Math SMTP Server Configuration

Šarūnas Burdulis

Version 1, August 3, 2005

Contents

1 Requirements for SMTP Service at Math

We need SMTP service, which meets the following criteria:

1. Supports spam and virus filters via preferably uniform interface. Scanning should be done at the early DATA stage of SMTP transaction, to allow for high-score spam and malware-infected messages to be discarded without generating bounce replies (collateral spam) and without further processing.

2. Flexible configuration for mail relay in combination with sender authentication.

3. Support for secure SMTP connections via TLS, preferably via STARTLS, i.e. using the standard TCP port 25 initially with unencrypted connection and then upgrading the existing connection to TLS mode.

After evaluating Exim, Postfix, Qmail and Sendmail, Exim 4 was selected. Postfix and Qmail use the so-called "sandwich" technique for passing mail through spam and virus filters: they listen on port 25, accept the message and forward it to the filter service listening on some other network port. Filter then resubmits the processed message back to the real SMTP, listening on some arbitrary port, which complicates the setup unnecessarily. Sendmail seems to meet most of the criteria, but the configuration is still cumbersome, if not ryptic. Exim 4 however, at least as packaged by Debian, meets our needs essentially with the default settings. It is also the default SMTP server in Debian.

2 Exim 4

Exim 4 is installed on `gauss` as from the following Debian packages: `exim4`, `exim4-base`, `exim4-config` and `exim-daemon-heavy`. The daemon package exists in several flavors and we need the `-heavy` variant, which already includes Exiscan (Exim patch providing interface to content filters) and support for TLS.

We use a single-file mode for Exim 4 configuration for simplicity and the configuration is in /etc/exim4/exim4.conf. Below are the modifications made to the default distribution file.

- Add gauss and its aliases to local_domains. The initial value was @, which expands into local host's name. The name local_domains is used later in the configuration to identify domains that are to be delivered on the local host.

  ```
  domainlist local_domains =
      @:gauss.dartmouth.edu:math.dartmouth.edu:www.math.dartmouth.edu:localhost
  ```

- Domain to be added to unqualified addresses (i.e. recipient root becomes root@gauss.dartmouth.edu):

  ```
  qualify_domain = gauss.dartmouth.edu
  ```

  We leave sender_unqualified_hosts and recipient_unqualified_hosts values empty, therefore unqualified addresses are accepted from local callers only, which is the default.

- Add gauss's external interface using IP number 129.170.28.37:

  ```
  local_interfaces = 127.0.0.1:129.170.28.37
  ```

- Allow SMTP relay from localhost, Bradley Hall and Choate House:

  ```
  hostlist relay_from_hosts = 127.0.0.1 : 129.170.28.0/24 : 129.170.147.0/24
  ```

3 Spam Filtering

Exiscan does content filtering by interfacing with third party utilities like SpamAssassin. We configure Exim to check all incoming mail for spam patterns at DATA stage, that is, after addressing rules were satisfied, all data received, but before starting the delivery. If high-score spam is detected, that message is rejected with SMTP response code 550 — it's now up to SMTP sender to decide what to do with the rejected message. All message reject actions (spam and malware, see below) are logged in /var/log/exim4/rejectlog. Spam detection itself is done by SpamAssassin (Debian package spamassassin). On system boot spamd daemon starts, listening on TCP 127.0.0.1:783 and this is were Exiscan sends message body for scanning. spamd configuration is in /etc/defaults/spamassassin. The only change from the distribution default is to comment out OPTIONS="-c -H" line — it tells spamd to create user preferences files, which would be confusing in case of messages with multiple recipients.

Below are the options which had to be added to /etc/exim4/exim4.conf to enable and configure Exiscan with SpamAssassin.

MAIN section. How to connect to spamd:

```
spamd_address = 127.0.0.1 783
```

ACL section, acl_check_data:

```
# high-score spam (score above 12.0), reject:
deny message = This message scored $spam_score spam points.
   spam = nobody:true
   condition = ${if >{$spam_score_int}{120}{1}{0}}
```

```
# all messages, add headers telling that message was scanned:
warn message = X-Spam-Checker: Spamassassin on $primary_hostname
   spam = nobody:true
warn message = X-Spam-Score: $spam_score ($spam_bar)
   spam = nobody:true

# identified as spam (but not high-score), add headers:
warn message = X-Spam-Report: $spam_report
   spam = nobody
warn message = X-Spam-Status: Yes
   spam = nobody
warn message = Subject: *** SPAM: $spam_score *** $h_Subject
   spam = nobody
```

Exiscan website [1] has detailed explanation of the syntax above as well as of all the options available. See exiscan-acl-spec.txt, exiscan-acl-examples.txt and HOWTO (Exim-SpamAndVirusScanning.pdf). All three files are also saved locally with this document.

The score required to trigger spam condition to true is set in /etc/spamassassin/local.cf:

```
required_score = 3
```

4 Virus Detection

All mail received by SMTP server is scanned for malware at DATA stage. This is done by using Exiscan to submit message contents to ClamAV virus scanner daemon. Of course the appropriate daemon (clamd) has to be running and listening on specified TCP port. clamd is part of the ClamAV anti-virus suite. The relevant packages in Debian are clamav, clamav-base, clamav-daemon and clamav-freshclam. The latter package is yet another daemon which is responsible for periodically (hourly) checking for and downloading virus signature updates. Configuration files for ClamAV are in /etc/clamav/. The configuration is quite simple, for example:

```
TCPSocket 3310
TCPAddr 127.0.0.1
User clamav
... ... ...
```

Then it's necessary to add user clamav to Debian-exim group:

```
# adduser clamav Debian-exim
```

The following is added to the MAIN section of /etc/exim4/exim4.conf to enable Exiscan-to-ClamAV communication:

```
av_scanner = clamd:127.0.0.1 3310
```

And the following is added to ACL section, just before spamd entries:

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

```
   demime  = *
   malware = *
warn message = X-Malware: Found clean by ClamAV on $primary_hostname
```

That is, all messages identified as malware are rejected by SMTP. Clean messages (i.e. the rest of them) are marked as ”found clean” by adding X-Malware header of our choice. See [1] and the documents mentioned in the previous section for more on Exiscan syntax and options regarding clamd.

5 Authentication and TLS

We want to enable SMTP relay on gauss for remote and/or mobile users. The relay should only be available to users, who can authenticate on gauss. Exim 4 supports quite a few authentication techniques and user credential sources (see “SMTP Authentication” in [2] for all the options available). We are interested in using 1) the current UNIX-style username and password database on gauss (/etc/passwd,shadow), which will let use the same username/password pair as for IMAP or SSH logins; 2) PAM, as this should make changing the username/password backend easier (for example, when introducing department-wide authentication via Math-LDAP). It appears that the optimum way to implement this is to use Simple Authentication and Security Layer (SASL): SASL daemon will do the communication between Exim and PAM, which in turn will use /etc/passwd. In addition we will need secure encrypted communication between the mail client and Exim in order to not expose authentication credentials in plain text. The latter is achieved by using Transport Layer Security (TLS).

PAM PAM is already in use on gauss and is configured to use /etc/passwd,shadow (use of pam_unix.so in /etc/pam.d/common*). No changes needed.

SASL Package in Debian Sarge is sasl2-bin. Configuration is in /etc/default/saslauthd, and PAM is the default authentication mechanism: MECHANISMS="pam". Accordingly, /etc/init.d/saslauthd starts saslauthd (SASL daemon) with the -a pam switch. saslauthd creates a local socket /var/run/saslauthd/mux for communication with other processes like Exim. Upon installation of SASL, system group account sasl is created too. We need to add user Debian-exim to this group:

```
# adduser Debian-exim sasl
```

Default configuration of Exim 4 in Debian already allows relaying for authenticated connections. The setting is in ACL section of exim4.conf:

```
accept authenticated = *
```

What we need to do in Exim 4 configuration, is to set the desired authentication mechanism as well as the condition that would prevent authentication if TLS connection could not be established. This is done by having the following in AUTHENTICATION CONFIGURATION section of exim4.conf:

```
# Authenticate against local passwords using sasl2-bin
# Requires exim_uid to be a member of sasl group, see README.SMTP-AUTH
plain_saslauthd:
   driver = plaintext
   public_name = PLAIN
   # don’t send system passwords over unencrypted connections
   server_advertise_condition = ${if eq{$tls_cipher}{}{0}{1}}
   server_condition = ${if saslauthd{{$2}{$3}}{1}{0}}
   server_set_id = $2
   server_prompts = :
```

Of course we have to “enable” TLS in Exim for TLS connections to be possible. Below are the relevant entries from the MAIN section of `exim4.conf`. Certificate for `math.dartmouth.edu` signed by the Dartmouth CertAuth1 as well as the corresponding private key should be put into `/etc/exim4/tls/` as it is obvious from the configuration (for more on certificates see [3]):

```
# advertise STARTTLS capability
tls_advertise_hosts = *

# Defines where your SSL-certificate and SSL-Private Key are located.
# This requires a full path. The files pointed to must be kept ’secret’
# and should be owned my root.Debian-exim mode 640 (-rw-r-----). Usually the
# exim-gencert script takes care of these prerequisites.
tls_certificate = /etc/exim4/tls/math.cert
tls_privatekey = /etc/exim4/tls/math.key

# A file which contains the certificates of the trusted CAs (Certification
# Authorities) against which host certificates can be checked (through the
# ‘tls_verify_hosts’ and ‘tls_try_verify_hosts’ lists below).
tls_verify_certificates = /etc/ssl/certs/ca-certificates.crt

# A weaker form of checking: if a client matches ‘tls_try_verify_hosts’ (but
# not ‘tls_verify_hosts’), request a certificate and check it against
# ‘tls_verify_certificates’ but do not abort the connection if there is no
# certificate or if the certificate presented does not match. (This
# condition can be tested for in ACLs through ‘verify = certificate’)
tls_try_verify_hosts = *
```

In addition we have `ident` callbacks disabled. If enabled, `ident` service should also be enabled on computer where SMTP client runs. If `ident` service is not enabled or blocked by firewall(s), making the callback will result in delay (until callback `ident` request timeouts), which appears to be confusing to users by hinting at “slow response/connection” etc.

```
# The settings below, which are actually the same as the defaults in the
# code, cause Exim to make RFC 1413 (ident) callbacks for all incoming SMTP
# calls. You can limit the hosts to which these calls are made, and/or change
# the timeout that is used. If you set the timeout to zero, all RFC 1413 calls
# are disabled. RFC 1413 calls are cheap and can provide useful information
# for tracing problem messages, but some hosts and firewalls have problems
# with them. This can result in a timeout instead of an immediate refused
# connection, leading to delays on starting up an SMTP session.
rfc1413_query_timeout = 0s
```

After Exim 4 has been restarted with the new configuration, we can (and should) test it by imitating SMTP client connection. For instance, do `telnet gauss 25` and then, after Exim on `gauss` responds with “220...”, type `ehlo` ***`yourhostname`***:

```
hilbert:˜$ telnet gauss 25
```

```
Trying 129.170.28.37...
Connected to gauss.dartmouth.edu.
Escape character is ’ˆ]’.
220 gauss.dartmouth.edu ESMTP Exim 4.50 Wed, 03 Aug 2005 14:46:34 -0400
ehlo hilbert
250-gauss.dartmouth.edu Hello hilbert.dartmouth.edu [129.170.28.34]
250-SIZE 52428800
250-PIPELINING
250-STARTTLS
250 HELP
```

The response from Exim should include `250-STARTTLS` (advertising the capability of the SMTP server to upgrade the existing plain text connection to TLS), but there should be no lines advertising authentication possibilities, as we only allow authentication if, and only if, the connection was upgraded into secure, encrypted TLS connection.

E-mail client configuration should be set to use `math.dartmouth.edu` as SMTP server, port 25, SMTP authentication enabled, username — the same as on `gauss`, TLS required. This is only needed when outside Choate and Bradley networks, but will work anywhere, including Choate and Bradley, which is why it is convenient to use on a laptop, for example.

When sending e-mail while SMTP authentication enabled, user should be prompted for their password on `math.dartmouth.edu` (or whatever name was used for `gauss` in outgoing server configuration; if it is not `math.dartmouth.edu`, there will also be a warning of server name mismatch with the one used in server certificate). Such an e-mail, when received, should include the following headers, indicating the use of TLS:

```
Received: from hilbert.dartmouth.edu ([129.170.28.34]:53191)
        by gauss.dartmouth.edu with esmtpsa
        (TLS-1.0:DHE_RSA_AES_256_CBC_SHA:32)
        (Exim 4.50) id 1E0OaC-0004IM-EO for Sarunas.Burdulis@Dartmouth.edu;
        Wed, 03 Aug 2005 15:07:16 -0400
```

The corresponding `/var/log/exim4/mainlog` entry should indicate authentication and the username used (`A=plain_saslauthd:`*`username`*):

```
2005-08-03 15:07:16 SMTP connection from [129.170.28.34]:53191 I=[129.170.28.37]:25
        (TCP/IP connection count = 1)
2005-08-03 15:07:16 1E0OaC-0004IM-EO <= sarunas@math.dartmouth.edu
        H=hilbert.dartmouth.edu [129.170.28.34]:53191 I=[129.170.28.37]:25 P=esmtpsa
        X=TLS-1.0:DHE_RSA_AES_256_CBC_SHA:32 CV=no DN="" A=plain_saslauthd:sarunas
        S=1553 id=42F115BF.2060800@math.dartmouth.edu T="demo0"
        from <sarunas@math.dartmouth.edu> for Sarunas.Burdulis@Dartmouth.edu
2005-08-03 15:07:16 SMTP connection from hilbert.dartmouth.edu [129.170.28.34]:53191
        I=[129.170.28.37]:25 closed by QUIT
```

While for the non-authenticated case corresponding header and log entries would be:

```
Received: from hilbert.dartmouth.edu ([129.170.28.34]:57487)
     by gauss.dartmouth.edu with esmtp
     (Exim 4.50) id 1E0OfW-0004LN-8l for Sarunas.Burdulis@Dartmouth.edu;
     Wed, 03 Aug 2005 15:12:46 -0400
```

and

```
2005-08-03 15:12:46 SMTP connection from [129.170.28.34]:57487 I=[129.170.28.37]:25
        (TCP/IP connection count = 1)
2005-08-03 15:12:46 1E0OfW-0004LN-8l <= sarunas@math.dartmouth.edu
        H=hilbert.dartmouth.edu [129.170.28.34]:57487 I=[129.170.28.37]:25 P=esmtp
        S=1514 id=42F1170C.2000703@math.dartmouth.edu T="demo0"
        from <sarunas@math.dartmouth.edu> for Sarunas.Burdulis@Dartmouth.edu
2005-08-03 15:12:46 SMTP connection from hilbert.dartmouth.edu [129.170.28.34]:57487
        I=[129.170.28.37]:25 closed by QUIT
```

References

[1] Exiscan website.

[2] Exim 4.40 Specification.

[3] Sysadmin doc “Math Servers’ Certificates” in
`gauss:/usr/local/share/doc/sysadmin/certificates/`.