==========
==========
MAILMISTRESS MAILING LIST SERVER FOR HAIKU

USER INSTRUCTIONS
This document covers version 0.8
==========
==========

Contents

==========
==========

1. Copyright & license

2. Intro & prerequisites

MailMistress is a server program for Haiku designed to work in tandem with the built in Haiku Mail daemon. MailMistress can be set up on any computer with an internet connection. It does not need a static or publicly visible IP address, thus your MailMistress server can sit behind a NAT gateway. MailMistress does not contain a POP/IMAP/SMTP server and will need to work in tandem with one which sends/receives email for your domain such as the one provided by your internet hosting provider or your existing company mail server.

This is a feature not a limitation. It was specifically designed for use by organisations which had an existing mail server handling mail for their domain but for which they did not have full 'root' administrative access such as at an ISP or locked away in the vaults of corporate IT.

MailMistress requires the Haiku Mail daemon to be setup to receive mail from your mailing lists incoming address. For the purposes of this document we will use the a discussion list as the working example. To post to this list members send their messages to discuss@example.com which is a standard POP/IMAP mailbox on the example.com mailserver provided by the internet hosting company.

MailMistress is configured to pick up mail for this mailbox as a standard POP/IMAP client using the Haiku mail daemon and distribute it to the lists members. In order for this to work you need to follow some simple rules when configuring the Haiku mail daemon which will be discussed in a minute in section 3.

A second mailbox is necessary as the outgoing address for the list, to which any bounces will be received.

This prevents bounces from individual subscribers being echoed to all subscribers on the list.

The convention is for this address to be same as the lists incoming address but with -bounces appended, ie discuss-bounces@example.com

THESE *MUST* BE TWO SEPARATE POP/IMAP MAILBOXES NOT AN ALIAS TO THE SAME BOX!

In order to determine if a poster to the list is authorised to post, an authentication program 'plug in' must be provided by you, the server/list administrator.

This plug in architecture allows the authentication process to be tailored on a per list basis to the specific needs of that list. For example you might want to connect to your MySQL/PostgreSQL/xyzSQL database to determine if the poster is a subscriber, you may want to connect to an LDAP or AD server or simply consult a simple text file of valid posters.

The authentication program can be written in whatever language you like so long as it can run on Haiku, accept command line arguments, write to a text file, and return an exit status number to the OS. Compiled C/C++, Perl or a shell script are the most common choices. Details of the API and how to write a plugin are discussed in section 4.

In addition MailMistress allows per list configuration of a customised bounce message to unauthorised posters, a customisable limit to the size of an email, the choice of whether to allow attachments or not and the option to save all messages for archive purposes. The customised bounce message is stored in a plain text file which must be less than 50KB (more than enough-just look at the size of this README file!).

The system will not process emails larger than 100MB, as a sanity check, bearing in mind most internet mail servers will not accept emails more than a few MB (usually 2-5MB)

3. Configuration of Haiku Mail Daemon

EVEN IF YOU ARE FAMILIAR WITH THE HAIKU MAIL DAEMON READ THIS SECTION
MAILMISTRESS WILL NOT WORK IF THE MAIL DAEMON IS NOT SETUP IN A PARTICULAR WAY

On the server which handles mail for your domain setup an incoming (IC) and outgoing (OG) email box for your list.

The recommended convention is to use the same username (i.e before the @ sign) for both but with -bounces added to the OG address.

For example the Discussion list which we're using as our example uses discuss@example.com & discuss-bounces@example.com

Members send emails to discuss@example.com (the IC address) when they want to post to the list.

If a message sent out from the list to a subscriber bounces back -- perhaps because that subscribers email address is no longer working, then the bounce will goto discuss-bounces@example.com (the OG address).

As mentioned earlier these MUST be SEPARATE mailboxes NOT aliases to the same box otherwise you will get into all sorts of messy infinite loops.

The Haiku Mail Daemon (using the Email Preferences app) must be configured to pick up and send mail for these two mailboxes.
At present Haiku defaults to using /boot/home/mail as the parent directory for all mail accounts. This may change in later Haiku releases, particularly when Haiku becomes multiuser, but this doesnt matter.

We recommend you pick a parent directory for all the MailMistress folders and a

directory called 'lists' in this system default directory is as good as any.

Below this parent directory we recommend you create a sub directory for each mailing list your server handles and for ease of identification use the lists IC email address as the directory name.
e.g /boot/home/mail/lists/discuss@example.com/

Below that we recommend creating several sub directories called 'in' 'out' 'bounces-in' & 'archive'
Your filesystem should thus look like this:

```
/boot/home/mail/lists
                     |
                     |-/discuss@example.com
                                          |
                                          |-/archive
                                          |-/bounces-in
                                          |-/in
                                          |-/out
```

These are the pathnames we shall use throughout the remainder of this example.

Messages received for the list which are waiting to be distributed to subscribers are put into the 'in' folder, messages being distributed to subscribers are put in the 'out' folder . If you enable the Archive facility (described later in section 4) all emails sent to the list will be dumped into 'archive' after successful processing.

Any messages which are sent to discuss-bounces@example.com (i.e delivery failure bounces) will go into 'bounces-in'.

You dont have to use this layout but it is highly recommended you use something along these lines to keep things logical. It is essential that each list has its own unique 'in' folder and (if needed) its own unique 'archive' folder. What they're called, or where they are, is up to you, but you CANNOT share them between lists.

Create your filesystem layout for your list then launch the Haiku Email Preferences App

On the Settings tab make sure 'start mail services on startup' is checked and that it is set to check for mail periodically. How often is upto you, bearing in mind the amount of traffic you expect your list to have, and how long a delay you're prepared to have between someone sending a message and MailMistress processing it.

On the Accounts tab click Add. You need to setup both of your lists mail accounts - the IC address (discuss@example.com) and the OG address (discuss-bounces@example.com)

Set the Account name to the email address (e.g discuss@example.com or discuss-bounces@example.com) and the Real Name to the same. THIS IS IMPORTANT. MailMistress won't work otherwise.

Enter the appropriate server addresses, usernames & passwords for both mailboxes.

Once the setup wizard has finished you will see the accounts listed in the left hand column of the Preferences window. Under each account name there is an Incoming & Outgoing section

Click each accounts Incoming section and make sure 'Partically download messages larger than....' is UN checked. Make sure 'Remove mail from server when

deleted' IS checked.

Set the Destination (In) folder for the IC address to
/boot/home/mail/lists/discuss@example.com/in (or what ever folder layout you're using)

Set the Destination (In) folder for the OG address to
/boot/home/mail/lists/discuss@example.com/bounces-in (or what ever folder layout you're using)

Click each accounts Outgoing section
Set the Destination (Out) folder for both accounts to
/boot/home/mail/lists/discuss@example.com/out (or what ever folder layout you're using)

It should be noted that some SMTP servers will reject mail from systems which only have a hostname set rather than a fully qualified host&domain name (FQDN), at the time of writing a default Haiku installation uses the hostname 'shredder' with no domain name and some SMTP servers will reject this.

To fix this open a Terminal (command line) and use the hostname command to set a FQDN of your choice. localhost.localdomain is fine. Unlike most Unix-like systems Haiku automatically persists this hostname change across reboots so you only need to do this once

4. Configuration of MailMistress

MailMistress is configured via a simple plain text config file which must be called MailMistress.conf and placed in the same folder as the server program, the file contains a number of directives which configure various parameters of the server.

Each directive is placed in either a <GLOBAL> section or a <LIST> section.

There must be one <GLOBAL> section terminated with </GLOBAL>

There must be at least one <LIST> section terminated with </LIST>

There can be as many <LIST> sections as needed. One for each list this server handles.

The <GLOBAL> section defines server wide configuration. The <LIST> section defines per-list configuration.

Some directives are mandatory, some are optional. Some directives can be placed in both a <GLOBAL> and <LIST> section.

The sample .conf file provided with the software contains every possible directive with descriptive comments. These comments make the conf file pretty self explanatory but this README file mentions a few in a little more detail for clarity.

The creation & use of the authentication program (AUTH directive) will be discussed in section 5

The PLAINTEXTONLY directive is perhaps worthy of a little explanation. This directive, which can appear in either a <GLOBAL> or <LIST> section (or both) sets whether the server as a whole or a specific list will allow messages which contain anything other than plain text such as HTML text or attached files. There are 3 possible values (Y, N or H) with Y being the most restrictive and N (the default) being the least restrictive. If the directive is specified in both a <GLOBAL> and <LIST> section then the most restrictive setting takes

precedence. Y means messages can only contain plain text (not HTML text, and no attachments), H means messages can contain plain text or HTML text but no attachments, N means messages can contain anything (plain text, HTML text or attachments)

5. Authentication program plug in

The authentication program is a separate program provided by you the server/list admin which MailMistress uses to determine if a particular poster is authorised to post to the list. You can provide one program per list or one program can be shared between lists.

MailMistress passes the list IC address to the authentication program with each authentication request so that you can determine which list it relates to if needbe.

The authentication program also returns to MailMistress a list of all the subscribers to which the message should be distributed.

How it determines if a poster is authorised, and where it obtains the list of subscribers is up to you, depending on the needs of your particular list.

For example it may connect to a database server and run an SQL query, or consult a simple text file, or connect to an LDAP / AD server.

Communication between MailMistress and your authentication program takes place using command line arguments (for MailMistress to pass data into your program), a text file in the servers temp directory (for your program to pass the list of subscribers back to MailMistress) and the operating system program exit status code (for your program to indicate the result of the authentication to MailMistress).

The temp text file is created for you by MailMistress and the full path string to it is passed into your program via one of the command line args. When your program runs this file will exist but be empty (and closed). Your program should write into it one recipient address per line (with lines delineated with \n) and close the file. The senders email address is also passed to your program as a command line arg along with the lists IC address. The presence of the list IC address allows you to write an authentication plug in which handles more than one list if you wish.

When your program exits it must return one of the following exit codes to the operating system to indicate to MailMistress the result:

0=Poster is authorised to post. Temp file is filled out with recipient addresses
1=Error occured (e.g if you cant connect to a database etc). The incoming email will be kept and MailMistress will try again later.
2=Poster is NOT authorised to post. Bounce the message back to them.
3=Poster is NOT authorised to post. Silently discard the message.
100=TEST intercepted
Any other value will be interpreted as if it was a 1.

Your authentication program should never delete the temp file passed to it by MailMistress. MailMistress will do this when your program exits, even if the file was left empty.

If in the somewhat unlikely event that the poster is authorised to post but there are no subscribers it is safe to return 0 with an empty temp file

Command line arg1 is the senders email address
Command line arg2 is the full path to the temp file
Command line arg3 is the lists IC address

Your program must check if both of the first two args are "TEST" (in capitals) and if so return 100. This check is used by MailMistress when it starts up to check the authentication program exists and is executable.

A skeleton pseudo-C based auth program is included as authexample.cpp as an example
You may write your program in whatever language you wish so long as it fulfills the API requirements above.

Have fun. Don't abuse email. It's a wonderful open technology. Bulk Mail responsibly!