

---

# MailSigger

Karsten Schulz <mailsigger@karstenschulz.name>

Revision 0.4

Revision History  
2007-04-26

KS

## Table of Contents

|   |   |
|---|---|
| Overview .....                          | 1 |
| Downloads .....                         | 1 |
| News .....                              | 2 |
| How it works .....                      | 2 |
| Installation .....                      | 3 |
| Download and install the program .....  | 3 |
| Create an user .....                    | 4 |
| Create an working directory .....       | 4 |
| Configuration .....                     | 4 |
| Setup the parameters .....              | 4 |
| Assignment of the signature files ..... | 5 |
| Configure Postfix and MailSigger .....  | 6 |
| Test the installation .....             | 7 |
| Alternatives to MailSigger .....        | 7 |
| Disclaimer and License .....            | 7 |

## Overview

MailSigger is a small Python program which is intended be installed on the MTA in your network. Depending on the sender address, a disclaimer file will be attached to the outgoing email.

It handles plain text emails as well as MIME emails. This introduction shows, how to make it work with the Postfix MTA.



### MailSigger depends on Python 2.5!

MailSigger needs a current Python 2.5 [<http://www.python.org/>] to run!

Hint: when changing some Python import statements, you can make `mailsigger.py` work with earlier versions (try 2.3+ and change the mime-imports). But this is not recommended, nor is it supported.

## Downloads

- download the latest program archive: mailsigger-0.4.tar.bz2 [../downloads/mailsigger-0.4.tar.bz2]
- read the documentation as pdf: mailsigger-0.4.pdf

## News

- [2007-04-26] - MailSigger Version 0.4 released
  - bugs fixed
  - improved MIME-Email handling
- [2007-02-21] - MailSigger Version 0.3 released
- [2007-02-14] - Start of the project

## How it works

MailSigger reads a whole email from standard input. Then it examines the sender and a table with sender/attachment-pairs. The two columns in the table are:

- a sender address or a sender domain
- a path to a file, which contains the disclaimer/attachment

If it finds an entry for a sender, it will attach the given file to the email. You can both define attachments for an individual sender, or for a whole domain.

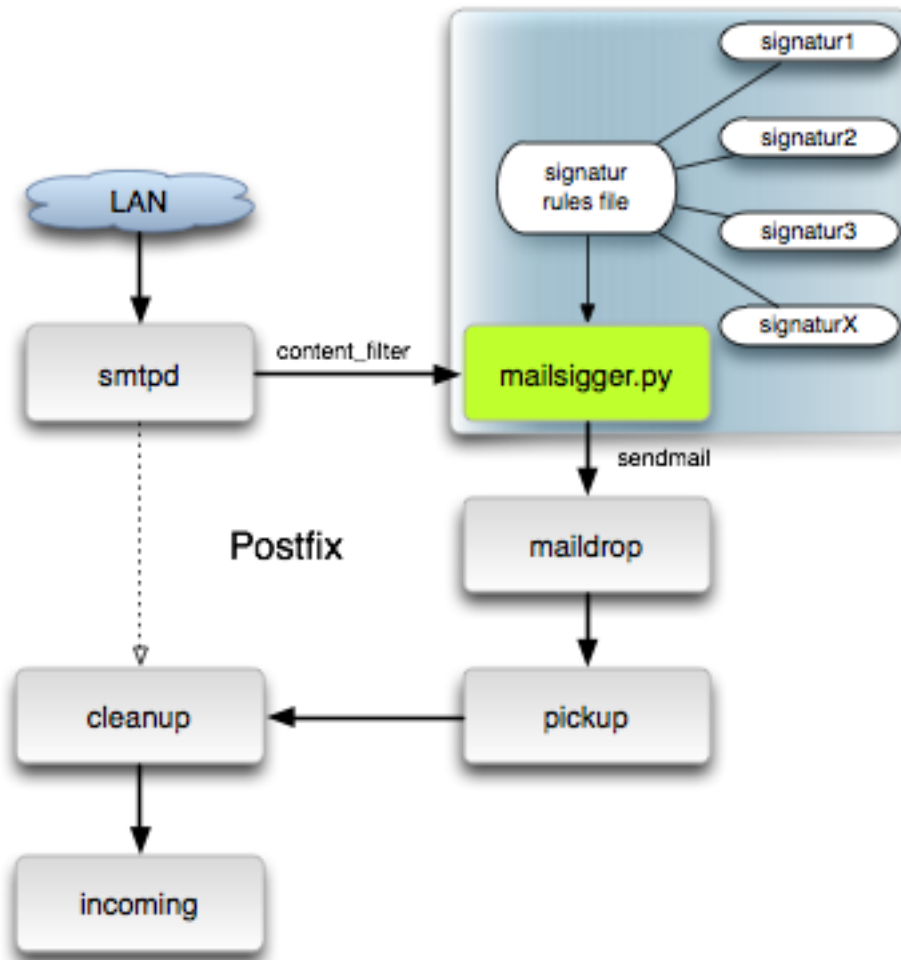
If no signature file is found, or an error in the processing occurred, the email will not be altered! After processing, the email will be printed to standard output and can be feeded back to Postfix, which will now deliver the email.

MailSigger can be integrated into an existing Postfix installation very simply. The Postfix parameter `content_filter` is used to glue these things together.

The filtering by MailSigger takes place independently of a possibly already existing filtering by Postfix (e.g. with `spamassassin` or `amavis`).

MailSigger version 0.4 was testet in 4 little networks with a daily load from about 2500 emails a day.

### **Figure 1. Integration of MailSigger into a Postfix installation**



## Installation

At the moment the only supported method to install is to do it by hand. This is not so bad, because the archive contains only two files, which both have to be copied into `/usr/local/bin/` on your system.

After that, you have to create a signatures rules file and the signature and/or disclaimer-files.

The last step is to change Postfix' configuration to use the new filter `MailSigger`.

Let's see it step by step!

## Download and install the program

After downloading `mailsigger-0.4.tar.bz2` [`../downloads/mailsigger-0.4.tar.bz2`] just unpack it and copy both files `mailsigger` and `mailsigger.py` into the directory `/usr/local/bin`.

The file `mailsigger` is a shell script, which was taken from the Postfix documentation. It connects the MTA with a filter program, and this filter program will be our `mailsigger.py`.

```

host:~ # wget http://www.karstenschulz.name/assets/downloads/mailsigger-0.4.tar.bz2
...
host:~ # tar xvjf mailsigger-0.4.tar.bz2
host:~ # cd mailsigger-0.4
host:~ # cp mailsigger* /usr/local/bin
  
```

## Create an user

Now we create an user, who will be used to run the scripts. This is for security reasons. If there is a bug in `mailsigger.py` (and believe me, there will be tons of it), the system won't be compromised at all.

```
host:~ # useradd -s /bin/false mailsigger
```

## Create an working directory

We need a spooling directory for MailSigger. MailSigger needs read and write access to this directory. At the moment, this spooling directory can also be used for MailSigger's logfile.

```
host:~ # mkdir /var/spool/mailsigger
host:~ # chown mailsigger /var/spool/mailsigger
host:~ # chmod 0700 /var/spool/mailsigger
```



### Location of the working directory

If you wish to use another directory make sure, that the shell variable `INSPECT_DIR` in the shell script `/usr/local/bin/mailsigger` is changed accordingly!

That's it. We are finished with installing! Not we have to configure some things.

## Configuration

We have to create a signature rules file and some signature files. And we must tell MailSigger, where to find these files.

Because there are different encodings of text possible, we have to declare the encoding of our signature files, too. This is done by setting the parameter `SIGNATURE_ENCODING` to a proper value.

## Setup the parameters

Please take a look at the beginning of the file `/usr/local/bin/mailsigger.py`.

### Example 1. Configuration of `/usr/local/bin/mailsigger.py`

```
#####
# please configure the following values:                                     #
#####

# configure different sig-files for different sender here:
SIGNATURE_RULES_FILE = '/etc/Postfix/signature_rules'

# which encoding do you use in your different signature files?
SIGNATURE_ENCODING = 'utf-8'

# loglevel, one of ERROR or DEBUG (atm)
VERBOSITY = logging.DEBUG

# logfile
LOGFILE = '/var/spool/mailsigger/mailsigger.log'
```

```
#####
# end of configuration                                     #
#####
```

This example shows, that in `SIGNATURE_RULES_FILE` the file `/etc/Postfix/signature_rules` is declared as the file, which contains the rules for appending signature files according to the sender.

The value for `SIGNATURE_ENCODING` is set to `utf-8`. This is the standard value for most current systems. Maybe you have an editor, which saves its textfiles in `iso-8859-1` encoding. I hope, you don't use `cp1252` encoding, which was used by some weird systems. Please have a look at the documentation of your editor to learn the correct setting.

With `VERBOSITY` you can control the amount of messages in the logfile. At the moment, you can use the following settings:

**Table 1. Protokollstufe**

| Stufe         | Bedeutung  |
|---------------|--|
| logging.DEBUG | writes debugging messages. Use this right after installation |
| logging.ERROR | writes only error messages.                                  |

And last, you can tell MailSigger with the parameter `LOGFILE`, where to write its messages.



**Write access to the logfile**

The user account, which is used to run MailSigger must of course be able to write to the logfile. Please make sure, that he has sufficient rights.

Besides that, please make sure, that he is able to read the signature rules file and the signature files

## Assignment of the signature files

In the file `SIGNATURE_RULES_FILE` you can configure, which signature file is to be appended to an email depending on its sender.

In our example, this is the file `/etc/postfix/signature_rules`. This file looks like this:

**Example 2. Assignment of the disclaimer files**

```
# sender          # file with signature
boss@mycompany.com  /etc/Postfix/boss-sig.txt
kurt@mycompany.com  /etc/Postfix/kurt-sig.txt
no-sig@mycompany.com
@mycompany.com      /etc/Postfix/sig-for-all.txt
```

The senders `boss@mycompany.com` and `kurt@mycompany.com` both get their own, individual signatures

from different files.

The sender `no-sig@mycompany.com` get no signature appended at all. All other senders from the domain `mycompany.com` get the file `/etc/Postfix/sig-for-all.txt` appended.

As you can see, a comment line is preceded by a `#` at the very first position and empty lines are allowed, too.

## Configure Postfix and MailSigger

The shellsript `mailsigger` is taken from the excellent documentation of Postfix. Please take a look at `FILTER_README.html` [[http://www.Postfix.org/FILTER\\_README.html](http://www.Postfix.org/FILTER_README.html)]. Its purpose is, to connect Postfix with an external filter program like `mailsigger.py`.

### Example 3. Configuration options in `/usr/local/bin/mailsigger`

```
#!/bin/sh

# Simple shell-based filter. It is meant to be invoked as follows:
#   /path/to/script -f sender recipients...

INSPECT_DIR=/var/spool/mailsigger
SENDMAIL="/usr/sbin/sendmail -G -i"
```

Please edit the lines containing `INSPECT_DIR` und `SENDMAIL` to fit your needs. `INSPECT_DIR` is the working directory, which we have created earlier..

The last step to be done is to configure a new transport. We will tell Postfix to use our MailSigger for every email, which is received from the clients in our network. We have to edit

`/etc/Postfix/master.cf`:

### Example 4. Postfix's `master.cf`

```
# =====
# service type private unpriv chroot wakeup maxproc command + args
#          (yes)   (yes)   (yes)   (never) (100)
# =====
127.0.0.1:10025 inet      n       -       n       -       -       smtpd
    -o content_filter=
    -o local_recipient_maps=
    -o smtpd_helo_restrictions=
    -o smtpd_client_restrictions=
    -o smtpd_sender_restrictions=
    -o smtpd_recipient_restrictions=permit_mynetworks,reject_unauth_destination
    -o mynetworks=127.0.0.0/8

smtp-amavis      unix      -       -       n       -       2       smtp
    -o smtp_data_done_timeout=1200
    -o disable_dns_lookups=yes

❶ mailsigger    unix      -       n       n       -       -       pipe
   flags=Rq user=mailsigger argv=/usr/local/bin/mailsigger -f ${sender} ${recipient}

❷ 192.168.0.1:smtp      inet      n       -       n       -       -       smtpd
```

```
-o content_filter=mailsigger
127.0.0.1:smtp inet n - n - - smtpd
```

- ❶ The new transport *mailsigger* calls the script `/usr/local/bin/mailsigger` with the userid *mailsigger*
- ❷ Through this transport, all mails, which reach our Postfix via the socket `192.168.0.1:25`, will be routed through MailSigger. Please adapt the ip address to fit your needs.

## Test the installation

Now we can test our installation. First we filter an email by hand:

```
host:~ # mailsigger.py < mytestmail
```

You should see the email on standard output. If the sender is configured to have an signature, `mailsigger.py` should print your email with the signature appended! If not, the email should not be altered.



### Ownership of your Logfile

If your test was the first run of `mailsigger.py`, your logfile `/var/spool/mailsigger/mailsigger.log` will now probably be owned by *root*. Please change the ownership to the user *mailsigger*, because this user need of course write access to this file!

Now you can check sending an email:

```
host:~ # mailsigger -f sender@here receiver@there < mytestmail
```

Please read the logfiles of MailSigger and of Postfix! If everything is fine, you can activate the system with the command:

```
host:~ # postfix reload
```

## Alternatives to MailSigger

To append disclaimer to emails, there is another fine program out there: `alterMIME` [<http://www.pldaniels.com/altermime/>] by Paul L. Daniels. If your Postfix is under heavy duty, you may want use a compiled C program, instead of a Python filter.

## Disclaimer and License

Because the program is licensed free of charge, there is no warranty for it! Please test, test and test your installation before you use it in a production environment! Make sure, that your system won't loose any emails! Look after the logfiles and into the working directory of **mailsigger.py** at a regularly basis. Ehm, and test it!

This software ist licensed under the GPL version 2. Please read GPL Version 2

[<http://www.gnu.org/licenses/gpl.html>] carefully.

This document was created with Stuart Rackham's wonderful asciidoc  
[<http://www.methods.co.nz/asciidoc/>]!

Have fun and feel free to contact me [<mailto:mailsigger@karstenschulz.name>]