Exim 4 for Debian

Contents

1   Introduction

If you're reading this, you have found the README.Debian file. This is good, thanks! Please continue reading this file in its entirety. It is full of important information and has been written with the questions in mind that keep popping up on the mailing lists.

1.1   How to find your way around the Documentation

Exim comes with very extensive documentation. Here is how to find it.

1. A lot of information about Debian's Exim 4 packaging can be found in this document.

2. The packages contain a lot of Debian-specific man pages. Use the apropos exim command to get a list.

3. Most files that control the default configuration are documented in the exim4-config_files(5) man page, which is symlinked to the file names. man <filename> should lead you to the page.

4. The Debian Exim 4 packages have their own Home Page[1] which also links to a User FAQ.

5. The very extensive Upstream documentation is shipped

    a) in text form (/usr/share/doc/exim4-base/spec.txt.gz) with the binary packages.
    b) in HTML in the package exim4-doc-html
    c) as a Texinfo file in the package exim4-doc-info

Please note that documentation found on the web or in other parts of the Debian system (such as the Debian Reference) might be outdated and thus give wrong advice. In doubt, the documentation listed above should take precedence.

1.2   Getting Support

For your questions and comments, there is a Debian-specific mailing list[2]. Please ask Debian-specific questions there, and only write to the upstream

exim-users mailing list if you are sure that your question is not Debian-specific. Debian-specific questions are more likely to find answers on our pkg-exim4-users mailing list, while complex custom configuration issues might be more easily solved on the upstream exim-users mailing list because of the broader and more experienced audience there. You can subscribe to pkg-exim4-users[2] via the subscription web page; you need to be subscribed to post.

If you think that your question might be more easily answered if one knows a bit about your configuration, you might want to execute `reportbug --subject="none" --offline --quiet --severity=wishlist --body="none" --output=exim4.reportbug exim4-config` on the system in question, answer yes to both "include [extended] configuration" questions and include the contents of the exim4.reportbug file generated by this command with your question. Please check whether the file contains any confidential information before sending.

1.3 Packaging

Similar to the Apache2 package, Exim 4 is an entirely different package that does not currently offer a smooth upgrade path from Debian's Exim 3 packages.

It is the first Exim package in Debian that can be configured using debconf. However, the entire configuration framework is extremely flexible, allowing you to get exactly the amount of control you need for the job at hand.

The development web page[1] contains a lot of useful links and other information. The subversion repository of the Debian package is available for public read-only access and is linked from the development web page.

1.3.1 Feature Sets in the daemon packages

To use Exim 4, you need at least the following packages:

exim4-base
support files for all Exim MTA (v4) packages

exim4-config
configuration for the Exim MTA (v4)

exim4-daemon-light
lightweight exim MTA (v4) daemon

4

Just apting the metapackage exim4 will pull in the other packages per dependency. You'll get an exim daemon with minimal feature set (no external lookups).

If you need more advanced features like LDAP, sqlite, PostgreSQL and MySQL data lookups, SASL and SPA SMTP authentication, embedded Perl interpreter, and exiscan-acl for integration of virus-scanners and SpamAssassin, you can replace exim4-daemon-heavy instead of exim4-daemon-light. Additionally, the source package offers infrastructure to build your own custom-tailored exim4-daemon-custom which exactly fits your special local needs. The infrastructure to do so is already in place, see debian/rules for instructions.

1.3.2   How to build a custom daemon

The process of building a custom daemon is partially documented in the debian/rules file in the source package. Patches for more documentation are welcome.

2   Configuration of Exim 4 in the Debian packages

Generally, the Debian Exim 4 packages are configured through debconf. You have been asked some questions on package installation, and your initial Exim configuration has been created from your answers. You can repeat the configuration process any time by invoking dpkg-reconfigure exim4-config. If you are an experienced Exim administrator and prefer to have your own, hand-crafted, non-automatic Exim configuration, you will find information about how to do so in Section 2.1.6, "Using a completely different configuration scheme".

The debconf-driven configuration is mainly geared for a one-domain shell account machine/workstation with local delivery as suggested by the original upstream default configuration. If you configure the packages to handle more than one local domain, all local domains are treated identically. The domain part is not used for routing and filtering decisions.

Despite the default configuration being extended somewhat from the original upstream, chances are that you'll need to manually change the Exim configuration with an editor if you intend to do something that is not covered by the debconf-driven configuration. It has never been the

packages' intention to offer all possible configuration methods through debconf. The configuration files are there to be changed, feel free to do so if you see fit. The Debian Exim 4 maintainers have tried to make the configuration as flexible as possible so that manual intervention can be minimized.

If you need to make manual changes to the Exim configuration, please be familiar with how Exim works. At minimum, have read this README file and the manpages delivered with the Debian Exim 4 packages, and /usr/share/doc/exim4-base/spec.txt.gz chapters 3 and 6. spec.txt.gz is an excellent reference.

Please note that while most free-form fields in the debconf-driven configuration have the entered string end up verbatim in Exim's configuration file (and thus using more advanced features like host, address and domain lists is possible and will probably work), this is not officially supported. Only plain lists are supported in the debconf dialogs. You may use more advanced features, but they may stop working any time during upgrades.

2.1   The Configuration System

2.1.1   The Debconf questions

In this section, we try to document and explain the debconf questions, which are themselves limited to a small screen of information and might leave questions unanswered. Since you can usually read this file only after having answered the questions, the process can always be repeated by invoking `dpkg-reconfigure exim4-config`. /etc/exim4/update-exim4.conf.conf, documented in the update-exim4.conf manual page, is a simple shell-script snippet used to store the answers that you passed to debconf when initially configuring Exim. You may also modify this file with an editor of your choice. The package maintainer scripts can handle this and will preserve your changes.

2.1.1.1   General type of mail configuration

This is the main configuration question which will control which of the remaining questions are presented to you. It also controls things like daemon invocation and delivery of outgoing mail.

2.1.1.1.1   internet site; mail is sent and received directly using SMTP

This option is suitable for a standalone system with full internet connectivity.

- The Exim SMTP daemon will accept messages to local domains, and deliver them locally.

- Outgoing mail will be delivered directly to the mail exchange servers of the recipient domain.

2.1.1.1.2   mail sent by smarthost; received via SMTP or fetchmail

This option is suitable for a standalone client system which has restricted internet connectivity, for example on a residential connection where an SMTP smarthost is used. Some ISPs block outgoing SMTP connections to combat the spam problem, thus requiring the use of their smarthosts. It is generally a good idea to use the ISPs smart host if one is connected with a dynamic IP address since quite a few sites do not accept mail directly delivered from a dial-in pool.

fetchmail can be used to retrieve incoming mail from the ISP's POP3 or IMAP mail server and deliver it to Exim via SMTP.

- The Exim SMTP daemon will accept messages to local domains, and deliver them locally.

- Outgoing mail will always be delivered to the smarthost configured in exim4.

2.1.1.1.3   mail sent by smarthost; no local mail

This option is suitable for a client system in a computer pool which is not responsible for a local e-mail domain. All locally generated e-mail is sent to the smarthost without any local domains.

2.1.1.1.4   local delivery only; not on a network

This option is suitable for a standalone system with no networking at all. Only messages for configured local domains are accepted and delivered locally; messages for all other domains are rejected:
"Mailing to remote domains not supported".

2.1.1.1.5   no configuration at this time

This option disables most of Debian's automatisms and leaves exim in an unconfigured state. update-exim4.conf will still copy
/etc/exim4/exim4.conf.template
or concatenate the files from /etc/exim4/conf.d, and will not generate any configuration control macros. Unless you manually edit the configuration source, this will leave Exim with a syntactically invalid configuration file, thus in a state where the daemon won't even start.

Only choose this option if you know what you're doing and are prepared to create your own Exim configuration.

dpkg-conffile handling is still in place, and you will be offered updates for configuration snippets, as soon as they become available.

2.1.1.2   System mail name

The "name" is the domain name used to "qualify" mail addresses without a domain name.

This name will also be used by other programs. It should be the single,full domain name (FQDN).

For example, if a mail address on the local host is foo@example.org, then the correct value for this option would be example.org.

Exim, as a rule, handles only fully qualified mail addresses, that is, addresses with a local part, an @ sign and a domain. If confronted with an unqualified address, that is, one without @ sign and without domain, first thing exim does is qualify the address by adding the @ sign and a domain.

This qualification happens for all addresses exim encounters, be it sender, recipient or else.

The domain name used to qualify unqualified mail addresses is called "mail name" on Debian systems and entered in this debconf dialog. What you enter here will end up in /etc/mailname, which is a file that might be used by other programs as well.

In some configuration types, the package configuration will offer you, at a later step, to hide this name from outgoing messages by rewriting the headers.

2.1.1.3   IP addresses to listen on for incoming SMTP connections

Please enter a semicolon-separated list of IP addresses. The Exim SMTP listener daemon will listen on all IP addresses listed here.

An empty value will cause Exim to listen for connections on all available network interfaces.

If this system does only receive e-mail directly from local services (and not from other hosts), it is suggested to prohibit external connections to the local Exim daemon. Such services include e-mail programs (MUSs) which talk to localhost only as well as fetchmail. External connections are impossible when 127.0.0.1 is entered here, as this will disable listening on public network interfaces.

Do not change this unless you know what you are doing. Altering this value could post a security risk to your system. For most users, the default value is sufficient.

2.1.1.4   Other destinations for which mail is accepted

Please enter a semicolon-separated list of recipient domains for which this machine should consider itself the final destination. These domains are commonly called 'local domains'. The local hostname and 'localhost' are always added to the list given here.

By default all local domains will be treated identically. If both a.example and b.example are local domains, acc@a.example and acc@b.example will be delivered to the same final destination. If different domain names should be treated differently, it is necessary to edit the config files afterwards.

The answer to this question ends up in the list of domains that Exim will consider local domains. Mail for recipients in one of these domains will be subject to local alias expansion and then delivered locally in the appropriate configuration types.

2.1.1.5   Domains to relay mail for

Please enter a semicolon-separated list of recipient domains for which this system will relay mail, for example as a fallback MX or mail gateway. This means that this system will accept mail for these domains from anywhere on the Internet and deliver them according to local delivery rules.

Do not mention local domains here. Wildcards may be used.

The answer to this question is a list of the domains for which Exim will relay messages coming in from anywhere on the Internet.

2.1.1.6   Machines to relay mail for

Please enter a semicolon-separated list of IP address ranges for which this system will unconditionally relay mail, functioning as a smarthost.

You should use the standard address/prefix format (e.g. 194.222.242.0/24 or 5f03:1200:836f::/48).

If this system should not be a smarthost for any other host, leave this list blank.

Please note that systems not listed here can still use SMTP AUTH to relay through this system. If this system only has clients on dynamic IP addresses that use SMTP AUTH, leave this list blank as well. Do NOT list 0.0.0.0/0!

2.1.1.7   IP address or host name of the outgoing smarthost

Please enter the IP address or the host name of a mail server that this system should use as outgoing smarthost. If the smarthost only accepts your mail on a port different from TCP/25, append two colons and the port number (for example smarthost.example::587 or 192.168.254.254::2525). Colons in IPv6 addresses need to be doubled.

If the smarthost requires authentication, please refer to Section 2.3, "SMTP-AUTH" for notes about setting up SMTP authentication.

Multiple smarthost entries are permitted, semicolon separated. Each of the hosts is tried, in the order specified (See Exim specification, chapter 20.5).

2.1.1.8   Hide local mail name in outgoing mail

The headers of outgoing mail can be rewritten to make it appear to have been generated on a different system, replacing the local host name in From, Reply-To, Sender and Return-Path.

2.1.1.9   Visible domain name for local users

If you ask Exim to hide the local mail name in outgoing mail, it will next ask you for the domain name that should be visible for your local users. These information is then used to establish the appropriate rewriting rules.

2.1.1.10   Keep number of DNS queries minimal (Dial-on-Demand)

In normal mode of operation Exim does DNS lookups at startup, and when receiving or delivering messages. This is for logging purposes and allows keeping down the number of hard-coded values in the configuration.

If this system does not have a DNS full service resolver available at all times (for example if its Internet access is a dial-up line using dial-on-demand), this might have unwanted consequences. For example, starting up Exim or running the queue (even with no messages waiting) might trigger a costly dial-up-event.

This option should be selected if this system is using Dial-on-Demand. If it has always-on Internet access, this option should be disabled.

2.1.1.11   Delivery method for local mail

Exim is able to store locally delivered mail in different formats. The most commonly used ones are mbox and Maildir. mbox uses a single file for the complete mail folder stored in /var/mail/. With Maildir format every single message is stored in a separate file in /Maildir/.

Please note that most mail tools in Debian expect the local delivery method to be mbox in their default.

2.1.1.12   Split configuration into small files

Our packages offer two (actually three, see Section 2.1.6, "Using a completely different configuration scheme") possibilities:

1. Generate Exim's configuration from
   /etc/exim4/exim4.conf.template,
   which is basically a normal Exim run-time configuration file which
   will be supplemented with some macros generated from Debconf in
   a post-processing step before it is passed to exim.

2. Generate Exim's configuration from the multiple files in /etc/exim4/conf.d/.
   The directories in /etc/exim4/conf.d/ correspond to the sections of the Exim run-time configuration file, so you should easily find your way around there.

Splitting the configuration across multiple files means that you have the actual configuration file automatically generated from the files below /etc/exim4/conf.d/ by invoking update-exim4.conf. Each section of Exim's configuration has its own subdirectory and the files in there are supposed to be read in alphanumeric order.
router/00_exim4-config_header is followed by
router/100_exim4-config_domain_literal, . . .

If you chose unsplit configuration, update-exim4.conf builds the configuration from /etc/exim4/exim4.conf.template, which is basically the files from /etc/exim4/conf.d/ concatenated together at package build time, and thus guarantees consistency on the target system.

In both cases, update-exim4.conf generates exim configuration macros from the debconf configuration values and puts them into the actual configuration file, which is then used by the Exim daemon. See the update-exim4.conf manual page for more in-depth information about this mechanism.

Benefits of the split configuration approach:

- it means less work for you when upgrading. If we shipped one big file and modified for example the Maildir transport in a new version you won't have to do manual conffile merging unless you had changed exactly this transport.

- It allows other packages (e.g. sa-exim) to modify Exim's configuration by dropping files into /etc/exim4/conf.d. This needs, however quite exact syncing between the exim4 packages and the other, cooperating package.

Drawbacks of the split configuration approach:

- It is more fragile. If files from different sources (package, manually changed, or other package) get out of sync, it is possible for Exim to break until you manually correct this. This can for example happen if we decide to add a new option to the Debian setup of a later version, and you have already set this option in a local file.

Benefits of the unsplit configuration approach:

- People familiar with configuring Exim may find this approach easier to understand as exim4.conf.template basically is a complete Exim configuration file which will only undergo some basic string replacement before is it passed to exim.

- Split-config's fragility mentioned above does not occur.

Drawbacks of the unsplit configuration approach:

- Will require manual intervention in case of an upgrade.

If in doubt go for the unsplit config, because it is easier to roll back to Debian's default configuration in one step. If you intend to do many changes to the Debian setup, you might want to use the split config at the price of having to more closely examine the config file after an update.

We'd appreciate a patch that uses ucf and the 3-way-merge mechanism offered by that package. It might be the best way to handle the big configuration file.

If you are using unsplit configuration, have local changes to /etc/exim4/conf.d/ (either made by yourself or by other packages dropping their own routers or transports in) and want to re-generate /etc/exim4/exim4.conf.template to activate these changes, you can do so by using update-exim4.conf.template.

2.1.2   Access Control in the default configuration

The Debian exim 4 packages come with a default configuration that allows flexible access control and blacklisting of sites and hosts. The acls involved can be found in /etc/exim4/conf.d/acl, or in /etc/exim4/exim4.conf.template, depending on which configuration scheme you use. Most rejections of messages due to this mechanism happen at RCPT time. Local configuration of the mechanisms happens through data files in /etc/exim4 or via Exim macros that you can set in /etc/exim4/conf.d/main, so there is normally no need to change the files in the acl subdirectory in a split-config setup. If you use the non-split config, you need to edit /etc/exim4/exim4.conf.template, which, as a big dpkg-conffile, won't give you any advantage of the .ifdef scheme.

The data files are documented in the exim4-config_files man page.

The access lists delivered with the exim4 packages also contain quite a few configuration options that are too restrictive to be active by default on a real-life site. These are masked by .ifdef statements, can be activated by setting the appropriate macros, and are documented in the ACL files itself.

2.1.3   Using Exim Macros to control the configuration

Our configuration can be controlled in a limited way by setting macros. That way, you can switch on and off certain parts of the default configuration and/or override values set in Debconf without having to touch the dpkg-conffiles. While touching dpkg-conffiles itself is explitly allowed and wanted, it can be quite a nuisance to be asked on package upgrade whether one wants to use the locally changed file or the file changed by the package maintainer.

Whenever you see an .ifdef or .ifndef clause in the configuration file, you can control the appropriate clause by setting the macro in a local configuration file. For split configuration, you can drop the local configuration file anywhere in /etc/exim4/conf.d/main. Just make sure it gets read before the macro is first used. 000_local-macros is a possible name, guaranteeing first order. For a non-split configuration, /etc/exim4/exim4.conf.localmacros gets read before /etc/exim4/exim4.conf.template. To actually set the macro EXIM4_EXAMPLE to the value"this is a sample", write the following line

EXIM4_EXAMPLE = this is a sample

into the appropriate file. For more detailed discussion of the general macro mechanism, see the Exim specification, chapter 6.4, for details how macro expansion works.

2.1.4   How does this work?

The   script   update-exim4.conf   parses   the   /etc/exim4/update-exim4.conf.conf file and provides the configuration for the exim daemon.

Depending on the value of dc_use_split_config, it either

- takes all the files below /etc/exim4/conf.d/ and concatenates them together or

- uses exim4.conf.template as input.

The debconf-managed information from /etc/exim4/update-exim4.conf.conf is merged into the generated configuration file by generating a number of Exim configuration macros.

DCsmarthost, for example, is set to the value of dc_smarthost in /etc/exim4/update-exim4.conf.conf which holds the answer to "Which machine will act as the smarthost and handle outgoing mail?"

The result of these operations is saved as /var/lib/exim4/config.autogenerated, which is not a dpkg-conffile! Manual changes to this file will be overwritten by update-exim4.conf.

Please consult update-exim4.conf manpage for more detailed information.

update-exim4.conf is invoked by the init script prior to any operation that may invoke an exim process, and gives an error message if the generated config file is syntactically invalid. If you want to activate your changes to files in conf.d/ just execute invoke-rc.d exim4 restart.

2.1.5   How do I do minor tweaks to the configuration?

Some times, you want to do minor adjustments to the Exim configuration to make Exim behave exactly like you want it to behave. There are the following possibilities to modify Exim's behavior.

2.1.5.1   Adjustments supported by the debconf configuration

If you want to modify parameters that are supported by the debconf configuration, things are easy. Just invoke dpkg-reconfigure exim4-config or hand-edit /etc/exim4/update-exim4.conf.conf to your liking and restart Exim.

You can find explanation of the debconf questions in Section 2.1.1, "The Debconf questions". Additionally, /etc/exim4/update-exim4.conf.conf is documented in the update-exim4.conf man page.

2.1.5.2   Adjustments controlled by macros in the Debian Exim configuration

Some aspects of the Debian Exim configuration can be controlled by Exim macros. To find out about these, you need basic understanding of Exim configuration. Just look in our Exim configuration and see which macro needs to be set to a different value to alter Exim's behavior.

Section 2.1.3 "Using Exim Macros to control the configuration" gives a closer explanation about how to do this.

2.1.5.3   Making direct changes to the Debian Exim configuration

You can, of course, make direct change to the configuration. All configuration files in /etc/exim4 are dpkg-conffiles, and you can thus edit them any time. Your changes will be preserved through updates. You need to know about how to configure Exim to be successful.

If you use unsplit configuration, edit /etc/exim4/exim4.conf.template. If you use split configuration, edit the Exim configuration snippets in /etc/exim4/conf.d.

More information about how the Exim configuration is built can be found in this document and in the update-exim4.conf manual page.

2.1.6   Using a completely different configuration scheme

If you are an experienced Exim administrator, you might feel working with our pre-fabricated configuration cumbersome and complex. You might feel right if you need to make more complex changes and do not need to receive updates from us. This section is going to tell about how to use your own configuration.

But, you might profit from keeping the Debian magic. Most files that come with Debian exim4 are conffiles. Debian is going to care about your changes and keeps them around. Additionally, a lot of configuration options can be overridden with a macro, which does not require you to actually change our configuration file. A lot of people are using our configuration scheme, and maybe it is going to save you a lot of time if you decide to spend some time familiarizing yourself with our scheme.

2.1.6.1   Override exim4-config configuration magic

If you are only running a small number of systems and want to completely disable Debian's magic, just take your monolithic configuration file and install it as /etc/exim4/exim4.conf. Exim will use that file verbatim. To have something to start, you can either take /etc/exim4/exim4.conf.template, run
`update-exim4.conf --keepcomments --output /etc/exim4/exim4.conf,` or use upstream's default configuration file that is installed as

/usr/share/doc/exim4-base/examples/example.conf.gz. You are going to lose all magic you get from packaging though, so you need to be familiar with Exim to build an actually working config.

/var/lib/exim4/config.autogenerated, the file generated by update-exim4.conf, is ignored as soon as /etc/exim4/exim4.conf is found. You should not edit /etc/exim4/exim4.conf directly when Exim is running, because the forked processes Exim starts for SMTP receiving or queue running would use the new configuration file, while the original main exim-daemon would still use the old configuration file.

Some third-party HOWTOs that reference Debian and claim to make things easy suggest dumping a pre-fabricated, static config file to /etc/exim4/exim4.conf. This is considered bad advice by the Debian maintainers since you are going to disable all updates and service magic that Debian might deliver in the future this way. If you do not know exactly what you're doing here, this is a bad choice. We try to comment on external HOWTOs found on the web in the Debian Exim4 User FAQ[3] to help you find out which advice to follow.

2.1.6.2   Replacing exim4-config with your own exim4 configuration package

We split off Exim's configuration system (debconf, update-exim4.conf, and the files in /etc/exim4/conf.d) to a separate package, exim4-config. If you want to, you can replace exim4-config by something entirely different. The other packages don't care. Your package needs to:

- Provides: exim4-config-2, Conflicts: exim4-config-2,exim4-config

- drop the Exim 4 configuration either into
  /var/lib/exim4/config.autogenerated or into
  /etc/exim4/exim4.conf.

Your package must provide an executable update-exim4.conf that must be in root's path (/usr/sbin recommended). The init script will invoke that executable prior to invoking the actual exim daemon. If you do not need that script, have it exit 0.

If you want to create your own configuration packages, there is a number of helpers available.

- The Exim 4 Debian svn repository holds sources for a exim4-config-simple package which contains a simple, not debconf-driven config-

uration scheme as an example which can be used as a template for a classical, exim4.conf based configuration scheme.

- The Exim 4 Debian svn repository holds sources for a exim4-config-medium package which contains the conf.d driven configuration of the main package with the debconf interaction removed. This can be used to create your own non-debconf configuration package that uses the conf.d mechanism.

- Finally, you can invoke the script debian/config-custom/create-custom-config-package which will create a new source package"exim4-config-custom" with the debconf-driven config scheme of exim4-config for your local modification.

Please note that exim4-config-simple and exim4-config-medium are only targeted to be used as a template. The configurations contained are not suitable for productive use. Of course, the Debian maintainers appreciate any patches you might find suitable. The scripts in exim4-config-simple and exim4-config-medium may not work at all in your environment. Unfortunately, they have not been updated in a long time as well. We are willing to accept patches.

See the development web page for links to the subversion repository.

Exchanging the entire exim4-config package with something custom comes particularly handy for sites that have more than a few machines that are similarly configured, but do not want to use the original exim4-config package. Build your own exim4-config-custom or exim4-config-foo, and simply apt that package to the machines that need to have that configuration. Future updates can then be handled via the dpkg-conffile mechanism, properly detecting local modifications.

In the future, it might be possible that Debian will contain multiple flavours of Exim4 configuration. However, these packages would have to be maintained by someone else because the exim4 package maintainers think that the scheme delivered with exim4-config is the least of all evils and would rather not spend the time to maintain multiple configuration schemes while only actually using one. It would be nice to have a configuration scheme using a monolithic config file, managed by ucf in three-way-merge mode. If anybody feels ready to maintain it, please go ahead.

2.2   Using TLS

2.2.1   Exim 4 as TLS/SSL client

Both exim4-daemon-heavy and exim4-daemon-light support TLS/SSL using the GnuTLS library and STARTTLS. Exim will use TLS via STARTTLS automatically as client if the server Exim connects to offers it.

This means that you will not need any special configuration if you want to use TLS for outgoing mail. However, if your server setup mandates the use of client certificates, you need to amend your remote_smtp and/or remote_smtp_smarthost transports with a tls_certificate option. This is not commonly needed.

The certificate presented by the remote host is not checked unless you specify a tls_verify_certificate option on the transport.

TLS on connect is not natively supported.

2.2.2   Enabling TLS support for Exim as server

You should have created certificates in /etc/exim4/ either by hand or by usage of the exim-gencert (which requires openssl). exim-gencert is shipped in /usr/share/doc/exim4-base/examples/ and takes care of proper access privileges on the private key file.

If you are on a host that does generate little entropy, you might want to manually generate the Diffie-Hellman parameters by invoking /usr/share/exim4/exim4_refresh_gnutls-params. For this, you need either gnutls-bin or openssl installed.

Now, enable TLS by setting the macro MAIN_TLS_ENABLE in a local configuration file as described in Section 2.1.3, "Using Exim Macros to control the configuration".

After this configuration, Exim will advertise STARTTLS when connected to on the normal SMTP ports. Some broken clients (most prominent example being nearly all versions of Microsoft Outlook and Outlook Express, and Incredimail) insist on doing TLS on connect on Port 465.
If you need to support these, set SMTPLISTENEROPTIONS='-oX 465:25 -oP /var/run/exim4/exim.pid' in /etc/default/exim4 and "tls_on_connect_ports=465" in the main configuration section.

The -oP is needed because Exim does not write an implicit pid file if -oX is given. Without pid file, init script and cron job will malfunction.

It might be appropriate to add "+tls_cipher +tls_peerdn" to any

19

log_selector statement you might already have, or to add a log_selector statement setting these two options in a local configuration file. These options have Exim log what cipher your Exim and the peer's mailer have negotiated to use to encrypt the transaction, and they have Exim log the Distinguished Name of the peer's certificate.

By default, exim asks a client for a certificate and logs its properties, but does not do any verification of the certificate. Some clients (most notably, for example, Incredimail (verified in February 2008)) choke on the certificate request and issue a generic error message. The client certificate request can be disabled by setting a macro in Exim configuration.

2.2.3 Diffie-Hellman parameters

This version of Exim is compiled against GnuTLS. GnuTLS is a replacement for the restrictively licensed OpenSSL libraries. GnuTLS does not support varying its Diffie-Hellman parameters. Therefore tls_dhparam settings are ignored in Exim's configuration file, and no dhparam file is generated by exim-gencerts. GnuTLS uses Diffie-Hellman parameters that are computed when they are needed. When someone sends STARTTLS, Exim will compute these parameters and then store these parameters in a cache file located in Exim's spool directory (/var/spool/exim4/gnutls-params).

The daily cron job and the script
/usr/share/exim4/exim4_refresh_gnutlsparams
take care of new Diffie-Hellman parameters. If neither gnutls-bin nor openssl are installed, the gnutls-params file is removed and Exim regenerates the file on the fly during the next incoming TLS connection. Systems generating little entropy might hang in this situation after clients invoking a STARTTLS command.

To avoid this behavior, which can possibly lead to a DoS condition, if the daily cron job finds openssl or gnutls-bin installed, it will regenerate the gnutls-params file outside of Exim and only replace the gnutls-params file after a new one has been successfully generated. If the new file generation does not finish after an hour, the process is killed to avoid sustained entropy depletion. If the gnutls-params file gets older than two weeks, the daily cron job starts sending out warning messages.

It is "more secure" when you have the gnutls-params file regenerated more often. You can delete it any time you wish without any need for synchronization. Exim will regenerate it automatically. Alternatively, you

can manually invoke /usr/share/exim4/exim4_refresh_gnutls-params to regenerate the file.

NOTE! The fact that GnuTLS does not support generated Diffie-Hellman parameters does NOT make it less secure.

For more reference, you can refer to /usr/share/doc/exim4-base/spec.txt.gz, section 38.

2.2.4   Troubleshooting

If Exim complains in an SMTP session that TLS is unavailable, the Exim mainlog or paniclog frequently has exact information about what might be wrong. Fo example, you might see

2003-01-27 19:06:45 TLS error on connection from localhost [127.0.0.1] (cert/key setup): Error while reading file)

showing that there has been an error while accessing the certificate or the private key file.

Insuffient entropy available is a frequent cause of TLS failures in Exim context. If Exim logs "not enough random bytes available", or simply hangs silently when an encrypted connection should be established, then Exim was unable to read enough random data from /dev/random to do whatever cryptographic operation is requested. Please check that your /dev/random device is setup properly.

A process that regularly consumes a lot of entropy is the regeneration of the Diffie-Hellman parameters. See Section 2.2.3, "Diffie-Hellman parameters" for more information.

2.3   SMTP-AUTH

Exim can do SMTP AUTH both as a client and as a server.

AUTH PLAIN and AUTH LOGIN are disabled for connections which are not protected by SSL/TLS per default. These authentication methods use cleartext passwords, and allowing the transmission of cleartext passwords on unencrypted connections is a security risk. Therefore, the default configuration configures Exim not to use and/or allow AUTH PLAIN and AUTH LOGIN over unencrypted connections.

It is thus recommended to set up Exim to use TLS to encrypt the connections. Please refer to Section 2.2, "Using TLS" for documentation about this. Note that most Microsoft clients need special handling for TLS.

2.3.1   Using Exim as SMTP-AUTH client

If you want to set up Exim as SMTP AUTH client for delivery to your internet access provider's smarthost put the name of the server, your login and password in /etc/exim4/passwd.client. See the man page for exim4-config_files(5) for more information about the required format.

If you need to enable AUTH PLAIN or AUTH LOGIN for unencrypted connections because your service provider does support neither TLS encryption nor the CRAM MD5 authentication method, you can do so by setting the AUTH_CLIENT_ALLOW_NOTLS_PASSWORDS macro. Please refer to Section 2.1.3, "Using Exim Macros to control the configuration" for an explanation of how best to do this.

/etc/exim4/passwd.client needs to be readable for the exim user (user Debian-exim, group Debian-exim). It is suggested that you keep the default permissions root:Debian-exim 0640.

2.3.2   Using Exim as SMTP-AUTH server

The configuration files include many, verbosely commented, examples for server-side smtp-authentication which just need to be uncommented.

If you need to enable AUTH PLAIN or AUTH LOGIN for unencrypted connections because your clients neither support TLS encryption nor the CRAM MD5 authentication method, you can do so by setting the AUTH_SERVER_ALLOW_NOTLS_PASSWORDS macro. Please refer to Section 2.1.3, "Using Exim Macros to control the configuration" for an explanation of how best to do this.

If you want to authenticate against system passwords (e.g. /etc/shadow) the easiest way is to use saslauthd in the Debian package sasl2-bin. You have to add the exim-user (currently Debian-exim) to the sasl group, to give exim permission to use the saslauthd service.

The Debian exim4 maintainers consider using system login passwords a bad idea for the following reasons:

- A compromised password will give access to a system account.

- E-Mail passwords could accidentally be transmitted unencrypted.

- E-Mail passwords are likely to be stored with the client software, which greatly increases the chance of a compromise.

2.4   How the Exim daemon is started

The Debian Exim 4 packages' init script is located in /etc/init.d/exim4. Apart from the functions that are required by Debian policy and the LSB, it supports the commands what, which executes exiwhat to show what your Exim processes are doing, and force_stop which unconditionally kills all Exim processes.

The init script can be configured to start listening and/or queue running daemons. This configuration can be found in /etc/default/exim4. This file is extensively documented.

2.5   Miscellaneous packaging issues

2.5.1   The daily cron job

Exim4's daily cron job (/etc/cron.daily/exim4-base) does basic housekeeping tasks:

- It reads /etc/default/exim4, so you can use this file to change any of the variables used in the cron job.

- It is a no-op if no Exim4 binary is found.

- If $E4BCD_DAILY_REPORT_TO is set to a non-empty string, the output of eximstats is mailed to the address given in that variable. The default is empty, so no reports are sent. Options for eximstats can be given in $E4BCD_DAILY_REPORT_OPTIONS.

- A non-empty paniclog is a nearly sure sign of bad things going on. Thus, the cron job will send out warning messages to the syslog and root if it finds the panic log non-empty. Please note that the paniclog is not rotated daily, so existing issues will be reported daily until either the paniclog is rotated due to its sheer size, or you manually move it away, for example by calling logrotate -f /etc/logrotate/exim4-paniclog from a shell. Just in case your system logs transient error situations to the panic log as well (see, for example, Exim Bug 92[4]), you can configure $E4BCD_PANICLOG_NOISE to a regular expression. If the paniclog contains only lines that match that regular expression, no warning messages are generated. If you want to disable paniclog monitoring completely, set $E4BCD_WATCH_PANICLOG to

no. $E4BCD_WATCH_PANICLOG=once will rotate a non-empty paniclog automatically after sending out the warning e-mail.

- It tidies up the retry and hints databases.

- If TLS is enabled, it regenerates the GnuTLS parameter file. If that process fails (maybe because your system being short of entropy), and the gnutls-params file thus gets older than $E4BCD_GNUTLS_PARAMS_MAXAGE, the cron job logs this to syslog and sends out a warning message to root.

2.6 Using Exim with inetd/xinetd

Exim4 is run as a separate daemon instead of inetd/xinetd for two reasons:

Ease of maintenance:
update-inetd is difficult to impossible to handle correctly (Just check the archived bug reports of Exim.) and update-inetd seems to be unmaintained for a long time, nobody dares to touch it. To quote Mark Baker, the maintainer of Exim (v3): "I really wish I had never used inetd in the first place, but simply set up exim to run as a daemon, but it's too late to change that now."

Extended features
Running from inetd interferes with Exim's resource controls (e.g it disables smtp_accept_max_per_host and smtp_accept_max).

If you introduce bugs on your systems by running from (x)inetd you are on your own! If you want to run exim from xinetd, follow these steps:

1. Disable Exim 4's listening daemon by executing
   `update-exim4defaults --queuerunner queueonly`

2. Create /etc/xinetd.d/exim4 service smtp
   {
   disable= no
   flags = NAMEINARGS
   socket_type = stream
   protocol = tcp
   wait= no
   user= Debian-exim
   group = Debian-exim
   server= /usr/sbin/exim4
   server_args = exim4 -bs
   }

3. Run invoke-rc.d exim4 restart; invoke-rc.d (x)inetd restart

If you want to use plain inetd, insert following line into /etc/inetd.conf:
smtp streamtcpnowaitDebian-exim/usr/sbin/exim4 exim4 -bs

2.7   Handling incoming mail for local accounts with low UID

Since system accounts (mail, uucp, lp etc) are usually aliased to root, and root's mailbox is usually read by a human, these account names have started to be a common target for spammers. The Debian Exim 4 packages have a mechanism to deal with this situation. However, since this derives rather far from normal behavior, it is disabled by default.

To enable it, set the macro FIRST_USER_UID to a numeric, non-zero value. Incoming mail for local users that have a UID lower than FIRST_USER_UID is rejected with the message "no mail to system accounts". Incoming mail for local users that have a UID greater or equal FIRST_USER_UID are processed as usual. Therefore, the default value of 0 ensures that the mechanism is disabled. On Debian systems, setting FIRST_USER_UID to 500 or 1000 (depending on your local policy) will disable incoming mail for system accounts.

Just in case that you need exceptions to the rule, /etc/exim4/lowuid_aliases is an alias file that is only honored for local accounts with UID lower than FIRST_USER_UID. If you define an alias for such an account here, incoming mail is processed according to the alias. If you alias the account to itself, messages are delivered to the account itself, which is an exception to the rule that messages for low-UID accounts are rejected. The format of /etc/exim4/lowuid_aliases is just another alias file.

2.8   How to bypass local routing specialities

Sometimes, it might be desireable to be able to bypass local routing specialities like the alias file or a user-forward file. This is possible in the Debian Exim4 packages by prefixing the account name with "real-". For a local account name "foo", "real-foo@hostname.example" will result in direct delivery to foo's local Mailbox.

This feature is by default only available for locally generated messages.

If you want it to be accessible for messages delivered from remote as well, set the Exim macro COND_LOCAL_SUBMITTER to true. If you do not want this at all, set the macro to false. Please note that the userforward router uses this feature to get error messages delivered, i.e. notifying the user of a syntax error in her .forward file.

2.9   Using more complex deliveries from alias files

Delivery to arbitrary files, directory or to pipes in the /etc/aliases file is disabled by default in the Debian Exim 4 packages. The delivery process including the program being piped to would run as the exim admin-user Debian-exim, which might open up security holes.Invoking pipes from /etc/aliases file is widely considered obsolete and deprecated. The Debian Exim package maintainers would like to suggest using a dedicated router/transport pair to invoke local processes for mail processing. For example, the Debian mailman package contains a /usr/share/doc/mailman/README.Exim4.Debian file that gives a good example how to implement this. Using a dedicated router/transport pair have the following advantages:

- The router/transport pair can be put in place by another package, giving a well-defined transaction point between Exim 4 and $PACK-AGE.

- Not allowing pipe deliveries from alias files makes it harder to accidentally run programs with wrong privileges.

- It is possible to run different pipe processes under different accounts.

- Even if only invoking a single local program, it is easier to do with your dedicated router/transport since you won't need to change this file, making automatic updates of this file possible for future versions of the Exim 4 packages. If you do local changes here, dpkg conffile handling will bother you on future updates.

If you insist on using /etc/aliases in the traditional way, you will need to activate the respective functions by setting the transport options on the system_aliases router appropriately. Macros are defined to make this easier. See /etc/exim4/conf.d/router/400_exim4config_system_aliases for information about which macros are available. You might find the address_file, address_pipe and/or address_directory transports that are

used for the userforward router helpful in writing your own transports for use in the system_aliases router.

If any of your aliases expand to pipes or files or directories you should set up a user and a group for these deliveries to run under. You can do this by setting the "user" and – if necessary – a "group" option and adding a "group" option if necessary. Alternatively, you can specify "user" and/or "group" on the transports that are used.

2.10   Putting Exim 4 and UUCP together

UUCP is a traditional way to execute remote jobs (e.g. spool mails), and as a lot of old things there are much more than one way to do it. However, today, the ways to handle it have boiled down to more or less two different ways.

Our recommendation is to use bsmtp/rsmtp wherever possible, because it supports all kinds of mail addresses (also the empty ones in bounces), and is also better from the security point of view.

2.10.1   Sending mail via UUCP

2.10.1.1   rmail with full addresses

rmail is the oldest way to transfer mail to a remote system. However, today it is normally required to use addresses with full domains for that (Well, they look like any normal address for you, and we do not tell about the other way to not confuse you ;). If you want this, you can use this transport:

rmail:

```
debug_print ="T: rmail for $pipe_addresses"
driver=pipe
command = uux - -r -a $sender_address -gC $domain_data!rmail
$pipe_addresses
return_fail_output
user=uucp
batch_max = 20
```

However, all recipients are handled via the command line, so you are discouraged to use it.

2.10.1.2   bsmtp/rsmtp

This is a more efficient way to transfer mails. It works like sending SMTP via a pipe, but instead of waiting for an answer, the SMTP is just batched; from this is also the name batched SMTP or short bsmtp.

Furthermore, this way won't fail on addresses like " "@do.main. If you want this, please use this, if the remote site uses rsmtp (e.g. is Exim 4):

rsmtp:

```
debug_print ="T: rsmtp for $pipe_addresses"
driver=pipe
command = /usr/bin/uux - -r -a $sender_address -gC $domain_data!rsmtp
use_bsmtp
return_fail_output
user=uucp
batch_max = 100
```

and this if it wants bsmtp as the command:

bsmtp:

```
debug_print ="T: bsmtp for $pipe_addresses"
driver=pipe
command = /usr/bin/uux - -r -a $sender_address -gC $domain_data!bsmtp
use_bsmtp
return_fail_output
user=uucp
batch\_max = 100
```

Of course, these examples can be extended for e.g. compression (but you can also use ssh for compression, if you want).

2.10.1.3   The router

You need a router to tell Exim 4 which mails to forward to UUCP. You can use this one; please adopt the last line. Of course, it is also possible to send mail via more than one way.

uucp_router:

```
debug_print ="R: uucp_router for $local_part@$domain"
driver=accept
require_files= +/usr/bin/uux
```

28

```
domains = wildlsearch;/etc/exim4/uucp
transport = rsmtp
```

The file /etc/exim4/uucp looks like:
.do.main uucp.name.of.remote.side

2.10.1.4   Speaking UUCP with the smarthost

If you have a leaf system (i.e. all your mail not for your local system
goes to a single remote system), you can just forward all non-local mail
to the remote UUCP system. In this case, you can replace "domains = ..."
with "domains = ! +local_domains", but then you need also to replace
$domain_data in the transport by the UUCP-name of your smarthost. The
file /etc/exim4/uucp is not needed in this case.

2.10.2   Receiving mail via UUCP

2.10.2.1   Allow UUCP to use any envelope address

Depending how much you trust your local users, you might use
trusted_users and add uucp to it or use local_sender_retain=true and lo-
cal_from_check=false.

2.10.2.2   If you get batched smtp

Allow uucp to execute rsmtp via
commandsrmail rnews rsmtp
in your /etc/uucp/sys, and ask the sending site to use rsmtp (and not
bsmtp) as the batched command.

3   Updating from Exim 3

If you use exim4-config from Debian, you will get the debconf based con-
figuration scheme that is intended to cover the majority of cases.

If exim4-config is installed while an Exim 3 package is present on the
system, exim4-config tries to parse the Exim 3 config file to determine the
answers that were given to eximconfig on Exim 3 installation. These an-
swers are then taken as default values for the debconf based configuration
process. Be warned! eximconfig from the Exim 3 packages does not record

the explicit answers given on Exim 3 configuration. So we have to guess the answers from the Exim 3 configuration file /etc/exim/exim.conf, which is bound to fail if the config file has been modified after using eximconfig.

This is the reason why we refrained from doing a "silent update", but only use the guessed answers to get reasonable defaults for our debconf based configuration process.

Please note that we do not use the exim_convert4r4 script, but try to configure the Exim 4 package in the same way Exim 3 was. This will hopefully aid future updates.

If you have used a customized Exim 3 configuration, you can of course use exim_convert4r4, and install the resulting file as /etc/exim4/exim4.conf after careful inspection. Exim 4 will then use that file and ignore the file that it generated from the debconf configuration. To aid future updates, we do, however, encourage you not to use the exim_convert4r4-generated file verbatim but instead drop appropriate configuration snippets in their appropriate place in /etc/exim4/conf.d.

4   Misc Notes

4.1   PAM

PAM: On Debian systems the PAM modules run as the same user as the calling program, so they cannot do anything you could not do yourself, and in particular cannot access /etc/shadow unless the user is in group shadow. – If you want to use /etc/shadow for Exim's SMTP AUTH you will need to run exim as group shadow. Only exim4-daemon-heavy is linked against libpam. We suggest using saslauthd instead.

4.2   Account name restrictions

In the default configuration, Exim cannot locally deliver mail to accounts which have capitals in their name. This is caused by the fact that Exim converts the local part of incoming mail to lower case before the comparision done by the check_local_user directive in routers is done.

The router option caseful_local_part can be used to control this, and we decided not to set this option in the Debian configuration since it would be a rather big change to Exim's default behavior.

4.3   No deliveries to root!

No Exim 4 version released with any Debian OS can run deliveries as root. If you don't redirect mail for root via /etc/aliases to a nonprivileged account, the mail will be delivered to /var/mail/mail with permissions 0600 and owner mail:mail.

This redirection is done by the mail4root router which is last in the list and will thus catch mail for root that has not been taken care of earlier.

4.4   Debugging maintainer and init scripts

Most of the scripts that come with this Debian package do a set -x if invoked with the environment variable EX4DEBUG defined and non-zero. This is particularly handy if you need to debug the maintainer scripts that are invoked during package installation. Since dpkg redirects stdout of maintainer scripts, calling dpkg with EX4DEBUG set might yield interesting results. If in doubt, invoke the maintainer scripts with EX4DEBUG set manually directly from the command line.

4.5   SELinux

There is no SELinux policy for Exim4 available so far. Until this is resolved, users should use postfix or sendmail if they intend to run SELinux.

The Debian Exim4 maintainers would appreciate if somebody could write an SELinux policy. We will gladly use them in the Debian packages as long as there is somebody available to test, debug and support.

4.6   misc

- convert4r4 is installed as /usr/sbin/exim_convert4r4.

- The charset for $header_foo expansions defaults to UTF-8 instead of ISO-8859-1.

- Marc Merlin's Exim 4 Page[5] has a lot of ACL examples.

- For an example of Exim usage in a large installation, see Tony Finch's paper[6] about the Exim installation at University of Cambridge.

5 Debian modifications to the Exim source

Patches by Steve Haslam[9]:
boolean_redefine_protect [src/mytypes.h] Surround the definition of TRUE and FALSE macros with #ifndef /#endif, in case some other header defines them (from mixing No Perl and Exim, istr)

Other stuff

- link exim dynamically against pcre.

- The main binary is /usr/sbin/exim4:

    - src/globals.c was changed to use 'US BIN_DIRECTORY "/exim4"' as default for exim_path.

    - changed default for $exim_path (modulo lower/upper case) from BIN_DIRECTORY/exim to BIN_DIRECTORY/exim4 in exicyclog.src, exim_checkaccess.src, eximon.src, exinext.src, exiqgrep.src, exiwhat.src.

    - OS/Makefile-Linux:$EXIWHAT_MULTIKILL_ARG=exim4

- [8]localscan_dlopen .patch: Allow to use and switch between different local_scan functions without recompiling Exim. Use local_scan_path = /path/to/sharedobject to utilize local_scan() in /path/to/sharedobject.

- changes to the documentation to have the Debian-specific mailing list[9] mentioned where the official exim-users list[10] is mentioned

6 Credits

Andreas Barth[11]
UUCP documentation

Dan Weber, Ryen Underwood
inetd/xinetd documentation

References

[1] http://pkg-exim4.alioth.debian.org/

[2] http://lists.alioth.debian.org/mailman/listinfo/pkg-exim4-users

[3] http://wiki.debian.org/PkgExim4UserFAQ

[4] http://www.exim.org/bugzilla/show_bug.cgi?id=92

[5] http://marc.merlins.org/linux/exim/

[6] http://www-uxsup.csx.cam.ac.uk/~fanf2/hermes/doc/talks/2005-02-eximconf/

[7] http://www.arise.demon.co.uk/exim-patches/

[8] http://marc.merlins.org/linux/exim/files/sa-exim-current/

[9] mailto:pkg-exim4-users@lists.alioth.debian.org

[10] mailto:exim-users@exim.org

[11] mailto:aba@not.so.argh.org