

# SingleUserUnixInstall

## Single-user SpamAssassin installation

This page will take you through a complete single-user [SpamAssassin](#) installation on a typical Unix account.

Many systems already have [SpamAssassin](#) and the support packages installed, in which case this whole page is unnecessary for you. However, if you want to run the newest version of [SpamAssassin](#) and all the related packages, you can always guarantee the setup by installing it yourself in your own directory. This also requires, of course, for you to update it yourself as new versions are released.

Note that, unfortunately, almost every Unix installation is slightly different, and that if you just blindly follow the commands here, this may not work, and you could even lose some mail. In other words, if you really don't know Unix, you may want to get someone who knows it better to help you with this install. Of course, if an up-to-date version of [SpamAssassin](#) with the appropriate plugins is already installed on the machine (you can check with `which spamassassin`), you don't need to reinstall it.

If you know Unix but not [SpamAssassin](#) and procmail, this page will hopefully provide some useful details. This is largely a distillation of the directions in the [<http://www.spamassassin.org/full/3.0.x/dist/INSTALL> INSTALL] file. Also note that this is fairly complex, and that if you're intimidated, you're probably better off getting someone (like the sysadmin) to install it for you. Many ISPs, for example, now offer [SpamAssassin](#) as a free option as part of Unix accounts.

## Overview

We're going to install [SpamAssassin](#) 3.0.2, and add in SPF, Razor, Pyzor, and DCC. We're going to set up a mistake-based Bayesian learner (including forwarding the mail to another account), as described at [ProcmailToForwardMail](#). This assumes that your system already has procmail installed, and has a new enough Perl and Python to work with this software, plus a number of the standard Perl modules, such as Net::DNS and DB\_File. You need these two installed to use DNSBLs and Bayes, both of which are important for good performance.

## Setting up your path

If your system now or in the future may have other copies of spamassassin or the other packages installed on it, we want to make sure that it uses the version we're installing. We do this by having the shell look first in our local bin directory. Even more important, we need to tell Perl where to find the packages we're installing locally, and we try to work around a language bug with some versions of Perl 5.8.

We can do this with bash by entering the following lines at the top of the `.bashrc` (`pico .bashrc`):

```
export PATH=$HOME/bin:$HOME/perl5/bin:$PATH
export MANPATH=$HOME/man:$HOME/perl5/man:$MANPATH
export PERL5LIB=$HOME/lib/perl5/:$HOME/lib/perl5/site_perl/5.8.3:$PERL5LIB
export LANG=en_US
```

The `5.8.3` should be replaced with the version you get when entering `perl -v`.

After saving and exiting (Choose Ctrl-X, then y, then enter), we reload the `.bashrc` with the command `cd ~/. .bash_profile`. The same commands would work if your shell is sh, ksh, or zsh, by editing the corresponding rc file.

In `csh` `.cshrc` and `tcsh` `.tcshrc`, you would add the following lines:

```
setenv PATH $HOME/bin:$HOME/perl5/bin:$PATH
setenv MANPATH $HOME/man:$HOME/perl5/man:$MANPATH
setenv PERL5LIB $HOME/lib/perl5/:$HOME/lib/perl5/site_perl/5.8.3:$PERL5LIB
setenv LANG en_US
```

## Installing SpamAssassin

We're going to download [SpamAssassin](#) and the other packages into `$HOME/src`:

```
cd $HOME
mkdir src
cd src
wget http://www.apache.org/dist/spamassassin/Mail-SpamAssassin-current.tar.gz
tar xvfz Mail-SpamAssassin-current.tar.gz
cd Mail-SpamAssassin-3.0.2
perl Makefile.PL PREFIX=$HOME && make && make install
```

Push enter 4 times (i.e., the defaults are all fine).

## Testing installation

Make sure we're working with the version we just installed by entering `which spamassassin` and we should see something like `/home/myusername/bin/spamassassin`.

Enter `spamassassin < sample-spam.txt`. You should see a message that spamassassin is creating user preferences file and then see the output of the message with the [SpamAssassin](#) markup.

If that doesn't work, look at the debug output with `spamassassin -D < sample-spam.txt` and then perhaps take a look at [FixingErrors](#).

## SPF support

[SpamAssassin](#) 3.0 supports SPF to detect and penalize header forgery. This requires `Mail::SPF::Query`, a relatively new package that's not yet installed on most machines. You can confirm whether you have it by entering `perl -e 'require Mail::SPF::Query'`. If you get the error "Can't locate Mail/SPF/Query.pm in @INC..." you need it, if you get no feedback you can skip to the next section.

To install SPF, do the following:

```
cd $HOME/src
wget http://spf.pobox.com/Mail-SPF-Query-1.997.tar.gz
tar xvfz Mail-SPF-Query-1.997.tar.gz
cd Mail-SPF-Query-1.997
perl Makefile.PL PREFIX=$HOME && make && make install
```

You can test this installation (and that PER5LIB is set correctly) with `perl -e 'require Mail::SPF::Query'`.

## Razor support

To install the packages that Razor requires, do the following:

```
cd $HOME/src
wget http://unc.dl.sourceforge.net/sourceforge/razor/razor-agents-sdk-2.03.tar.gz
tar xvfz razor-agents-sdk-2.03.tar.gz
cd razor-agents-sdk-2.03
perl Makefile.PL PREFIX=$HOME && make && make install
```

To install Razor:

```
cd $HOME/src
wget http://unc.dl.sourceforge.net/sourceforge/razor/razor-agents-2.67.tar.gz
tar xvfz razor-agents-2.67.tar.gz
cd razor-agents-2.67
perl Makefile.PL PREFIX=$HOME && make && make install
razor-client
razor-admin -create
razor-admin -discover
razor-admin -register
```

It should then say "Register successful...". (Note that you may need to enter the last command a couple times to reach the registration server; if it says "Error 202", try "razor-admin -register" again.)

# Pyzor support

To install Pyzor:

```
cd $HOME/src
wget http://unc.dl.sourceforge.net/sourceforge/pyzor/pyzor-0.4.0.tar.bz2
tar xvfj pyzor-0.4.0.tar.bz2
cd pyzor-0.4.0
python setup.py build
python setup.py install --home=$HOME
pyzor discover
```

If you get the following error message, define PYTHONPATH to point at \$HOME/lib/python:

```
Traceback (most recent call last):
  File "<stdin>", line 1, in ?
ImportError: No module named pyzor.client
```

# DCC support

To install DCC:

```
cd $HOME/src
wget http://www.dcc-servers.net/dcc/source/dcc-dccproc.tar.Z
tar xfvz dcc-dccproc.tar.Z
cd dcc-dccproc-*
./configure --disable-sys-inst --disable-server --disable-dccm \
--disable-dccifd --homedir=$HOME/dir --bindir=$HOME/bin
make && make install
```

# Test spamassassin installation

First, create your Bayes databases by entering `sa-learn --sync`.

You should now have all the packages you need installed. You can test this by entering `spamassassin -D < $HOME/src/Mail-SpamAssassin-3.0.0/sample-nonspam.txt` and carefully reviewing the output. Specifically, look for the following lines:

```
debug: bayes: found bayes db version 3
debug: is DNS available? 1
debug: registering glue method for check_for_spf_helo_pass (Mail::SpamAssassin::Plugin::SPF=HASH(0x8d21990))
debug: Razor2 is available
debug: Pyzor is available: /home/username/bin/pyzor
debug: DCC is available: /home/username/bin/dccproc
```

These lines confirm, in order, that DB\_File, Net::DNS, Mail::SPF::Query, Razor, Pyzor, and DCC are all correctly installed and configured.

# Configure procmail

Copy the sample `.procmailrc` from [ProcmailToForwardMail](#). The easiest way to do this is:

```
cd $HOME
wget http://wiki.apache.org/spamassassin-data/attachments/ProcmailToForwardMail/attachments/procmailrc.forward.txt
mv procmailrc.forward.txt .procmailrc
```

It's essential that you edit that file with your correct public and private addresses. Do this with `pico .procmailrc`.

If you don't want your mail forwarded to another account, you can instead use the example procmail file by entering `cp $HOME/src/Mail-SpamAssassin-3.0.0/procmailrc.example $HOME/.procmailrc`.

# Configure .forward

Follow the steps in the first section of [UsedViaProcmail](#) to enable procmail.

Specifically, if your system supports .forward files (as opposed to .qmail) and is not already processing mail through procmail, then edit your .forward. Replace user with your username (which you can discover by entering `whoami`) and entering the correct procmail path (which you can discover with `which procmail`):

```
cd $HOME
pico .forward
"|IFS=' ' && exec /usr/bin/procmail -f- || exit 75 #user"
```

Choose Ctrl-X, then y, then enter to save.

## Test mail installation

Now, you should be ready to send some test emails and ensure everything works as expected. First, send yourself a test email that doesn't contain anything suspicious. You should receive it normally, but there will be a header containing "X-Spam-Status: No".

Now, send yourself a copy of the GTUBE test string to check to be sure it is marked as spam. That string is:

```
XJS*C4JDBQADN1.NSBN3*2IDNEN*GTUBE-STANDARD-ANTI-UBE-TEST-EMAIL*C.34X
```

This email will be recognized as spam and put in the almost-certainly-spam folder. You should be able to see it by entering `less $HOME/mail/almost-certainly-spam`.

If your test non-spam email doesn't get through to you, immediately rename your .forward file until you figure out cause of the the problem, so you don't lose incoming email. `mv .procmailrc .procmailrc.broken`.

Note: one possible cause for problems is the use of smrsh on the MTA system; see [ProcmailVsSmrsh](#) for details.

## Follow-up

You'll want to subscribe to the spamassassin-announce [<http://lists.sourceforge.net/lists/listinfo/spamassassin-announce> list] to be alerted when new updates come out. Follow the same steps as your original install (with the new filename, of course), and the `make install` will automatically overwrite old versions.

If you want to install custom rules, such as those at [CustomRulesets](#), just `cd $HOME/etc/mail/spamassassin` and `wget` the ones you want. Note that many of these rules have already been incorporated into [SpamAssassin](#) 3.0 and so you'll have an unduly high risk of [FalsePositives](#) if you use both.

If you're using [SpamAssassin](#) for non-commercial use, you may also want to turn on the MAPS rules, which are useful DNSBLs. Edit the `user_prefs` by entering `pico $HOME/.spamassassin/user_prefs` and add the following 4 lines:

```
score RCVD_IN_MAPS_RBL 2.0
score RCVD_IN_MAPS_DUL 1.0
score RCVD_IN_MAPS_RSS 2.0
score RCVD_IN_MAPS_NML 2.0
```

## Contributors

- [DanKohn](#)