

# verot.net - How to setup Spamassassin to remotely remove spam messages from an IMAP account, and trigger KMail to update its cached IMAP folders

[kmail\\_spam.htm](#)

[SpamAssassin](#) can be integrated in KMail. However, it is not useable, as KMail is not able to run spam checks in the background, and the whole application freezes for a while when retrieving new emails. This how-to explains a way to get SpamAssassin to work remotely on an IMAP account, and then trigger KMail to update itself. We will also see how to regularly train SpamAssassin.

## Install SpamAssassin

First, you need to install [SpamAssassin](#). Check that it works properly. My configuration file `~/.spamassassin/user_prefs` shows:

```
required_score      5.0
report_safe         0
use_bayes           1
bayes_auto_learn    1
skip_rbl_checks     0
use_razor2          1
use_dcc             1
use_pyzor           1
ok_languages        ar en fr es
ok_locales          ar en fr es
score SUBJ_ILLEGAL_CHARS 0
```

## Install *isbg* (IMAP Spam Begone)

[isbg](#) is a script that makes it easy to scan an IMAP inbox for spam using SpamAssassin and get your spam moved to another folder. Unlike the normal mode of deployments for SpamAssassin, *isbg* does not need to be involved in mail delivery, and can run on completely different machines to where your mailbox actually is.

We will use *isbg* in a cron job to periodically check our remote IMAP inbox, and move spam messages to another folder. Simply download the python script, and put it in your home directory.

Firstly, you need to create a folder called *spam* in your IMAP inbox. In KMail, simply create the folder, and make sure that your remote IMAP account is synchronized. I use cached IMAP folders for my email account.

Run the script with the argument `--savepw` in order to see if your arguments are correct, and to save the password (encrypted). Try the following, with your email server and email account as arguments:

```
./isbg.py --verbose --imaphost mail.domain.net --imapuser user@domain.net
--savepw
```

It should check emails in your remote IMAP inbox, and will save the password. The script will tell you how many messages were parsed, and how many were spam. The spam messages are tagged, and moved to

the *spam* folder you just created. Note that if you run the script again, only new messages will be parsed.

We will now run the *isbg* periodically. What we want is a script that can filter out spam messages in the remote IMAP folder, and then trigger KMail to synchronise itself. We can do this via D-Bus (or DCOP). We will essentially replace the built-in KMail interval mail checking feature, and do it via a cron job. But we will retrieve new messages only **after** filtering out spam.

## The script

Here is the script, compatible with D-Bus. Make it executable, and set it to run every 10 minutes.

```
#!/bin/bash
./isbg.py --delete --expunge --imaphost mail.domain.net --imapuser
user@domain.net &> /dev/null
export DISPLAY=:0
qdbus org.kde.kmail /KMail org.kde.kmail.kmail.checkAccount user@domain.net &>
/dev/null
exit;
```

The same script, for DCOP:

```
#!/bin/bash
./isbg.py --delete --expunge --imaphost mail.domain.net --imapuser
user@domain.net &> /dev/null
dcop --user xxx kmail KMailIface checkAccount user@domain.net &> /dev/null
exit;
```

The script above simply calls *isbg.py*, which filters out spam messages from the remote IMAP account, and then calls KMail via D-Bus/DCOP to synchronize the account locally. Note the use of *--delete --expunge* which actually deletes the spam messages from your inbox (after copying them in the spam folder), preventing the same messages to be caught by *isbg* at each script run. If KMail or Kontact is not started, the D-Bus/DCOP call will silently fail. Note that you need to specify your username in the D-Bus/DCOP call, this is required to make use of D-Bus/DCOP in a cron job.

## Training SpamAssassin

This page doesn't explain how to set up SpamAssassin, not how to configure it to suit your own needs. However, here are two scripts that I use to train SpamAssassin's Bayesian filters. I set it to periodically train itself on my cached IMAP accounts.

I run this first once a day, to train SpamAssassin on my spam messages:

```
sa-learn --spam
~/ .kde/share/apps/kmail/dimap/.#####.directory/.INBOX.directory/spam/cur
```

I run this first once a week, to train SpamAssassin on my non-spam messages:

```
sa-learn --ham ~/ .kde/share/apps/kmail/mail/sent-mail/cur
sa-learn --ham ~/ .kde/share/apps/kmail/dimap/.#####.directory/INBOX/cur
sa-learn --ham
```

```
~/ .kde/share/apps/kmail/dimap/.#####.directory/.INBOX.directory/Clients/cur
sa-learn --ham
~/ .kde/share/apps/kmail/dimap/.#####.directory/.INBOX.directory/Friends/cur
sa-learn --ham
~/ .kde/share/apps/kmail/dimap/.#####.directory/.INBOX.directory/...../cur
```

Of course, replace the `#####` with your own directory name, for the IMAP account that you want to train SpamAssassin with. You can train SpamAssassin on several accounts, directories, etc...

## A last thing...

You may prefer to use SSL for the above. You can, using `--ssl` when you call `isbg`. You will need to have Python compiled with [SSL support](#). But then, if you also set KMail to retrieve emails in SSH, you have the problem of the [SSL certificates warnings](#). Nothing is perfect...