Spam and Virus Scanning with Exim 4

using the Exim Content Scanning Extension and/or SA-Exim

Mini-HOWTO

©2003-2006 Tim Jackson (tim@timj.co.uk)
V1.0.12 (12-Nov-2006)

Table of Contents

1. Disclaimer/Copyright/Distribution

1.1. No Warranty

Whilst this document is offered in good faith and at no charge in the hope that it will be useful, the author provides NO WARRANTY WHATSOEVER as to the accuracy or otherwise of any of its contents. Use of any information provided in this document is entirely at your own risk.

1.2. Copyright

1.3. Licensing/Distribution Permissions

2. Introduction - what's all this?

This document is mostly a “quick start” guide which briefly describes how to install and configure the following items of software, so that they work together to provide real-time spam and virus scanning:

- Exim 4.x (this version refers to 4.50+ but most parts are applicable to earlier versions of Exim 4)
- The Exim Content Scanning Extension, formerly known as “Exiscan-ACL” (very old implementations also called “Exiscan”, though these used a different methodology)
- SA-Exim (this document refers to version 4.x but is applicable to 3.x too)
- SpamAssassin (this document refers to version 3.x, though most is applicable to 2.x)
- Clam Antivirus/Sophos Antivirus/sophie

In here I hope to summarise much of the information already available from various sources on the Web, therefore helping those new to using these tools to understand the options available, and successfully get a basic configuration up and running. It is *not* a comprehensive reference, though, and is definitely not a substitute for reading the documentation for the individual pieces of software.

The instructions should be moderately generic, though they include some extra packaging/installation information for those using GNU/Linux and Red Hat Enterprise/Fedora-based systems (and derivatives thereof, e.g. CentOS). If you're using something else (e.g. deb packaging), you will need to refer to other sources for detailed packaging and installation advice, although the general configuration details given here should remain similar or the same. Details on how to acquire the various items of software are provided in-line.

In all cases, if you're having trouble with something, read the software documentation thoroughly, understand how the instructions given there relate to what you've done, and troubleshoot based on that. If that fails, there are mailing lists available for the various pieces of software:

- Exim: http://www.exim.org/mailman/listinfo/exim-users
- SA-Exim: http://lists.merlins.org/lists/listinfo/sa-exim
- SpamAssassin: http://spamassassin.org/lists.html
- Clam Antivirus: http://www.clamav.net/ml.html
- Sophie: http://www.vanja.com/list/listinfo.cgi/vtools

This document assumes you are using Exim 4. If you are still using Exim 3, you should upgrade; Exim 3 has been obsolete and unsupported for over 4 years. The instructions in this document will not work on Exim 3.

Important Note: This document assumes a basic familiarity with Exim and SMTP. If you are new to Exim, you are strongly recommended to install Exim on its own, and become familiar with how it works, before attempting the projects described in this document.

3. Mail Content Scanning Concepts and Options

3.1. Introduction

In these days of ever-increasing junk e-mail, it's increasingly necessary to employ a variety of methods to reduce the amount of unwanted 'spam'. There are many ways of doing this (DNS-based blocking lists, sender verification etc. - see more info at http://slett.net/spam-filtering-for-mx/) but this document looks at one particularly powerful and common requirement: to be able to make decisions on whether or not to accept particular messages based on their content. This does not by any means preclude or supersede the use of other solutions such as DNSBL's; in fact, you'll find the most effective spam-stopping solutions involve a combination of different methods. Since it is relatively "expensive" in terms of resource usage (CPU and bandwidth), content scanning should be a "fallback" - a check of last resort.

There are two primary ways of handling mail scanning:

a) Accept all messages, then apply some sort of filtering, and bounce any that you don't want
b) Scan incoming messages as they arrive, and reject at SMTP DATA time

These are explained and discussed below.

3.1.1. Accept all messages, then scan

This is the "old fashioned" way, that was commonly used a number of years ago. The normal route is to accept the mail and then set up some kind of router which passes the mail to an external scanner (typically SpamAssassin), then re-inject back into Exim. This has been covered in detail before by many people (including on the exim-users list), and details on how to set it up with SpamAssassin used to be found at the following URL, although as at November 2006 it seems to be dead:

http://dman.ddts.net/~dman/config_docs/exim4_spamassassin.html

The main problem with this method is what to do with a mail once it's classified as spam. You could:

- discard it (to /dev/null), but then there's no indication to the sender, which is not good if you get a "false positive" (i.e. legitimate mail is classed as spam).
- bounce it, but the sender is often forged, in which case the bounce will either bounce (and end up frozen in your queue), or go to some innocent bystander (so-called "collateral spam"). **Big problem – DO NOT DO THIS!**
- move it into a separate mailbox, but then someone has to check it, and the sender has no idea their mail might be delayed (unless you send a notification, but then see the above point about bouncing and it all gets rather clumsy).

Additionally, you may often end up scanning a particular mail multiple times - once for each recipient.

None of these are ideal, which is why many people choose the second method, which is to scan at SMTP time.

3.1.2. Scan incoming messages as they arrive (while the SMTP session is open)

This is a more advanced way of rejecting spam and viruses, the main focus of this document. What happens in this case is that after the remote server has connected to your Exim server and sent the message envelope and body, your server scans the complete mail and makes a decision on whether to accept or reject it *before* sending the final SMTP confirmation. If it chooses to reject the mail, it can therefore send an SMTP reject code (550) and the problem of what to do with the mail is left entirely to the originating server. Frequently, the "originating server" is not really a mail server at all, but some "spamware" software or perhaps an open proxy, in which case it is unlikely a bounce will be generated. So this method also helps to reduce "collateral spam". This is increasingly important; most common e-mail-borne viruses at the current time fake "From" headers of e-mails, implicating innocent people. Recent virus outbreaks have consequently resulted in blizzards of bounces and "you sent us a virus" warnings to completely innocent parties – it is every server administrator's responsibility to reduce the risk of sending "collateral spam" as much as possible.

This (scan at SMTP time) is the approach that will be documented here. Before you begin, you should be aware that this approach can be costly on resources, and so it isn't recommended in situations where CPU time (in particular) is at a premium. If you're handling high volumes of mail, you're likely to need hefty resources to implement a solution along these lines.

There is, however, one main inherent disadvantage of this method: where spam is concerned you will, largely, lose the ability to provide per-user options on filtering (blacklists/whitelists etc.), because each mail that comes in is scanned only once, and can only be rejected or accepted, but it might have a number of different recipients all on your server.

There are some 'hybrid' solutions/workarounds you could use to mitigate this limitation, such as:

a) all mails over a certain (high, e.g. 25) SpamAssassin “spam score” (see later) are discarded at SMTP time, but for any mails that are accepted, you then have customisable per-user filtering after that.

b) you conditionally scan mails based on one or more recipients having selected scanning (i.e. if one of the recipients wants scanning, everyone gets it whether they like it or not)

c) using the method outlined by Tom Kistner on http://www.exim.org/pipermail/exim-users/Week-of-Mon-20030317/051275.html whereby you allow only one recipient per mail (by temporarily rejecting second and subsequent recipients for a mail at SMTP RCPT time) but this is considered by some to be rather clumsy, and it will delay delivery to the second or subsequent recipients (because the originating server will delay for some period before retrying the mail with the other recipient(s)). A more powerful variant of this, as outlined by Alan Flavell, is to allow multiple recipients per SMTP session, but only if their scanning preferences match. See http://www.exim.org/pipermail/exim-users/Week-of-Mon-20031006/061151.html. There are also some examples on the Exim wiki at http://www.exim.org/eximwiki/ExiscanExamples

Either way, you could also use the Exim RCPT ACL to provide a certain degree of per-user-configurable black/whitelisting based on sender address or enable the use of DNSBL's selectively, but that is beyond the scope of this document.

4. What Software To Use?

You can consider that you'll need three types of software (other than the core Exim software itself) to do spam and virus scanning at SMTP time. You'll need:

1. Some kind of Exim patch/module to handle the mail during transmission and pass it on to the spam and virus scanning software
2. Spam scanning software
3. Virus scanning software

The options are discussed below.

4.1. Exim patch/module

As of Exim 4.50, there is an (optional) universal content scanning system integrated into the Exim distribution called the “Content Scanning Extension”. This is what was formerly called Exiscan ACL (http://duncanthrax.net/exiscan-acl/). This integrates virus, spam and other forms of scanning into Exim. It is elegant and consistent with the rest of the Exim configuration, because it uses the normal Exim ACL system. So you can just use this if you want, and this is probably the simplest solution. In any case we'll use this for virus scanning.

However, if you want more options and more detailed control of *spam* scanning (only), and/or wish to take advantage of SpamAssassin's report_safe option, you might prefer to use SA-Exim (http://marc.merlins.org/linux/exim/sa.html). In particular, SA-Exim lets you easily save rejected messages to a file, which you might find handy if you're jittery about rejecting 'possible spam' (this is also possible with the Content Scanning Extension, although marginally less simple to set up). SA-Exim is a “local_scan” module, meaning that it is configured separately to Exim and does not integrate at all with the Exim ACL system. Exiscan and SA-Exim can co-exist peacefully, so you can use the Content Scanning Extension for virus scanning and SA-Exim for spam scanning if you want.

It's suggested that you read the introductory information given in the respective documentation for the two pieces of software, to decide which you prefer for spam scanning. If in doubt, you can always install both, and then switch between them at leisure by simply adjusting your Exim configuration.

4.2. Spam scanning software

If you're going to be doing spam scanning, the most important part of the jigsaw is a piece of software which can scan message contents and make some kind of recommendation as to the 'spamminess' of the message. For the purposes of this document, we will use the most common software, SpamAssassin (http://www.spamassassin.org/). You should bear in mind, however, that there are competing solutions (bogofilter, spamprobe, crm114 etc.) which might be better suited to your needs. In that case, however, these instructions won't help you.

4.3. Virus scanning software

You've got a number of choices here. The Content Scanning Extension supports a number of virus scanners directly, and supports virtually any scanner (or multiple scanners) indirectly via it's generic command-line scanner configuration.

Some common choices, all of which provide a memory-resident daemon (this is good to ease resource load and generally speed things up) are:

- Clam Antivirus (Free software, widely used and highly regarded)
- Sophos Antivirus (commercial; either using the SAVI library in conjunction with a daemon such as 'sophie', or with the commandline-based 'sweep' scanner)
- Kaspersky Antivirus (commercial; has it's own daemon)

Clam AV (http://www.clamav.net/) is free (in both senses of the word) software and can provide a totally free virus scanning solution. Alternatively, you can download a fully-working version of Sophos (with limited trial

licence) from the Sophos website (http://www.sophos.com/), and a trial is also available for Kaspersky (http://www.kaspersky.com/).

Please see the Content Scanning Extension documentation for further information about your options when selecting a virus scanner.

This document details how to install either Clam Antivirus or Sophos Antivirus (with the 'sophie' daemon).

5. Installing the software

5.1. Installing SpamAssassin

You can download the SpamAssassin source from http://spamassassin.apache.org/. The latest stable version at the time of writing is 3.1.7 and you're advised to use that (and keep up to date with new releases). If you like living on the edge (and beating the latest spam techniques), you might like to try the latest CVS version.

5.1.1. Installing using RPMs

Many systems, including some Linux distributions such as Fedora, now include SpamAssassin. In this case, installation may be as simple as "yum install spamassassin" or similar. If not, you have a number of options:

- download a pre-built binary package from the links on the SpamAssassin website, if there is a suitable one available for your system
- use a third party RPM repository such as DAG (http://dag.wieers.com/), http://centos.karan.org/ etc.
- rebuild your own RPM, either using the spec file provided with SpamAssassin, or using a suitable base spec/source RPM package from a similar distribution or third party repository. For example, the Fedora Core RPM package normally rebuilds easily on recent versions of Red Hat Enterprise/CentOS. If you're unsure about how to build an RPM from source, see Appendix A.

If you use an "enterprise" distribution like CentOS where updates are generally confined to conservative bugfixes and security updates, you're in an interesting position: using the OS-supplied SpamAssassin version eases your update burden but means you aren't getting the latest software in a fast-moving area. However, branching out on your own by building your own packages or installing from scratch means taking on the burden of maintenance and updates yourself. A compromise might be to use a third party repository that you trust that provides more up to date versions. You still don't get the "support" of the base OS vendor, but at least you don't have to rebuild things yourself.

You'll also need some Perl modules that may not be installed by default. Please see the SpamAssassin INSTALL file for full details of the required modules, but these may (depending on your installation) include:

- Time::HiRes
- Digest::MD5
- MIME::Base64

Check your installation media/yum repositories to see if they are pre-packaged already. RPM-based systems should look for packages beginning with perl-XXX; e.g. "perl-Time-HiRes".

You may also need some others, such as:

- Digest::SHA1
- Digest::HMAC_MD5
- Test::Simple
- Net::DNS

You can either install these additional modules from CPAN using the 'normal' Perl method (perl -MCPAN...), or if (like me), you prefer to install them as RPMs and they're not available as part of your normal package options, you can use the 'cpan2rpm' utility from CPAN (available from Fedora Extras, for Fedora users). Other similar utilities (e.g. 'cpanflute2') are also available and will probably do the job equally well. If you're installing SpamAssassin RPMs via yum from a "complete" repository like Fedora Extras, all the required dependencies should normally be pulled in automatically.

5.1.2. Installing without RPMs

Follow the instructions in the SpamAssassin distribution.

5.1.3. Testing

In either case, assuming you've managed to get SpamAssassin installed, you should now follow the 'testing' instructions in the INSTALL file provided with SpamAssassin to test it, namely:

```
spamassassin -t < sample-nonspam.txt > nonspam.out
spamassassin -t < sample-spam.txt > spam.out
```

(The sample-spam and sample-nonspam files are provided as part of the SpamAssassin distribution, and can be found in the documentation directory of SA, typically /usr/share/doc/spamassassin-XXX/.)

With a bit of luck, all should be working. If not, check you've got all the necessary Perl modules installed (see the INSTALL file) as that's the main cause of problems.

Once it's working, you can move on to setting up the Content Scanning Extension and/or SA-Exim.

5.1.4. Security reminder

Before you start the daemon (we'll come to that later, after configuration), whichever way you installed SpamAssassin, ensure that your firewall is blocking port 783/tcp from the world at large. This is the port used by default by the SpamAssassin daemon to accept messages for processing and return results.

5.2. Installing the Content Scanning Extension

Skip this section if you're only doing spam scanning (not virus scanning), and you've chosen to use SA-Exim to handle things.

5.2.1. Introduction

The Content Scanning Extension was formerly a separate source-level patch called Exiscan, written by Tom Kistner. It is now fully integrated into the main Exim package, though compilation and use of it is still optional.

5.2.1.1. Note about older versions

You may be confused by references on the website and mailing lists (especially archives) to Exiscan and the Content Scanning Extension, because there are at least three different things that people may be referring to:

- 'Original' Exiscan – this was the only option until Exim 4.14. It uses custom Exim configuration options starting with exiscan_*. This is now long since deprecated, is not maintained for current Exim releases and should no longer be used.
- Exiscan-ACL – introduced publically with the release of Exim 4.20, this was an overhauled version of Exiscan which generalised much of the configuration to integrate neatly into the Exim ACL system. This is a very slick idea which provides maximum configuration flexibility and familiar rules (if you're used to Exim ACL's).
  Exiscan-ACL also sub-divides into two distinct categories:
  - release <= 14: just provides extra options in the DATA ACL, including the “demime” option
  - release >= 15: provides an entirely new ACL; acl_smtp_mime, which is run for each MIME part in a message. “demime” is now deprecated.
- The Content Scanning Extension: this is simply Exiscan-ACL integrated into the main Exim source from version 4.50 onwards. It is still maintained by Tom Kistner. Some people still informally refer to the Content Scanning Extension as 'Exiscan'.

Just be aware that discussions and other documentation you may find on the Internet might be referring to any one of these! This document now deals solely with the Content Scanning Extension (and implicitly Exiscan-ACL); the last version of this document which discusses 'original' Exiscan is v1.0.8.

For older versions (< 4.50) of Exim, you need a version of Exiscan-ACL designed specifically for the version of Exim that you're running, and you can download these from http://duncanthrax.net/exiscan-acl/ .

5.2.2. Installing as part of an Exim RPM

See section 5.6.

5.2.3. Installing without using RPMs

To use Exim with the Content Scanning Extension, you will need to compile it using the WITH_CONTENT_SCAN compile-time option. For more information please review the Exim Specification (Manual) which has fuller details.

For older versions of Exim, to install Exiscan-ACL, untar the source file, and use the “patch” command to patch the Exim source tree. You'll then need to recompile Exim. More details are in the Exiscan-ACL documentation.

5.3. Installing SA-Exim

Skip this section if you're not doing spam scanning, or you have chosen to use the Content Scanning Extension to handle your spam scanning.

SA-Exim uses Exim's local_scan interface to scan messages as they are received, which is what we want to do. You can download it at http://marc.merlins.org/linux/exim/sa.html.

SA-Exim can be used in two ways:

- As a simple local_scan.c replacement
- As a loadable module, using Derek 'dman' Hudson's dl_open patch for Exim

Please note: to avoid confusion, any references to a 'local_scan' config option you may see in mailing list postings imply that the 'dlopen' patch is installed. So, to reiterate, there are two distinct options:

Choice 1 (with dlopen patch – this is what many RPM packages that provide SA-Exim do)
- Compile Exim with dlopen patch
- Compile SA-Exim as .so
- Use the (non-standard, provided by dlopen patch) "local_scan_path" option in Exim config file to tell the dlopen patch which local_scan modules to load. For example:
  ```
  local_scan_path = /usr/libexec/exim/sa-exim.so
  ```

Choice 2 (without dlopen patch)
- Compile Exim without dlopen patch, but with sa-exim.c in the Exim source tree
- That's it. No config options required.

SA-Exim provides a file to replace the (dummy) src/local_scan.c in the default Exim source tree, so you'll need to replace this and recompile Exim. You'll also need sa-exim.h (do “make sa-exim.h” in the SA-Exim source directory). If you're installing Exim as an RPM, this is all done automatically.

5.4. Installing Clam Antivirus

These instructions apply to ClamAV 0.88.6, and have changed since earlier versions of this document and ClamAV.

5.4.1. RPM Installation

Many distributions, including Fedora (via Fedora Extras) and PLD, now include Clam Antivirus packages, so if

you're using those you can probably just go ahead and install the RPM or other package as supplied. For Fedora Extras or packages rebuilt from there, you need the following packages:

- clamav (this is the base package)
- clamav-data (the actual virus database)
  clamav-lib (scanning libraries)
- clamav-server (this provides the Clam AV daemon)
- clamav-update (this provides the Freshclam update tool)

If you're using another distribution, you may need to compile a fresh RPM, one way or another. I recommend using the ClamAV SRPM package from Fedora Extras as a base:

http://download.fedora.redhat.com/pub/fedora/linux/extras/development/SRPMS/

Versions of this recompiled for Red Hat Enterprise (and derived distributions such as CentOS) can be found at http://centos.karan.org/

To make things easier, you should also install a package which configures an instance of Clam AV to run with appropriate permissions for Exim to use. At the time of writing, a package for Fedora Extras is available as a sub-package of Exim by installing “exim-clamav”. Unfortunately this package does not currently create a separate UNIX user and instead runs as the same user that Exim runs as – this is not terrible, but a separate user would add some extra security. When you install exim-clamav, a daemonised ClamAV instance is configured to run as the “exim” user so that it can read the temporary files in /var/spool/exim/scan that are created during mail reception for scanning purposes.

5.4.2. Installing without an RPM

Follow the instructions given with your distribution or, if installing from source, in the Clam AV documentation. As part of the process, you should create an unprivileged user (e.g. 'clamexim') for the daemon to run as. Ideally this should be separate to the user that Exim runs as. If it is, then either:

- add it as an additional user in the UNIX user group that Exim runs as (e.g. 'exim') in /etc/group and enable the AllowSupplementaryGroups option in your ClamAV config file, or
- set its default group to be the same group that Exim runs as (e.g. 'exim')

Alternatively, you can run it as the same user that Exim runs as, but this gives the daemon additional privileges. Either way, make sure that the user can read the temporary files in /var/spool/exim/scan that are created during mail reception for scanning purposes.

5.5. Installing Sophos Antivirus/sophie

Sophos Antivirus itself consists primarily of three parts:

- Virus database
- Command-line scanner
- libsavi shared library

The virus database itself is of course the most important part and what you are (primarily) paying for. Beyond that, however, you have a choice: you can either use the command line scanner (“sweep”) directly, or you can use some third party software (e.g. a daemon), interfaced via the SAVI interface using the libsavi shared library.

In this documentation, I'll focus on using a third party daemon (sophie) with the libsavi library, since this is the most efficient way of scanning, though it's perfectly possible to use the command line scanner with Exiscan – see other documents for details.

The most common daemon used with Sophos is the 'sophie' daemon, which is Free software and available from http://www.clanfield.info/sophie/.

To install sophie without using RPMs, follow the instructions with the package. To install it as an RPM, read on.

5.5.1. Installing as RPM

To install sophie and Sophos Antivirus as RPMs, you can use my spec files found at http://www.timj.co.uk/linux/ . Please note that I do not use sophie myself any more, so these are old and not maintained.

5.5.2. Virus database updates

Either way, you may find the sophos-update script from http://englanders.cc/~jason/patches.php useful to auto-download updates to the antivirus database.

5.6. Compiling/installing Exim as an RPM

If you use an RPM-based system (e.g. Red Hat/Fedora/SuSE/PLD etc.), you may wish to compile/install Exim, together with any associated patches/modules (Content Scanning Extension, Exiscan and/or SA-Exim), as an RPM. To do so, there are a variety of choices. Many distributions provide good packages already. Up to date information about package availability for Fedora and Red Hat-based distributions (amongst others) is shown on the Exim Wiki at http://www.exim.org/eximwiki/ObtainingExim.

In the absence of a specific RPM for your distribution, I recommend using the Exim source RPM from Fedora Extras as a base and rebuilding this. You can find the latest package here:

http://download.fedora.redhat.com/pub/fedora/linux/extras/development/SRPMS/

To recompile from SRPM, it may be as simple as picking one of the SRPMs and doing “rpmbuild –rebuild exim-xxx.src.rpm”, or you may need to first install the SRPM, optionally modify the source files (e.g. the makefile or other build options; the spec file provides a number of choices) and then recompile with “rpmbuild -ba exim.spec”.

6. Configuring the software

Before you start, bear in mind that what I give here is very much a 'quick start' guide to important/common options, and is **not a substitute for reading the documentation** for the various pieces of software. The options provided are examples and the appropriate settings may vary significantly depending on your particular environment.

6.1. Configuring SpamAssassin

SpamAssassin itself works pretty well 'out of the box', although you'll need to configure the Content Scanning Extension/Exiscan/SA-Exim properly.

6.1.1. For Content Scanning Extension/Exiscan users

There is one option – report_safe - you'll probably need/want to change if you're using the Content Scanning Extension. You do this in the local config file, (typically /etc/mail/spamassassin/local.cf):

```
report_safe 0
```

(This option changes the message body of suspected spam by writing the SpamAssassin report in the body and attaching the original message as a MIME part. Some people like this, some people don't. It is supported by SA-Exim v3.1+ but not currently by the Content Scanning Extension/Exiscan as they don't allow modification of the message body by SpamAssassin. Therefore, this option has no effect but it's probably wise to disable it for performance and consistency.)

6.1.2. For SA-Exim users

You may want to make use of the “rewrite_header” option. This can be used to modify headers (typically the “Subject” header) of suspected spam, to aid in “flagging” it. Otherwise, the spam reports will only be found in special message headers (e.g. X-Spam-Flag, X-Spam-Status, X-Spam-Level, X-Spam-Report). In an ideal world this would be unnecessary, but unfortunately many common but useless clients such as Micros*** Lookout and Lookout Express can't filter on non-standard headers, so you may need to set this option to 1 if those users want to be able to filter 'possible spam' into a separate folder.

NOTE: this setting is ignored and you need to configure this separately if you're using Exiscan-ACL!

Also, SA-Exim adds a number of headers to mails that pass through it; you probably don't want SpamAssassin's Bayesian database to take any notice of these, so note this using the bayes_ignore_header functions.

```
bayes_ignore_header    X-SA-Exim-Mail-From
bayes_ignore_header    X-SA-Exim-Scanned
bayes_ignore_header    X-SA-Exim-Version
```

6.1.3. For all users

SpamAssassin in general is very configurable, so it's worth reading the manual to find out settings which you might want to customise for your site. However, one key setting to bear in mind is **required_hits** – default 5.0, this sets the number of points which a mail must score to be *marked* as spam. Note that this *isn't* the score at which mail will be rejected outright; that is set separately in SA-Exim or Exiscan, depending on which you are using.

You should also note that there are many more useful things you can put in the global configuration file, including black/white lists and your own rules.

As you will see later, SpamAssassin places various reports in the headers of scanned e-mails showing details of what rules, if any, were triggered. You can customise this (see SpamAssassin documentation) but I find the following useful:

```
clear_report_template
```

```
report "_YESNO_, hits=_HITS_ required=_REQD_ tests=_TESTS_ autolearn=_AUTOLEARN_
version=_VERSION_ _REPORT_"
```

Furthermore, if you want to take advantage of SpamAssassin's auto-whitelist and Bayesian auto-learning features (please read the SpamAssassin documentation for more information about these), you'll probably need to set up the directory where this is stored. Find out the UNIX user that Exim runs as (typically 'exim', with uid 93 if you installed on a Red Hat/Fedora system from RPM, otherwise possibly 'mail' or 'mailnull') and locate the home directory for that user (from /etc/passwd). This may be something like /var/spool/exim. Then, in the home directory, do:

```
mkdir .spamassassin
chown exim.exim .spamassassin
chmod 700 .spamassassin
```

Substitute the Exim user/group for “exim.exim” if that's not right. SpamAssassin's auto-whitelist database file will be stored in this directory once you start using it.

6.2. Configuring the Content Scanning Extension/Exiscan

You need to enter some options into your Exim config file (typically /etc/exim/exim.conf or /etc/exim/exim4.conf). This guide now only refers to the Content Scanning Extension/Exiscan-ACL; if you're using the 'original' Exiscan, either upgrade or read v1.0.8 of this guide.

6.2.1. Overview/setting up DATA ACL

The Content Scanning Extension provides additional functions in the DATA ACL, and provides an entirely new ACL (acl_smtp_mime) which is called for each MIME part in a message. Under normal circumstances, a copy of Exim build with the Content Scanning Extension will have a number of sample configurations defined in the default config file, so it's worth checking through that as well as reading the “Content scanning at ACL time” chapter in the Exim Specification. Otherwise, define an ACL to be used (I'll call it 'acl_check_data' here) in the main part of your Exim config:

```
acl_smtp_data = acl_check_data
```

and, down in the ACL section of your config file (somewhere after the “begin acl” line), start that ACL:

```
acl_check_data:
```

Additionally, since some content scanning may take place per-MIME-part in e-mails (the “MIME ACL”), you should probably start that one too:

```
acl_smtp_mime = acl_check_mime
```

(See notes below about functionality which is currently available either using the DATA ACL using “demime”, or the newer way using the MIME ACL)

Now, the beauty of the Content Scanning Extension/Exiscan-ACL is that you can customise the exact rules to suit your own circumstances in the same way that the normal Exim ACLs give you unparallelled flexibility. However, that means that there are an infinite number of possibilities for configuring it. I'll outline here some common rules to get you started, but you're going to need to make sure you understand how ACLs work if you want to go beyond these (and it's worth doing, because you'll appreciate the amazing flexibility of Exim in general!)

Some other references include:

- http://www.exim.org/eximwiki/ExiscanFilenameBlocking
- http://www.exim.org/eximwiki/ExiscanExamples

6.2.2. MIME checking settings

You can configure the Content Scanning Extension to check whether a message contains invalid MIME. It can give you a number indicating the severity of MIME errors in the message and you can use this to reject badly malformed messages (which usually indicates malicious content). For example, in the DATA ACL:

```
deny    message       = This message contains malformed MIME ($demime_reason).
        demime        = *
        condition     = ${if >{$demime_errorlevel}{2}}
```

This will reject messages with severe (>=level 2) MIME errors.
Note that the above functionality requires Exim to be compiled with the WITH_OLD_DEMIME option.

To replicate this functionality without using the old “demime” option, you have to write equivalent rules using normal ACLs and the variables that the Content Scanning Extension provides. I do not currently have sample configurations for this built, although a number of discussions have taken place on the exim-users list to this effect. If I can find a consistent set of rules I may include them (or a link) in a future version this document.

6.2.3. File attachment blocking settings

It's generally considered prudent to block certain types of file attachment which are rarely used 'for real' but are sometimes used as virus carriers, as a first line of defence. This is less effective than it used to be, but can occasionally stop new viruses or variants which haven't yet made it into your virus scanner's database. Such attachments are identified by their file extension and a simple subset to block might be .bat/.com/.exe/.pif/.scr/.vbs .

To set up the Content Scanning Extension to block the above extensions, there are two ways:

6.2.3.1. The 'old' way

The 'old' way (which is still supported, for backwards compatibility, if you compiled Exim with the WITH_OLD_DEMIME option) is to use the DATA ACL in conjunction with a rule similar to the following:

```
deny    message       = Mail contains blacklisted attachment (.$found_extension)
        demime        = bat:com:exe:pif:prf:scr:vbs
```

6.2.3.2. The 'new' way

With recent versions of the Content Scanning Extension/Exiscan ACL which include the new MIME ACL, some functions have been generalised. The MIME ACL is called repeatedly for each MIME part of a submitted message. In this, a number of variables are available (see the Exim Specification). The $mime_filename variable allows you to do any operation you want based on the sender-suggested filename of an attachment, and this includes rejecting based on the extension. For example, in the MIME ACL you could do something like this:

```
deny   message   = Mail contains blacklisted attachment ($mime_filename)
       condition = ${if match \
                        {${lc:$mime_filename}} \
                        {\N\.\s*(bat|com|exe|pif|prf|scr|vbs)\s*$\N}}
```

(the \s* fragments are to catch malware that attempts to confuse the user/user's software by inserting spaces into the filename)

6.2.4. Spam scanning settings

Skip this if you're not doing spam scanning, or are doing it with SA-Exim

First, you need to define your spam scanner's TCP socket location. Assuming it's on the local machine and

listening on port 783, add the following option in the *main part of the Exim config (not the DATA ACL!):*

```
spamd_address = 127.0.0.1 783
```

Back in the DATA ACL, there are a wide variety of options available for spam scanning. Let's assume you want the following:

- All mails (whether or not they are spam) should have an X-Spam-Score header , containing the SpamAssassin score for that mail
- All mails (whether or not they are spam) should have an X-Spam-Report header, containing SpamAssassin's report on the mail

Assuming that's the case, add your first spam ACL rules in the DATA ACL:

```
warn  message = X-Spam-Score: $spam_score
      spam = exim:true

warn  message = X-Spam-Report: $spam_report
      spam = exim:true
```

The “exim” in the above rules refers to the UNIX user whose SpamAssassin preferences will be used. We discussed earlier setting up the .spamassassin folder in the Exim user's home directory so that SpamAssassin can store Bayes and auto-whitelist entries there. Therefore, use the “exim” user (or whichever user Exim runs as) here.

If you want to add a “**** SPAM ****” marker to the start of subject lines if a mail contains spam (see the discussion above about rewrite_subject in the SpamAssassin configuration for more info) then add this ACL rule:

```
warn  message = Subject: **** SPAM **** $h_Subject
      spam = exim
```

A more detailed method (written and used by myself) that allows per-user configuration of Subject line tagging is described at http://www.exim.org/eximwiki/SpamTagSubjectHeaderPerUser.

Now, to do our SMTP-time rejection, we need to add a 'deny' ACL rule. Decide the score at which you wish to reject and multiply it by 10 to find the rejection threshold. Assuming you choose 10 (a reasonable starting value), you should add a rule similar to the one below: (the '100' is the rejection threshold as calculated above)

```
deny   message = This message scored $spam_score spam points.
       spam = exim:true
       condition = ${if >{$spam_score_int}{100}}
```

6.2.5. Virus scanning settings

Skip this if you're not doing virus scanning

You need to configure Exim to pass messages to your virus scanner. First, in the *main part of the Exim config (not the DATA ACL!)* define your virus scanner using the av_scanner option. The Exim manual details how to use this, but here are common settings:

For Clam Antivirus:
```
av_scanner = clamd:/var/run/clamd.exim/clamd.sock
```
(This scans using clamd's UNIX socket, which is the way recommended by the ClamAV team and configured by the recommended ClamAV RPMs. Adjust the path to the socket if the above doesn't match yours. Otherwise, you could use TCP sockets [which can be handy for debugging] – configure it in your clamd config file and then use: “av_scanner = 127.0.0.1 3310” or equivalent in the Exim config).

For Sophos with sophie:
```
av_scanner = sophie:/tmp/sophie
```

Now you need to set up an ACL rule to do the virus scanning. Like checking for “bad extensions”, you can do this either in the MIME ACL or the DATA ACL.

6.2.5.1. In the DATA ACL

Something such as the following rule should work well:

```
deny  message  = This message contains a virus or other harmful content ($malware_name)
      malware  = *
```

6.2.6. Ending the ACL

Finally, close the acl_check_data and/or acl_check_mime ACL(s) with a final 'accept' line:

```
accept
```

so that if mails don't contain viruses/spam, they will be accepted. Don't forget this, or you will end up rejecting everything!

6.3. Configuring SA-Exim

SA-Exim is a highly configurable piece of software, and has an exceptionally well-commented config file. The exact options that you want will very much depend on your personal preferences and environment. However, I'll give you some pointers here towards options you may typically want to look at.

6.3.1. The SA-Exim Config File

The SA-Exim config file can typically be found residing at /etc/exim/sa-exim.conf

The most important 'basic' options are:

- SAEximRunCond – equivalent to Exiscan's ACL 'condition', this determines when and if SA-Exim will scan a message. You could just set it to '1' to scan all messages, but there are some circumstances when you might not want to – see below.
- SAEximRejCond – this determines whether mails are actually rejected if they are found to be spam. You don't really want to reject messages addressed to postmaster@yourdomain/abuse@yourdomain, so the default configuration for this option checks some flags which can be set in an Exim ACL to avoid this – see below.
- SApermreject – this is the 'threshold' spam score. If a message scores higher than this, it will be rejected. You might be a bit nervous about this setting, so if you like you might set it higher and slowly reduce it. 12.0 (the default) is unlikely to catch any real mail, though, and in fact you can probably safely pull it down to 11.0 or even less without rejecting real mail. Experiment!
- SApermrejectSavCond – this determines whether messages that have been rejected as spam are saved anywhere. If you want to save them, set this to 1 (or use some kind of condition) and then set SApermrejectsave to the location where you want to save messages.
- SAmsgpermrej – this is the message used to reject spam with if it exceeds the threshold. If a real mail is inadvertently bounced as spam, the sende r should see this message in the bounce which they get.

6.3.2. Setting Exim ACLs

To avoid rejecting messages to postmaster@/abuse@, you should add the following ACL part to Exim's RCPT ACL (in the Exim config file), somewhere near the top:

```
warn    message       = X-SA-Do-Not-Rej: Yes
        local_parts   = postmaster:abuse
```

This will add a header “X-SA-Do-Not-Rej: Yes” to any messages addressed to postmaster@/abuse@, and

then (according to the default SA-Exim config), suppress rejection of spam to these addresses. Messages that are classed as spam will still receive the spam headers, however, which is a good compromise – the mail can still be filtered into a separate folder in your (or your users') mail client, but in the event that it's actually a genuine mail (typically an abuse complaint), you will have the opportunity to review this folder and read it.

6.4. Configuring Clam Antivirus

If you use the recommended RPM packaging as supplied, things should be set up more or less correctly by default and ClamAV is ready to go. You just need to set up virus database updates, so that you get the latest virus definitions regularly.

6.4.1. Virus database updates

ClamAV includes 'freshclam', a utility (which can be run either standalone or as a daemon) to automatically check for and download virus database updates. If you installed as an RPM, this should set it up to start and monitor for updates. However, before first use you may need to:

- edit /etc/sysconfig/freshclam and comment out the line:

  FRESHCLAM_DELAY=disabled-warn

  which prevents network access by default (so that you have to explicitly enable it).
- comment out the “Example” line in /etc/freshclam.conf, and set a local database mirror using the DatabaseMirror option. The comments in the config file are largely self explanatory.

For an RPM-installed package, the regular updates are kicked off via an entry in /etc/cron.d/clamav-update.

6.4.2. Non-RPM installation

You'll probably need/want to change quite a few of the default settings for Clam Antivirus. First, just to check, make sure you followed the instructions in the earlier section about installing Clam Antivirus, particularly with respect to making sure that clamd is running as an unprivileged user that can access /var/spool/exim/scan.

For reference, the key configuration options to check are outlined below:

1. Commented out the line which has “Example” on it. (Without doing this, ClamAV won't start)
2. Set the LogFile to /var/log/clamav/clamd.log
3. LogVerbose and LogTime enabled. You'll find these useful if trying to track down problems.
4. Set PidFile /var/run/clamav/clamd.pid
5. Set “DataDirectory /usr/share/clamav” as this is where the RPM normally installs the virus database files. Check, if you're not installing from RPM.
6. Enabled the StreamSaveToDisk and “StreamMaxLength 10M” options
7. Set the “MaxThreads” option to 10
8. Enable the “User clamav” option – better not to run Clam Antivirus as root.
9. **Enable the “AllowSupplementaryGroups” option, to give ClamAV the required access to the temporary scan files created by Exiscan.** (This is a crucial step and is often forgotten)
10. Enable the 'ScanMail' option. This enables the MIME functions in ClamAV.

7. Getting it all running/Testing

Assuming you've followed this document through, you should now be ready to start your chosen combination of software and get things working.

7.1. Starting the daemons

7.1.1. SpamAssassin

On many systems, including Red Hat-alike ones where you install SA as an RPM:

- Execute “service spamassassin start”. You should then have a process called 'spamd' running, hopefully with the '-d' and '-a' options enabled.
- Execute “chkconfig spamassassin on” just to make sure it will start on reboot.

otherwise:

- Execute “spamd -d -a” and arrange for this to happen on bootup.

7.1.2. Clam Antivirus

If you installed as an RPM:

- Execute “service clamd.exim start”. You should then have a process called “clamd.exim” running.
- Execute “chkconfig clamd.exim on” to make sure it will start on reboot.

Freshclam is normally run from cron; the RPMs I suggested install /etc/cron.d/clamav-update to do this.

otherwise:

- Just run 'clamd'

7.1.3. Sophie

If you installed as an RPM:

- Execute “service sophie start”. You should then have a process called “sophie” running.
- Execute “chkconfig sophie on” just to make sure it will start on reboot.

otherwise:

- read the Sophie documentation

7.1.4. Exim

Start Exim in your usual way (typically “service exim start” if you installed using an RPM)

7.2. Testing

7.2.1. Introduction

In the following sections I describe how to do low-level testing to check that your setup is working. This involves using “telnet” to connect to your mailserver and issue SMTP commands. It's good to understand how this works (and, indeed, you should). However, it tends to get tedious doing all this typing, especially when things aren't quite working and you need to debug. Therefore, I highly recommend using an SMTP debug tool. Personally I recommend and use the superb “swaks” from John Jetmore (http://jetmore.org/john/code/). This is written in Perl and has a number of CPAN dependencies though for Fedora users it is available from Fedora Extras (“yum install swaks”). Alternatively, Tony Finch from Cambridge University has written a simple SMTP client in C which performs a similar job: http://www-uxsup.csx.cam.ac.uk/~fanf2/hermes/src/smtpc/ .

If you use one of these SMTP testers, please substitute them as appropriate in the following test instructions; the principles are the same but you can save yourself some typing.

7.2.2. Spam scanning

The first test is to check that your spam scanning is working. Assuming that you've followed the instructions here, there are three possibilities for what happens to a mail when it comes in:

- It scores less than SpamAssassin's *required_hits* value, and is accepted as normal
- It scores more than SpamAssassin's *required_hits* value, but less than the rejection threshold set in SA-Exim/Content Scanning Extensions, in which case it is marked as spam (either in the headers, or in the subject, depending on how you configured things) but accepted
- It scores more than the rejection threshold set in SA-Exim/Exiscan, and is rejected at DATA time.

You therefore need to test these three circumstances. Before you do this, if you have virus scanning enabled via the Content Scanning Extension/Exiscan-ACL, temporarily disable it by commenting out the relevant ACL lines, just so we can eliminate one source of possible problems.

First, telnet to your mailserver on port 25 and send yourself a plain 'test' e-mail:

```
telnet your.server 25
```
[Wait for connection and the server banner, e.g. "220 your.server.name ESMTP Exim 4.xx ..."]
```
HELO test
```
[Server should respond: 250...]
```
MAIL FROM: <you@your.address>
```
[Server should respond: 250...]
```
RCPT TO: <you@your.address>
```
[Server should respond: 250...]
```
DATA
```
[Server should respond: 354...]
```
From: you@your.address
To: you@your.address
Subject: a test mail

test
.
```
[Server should respond: 250, possibly after a short pause whilst scanning takes place]

With a bit of luck, you should now have a message in your mailbox. Look at the source. It should have one or two additional headers (varying according to your software, versions, options and where you send the mail from), possibly something like this:

```
X-Spam-Status: No, hits=1.1 required=5.0
           tests=NO_REAL_NAME, MSGID_FROM_MTA_SHORT
           autolearn=no version=2.60
X-Spam-Level: *
```

If so, excellent! If not, or you got a temporary error (4xx) when you sent your mail, check your Exim main and reject logs to find out what happened.

Assuming that worked, now try sending a mail that's a bit spammy. Telnet to your server again, as before, but use something like this as the mail:

```
From: test@test
Subject: MAKE MONEY FAST!!! $$$        34fs4

viagra
```

The mail should be accepted, and if you look at the headers when you receive it, you should find something like this (again, will vary depending on whether you are using SA-Exim or Exiscan):

```
X-Spam-Status: Yes, hits=6.0 required=5.0
           tests=CASHCASHCASH,MISSING_HEADERS,NO_REAL_NAME,PLING_PLING,
                 SUBJ_HAS_SPACES,SUBJ_HAS_UNIQ_ID
           autolearn=no version=2.60
X-Spam-Level: *********
X-Spam-Checker-Version: SpamAssassin 2.60 (1.212-2003-09-23-exp) on your.server.example
X-Spam-Report:   ---- Start SpamAssassin results
  6.00 points, 5 required;
  *  0.8 -- From: does not include a real name
  *  1.7 -- Subject contains lots of white space
  *  1.3 -- Subject contains a unique ID
  *  0.5 -- Missing To: header
  *  0.0 -- Contains at least 3 dollar signs in a row
  *  1.7 -- Subject has lots of exclamation marks
  ---- End of SpamAssassin results
X-Spam-Flag: YES
```

You'll notice the X-Spam-Flag header, which can be useful to filter “possible spam” into a separate folder in your mail client. (The X-Spam-Level header can also be useful for this; it contains a number of asterisks corresponding to the spam rating of the mail).

If you are using SA-Exim, you may also get additional headers useful for debugging such as:

```
X-SA-Exim-Mail-From: you@your.address
X-SA-Exim-Version: 3.1 (built Sat Oct  4 09:02:58 BST 2003)
X-SA-Exim-Scanned: Yes
```

If you don't like these headers, you can get rid of them – see Appendix B.

Finally, let's try to get a message rejected. Telnet in again, but use this as your message:

```
From: test@test
Subject: MAKE MONEY FAST!!! $$$        34fs4
MiME-Version: 1.0

VIAGRA!!!!
AS SEEN ON NATIONAL TV, WORK FROM HOME!!!
GUARANTEED 100% THIS IS NOT SPAM!!!
```

This should get rejected, so instead of a final 250 code from the server, you should receive a 550. The message is rejected, and won't end up in your mailbox. If you're using SA-Exim and you've enabled saving of rejected messages, you should be able to see it in the save directory.

Note that there is also a 'magic' test phrase (referred to as 'GTUBE') which, if used in a mail, will trigger SpamAssassin to reject it (rather like the widely-used 'EICAR' test string for virus engines:

```
/XJS\*C4JDBQADN1\.NSBN3\*2IDNEN\*GTUBE-STANDARD-ANTI-UBE-TEST-EMAIL\*C\.34X/
```

7.2.3. Virus scanning

Assuming your spam scanning is now working (if you're doing it), it's time to test virus scanning. Re-enable it if you temporarily disabled it earlier.

Before you start, you'll need something to test for. A good test is the Eicar test signature, which is a 'pretend' virus recognised by most anti-virus software. You can download it from http://www.eicar.org/anti_virus_test_file.htm, or just paste the next line into a file (this is the 'virus'):

```
X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*
```

First, try sending yourself the file by telnet. Telnet in as before, and send the above line after DATA. Hopefully, you'll get a response something like this:

```
550 This message contains a virus or other harmful content (Eicar-Test-Signature).
```

If so, great! Looks good. Now use an e-mail client to send yourself viruses (through your server) using different forms such as enclosing in a ZIP file etc. It should all be detected and rejected.

7.2.3.1. Manual testing with Clam Antivirus

If you're having problems, and are using Clam Antivirus, you can test if it's working generally by:

- turning on TCP connections to ClamAV (rather than a UNIX socket)
  - check your clamd config (e.g. /etc/clamd.d/exim.conf for an RPM installation via exim-clamav)
- placing a test file somewhere on your filesystem
- checking that the test file can be read by the user running your Clam AV daemon
- telnetting to port 3310 (or whatever port you configured) on the machine running Clam AV and initiating a scan.
  - Type “SCAN /path/to/eicar.com”

Observe the response. Ideally you'll get a response something like “/path/to/eicar.com: Eicar-Test-Signature FOUND”. If not, you'll need to dig deeper. If you do, then try putting the Eicar file in /var/spool/exim/scan, setting it to be owned by [eximuser].[eximgroup] with permissions 0640 and scanning it then. If that fails, you've got a permissions problem. Check in particular that the ClamAV user is either in Exim's group, or a supplementary member. Note that by default Clam *does not inherit* supplementary groups; that is, if you have an entry like this in your groups file:

```
exim:93:clamav
```

It *will not* work unless you explicitly enable the AllowSupplementaryGroups option in your ClamAV daemon config file, and restart ClamAV. This problem comes up repeatedly on the exim-users list.

8. Appendix A – Building RPMs from source

If you want to build an RPM from source, you need three things: the source to the program (typically a .tar.gz file), an RPM 'specification' (spec) file, and (possibly) any ancillary external scripts/patches which are needed to build the software.

The command to build an RPM is “rpmbuild” on recent versions of RPM, or “rpm” on older versions. You may need to install the package “rpm-build” from your distribution, if it isn't already installed.

8.1. Using a build tree

To build things the 'proper' (conventional) way, you'll first need an RPM build tree. Typically, you'll have an example one installed which is writeable by the root user, in /usr/src/redhat. It's not recommended to build things as root, however, so it's best to reproduce this tree somewhere else and make it writeable by an unprivileged user. Then, in the home directory of the unprivileged user you're going to use to compile things, create an “.rpmmacros” file, containing a single line with this in: “%_topdir /path/to/your/rpm/build/tree”.

Now, to build an RPM, you need a spec file and program source files.

8.1.1. Tarball building

Occasionally, the piece of software you're using provides a spec file within it's tarball and if this is set up correctly and no external files are needed, you may be able to build an RPM directly from the tarball, like so:

```
rpmbuild -tb <file>.tar.gz
```

8.1.2. Normal building

Otherwise, need to first place the program sources and any ancillary files in the SOURCES directory of your RPM build tree. Next, put the spec file in the SPECS directory of your RPM build tree and change into that directory. Then type:

```
rpmbuild -ba <specfile>
```

where <specfile> is the name of the spec file for the program that you want to build. Remember this may be “rpm -ba <specfile>” on older systems.

If all goes well and there are no errors, the finished RPM should be in the RPMS directory of your build tree, in the i386 directory if you're building on an i386 machine (or possibly the 'noarch' directory if it's non-native; not the case for Exim). You'll also have a source RPM, which includes all the source files and the spec file itself, in the SRPMS directory of your build tree.

8.2. More Information

For more information about RPMs/spec files etc., please visit http://www.rpm.org/ .

9. Appendix B – Removing headers

This advice applies generally if you wish to remove specific headers from an e-mail, but is mainly applicable if you wish to remove headers such as SA-Exim's X-SA-Exim-Mail-From header.

The best way is to add a system filter rule. Outlined below are methods for if you do or don't already have a system filter. These assume you just want to blanket remove headers from all e-mails; you could use conditions etc. to remove them form only certain mails.

9.1. If you don't already have a system filter

First, you need to create a filter file. You can put this wherever you like, but a good place might be /etc/exim/system.filter. Put the following lines in it (note in particular the '#Exim filter' line which is crucial):

```
# Exim filter
headers remove X-SA-Exim-Scanned:X-SA-Exim-Mail-From
```

The above example removes the X-SA-Exim-Scanned and X-SA-Mail-From headers; adjust to your taste.

Now you need to tell Exim to use your filter as the 'system filter'. Add this to the main (top) part of your Exim configuration file:

```
system_filter = /etc/exim/system.filter
```

You should now find that messages passed through your system will have the headers you defined in the system.filter file removed.

9.2. If you already have a system filter

Just add the following line to the start of your filter (the following example removes X-SA-Exim-Scanned and X-SA-Exim-Mail-From; adjust to your taste).

```
headers remove X-SA-Exim-Scanned:X-SA-Exim-Mail-From
```

10. Appendix C – The GNU Free Documentation License

Version 1.2, November 2002

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- **A.** Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- **B.** List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- **C.** State on the Title page the name of the publisher of the Modified Version, as the publisher.
- **D.** Preserve all the copyright notices of the Document.
- **E.** Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- **F.** Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- **G.** Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- **H.** Include an unaltered copy of this License.
- **I.** Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- **J.** Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- **K.** For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- **L.** Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- **M.** Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- **N.** Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- **O.** Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warrany Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.