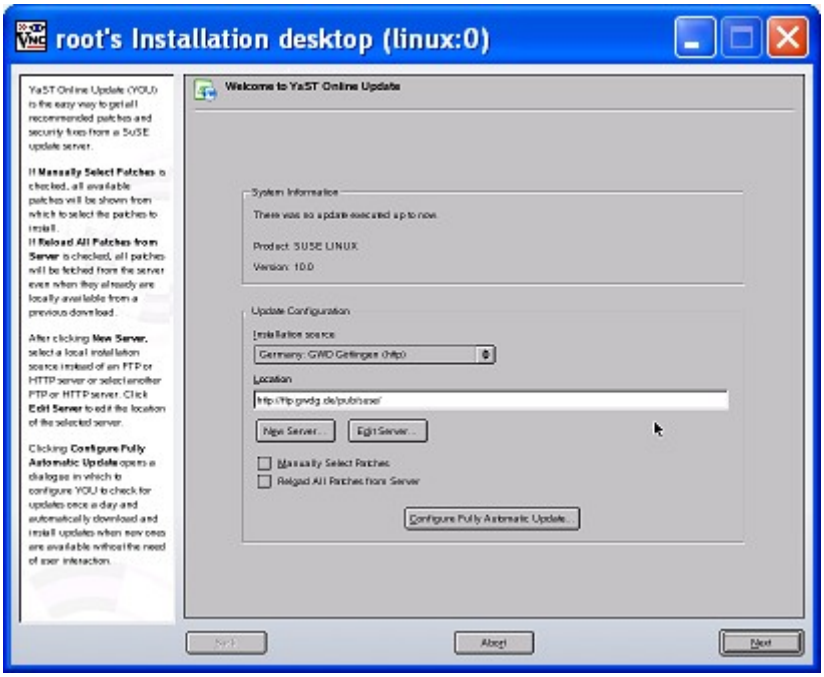
Now YaST saves the network configuration and moves onto the Internet test. If the server has Internet access at this point you should allow it to do this, as it includes an on-line update of all packages you've just installed. Skip the test if you don't have Internet access.



After the Internet test is complete, click **Next**.

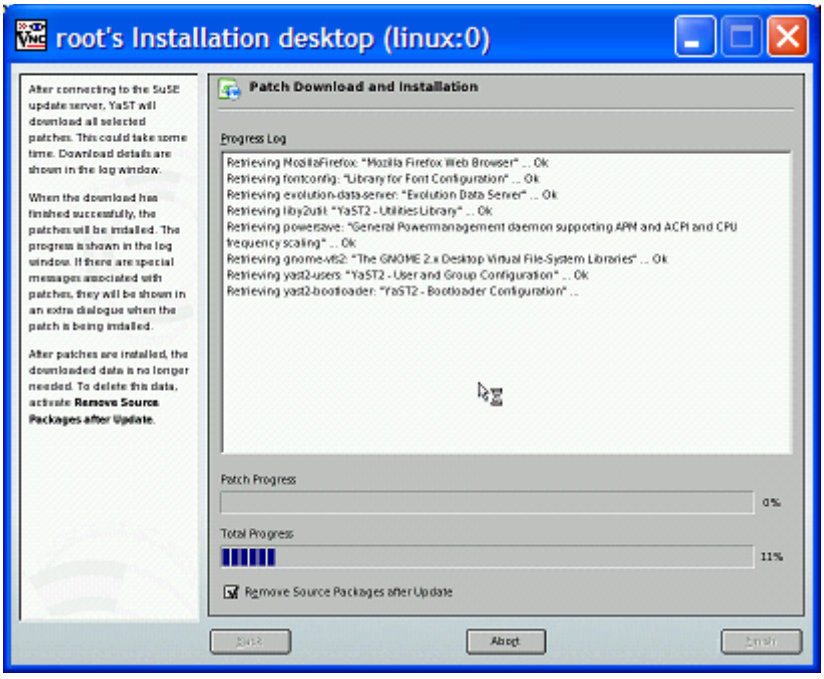


If the test was successful you'll be presented with an **Online Updates Available** message. Accept the defaults and click **OK**.

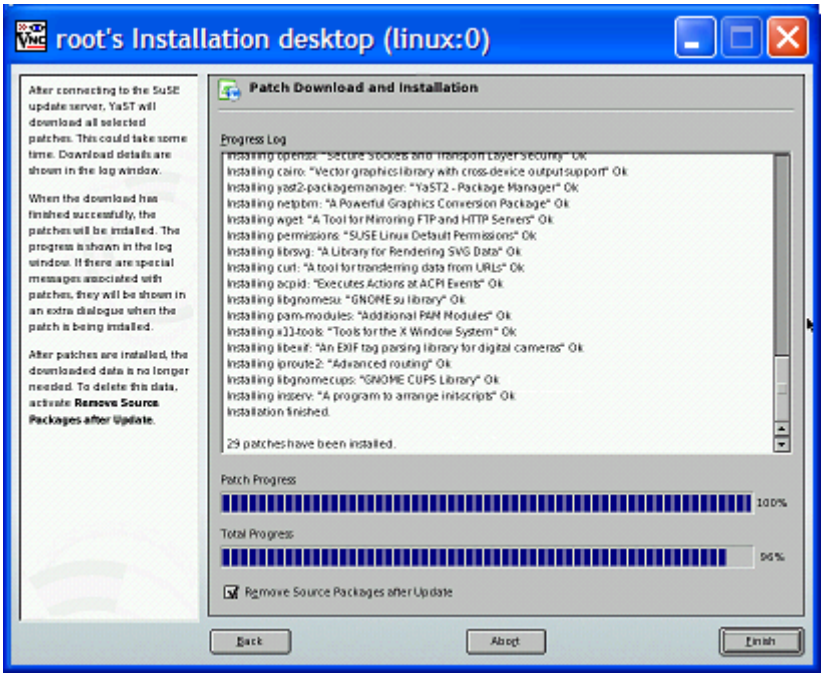


At the **Welcome to YaST Online Update** screen, clear the tick from the **Manually Select Patches** box then click **Next**.

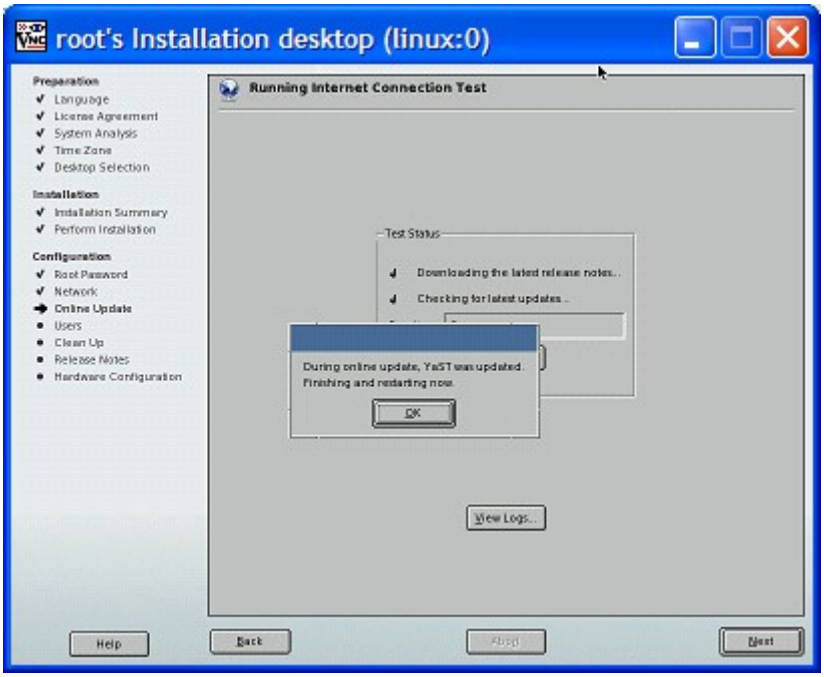
If you are warned about a kernel patch at this time select **Skip Patch** else the installation might go south during the next phase of hardware detection.



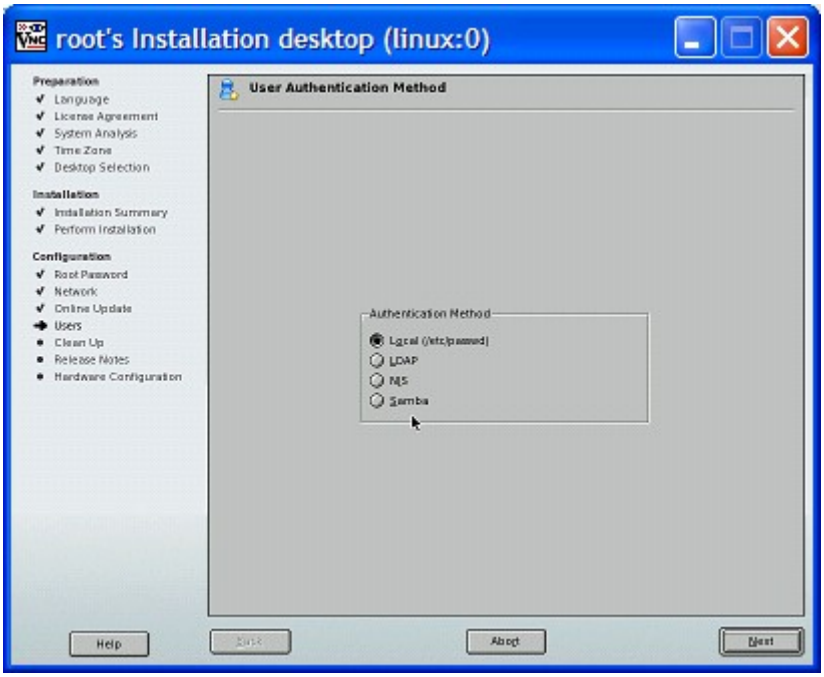
Now get another coffee and relax a while...



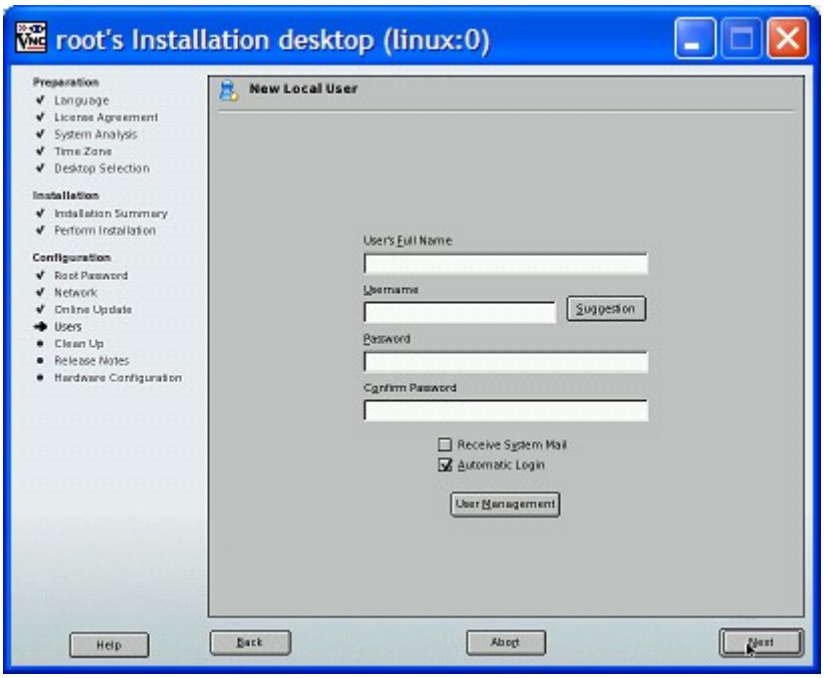
Once it's finished, click the default **Finish** button.



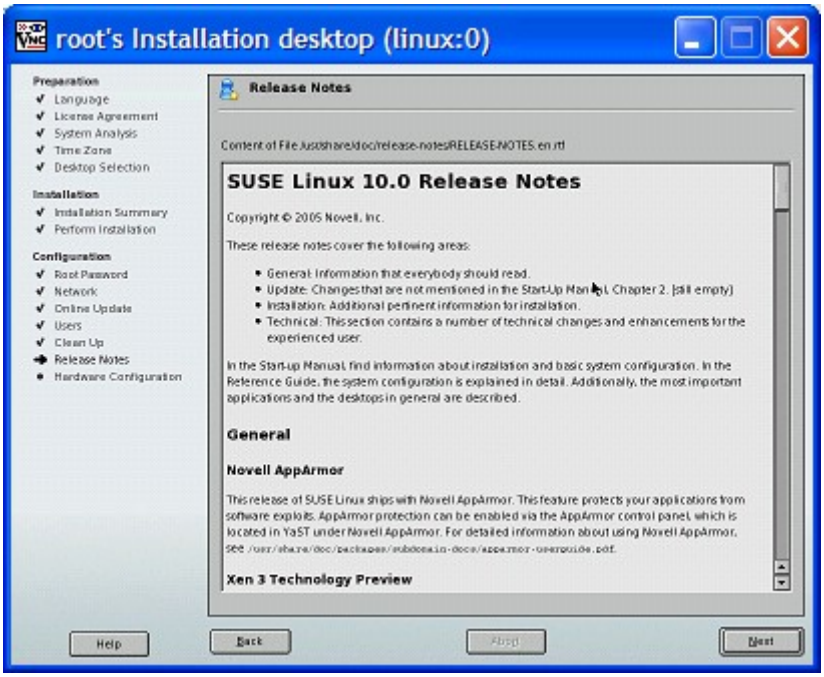
Due to YaST itself (the installer) being updated, it needs to be re-started and you will be notified of this. Click **OK** as you don't have a choice.



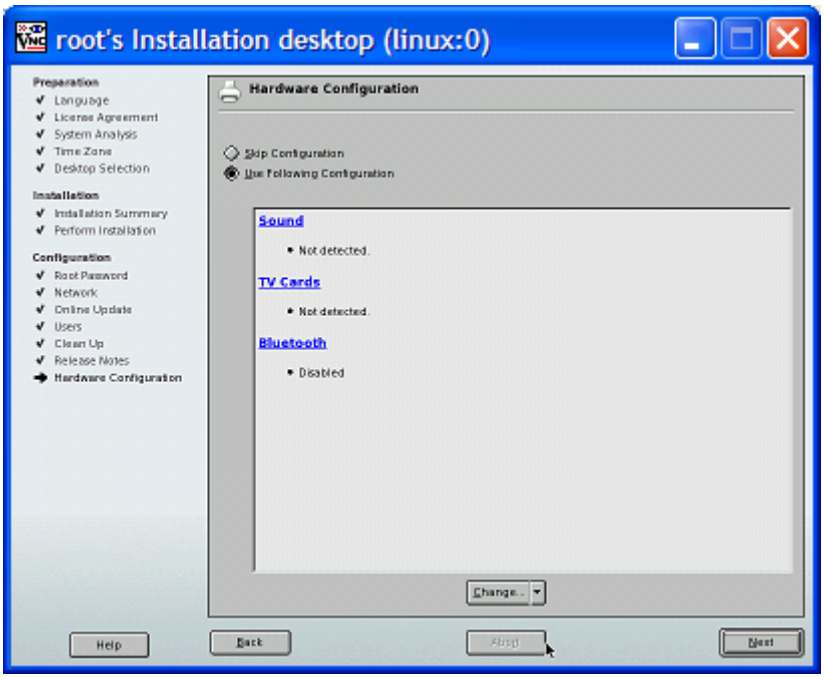
The next screen you see relates to how Linux will authenticate users trying to log in. Leave it at the default of **Local** and click **Next**..



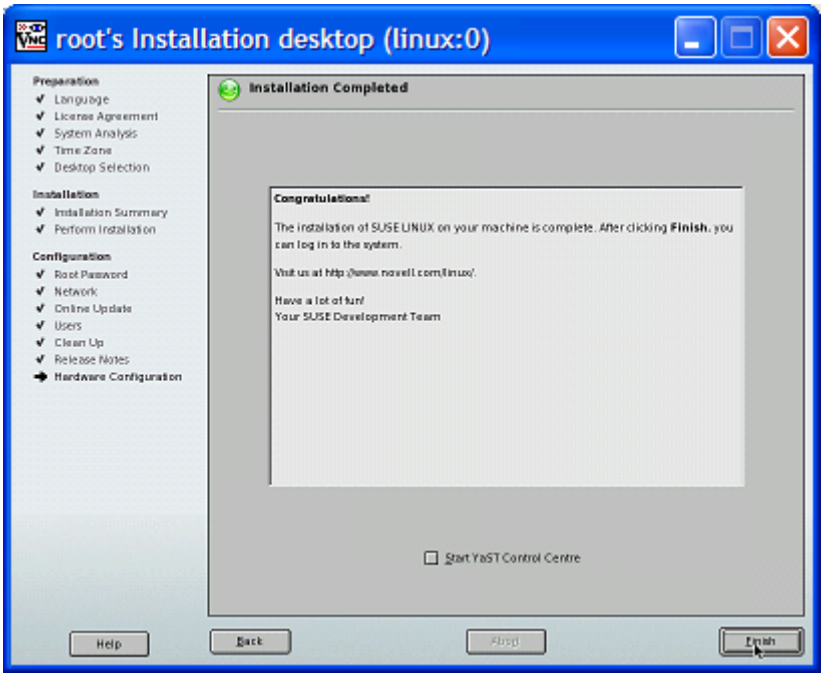
At the **New Local User** screen, leave it blank as you don't need to create any additional local users so just click **Next** and accept the **Empty User** dialog that pops up.



After reading the release notes, because they are really worth it, click **Next** to move onto graphics, sound, TV cards and bluetooth.

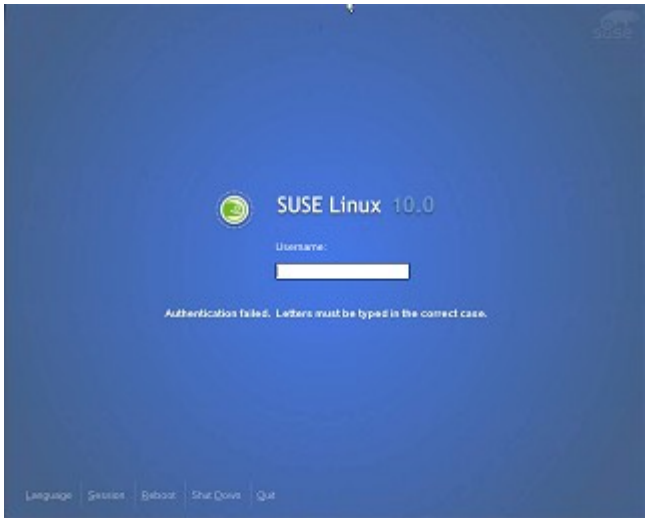


The example screenshot above will most likely be different than yours, which should include a graphics option (due to me installing via a VNC connection). To the best of your knowledge, check that the auto-detect settings are correct then click **Next.**



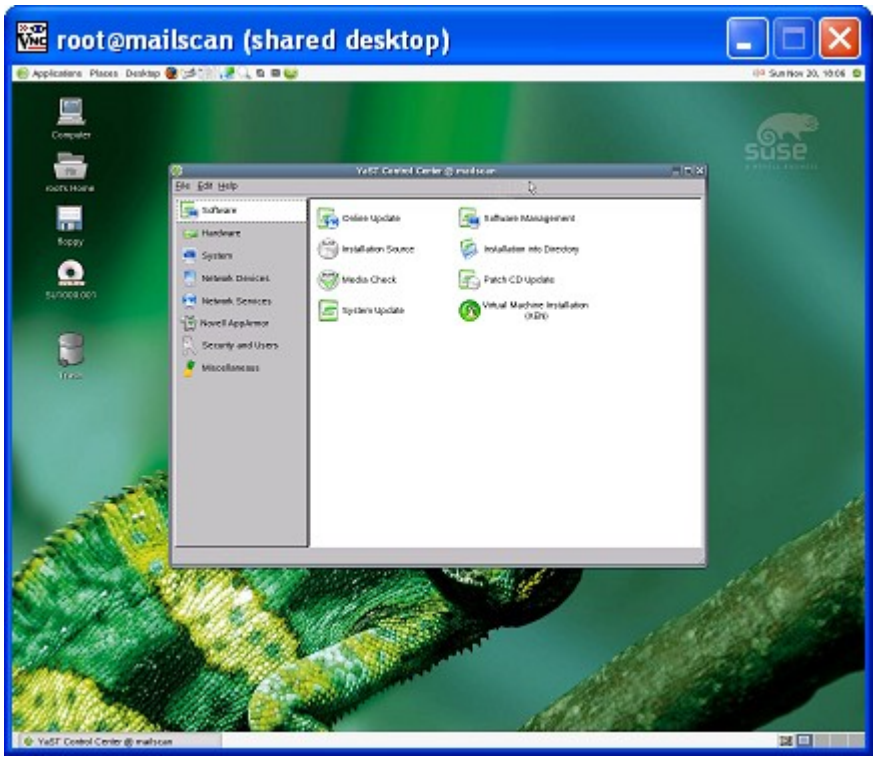
Now the installation is complete. Click **Finish** for the installation to exit and present a login screen.

Installing additional Software Packages

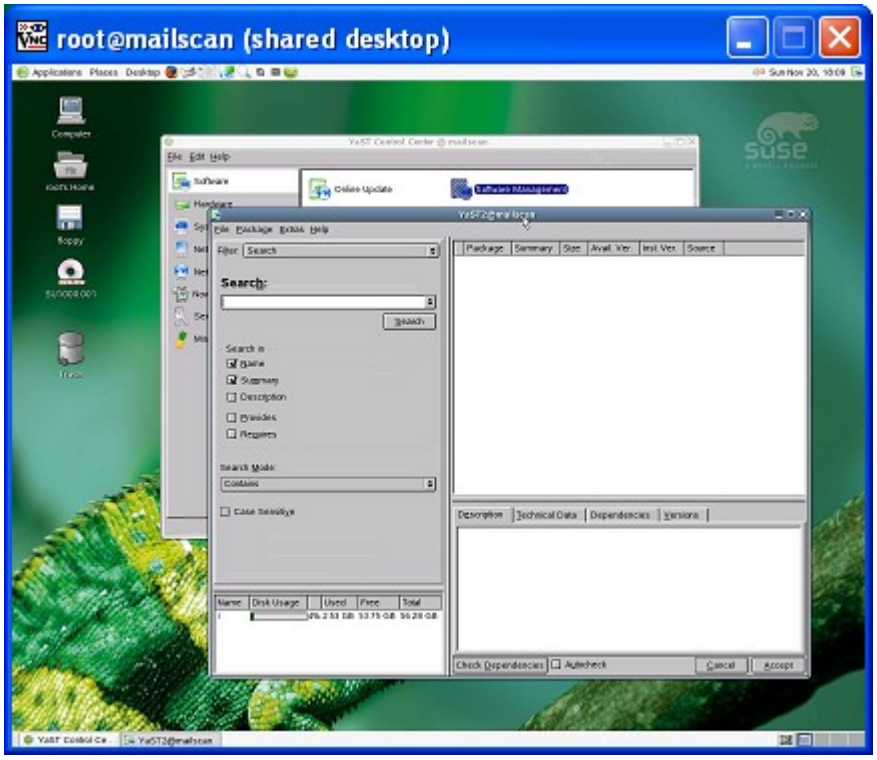


Login as the user **root**

Now you need to install some additional software. To start, select the **Desktop** menu then click on **YaST**.

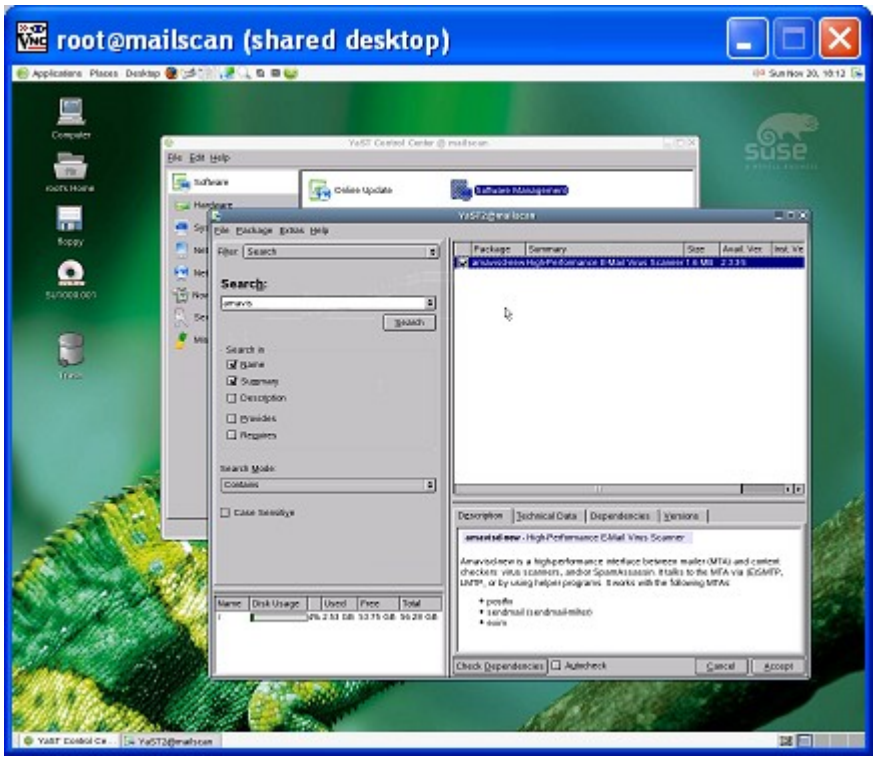


Select the **Software Management** link in the right hand pane.



You will be presented with a search box, making it easier to find SuSE software packages.

In the search box, type **amavis** then press **[Enter]**.



Select the **amavisd-new** package (this program controls how other programs scan e-mail, such as anti-virus and anti-spam scanners, as well as having control over attachment blocking and a few other options) by clicking the box to the left of the package name.

Now repeat the process for the following:

- **apache2**
  The famous Apache web server v2

- **apache2-prefork**
  There are 2 ways Apache can run. Using the pre-fork method, each server runs in it's own process meaning it takes more memory but is more stable. Using the MPM (Multi-processing Module) method increases performance for the sake of reliability. I like my boxes stable, hence my preference, but you can choose the other.. doesn't make any difference to the configuration.

- The following apache2 Plugin modules:
  - **apache2-mod_perl**
  - **apache2-mod_php4**

- **gcc**
  gcc is a GNU C compiler, used to compile DCC.

- **gd**
  A drawing library for programs that use PNG or JPEG output.

- **libmcrypt**
  Used to encrypt data using various encryption methods. E-mail will be held in the MySQL database in encrypted form, ensuring no prying eyes can see any of the e-mails even if they access the database.

- **mysql** *(If you decide to use an existing MySQL DB server, bear in mind there are some changes you need to make to MySQL which I'll explain later, that you may not think are a good idea for your other databases running on it... it's your choice, of course!)*
  The famous MySQL database, used to store most program settings along with all unconfirmed e-mail and e-mail statistics.

- The following perl encryption modules:
  - **perl-crypt-blowfish**
  - **perl-crypt-CBC**
  These are the perl programs that will actually encrypt relevant data.

- **perl-DBD-mysql**
  A Perl database interface library.

- **php4**
  A famous scripting language commonly used for web programming. This is what Maia Mailguard is written in.

- **php4-bcmath**
  is a library of binary calculator functions for php.

- **php4-gd**
  PHP functions to manipulate graphics using the gd library

- **php4-imap**
  PHP interface use to authenticate against an IMAP source

- **php4-ldap**
  PHP interface to LDAP

- **php4-mcrypt**
  PHP interface to the mcrypt library.

- **php4-mysql**
  PHP functions to access a MySQL database.

- **php4-pear**
  PHP interface to another scripting language called pear, which Maia uses.
  There are also a few pear packages you'll need to install but this aren't included with SuSE, so you'll do this after the initial installation.

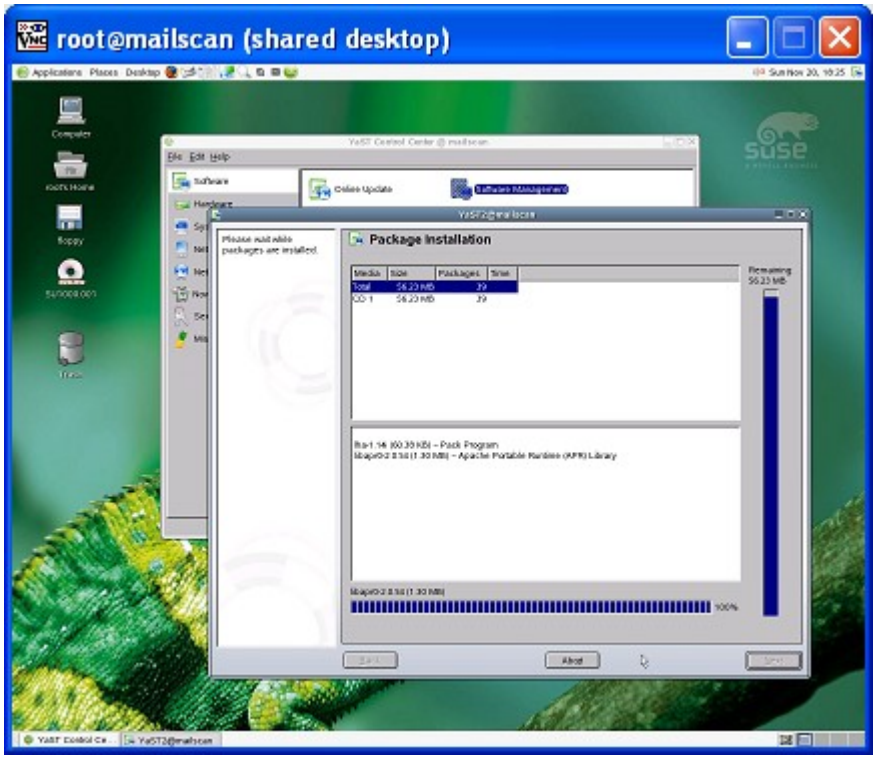
- **php4-session**
  A PHP library that allows subsequent accessing of data.

Removing the Beagle Desktop search tool

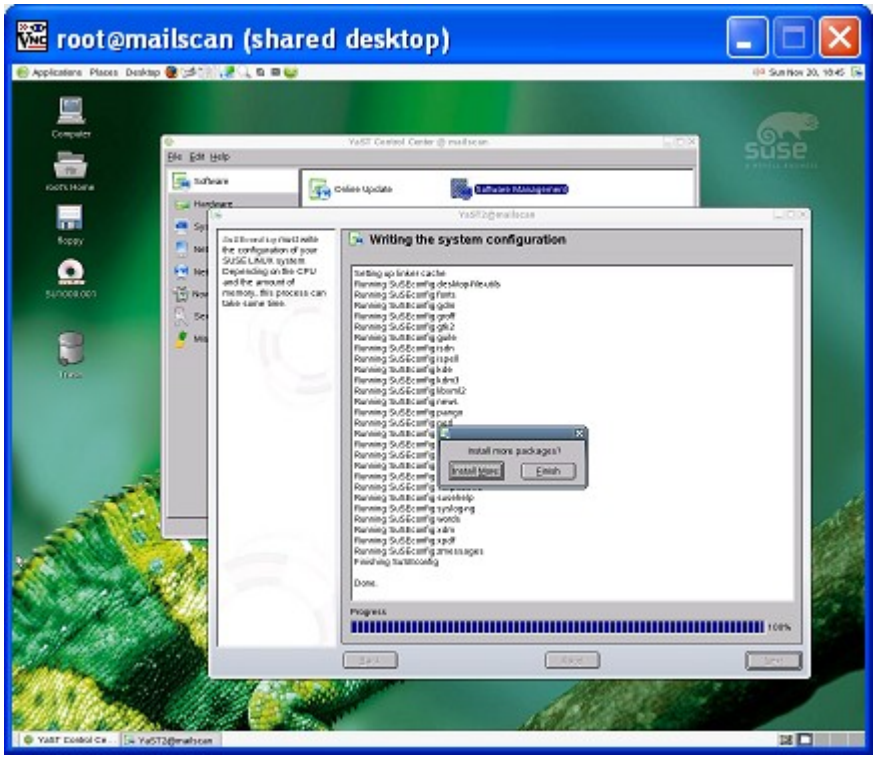
By default the beagle desktop search tool will be installed. This creates a scheduled (cron) job that runs at 4:30am every morning. This won't cause any great concern for your system's overall performance, but it is probably best to remove it, as it doesn't add value to the solution.

Search for and remove the tick from the **beagle** application (leave libbeagle as it's a dependency for other software).

Now click **Accept**. You will see a message about additional packages that must be installed due to dependency issues. Click **Continue** for the installation to start.



The installation will only take a minute or two.



Once it's finished, click **Finish**.

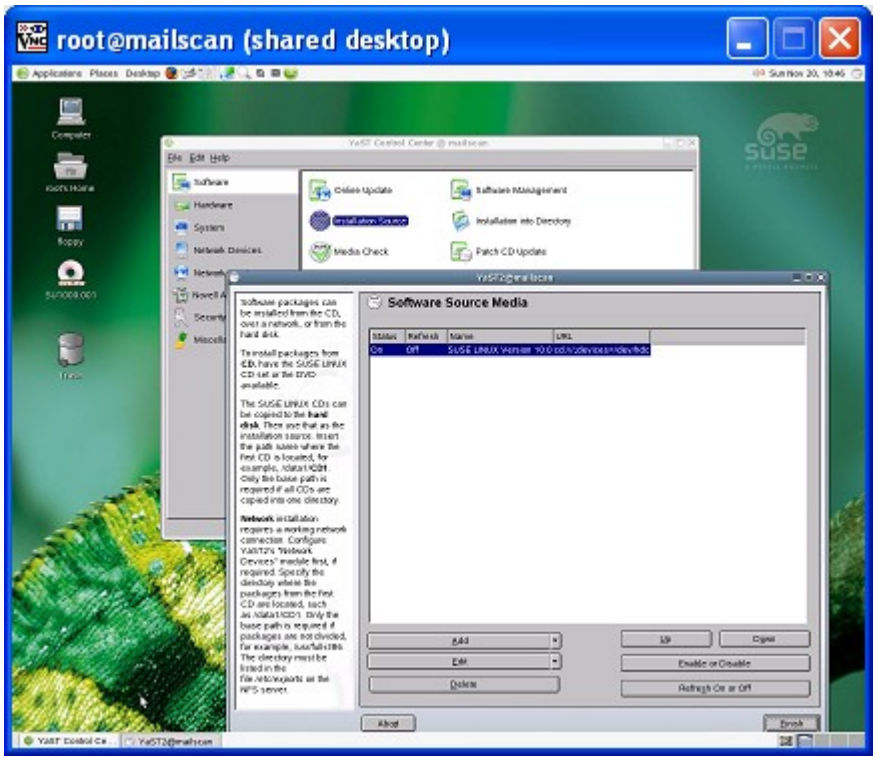
Strangely some packages that are available on the Novell SuSE 10.0 retail DVD and the OpenSuSE OSS 10.0 DVD aren't available on the Novell Evaluation DVD... so to get these extra hidden gems, you'll need

to add an extra software repository to YaST so it can find the additional packages needed.

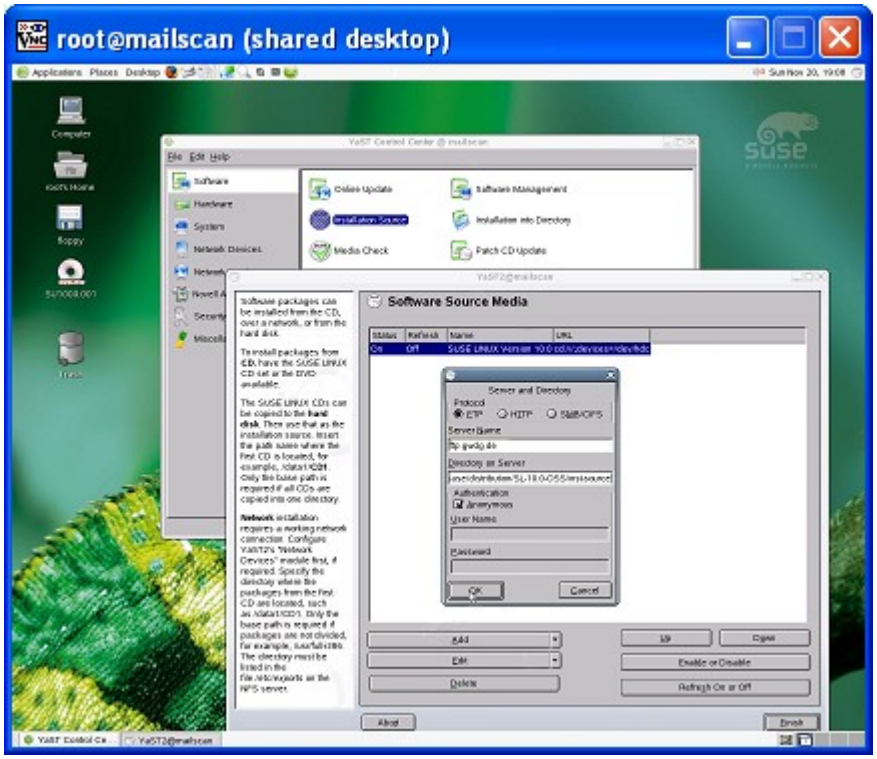
I'm going through this now instead of before when you installed the first set of packages just to highlight what packages aren't installed by default. You could have of course added this installation source earlier and just installed the lot in one go.

In **YaST**, click **Installation Source**.

To add an on-line software repository for SuSE to install from, click the **Add** button then choose either the **ftp** or **http** options. You should enter details for a mirror of a server close to yourself. A list of OpenSuSE mirror sites is available here and a Novell CoolSolutions feature explaining it in detail is available here

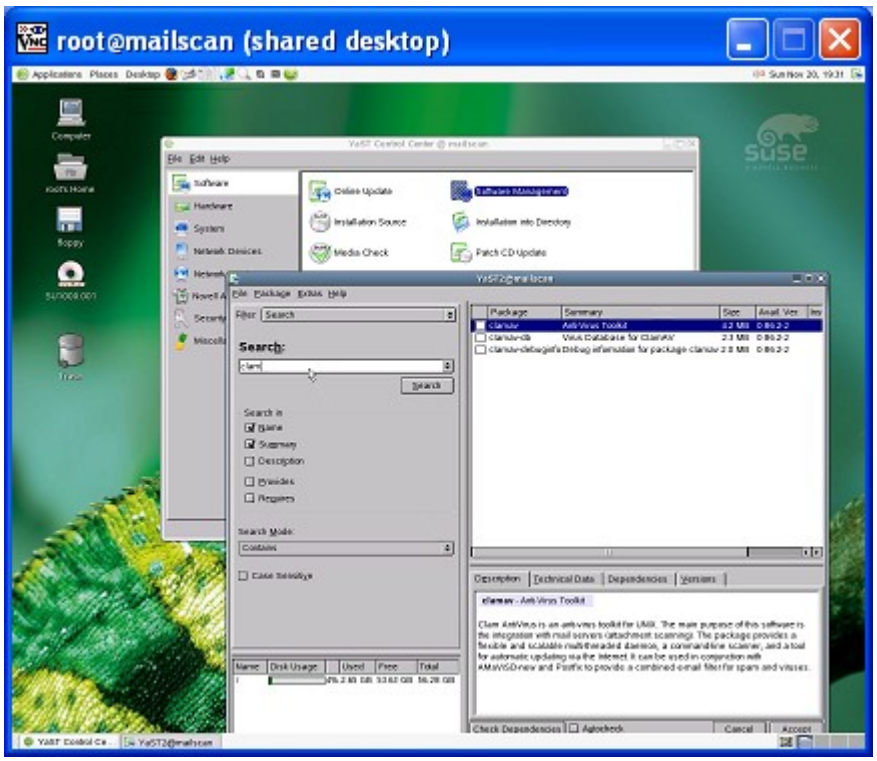


If you are lazy, you can use the source that I chose, and follow the screen shots. If a mirror doesn't work for you, try another. In my case I clicked **Add**, then chose **ftp...** with the following options:



In the **Server Name** field, I typed in: **ftp.gwdg.de**
In the **Directory on Server** field, I typed in: **/pub/opensuse/distribution/SL-10.0-OSS/inst-source/**

Once you've entered your new installation source, click **OK** and it will show as a status of **On** second in the list (with your local DVD being the first).



Now click **Finish** then go back to **Software Management**.

It may take a few minutes for the **Reading package information...** dialog to disappear due to YaST reading the online software repository package list.

Now search for and install the following packages, as you did previously:

- **apache2-mod_auth_mysql**
  An apache module to authenticate users through a MySQL database

- **clamav**
  An anti-virus toolkit for UNIX

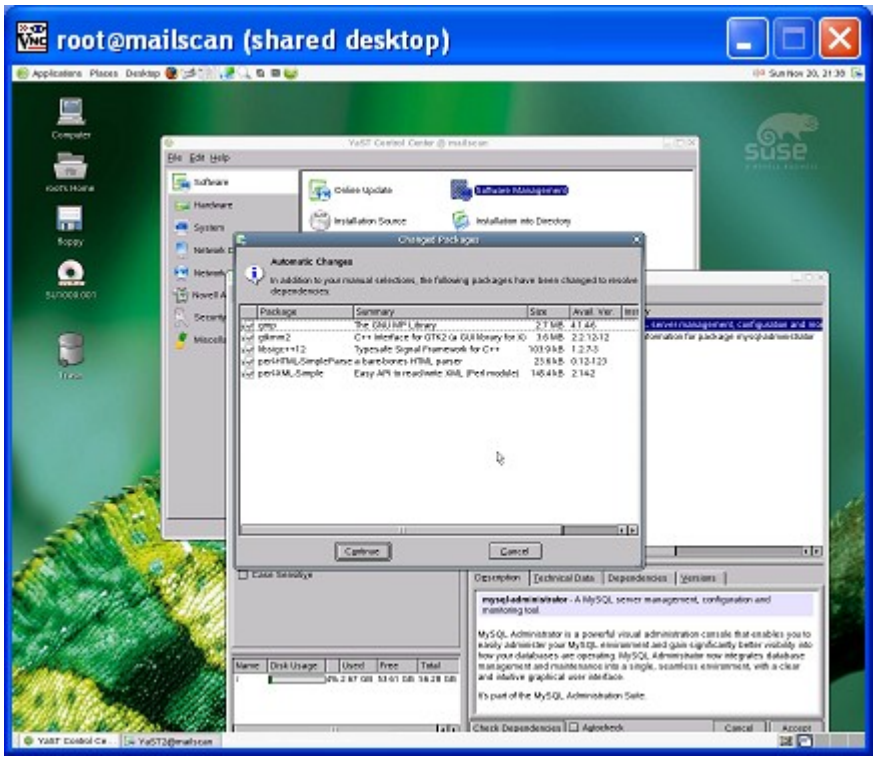
- **clam-db**
  An initial clamav database. You will receive a pop-up message saying if you use freshclam (an updater for the clamav signatures) you don't need the database. Just click **OK** anyway.

- **mysql-administrator**
  A graphical administration and maintenance program to manage MySQL servers. This is an optional package as you can also use the mysql command line tools, but I find it a little easier using this GUI, especially for newcomers to MySQL/Linux.

- **Perl-data-UUID**
  Provides a framework for producing unique ID's.

- **perl-template-toolkit**
  A template system for Perl.

- **razor-agents & perl-razor-agents**
  Programs to communicate with the Razor network to exchange signatures of SPAM.



Now click **Accept**. You will see a message about additional packages that must be installed due to dependency issues. Click **Continue** for the installation to start.

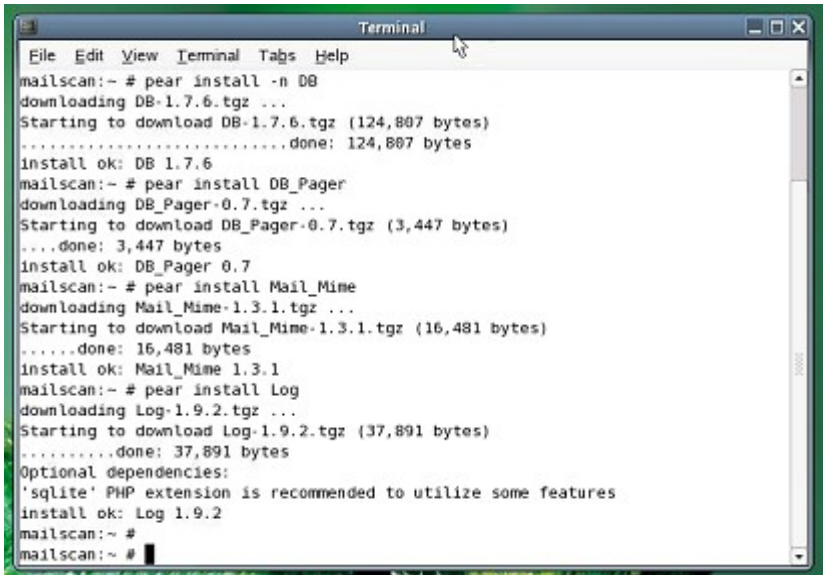
Click on **Finsh** at the end when prompted to install more software.

Since there's new software now installed, it's a good time to check for any updates, so in YaST under the **Software** group select **Online Update** and as before during the initial installation, clear the tick box option for **Manually select patches** then click **Next** and let it do it's thing.

If you are prompted for a kernel update at this time, select the option to install it.
It won't take long this time as you've already updated most of the software.

Installing additional pear modules

There are a few extra dependencies to install for the pear package you selected earlier, which Maia Mailguard requires.



Open a terminal window by right-clicking on the desktop, then select **Open Terminal**

If you need to go through a proxy server, pear is one of the programs that do not honor use the SuSE proxy settings, so you will need to configure this separately.

To set the proxy server, in the terminal run:

```
pear config-set http_proxy http://username:password@ServerOrIP:port
```

where **username** is an authorised proxy user *(only required if proxy authentication is needed)*
where **password** is an authorised proxy user password *(only required if proxy authentication is needed)*
where **ServerOrIP** is the DNS name or IP address of the proxy server
where **port** is the tcp port the proxy server is listening on.

If you cannot install the necessary pear modules in this manner, you will have to manually download the required pear modules from http://pear.php.net and run the same installation commands as below, but substitute the package name with the filenames you downloaded.

In the terminal, run:

```
pear install -a http://pear.php.net/get/PEAR-1.3.3.tgz
pear install -a Net_SMTP
pear install DB
pear install DB_Pager
pear install Log
pear install Mail_Mime
pear install Net_IMAP
pear install Net_POP3
pear install Image_Color
pear install -f Image_Canvas
pear install -f Numbers_Words
pear install -f Numbers_Roman
pear install -f Image_Graph
```

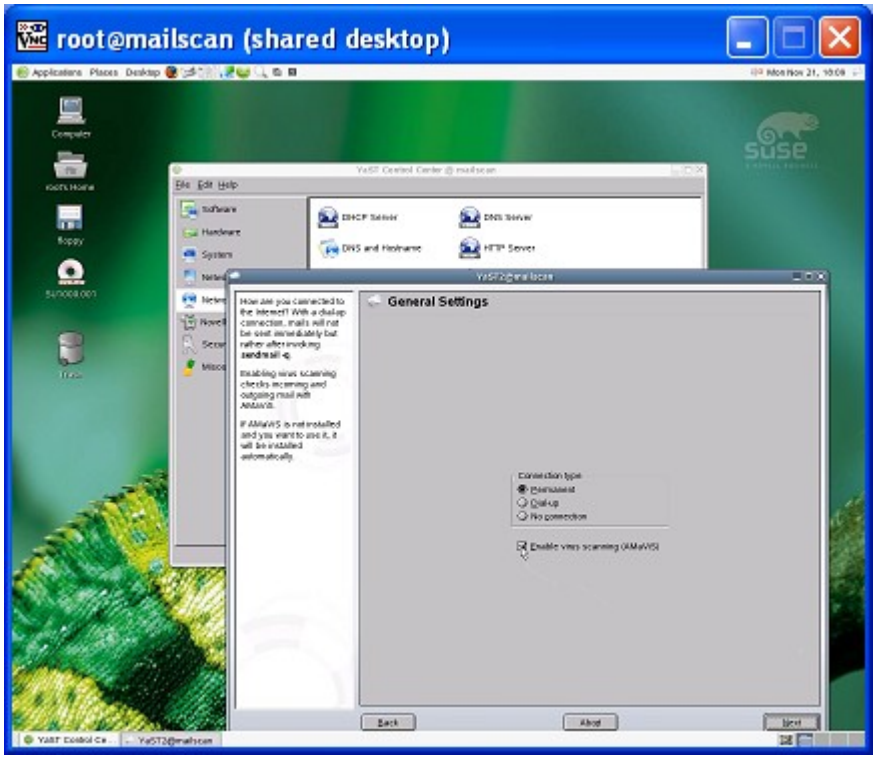
The -a switch tells pear to install any required and optional dependencies
The -f switch is to force install of 'non-stable' releases. These packages are either in Beta or Alpha. Without them chart graphing in Maia Mailguard won't work.

Pear will attempt to go off to the Internet, fetch the packages and install them for you.

You're now done with the basic foundation needed for an excellent mail scanning gateway. There are a couple of extra programs you'll need to download, but that can all happen in good time. Next you start configuring the mail components you've already installed.

Setting up postfix

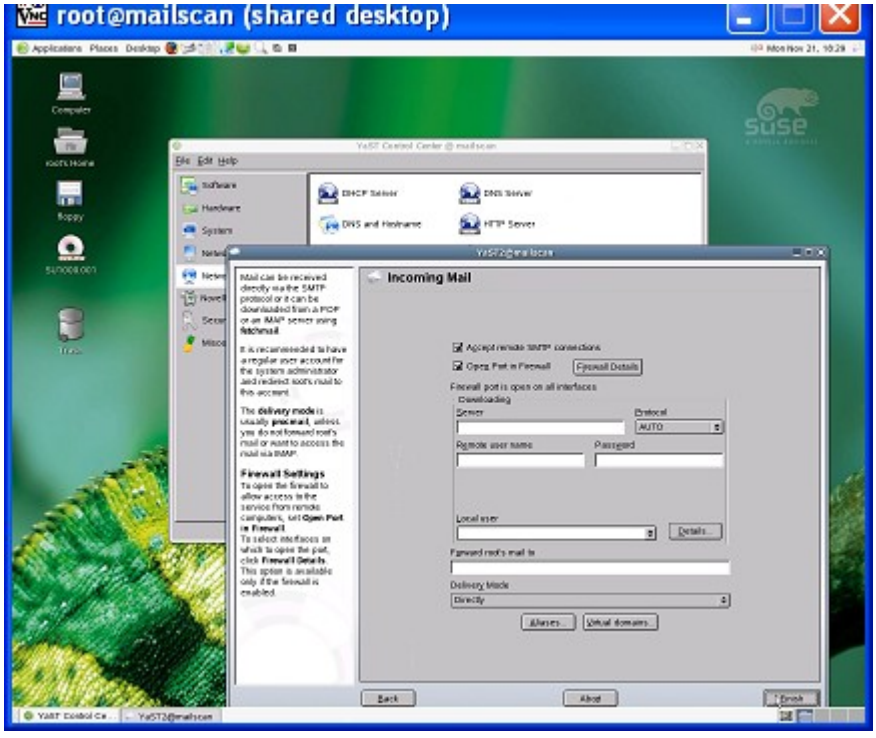
To start you need to configure the postfix MTA.



Start YaST, select **Network Services** from the left pane, then select **Mail Transfer Agent**.

Select the check box **Enable virus scanning (Amavis)** and click **Next**.

At the **Outgoing Mail** screen, just hit **Next**. If you're on dial-up or if you really want to use your ISPs' e-mail relays, you'll need to fill in those details then click **Next**.

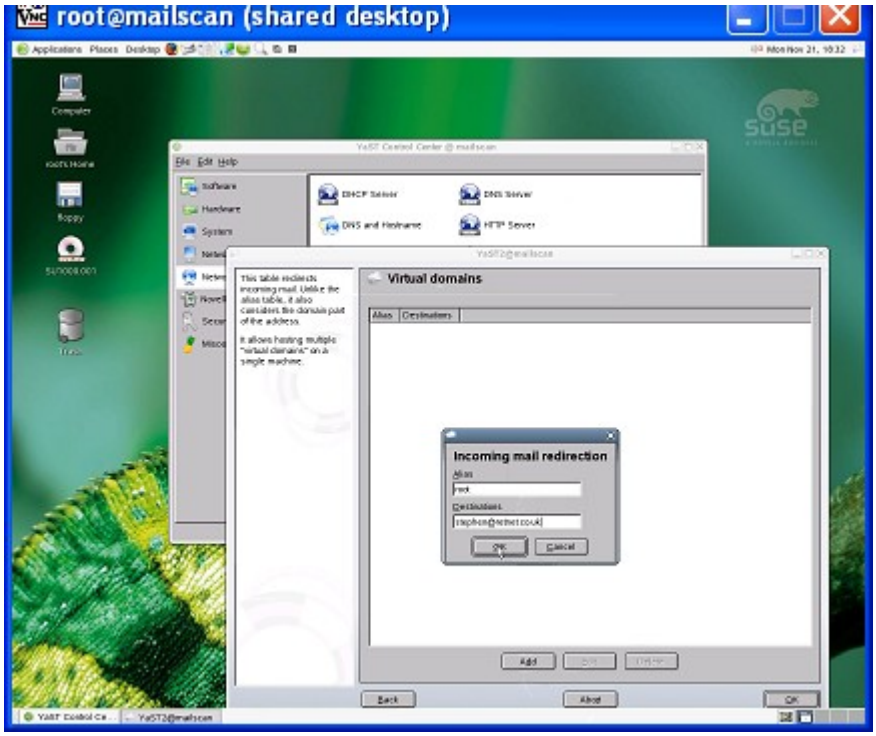


The **Incoming Mail** screen however will need a couple of changes.

Click the check box **Accept remote SMTP connections** else no external mail server will be allowed to send you mail.

Ensure the **Open Port in Firewall** option is selected.
If you've disabled your firewall this option will be greyed out.

If you need to collect your mail from your ISPs' POP3, IMAP etc server, fill in the details.



If you want to receive e-mail sent to the local root user, use the **Virtual domain...** option to forward mail from root to yourself. The **Forward root's mail to** option is only valid for other local users of the gateway itself.

I personally do this so if anything is sending mail to root on that box, such as scheduled cron jobs, or anything/one else then I want to know about it, and so should you else some problems may go unnoticed for a long time.

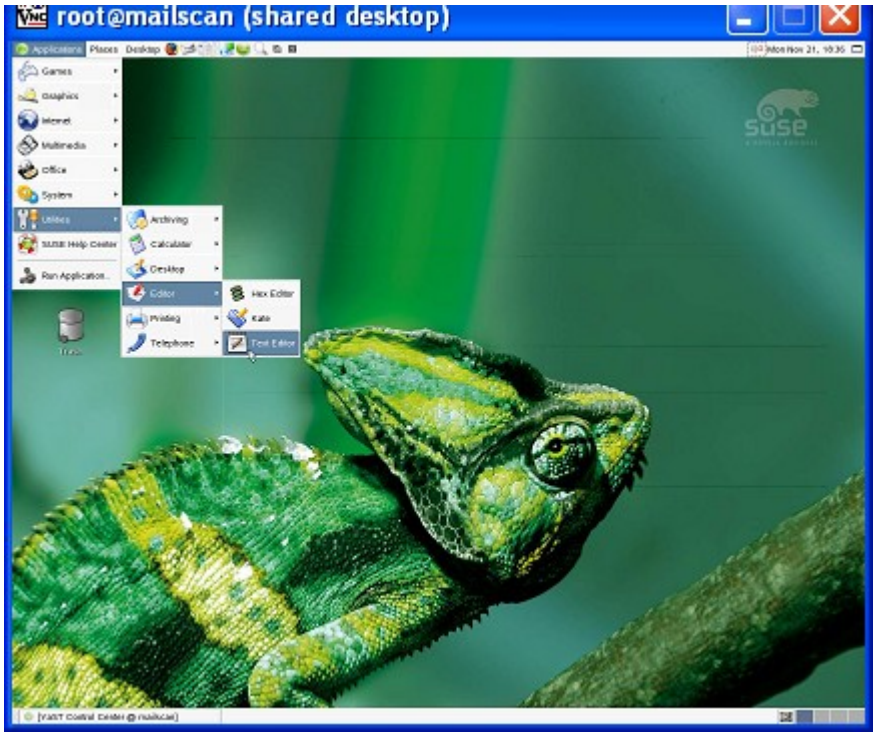
The reason you use a virtual domain and not simply an alias to do this, is that an alias will re-direct a recipient to another 'local' recipient – i.e. another local user on the postfix server itself, whereas virtual domains will redirect mail to another remote domain and/or e-mail address as required. As far as mail redirection is concerned, all your local users are 'remote' to this box.

So in this case, in the Alias field type **root** then in the Destinations box type your own e-mail address, such as **stephen@retnet.co.uk**.

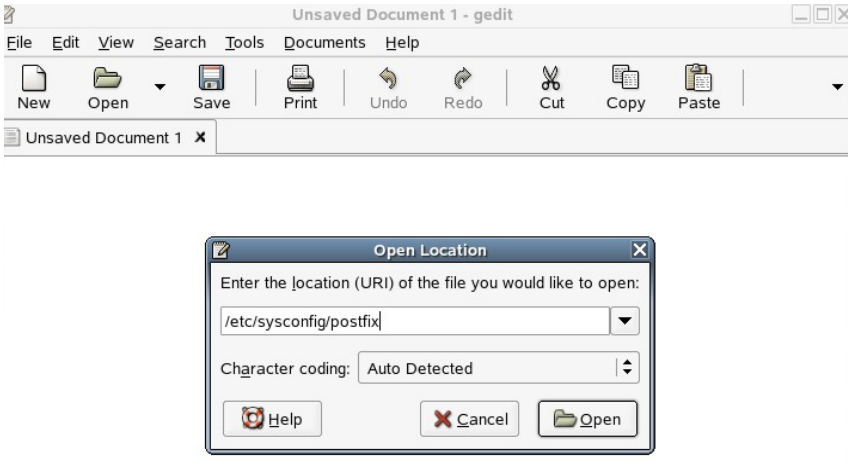
I also use the virtual domains section as I run multiple websites on different domains, so mail can come in addressed to many different accounts. I use the **Virtual domains...** option to redirect incoming mail for those domains to my own e-mail account for addresses such as abuse, webmaster etc.

Click OK twice you've finished here, then the **Finish** button to save the changes.

One last change to postfix at this point which isn't available through YaST is to tell it to relay all e-mail for your domain to your regular mail server.



Start your text editor by selecting the **Applications** menu, then **Utilities, Editor** then **Text Editor**.



Open the file **/etc/sysconfig/postfix** by clicking the **File** menu, selecting **Open Location**, typing in **/etc/sysconfig/postfix** then click **Open**.

Scroll to the end of the file and add the following line:

POSTFIX_ADD_RELAY_DOMAINS=”retnet.co.uk”

where **retnet.co.uk** is your own e-mail domain.

This tells postfix what destination domains it needs to relay. If this change wasn't done, postfix would think all mail to retnet.co.uk was local and try to deliver to a user local on the gateway, which obviously won't work.

Save the file and exit.

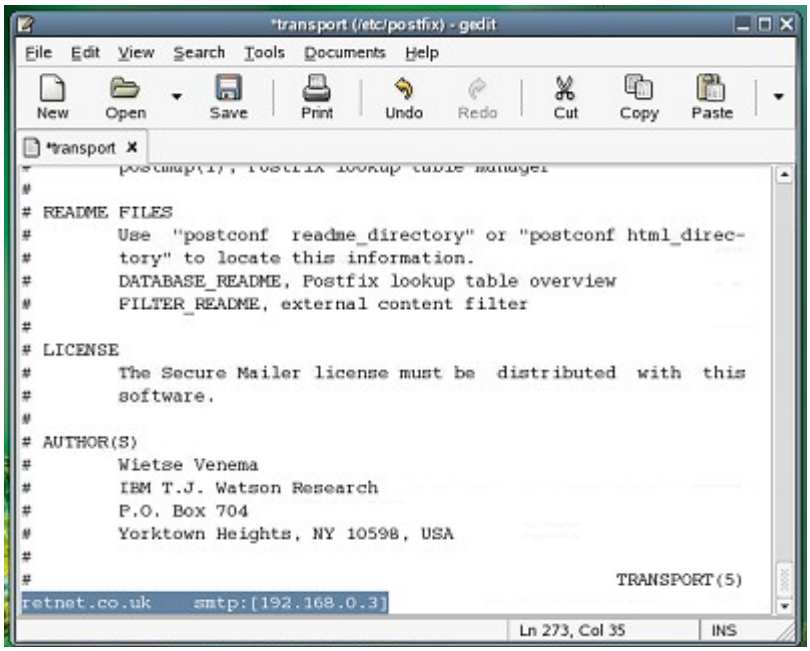
For this change to take affect, open a terminal console by right-clicking anywhere on the desktop and select **Open Terminal**.

In the console, run:

SuSEconfig

This will update the postfix configuration file **/etc/postfix/main.cf** with the setting you just made. The reason you didn't just edit the main.cf file directly is that if you did this you would no longer be able to make changes to postfix via YaST as it would know the file was updated by something other than itself and won't touch it.

Now you need to tell postfix where to relay mail bound for retnet.co.uk. Open the file **/etc/postfix/transport** the same way as before.



Scroll to the end of the file and add the following:

retnet.co.uk smtp:[192.168.0.3]

where retnet.co.uk is your e-mail domain name, and

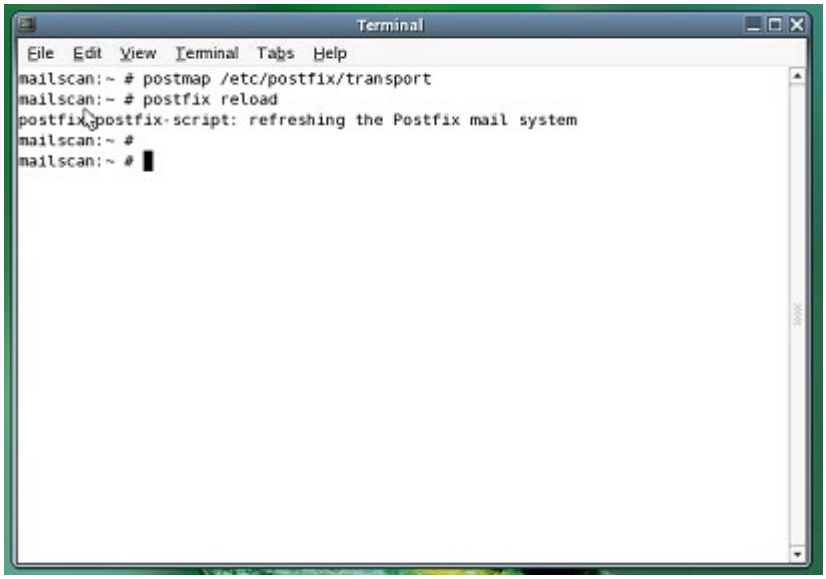
smtp:[192.168.0.3] tells postfix to relay mail bound for retnet.co.uk to 192.168.0.3 which is my real existing mail server (as opposed to a gateway as this is), using the SMTP protocol.

You could also use a host name instead of the IP address. Oh, and the square brackets are important in this case. By default postfix will perform a DNS lookup on the mail exchange (MX) record associated with whatever you type in after the **smtp:** . Using the square brackets stops postfix from performing an

MX lookup on the transport you specify.

Save this file and exit the text editor.

Postfix reads the transport file in a database format, so you need to convert it then reload postfix to pick up the changes.



Open a terminal console by right-clicking anywhere on the desktop and select **Open Terminal**.

In the console, run:

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

The **postmap** command will convert the transport file into transport.db which postfix will read, and the **postfix reload** command tells postfix to re-load all configuration files which will pick up the change.

Note at this stage there are quite a few anti-spam settings that haven't been applied along with other defaults which you may want to change such as the default message size limit which is 10 Mb. These will be covered after the initial configuration is confirmed as working correctly.

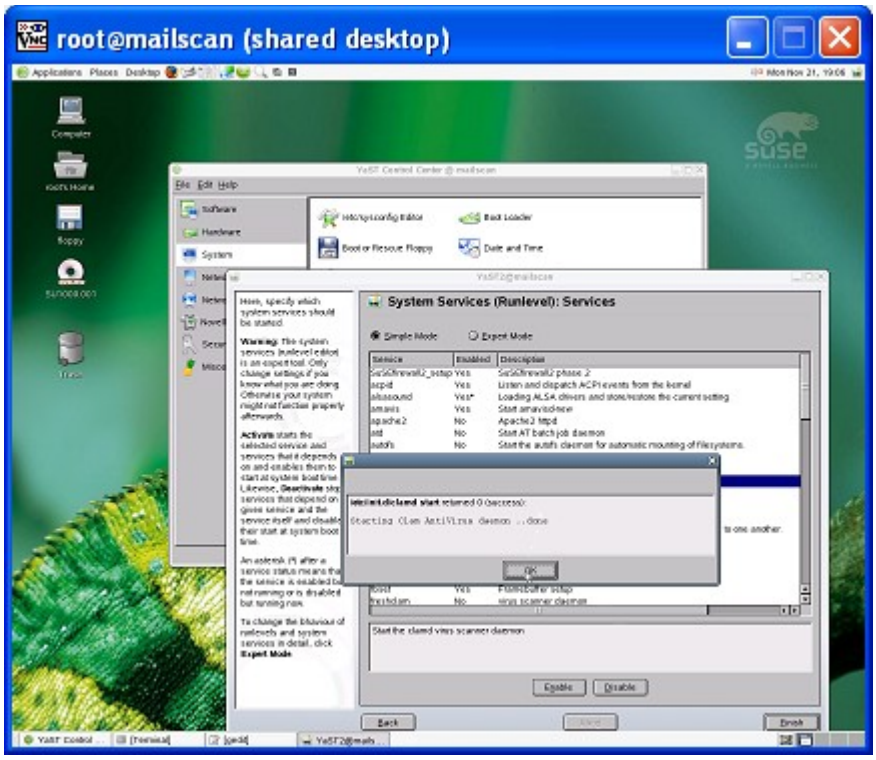
Setting up ClamAV

"Clam AntiVirus is a GPL anti-virus toolkit for UNIX. The main purpose of this software is the integration with mail servers (attachment scanning). The package provides a flexible and scalable multi-threaded daemon, a command line scanner, and a tool for automatic updating via Internet. The programs are based on a shared library distributed with the Clam AntiVirus package, which you can use with your own software. Most importantly, the virus database is kept up to date ."

Quoted from the Clam Antivirus website at http://www.ClamAV.net

By default amavisd calls ClamAV to scan e-mails that come in. ClamAV can be contacted either via a network port or a local socket (a special file). By default it uses a TCP socket which amavisd is already set up for.

ClamAV at this point isn't configured to run as a service, so fire up **YaST**, select **System** then **System Services (RunLevel)**



Scroll down to the **clamd** service and click **Enable**. Click **OK** to the message pop-up.

To make sure the virus signature database gets updated, you will also need to enable the **freshclam** service, listed further down.

After enabling both services click **Finish** then **Yes** to save changes and close **YaST**.

By default freshclam will check for updates every 2 hours via HTTP, which can be changed by editing the file **/etc/freshclam.conf** accordingly and re-starting the service by running **/etc/init.d/freshclam restart**

If you need to go via a proxy, edit **/etc/freshclam.conf**, scroll down to the **# Proxy Settings** section and edit the settings as required. An example might look like:

```
# Proxy settings
# Default: disabled
HTTPProxyServer proxy.com
HTTPProxyPort 8080
HTTPProxyUsername myusername
HTTPProxyPassword mypass
```

You can also check to ensure the updates are working by running **cat /var/log/mail** in a terminal and looking for freshclam notices.

Setting up razor2 agents

"Vipul's Razor is a distributed, collaborative, spam detection and filtering network. Through user contribution, Razor establishes a distributed and constantly updating catalogue of spam in propagation that is consulted by email clients to filter out known spam. Detection is done with statistical and randomized signatures that efficiently spot mutating spam content. User input is validated through reputation assignments based on consensus on report and revoke assertions which in turn is used for computing confidence values associated with individual signatures."

Quoted from Vipul's Razor web site at
http://razor.sourceforge.net

The razor client is automatically picked up by SpamAssassin as long as it's installed.

It send requests to public Razor servers on the Internet asking if an e-mail is known spam, by sending a special checksum of the e-mail it creates which guarantees it is unique to that e-mail (no e-mails are actually sent so your data is safe).

It sends these requests on outbound TCP port 2703, so your Internet firewall will need to let this out, and associated replies from your mail gateway.

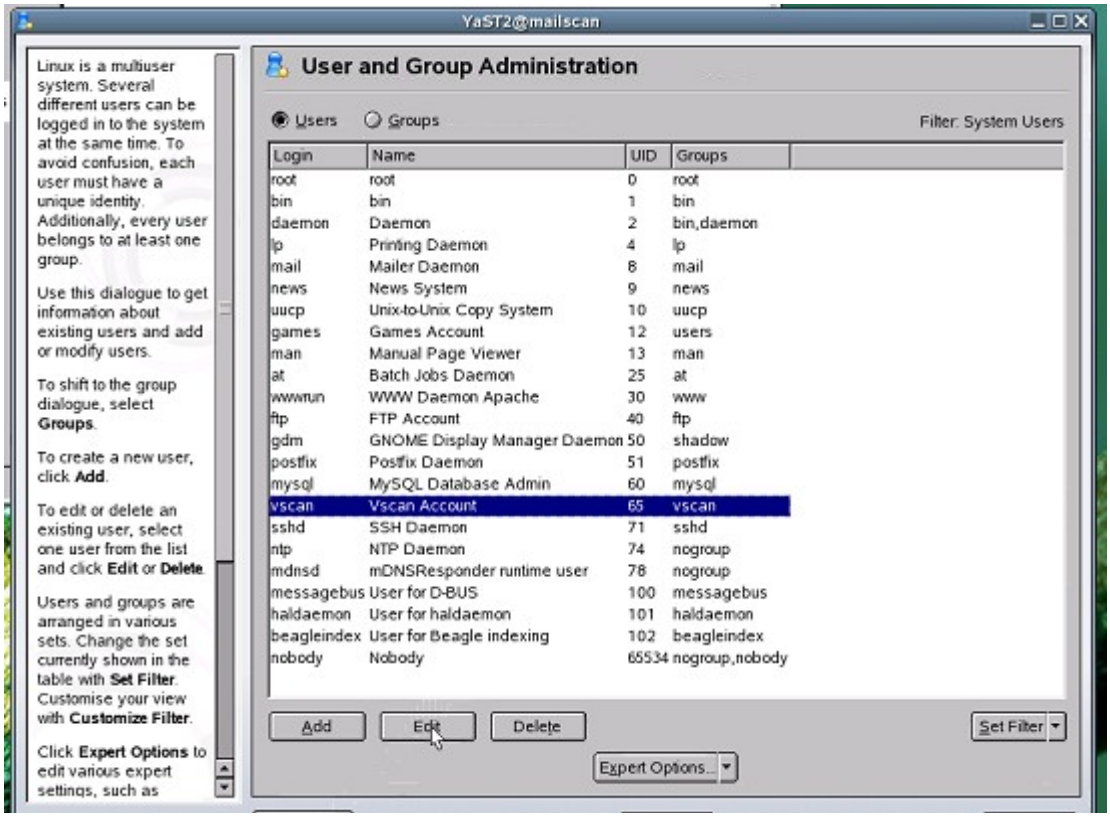
To configure razor2, you need to create some defaults and register it with the razor network using the **razor-client** & **razor-admin** utilities.

You need to run the **razor-admin** utility as the **vscan** user so it's configuration file gets created in the right place with the correct permissions, but because the **vscan** user is a special system user it cannot normally run interactively on the system, but this can be changed by assigning a '**Login shell**' to the user.

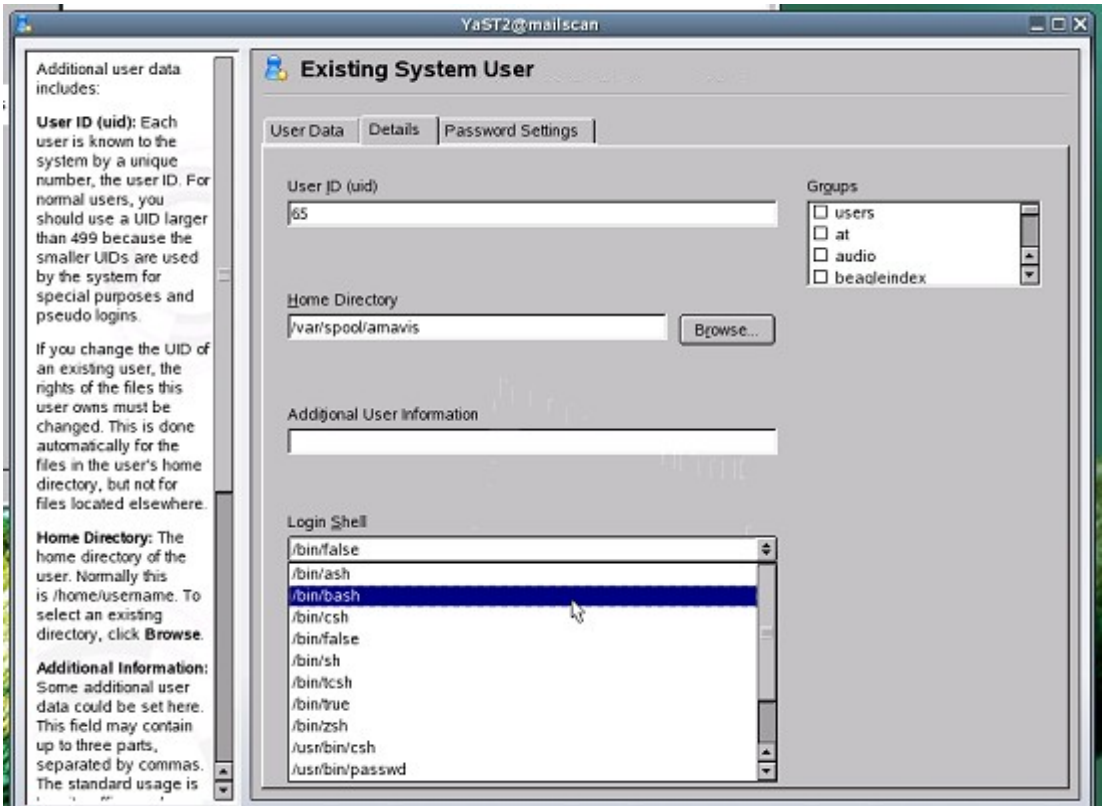
The reason that razor is run as this user is that it's the default for amavisd-new. From the diagram at the very start of this document you can see that razor is called by SA, which in turn is called by amavisd-new. All of these systems run as the vscan user so that they can easily pass the message between each other without problems.

After testing, you will set this back to help keep your system a little more secure.

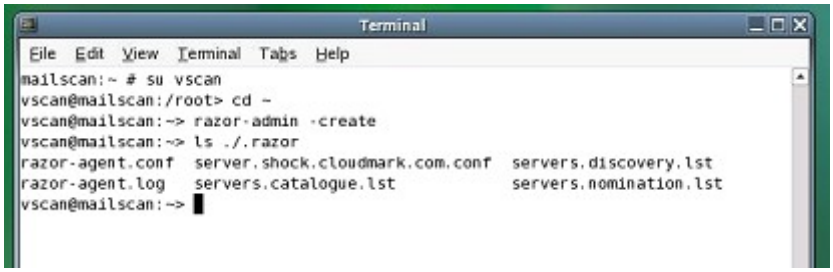
Open **YaST,** select **Security and Users** then **User Management**.



Select the **Set Filter** pull down menu and choose the **System Users** filter option. Now scroll down, select the **vscan** user and click the **Edit** button.



Select the **Details** tab and in the **Login shell** menu, select the **/bin/bash** option, then click **Accept** then **Finish**.



Now change to the vscan user and change into it's home directory by running

```
su vscan
cd ~
```

Create a default configuration for the vscan user by running

```
razor-admin -create
```

*There is a space between **admin** and **-create***

You can see the directory and configuration files it created by listing files in the **.razor** directory. The period at the start of the directory makes it hidden so you won't see it by default.

Now register razor2 with the razor network.

You should first let razor discover publicly available servers, then register it. You can either let razor choose a user name and password for you, choose the user name but let razor generate a password or you can specify the lot.

As the **vscan** user, run

```
razor-admin -discover
razor-admin -register
```

If you receive a 202 error while trying to register, try it a few more times and it should work. It's just a sign of busy servers.

You can specify a username and password if you like. Just add

-user=foo **-pass**=foopass

after the **-register** switch.

razor-admin will tell you if it was successful and where the identity information is stored.

Exit the vscan shell by running

```
exit
```

Normally after installing and configuring razor2 agents you would also re-start amavisd-new to let SpamAssassin (yep, the dependencies are nested...) pick up the changes but in this case there are other changes you are going to make before you need to do this.

Install and setting up DCC

"The DCC or Distributed Checksum Clearinghouse is an anti-spam content filter. As of mid-2004, it involves millions of users, tens of thousands of clients and more than 250 servers collecting and counting checksums related to more than 150 million mail messages on week days. The counts can be used by SMTP servers and mail user agents to detect and reject or filter spam or unsolicited bulk mail. DCC servers exchange or "flood" common checksums. The checksums include values that are constant across common variations in bulk messages, including "personalizations."

The idea of the DCC is that if mail recipients could compare the mail they receive, they could recognize unsolicited bulk mail. A DCC server totals reports of checksums of messages from clients and answers queries about the total counts for checksums of mail messages. A DCC client reports the checksums for a mail message to a server and is told the total number of recipients of mail with each checksum."

Quoted from the Distributed Checksum Clearinghouse website
http://www.rhyolite.com/anti-spam/dcc/

First a blurb... it's worth it.

The DCC software is actually a collection of both server and client based programs. You will be installing only one of the client programs as this installation isn't intended on reaching the sort of throughput that dictates you install your own DCC server (around 100,000 e-mails per day).

The DCC client works on a different method to razor in that it doesn't actually detect spam, but does a great job in detecting mass e-mails. The idea here is that spam needs to reach a massive audience to gain any tangible returns. It sends checksums of incoming e-mail to DCC servers and checks how many identical e-mails have been seen by the servers. Based on this information it can trigger DCC to mark it as a mass e-mail.

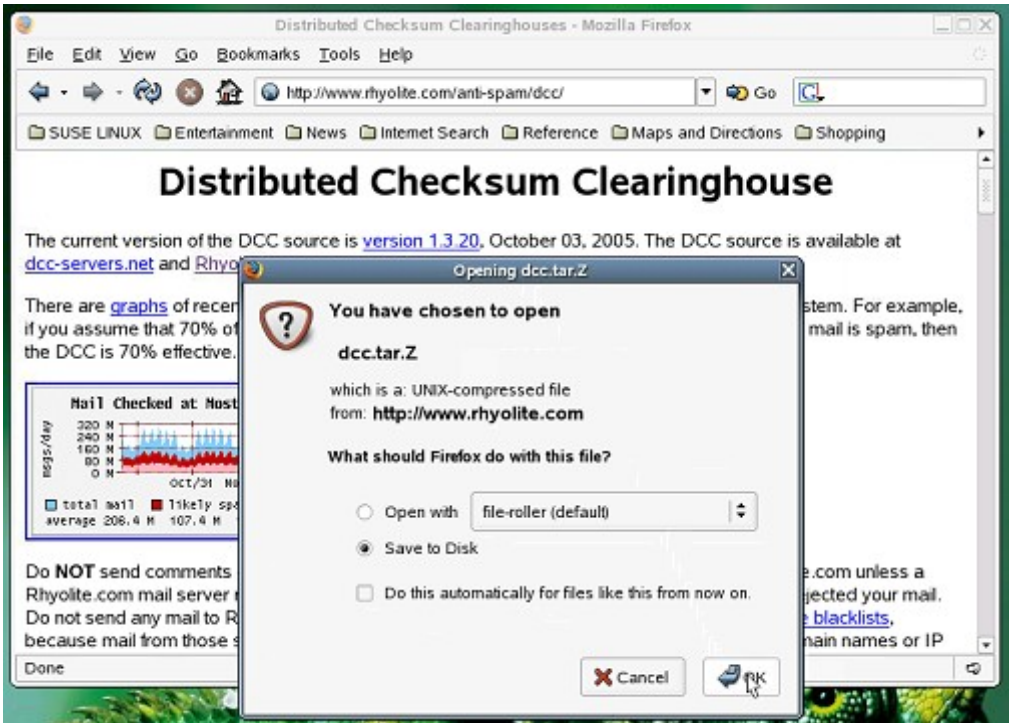
Unfortunately this also means some heavy mailing lists, such as SuSE and others may trigger DCC rules and add a score that will lean their total score towards spam. On the up side, the scores that are added are small enough that almost all of these will still pass through as long as they are properly addressed, have come from correctly configured e-mail servers and conform to correct Internet e-mail standards.

To overcome some clean e-mail that may be triggered, DCC also comes with a good out-of-the-box list of addresses that are known to send out legitimate mass e-mails and can be configured (more on this in the later Tweaks and Tightening section).

DCC will be automatically detected and used as a plug-in by SpamAssassin, so all you have to do is install it and the system will do the rest for you.

Before you start, you will need to allow outgoing packets on **UDP port 6277** and inbound replies from **UDP port 6277** on your firewall.

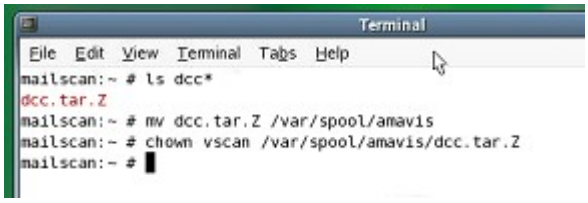
You will need to read your router/firewall manual for help if you don't know how to do this. The built in SuSE firewall will work with DCC without modification.



You will need to download the software from the Distributed Checksum Clearinghouse homepage at http://www.rhyolite.com/anti-spam/dcc/

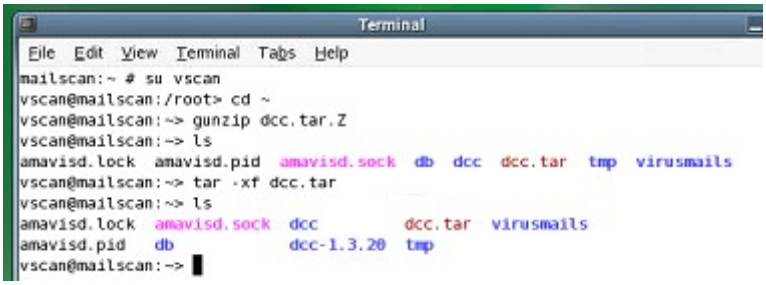
click the hyperlink on the main page with the version (currently 1.3.20), select the **Save to Disk** option then click **OK**.

Save the file even if it gives you the option of opening it, as opening it will fail. I'm not going to get into details here, but it's due to the type of archive it is along with what appears to be a security concern with opening these archive. Rest assured this one is safe.



Because this program needs to be installed as the user running it (vscan), at this stage it's easiest to copy the archive and change ownership to the vscan user and it's home directory.

```
mv ./dcc.tar.Z /var/spool/amavis
chown vscan /var/spool/amavis/dcc.tar.Z
```

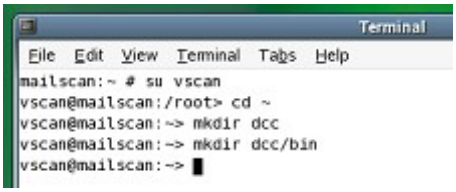


Now to extract the archive. In a terminal run

```
su vscan
cd ~
gunzip dcc.tar.Z
tar -xf dcc.tar
```

You can now see a directory called dcc-1.3.20 (at the time of writing this. The version may differ slightly).

Before installing dcc for the vscan user to use, you need to create a couple of directories inside the vscan home directory for placement of the dcc programs and configuration files.

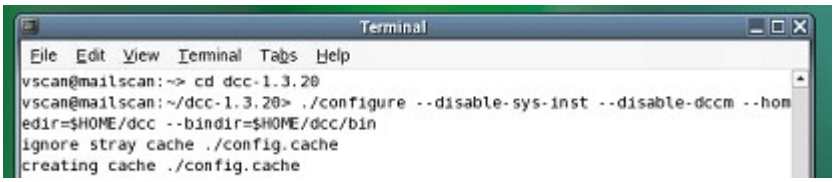


In the terminal window, still as the vscan user and make the directories required by running

```
mkdir dcc
mkdir dcc/bin
```

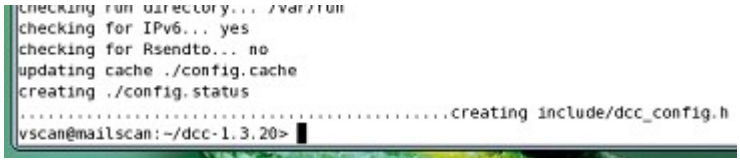
The dcc configuration files will end up in **/var/spool/amavis/dcc** and the client program will end up in **/var/spool/amavis/dcc/bin**

Now it's time to compile and install the software.



Still as the vscan user, change to the dcc-1.3.20 directory then run the configure script, by running

```
cd dcc-1.3.20
./configure --disable-sys-inst --disable-dccm --homedir=$HOME/dcc --bindir=$HOME/dcc/bin
```



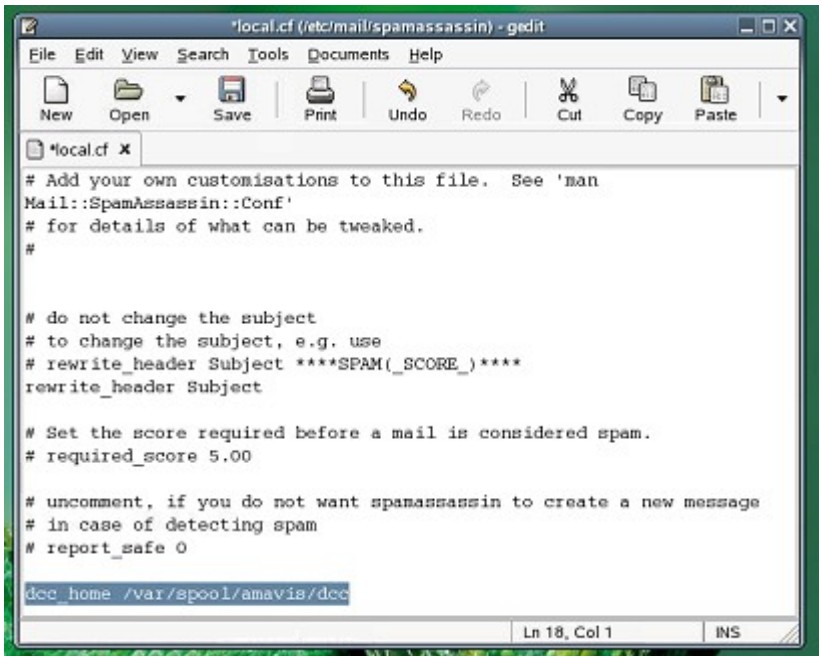
As long as you don't see any error messages, all went well.

Now install dcc by running

make install

Again as long as no errors appear, it worked just fine.

Because SpamAssassin uses DCC as a plugin, SA will need to know where to look for DCC so you will need to edit the SpamAssassin configuration file.



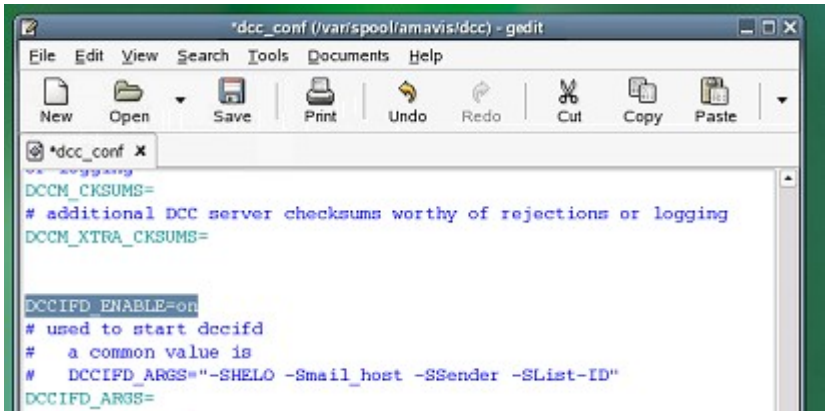
Start Gedit, select **File – Open Location** and type in **/etc/mail/spamassassin/local.cf**

Add to the end of the file the following text:

dcc_home /var/spool/amavis/dcc

Save the file.

Lastly you need to configure DCC to start during system boot.



In Gedit open the DCC configuration file **/var/spool/amavis/dcc/dcc_conf**

Scroll down the file and change the option

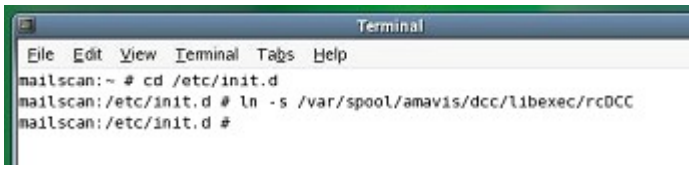
DCCIFD_ENABLE=off

to read

DCCIFD_ENABLE=on

Save the file and exit.

Create a link to the startup file for the dccifd service so that you can control it using the YaST System Services (RunLevel) editor, then Enable the service.

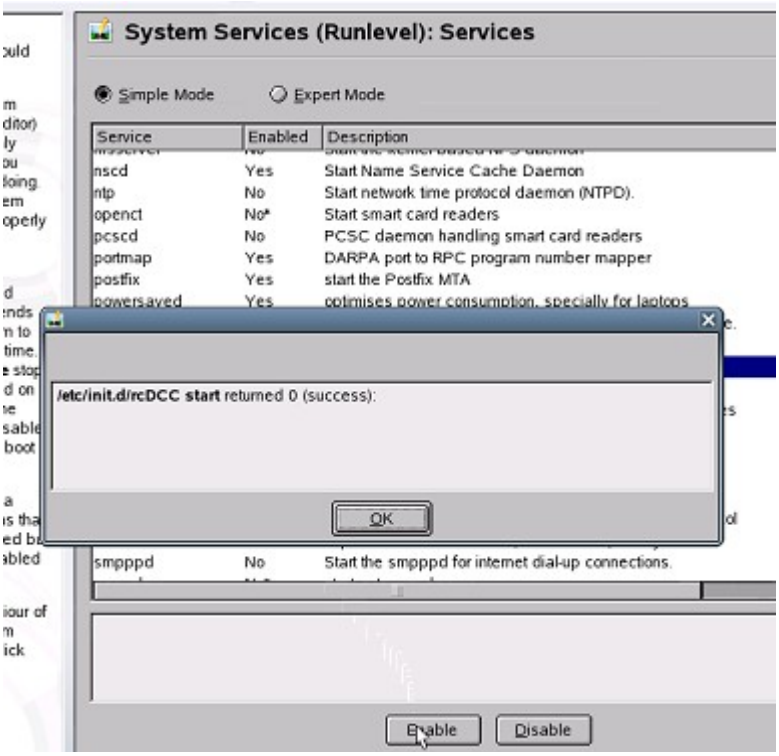


Open a terminal window as root, and run

cd /etc/init.d
ln -s /var/spool/amavis/dcc/libexec/rcDCC

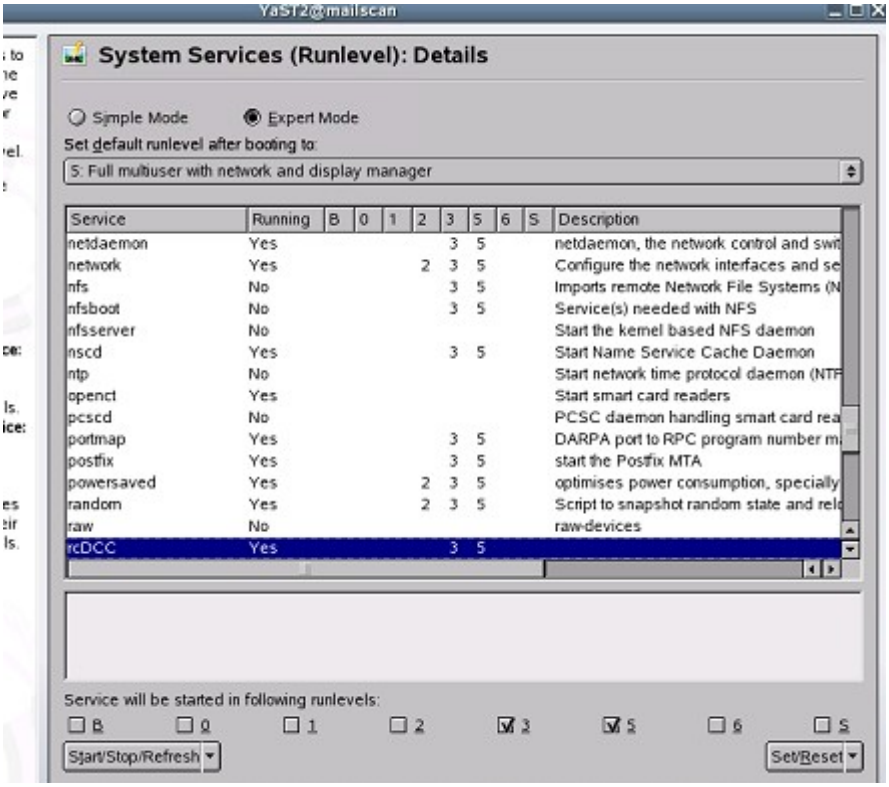
The **ln** command will create a link of the same name to the startup file rcDCC, in the current directory being /etc/init.d/

Now set the service to start automatically



Do this by starting **YaST**, going to **System, System Services (RunLevel)**, scroll down to **rcDCC** and click **Enable**, then the **OK** button.

You can see that although the status returned was **success**, in the **Enabled** column it still lists **rcDCC** as **No** but now with a star ***** against it. This is because the rcDCC startup script doesn't have information to tell YaST at what point during the system boot up it should start the service, which you will now do manually.



Click the **Expert Mode** radio button at the top of the screen, scroll down and highlight the **rcDCC**

service, then select the **3** and **5** RunLevel check boxes.

Click **Finish** and **Yes** to save the changes. If you re-run the **System Services(RunLevel)** editor again you will now see a **Yes** in the **Enabled** column.

Setting up amavisd-new

*"**amavisd-new** is a high-performance interface between mailer (MTA) and content checkers: virus scanners, and/or SpamAssassin. It is written in Perl for maintainability, without paying a significant price for speed. It talks to MTA via (E)SMTP or LMTP, or by using helper programs. Best with Postfix, fine with dual-sendmail setup and Exim v4, works with sendmail/milter, or with any MTA as a SMTP relay."*

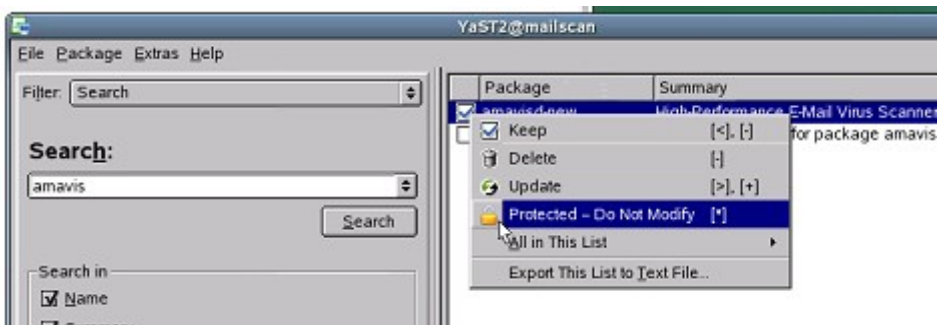
Quoted from the amavisd-new website at http://www.ijs.si/software/amavisd

This program is the 'glue' between postfix and the anti-spam/virus scanners but it also does a few other things such as attachment blocking and spam and virus detection notifications via e-mail.

Because SuSE 10.0 installs a much newer version than Maia Mailguard is expecting you will have to tell YaST not to update this package any more so your changed executable doesn't get overwritten.

You will be using versions of the executable and configuration file from the Maia Mailgaurd website, so at this point all you will do is set the YaST package amavisd-new to protected, thus stopping any SuSE updates from taking effect.

Open YaST, select **Software Management**



In the **Search** field type in amavis then click **Search**. Right-click the amavisd-new package and select the option **Protected – Do Not Modify**.

Due to a flaw(?) in YaST this change won't stick unless you install or deinstall something else from the same media as where the application you want to protect was originally installed from, so for this build I searched for and selected to install a package called **Subversion**, clicked on the **Version** tab then selected the 'other' one in the list (they're both identically labeled so you can't tell which version is on the DVD vs online). Subversion is a small version control program you can use to download the latest source code from online repositories such as Maia Mailguard.

Now click **Accept**. You will notice only the subversion package gets installed.

If you want to ensure the amavisd-new package is in fact protected now, click the **Install More** button and search for the amavisd-new package again and you will see it now has a protected icon. If it doesn't, try installing something again but choosing the alternate location.

If your server has less than 384 Mb RAM, you may want to also reduce the number of amavisd-new processes that run concurrently to save on system resources. To do this, reduce the **$max_servers** setting from it's default of 2 down to 1 in the amavisd.conf file. If you do this it is assumed you're not processing anywhere near 30,000 e-mails per day, given you're setting this system up on such a low end box.

Setting up MySQL

The next task is to setup MySQL to store settings, statistics and e-mail for Maia Mailguard.

You will need to configure MySQL to start automatically during system boot, so run **YaST** and from the **Services** menu in the left pane click on **System Services (RunLevel)** in the main window.

(I won't include a picture this time as you should know how to get there now.)

Scroll down and select the **mysql** service then click **Enable**.

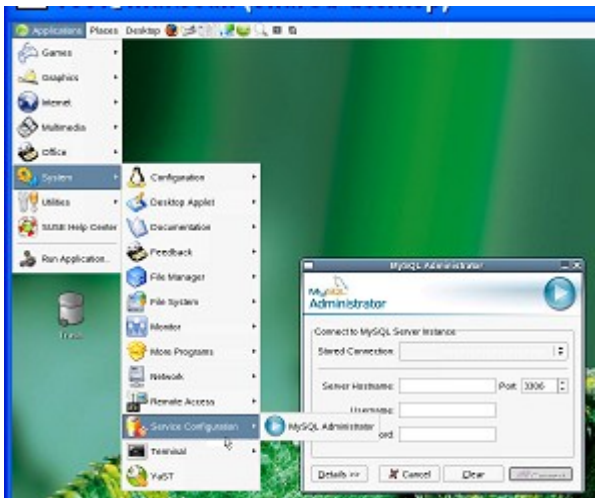
Being the first time you start mysql, you'll see a special message pop-up with instructions on how to change the mysql root password.

Click **OK** to the message, **Finish** then **Yes** to save changes and close YaST.

I strongly suggest you follow the instructions to change the mysql root password, by opening a new terminal console as root and run the commands:

```
mysqladmin -u root password newpassword
mysqladmin -u root -h mailscan.retnet.co.uk password newpassword
```

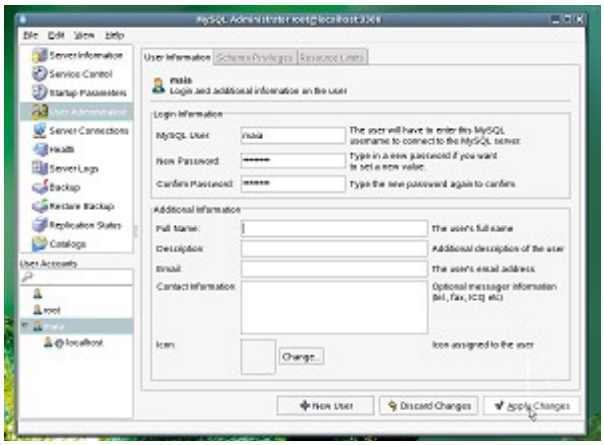
where you type in everything in **bold** exactly as it is above, and your own details for the other parts.



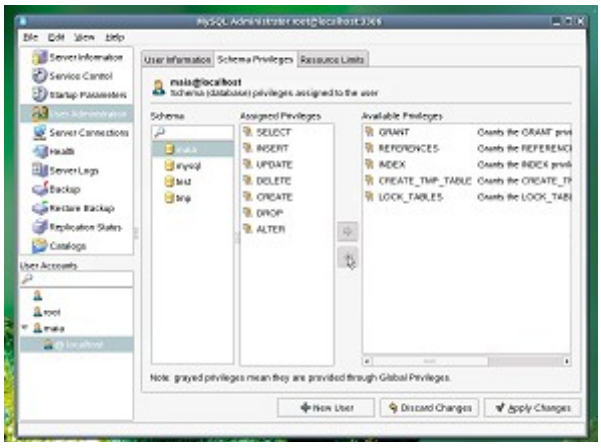
Now you should create a database (called a Schema) and user for Maia Mailguard to use. Fire up **mysql-administrator** by selecting the **Applications** menu, **System**, **Service Configuration**, then **MySQL Administrator**.

Log in as root.

To create a new database, click on the **Catalogs** link then right-click under the mysql and test schema's, and choose **Create Schema**. Type in a schema name.. I'll be calling mine maia for this example.



Create the maia user by selecting **User Administration**, right-click in an empty area underneath the other users and select **New User**. Type in the name and password you want to use for the database connection.



Now click on **@localhost** under new_user then select the **Schema Privileges** tab.

Highlight the maia database then in the **Available Privileges** list, select the following privileges which you add by clicking the **Left arrow** button: **Select, Insert, Update, Delete, Create, Drop, Alter**.

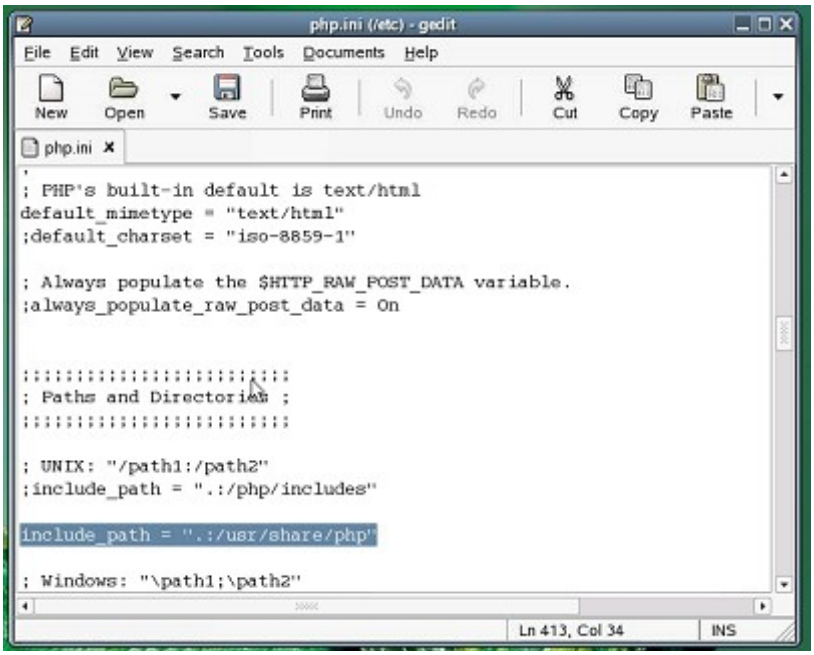
Click the **Apply Changes** button to save.

If you get an error about duplicate users, cancel the changes, delete the maia user and do it again, this time without pressing Apply Changes before clicking the @localhost link.

You're all done here so close MySQL Administrator.

Configure PHP / Apache2

Maia uses the PHP language extensively but a path setting in the /etc/php.ini file needs to be modified slightly for it to work.



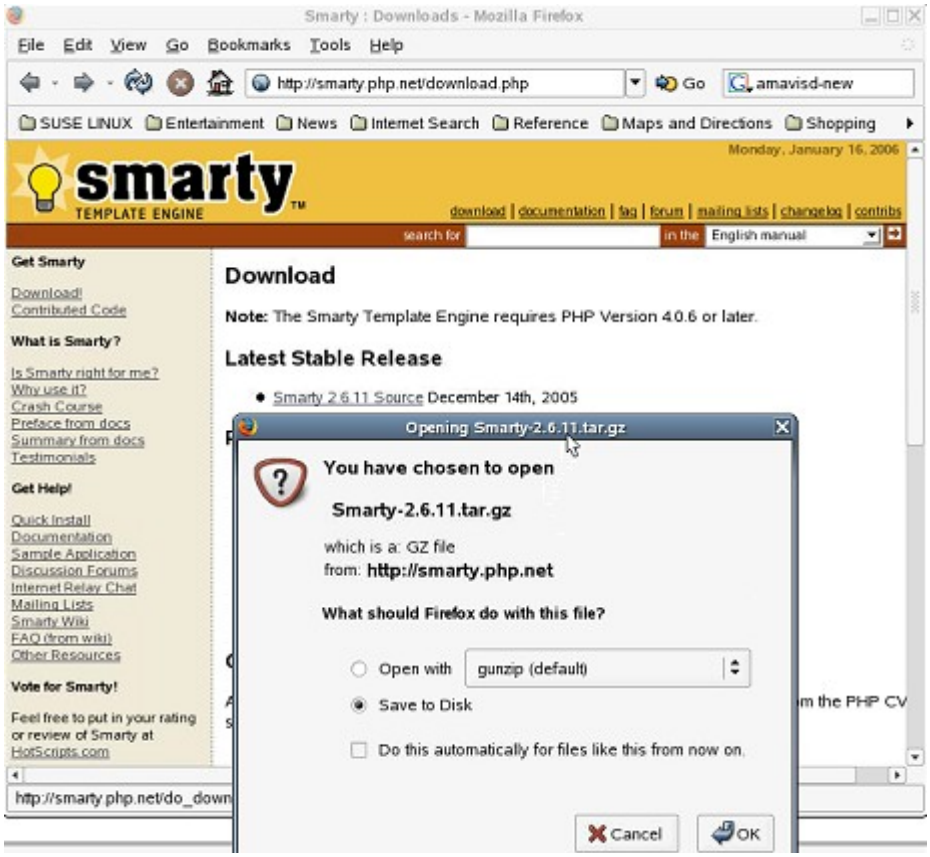
Using Gedit open **/etc/php.ini**

Scroll down and change the **include_path** setting to read:

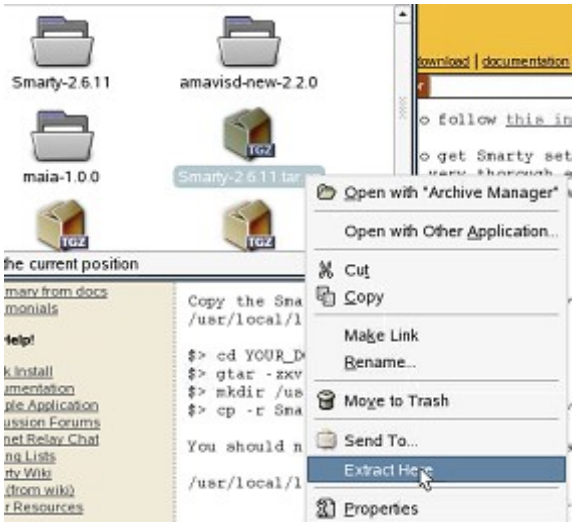
include_path = ".:/usr/share/php";

By default it doesn't include **.** (dot) path meaning the current directory and Maia Mailguard will not work without this.

Maia Mailgaurd also uses the Smarty template system for PHP, allowing easier development of themes, so now you need to download and install Smarty.



In a web browser, goto http://smarty.php.net, click the download link and download the latest stable release.



Extract the downloaded file by opening a new file browser window from the **Places** desktop menu, right-click the Smarty-2.6.11.tar.gz file and select the **Extract Here** option. This will create a **Smarty-2.6.11** or similar directory in /root

Now create the required **Smarty** directory in the php shared root directory by running in a terminal window as root:

```
mkdir /usr/share/php/Smarty
```

Lastly copy the Smarty library files into the directory you just created by running:

```
cd /root/Smarty-2.6.11
cp -r ./libs/* /usr/share/php/Smarty
```

Also the web server will need to be running (fairly obvious), so to configure apache2 to start automatically during system boot, run **YaST**, select **Services** in the left pane then **System Services (RunLevel)** in the main window.

(Again, I won't include a picture this time as you should know how to get there now.)

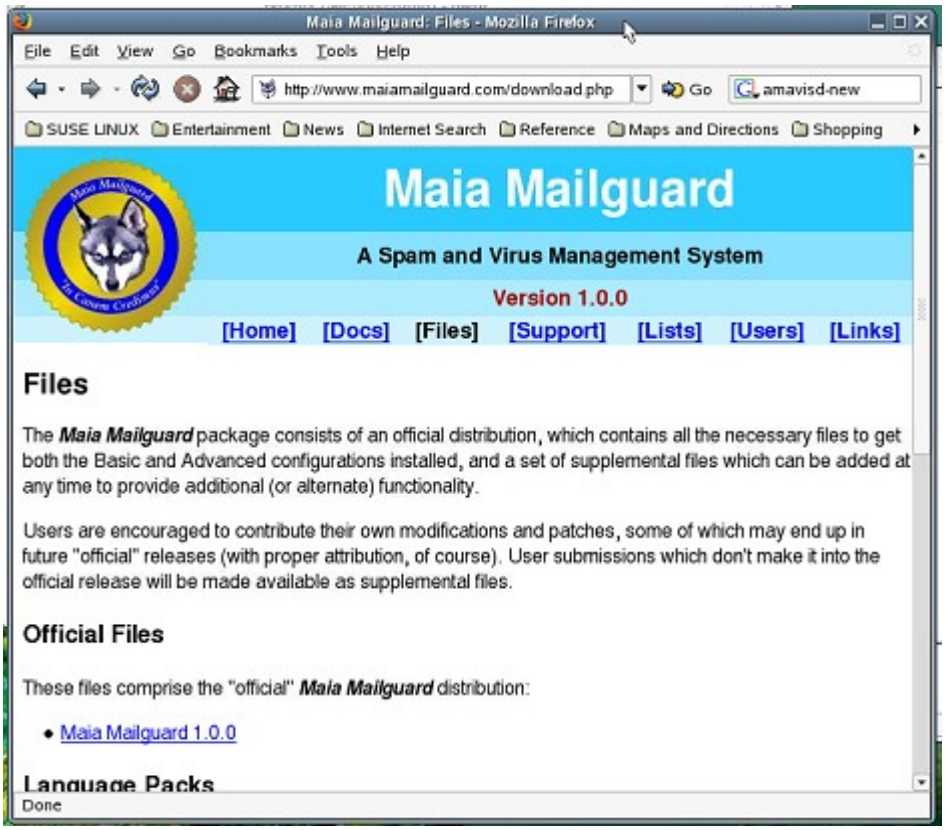
Select the **apache2** service then click **Enable** and click OK and Finish.

Maia Mailguard v1.0.0 installation

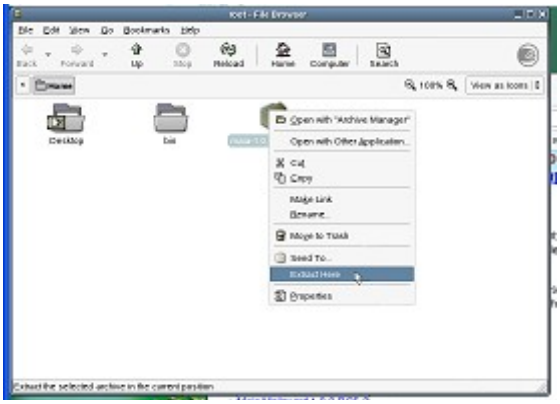
*"**Maia Mailguard** is a web-based interface and management system for the popular amavisd-new e-mail scanner and SpamAssassin. Written in Perl and PHP, **Maia Mailguard** gives end-users control over how their mail is processed by virus scanners and spam filters, while giving mail administrators the power to configure site-wide defaults and limits."*
Quoted from the Maia Mailguard web site at
http://www.maiaMailguard.com

Now it's time to configure the final part of this solution. The web front end management to your e-mail scanning gateway.

Download Maia Mailguard v1.0.0



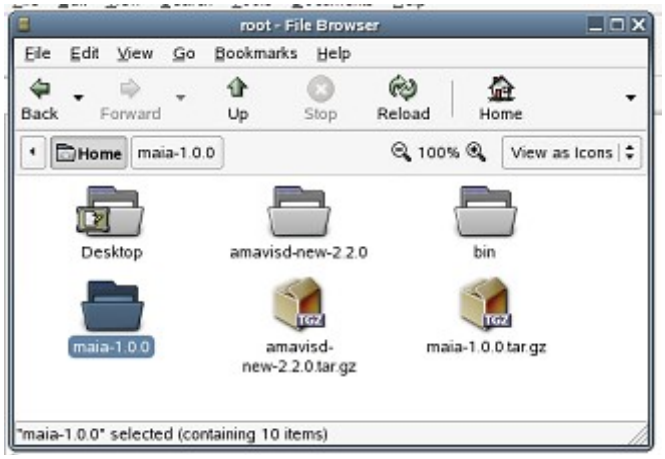
Head over to http://www.maiaMailguard.com/download.php and download version 1.0.0



Open the downloaded archive by opening **root's home** folder (the default), right-click the downloaded

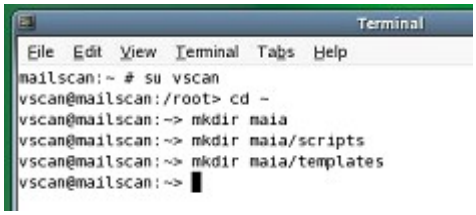
file and select the **Extract Here** option.

This will create a **maia-1.0.0** directory in **/root** with all the extracted files.



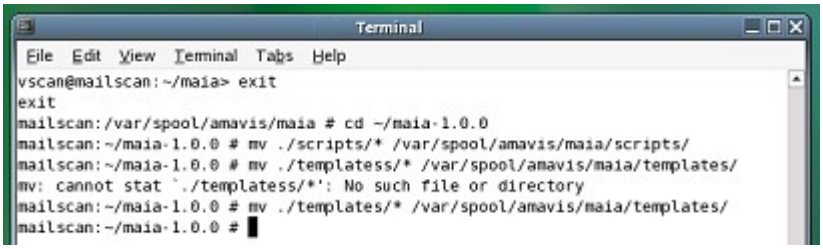
Configure scripts and templates

Maia Mailguard depends on a few templates and scripts to run outside of the web server's path, so now you need to setup the directory structure and configure them for your server.



Open a terminal window, change to the **vscan** home directory and create the maia directories as follows:

```
su vscan
cd ~
mkdir maia
mkdir maia/scripts
mkdir maia/templates
```

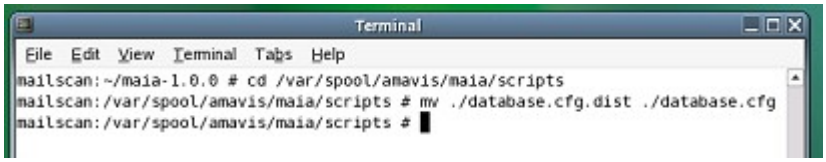


Now change back to root and copy the scripts and templates into their new home, by running

```
exit
cd ~/maia-1.0.0
cp ./scripts/* /var/spool/amavis/maia/scripts/
cp ./templates/* /var/spool/amavis/maia/templates/
```

Because you'll be starting with the default maia configuration, now rename the database template file to a

live version.



Rename the file by running

```
cd /var/spool/amavis/maia/scripts
mv database.cfg.dist database.cfg
```

For these files you need to modify their ownership and permissions, which as usual is easier to do at the command line.

Back in the terminal window, run

```
cd /var/spool/
chown -R vscan:vscan ./amavis/maia
chown -R :www ./amavis/maia/templates
chmod 640 ./amavis/maia/templates/*.tpl
chmod 750 ./amavis/maia/scripts/*.pl
chmod 640 ./amavis/maia/scripts/database.cfg
chmod 711 ./amavis
```

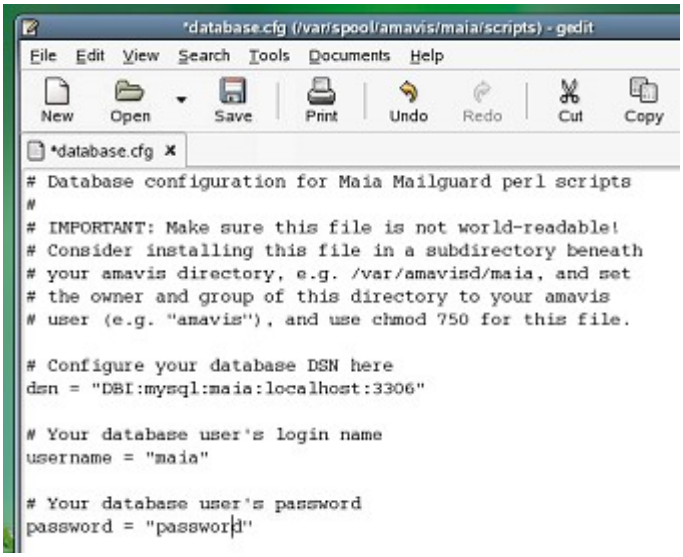
These commands are explained as follows:

chown changes the user and group ownership of the directory and -R tells chown to do it recursively (i.e. all files & folders beneath that level will also be affected).

You need to change the group membership of the templates to the www group so that the web server can read them. If you don't, you will never be able to login and will receive various errors.

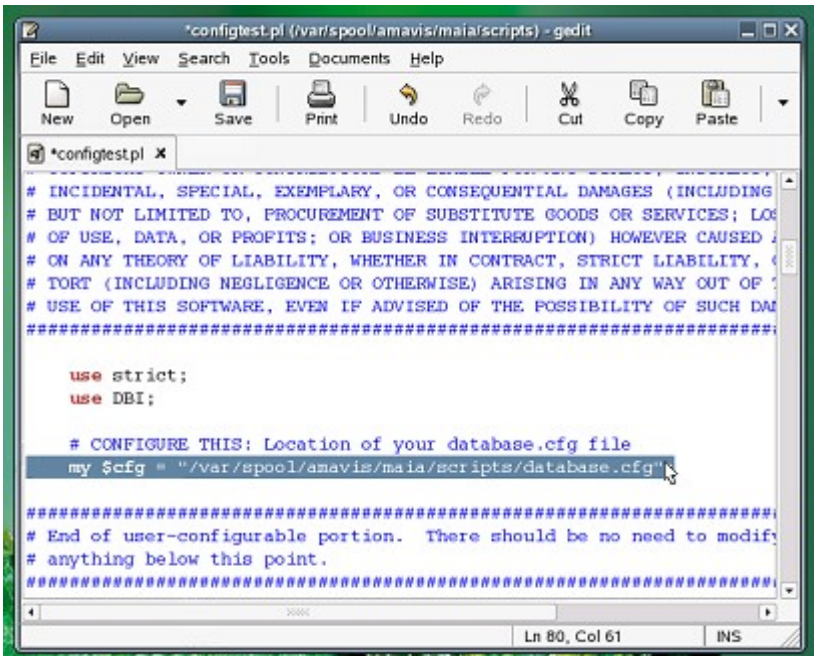
chmod modifies the file rights of the files or directories specified. The numbers are shorthand for what rights the user/group/other has to that file or directory. 711 is a set of permissions added to the amavis directory which will allow the apache web server to read the template files.

Now you need to edit the **database.cfg** file to tell Maia where your database is, along with login credentials.



Using Gedit open **/var/spool/amavis/maia/scripts/database.cfg** and change the database username and password to what you setup earlier in mysql. I suggest for a production system you don't use maia or amavis as the username as these seem a little obvious, even though I'm using that for this guide.

Finally for this configuration stage, you'll need to edit the perl scripts in **maia/scripts/** to tell them where the **database.cfg** file is located.



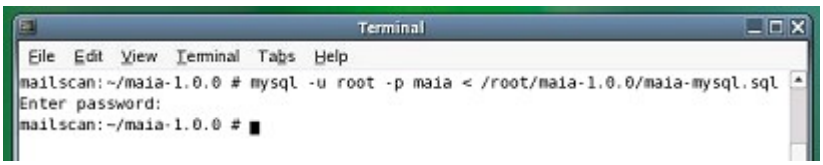
Use Gedit, change the following maia perl scripts as outlined below

File to change	*Line to change from*	*Line to change to*
configtest.pl	my $cfg = “/var/amavisd/maia/scripts/database.cfg”;	my $cfg = “/var/spool/amavis/maia/scripts/database.cfg”;
expire-quarantine-cache.pl	my $cfg = “/var/amavisd/maia/scripts/database.cfg”;	my $cfg = “/var/spool/amavis/maia/scripts/database.cfg”;
load-sa-rules.pl	my $system_rules_dir = “/usr/local/share/spamassassin”;	my $system_rules_dir = “/usr/share/spamassassin”;

File to change	*Line to change from*	*Line to change to*
	my $user_rules_dir = “/var/amavisd/.spamassassin”;	my $user_rules_dir = “/var/spool/amavis/.spamassassin”;
	my $cfg = “/var/amavisd/maia/scripts/database.cfg”;	my $cfg = “/var/spool/amavis/maia/scripts/database.cfg”;
process-quarantine-sub.pl	my $cfg = “/var/amavisd/maia/scripts/database.cfg”;	my $cfg = “/var/spool/amavis/maia/scripts/database.cfg”;
	my $key_file = undef;	my $key_file = “/var/spool/amavis/maia.key”;
process-quarantine.pl	my $subroutine = “/var/amavisd/maia/scripts/process-quarantine-sub.pl”;	my $subroutine = “/var/spool/amavis/maia/scripts/process-quarantine-sub.pl”;
	my $pid_file = “/var/amavisd/.process-quarantine.pid”;	my $pid_file = “/var/spool/amavis/.process-quarantine.pid”;
send-quarantine-digests.pl	my $cfg = “/var/amavisd/maia/scripts/database.cfg”;	my $cfg = “/var/spool/amavis/maia/scripts/database.cfg”;
! Do not put a trailing slash on the URL !	my $base_url = “http://www.example.com/”;	my $base_url = “http://mailscan.retnet.co.uk/mail”;
	my $template_dir = “/var/amavisd/maia/templates/”;	my $template_dir = “/var/spool/amavis/maia/templates”;
send-quarantine-reminders.pl	my $cfg = “/var/amavisd/maia/scripts/database.cfg”;	my $cfg = “/var/spool/amavis/maia/scripts/database.cfg”;
stats-snapshot.pl	my $cfg = “/var/amavisd/maia/scripts/database.cfg”;	my $cfg = “/var/spool/amavis/maia/scripts/database.cfg”;

Setup the Maia database tables in MySQL

To import the database schema into MySQL is a single terminal command.



In a terminal window as root, run

mysql -u root -p maia < /root/maia-1.0.0/maia-mysql.sql

You will be prompted for the root password (**maia** is not the password in the line above. This is actually the database name). After entering this, as long as no errors appear it has worked successfully.

Replace amavisd-new with a maia patched version

You need to replace both the executable and configuration file.

To start the change, shutdown amavisd, which is easier at the command line.... run

/etc/init.d/amavis stop

Rename the original executable version of amavis by running

```
mv /usr/sbin/amavisd /usr/sbin/amavisd-orig
```

Now copy the new version over using the same name as the original by running

```
cp ~/maia-1.0.0/amavisd-maia /usr/sbin/amavisd
```

To replace the SuSE default amavisd configuration file, copy the maia version over:

```
mv /etc/amavisd.conf /etc/amavisd.conf.orig
cp ~/maia-1.0.0/amavisd.conf.dist /etc/amavisd.conf
```

The following configuration changes are initial only. You should visit http://www.ijs.si/software/amavisd/ and have a read to see what other options are available.

Using Gedit, open **/etc/amavisd.conf**

*Tip: It's generally easier to read these files without the text wrapping over lines. To turn this off, select the **Edit** menu then **preferences**. Select the **View** menu and unselect the option **Enable text wrapping**.*

Scroll down the file and check/change the following:

Change the **$daemon_user** and **$daemon_group** to the user/group that will run amavisd, so that it looks like:

```
$daemon_user = 'vscan';
$daemon_group = 'vscan';
```

Change $mydomain to your e-mail domain name, so that it looks similar to:

```
$mydomain = 'retnet.co.uk';
```

Set the $MYHOME variable to the vscan home directory, so that it looks like:

```
$MYHOME = '/var/spool/amavis';
```

Scroll down to the **$log_level** line. You will want to increase the log verbosity for testing purposes, so change the default setting of **0** up to **5**. Once you're done testing, decrease it down to **2**. The level of 2 will show all scores for each test that matched each e-mail which *will* be very handy in troubleshooting scores of e-mails that either pass through or get caught.. and you will get these requests.

```
$log_level = 5;
```

Scroll down a little further and make sure SpamAssassin network tests are enabled, by checking that

$sa_local_tests is set to zero (it should be by default)

```
$sa_local_tests_only = 0;
```

Scroll down a couple of lines further and change **amavis** and **password** to whatever you are going to setup in MySQL for the username/password combination in MySQL which is setup in the next section:

```
@lookup_sql_dsn = (['DBI:mysql:maia:localhost, 'user', 'password'] );
```

Scroll down to the **$virus_admin** option. I choose not to send e-mail notification of spam or viruses to an admin, as there's too much spam to contemplate sending notifications for those, and most viruses are sent from computers infected with mass mailing worms or viruses of their own.

I suggest you put a comment, (#) hash symbol, in front of the **$virus_admin** option so that it looks like

```
#$virus_admin= “virusadmin\@mydomain”;
```

However if you do want these notifications, change **virusadmin** to your own name, and enjoy the extra mail!

Just below these lines, comment out all the **@addr_extension** lines. These lines are used for what's called 'plus addressing'. When these options are enabled, if spam and/or viruses are configured to be passed through the system (and usually tagged to identify them as spam/virus), amavisd-new will add the suffix as defined by the options to the user's name e.g. jsmith+banned@retnet.co.uk.

E-mail servers that recognise this form of addressing will automatically place the e-mail in a sub folder of the same name in the users mailbox (wouldn't it be nice if most well known commercial systems handled this!), but most don't and simply bounce the message as they see the address extension as a literal part of the user's name.

Make the lines look like:

```
#@addr_extension_virus_maps
#@addr_extension_spam_maps
#@addr_extension_banned_maps
#@addr_extension_bad_header_maps
```

Set your gateway hostname by changing **$myhosname** to look like

```
$myhostname = 'mailscan.retnet.co.uk';  # must be a fully-qualified domain name!
```

Find the anti-virus section starting with:

```
['ClamAV-clamd',
```

and change the **\&ask_daemon** line, so that the lot looks like:

```
['ClamAV-clamd',
  \&ask_daemon, ["CONTSCAN {}\n", "127.0.0.1:3310"],
  qr/\bOK$/, qr/\bFOUND$/,
  qr/^.*?: (?!Infected Archive)(.*) FOUND$/ ],
```

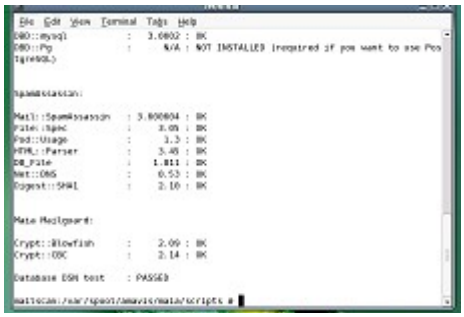
You're finished with this file now, so save the file and close the text editor.

Now re-start the modified amavisd-new service by running

```
/etc/init.d/amavis restart
```

If all goes well you will see a **green** **done** notification.
If you get a @decoders error, make sure you have downgraded the amavisd configuration file as I pointed in earlier in 'Setting up amavisd-new'.

Confirm your Maia configuration



To test the setup to date, in your terminal window, run

```
/var/spool/amavis/maia/scripts/configtest.pl
```

If the test passes, everything **except** DBD::Pg will return a result of passed or OK.

The DBD::Pg test is only applicable if you are using a PostgreSQL database server, which you aren't if you've followed my instructions.

Install the website files

It's now time to install the web site files that will be served up by Apache.

Typically these files are installed in a directory such as mail, so that you visit a URL such as http://mailscan.retnet.co.uk/mail but you can install them anywhere you like.

It's quicker using the command line so this is what you will use.. it's also good practice.

Open a terminal window as root, make a mail directory off the root of the web server, change into the **/root/maia-1.0.0** directory then copy all the files in the php directory over, by running

```
mkdir /srv/www/htdocs/mail
```

```
cd ~/maia-1.0.0
cp -R ./php/* /srv/www/htdocs/mail
```

Lastly you need to change permissions on the ../mail/themes directory to allow the web server write access.

To do this, in the console run

```
chgrp vscan /srv/www/htdocs/mail/themes/*/compiled
chmod 775 /srv/www/htdocs/mail/themes/*/compiled
usermod -G vscan wwwrun
```

These settings in order:

- change group membership to the vscan group for the matching directories
- change file permissions for the matching directories
- Add the web server user to the vscan group

And lastly re-start the Apache2 web server to pick up the new group assignment from the previous command, by running

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

Configure the PHP website

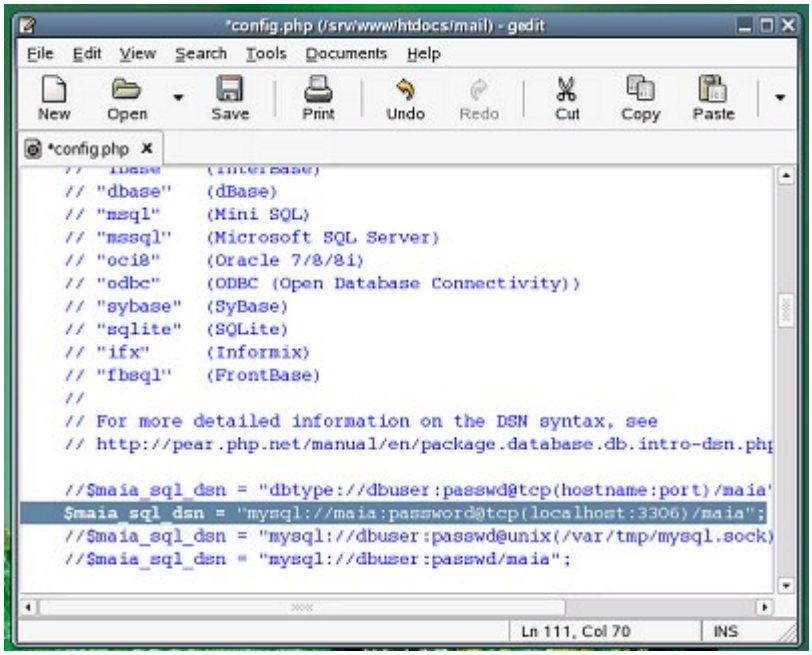
You now need to make a couple of modifications to the website configuration file, to tell it where the database is and the type of authentication into Maia Mailguard you want to use.

First rename the template configuration file by running

```
cd /srv/www/htdocs/mail
mv ./config.php.dist ./config.php
```

Now edit the file using Gedit

Tip: You can run gedit from the command line saving you a few mouse clicks. In the /mail directory run ***gedit ./config.php*** *to run gedit with the config.php file reading for editing.*



Using Gedit, open the file **/srv/www/htdocs/mail/config.php**

Scroll down to the **$maia_sql_dsn** line and change amavis to your database username and change passwd to your database user's password, so that it looks something like:

```
$maia_sql_dsn = "mysql://maia:password@tcp(localhost:3306)/maia";
```

Also if you choose to enable pie graph charts on the statistics pages (more on that later), the default graphing font is a less than ideal, so to improve it you can specify your own by scrolling down a little more to the **$chart_font='';** section and change it to look like:

```
$chart_font='VeraBd';
```

The font specified can be any truetype font on your system. With no path specified PHP will look in **/usr/X11R6/lib/X11/fonts/truetype/** and will also assume a .ttf extension. If the truetype font you want to use is installed elsewhere on your system, you will need to explicitly specify it here. For example to explicitly specify the VeraBd font it would look like:

$chart_font=”/usr/X11R6/lib/X11/fonts/truetype/VeraBd.ttf';

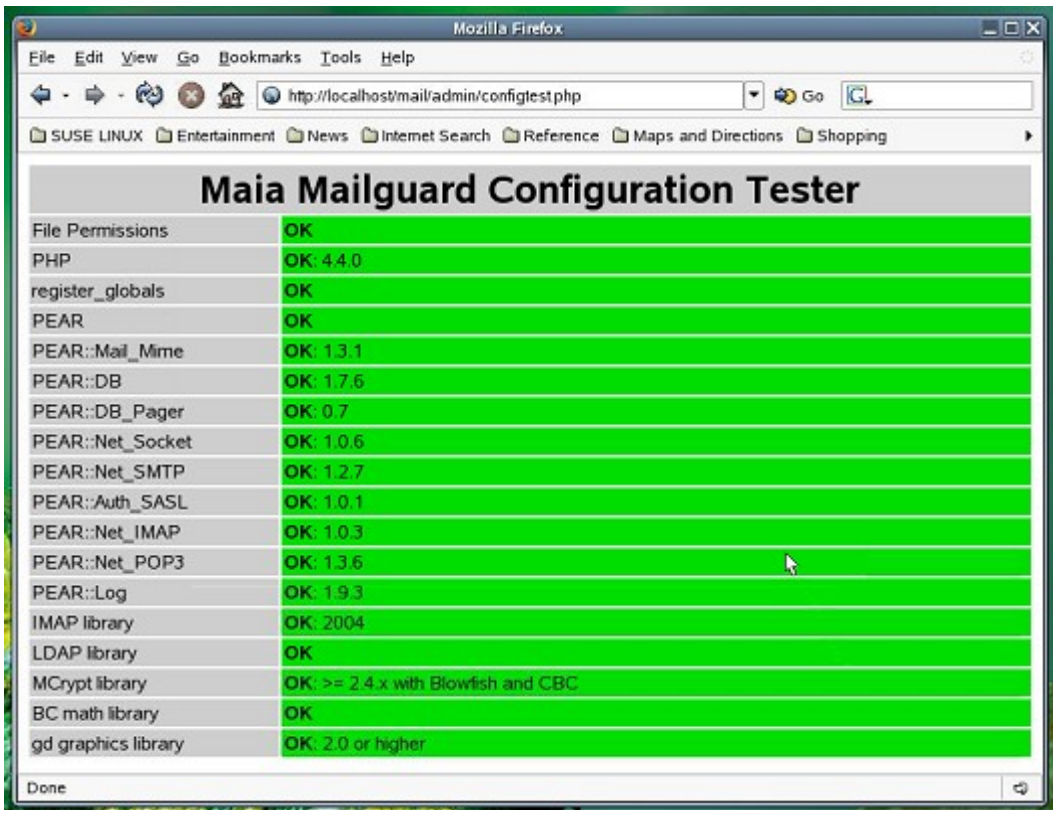
One last note on fonts is that if you already have a valid Windows license, you are allowed to use any truetype font on that windows system as well, by copying them from the C:\Windows\Fonts directory.

Save the file and exit.

In this example we are using Internal authentication, meaning users will be setup for authentication against the Maia database, although you can configure it for LDAP v2, Exchange 5.5 (experimental), POP3, IMAP or SQL (another SQL database) authentication. Read the Maia Mailguard installation notes on their website for more information on extending this functionality.

Test your Maia Mailguard configuration

This is quite easy. Just open configtest.php in your Firefox web browser



In your web browser goto the address **http://mailscan/mail/admin/configtest.php**

As long as you got no errors, which should be the case if you've followed this guide, then it's almost good to go. (Re-visit the installation or configuration section for any related errors)

Generate your encryption key

Maia can store e-mail in MySQL in encrypted form to protect sensitive e-mail from prying eyes. Although optional I think this should be mandatory considering the potential for abuse.

It uses 56-bit blowfish encryption so it's quite reasonably secure.

To create your key file, in a terminal window and as root run

```
cd /var/spool/amavis/maia/scripts
./generate-key.pl > /var/spool/amavis/maia.key
```

It's also good practice to keep a separate backup copy of this file in a safe place just in case something should happen to the original, as you'll need it to recover any e-mail in the database.

Load SpamAssassin rules into Maia Mailguard

Amavisd-new will use SpamAssassin to perform spam checks on incoming e-mails and Maia Mailguard will read the e-mail headers that are added as a result.

In Maia Mailguard you can see what rules were triggered and the score for each rule for an e-mail by opening that e-mail in the quarantined e-mail view (will only show basic text or html, and won't run anything so you can even view virus e-mails safely this way), or by checking the system wide SpamAssassin statistics page.

For Maia Mailguard to tally up what rules are being used and their associated scores, you will need to run a script to load those details into the MySQL database.

Open a terminal window and as root run the following command

```
/var/spool/amavis/maia/scripts/load-sa-rules.pl
```

It will display each rule as it find it.

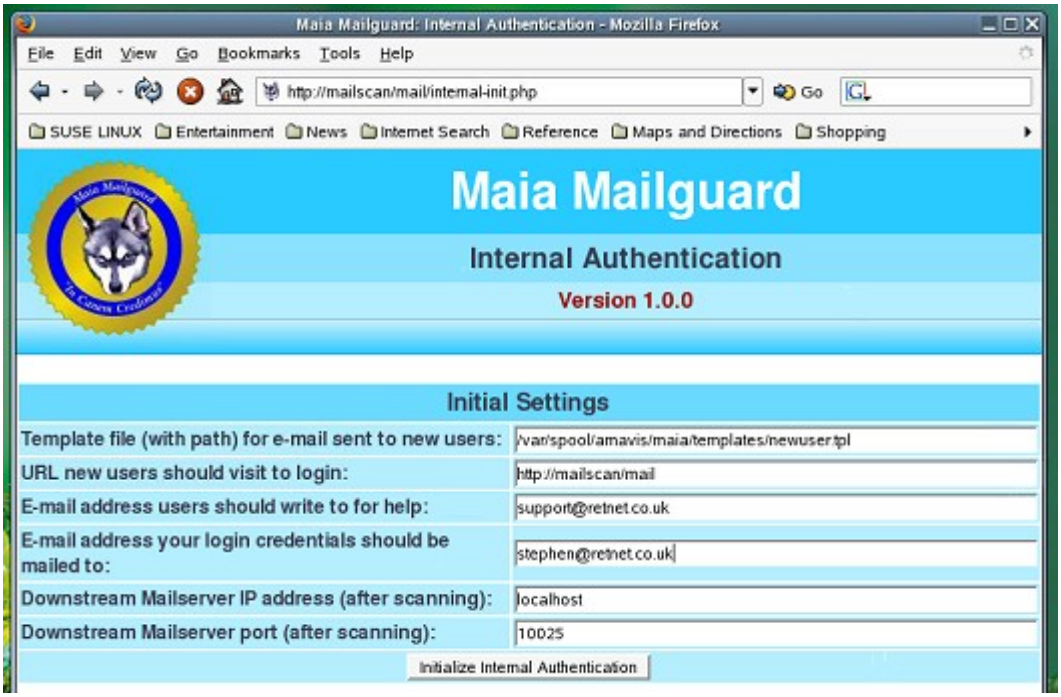
IMPORTANT NOTE: Every time you add new SpamAssassin rules or change their associated scores you will need to re-run this script in order to update the Maia Mailgaurd database. If you don't, no harm will come of it, although you will not be able to see associated statistics of SpamAssassin rules and scores. You can alleviate some pain here by adding this script to a cron job for the root user to run say once per day, so even if you forget to run it yourself, the system will be automatically updated.

First time login

If you are going to use any authentication method other than internal authentication such as POP3, IMAP or LDAP you do not need to perform the Internal authentication setup and can move onto registering yourself as the Super Administrator that is detailed next.

Internal authentication setup

If you are using the **Internal Authentication** method (as I am), open your web browser and hit the address: http://mailscan/mail/internal-init.php



As long as your using the Internal authentication method (as is the case in this guide) you will see this screen. *If you just get a blank screen, make sure permissions are correct for the /mail/templates directories and that php.ini has the current path in it's include_path statement.*

In the **Template file** field, type in the full path, of

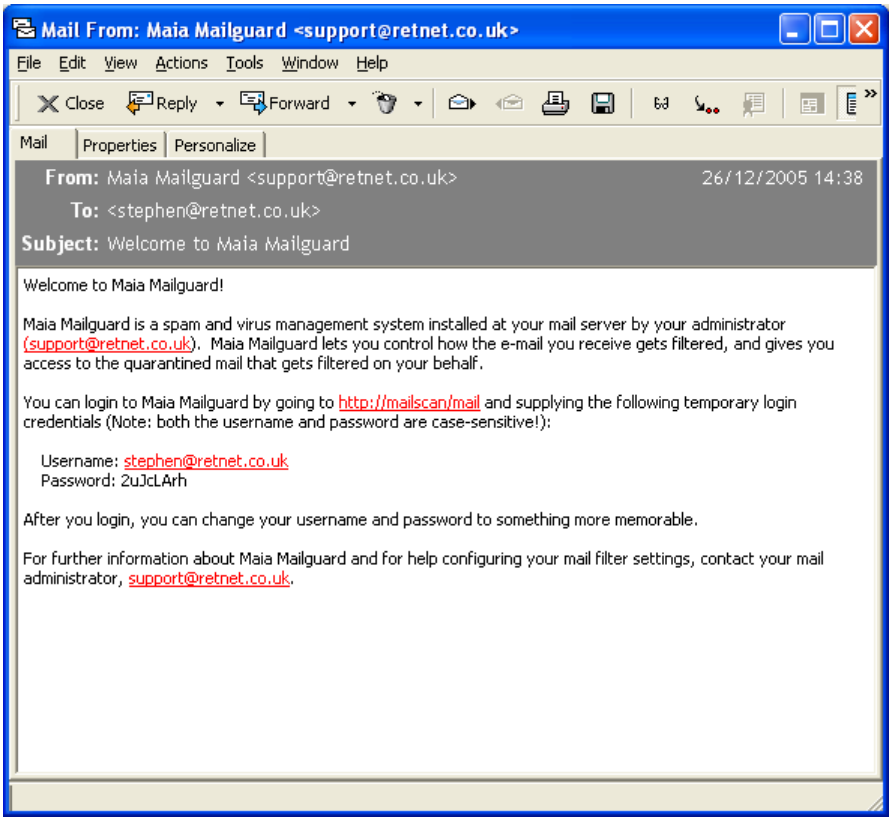
/var/spool/amavis/maia/templates/newuser.tpl

In the **URL new users** field, type the full URL to your e-mail gateway login page, being something like

http://mailscan/mail

And the other settings should be self explanatory.

Hit **Initialize Internal Authentication**



You should at this point see an e-mail in your regular inbox. If not, there are 3 common reasons:

1) You typed in the wrong path to newuser.tpl in the internal-init.php file(will be seen as "fopen" errors in the Apache2 error log file at /var/log/apache2/error_log)
2) Permissions are not correctly set on the template directory so that the wwwrun user that Apache2 runs as cannot read the newuser.tpl file (will be seen as "fopen" errors in the Apache2 error log file)
3) Postfix is not set correctly to relay for your domain in /etc/postfix/main.cf. Most common error I've seen is there should not be an @ symbol in the relay_domains option.

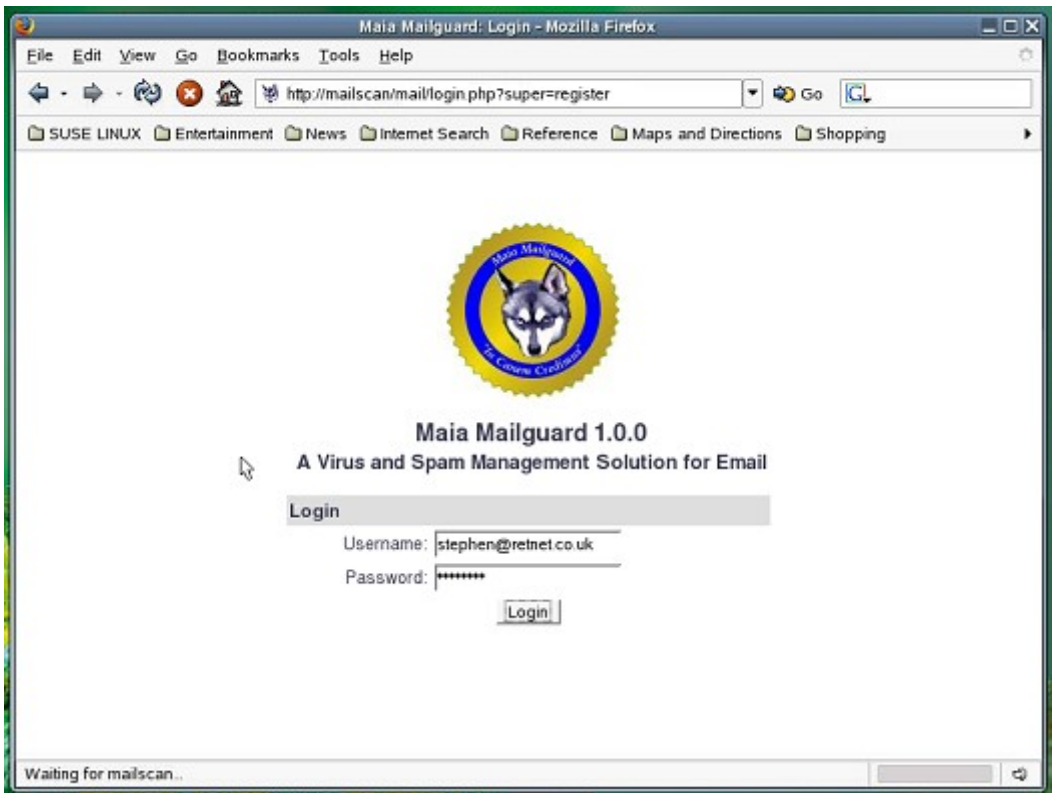
By the way, this is a test box and doesn't actually exist, so don't bother trying to log in with either these user names or passwords on my own website as they simply won't work... because they don't exist.

Note down the **user name** and **password**.

Super administrator registration

Now to register yourself as the super administrator, log in via the web page:

http://mailscan/mail/login.php?super=register



Once logged in, you'll be at your personal **Welcome** page.

The settings you see on this screen are for your own personal account, as denoted by the **User** under the **Welcome** title at the top.

The Current protection level setting is a predefined set of options for what checks take place and at what score e-mail is considered spam. These are configured in /srv/www/htdocs/mail/config.php under the // **Default Protection Level** section and the default settings are:

Off – No scanning at all occurs (default)

Low – Virus protection only

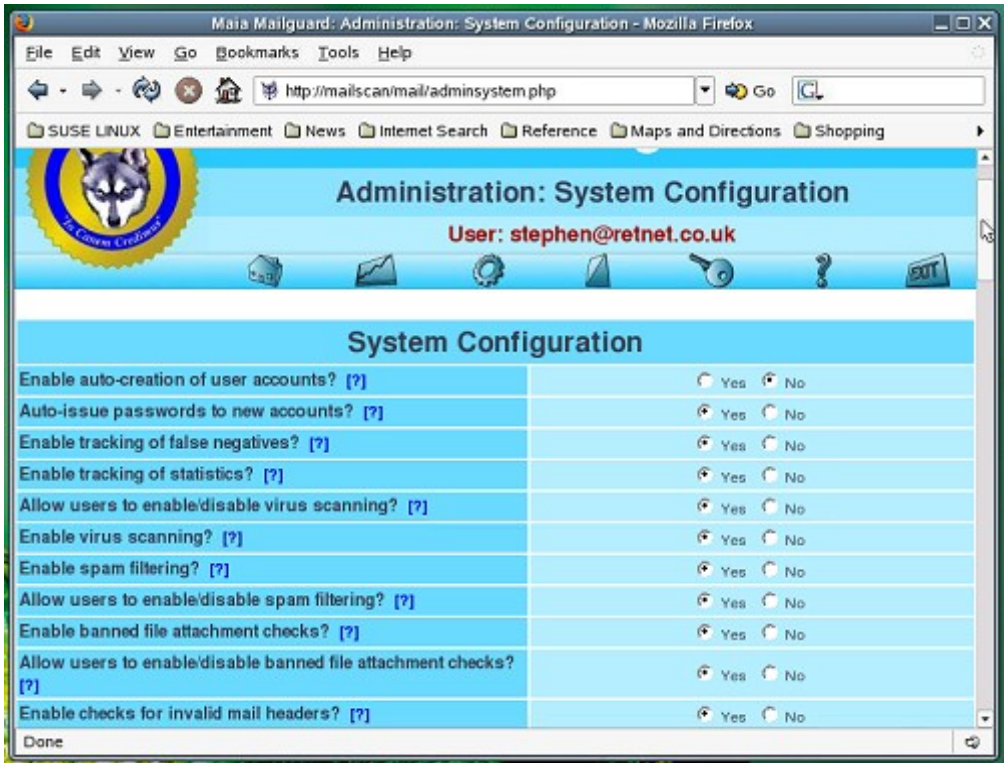
Medium – Spam is marked at a score of 5 but not quarantined, all attachment types and e-mails with bad SMTP headers are allowed

High – E-mails are tagged with spam scoring information at a score of 1 and quarantined at a score of 5, all attachments types, bad headers and viruses are quarantined.

For testing purposes you will be creating custom protection settings, so for the moment this area won't apply. After testing, set the protection levels how you see fit, then use the predefined protection level for all users for easier day-to-day use.

The Cache Contents area is a quick view of what e-mail is currently quarantined in your cache and at the bottom of the screen are some quick overall stats on how many spam and viruses have been caught for yourself and the entire system.

It generally makes sense to setup the overall site settings first, so click the **key icon**  (third from right) to enter the admin section of the site then the **System Configuration** link.



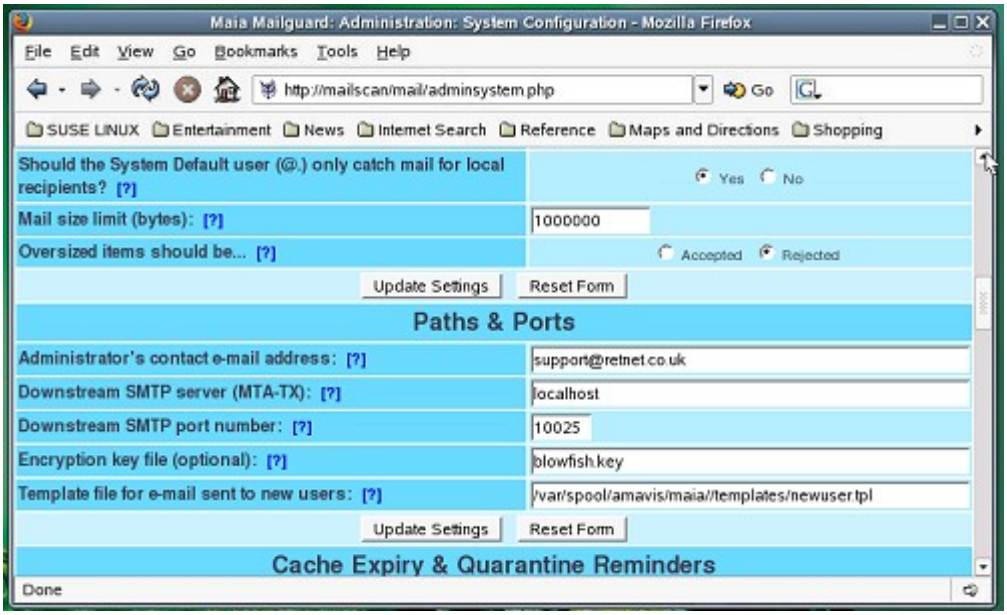
Of course you can change anything you like in here, but to start with it's best to leave most at their default until after you've read the associated help – click on the **?** symbol next to each item for an explanation.

The first option, **Enable auto-creation of user accounts?** can be either a major headache or an administrative life saver. Leaving it set to **No** means an administrator will need to create each user account on the system manually. Changing to **Yes** will enable Maia Mailgaurd to automatically create a user account the first time an e-mail is sent to a user of any configured domains. If your postfix installation does not verify the recipient names are valid e-mail addresses, this can create potentially thousands of dummy accounts, usually caused by spam dictionary attacks. Information on configuring postfix to perform recipient verification is available in the Tweaks and Tightening section.

The **Mail size limit** is of particular note as mail size settings can be configured in postfix, SpamAssassin, Maia Mailguard and MySQL, all of which have different meanings.

The size description here relates to the mail size in raw (encoded) format, so if the e-mail has binary attachments, then the 'human readable' limit is somewhere around 700 - 750kB in binary form, due to the way binary attachments are encoded for SMTP delivery.

This setting relates to the upper size limit of e-mails that Maia Mailguard will perform content-filter checking on. Note that the MySQL **max_allowed_packet** setting in **/etc/my.cnf** must be set to AT LEAST the same value as the Maia Mailguard **Mail size limit**. If you do not, Maia will fail to quarantine e-mail between the max_allowed_packet size and the Mail size limit, which you will never want.



If you want to automatically trust any e-mails that are larger than this, change the **Oversized items should be** option to **accept**.

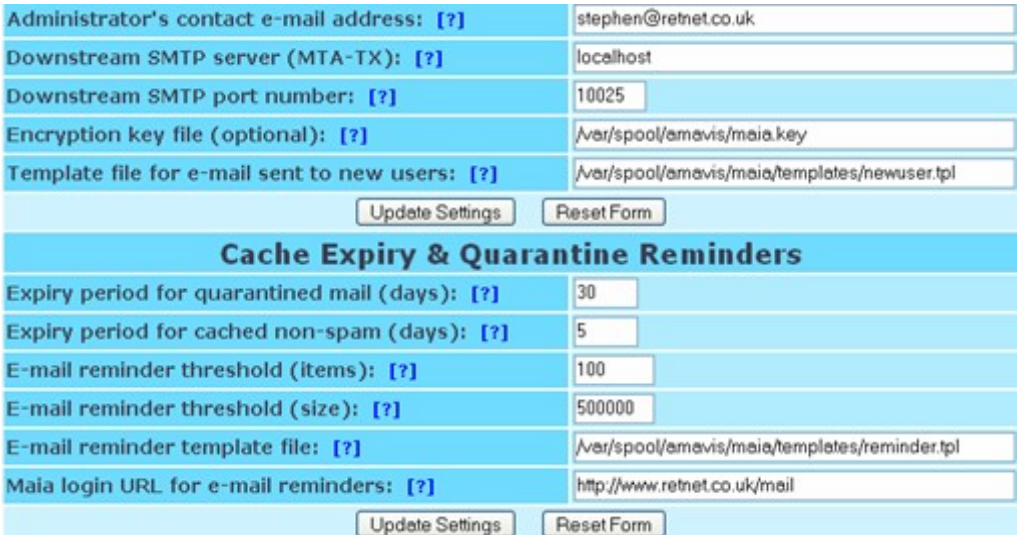
To keep things simple, it's generally a good idea to set the **Oversized items should be** setting to reject, and keep the **Mail size limit** (*note the postfix default is 10Mb, while in Maia it's* 1Mb) the same in Maia Mailguard as it is in postfix.

So for this guide, change the **Mail size limit** to read **10240000**

A quick jump from the current setup of Maia Mailguard if you want to increase the mail size limit from the postfix default of 10Mb:

If you do this, you need to edit the postfix configuration file **/etc/sysconfig/postfix** and change the setting **POSTFIX_ADD_MESSAGE_SIZE_LIMIT="10240000"** to the number of bytes for your upper mail size limit. After this, you'll need to open a terminal console and run **SuSEconfig** to update the postfix configuration file and lastly run **postfix reload** for postfix to read the new setting. Then in Maia Mailguard, change the **Mail size limit** setting to be the same. And lastly the MySQL setting if you want the option of quarantining these.

Lastly on e-mail sizes is a note on SpamAssassin. It will by default only scan for spam in e-mails up to 250Kb in size. The thinking by the Apache group who develop SA is that no spammer in their right mind would send e-mails larger than this, else it would end up costing the spammer too much money to send the e-mails out. To date I've never seen a spam e-mail over 20Kb in size so you're fairly safe with this default setting.



Scroll down and add the full path to the **Encryption key file (optional):** field to read

/var/spool/amavis/maia.key

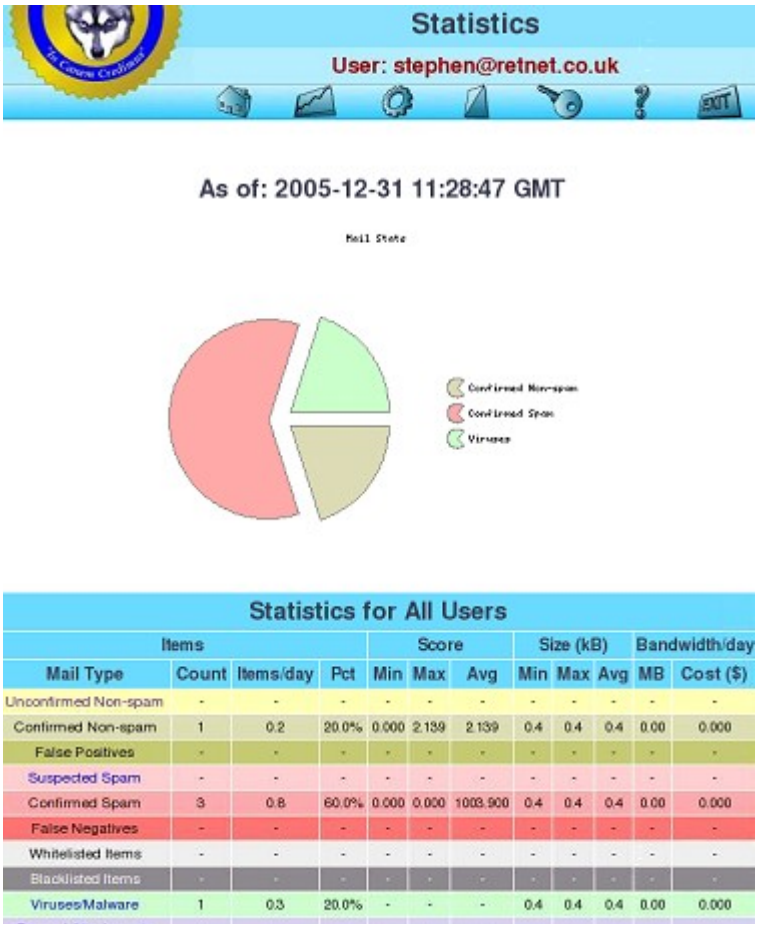
In the following section **Cache Expiry & Quarantine Reminders** change the **E-mail reminder template file:** setting to be the full path of the file, like

/var/spool/amavis/maia/templates/reminder.tpl

You need to do this else Maia Mailguard won't find the template when it tries to e-mail end users to remind them to confirm their spam and non-spam (clean mail). You can also customise the reminder template, which is covered in the Maia Mailguard installation guide at http://www.maiamailguard.com/install.php under the **18. Customise the e-mail templates** section.

For this guide you won't be setting up per-user settings, thus negating the need for sending reminders but at least this setting is now correct if you ever decide to use it.

The **Display** and **Virus Information** sections are pretty self explanatory, although it's worth noting there are license restrictions on how much you can change the display without having to pay the author which allows you to remove all traces of the Maia Mailguard logo from the site. Read the license note at http://www.maiaMailguard.com/license.php .



Bandwidth Accounting lets you see how much your e-mail is really costing you on average per day. Enter your currency symbol and the cost per gigabyte that your bandwidth costs. If you have a leased line and pay for truly unlimited bandwidth, see if you can get some accounting info on cost and bandwidth utilisation statistics from your ISP to help you figure out and average.

Site Statistics Reporting is a nice way of letting the people who develop Maia Mailguard know how well it's working in real life at other sites. If you're interested in joining in, you'll need a Maia ReporterID which you can read about by clicking the **?** against the **Your site's Maia Reporter ID:** field.

Statistics for All Users

	Items			Score			Size (kB)			Bandwidth/day	
Mail Type	Count	Items/day	Pct	Min	Max	Avg	Min	Max	Avg	MB	Cost ($)
Unconfirmed Non-spam	-	-	-	-	-	-	-	-	-	-	-
Confirmed Non-spam	1	0.2	20.0%	0.000	2.139	2.139	0.4	0.4	0.4	0.00	0.000
False Positives	-	-	-	-	-	-	-	-	-	-	-
Suspected Spam	-	-	-	-	-	-	-	-	-	-	-
Confirmed Spam	3	0.8	60.0%	0.000	0.000	1003.900	0.4	0.4	0.4	0.00	0.000
False Negatives	-	-	-	-	-	-	-	-	-	-	-
Whitelisted Items	-	-	-	-	-	-	-	-	-	-	-
Blacklisted Items	-	-	-	-	-	-	-	-	-	-	-
Viruses/Malware	1	0.3	20.0%	-	-	-	0.4	0.4	0.4	0.00	0.000

The **Charts** section will allow the display of graphs representing the statistics it accumulates.

Change this to Yes if you want users to have the ability to see these charts.

Users, including yourself will also need to set the miscellaneous personal setting 'Display Graphical Charts?' to Yes – This is explained in the next section.

An example of what the charts should look like for spam is here. The chart for viruses is basically the same.

To access site wide spam, virus and SpamAssassin rule statistics, at the Statistics page (by default your own stats) click the link at the bottom of the page.

To finish scroll to the bottom and click **Update Settings**.

With the base configuration out of the way you need to understand the different levels of settings in Maia Mailgaurd being

- Domain: System Default (@.)

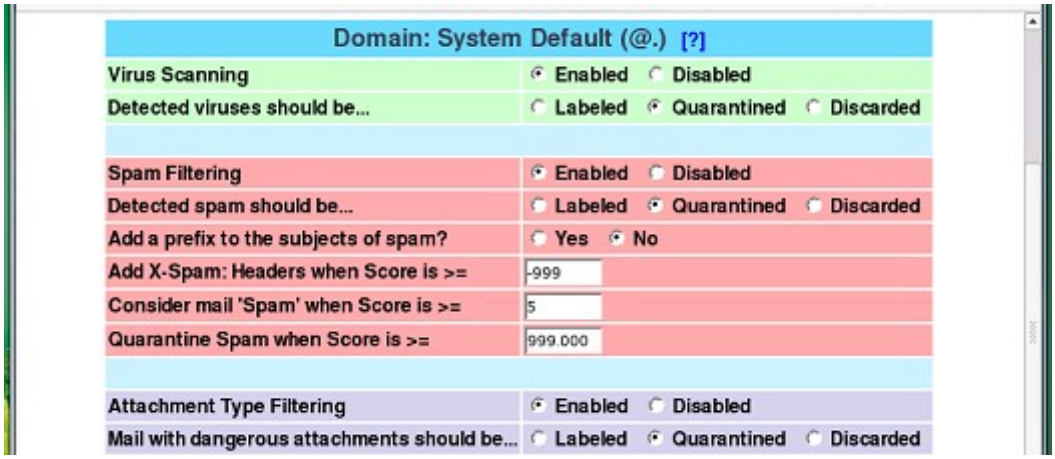
- System Default User (@.)
- Domain defaults (e.g. retnet.co.uk), and
- User defaults (e.g. Stephen@retnet.co.uk).

Any domain settings will override the system default settings and any personal settings will override the system default user and both the domain and system defaults.

E-mail for domains that are relayed by the gateway but not specified in Maia Mailgaurd will be scanned based on the **Domain: System Default (@.)** settings.

This means as an administrator although you set up the defaults, a user with a Maia account can override settings for their own e-mail needs, for example by decreasing the score threshold for spam or disabling virus checking altogether (although these options can be disabled by the admin as well so end users cannot change them).

So now you should configure the **Domain: System Default @** settings. Click the **key icon** then select the **Domains** link then the **System Default @.** link to open the default domain properties page.



By default no virus or spam scanning takes place, so you should probably change these.

In the **Virus Scanning** field, change it to **Enabled** and in the **Detected viruses should be** field select **Quarantined**. This will let you rescue an e-mail caught with a virus if you need to.

In the **Spam Filtering** field, change it to **Enabled** and in the **Detected spam should be** field select **Quarantine**. By quarantining spam you can recover false positives and can also confirm the e-mails as spam which makes the Maia statistics page more accurate.

There's no real need to **prefix** the subject of spam if you're quarantining it, so leave this set to **No**.

At the **Add X-Spam: Headers when Score is >=** field, change it to **-999** This will mean all incoming e-mail will have headers added to them which will help you identify why certain e-mail was or was not identified as spam.

At the **Consider mail 'Spam' when Score is >=** field, change it to **5** This is the SpamAssassin default value and is a reliable setting for me. After some time you may find that some spam e-mails are sneaking in just under the 5.0 score limit, at which point you may decide to reduce this value, but for the time

being, I'd suggest **5** is a good start.

The **Quarantine Spam when Score is >=** value is only used when spam is labeled and passed through instead of quarantined. The idea here is that if a spam e-mail registers a high enough score, then don't send it through to the end user and automatically consider it spam. If you choose to quarantine spam anyway, this setting is irrelevant and you can leave it.

Change the **Attachment Type Filtering** to **Enabled** and set it's action to **Quarantined**.

At this point leave the **Bad Header Filtering** at **Disabled**. The tests done for this check to see if the e-mail header conforms to proper Internet standards. If it doesn't, chances are it's a malicious e-mail or spam sent from a home made program that's been written in a sloppy manner. In this case the first e-mail will be sent using the telnet program and won't conform to it's strict tests. You can turn it on after the test to ensure it's all setup correctly.

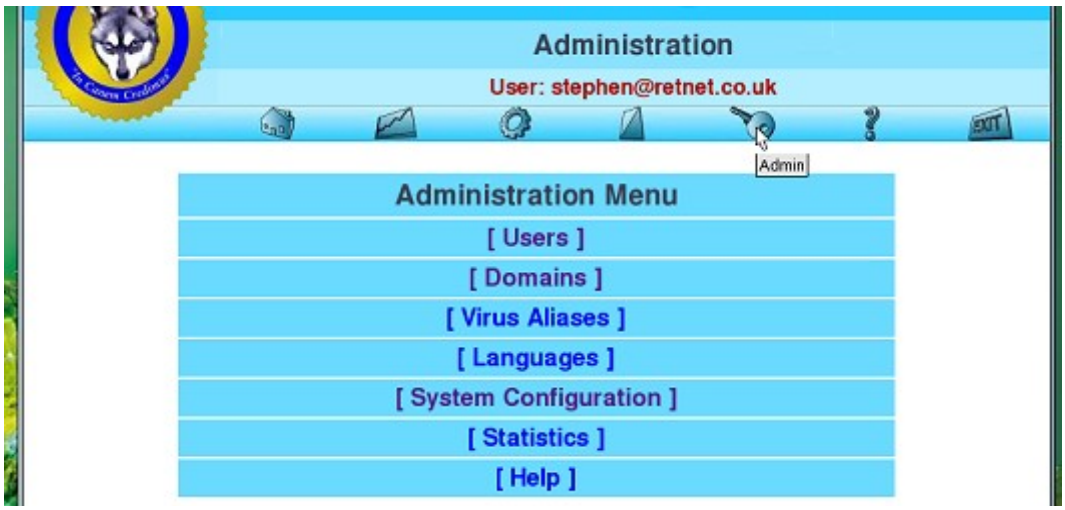
To properly train SpamAssassin and produce accurate statistics in Maia, you should change the **Should non-spam items be cached?** to **Yes**. This allows Maia Mailguard to train SpamAssassin on what is clean e-mail which will over time help your system become more accurate. If you don't, the SpamAssassin database will become biased towards spam and you will get more falsely caught spam than you would otherwise.

It also is a good way of identifying spam that passes through. By telling Maia Mailguard an e-mail it thinks is non-spam is actually spam, it helps it to report this to SpamAssassin which will learn to identify that and similar e-mails in future as spam.

Click the **Update This Domain's Defaults** button.

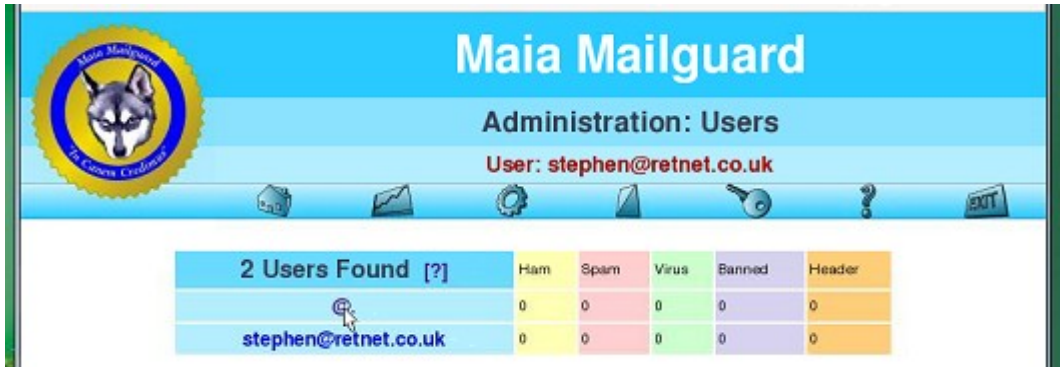
Now that you've configured the default domain settings you should set the default user settings.

To change the default user settings, you edit the configuration for the System Default User (**@.**) by clicking the **key** icon,



selecting the **Users** link,

clicking the **Find Users** button

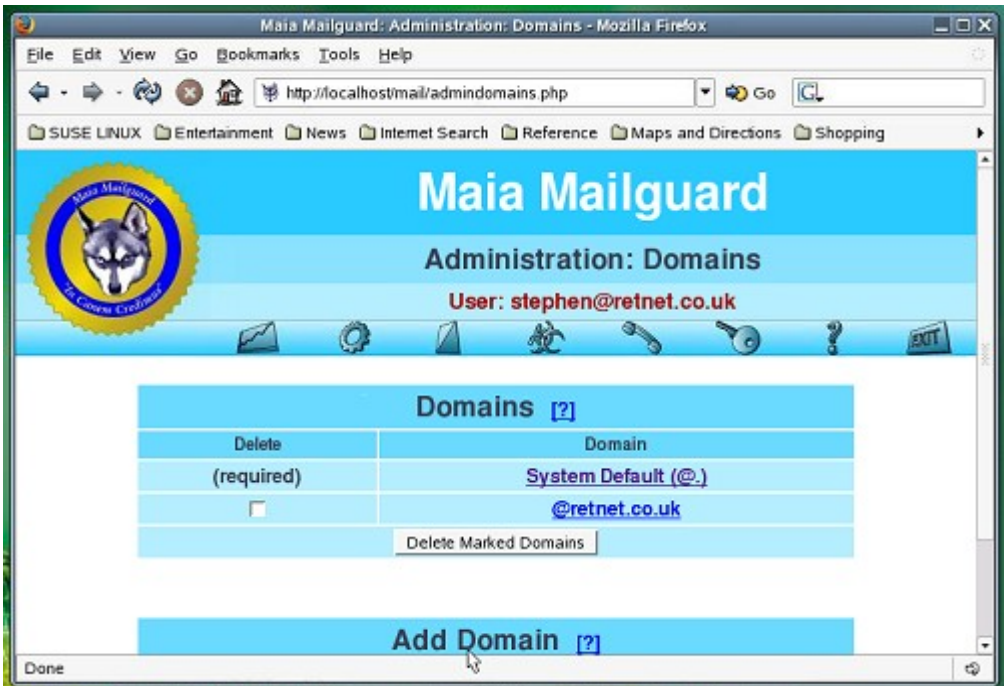


then selecting the **@.** user.

You will now see under the Welcome sign the user you are impersonating, being **System Default User (@.)**.

Click the **Cog** menu button  (3rd from the left) and change any miscellaneous settings as you wish then click the **Update Settings** button. The settings you enter here will be used as defaults for all new users that are created within Maia.

Now that you've finished setting up all the template settings and global configuration, it's time to configure your own e-mail domain.



To create your own domain in Maia Mailguard, click the **key icon** then the **domains** link again.

In the **Add** Domain setion, in the **New domain:** field enter in your own domain name such as:

@retnet.co.uk

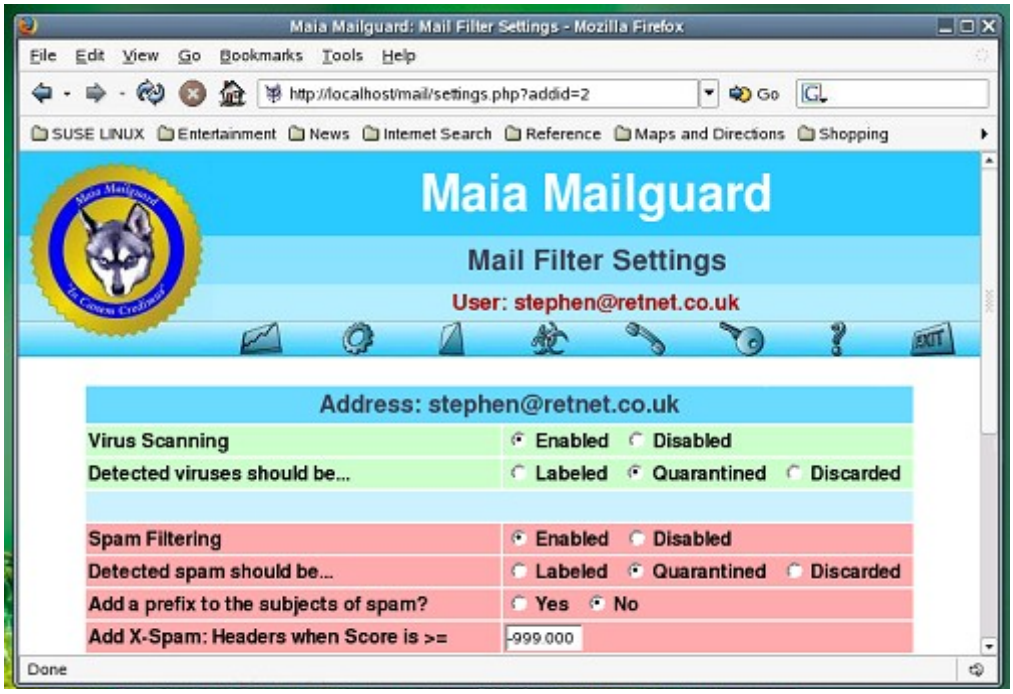
then click the **Add Domain** button.

Because you've already configured the **System Default (@).** domain, your new domain will automatically be created with those defaults. Note if you change the system defaults, this will not change your domain settings, as it will only take affect on newly created domains.

Now although you've created the **System Default (@.)** and **@retnet.co.uk** settings, you should check and change your own personal user before testing.

This is because although you have setup all the necessary defaults, your own account was created beforehand, based on the Maia Mailguard defaults so no scanning or graphs of any sort will be done. This is deliberate on the authors part, to make the initial setup transparent to your e-mail system.

To change your own settings you need to revert back to your own user so just click on the **Key** icon again. This will change your current user back to yourself, which you can see from the user that's identified above the menu icons. Now click the **Cog** icon, then your **Primary Address** link, which takes you to your user specific scan settings.



To keep it all uniform I suggest you start with the same settings as in the domain defaults. Set them up then click the **Update This Addresses' Settings** button at the bottom of the page.

The very last stage before testing the lot, is to setup the scripts you installed in **/var/spool/amavis/maia/scripts** to run at regular intervals. You will do this using cron.

Scheduling the maintenance scripts

Open a terminal console, and as root open the crontab editor by running:

```
crontab -u vscan -e
```

The **crontab** utility uses a basic text editor called **vi**, which although very powerful, is also quite difficult to get a handle on, so I'm only going to point out how to do the bare minimum.

When **vi** starts up, you will see a blank page. To start typing text into it you need to press the letter **i** which is the command for INSERT. When you press **i**, you will see --- INSERT --- displayed at the bottom of the screen.

If you make a mistake, the backspace key won't work. Move to the start of the mistake and use the **Delete** key instead.

The command syntax for cron is:

[minute] [hour] [day of month] [month] [day of week] [command]

You can use star * which means 'any', so a * in the minute column would mean that particular task will run every minute.

You can also divide time using **/x** where **x** is the divider, so for example if you wanted something to run every 15 minutes, you could type ***/15** in the minute column.

In the **day of week** column you can represent days using their first 3 letters.

Enough of that detour for now. If you are interested, you can find out a lot more about **cron** at http://www.unixgeeks.org/security/newbie/unix/cron-1.html

As a quick overview as to the scripts you will configure for scheduling and why:

expire-quarantine-cache.pl is used to help you automatically manage your maia database. If left all on it's own the database would very quickly fill up your entire hard disk. This script deletes e-mail that has been around for too long (5 days for non-spam, 30 days for spam by default) which is configurable in the Maia Admin settings.

load-sa-rules.pl will update the Maia Mailgaurd score tables with new rules and updated scores. Although you can just run this manually whenever you change a SpamAssassin score or add a new rule, I choose to run it automatically to ensure all scores and rules are always correct.

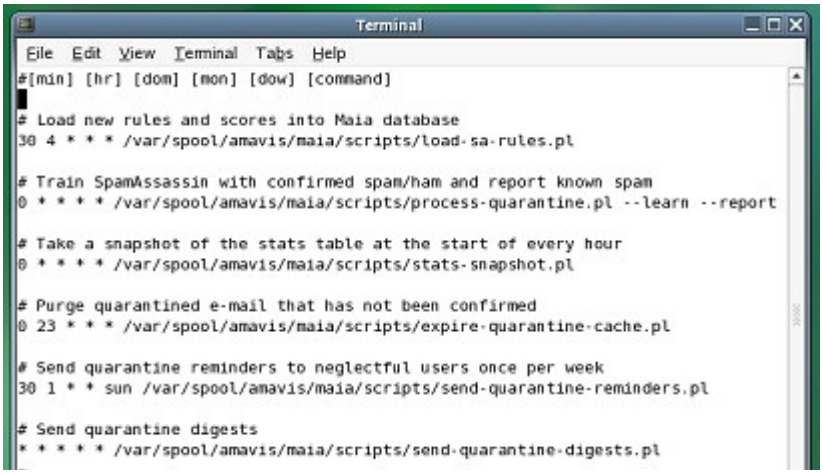
process-quarantine.pl is used on e-mail that you have confirmed in Maia Mailguard as non-spam or spam. This script can tell SpamAssassin to learn from the e-mail you've confirmed so it will become more accurate the longer you use it. If an identical e-mail has previously been learned as spam or non-spam it will not learn it again, so you may find that although there were 1000 spam e-mails processed, only 400 were learned... this is nothing to worry about. It will also report spam to the Razor and DCC networks.

send-quarantine-digests.pl sends a digest of all quarantined e-mails to your users. You can configure the sorting of the lists by changing the necessary values within this script. To actually send digests, the user needs to have their personal setting option **E-mail digest Interval?** set to at least 1. A value of 0 or less disables it. This script needs to run every minute as the configurable digest interval is also in minutes. There could be confusion if for example a user is configured to receive digests twice a day but you only run the script once per day...

You could also just tell your users they will get it once/twice per day if they set it to 1, then you can control the frequency using cron.

send-quarantine-reminders.pl is used when you have delegated spam/non-spam management to the end users themselves. This can potentially save you as an administrator a lot of time. If users become careless and don't confirm their e-mail, this script will send them a gentle reminder

stats-snapshot.pl updates the maia database with e-mail statistics which you can use by pulling the data from the maia database directly, as trend graphing functions are not currently available within Maia Mailgaurd (although current statistical graphs are).



Now moving on, back in your terminal console you should be in **insert mode**, so type in the following:

```
#[min] [hr] [dom] [mon] [dow] [command]

# Load new rules and scores into Maia database
30 4 * * * /var/spool/amavis/maia/scripts/load-sa-rules.pl

# Train SpamAssassin with confirmed spam/non-spam and report known spam
0 * * * * /var/spool/amavis/maia/scripts/process-quarantine.pl --learn --report

# Take a snapshot of the stats table at the start of every hour
0 * * * * /var/spool/amavis/maia/scripts/stats-snapshot.pl

# Purge quarantined mail that has not been confirmed
0 23 * * * /var/spool/amavis/maia/scripts/expire-quarantine-cache.pl

# Send quarantine reminders to neglectful users once per week
30 1 * * sun /var/spool/amavis/maia/scripts/send-quarantine-reminders.pl

# Send quarantine digests
* * * * * /var/spool/amavis/maia/scripts/send-quarantine-digests.pl
```

To save the file and exit, press the **[ESC]** key then press in order (i.e. not at the same time..):

```
:wq
```

then press **[Enter]**

Pressing the ESC key puts you in 'command' mode. You will see the ---INSERT--- tag disappear when you do this. The :wq command tells **vi** to write the file and quit. You don't need to specify a filename because the **crontab** program did that for you.

Be aware the cron scheduler will e-mail the user of any output that these scripts create, in this case being the **vscan** user. These can be handy for some troubleshooting and ensuring the maintenance scripts are running correctly.

If you would like to receive these e-mails, you need to add an e-mail **Alias** in postfix via YaST for the **root** user, on the **Incoming Mail** screen, which is explained in the **Configure Postfix** section.

Finally, after all this, it's time for testing...

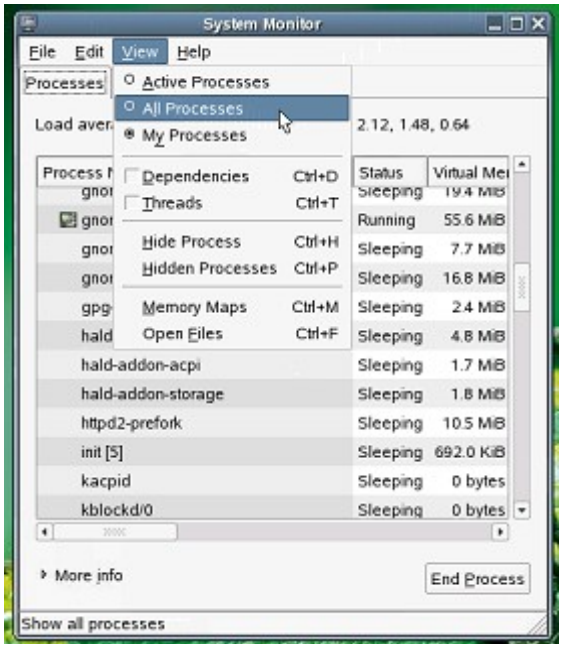
Checking all processes start correctly

The first test, is to simply make sure that everything will start after a reboot.

From the **Desktop** menu select **Log Out**, then choose **Restart the Computer**.

Once it's restarted and logged in as root, check that all required e-mail related services have started.

Run the **Gnome System Monitor** by clicking the **Applications** menu, then **System**, **Monitor**, and lastly the **Gnome System Monitor** program itself.



After it's started, change the view to see all running processes by clicking the **View** menu then select the option **All processes**. Also select the **View Dependencies** option in the same menu to see each how each

process is related.

Confirm the following processes are listed. If any are not, run the **System Run Level (Editor)** in YaST and make sure under the **Enabled** column the service is set to **Yes**

amavisd (master) *(Amavisd-new e-mail content scanner)*
amavisd (virgin child)
amavisd (virgin child)

clamd *(ClamAV anti-virus service)*

dccifd *(DCC client)*
dccifd

freshclam *(ClamAV database updater)*

httpd2-prefork *(Apache2 and associated services)*

master *(Postfix and associated services)*
pickup
qmgr

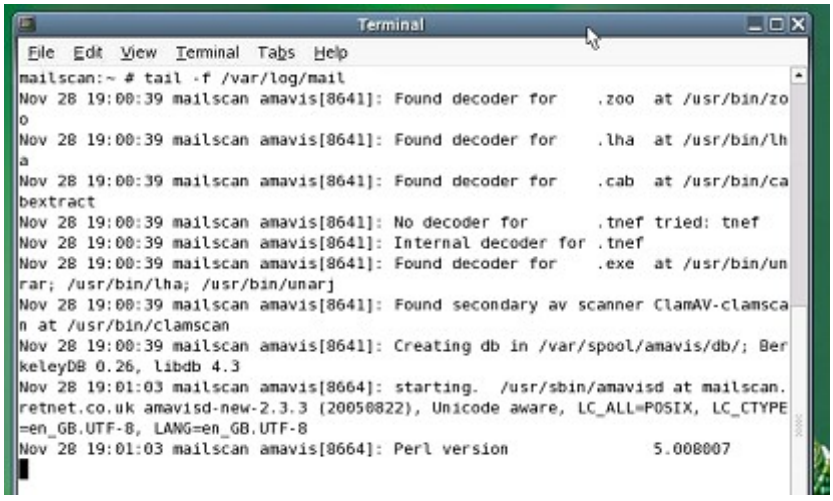
mysqld_safe *(MySQL and associated services)*
mysqld

You won't see processes for SpamAssassin or Razor2 because these don't run as services.

First test e-mail sent from the gateway itself

For testing and troubleshooting you will get to know the postfix mail log pretty well. It's located at **/var/log/mail**

One of the most handy command line tools I use is called **tail**. This utility will display the last x number of lines of a file. Where it really shines is in it's simple ability to monitor a text file and display it's changes in real-time, using a **-f** switch.

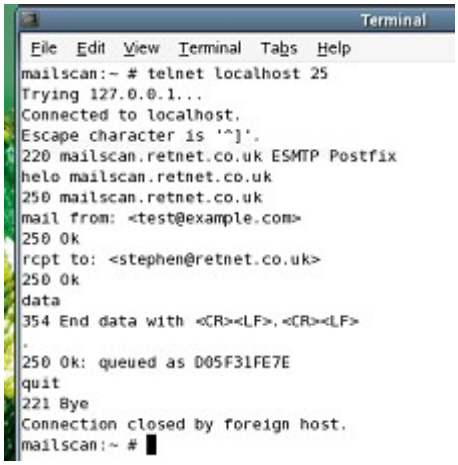


Open a terminal console as root and run

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

You will notice you are not returned to a command prompt. When you want to quit tail, just press **crtl-c**

To send a basic e-mail you will use the **telnet** program and directly communicate with the smtp postfix service.



Open another terminal console and run

telnet localhost 25

This will open an e-mail connection with the postfix smtp gateway, and you will be presented with a response similar to

220 mailscan.retnet.co.uk ESMTP Postfix

Now type in everything below. If you make a mistake just hit enter (because you can't delete). You will receive an error but don't worry and just type in the line again.

mail from: <test@example.com>
rcpt to: <stephen@retnet.co.uk>
data
test
.

where **stephen@retnet.co.uk** should be you're own e-mail address.

Note the period (.). This tells the SMTP gateway it's the end of the e-mail transmission.

If your syntax was correct your will receive a

250 Ok: queued as xxxxxxxx

If it wasn't correct you will receive an error. Just try typing it in again from the **mail from:** part.

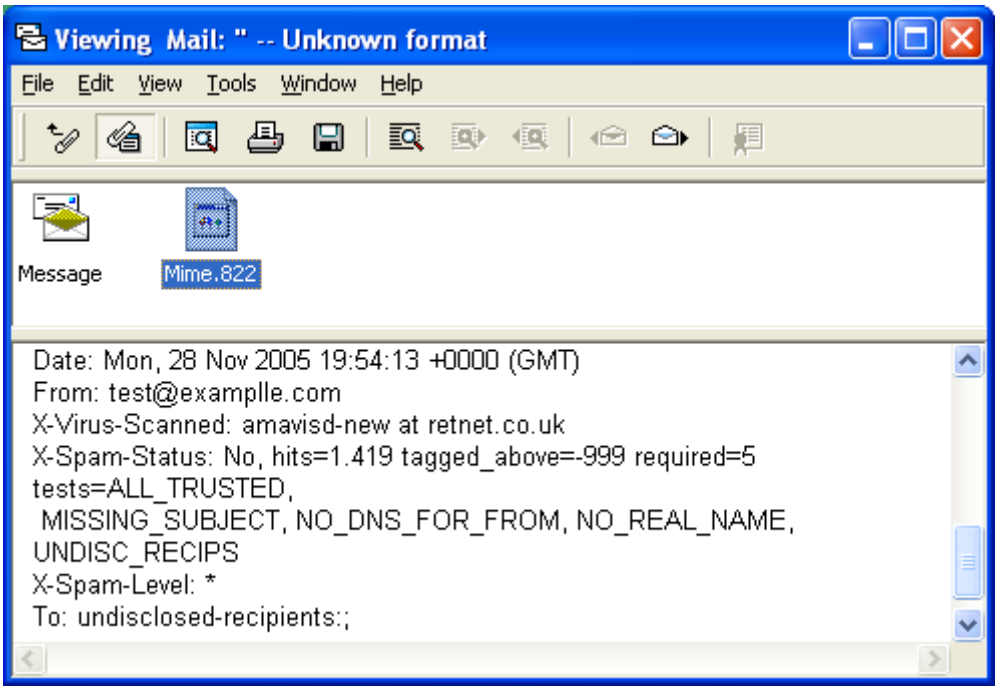
Typical SMTP error codes include:

421 Service not available, closing transmission channel (This may be a reply to any command if the service knows it must shut down)

450 Requested mail action not taken: mailbox unavailable (E.g., mailbox busy)
451 Requested action aborted: local error in processing
452 Requested action not taken: insufficient system storage
500 Syntax error, command unrecognized (This may include errors such as command line too long)
501 Syntax error in parameters or arguments
502 Command not implemented
503 Bad sequence of commands
504 Command parameter not implemented
550 Requested action not taken: mailbox unavailable (E.g., mailbox not found, no access)
551 User not local; please try
552 Requested mail action aborted: exceeded storage allocation
553 Requested action not taken: mailbox name not allowed (E.g., mailbox syntax incorrect)
554 Transaction failed

Check your regular e-mail account and you should have now received an empty e-mail from test@example.com.

To make sure Maia ran both virus and spam scanning, you need to view the header of the e-mails, which in almost all e-mail clients is hidden by default.



If you use Microsoft Outlook, with the e-mail open, select the **View** menu then **Options...**

In GroupWise 6.5, select the e-mail (don't open it) then select the **Action** menu then **View**.

Other clients will vary. In any case you should see some **X-** lines like above in the picture.

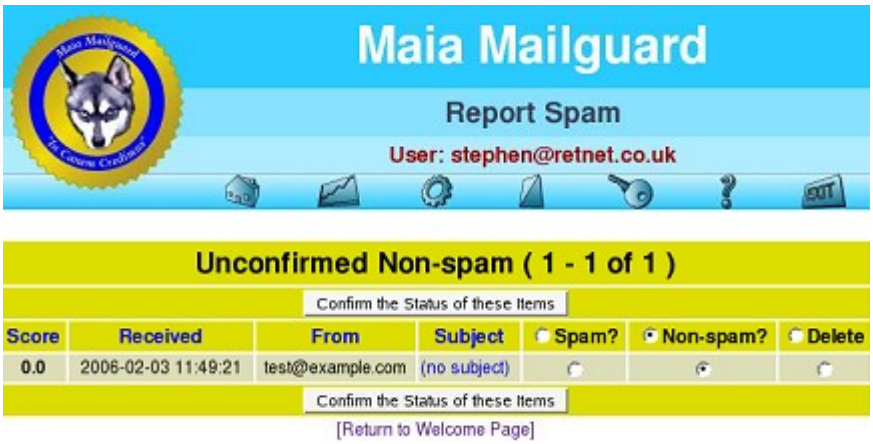
These lines tell you the e-mail was scanned for viruses as well as spam, and the corresponding spam tests that the e-mail 'matched' against, triggering a score. Scoring is an extremely complex subject, suffice to say it's almost always right.

If you want to read up on how scores are deduced, you should read the information available at the websites of Maia Mailguard, amavisd-new and SpamAssassin as they all play a part. Be sure to read them in that order though, as some settings in amavisd are controlled by Maia Mailguard, and some settings in SA are controlled by amavisd and Maia Mailguard, so it can get confusing if you read about

different scoring systems the other way around, as they don't mention the next layered application that may have an effect.

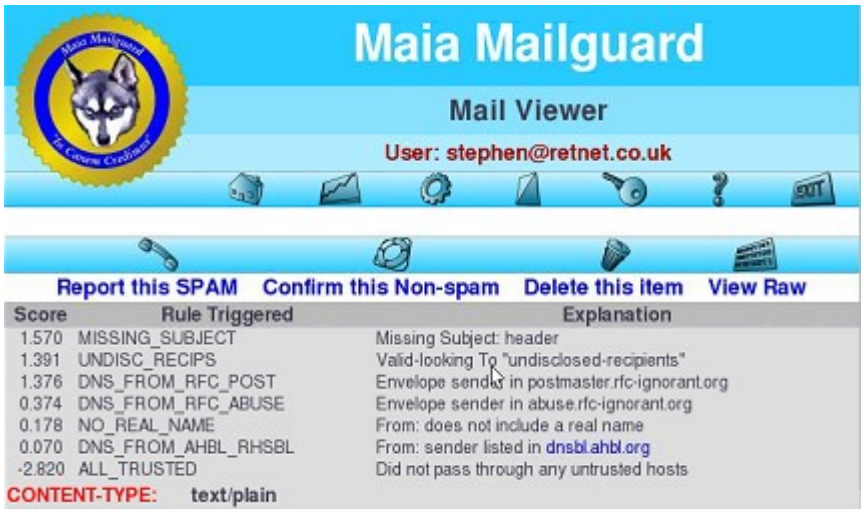
An alternative to viewing the results is to use Maia Mailguards built-in message viewer.

Note the superuser always has read rights to all e-mail, but other per-domain administrators can be denied this access, and if you configure individual users, they won't have access to anyone else's e-mail cache.



Click on the **home** icon in the menu and you will see in the Cache Contents box **You have 1 items in your non-spam cache**. Click the **Report/Confirm** link to take you into the non-spam cache.

You should see the test e-mail, with it's associated score. By default you would normally just click the **Confirm the Status of these Items** button, but this time around you will want to check the e-mail headers, so click on the Hyperlink for the subject, in this case **(no subject)**.



Here you can see what tests were triggered against this e-mail and the scores assigned to them.

Click the **Confirm this Non-spam** link to confirm that e-mail as clean.

Now to perform some other basic virus and spam tests....

Using Mozilla Thunderbird e-mail client for testing

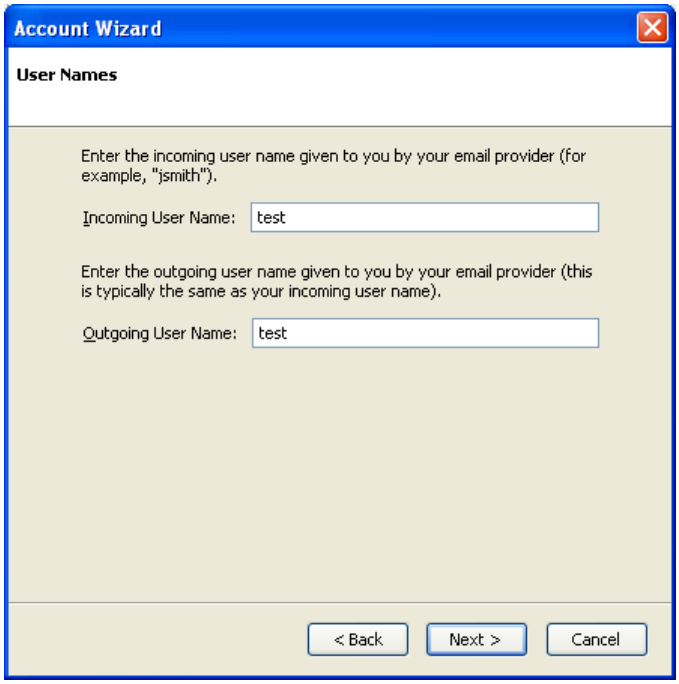
You have now used telnet, which if you haven't used before for SMTP testing, is good to know for quick troubleshooting in many situations.

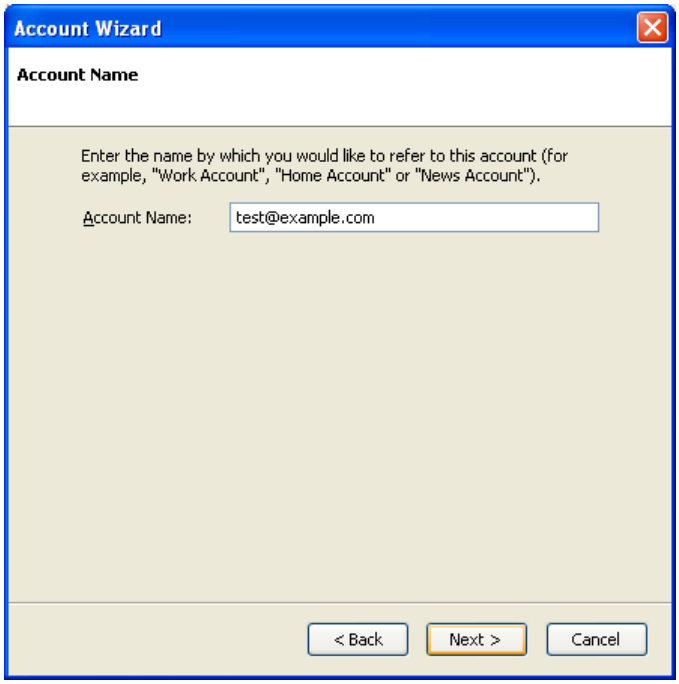
To make things easier you'll now move onto using a real mail client from your own PC, not the e-mail gateway, as it's quicker and less error prone. You can use any client that supports it's outbound destination as an SMTP host, such as Microsoft Outlook Express, KMail, Evolution, Mozilla Thunderbird or many others that are available.

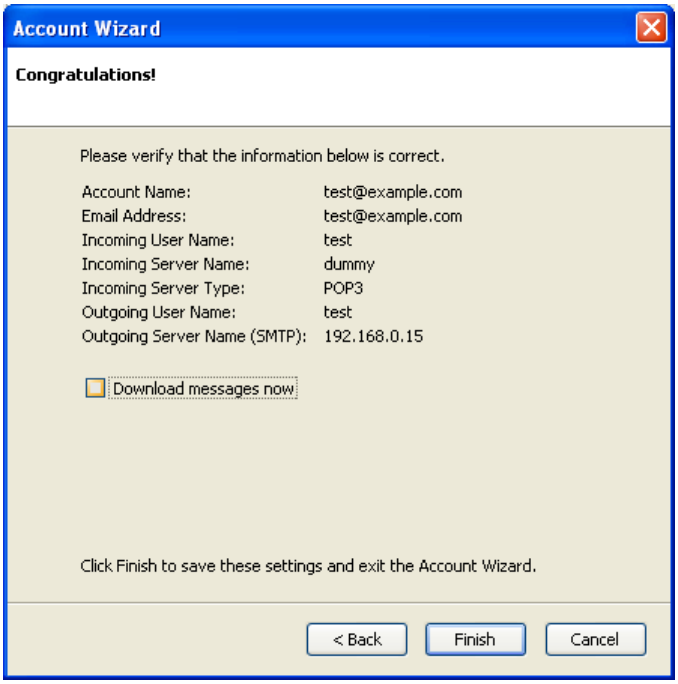
For ease of this guide, I'm going to describe the use of Mozilla Thunderbird, as it runs on Windows, Mac OS X and Linux i686 (PII or AMD K6 and above) systems. I'm not going to describe the installation of this client, as it varies between platforms, suffice to say you can go to http://www.mozilla.org and download it, then run the install, which is quite easy.

When setting up Thunderbird for your tests you don't want to use your real name or e-mail address, so I'll assume you've setup Thunderbird as the user **Test User** with an e-mail address of **test@example.com**. You will also need to enter a name of an incoming mail server. Just type in anything to get past that setup screen, and ignore any errors.

To make it easier I've included screen shots of what each setup screen should look like. These may be slightly out of date by now, but the idea is the same. If you do not see exactly the same screen shots, once setup, have a look in your account settings for the equivalent setting and change accordingly.

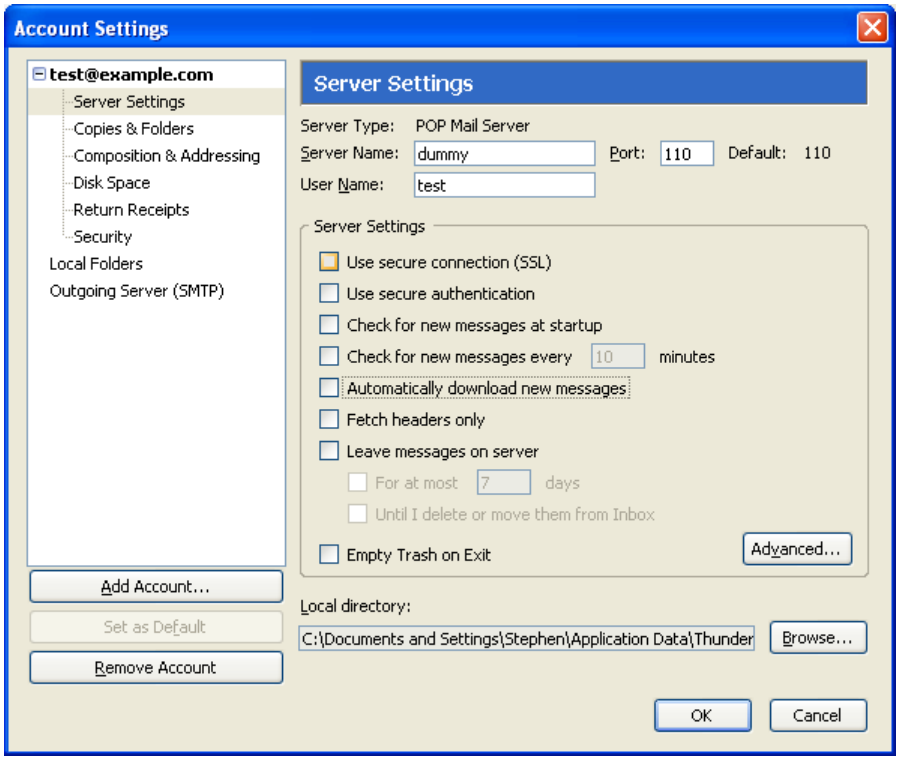


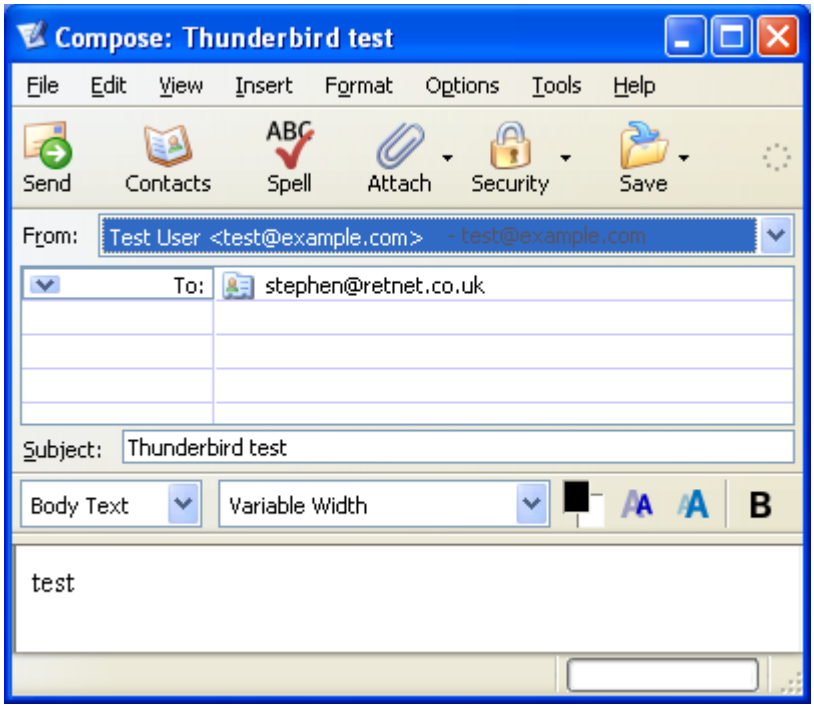




Remove the tick from the **Download messages now** option on the last page and click Finish.
If you see an error regarding 'failed to connect to' just ignore it. This is Thunderbird trying to connect to your mail server or check for incoming messages, which doesn't exist. To stop all the errors regarding incoming mail server not being found, edit the account properties from the **Tools – Account settings...** menu option .

Click the **Server Settings** tab and clear all options on that page then click the **OK** button.





To confirm the Thunderbird client is setup correctly, send a simple test message through to your regular e-mail account.

Now check your regular e-mail account to confirm it came through okay. If not, check the /var/log/mail file for errors. If there was a problem sending the e-mail from Thunderbird, the client will tell you about it.

Testing virus and spam filtering

Now that's out of the way, there's a great simple test for spam catching called the GTUBE (Generic Test for Unsolicited Bulk Email) test.

This test is simply a special text string that you insert into the body of an e-mail, and if the spam software is designed to catch it, which SpamAssassin is, Maia Mailguard will correctly quarantine it.

There is also a similar test for the virus checker called the Eicar Test Virus, which again is a special text

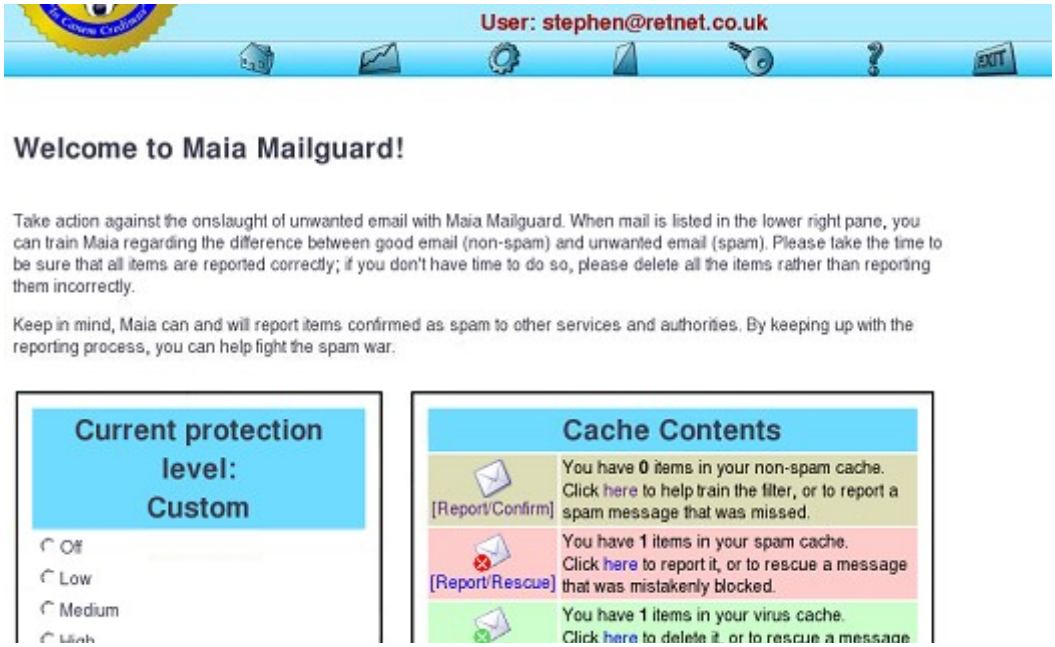
string inserted into the body of an e-mail.

For the spam test string go to http://spamassassin.apache.org/gtube and copy the text from there**.**

(I originally included the test text here but some web scanners blocked the download of this document for containing the test itself!)

For the virus test string go to http://www.eicar.org/anti_virus_test_file.htm and copy the text string from there.

So create 2 new e-mails, one with each test string and send them through, then go back to your **Welcome** page to check the results.



Click the **Report/Rescue** link to the **spam cache** and you'll see the score it got... that is about as high as you're ever going to see an e-mail get triggered.

Click the message subject link and you'll see why it scored that high.

To confirm this e-mail as spam, click the **phone** icon 

Now return to the **Welcome** screen then follow through the virus link.

You can see what virus it was recognised as, and read the e-mail by clicking the subject link. Maia will not execute any code within an e-mail whatsoever, so it's quite safe to do this.

If the virus scanner got it wrong, and it was in fact a legitimate clean e-mail, you can re-classify the e-mail as **non-spam** by clicking in that radio button then clicking the **Confirm the Status of these Items** button. In this case, it was right, so don't change its status and click the **Confirm** button.

Virus scanning is partly controlled by Maia Mailguard (Enabled/Disabled and an action), amavisd-new and ClamAV. These details will be covered further in the **Tweaks and Tightening** section.

Testing attachment filtering

Attachment blocking is mostly on a site-wide basis. You can define rules with lists of users that different attachment blocking sets are designed for, but for the meantime we will assume you want to block at least some attachment types site-wide.

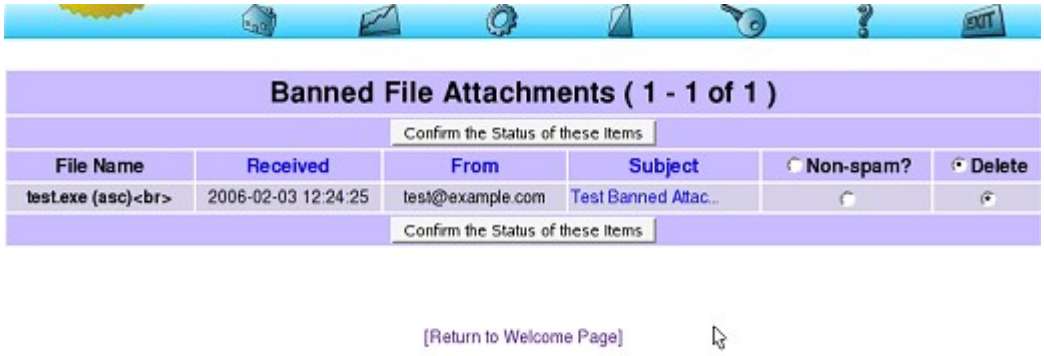
Head over to the amavisd-new website to see how to implement custom lists.

By default, when attachment filtering is enabled, the following file types will be blocked:

Any file with an extension of: **exe, vbs, pif, scr, bat, cmd, com or cpl**
Any file with an extension anywhere in the name (such as with double extension type files like program.exe.txt): **exe, vbs, pif, scr, bat, cmd, com, cpl or dll**
MIME type attachments of: **x-msdownload, x-msdos-programs or hta**
File types of the type: **exe-ms** (even if someone renames a .exe, it will still be caught by this)

Using Mozilla Thunderbird or whatever e-mail client supports sending to an SMTP host as it's outbound destination, send a test message through the gateway with with one of the banned extension types, such as an executable.

Once sent, go to the Maia Mailguard interface, then click on the banned files quarantine and you will now see the test e-mail.



When you click the **Banned File Attachment** link, you will notice it tells you the file name that was banned, along with it's type in brackets. In this case I simply created a text file and called it a .exe, and **amavisd** is clever enough to tell you it's an ASCII type file, even though it's got a different extension.

To enable attachment filtering for all users, click the **key** icon in the Maia Mailguard menu, select **Domains**, then your external domain, and enable the setting in there. To ensure it is automatically enabled for any new domains you configure, repeat the process for the **System default (@.)** domain as well.

Testing bad header filtering

Header filtering checks that the SMTP header of an e-mail conforms to strict Internet standards. As you left it Disabled before, when initially configuring you user accounts' settings in Maia Mailguard, you will now have to Enable it.

Open your personal e-mail account settings, by clicking the **cog** icon in the Maia Mailguard menu then selecting your **Primary address**.

Change the **Bad Header Filtering** option to Enabled, and ensure the **Mail with bad headers should be...** is set to **Quarantined**.

Now as you did earlier, use telnet to send through a simple e-mail. Because the headers won't be strictly correct, you should see this e-mail get caught by this filter.

The telnet commands will look something like this:

```
telnet 192.168.0.15 25
mail from: <test@example.com>
rcpt to: <stephen@retnet.co.uk>
data
.
quit
```

Obviously change the IP with the address of your gateway and the e-mail address you send the test to.

If your syntax was correct your will receive a

250 Ok: queued as xxxxxxxx

In Maia Mailguard, check your regular e-mail account **spam quarantine** and you should have now received an empty e-mail from test@example.com.

Click on the link to the quarantined item to have a closer look

Normally you would just confirm these e-mails and let Maia delete them. Notice there is no 'Spam or non-spam' options. This is because the bad headers check happens before the other checks, and as soon as amavisd catches an e-mail it stops processing any further which saves scanning resources and classifies it accordingly.

You can click the **Confirm the Status of these Items** now to clear that e-mail out of the system.

To enable bad header filtering for all users, click the **key** icon in the Maia Mailguard menu, select **Domains**, then your external domain, and enable the setting in there. To ensure it is automatically enabled for any new domains you configure, repeat the process for the **System default (@.)** domain as well.

Testing oversized e-mails

Before jumping right into this, you will need to understand a little about how this gateway and it's components independently deal with e-mail sizes.

Working backwards...

The Maia Mailguard **Mail size limit** option is located in the **System Configuration** section under the **Administration Menu** (Key icon) and it's default maximum size in bytes is set to 1Mb. This determines the largest e-mail that will be scanned for content filtering at all. Very large e-mails are CPU intensive, so you can choose to save some of that load and not scan e-mails over this size.

The Maia Mailguard **Oversized items should be...** option controls whether the oversized e-mail should simply be delivered anyway, or rejected. There is no quarantine for oversized items.

Note that as these settings are based in the **System Configuration** they are global, and there are no domain or user specific settings related to e-mail sizes.

Your MySQL server must also have the setting **max_allowed_packet** in **/etc/my.cnf** set to AT LEAST the same value as the Maia Mailguard **Mail size limit**. This setting relates to the single largest record that MySQL can store. Because Maia Mailguard will not break up an e-mail, when it is quarantined MySQL must be able to store the entire e-mail in one go. If the max_allowed_packet is smaller than the Mail size limit, Maia will try and fail to quarantine e-mail over the max_allowed_packet size... a situation you will never want.

PHP, the programming language that Maia Mailguard is primarily built on also has a built in memory limit, which by default is set to 8Mb. If this setting is too small the e-mail will be cached but you won't be able to release it, if caught by one of the other filters such as bad headers or attachment filtering.

You will need to edit the PHP configuration file **/etc/php.ini** and set the option **memory_limit** to a lot more than the Mail size limit you've set in Maia Mailguard. From my testing I found I needed to set it to

90Mb to successfully view/release a 10Mb e-mail. If anyone knows why this is the case please let me know. If you choose a larger e-mail size you will need to play with this memory setting to find the minimum needed to view your largest cached e-mail.

For this change to take effect you will need to re-start the apache web server by opening a terminal console on the gateway as root, then running **apache2ctl restart**.

Amavisd uses SpamAssassin to scan for spam, and by default it has an internal limit of 200Kb (although SA's default is 250Kb). Any e-mail larger than the amavisd limit is simply skipped for spam scanning. This is done to save time and resources (running 600+ tests would take it's toll on *every* e-mail) and it's extremely unlikely that a spam e-mail will be over this size due to the associated costs in terms of bandwidth and time to spammers to send out e-mails much larger than this. So the defaults are good to use but if you want to change it, on the gateway edit the file **/etc/amavisd.conf** and increase the option **$sa_mail_body_size_limit** to your needs.

If you change this setting you will need to re-start amavisd by opening a terminal console as root on the gateway and running **/etc/init.d/amavis restart**

Also note that amavisd needs enough physical RAM to cache an entire e-mail, as it stores the e-mail there for quicker scanning. By default amavisd runs with 2 child processes so you will need enough free RAM to potentially hold 2x your largest e-mails along with all the other things running on your gateway (hence the recommendation for plenty of RAM at the very start of this guide).

Lastly postfix has it's own default message size limit of 10,240,000 bytes which is roughly 10Mb (depending on how you calculate it.. some people use 10,000,000 10^6 as 10Mb, while others use 10,240,000 1,024x1000 as 10Mb. Yes it's confusing... check out http://en.wikipedia.org/wiki/Megabyte for information) . This should generally be set to a smaller limit than Maia Mailguard, else Maia could choke trying to deal with messages that are too large. To change this setting, on the gateway edit the file **/etc/sysconfig/postfix** and change the option **POSTFIX_ADD_MESSAGE_SIZE_LIMIT** to a number in bytes the same as or smaller than the same setting is Maia Mailguard. This effects inbound and outbound e-mail, so if you need to be able to send larger e-mails than you receive, your easiest option is to set our e-mail server to send directly out to the Internet or configure another gateway for outbound messages only. You can configure postfix to use a separate path for outbound e-mail to bypass this limitation – check the postfix website for details on setting up custom master processes in master.cf.

For this change to take effect you will need to open a terminal console on the gateway as root and run **SuSEconfig** to update the live postfix configuration file, then reload postfix by running **postfix reload**.

So at the end of this, there's a simple question you have to ask yourself. Is it okay to automatically accept and pass e-mails that are too large for my gateway to scan and manage?

If the answer is no, which it almost always will be, then you'll configure all the settings to similar values, but if the answer is yes, then there will be less load on Maia Mailguard in having to handle those e-mails, and you can keep Maia Mailgaurd, PHP and MySQL configurations at (what I consider to be) reasonable levels.

So lets move onto an example.

Lets say you want the gateway to handle e-mails up to 10Mb in size, and anything else should be rejected:

Your end e-mail server will of course also need to accept e-mails of this size for it to work. All the gateway components you've installed base their sizes on the converted MIME format of an e-mail which

can be up to around 30% larger, so you need to take that into account.

For a small bit of mail format info, all e-mail sent via SMTP is sent in plain text. This means any binary attachments must be converted to a format in plain text before they can be sent. MIME is a popular format that all SMTP gateways understand, and using it's algorithm on a binary file will produce an ASCII file of around 30% larger than it's original size.

I suggest you make your actual settings a little more liberal than any given directive, to save yourself and boss the grief of arguments with users over how large an e-mail really is... applying a small buffer by accepting slightly larger e-mail allows you to easily be sure that any e-mail that's rejected due to size was well beyond your company policy.

Taking a 10Mb e-mail as an example, you will need to account for the conversion of this e-mail, based on the fact it is most likely a binary file, into MIME format which will cause the size of the actual e-mail to increase to around 13.5MB in size, or 13,500,000 bytes. For this example we'll set the limits at 14,000,000 bytes for good measure.

On the gateway server, open the postfix configuration file **/etc/sysconfig/postfix** using Gedit

Scroll down to the option (near the bottom of the file) **POSTFIX_ADD_MESSGE_SIZE_LIMIT** and change the value to read **14000000**

Save and close the file.

Open the PHP configuration file **/etc/php.ini** using Gedit

Scroll down to (or search for) the option **memory_limit** and change it to read 9**0M**

If this setting is too low the Maia Mailguard e-mail viewer and PHP will generate errors on the server with a '**allowed memory size of xxxx bytes exhausted'**.

Save and close the file.

Open the mysql configuration file **/etc/my.cnf** using Gedit

Scroll down to the option **max_allowed_packet** and change the setting to read **15M** being just a little larger than required, for breathing space... you **really** don't want this to go wrong.

Save the file and close Gedit.

Now to enable these changes, on the gateway open a terminal console and run the following commands:

```
SuSEconfig
postfix reload
```

SuSEconfig will update the /etc/postfix/main.cf file with your changes, and postfix reload will tell postfix to re-read it's configuration to pick up the changes

now run

```
apache2ctl restart
```

This tells the apache2 web server to re-start which will re-read the php settings.

and

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

This re-starts mysql which will cause it to pick up the change you made.

Lastly change the Maia Mailguard setting

Go to the **System Configuration** page via the **Administration menu** and change the **Mail size limit** to read **14000000**

Scroll to the bottom of the page and click the **Update Settings** button.

Now to test it.

To do this accurately you need to create files of exact size, then attach these to your test e-mails.

Running Windows you can use a freeware utility called **File Generator** which you can download from http://www.soft32.com/download_76964.html

Running Linux, you can use a utility that's already installed on your computer, called dd.

To use dd to make a 10MB test file, open a terminal console and run

```
cd ~
dd if=/dev/zero of=./test50 bs=1000000 count=10
```

To explain the above, running **cd ~** will ensure you are placed in your home directory.

In the dd command, **if** is the input file, in this case a special device called zero which is actually an area of memory containing all zeros, **of** is the output file name, **bs** is the byte size which in this case is 100,000 bytes, and lastly **count** is the number of times to repeat **bs** which in this case is 10, which gives us our 10MB attachment.

In Mozilla Thunderbird, create a test e-mail and attach the 10MB file you made, then send it.

You can watch the processing of the e-mail on the gateway by running

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

This e-mail will not be scanned for spam, as it's too large, so the only header information you will see in the e-mail that's delivered is that it's been virus scanned.

To test the gateway for rejection is quite easy... just send a larger attachment through. From your e-mail client you will see it never gets there. You will receive an error about the size limit.

Normally it won't be an e-mail client talking to the gateway but instead another e-mail server. In those cases, the other e-mail server will throw out the e-mail and send the original sender a message saying the e-mail was too large.

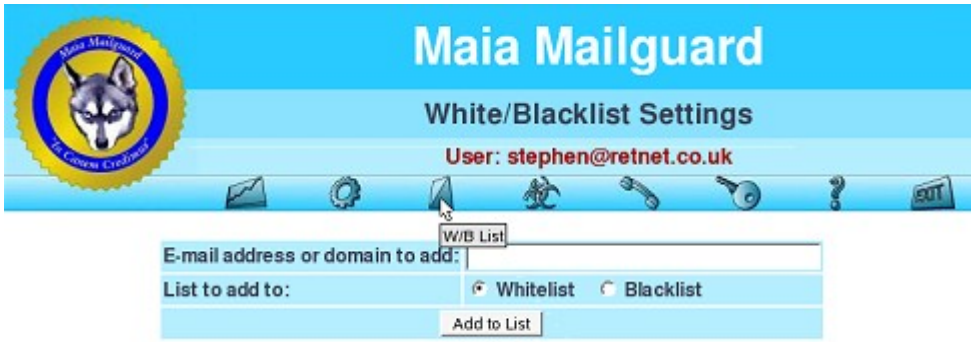
Note that because it is postfix that blocks the oversized e-mails, you will never see anything listed in the oversized statistics of Maia Mailguard for these, unless you set the postfix limit higher, but this would just waste bandwidth and gateway processing time.

The only way this rule will be triggered in Maia Mailguard is if Maia's setting is smaller than that of postfix i.e. postfix accepts the message and passes it to Maia which then rejects/passes based on your **System Configuration** settings.

Testing Whitelists/Blacklists

Lastly you'll test Whitelists & Blacklists.

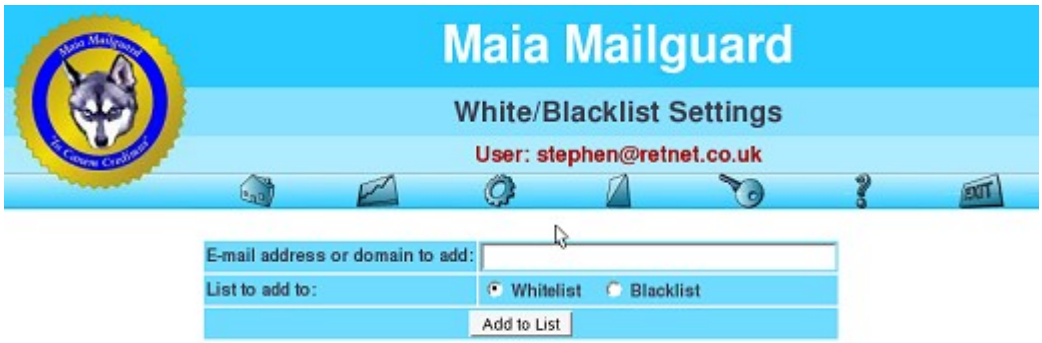
These should be used sparingly, and only if e-mail from the same sender is constantly being caught (or not) as spam when in fact it isn't.



In Maia if you click the 3rd icon from the left, something like a **box** shape, you will be in you own personal **White/Blacklist Settings** from where you can add users to your list. Whitelist users are those who you trust unconditionally to never send you spam. Blacklist users are those who seemed determined to fill your inbox with spam, but in reality this usually ends up being a block list for commercial marketing (although your users should be marking these as junk themselves in their e-mail client.

You can also setup a whitelist or blacklist for an entire domain or the system default.

Click the **key** icon then the **Users** link, click the **Find Users** button then click the **@retnet.co.uk** domain.



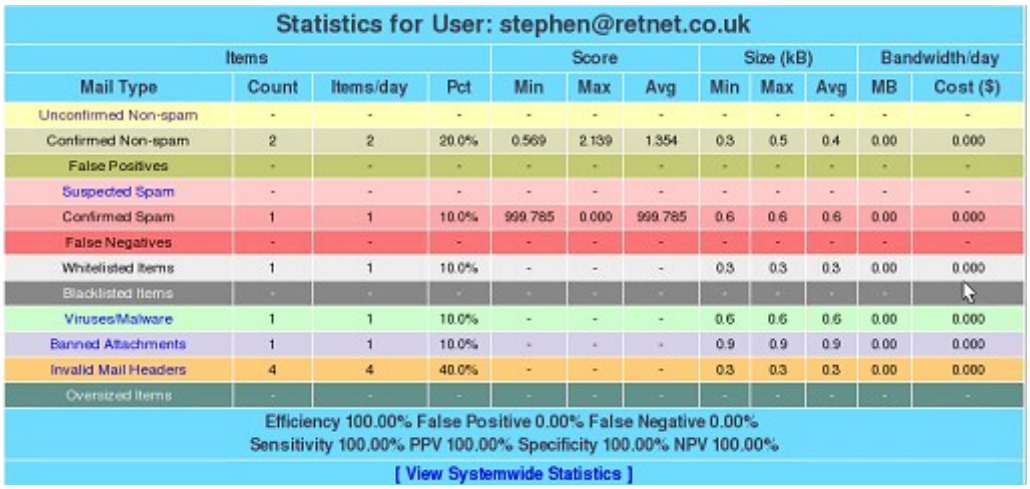
Now when you click the the **Whitelist/Blacklist Settings** button you will notice just under the heading it says **Default User for Domain @ retnet.co.uk** which tells you that you're now changing the defaults for the entire domain and not just yourself.

So let's test it.

While still in the domain **White/Blacklist Settings** page, add your test user **test@example.com** to your **Whitelist**.

Now using Mozilla Thunderbird, send an e-mail to your account, with the message body being the spam GTUBE code.

To see what's now happened in Maia, go back to the **Admin** menu by clicking the **key** icon, select the **Users** link, then click the **Find Users** button and select your own named account. Doing this will switch you back to your **Welcome** page.



Click the **stats** icon now will now show 1 **Whitelisted Item**. Note this e-mail, although it passed, is not cached in the non-spam cache like regular e-mails. The reason is you've already told Maia to ignore all checks and to assume it is clean.

Conversely, e-mail from blacklisted senders are never cached either and are always silently discarded,

apart from the counter in the statistics page.

If you as an administrator, set up e-mail senders as White or Black listed and give end users the right to change their personal lists, be aware they can override your defaults by adding that sender to their own white or black list.

One last note on White/Black lists is something called the **Automatic White List** (AWL for short) that's used by SpamAssassin by default. This isn't a white/black list as in Maia Mailguard, but a single list with scores against sender e-mail addresses that's automatically managed by SA.

The purpose of this list is to bias e-mails as they come from known senders, so if for example someone has sent you 50 clean e-mails and they then send something that appears to SpamAssassin as spam, due to their history of sending clean e-mail, SA will reduce the spam score of the e-mail based on their score in the AWL. SpamAssassin will then add a few points to the users' score in the AWL towards 'spaminess'.

On the flip side it also means that repeat offenders of spam quickly get caught out regardless of how they've changed the e-mail message contents, due to their history of sending rubbish e-mails.

To read a more details description of the AWL, point your web browser to this SpamAssassin info page:

http://wiki.apache.org/spamassassin/AutoWhitelist

Well that's just about done. You should also run the maintenance scripts manually as the vscan user to ensure they work without a problem. Output from these scripts will mostly appear in the terminal so any problems should be easily noticed. Be sure to check the digests that are sent include the correct url link (try the link out to make sure!).

To clean up, you should tune down the **$log_level** setting in amavisd.conf to **2** instead of **5** then restart amavisd by running **/etc/init.d/amavis restart**, as too much information just creates a lot of noise that's hard to listen to if you're trying to troubleshoot a problem in the log files later on.

You should also remove the login shell setting for the **vscan** user that you created near the start of this guide. To do this, start **Yast**, select **Security and Users** then **Edit and create users**. Change the filter to **System users** edit the **vscan** user, select the **Details** tab and change the **Login shell** option back to it's original value of /bin/false. Click **Accept** then **Finish** to save the changes.

You will probably also need to make 1 more change from the following section, labeled **Permit e-mail from trusted networks** which specifies what internal hosts, other than the gateway itself, are allowed to send e-mails to foreign domains... I'm going to take a wild guess here and suggest you probably do send out external e-mail, so you'll at least want to read that bit to see if it applies to you.

There's also quite a few really handy tips in tightening up your gateway in the following section, many of which work right at the front gate – the postfix SMTP connector, which will reduce the amount of spam before it even enters your network. It's more intense with no screen shots (you're probably glad of that by now) and if you've followed me this far from the start, I know you'll be able to cope quite easily with it.

Backing it up

The best way to avoid a hard drive failure is to have a RAID system in place, such as a mirrored or RAID 5 array. There is simply no substitute. I have a step-by-step guide on my website on how to create a bootable software RAID 1 array, which includes monitoring your RAID and using S.M.A.R.T. based monitoring to pick up disks that are on their way out.

I'm not going to spell out exactly how to back up the relevant configuration as there are simply too many ways to do this and most people have their own preference so instead I will point you to what needs to be backed up and how that can be achieved.

For most of the system configuration files, simply copy the **/etc** directory and all subdirectories.

For the website, copy the **/srv/www** directory and all subdirectories.

For the vscan user, copy the **/var/spool/amavis** directory and all subdirectories.

For MySQL, use the **mysqldump** tool to export all databases into a backup.sql file and copy that file.

Optional: Copy the root home directory at **/root** which includes all software you downloaded during the build.

To backup all of these files I personally use rsync, which is a YaST package. You can read more about rsync at http://rsync.samba.org/

You could alternatively use tar to create a simple backup file of all the above then copy that single file using cron via ftp, rsync, scp or any other method you can think of (all the the mentioned utilities are available via YaST).

Deciding the method that best suits your infrastructure then finding a solution to match is the best way of approaching this, and once you know what you want to do it will just take a little Internet searching to find all the answers you need.

Tweaks and Tightening

The system already described in this article will drastically reduce the amount of spam and viruses getting to your e-mail server, and this section show you how to tighten down the hatches even further.

System Updates

If your system is running well, unless there are security updates for the specific components used you could do worse than leave it alone.

If you do want to keep it up to date, start an Online Update from within YaST and follow the prompts. You can also configure fully automatic updates within this application.

To upgrade any additional software such as DCC, Razor2 or Maia Mailguard, you will need to carefully read the upgrade notes and test the update before going live.

Stopping spam and viruses before they even get in

These settings are additional checks performed by the postfix SMTP connector, that are performed at different stages throughout the e-mail conversation it has with the outside world.

The file you will edit for this set of restrictions is **/etc/sysconfig/postfix**, which is the YaST template that creates the postfix configuration file **/etc/postfix/main.cf**

The set of restrictions listed here will follow a typical SMTP conversation, the first of which is the sender identifying itself.

Reject if the sender doesn't identify itself

The sender should identify itself using an EHLO or HELO command and if it doesn't then there's a good chance it's a poorly written virus or script. All commercial and well known free e-mail servers and clients will tell you who they are before sending you e-mail.

To deny any sender that doesn't identify itself, add the following line to the very end of the **/etc/sysconfig/postfix** file:

POSTFIX_ADD_SMTPD_HELO_REQUIRED=”yes”

To test this change, save the file, in a terminal run **SuSEconfig** then **postfix reload** and try to start an SMTP conversation by running

telnet mailscan 25
mail from: <test@example.com>

Right there you will see a **503** error, and the sender will get no further until the connection times out or it stops trying to send the e-mail.

All of the sender restrictions are also applied to internal senders, i.e. Your e-mail server, but don't worry, the next 'restriction' allows all internal e-mail out, even if it's not properly formatted.

Permit e-mail from trusted networks

This is implied if no sender restrictions are in place, but as you are about to put some in, you will need to implicitly make this statement.

It references a self-calculated list called **mynetworks** that contains "trusted" SMTP clients that have more privileges than strangers. Restrictions are tested in order and the first to match will be applied. If none match, 'allow' is implied.

It is required in order to allow relaying from hosts other than the gateway itself onto external domains, which if you point your current e-mail server to this gateway, it will need doing.

To make this change, which you must do else you won't be able to relay e-mail out of your network via this gateway, add the following line to the very end of the **/etc/sysconfig/postfix** file:

POSTFIX_ADD_SMTPD_SENDER_RESTRICTIONS=”permit_mynetworks”

By default postfix will use the subnet of your gateway as a trusted subnet, so any other host on the same subnet will be “trusted”. If your e-mail server is actually on another subnet, you will need to specify this explicitly.

If you do need to do this, add the following line to the very end of the **/etc/sysconfig/postfix** file:

POSTFIX_ADD_MYNETWORKS=”192.168.0.0/24”

where 192.168.0.0 is the subnet you want to trust, and /24 is the number of bits in the subnet mask (in this case that will be 255.255.255.0). To trust only a single host such as your internal e-mail server, change it to something like 192.168.0.100/32. If you want to trust multiple hosts and/or subnets, separate them with commas.

To test this change, save the file, in a terminal run **SuSEconfig** then **postfix reload.**

On a computer in an 'untrusted' network, start an SMTP conversation by running

telnet mailscan 25
mail from: <test@example.com>
rcpt to: <stephenintheuk@yahoo.co.uk>

Substitute **stephenintheuk@yahoo.co.uk** for your own external e-mail address.

Right there you will see a **554** error, with postfix not willing to relay that e-mail. Now repeat the test from a computer on a 'trusted' network and it will work (as long as your MYNETWORKS statement is correct).

Reject non fully qualified sender's address

Have you ever received an e-mail from joe@yourcompany.com when you know that Joe doesn't exist in your company? This is a simple trick some malicious programs play to try to gain your confidence in

opening the e-mail or running an infected attachment. They send the e-mail as simply <joe> instead of <joe@company.com>. Your e-mail server will usually be default fill in the @company.com with your own company domain, thus appearing as a legitimate local e-mail.

To deny any sender that doesn't specify a full e-mail address in the MAIL FROM command, add the following line to the very end of the **/etc/sysconfig/postfix** file:

POSTFIX_ADD_SMTPD_SENDER_RESTRICTIONS=”reject_non_fqdn_sender”

If you already have an option listed, separate them with whitepace meaning any space, tab or new line (enter) so it could look something like:

POSTFIX_ADD_SMTPD_SENDER_RESTRICTIONS=”
permit_mynetworks
reject_non_fqdn_sender”

To test this change, save the file, in a terminal run **SuSEconfig** then **postfix reload** and try to start an SMTP conversation by running

telnet mailscan 25
mail from: <test>
rcpt to: <stephen@retnet.co.uk>

Substitute **stephen@retnet.co.uk** for your own e-mail address.

Right there you will see a **504** error, with postfix not accepting that user.

Reject from unknown sender domains

The assumption here is all incoming Internet e-mail should have originated from a real, existing Internet domain, so if postfix can't verify that the domain of the sender as specified in the MAIL FROM address exists, then reject the e-mail.

For this restriction, add the following line to the very end of the **/etc/sysconfig/postfix** file:

POSTFIX_ADD_SMTPD_SENDER_RESTRICTIONS=”reject_unknown_sender_domain”

If you already have the **sender_restrictions** line in your postfix file, just add additional parameters to it separated by commas and blank lines so that it looks like:

POSTFIX_ADD_SMTPD_SENDER_RESTRICTIONS=”
permit_mynetworks
reject_non_fqdn_sender
reject_unknown_sender_domain”

To test this change, save the file, in a terminal run **SuSEconfig** then **postfix reload** and try to start an SMTP conversation by running

telnet mailscan 25
mail from: <test@dummy.domain>
rcpt to: <stephen@retnet.co.uk>

Substitute **stephen@retnet.co.uk** for your own e-mail address.

Right there you will see a **450** error, with postfix not accepting the sender's domain.

The next set of restrictions relates to the recipients.

Permit e-mail from trusted networks

For the same reasons as the sender restrictions, allow anything from trusted hosts or networks to go through.

POSTFIX_ADD_SMTPD_RECIPIENT_RESTRICTIONS=”permit_mynetworks”

Reject non fully qualified recipients addresses

All e-mail coming into your company should be destined to user@yourcompany.com and never to just user. Spammers can use this method to get around you trying to hide your domain name from the Internet. They simply connect to your e-mail server and start firing off e-mails to recipients without specifying your domain so they don't even have to know it!

To enforce that senders send e-mail to fully qualified e-mail addresses, add the following line to the very end of the **/etc/sysconfig/postfix**

POSTFIX_ADD_SMTPD_RECIPIENT_RESTRICTIONS=”reject_non_fqdn_recipient”

Again if you already the first setting in place, separate the options with whitepace.

To test this change, save the file, in a terminal run **SuSEconfig** then **postfix reload** and try to start an SMTP conversation by running

telnet mailscan 25
mail from: <test@example.com>
rcpt to: <stephen>

Substitute **stephen** for your own e-mail name (don't include the domain though).

Right there you will see a **504** error, with postfix not accepting that recipient.

Reject unauthorised destination domains

This check confirms that the domain in the RCPT TO command matches your domain name.

For this restriction, add the following line to the very end of the **/etc/sysconfig/postfix**

POSTFIX_ADD_SMTPD_RECIPIENT_RESTRICTIONS=”reject_unauth_destination”

To test this change, save the file, in a terminal run **SuSEconfig** then **postfix reload** and try to start an SMTP conversation by running

telnet mailscan 25
mail from: <test@example.com>
rcpt to: <stephen@dummy.domain>

Substitute the **stephen** part for your own e-mail name (don't include the domain though).

Right there you will see a **554** error, with postfix not accepting the recipient domain.

Reject unverified recipients

This is a pearl of a setting, where postfix will probe your e-mail server to see if the recipient actually exists on your internal e-mail server, before accepting the e-mail from the sender.

This setting therefore prevents undeliverable junk mail from ever entering your e-mail system and wasting resources on processing the e-mail or generating bounce replies that end up sitting on the gateway for days.

Postfix does this by starting a normal SMTP conversation with your e-mail server, and assumes if it receives from your server a **250 OK** response to the RCPT TO address, that the user exists. This probe is only performed once then the result is cached. If the recipient exists it will stay cached for 7 days before requiring a refresh. If the recipient does not exist that result will stay cached for 3 hours, so if someone was just trying to send a new starter some e-mail who didn't have an account yet, it wouldn't be long before the address was re-checked and any subsequent e-mails to that recipient would be allowed through.

For this restriction, add the following line to the very end of the **/etc/sysconfig/postfix**

POSTFIX_ADD_SMTPD_RECIPIENT_RESTRICTIONS=”reject_unverified_recipient”

If you already have the **recipient_restrictions** line in your postfix file, just add additional parameters to it separated by commas and blank lines so that it looks like:

POSTFIX_ADD_SMTPD_RECIPIENT_RESTRICTIONS=”
permit_mynetworks
reject_non_fqdn_recipient
reject_unauth_destination
reject_unverified_recipient”

Also note that Postfix say this is only a good option for low traffic sites... but I believe their idea of low traffic is something like under several hundred thousand of messages per day, which I believe should be fine for most of you out there. Another thing to take heart at is if you did not do this check the gateway would attempt to send the e-mail onto your e-mail server anyway.

To test this change, save the file, in a terminal run **SuSEconfig** then **postfix reload** and try to start an SMTP conversation by running

telnet mailscan 25
mail from: <test@example.com>

rcpt to: <dummyuser@retnet.co.uk>

Substitute **@retnet.co.uk** for your own domain name (leave the username as it is, unless you actually have a user called that.... in which case change it to a user that doesn't exist).

Right there you will see a **450** error, with postfix not accepting the recipient's name.

Tighter control over attachment filtering

File attachment filtering is performed by Amavisd, and this is one area that Maia Mailguard only supports being enabled or not and what to do when they are caught.

If you want to change the types of files caught from the default set, then you will need to modify **/etc/amavisd.conf**

Scroll down to the **$banned_filename** section then edit to your heart's desire.

Be sure to re-start Amavisd by running **/etc/init.d/amavis restart** in a terminal console.

Blocking e-mail delivery to local users of the gateway

Because this is a gateway, no one should ever be trying to send e-mail to locally setup users on the box itself. By getting rid of local e-mail delivery to the gateway makes it harder to break.

In **/etc/sysconfig/postfix** add to the end of the file

POSTFIX_ADD_MYDESTINATION = ””
POSTFIX_ADD_LOCAL_RECIPIENT_MAPS = ””
POSTFIX_ADD_LOCAL_TRANSPORT = error: local mail delivery is disabled

Save that file, now open **/etc/postfix/master.cf** and comment out the local delivery agent line, to look like

#local unix - n n - - local

Save that file, close Gedit and in a terminal console run

SuSEconfig
postfix reload

Setting a lower spam thershold

SpamAssassin has very good default settings, but still errs on the side of caution. Some specially crafted spam messages are designed to come in just under the default score of 5.000. After a few weeks I personally set mine set to 4.000 and have found this very reliable with not a single legitimate e-mail being caught yet, after it learned my initial e-mailing habits.

Because Maia doesn't automatically reject spam, but instead quarantines it, you can be safe in tweaking the spam catch score setting in Maia.

To do this, in Maia mailguard edit the options **Consider mail 'Spam' when Score is >=** and **Quarantine Spam when Score is >=** in your System Default domain (.@), local domain and your own personal e-mail account settings.

Remember in the current stable RC5 release of Maia Mailguard that the quarantine level will always be matched to the spam level. There is no distinction between these 2 values in this version.

BCC copies of e-mail to another address

Useful for security or HR/IT investigations, these options will add another address to the e-mail as a BCC so the original sender and recipient has no idea it's being done.

Edit **/etc/sysconfig/postfix** and add them like this:

 # Send a copy of every e-mail generated to another address
POSTFIX_ADD_ALWAYS_BCC = security@mydomain.com

 # Send a copy of every e-mail matching the sender
POSTFIX_ADD_SENDER_BCC_MAPS = 'hash:/etc/postfix/sender_bcc_maps'

 # Send a copy of every e-mail matching the recipient
POSTFIX_ADD_RECIPIENT_BCC_MAPS = 'hash:/etc/postfix/recipient_bcc_maps'

The sender_bcc_maps and recipient bcc_maps use a lookup file to match an address then send to it's corresponding bcc'd address. Note that neither address needs to be on your local domain, so you can match an external sender or recipient address.

You will need to create the files yourself as they do not exist by default, and don't include the .db extension.

An example would be if you wanted to receive a copy of all e-mail sent from john.doe@retnet.co.uk .

Create the file **/etc/postfix/sender_bcc_maps** and in it put something like:

 # This lookup table is used to match sender addresses then send a bcc'd copy
 # to a matching address.

john.joe@retnet.co.uk **stephen@retnet.co.uk**

Once you've created the file, save it then run

postmap /etc/postfix/sender_bcc_maps

which will create a hashed index table version of the same file and give it a /db extension.

Now edit **/etc/postfix/master.cf** and add the sender_bcc_maps option as described above

To finish off, reload postfix with the new changes by running

postfix reload

To read more about these or any other setting, visit http://www.postfix.org and search for those terms on the front page to find the relevant documentation.

Increasing scanning throughput

One way is to increase the number of scanning threads in amavisd-new.

To do this, increase the **$max_servers** setting in **/etc/amavisd.conf** and restart amavisd.

Also you will need to change the postfix **maxproc** setting in **/etc/postfix/master.cf** on the smtpd line that uses amavisd.

```
# ==========================================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)
# ==========================================================================
smtp      inet  n       -       n       -       2       smtpd -o
content_filter=smtp:[127.0.0.1]:10024
```

In the above example, maxproc is set to 2. This setting tells postfix how many connections it can use to talk to amavisd. If the **maxproc** setting is smaller than the **$max_servers** setting in amavisd.conf you're just wasting resources as postfix will never send more than 2 e-mails at a time to amavisd and amavisd will always have many unused threads, so to maximise efficiency and performance, make sure both settings match.

After the change to **master.cf**, restart postfix with a **postfix reload**

Another option is to put the SpamAssassin Bayesian database and AutoWhiteList into a MySQL database instead of using the default Berkley DB format. The process is quite simple by following instructions on https://secure.renaissoft.com/maia/wiki/SpamAssassin3SQLBayes

A third, but slightly more risky option is to setup a RAM drive and mount it as the temporary storage area used by amavisd-new (**/var/spool/amavis/tmp**). Because there can be a lot of disk spooling going on, especially with 15 or more amavisd-new processes running, doing this on higher loaded sites can show a huge performance increase, but requires plenty of extra memory.

I've never set one of these up, so if you're interested hop on over to the SuSE support forums at http://support.novell.com/forums/ for some helpful advice.

Basic Troubleshooting

It's all in the log files....

Every problem from e-mail delivery due to incorrect settings or network problems to troubleshooting a particular spam score can always be found in the logs.

The single most used part of all troubleshooting is the postfix mail log, located at **/var/log/mail**.

You can search the file easily using a basic command like:

cat /var/log/mail | grep user@example.com

to find all matching lines with user@example.com in them.

This log file also has information on scores applied by ClamAV, Freshclam (database updater) messages and amavisd.

Debugging

You can also turn up debugging in the following:

Amavisd-new:

- Increase the **$log_level=** setting in **/etc/amavisd.conf** to 5, re-start the amavis service by running **/etc/init.d/amavis restart** then check your **/var/log/mail** log file.

- Stop the amavisd-new service by running **/etc/init.d/amavis stop**

- Run amavisd manually using **su – vscan -c amavisd debug**

  This puts amavisd into full debugging mode and you'll see lots of info on stdout (your terminal screen)

- To debug exactly what SpamAssassin is doing, and the associated scores it assigns to an e-mail, run amavis using **su – vscan -c amavisd debug-sa**

Postfix:

- First backup your **/etc/postfix/main.cf** file by running
  cp /etc/postfix/main.cf /etc/postfix/maincf.orig

- Now edit main.cf and add the line **debug_peer_list = some.domain**
  where some.domain is the domain you want to debug.
  This can also be an IP address or some other pattern.

Mailq and qshape tools

Postfix has 2 great tools, one called **mailq** and another called **qshape**.

The **mailq** utility will display details of e-mails in the active or deferred queues, including which directory it's in and the sender's e-mail address. Just run **mailq** in a terminal and check the results that are displayed.

The **qshape** utility displays a table sorted by the destination domain with the largest number of e-mails and shows how long and how many e-mails are sitting in your mail queues (by default the active and deferred queues).

Finding this great utility in SuSE actually takes a little more digging as it's not in any of the 'standard' directories so I suggest you create a link to it in one of the more normal directories in Linux, by running:
ln /usr/share/doc/packages/postfix/auxiliary/qshape/qshape.pl /usr/sbin/qshape

Just run **qshape** in a terminal and check out the results.

To read more about mailq and qshape, visit http://www.postfix.org and search for those terms on the front page to find the relevant documentation.

SpamAssassin and the ALL_TRUSTED score being incorrectly added

SpamAssassin will automatically try to work out if the Received headers in an e-mail were added by a mail server on your network or not. Sometimes it gets this wrong, so you may see some spam e-mails that have an ALL_TRUSTED negative score against them, or worse, spam that has made it through because the negative score reduced the total e-mail score below your thresholds.

To overcome this issue, you can manually tell SpamAssassin what hosts and/or networks can be trusted.

To do this, edit your **/etc/mail/spamassassin/local.cf** file and add the following option:

trusted_networks 192.168.0.1 192.168.1.0/24

The above example will trust the host 192.168.0.1 and all hosts on the C-class network of 192.168.1.0.

Defining this setting ensures that no other hosts will be trusted and you will not get all all_trusted score for any e-mail coming in from any other hosts or networks.

You can visit this URL for more reading regarding this setting:

http://wiki.apache.org/spamassassin/TrustPath

Websites for more help

Also check out the websites for the software you're having a problem with, most of which have searchable archives of their mailing list as well as FAQ's and their own full documentation.

Links are back at the start of this document.

Well I hope even if you didn't follow it to the letter, that you have a slightly better understanding of this setup which is fairly common or at least you've been able to glean something of use from this.

Oh, and don't forget you can always e-mail me!