TmdaFaq

Table of Contents

Table of Contents

TmdaFaq

TMDA Frequently Asked Questions

NOTE: don't delete obsolete entries because it will break the numbering that people refer to the FAQ entries by. Just strikethrough them but leave them in place.

Contents

General questions

Can't spammers just setup an auto-responder to defeat TMDA?

In theory yes, but in practice this is not likely to happen. Most SPAM is unrepliable, so TMDA's confirmation requests are never delivered to them. They use non-valid return addresses as to not incur the cost of the tremendous number of bounces they generate. Using a valid return address to process all the bounces looking for confirmation messages to auto-reply to would defeat their economies of scale. It would also make them easy to block, track down and report, sue, etc.

In short, trying to thwart TMDA in this manner would defeat the cost-effectiveness of the bulk-mailing process. Simple economics keep us safe.

But should these facts change, TMDA could modify its (currently very simple) challenge/response to make it more difficult for a computer to auto-reply to. The level of difficulty could increase as much as is necessary for the sender to prove their humanity and legitimacy.

The idea is to keep the challenge/response as simple as possible to avoid inconveniencing legitimate senders, while at the same time difficult enough to thwart an automated response system. At the present time, TMDA offers the ability to confirm by a simple e-mail reply, or by clicking on a URL. There is no evidence to suggest that a more challenging procedure is even close to necessary.

If spammers do resort to auto-confirmation however, we've won a huge battle for the community by raising the bar and forcing them to leave a breadcrumb trail leading back to their lair. As the focus on legislation continues, and spamming becomes an increasingly illegal activity, who will take these great risks for such a little reward?

Why isn't TMDA written in Perl, C, C++, or Java?

TMDA is written in Python, which is a great prototyping language because development speed is very high, the standard library is very rich, and there are many 3rd party modules available. For TMDA this means we can implement any feature that we think of in a relatively short amount of time, get feedback, make changes, etc. This is great for trying out a large number of features, approaches, etc., to see what works, what doesn't, what is necessary, and what isn't in a new application. While TMDA-like programs are now popping up all over the place, remember that TMDA was the first of its kind, so there were no other applications to learn from initially.

Python is also a safe language. For example, unlike C or C++, buffers in Python are dynamically sized, so TMDA is not vulnerable to buffer overflows and all the accompanying security problems.

The question of peformance and scalability has often been raised due to the fact that Python is an interpreted language and thus executes slower than one with a native code compiler like C, C++, or Objective Caml. The fact is that most scalability or performance problems can be resolved through configuration changes (see FAQ 7.7 for example). In fact, some sites have reported running over 50,000 messages/day through TMDA without any problems. So in short, it doesn't make sense to work in a lower-level language and lose Python's productivity and reliability benefits.

Any plans to support Microsoft Exchange Server?

I'm not adverse to TMDA supporting Microsoft Windows, but someone else will have to do the work. So far, noone has stepped up to the plate.

For me, Windows is not a pleasant platform for software development. Mail server software such as Exchange Server costs $$, and it's not cheap. Most importantly, it isn't OpenSource, so it's like a black box which makes things extremely difficult when trying to debug complex problems.

Does TMDA have a web interface?

Yes, tmda-cgi from Gre7g Luterman and friends. See http://tmda.net/tmda-cgi/

Won't senders just refuse to confirm their messages?

A common misconception about systems based on whitelisting with challenges is that few of us are so irresistible that people will jump through extra hoops to talk to us.

In actual practice (5 years of using TMDA), I've not found this to be the case at all, and most TMDA users will agree. If someone took the time to craft and then send you a message, they will most likely take a few extra seconds to reply to a one-time confirmation request. Phrased another way, by writing to you, the sender also fulfills a need of their own---it's not purely to your benefit that you receive their message. If they refuse to confirm simply out of spite, that need goes unfulfilled (which is why this is so rare).

Once upon a time, people had the same sort of objections about authenticating a mailing list subscription; something we now take for granted. TMDA's confirmation mechanism is really no different.

In any case, TMDA provides several different ways to monitor who sent you mail, who didn't reply to their confirmation request, etc (see next FAQ entry).

Also, legitimate senders are faced with a TMDA confirmation request less than you might think. For one, a well constructed initial whitelist will encompass the vast majority of your correspondents. Second, TMDA has facilities to reduce this even further. For example, use of 'dated' addresses in your outgoing messages which allow anyone through until it expires. FAQ 5.5 details two different methods to insure that responses to your messages will not be challenged. In other words, if you are using TMDA properly, a challenge should only be issued for a new correspondent who is e-mailing you out of the blue. That is, 1) the sender is someone whom you've never received a message from before, and 2) their message is *not* in response to a message you've sent.

As proof, using both a whitelist (with auto-update) and the 'dated' address feature meant that only 6% of my legitimate corresponents had to confirm their messages to me. I think that's quite acceptable.

So in summary, most of your correspondents will not even see a confirmation request, and when they do, they will reply to it. I'm much more comfortable with this scenario than with a classifier which accidentally trashes a message it "thinks" is spam. At least with TMDA, the sender has the choice whether or not to let his message be delivered to you. With a classifier, he will assume it has been delivered when it has not been, which opens a communications rift.

How do you know when someone didn't confirm their message?

Although rare, there may be cases when someone can't (or won't) reply to your confirmation request, and you don't want to lose that message. TMDA provides several ways around this dilemma.

TMDA includes comprehensive logging of all incoming mail which can be parsed (by script or by eye) for such messages.

For the more paranoid, it also includes a "pending queue" browser that lets you manipulate your unconfirmed messages in different ways. Print summaries, view/delete/release/show/blacklist/whitelist, etc.

You can even have a copy of each unconfirmed message forwarded to another address.

So basically there is something for every level of trust or distrust.

Under which license is TMDA released? I'd like to use it in a commercial application.

TMDA is released under the GNU General Public License. For more about the GPL, see http://www.gnu.org/licenses/gpl.html

The copyright notice included with the code looks like:

```
# Copyright (C) 2001-2006 Jason R. Mastaler <jason@mastaler.com>
#
# This file is part of TMDA.
#
# TMDA is free software; you can redistribute it and/or modify it
# under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version. A copy of this license should
# be included in the file COPYING.
#
# TMDA is distributed in the hope that it will be useful, but WITHOUT
# ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
# FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License
# for more details.
#
# You should have received a copy of the GNU General Public License
# along with TMDA; if not, write to the Free Software Foundation, Inc.,
# 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA
```

What version of TMDA does the FAQ address?

The FAQ attempts to track the latest released version of TMDA, which currently is from the v1.1.x series.

Are there any other TMDA-like systems available?

If for whatever reason TMDA doesn't fit your needs, here are some other packages that provide similar functionality. Please let us know if you come across any others.

- Active Spam Killer.
- QuarantineMail (for Microsoft Outlook, commercial product).
- Qurb (for Microsoft Outlook, commercial product).
- Matador (for Microsoft Outlook, commercial product).
- ChoiceMail (for Microsoft Windows, commercial product).
- JunkJam (for Microsoft Windows, commercial product).
- Spam Arrest (commercial service).
- SpamCop Email (commercial service).
- qconfirm (qmail only).
- oSpam (qmail only).
- spamgard (Requires procmail).
- qsecretary from D. J. Bernstein (not yet released).
- AntiSpamSystemExtension.
- Mail::Address::Tagged Perl module.

Can TMDA be used in conjunction with other countermeasures such as RBL, Razor, Spamassasin, etc.?

(also see http://wiki.tmda.net/SpamAssassin)

The short answer to this question is, "Yes." But the question is frequently asked in other terms:

```
"TMDA seems to allows SPAM to get all the way into my mailserver. So instead of sitting in my mai
email earlier so that SPAM consumes less resources. Can TMDA do this?"
```

TMDA does not prevent you from using other spam prevention techniques. TMDA relies on a working Mail Transfer Agent (MTA) like qmail, postfix, exim or sendmail. So any of the other spam prevention techniques that work with the MTA can be employed in conjunction with TMDA.

I personally, use Realtime BlackLists (RBLs) configured into my postfix MTA. Any email that gets past the RBLs will be delivered to spamassassin. This setup, alone, works pretty well. However when I was relying on these techniques alone, an annoying amount of spam would still reach my mailbox. That's when I deployed TMDA as the last stage in my mail delivery process. TMDA catches everything that the RBLs and spamassissin allow through. In this situation, using TMDA does not actually consume any more disk space or bandwidth than not using TMDA since the other prevention techniques allow the SPAM through.

The general answer to this question is that TMDA can be deployed in addition to any SPAM prevention techniques that are already in use with your MTA or through procmail. Doing so will give you all of the resource benefits of other techniques but the greater SPAM blocking effectiveness of a whitelist-centric strategy.

*** NOTE FOR SPAMASSASSIN USERS: spamassassin has a habit of marking email sent to TMDA tagged addresses as spam. This is most troublesome when it happens on a reply to a confirmation request. The solution is to add some local tests to lower the spamlevel on email to TMDA tagged addresses. This is done by adding the following to ~/.spamassassin/user_prefs:

```
header TMDA_CONFIRM To =~ /-confirm-/
describe TMDA_CONFIRM To a TMDA confirm address
score TMDA_CONFIRM -4.00

header TMDA_DATED To =~ /-dated-/
describe TMDA_DATED To a TMDA dated address
score TMDA_DATED -4.00

header TMDA_SENDER To =~ /-sender-/
describe TMDA_SENDER To a TMDA sender address
score TMDA_SENDER -4.00

#header TMDA_KEYWORD To =~ /-keyword-/
#describe TMDA_KEYWORD To a TMDA keyword address
#score TMDA_KEYWORD -4.00
```

I've commented out the spamassassin rules for lowering the score on TMDA keyword addresses. Since anyone can use that kind of address, I still want spamassassin to take a guess at whether or not it thinks the email is spam. I provide the rules here for reference. If you want to use them just remove the # at the beginning of each line.

Also the above rules assume that you're using the default TMDA configuration for TAGS_CONFIRM, TAGS_DATED, TAGS_KEYWORD and TAGS_SENDER. If you've modified these variables, then you should change the rules above to make sure that they match your settings.

What software did you use to generate this FAQ?

MoinMoin

Why the funny looking e-mail addresses? (or, advantages of tagged messages)

One of the most common objections raised by those evaluating TMDA is that they don't like TMDA's tagged e-mail addresses because they "look funny". e.g,

jason-dated-1032297526.da8e7f@mastaler.com

instead of just:

jason@mastaler.com

Can TMDA be used in conjunction with othercountermeasures such as RBL, Razor, Spamassasin, etc.?

Indeed, if tagged messages didn't provide some significant advantages, there would be no need for them. Many whitelisting systems include the "cookies" within the Subject: header of the message instead of directly in the address like TMDA does.

Consider these advantages of the tagged address method:

- Easier and more reliable parsing of the encoded "cookies". The message will get processed correctly upon reply regardless of whether the Subject: header remains intact or not.
- Significant reduction in the number of legitimate senders who are actually faced with a confirmation request (see FAQ 1.5).
- Senders who respond to a message you send them won't have to confirm their reply, which many consider rude (see FAQ 5.5).
- Messages that you send which end up undeliverable (aka bounces) will be delivered to you and will not be quarantined by TMDA (see FAQ 5.4).

None of these advantages are available without tagged addresses leading to a much less flexible system. What do you care more about, aesthetics or functionality?

Also, when displaying a message, many mail clients (such as MS Outlook) only show the full name of the sender (e.g, Jason R. Mastaler) and hide the e-mail address itself. So, it's likely that many of your recipients won't even know that you are using a tagged address.

Also, consider why you think a tagged address looks "funny". It's simply because this is something you are not used to seeing. If tagged messages became some sort of standard, they would no longer seem that way. In any case, your identity remains intact regardless of what your e-mail address looks like, which is what's important. People change jobs, graduate from university, lose domain name disputes, and their e-mail address changes -- but their identity remains intact. I'm `Jason R. Mastaler' regardless of whether my e-mail address includes some extra characters in it or not.

Remember, you are not your e-mail address. And while I'm at it: You are not your job. You are not how much money you have in the bank. You are not the car you drive. You are not the contents of your wallet. You are not your fucking khakis. You are the all singing, all dancing, crap of the world.

If I still haven't convinced you, TMDA does provide a way to lessen the "ugliness" of tagged addresses. Instead of tagging your "From:" header, some users prefer to tag "Reply-To:" instead and leave "From:" intact. This preserves your e-mail identity yet still allows the full advantages of tagged messages. The headers of such a message might look like this:

```
From: "Jason R. Mastaler" <jason@mastaler.com>
Subject: Re: Vegas Baby
To: bobby@peru.com
Date: Wed, 02 Oct 2002 11:54:27 -0600
Reply-To: jason-dated-1034271964.56150d@mastaler.com]
```

TMDA's outgoing filter file can fully automate this tagging process.

What do I do when a spammer does confirm their message?

First, as documented, TMDA is a SPAM _reduction_ system; there is no way to completely eliminate spam other than severing your connection to the Internet with a scissors.

Occasionally, you will find that a piece of spam is confirmed. This is usually the result of a bounce message going to the confirmation address, telling you that the spammer's address doesn't exist or is over quota or something.

This happens so rarely, that I tend to ignore it. Remember that the idea is to keep the confirmation process as simple as possible for the benefit of legitimate senders, while still eliminating most spam. See FAQ 1.1 for more on this point.

If the confirmation was actually done by the spammer, a positive thing about this phenomenon is that you may now have a working contact address for them. The two useful things you can do with that information are A) report the spam message to Spamcop so that the spammer's ISP can get in touch with them about their activities, and B) blacklist the address in your TMDA config. Make sure you remove them from your whitelist, and add them to a blacklist.

If you are extremely sensitive to spam, TMDA provides a couple methods to make it difficult if not impossible for a spammer to use an auto-responder to confirm messages.

1. Edit confirm_request.txt and remove the Reply-To line. Now there is no address anywhere that an auto-responder can use to release the message. This means legitimate senders will have to pick out the confirm address from the body of the message instead of just hitting "Reply", so it will be slightly more inconvenient for them, but still not bad. See the TMDA Template HOWTO (http://tmda.net/howto-template.html) for detailed instructions on how to customize confirm_request.txt.
2. You can use tmda-cgi's web confirmation system, which allows you to imbed a URL in confirm_request.txt that when clicked on, will release the message. This is an alternative to the traditional e-mail confirmation method. See http://tmda.net/tmda-cgi/confirm.html for more information.

More difficult confirmation methods such as using graphics files with embedded words have been suggested, but "eye-candy" of this sort is vast overkill at this point. There isn't any evidence that something like this is even close to necessary. It's also potentially harmful--the more difficult you make the challenge, the more likely it is that a legitimate sender is not going to confirm their message.

Why does TMDA run on the server instead of my mail client?

I'm often asked about why TMDA is run on the server. That is, why TMDA was designed to work with the mail server software on the host that receives your incoming mail.

This is opposed to software which plugs into your mail client (e.g, Outlook) or a pop3 "proxy" which the end user can operate. In this scenario, TMDA would be run *after* you download your messages rather than before.

The answer is that I don't feel the client is the correct place for TMDA to operate.

For one, it delays sending of the confirmation requests since they would only be sent periodically when you download your mail rather than as soon as a message is received on the server. This can result in significant delays as to when you will see the confirmed messages. For example, if a sender in a distant timezone sends you a message just after you've retired for the evening, it might be 10-12 hours before you fire up your mail client for the next time, meaning 10-12 hours before that sender sees a confirmation request. By that time, she has retired for the evening, meaning an additional 10-12 hours before she can reply to the confirmation request.

This is silly. If TMDA resides on the server, the confirmation process happens instantaneously and automatically, regardless of whether you are available at that particular time or not.

The second reason is that it's a waste of time and bandwidth for the client to download spam messages just so they can be processed. With TMDA running on the server, the client only has to download legitimate messages. If you receive a large amount of spam, this can result in significant savings in time and cost (if you pay for your bandwidth by the byte).

If you don't have access to your mailhost, your ISP might have to set TMDA up for you. If that isn't possible, then switch to an ISP that does allow you to set TMDA up yourself. Afterall, it's your money!

That said, See FAQ 1.9 for a list of similar products, many of which do operate at the client level.

What's with those goofy release names?

Each TMDA release has a "codename" in addition to a version number. This serves no purpose other than to add colour to the development process. See the CODENAMES file in the distribution for more information.

What about challenge/response mail loops?

TMDA is not vulnerable to this. See FAQ 4.12 for a complete answer.

Setup and installation

I'm getting errors when running 'compileall' on my RedHat Linux system

Make sure you are running the minimum required version of Python (at least v2.3).

Mail Transport Agent issues

Why don't my user+detail addresses work with Sendmail virtualdomains?

If you are running Sendmail, you may have noticed that your user+detail addresses don't work if the message is sent to a virtual domain.

You need some special entries in the virtusertable (and a modern version of sendmail - at least 8.10.1 to get full functionality) to preserve the +detail portion of the address. For example, if you currently have the following in your virtusertable:

```
bobby@peru.com bobby
```

you can change this to:

```
bobby+*@peru.com bobby+%2
```

See the virtusertable section in your Sendmail's cf/README for the full details.

Why don't my user+detail addresses work with Sendmail /etc/aliases?

If you're running Sendmail, you may have noticed that your user+detail addresses don't work if the message is sent to a an alias.

If you have the following in /etc/aliases:

```
webmaster: johndoe
```

+detail is preserved for mail to johndoe+detail, but not for webmaster+detail.

Unfortunately, there is no way to preserve the +detail portion in /etc/aliases. You must use the virtusertable for this (see previous question).

I'm getting "553 sorry, that domain isn't in my list of allowed rcpthosts (#5.7.1)"

If you are seeing this message in your LOGFILE_DEBUG file, you are running a very old version of TMDA. Try upgrading.

Exim is rewriting return-path when BOUNCE_ENV_SENDER is set.

If Exim3 does not find a valid user from the specified sender when TMDA feeds it the confirmation message via SMTP it will rewrite the return-path headers with the reply-to confirmation address. This will cause

bounces to confirm.

If you've set your bounce sender to something like:

BOUNCE_ENV_SENDER = " spamdump@mydomain.com "

then the user "spamdump" needs to exist on the system.

You can cause Exim to drop, or do other things with, the bounced returns using an Exim system filter.

What is recommended for RECIPIENT_DELIMITER, '-' or '+'?

The two most popular recipient delimiters are:

```
'+', as in jason+foobar@mastaler.com
```

and

```
 '-', as in jason-foobar@mastaler.com
```

qmail's default is '-', while Sendmail's is '+'. Exim and Postfix do not come with a default, but allow the administrator to choose.

Unfortunately, not all recipient delimiters are created equal. On the tmda-users mailing list, there have been several reports of certain mail clients that can't reply to an address containing a '+'.

- AOL Webmail client.
- Lotus Notes.
- MSN.

There have been no such problems reported by users using '-' as their recipient delimiter. Therefore, if it's in your control, '-' is recommended. qmail, Postfix and Exim make this configurable, while Sendmail does not.

How do I establish a secure connection between TMDA and my SMTP server?

If your SMTP server is capable of using the starttls protocol, then you can use TMDA's SMTPSSL configuration variables to secure the connection between TMDA and the SMTP server.

In either /etc/tmdarc or in user specific ~/.tmda/config configuration files, set the following three variables.

- SMTPSSL = 1
- SMTPSSL_CERTFILE = "/path/to/certificate/file"
- SMTPSSL_KEYFILE = "/path/to/private/key/file"

The certificate and key can be in separate files or in the same file. You still need to set all three configuration variables in the latter case. Just point SMTPSSL_CERTFILE and SMTPSSL_KEYFILE to the same file.

Any user running TMDA will need to have read access to both the certificate and key.

In cases where mail accounts are virtual and are controlled by a single sytem account, the certificate and key files should be owned by the controlling system account with permissions 400.

In cases where multiple system accounts will run TMDA, one option is to create a supplementary group to which all TMDA users belong, and make the certificate and key readable by this group. Another option is for each TMDA user to have their own certificate and key.

I'm getting SMTPRecipientsRefused or SMTPSenderRefused errors

Make sure you're running a recent version of TMDA.

Can I use TMDA with a qmail relay?

Sure, here's how.

This FAQ assumes three things:

1. Mail comes in from the outside Internet.
2. Mail passes through TMDA to check for whitelists/blacklists.
3. If permitted (or confirmed), the mail is forwarded to a separate server.

Create a directory to be used as a holding area for TMDA, for example /var/spool/tmda . Add a new user to the system to handle incoming mail, and create the users home directory as the holding area. This FAQ will use 'tmda' as the account name. Be sure that the holding area is writeable only by the 'tmda' user.

- ```
  mkdir /var/spool/tmda
  useradd -d /var/spool/tmda tmda
  ```

  ♦ Now add an entry to your /var/qmail/control/virtualdomains file like this:
  ```
  example.com:tmda
  ```

This will send all mail for example.com to the tmda user, so that it can be handled by /var/spool/tmda/.qmail-default . Also add example.com to your /var/qmail/control/rcpthosts file.

Send qmail-send a HUP so that it re-reads virtualdomains. Now, set up /var/spool/tmda/.tmda/config however you like, and be sure to create a new crypt_key:

- ```
  tmda-keygen -b > /var/spool/tmda/.tmda/crypt_key
  chmod 600 /var/spool/tmda/.tmda/crypt_key
  ```

  ♦ Add the following to the /var/spool/tmda/.qmail file:
  ```
  |preline -f /usr/local/tmda/bin/tmda-filter
  |forward "$DEFAULT"@exchange.example.com
  ```

If exchange.example.com is not in DNS then you will need to add it to your smtproutes files. See the qmail-remote man page for details.

This also assumes that the mail server at exchange.example.com will accept mail to addresses @exchange.example.com just as it will for @example.com. If not then you can try some DNS tricks to get this to work.

Create a symbolic link from .qmail to .qmail-default

- `ln -s /var/spool/tmda/.qmail /var/spool/tmda/.qmail-default`

    - Finally, clean up the permissions on the /var/spool/tmda directory so that TMDA has permission to create any necessary files:

```
chown -R tmda /var/spool/tmda
chmod 700 /var/spool/tmda
```

Any messages sent to user@example.com will now be forwarded through to user@exchange.example.com after TMDA has validated the sender.

Why do confirmations trigger new requests under Sendmail?

You probably have an entry in your /etc/aliases for your username (see FAQ 3.2). A common mistake is to have "foo+*: foo". This used to be necessary with some older versions/configurations of sendmail, but isn't with current versions; you can just delete that line. If it's there, sendmail rewrites "foo+confirm-..." to "foo" before passing it off to procmail, so procmail has no chance of seeing the part after the "+".

TMDA server-side issues

How do I setup a whitelist?

You can allow senders, or groups of senders directly into your mailbox by adding lines to your incoming filter file. By default this is ~/.tmda/filters/incoming.

For example, the following line will allow both anyone@domain.dom and anyone@sub.domain.dom into your mailbox:

- ```
  from *@=domain.dom ok
  ```

If you have many individual addresses and/or expressions you'd like to whitelist, you may keep them in a seperate file and then have TMDA access it using the following line:

- ```
  from-file ~/.tmda/lists/whitelist ok
  ```

  - ~/.tmda/lists/whitelist would contain e-mail addresses and/or wildcard expressions, one per line. e.g,
    ```
    [mailto:king@grassland.com king@grassland.com]
    *@myisp.net
    *@cs.myuni.edu
    *@=mycompany.com
    bobby*@peru.com
    ```

See the TMDA Filter Specification for details on filter file syntax, options, and more examples. The collectaddys script in the contrib directory can be used to build an initial whitelist from your existing mail archives.

How do I setup an "auto-whitelist"?

TMDA has the ability to automagically add confirmed addresses to a whitelist. This way, each new sender only has to go through the confirmation process once.

If you want that behavior, set the CONFIRM_APPEND variable in your ~/.tmdarc to point to a file, something like:

- ```
  CONFIRM_APPEND = os.path.expanduser("~/.tmda/lists/whitelist_confirmed")
  ```

  - and add a rule to your incoming filter:
    ```
    from-file ~/.tmda/lists/whitelist_confirmed ok
    ```

Then TMDA will the add the senders from successfully confirmed messages to the 'whitelist_confirmed' file. A possible variation on this includes setting CONFIRM_APPEND to your main whitelist file rather than a secondary file.
    ```

How can I manage messages in my pending queue?

TMDA comes with a utility called 'tmda-pending' which can be used to release, delete, view, and generally manipulate messages in your pending queue. You can run it by hand, or periodically from cron. Run *tmda-pending -h* to get a listing of available options and usage examples. tmda-pending should be run by the user account that owns the pending queue, not by root (unless root is running TMDA).

There is also a web-utility called 'tmda-cgi' for managing your pending queue (among other things).

If you'd prefer an e-mail interface for pending queue management, check out tmda-ezplm (http://www.pongonova.net/tmda-ezplm).

Additionally, your can make your pending queue a Maildir, which, as an industry standard format, allows you to use any program that supports Maildir to browse, search, index, etc, your pending queue. For more information, see TmdaPendingAsMaildir.

The most common question about the pending queue is how old messages get purged. As of TMDA 0.77, this is done automatically for you with no intervention required.

The default cleanup interval is approximately once per 100 incoming messages received. To tweak this interval, or disable this feature entirely, see PENDING_CLEANUP_ODDS. When this feature is enabled, the default lifetime for a message in the pending queue is 14 days. Any message older than 14 days will be deleted. If you wish to raise or lower this threshold, see PENDING_LIFETIME

Here's an example of using tmda-pending in "batch" mode to write scripts which will manipulate your pending queue.

```
tmda-pending -Cbs -O 48h | /usr/bin/mail -s "TMDA new in pending" \
  you@yourdomain.dom > /dev/null 2>&1
```

What this will do is print a summary of any emails that have been in pending for 48h to see if an email has turned up in pending and if it hasn't been confirmed w/in 48h. Additionally, because of the "-C" option, each new email will only be shown once.

How do I allow my ezmlm mailing lists to pass through TMDA?

ezmlm sends out each list message with a different envelope sender address.

You can whitelist the ezmlm list using wildcard characters. Here is an example whitelist entry for the qmail mailing list:

- `qmail-return-*@list.cr.yp.to`

This will allow messages to get through when initially interacting with ezmlm to get subscribed, as well as after when list messages are delivered.

How do I prevent my auto-whitelist from being filled with 'dated' addresses?

If you are using TMDA's CONFIRM_APPEND feature to implement an "auto-whitelist", you may have noticed that it is being filled with the 'dated' addresses of other TMDA users, which isn't very helpful. This can be disconcerting to the other party who must confirm each of their messages to you.

Recent versions of TMDA recognize the *X-Primary-Address* header in incoming messages. See http://tmda.net/config-vars.html#PRIMARY_ADDRESS_MATCH for more on this feature.

So, TMDA users should configure their MUA to add an X-Primary-Address address field to their outgoing messages specifying the address they prefer be added to other user's whitelists. e.g,

X-Primary-Address: jason@mastaler.com

If you use tmda-sendmail or tmda-ofmipd to send your outgoing mail, you can do this with an ADDED_HEADERS_CLIENT entry in your ~/.tmda/config. e.g,

```
ADDED_HEADERS_CLIENT = {'X-Primary-Address' : 'jason@mastaler.com'}
```

Now, jason@mastaler.com will be whitelisted upon confirmation, whether I'm using a 'dated' address or not.

Which address headers are examined when looking for a filter match?

For each incoming message, TMDA compares addresses in the message to rules in your FILTER_INCOMING file (e.g, ~/.tmda/filters/incoming) to try and find a match.

for from* rules, all addresses in the "From:" and "Reply-To:" headers are matched, as well as the envelope sender address (often stored in Return-Path, but TMDA gets it from the environment variable $SENDER).

For to* rules, only the envelope recipient is matched. Some MTAs put that in Delivered-To, but others don't. TMDA gets this from the environment variable $RECIPIENT.

How do I prevent my postmaster from getting all my bounced confirmation requests?

Since the vast majority of spam is sent from a non-working address, most of your confirmation requests will be returned right back to you as undeliverable. Since TMDA's confirmation requests are sent out with a null '<>' Return-Path by default, your local postmaster will probably receive these bounces.

If you want to avoid this, the trick is to set BOUNCE_ENV_SENDER to something that goes nowhere (e.g, thl-spambounce@catseye.net). Then, when TMDA generates the confirmation request, it uses that address as the envelope sender instead of '<>'.

You'll also need a rule in your incoming filter that says:

```
to thl-spambounce@catseye.net drop
```

so that any mail that bounces (and therefore gets returned to thl-spambounce) will be automatically dropped.

See http://tmda.net/config-vars.html#BOUNCE_ENV_SENDER

How do I use TMDA with mailing lists?

Most of us have stopped integrating TMDA with mailing lists.

TMDA does provide the ability to accept mailing list messages as well as deliver them in a number of different ways, and to multiple locations, sort of like a scaled down procmail. However, if you aren't using this advanced functionality, I wouldn't bother.

Filtering your list traffic with TMDA will just necessitate extra time fiddling with your whitelist and/or creating keyword/sender addresses, and possibly cause delay in your seeing the messages if they end up stuck in your pending queue because of a misconfiguration.

Since TMDA doesn't filter based on content, it can't help you with spam on the list anyway. Once the spam gets distributed to the list, it's now just another list message as far as TMDA is concerned.

Instead, either use Gmane, or subscribe to lists with extension addresses that aren't handled by TMDA. I use the <user>-list-<whatever>@foo.dom convention. If you use procmail or maildrop, have your list traffic delivered before TMDA sees it.

That said, if you still want to mix TMDA with your mailing lists, here are some documents on the subject that might help.

- Tim Legant's Cookbook for using TMDA with mailing lists
- Andrew St. Jean's TMDA & Mailing Lists HOWTO

How can I use customized templates for TMDA's auto-responses?

See the TMDA Template HOWTO at http://wiki.tmda.net/TmdaOfmipdHowto

SPAM is getting in by spoofing my domain.

Occasionally you will receive spam coming from your own e-mail address, or an address within your domain. e.g,

```
Date: Sun Nov 9 20:21:11 CST 2003
From: "al" <jason@mastaler.com>
To: jason@mastaler.com
Subj: Get Free shipping! Curn like a porno star!
Actn: CONFIRM action_incoming
```

If the spam got through, this is most likely because you have whitelisted your own address, and/or your entire domain.

Most MTAs have ways of handling this problem before the message even reaches TMDA. For example, if the

message claims to be from a local address, but isn't being relayed from a machine on your network, the message can be rejected or discarded as a forgery.

In Postfix, I accomplish this using the following configuration:

In main.cf,

```
smtpd_recipient_restrictions =
          permit_mynetworks,
          check_sender_access hash:/etc/postfix/sender_checks,
          [...]
```

And then in /etc/postfix/sender_checks:

```
jason@mastaler.com DISCARD
```

So the 'permit_mynetworks' setting takes precedence and allows me to send mail from jason@mastaler.com as long as I'm doing so from within my network. Other attempts (obviously forgeries) are discarded. I DISCARD rather than REJECT, because otherwise I'd get back the rejection messages which would sort of defeat the purpose.

Sendmail and Exim (see FAQ 7.5 for one solution) likely have similar solutions.

If you are using qmail, Lou Hevly describes the Spamcontrol patch which allows you to add your address to ./badmailfrom and still send mail to yourself and others on your server.

Jesse Guardiani has also written a script for qmail that blocks forged messages claiming to originate from a local domain.

If for whatever reason you are not able to address this with your MTA, and are forced to deal with it with TMDA only, the simple solution is to not wildcard your domain. Instead, add each individual address to your whitelist, so spam coming from a random address in your domain won't get through.

For example, in your incoming filter file you could have:

```
from-file /root/local.emails.txt accept
# drop messages forged from local addresses as local.emails.txt above
# catches all valid ones
from *@=yourdomain.dom drop
```

Where local.emails.txt is a list of local email addresses generated by a small script that tallies valid user accounts and mail aliases on the system.

Also, don't whitelist your own address. When you need to send mail to yourself, Bcc: the message to a keyword address, an extension address which is not TMDA-protected, or one which you have granted access to in your incoming filter. e.g,

```
to jason-inbox@mastaler.com accept
```

Mark Horn wrote a script to make use of TMDA's FINGERPRINT system. See TmdaFingerprint.

Again though, the MTA is a better place to address this problem.

IOError: /home/foo/.tmda/pending/1024366714.22692.msg has no Return-Path header!

Starting with TMDA version 0.48, incoming messages must be given a Return-Path header to record the envelope sender address. This is later used for things like the CONFIRM_APPEND feature.

So, if your MTA has not been configured to do this, you'll notice the following type of error being raised in your LOGFILE_DEBUG when someone replies to a confirmation request:

- ```
  Uncaught Python 2.2.1 exception (Thu Jun 20 04:07:21 2002 UTC):
  ------------------------------------------------------------
  Traceback (most recent call last):
    File "/usr/local/tmda/bin/tmda-filter", line 50, in ?
      execfile(os.path.join(execdir, 'tmda-rfilter'))
    File "/usr/local/tmda/bin/tmda-rfilter", line 675, in ?
      main()
    File "/usr/local/tmda/bin/tmda-rfilter", line 607, in main
      verify_confirm_cookie(cookie_value)
    File "/usr/local/tmda/bin/tmda-rfilter", line 386, in verify_confirm_cookie
      raise IOError, \
  IOError: /home/foo/.tmda/pending/1024366714.22692.msg has no Return-Path header!
  ```

  ♦ If you use qmail, make sure that tmda-filter is invoked with 'preline' from your dot-qmail. e.g,

  ```
  |preline /usr/bin/tmda-filter
  ```

Exim users must configure Exim to add a Return-Path header to messages from their address_pipe transport. See the PreConfiguration page for more details.

What happens when two TMDA installations interact? Lost mail? Mail loops?

A common worry is that when TMDA user X sends mail to TMDA user Y, Y's confirmation request will be stuck in X's pending queue, and neither party will be aware of the exchange.

If X uses his common sense, this won't happen. He should simply make sure his message is repliable using one of TMDA's client-side options (see FAQ 5.5). TMDA auto-replies to the envelope sender of the message as all standards-compliant auto-responders should, so even if you don't want to tag your "From:" or "Reply-To" address, you should tag your envelope sender address. FAQ 5.4 details how to do tag your messages using a 'dated' envelope sender address.

Another common worry is that two TMDA installations will create a mail loop as they send confirmation requests back and forth. This will not happen, as TMDA is configured to not respond if the message contains identifying characteristics of a mailing list message, bounce message, or auto-response such as the vacation program (or another TMDA message!). Even if this fails, the mail-loop will be stopped by TMDA's auto-response rate-limiting feature that puts a ceiling on the number of messages it sends to a given address in a day.
  ```

How is the "responses" directory cleaned out?

'responses' cleans itself out. At any given time, it only contains the auto-responses generated by TMDA during the last 24 hours.

Why aren't confirmed/released messages deleted from pending?

They are. You must be using an old version of TMDA.

TMDA client-side issues

Can I send tagged messages with a non-Unix client like Microsoft Outlook?

Yes. The tmda-ofmipd program included with TMDA will allow any mail client capable of SMTP Authentication to send tagged mail. This includes most non-Unix clients. For complete information, see http://wiki.tmda.net/TmdaOfmipdHowto

My MUA doesn't support a /usr/sbin/sendmail interface. Can I still tag my outgoing messages?

Some MUAs don't support a sendmail command-line interface to send mail, but rather use direct SMTP only. These include Mozilla, as well as most all non-Unix mail clients. The answer is to use the tmda-ofmipd program to send your outgoing mail. See http://wiki.tmda.net/TmdaOfmipdHowto

Can I post to a TMDA protected mailing list with a 'dated' address?

You may have noticed that every time you post to a TMDA protected mailing list (such as tmda-users) with a 'dated' address you are asked to confirm.

TMDA will allow your post through if you are posting from the same address that you are subscribed with, or if your address matches a global "whitelist". Otherwise, you confirm your post, and that address is whitelisted so that future posts will go right through.

If you wish to continue posting to the list with a 'dated' address, you can use the 'tag' outgoing action documented at http://tmda.net/config-filter.html. For example,

```
to tmda-users@tmda.nett tag
     # Make sure envelope matches your subscribed address.
     envelope ext=list-tmda-users
     from dated
```

How can I make sure undeliverable bounces get returned to me?

When you send an outgoing message, you'd like to make sure that if the message is returned as undeliverable, it is delivered to you, and not challenged.

Such bounces are send with a null <> envelope sender address, so some users whitelist them with the following line in their FILTER_INCOMING:

```
from <> ok
```

The problem with this is that spammers sometimes send junk-mail from <>.

It also isn't sufficient to just whitelist mailer-daemon@yourhost.dom , as some bounces will be returned from the receiving host's SMTP server instead (e.g, mailer-daemon@remoteserv.dom). You could then whitelist mailer-daemon@*, but again some spammers send junk-mail from mailer-daemon addresses. Also, some MTAs might not send bounces from "mailer-daemon".

The best solution is to use TMDA's client-side features to tag your outgoing mail with a 'dated' envelope sender address. The envelope sender address is the address where bounces will be returned, so despite what your "From:" address is, you will still get back bounces.

You can do this easily using the 'tag' option in your FILTER_OUTGOING file. For example, I have the following in mine:

```
## use my bare address and a dated envelope (to catch bounces) with
## whitelisted recipients
to-file ~/.tmda/lists/whitelist tag envelope dated=10d from bare
to-file ~/.tmda/lists/wildcards tag envelope dated=10d from bare
```

When someone replies to my messages, will they get challenged?

No. TMDA provides several methods which safely allow replies to your mail messages to pass through unchallenged. This is advantageous because it reduces traffic and the number of confirmation requests to a bare minimum. If you play your cards right, the people you e-mail with will never even know your mailbox is protected by TMDA.

With tagged addresses:

If you make the `dated` tagged address type your default when you send outgoing messages via tmda-sendmail or tmda-ofmipd, their response will go through unhindered if it arrives within the timeout interval (default is 5 days, but can be customized).

TMDA can be also be configured to automatically add recipients to your whitelist as you send the message using the BARE_APPEND setting.

However, this isn't as reliable as using a dated address, as often the recipient will respond from a different address than you sent to.

Perhaps the best of both worlds is to use a 'dated' Reply-To address for unknown recipients, but also append their address to your whitelist. The example below shows how you might do this.

First, in ~/.tmda/config:

```
BARE_APPEND = os.path.expanduser("~/.tmda/lists/whitelist")
```

Next, in ~/.tmda/filters/outgoing:

```
# Use my bare address and a dated envelope (to catch bounces) with
# whitelisted recipients
to-file ~/.tmda/lists/whitelist tag envelope dated=10d from bare
```

```
# Catch-all rule for addresses that are not yet on the whitelist.
# Use a dated envelope (to catch bounces), a dated Reply-To for safe
# measure, and also append the recipient's address to the whitelist.
to * tag envelope dated=10d reply-to dated from bare=append
```

Of course, make sure you are checking the whitelist file on incoming messages. In ~/.tmda/filters/incoming:

```
from-file ~/.tmda/lists/whitelist ok
```

Without tagged addresses:

If you are opposed to using dated address in your outgoing messages (as some users are), then there is another technique that will help.

Every mail message you send has a unique `Message-ID` in the header that identifies that message. e.g,

```
Message-ID: <hh7kaoldcj.fsf@hrothgar.la.mastaler.com>
```

When people reply to that message, their mail client includes this Message-ID in one or more new headers, `References`, and/or `In-Reply-To`, e.g:

```
In-Reply-To: <hh7kaoldcj.fsf@hrothgar.la.mastaler.com> ("Jason R.
     Mastaler"'s message of "Mon, 24 Mar 2003 13:27:24 -0700")
```

TMDA can read these headers. The idea is to have TMDA accept messages which contain one of your Message-IDs in their 'References' or 'In-Reply-To' headers. This is proof that the reply is to a message that you sent, and therefore legitimate.

You can use `headers` or `headers-file` lines in your incoming filter file to accomplish this. For example,

```
# Accept replies to messages I've sent.
headers '^In-Reply-To:(.|\n)*<.*@hrothgar\.la\.mastaler\.com>' ok
headers '^References:(.|\n)*<.*@hrothgar\.la\.mastaler\.com>' ok
```

This strategy seems simple, but it works, and there is no evidence that spammers are including valid Message-IDs along with the spam that they send you.

NOTE: make sure that the string you use in your Message-IDs is not a valid hostname that a spammer could send mail to. In my case, 'hrothgar.la.mastaler.com' is not a real Internet hostname, so it works perfectly.

If you find that your user-agent is generating Message-IDs with a valid hostname and doesn't allow you to customize this, you can use TMDAINJECT to override the Message-ID of your outgoing messages with the string of your choice. For example, if you wanted your Message-IDs to contain the string `@foo.bar.yourdomain.dom`, you'd add the following lines to your ~/.tmda/config:

```
os.environ["TMDAIDHOST"] = "foo.bar.yourdomain.dom"
TMDAINJECT = "i"
```

Now when you send outgoing mail using either tmda-sendmail or tmda-ofmipd, your Message-IDs will look something like this:

```
Message-ID: <1056046770.22707.TMDA@foo.bar.yourdomain.dom>
```

You can then match `@foo.bar.yourdomain.dom` in your FILTER_INCOMING using a 'headers' entry as shown above.

If you'd like to use this strategy when posting to USENET because dated addresses are harvested and spammed before the address expires, you can do so. Of course, you won't be able to use TMDAINJECT to set your Message-ID as that only works with mail readers.

Some newsreaders allow the user to customize the Message-ID string, in which case you are set. With Gnus v5.10 and above for example, you'd use the following in your ~/.gnus:

```
(setq message-user-fqdn "foo.bar.yourdomain.dom")
```

If your newsreader doesn't provide this ability, request it!

Which mail clients (MUAs) are supported by tmda-ofmipd?

Any MUA that can send mail by connecting to an SMTP port, and which supports SMTP Authentication (RFC 2554) can use tmda-ofmipd.

For SMTP AUTH, tmda-ofmipd supports the following authentication methods: PLAIN, LOGIN, and CRAM-MD5. This covers the majority of clients.

The following MUAs have been known to work with tmda-ofmipd.

MUA	AUTH method
Apple Mail 2.546	CRAM-MD5
Becky! 2.00.11	CRAM-MD5
Edmax 2.96	CRAM-MD5
Eudora 5.1.1	CRAM-MD5
IMP Webmail 3.1	LOGIN
KMail 1.4.2	PLAIN
Mahogany 0.64.1	CRAM-MD5
Mew 2.2	CRAM-MD5
Mozilla 5.0	PLAIN
Mulberry 2.2.1	PLAIN
Netscape 4.78	PLAIN
Outlook 2000	LOGIN
Outlook Express 5	LOGIN
Pegasus Mail 4.01	CRAM-MD5
Pine 4.44	CRAM-MD5
Python's smtplib	CRAM-MD5
sSMTP 2.48	LOGIN
Sylpheed 0.8.2	CRAM-MD5
Thunderbird 1.5	PLAIN (over SSL)
Ximian Evolution 1.0.8	CRAM-MD5

My MUA doesn't support direct SMTP. Can I still use tmda-ofmipd?

Some Unix MUAs (e.g, Mutt, Gnus) only support a /usr/sbin/sendmail command-line interface for sending mail, and can't do direct SMTP.

To use tmda-ofmipd with such a client, you'll have to install a SMTP AUTH capable client program with a sendmail compatible wrapper. You'll then point your MUA at that, and when mail is sent, the wrapper will hand the message off to tmda-ofmipd.

You can use the sSMTP program to do this. Some links for sSMTP:

Source: ftp://metalab.unc.edu/pub/Linux/system/mail/mta/

Debian: http://packages.debian.org/unstable/mail/ssmtp.html

RPMs: http://www.rpmfind.net/linux/rpm2html/search.php?query=ssmtp

I've had success with the following ssmtp.conf file:

```
# Your tmda-ofmipd host:port
mailhub=nightshade:8025

# Set this to never rewrite the "From:" line (unless not given) and to
# use that address in the "from line" of the envelope.
FromLineOverride=YES
```

When you point your MUA at ssmtp, use the `-au' and` -ap' flags to pass along your tmda-ofmipd username and password. e.g, in your .muttrc:

```
set sendmail="/usr/bin/ssmtp -au jason -ap MyPassword"
```

If anyone has any other solutions to this problem, please let us know.

Why is CRAM-MD5 disabled in tmda-ofmipd when using remote authentication?

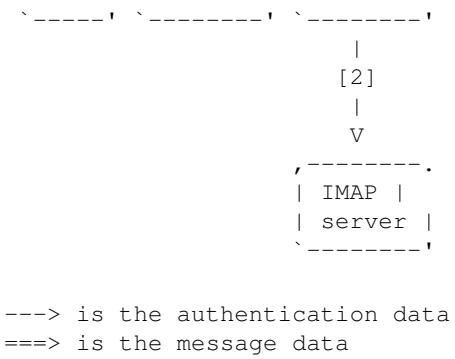
Unfortunately, CRAM-MD5 does not work with -R (--remoteauth) or -A (--authprog) authentication methods.

If you want to know why and how authentication works in tmda-ofmipd (and tmda-manager), read the explanation below.

In the following discussion, I'll use IMAP as the authentication method, although it applies as well to POP3, APOP and LDAP (with and without SSL).

Here is a little ascii schema to better visualize the flux:

```
,-----. ,--------. ,--------.
| MUA |-----[1]--->| ofmipd | | SMTP |
| |=====[3]===>| proxy |======[4]=====>| server |
```



1. the MUA sends authentication data (ie. username and password in clear) to ofmipd acting as a SMTP server.
2. ofmipd acts now as a IMAP client to authenticate against the IMAP server.
3. if the auth is OK, ofmipd accepts the message from the MUA.
4. ofmipd tag the address(es) and act as a client to send the message to the SMTP server.

The important point here is that ofmipd acts as a client to authenticate, and since a client has the clear text password, then ofmipd needs to get the clear text password from the actual MUA client.

This breaks down when using CRAM-MD5 since during SMTP authentication, the client password is converted into a hexdigest before transmission. Thus, tmda-ofmipd never receives the password in clear text, and thus can't verify it against an external source like an IMAP server.

However, given the username and a local password file (like /etc/tofmipd), tmda-ofmipd can lookup the users password, recalculate the hexdigest, and then compare that to the digest it received from the client. This is why CRAM-MD5 is supported with "authfile" authentication.

See http://www.faqs.org/rfcs/rfc2195.html for more on how CRAM-MD5 works.

If you want to use remote authentication but are concerned about sending passwords over the wire in the clear, you should use stunnel or ssh to encrypt the connection between the MUA and tmda-ofmipd.

Does tmda-ofmipd have TLS/SSL support?

For security reasons, many users are interested in creating an encrypted channel between their MUA and tmda-ofmipd which generally listens on port 8025 on the server.

tmda-ofmipd does not currently support this natively, but this can easily be accomplished using existing tools such as SSH and Stunnel.

Here are some references for how you might set this up:

- http://permalink.gmane.org/gmane.mail.spam.tmda.devel/3573
- http://permalink.gmane.org/gmane.mail.spam.tmda.devel/3576
- http://www.arda.homeunix.net/stunnelsetup.html

Recent versions of TMDA have support for better integration of tmda-ofmipd and tcpserver/xinet/stunnel, which allows tmda-ofmipd to correctly report the client's IP address, for example. Refer to the contrib/ofmipd-stunnel-xinetd directory.

Integration issues (with external resources)

How do I setup TMDA with vpopmail?

See Lou Hevly's TMDA/vpopmail tutorial, available at http://www.visca.com/tmda/tmda_vpop.html

Instructions for using tmda-ofmipd with vpopmail can be found at
http://tmda.net/tmda-vdomains.html#vpopmail

Also, to simplify adding TMDA to vpopmail accounts there is a vadduser-tmda wrapper script available in the contrib directory. Installation and usage instructions are included in the top of the script.

How can I prevent BBDB from asking about 'dated' addresses?

If you use BBDB with Emacs, you may have noticed that every time it sees a new 'dated' address it asks you whether it should be added.

To prevent this, you can either set bbdb-always-add-addresses to 'never, or use a BBDB hook to filter the addresses before adding them to the database. See http://my.gnus.org/Lisp/1012312767 for an example.

How do I integrate TMDA with the ezmlm mailing list manager?

TMDA is a good way to prevent SPAM from reaching your mailing lists, without requiring that the sender be subscribed to post.

TMDA can check whether the incoming message is from a subscriber and let it straight through, and otherwise prompt the sender to confirm. After which the message is delivered to the list, and future messages from that sender will not require confirmation.

TMDA integrates particularily easily with ezmlm. Mate Wierdl has written a script called `tmda4ezmlm' that sets up TMDA for an ezmlm mailing list:

http://www.csi.hu/mw/tmda4ezmlm

http://www.csi.hu/mw/tmda4ezmlm.sh

You need both files.

Can I use TMDA with Fetchmail?

Yes, albeit after some additional configuration. Simon Waldman has a HOWTO which explains how to do this.

Also see Hannu's HOWTO at http://wiki.tmda.net/TmdaFetchMailHowTo

Also see http://tmda.net/config-vars.html#RECIPIENT_HEADER In some fetchmail configurations, you can use RECIPIENT_HEADER in lieu of the RECIPIENT and EXT variables normally required by TMDA. Fetchmail should already provide SENDER in the environment.

Can I use TMDA with getmail?

Yes. getmail version 4 includes native support for TMDA; use the Filter_TMDA message filter. See the documentation on this class at http://pyropus.ca/software/getmail/configuration.html#conf-filters-tmda for details.

Any tips for using Gmane with TMDA?

If you haven't heard of Gmane, visit http://gmane.org/ for the scoop. Gmane is the best thing since TMDA, which is the best thing since sliced bread. Gmane will save you a ton of time and hassle when reading and posting to Internet mailing lists.

Here are some tips for the TMDA user that should smooth your Gmane experience.

First, whitelist auth@gmane.org . Gmane uses a challenge/response system similar to TMDA's to authorize posters, and this is the address that the confirmation requests are sent from. See http://gmane.org/post.php for more on this process.

You may prefer (as I do), to post to a mailing list with a 'dated' From or Reply-To in order to more easily accept direct replies to your messages.

As explained on the "Posting" page referenced above, Gmane's news-to-mail authorization script uses the From header to determine who sent the message. Thus, using a 'dated' From isn't wise because then you'll have to confirm every post. So the best idea is to use a 'bare' From, and a 'dated' Reply-To. Then you'll only have to confirm your first post to a group.

This scheme is complicated further by the fact that Gmane will send confirmation requests to Reply-To if it exists instead of From. However, a special header *Gmane-From* will reverse this behavior.

In summary, when posting to a Gmane group, use a 'bare' From, a 'dated' Reply-To, and add Gmane-From. It matters not what the value for Gmane-From is, as long as it's non-empty. Gmane-From will be stripped before your message is sent to the mailing list.

If you use Gnus to read news, here is an example of how you can accomplish this using 'gnus-posting-styles' from your .gnus:

```
(defun tmda-dated-address ()
  (shell-command-to-string "/path/to/tmda-address -dn"))

(setq gnus-posting-styles
      '(("gmane.*"
         (address "jason@mastaler.com")
         ("Gmane-From" "yes")
         ("Reply-To" tmda-dated-address))))
```

2007-02-24 17:18