9.1 Development schema .................................................. 23
9.2 Versioning schema .................................................... 23
9.3 Releasing Schema .................................................... 24

10 Manual page for isbg ................................................... 25
  10.1 SYNOPSIS .......................................................... 25
  10.2 DESCRIPTION ...................................................... 25
  10.3 OPTIONS .......................................................... 25
  10.4 EXAMPLES ........................................................ 27
  10.5 OVERVIEW ......................................................... 27
  10.6 SEE ALSO .......................................................... 28
  10.7 EXIT CODES ....................................................... 28
  10.8 BUGS .............................................................. 28

11 Manual page for isbg_sa_unwrap ................................... 29
  11.1 SYNOPSIS .......................................................... 29
  11.2 DESCRIPTION ...................................................... 29
  11.3 OPTIONS .......................................................... 29
  11.4 SEE ALSO .......................................................... 30
  11.5 BUGS .............................................................. 30

12 API Reference .......................................................... 31
  12.1 isbg .............................................................. 31
  12.2 Main Objects and functions .................................... 31
  12.3 Indices and tables ................................................ 32

13 IMAP Spam Begone ...................................................... 33
  13.1 Features ............................................................ 34
  13.2 Installation ......................................................... 34
  13.3 Usage .............................................................. 35
  13.4 Contact and about .............................................. 38
  13.5 License ............................................................ 38
isbg is a script and a python module 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. So this is the perfect tool to take good care of your ISP mailbox without having to leave it.

You can read the full documentation in Read the docs.
• Works with all common IMAP servers
• Works on Linux, MacOS X and Windows (even smartphones!)
• Can do IMAP over SSL
• Can remember your password
• Will work painlessly against multiple IMAP accounts and servers
• Is not involved in the mail delivery process, and so can run on any machine that can contact your IMAP server
• Highly configurable
• Sensible defaults so you don’t have to do any configuring :-)  
• Possibility to skip spam detection to stick only to the teach feature
• Don’t fail when meeting horrible and bad formed mail
• Lock file to prevent multiple instance to run at the same time (for cron jobs)
isbg install a python package module and also a script to use it isbg, it also install another script to unwrap messages: isbg_sa_unwrap.

## 2.1 Dependencies

isbg is written in the Python language. Python is installed by default on most Linux systems. You can find out more about Python at the Python home page.

Make sure you have SpamAssassin installed. All the necessary information can be found on the SpamAssassin wiki. SpamAssassin should be on your $PATH (it installs in /usr/bin/ by default)

To run, isbg also depends on some python modules.

- docopt for command line options.
- cchardet or chardet for encoding detection.
- xdg to found the .cache directory. xdg is not required, if it’s not installed, isbg will try to found .cache.

## 2.2 Install from source

From the main directory where you have download isbg, run:

```bash
$ python setup.py install --record installed_files.txt
```

It will install under /usr/local/. In installed_files.txt there should be the list of files installed. To uninstall them, use:

```bash
$ tr \n \0 < installed_files.txt | xargs -0 rm -vf --
```

In windows systems, you can build a windows installer using:
$ python setup.py bdist_wininst

## 2.3 install with PyPi

You also can install *isbg* from *Pypi*:

$ pip install isbg

To see the files installed you can use:

$ pip show isbg --files

And to uninstall it:

$ pip uninstall isbg
3.1 SpamAssassin

If you have never used SpamAssassin before, you’ll probably be quite nervous about it being too good and taking out legitimate email, or not taking out enough spam. It has an easily adjustable threshold to change how aggressive it is. Run the following command to create your preferences file:

```
$ spamassassin </dev/null >/dev/null
Created user preferences file: /home/rogerb/.spamassassin/user_prefs
```

You can then edit `$HOME/.spamassassin/user_prefs` and change the thresholds. You can also edit the system-wide settings in `/etc/spamassassin/locals.cf`. If you want to use the `--learnspambox` or `--learnhambox`, you’ll have to configure your spamassassin.

### 3.1.1 Configure your spamassassin

If you want to use `--learnspambox` or `--learnhambox` features, you have to add this configuration:

#### Allow Tell

You have to start `spamd` with the `--allow-tell` option. On Debian systems (Debian and Ubuntu), you have to edit `/etc/default/spamassassin` and replace:

```
OPTIONS="-D --create-prefs --max-children 5 --helper-home-dir"
```

by:

```
OPTIONS="-D --allow-tell --create-prefs --max-children 5 \ 
--helper-home-dir"
```
Don’t forget to restart your `spamd` server after that (`sudo service spamassassin restart on Debian`).

### 3.2 CLI Options

The default behavior of `isbg` is to not make any changes your Inbox unless you specify specific command line options. Consequently you can experiment without worry at the beginning.

Your first step is to create a new folder to receive suspected spam. I use one named ‘spam’.

Run `isbg` with the `--help` option to see what options are available or check its manual page with `$ man isbg`.

You can also unwrap `SpamAssassin` bundled emails with `isbg_sa_unwrap`, you can use the `--help` option to check the available options or `$ man isbg_sa_unwrap` to check its manual page.

### 3.3 How does it work?

IMAP assigns each message in a folder a unique id. `isbg` scans the folder for messages it hasn’t seen before, and for each one, downloads the message and feeds it to `SpamAssassin`. If `SpamAssassin` says the message is spam, then the `SpamAssassin` report is uploaded into your spam folder. Unless you specify the `--noreport` option, in which case the message is copied from your Inbox to the Spam folder (the copy happens on the IMAP server itself so this option is good if you are on a low bandwidth connection).

### 3.4 Multiple accounts

By default `isbg` saves the list of seen IMAP message unique IDs in a file in your home directory. It is named `.isbg-trackXXXX` where `XXXX` is a 16 byte identifier based on the IMAP host, username and port number. Consequently you can just run `isbg` against different servers/accounts and it will automatically keep the tracked UIDs separate. You can override the filename with `--trackfile`.

To run `isbg` for multiple accounts one after another, it is possible to use bash scripts like the ones in the folder “bash_scripts”. Since these scripts contain passwords and are thus sensitive data, make sure the file permissions are very restrictive.

### 3.5 Saving your password

If you don’t want `isbg` to prompt you for your password each time, you can specify the `--savepw` option. This will save the password in a file in your home directory. The file is named `~/.cache/isbg/.isbg-XXXX` where `XXXX` is a 16 byte identifier based on the IMAP host, username and port number (the same as for the multiple accounts description above). You can override the filename with `--passwdfilename`.

The password is obfuscated, so anyone just looking at the contents won’t be able to see what it is. However, if they study the code to `isbg` then they will be able to figure out how to de-obfuscate it, and recover the original password. (`isbg` needs the original password each time it is run as well).

Consequently you should regard this as providing minimal protection if someone can read the file.

---

1 You can see it in `Manual page for isbg` page or if you are reading this from github manpage.isbg doc
2 You can see it in `Manual page for isbg_sa_unwrap` page or if you are reading this from github manpage.isbg_sa_unwrap doc
### 3.6 SSL

**isbg** can do IMAP over SSL if your version of Python has been compiled with SSL support. Since Python 2.6, SSL comes built in with Python.

However you should be aware that the SSL support does NOT check the certificate name nor validate the issuer. If an attacker can intercept the connection and modify all the packets flowing by, then they will be able to pose as the IMAP server. Other than that, the connection will have the usual security features of SSL.

### 3.7 Read and Seen flags

There are two flags IMAP uses to mark messages, **Recent** and **Seen**. **Recent** is sent to the first IMAP client that connects after a new message is received. Other clients or subsequent connections won’t see that flag. The **Seen** flag is used to mark a message as read. IMAP clients explicitly set **Seen** when a message is being read.

Pine and some other mailers use the **Recent** flag to mark new mail. Unfortunately this means that if **isbg** or any other IMAP client has even looked at the Inbox, the messages won’t be shown as new. It really should be using **Seen**.

The IMAP specification does not permit clients to change the **Recent** flag.

### 3.8 Gmail Integration

**Gmail** has a few unique ways that they interact with a mail client. **isbg** must be considered to be a client due to interacting with the Gmail servers over IMAP, and thus, should conform to these special requirements for proper integration.

There are two types of deletion on a **Gmail** server.

- **Type 1**: Move a message to ‘[Gmail]/Trash’ folder.
  
  This “removes all labels” from the message. It will no longer appear in any folders and there will be a single copy located in the trash folder. Gmail will “empty the trash” after the received email message is 30 days old.
  
  You can also do a “Normal IMAP delete” on the message in the trash folder to cause it to be removed permanently without waiting 30 days.

- **Type 2**: Normal IMAP delete flag applied to a message.
  
  This will “remove a single label” from a message. It will no longer appear in the folder it was removed from but will remain in other folders and also in the “All Mail” folder.

Enable Gmail integration mode by passing **--gmail** in conjunction with **--delete** on the command line when invoking **isbg**. These are the features which are tweaked:

- The **--delete** command line switch will be modified so that it will result in a Type 1 delete.
- The **--deletetigherthan** command line switch will be modified so that it will results in a Type 1 delete.
- If **--learnspambox** is used along with the **--learnthendestroy** option, then a Type 1 delete occurs leaving only a copy of the spam in the Trash.
- If **--learnhambox** is used along with the **--learnthendestroy** option, then a Type 2 delete occurs, only removing the single label.

Reference information was taken from **gmail IMAP usage**.
3.9 Ignored emails

By default, *isbg* ignores emails that are bigger than 120,000 bytes since spam are not often that big. If you ever get emails with score of 0 on 5 (0.0/5.0), it is likely that *SpamAssassin* is skipping it due to size.

Default maximum size can be changed with the use of the *--maxsize* option.

3.10 Partial runs

By default, *isbg* scans 50 emails for operation: spam learn, ham learn and spam detection. If you want to change the default, you can use the *--partialrun* option specifying the number. *isbg* tries to read first the new messages and tracks the before seen to not reprocess them.

This is useful when your inbox has a lot of emails, since deletion and mail tracking are only performed at the end of the run and full scans can take too long.

If you want that *isbg* does track all the emails you can disable the *partialrun* with *--partialrun=0*. 
Please join our isbg mailing list if you use isbg or contribute to it! The mailing list will be used to announce project news and to discuss the further development of isbg.

You can also hang out with us on IRC, at #isbg on Freenode.

See the CONTRIBUTORS file in the git repository for more information on who wrote and maintains this software.
This program is licensed under the GNU General Public License version 3.
This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.
CHAPTER 6

6.1 Unreleased

6.1.1 New in 2.1.0

• Use of standard localization folders. You must store your password again.
• Added documentation in Read the docs.
• Added a default \texttt{--partialrun} of 50.

6.1.2 New in 1.00

\textbf{DEPRECATION NOTICE}: if you used the \texttt{--ssl} parameter in 0.99, you now need to stop using it! SSL is now used by default. If you want not to use it, please use the \texttt{--nossl} parameter.

• The CLI interface is now implemented with docopt.
• The README now includes the documentation.
• New command \texttt{--imaplist} lists the directories in IMAP account.
• Code now follows PEP-8 style guide.
• Renamed variables to be consistent.
• Added gmail integration (thanks to Orkim!).
• Added bash scripts for use with multiple accounts.
• SSL is now used by default and \texttt{--ssl} parameter is now a \texttt{--nossl} parameter.
• New command \texttt{--trackfile} now permits trackfile name configuration (thanks naevtamarkus!).
• New command \texttt{--partialrun} now enable isbg to run for only a few emails (thanks naevtamarkus!).

\footnote{To read a more detailed information about changes, see \textit{Changelog}.}
What has been done since last release.

## 7.1 Unreleased

- run spamassassin only once per mail check (thanks to rsmuc)

## 7.2 isbg 2.1.3 (20181024)

- don’t build the documentation during the install
- fix pip builds

## 7.3 isbg 2.1.2 (20180917)

- fix a bug with the interaction between –learnspambox and –dryrun
- remove mention of Python 2 in the docs

## 7.4 isbg 2.1.1 (201808729)

- Fix documentation
- Update setup.py in order to upload to PyPi
7.5 isbg 2.1.0 (20180614)

- Don’t raise a error when learning a too big email spam (spamc return code 98).
- Password is now stored as json data file.
- Using standard cache `xdg_cache_home` to store information, usually `$HOME/cache/isbg/`. If you have stored your password with and old releases you should re-store it and delete the old stored files, you should find the old ones as `$HOME/isbg`.
- We use `chardet` and `chardet` if installed for better detect the mails encoding.
- A more robust package, it can be used as a module and a script.
- Using `isbg/__main__` or `python -m isbg` to run as script.
- `python3` support.
- Modularized.
- Added a default –partialrun of 50. Use `--partialrun=0` to retain the old behaviour.
- Renamed `sa_unwrap` script to `isbg_sa_unwrap`, added command line options. Now it also depends of docopt.
- Documentation:  * Use `sphinx` and support to upload it to Read the docs.  * Added development documentation.  * Added `manpages`, built with `sphinx`.
- Examples: added a crontab example in the `bash_scripts` directory.

7.6 Old Releases

7.6.1 isbg 1.00 (20160106)

- The CLI interface is now implemented with docopt
- The README now includes the documentation
- New command –imaplist lists the directories in IMAP account
- Code now follows PEP-8 style guide
- Renamed variables to be consistent
- Added gmail integration (thanks to Orkim!)
- Added bash scripts for use with multiple accounts
- SSL is now used by default and “–ssl” parameter is now a “–nossl” parameter
- New command “–trackfile” now permits trackfile name configuration (thanks naevtamarus!)  
- New command “–partialrun” now enable isbg to run for only a few emails (thanks naevtamarus!)

7.6.2 isbg 0.99 (20100303)

- Drastic speed enhancement (thanks to Ajenbo!)
- deletehighterthen, fix expunge, movehamto (thanks to Ajenbo!)
- Learn spam/ham before scanning
- Call IMAP SEARCH command instead of fetching and checking manually
• Ignore lockfile when too old (4h by default)
• Switch to ignore lockfile
• Bug fix: SSL now work.
• Don’t crash anymore when parsing weird attachment (like MS Office files)

7.6.3 isbg 0.98 (20091201)

• Teach spam and ham from specific folders
• Compatibility from py2.4 to py2.6
8.1 Other stuff

- Export documentation.
- Add documentation about development.
- Easily upload to pypi.
- Auto report messages to Razor (high scoring ones that are definitely spam)
- Separate out messages that may be false positives (scores close to SpamAssassin thresholds) from the definite spam ones.
- Add documentation for the crontab usage.
- Remove the SpamAssassin report from a mail when it is in the ham folder.

8.1.1 Notes

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Dec.</th>
<th>Hex</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>9744</td>
<td>2610</td>
<td>2610</td>
<td>BALLOT BOX</td>
</tr>
<tr>
<td>9745</td>
<td>2611</td>
<td>2611</td>
<td>BALLOT BOX WITH CHECK</td>
</tr>
<tr>
<td>9746</td>
<td>2612</td>
<td>2612</td>
<td>BALLOT BOX WITH X</td>
</tr>
</tbody>
</table>
isbg uses github flow and, from v2.1.0 we use git-hub to deal with pulls and issues.

### 9.1 Development schema

We should work in a feature/* branch or bugfix/* branch and it should be attached to an issue. We can use git-hub issue to do it, e.g.:

```bash
$ git hub issue new --assign USER  # It creates the issue number 2
$ git branch feature/DESCRIPTION  # Creates the new branch
$ git checkout feature/DESCRIPTION # Change to the new branch
$ ...                            # DO changes and commit them
$ # Push the changes to github:
$ git push --set-upstream origin feature/DESCRIPTION
$ # Attach the changes to issue 2:
$ git hub pull attach -c master 2 feature/DESCRIPTION
```

If the git hub pull command doesn’t work, you can use the github web interface to do the pull request and attach it adding #2 to the pull request.

The working and stable features should be merged to master before a new release, and also closed. To close the user closes 2 in the commit message.

### 9.2 Versioning schema

We tag the new releases as:

```
v{major_release_number}.{minor_release_number}.{patch_release_number}
```

The current version number of isbg is stored in isbg/isbg.py
9.3 Releasing Schema

You should:

1. Update the the `__version__` var ./isbg/isbg.py.
2. Update ./NEWS.rst
3. Update ./Changelog.rst
4. Check the ./TODO.rst list and updated it.
5. Check if some changes should be updated in ./README.rst
6. If new files have been added or removed: Check ./MANIFEST.in.
7. If dependencies have been updated, added or removed check: ./setup.py, ./requirements.txt and/or ./requirements-build.txt.
8. Commit it to master.
9. Tag the new version in isbg github releases, add the news added for this version in NEWS.rst to the comment. Remember to add ‘v’ at the start of the version.
10. Login to readthedocs and update the stable version.
10.1 SYNOPSIS

isbg –imaphost <hostname> –imapuser <username> [options]
isbg –imaphost <hostname> –imapuser <username> –imaplist [options]
isbg (-h | --help)
isbg –usage
isbg –version

10.2 DESCRIPTION

isbg scans an IMAP Inbox and runs every entry against SpamAssassin. For any entries that match, the message is copied to another folder, and the original marked or deleted.

Works with all common IMAP servers, can do IMAP over SSL, can remember your password, will work painlessly against multiple IMAP accounts and servers, is not involved in the mail delivery process, and so can run on any machine that can contact your IMAP server and has the possibility to skip spam detection to stick only to the teach feature.

10.3 OPTIONS

–imaphost hostname  IMAP server name
–imapuser username  Who you login as
–imaplist  List imap directories
-h, --help  Show the help screen
--usage  Show usage information
--version  Show version information
--dryrun  Do not actually make any changes
--delete  The spams will be marked for deletion from your inbox
--deletehigherthan #  Delete any spam with a score higher than #
--exitcodes  Use exitcodes to detail what happened
--expunge  Cause marked for deletion messages to also be deleted (only useful if --delete is specified)
--flag  The spams will be flagged in your inbox
--gmail  Delete by copying to ‘[Gmail]/Trash’ folder
--ignorelockfile  Don’t stop if lock file is present
--imappassword passwd  IMAP account password. This however is a really bad idea since any user on the system can run ps and see the command line arguments
--imapport port  Use a custom port
--imapinbox mbox  Name of your inbox folder [Default: INBOX]
--learnspambox mbox  Name of your learn spam folder
--learnhambox mbox  Name of your learn ham folder
--learnthendestroy  Mark learnt messages for deletion
--learnthenflag  Flag learnt messages
--learnunflagged  Only learn if unflagged (for --learnthenflag)
--lockfilegrace=<min>  Set the lifetime of the lock file to [Default: 240.0]
--lockfilename  Override the lock file name
--maxsize numbytes  Messages larger than this will be ignored as they are unlikely to be spam
--movehamto mbox  Move ham to folder
--noninteractive  Prevent interactive requests
--noreport  Don’t include the SpamAssassin report in the message copied to your spam folder
--nostats  Don’t print stats
--partialrun num  Stop operation after scanning ‘num’ unseen emails [Default: 50]. You can run isbg without --partialrun with --partialrun=0
--passwdfilename file  Use a file to supply the password
--savepw  Store the password to be used in future runs. This will save the password in a file in your home directory. The file is named .isbg-XXXX where XXXX is a 16 byte identifier based on the IMAP host, username and port number (the same as for the multiple accounts description above). You can override the filename with --passwdfilename. The password is obfuscated, so anyone just looking at the contents won’t be able to see what it is. However, if they study the code to isbg then they will be able to figure out how to de-obfuscate it, and recover the original password. (isbg needs the original password each time it is run as well). Consequently you should regard this as providing minimal protection if someone can read the file.
--spamc  Use spamc instead of standalone SpamAssassin binary
--spaminbox mbox  Name of your spam folder [Default: INBOX.Spam]
--nossi  Don’t use SSL to connect to the IMAP server
--teachonly  Don’t search spam, just learn from folders
--trackfile file  Override the trackfile name
--verbose  Show IMAP stuff happening
--verbose-mails  Show mail bodies (extra-verbose)

(Your inbox will remain untouched unless you specify --flag or --delete)

10.4  EXAMPLES

10.4.1  Do your first run

$ isbg --imaphost mail.foo.com --imapuser rogerb@mail.foo.com
   --imaplist --savepw

   IMAP password for rogerb@mail.foo.org@mail.foo.org:

Will request the password for your user account and store it obfuscated for future use, after login, it will show the
IMAP folder list:

[u' INBOX''', u' INBOX.Esborranys''', u' INBOX.Spam''', u' INBOX.Sent''', u' INBOX.NOSPAM''', u'
   INBOX.Archive''', u' INBOX.Drafts''', u' INBOX.Trash''', u' INBOX.Paperera''']

10.4.2  Scan your account for spam

In future uses you can scan for spam with:

    isbg --imaphost mail.foo.com --imapuser rogerb@mail.foo.com

After some time, it will return the stats:

   0 spams found in 0 messages

   0/0 was automatically deleted

10.5  OVERVIEW

The amount of time it takes will be proportional to the size of your inbox and the amount of mails specified with
--partialrun. You can specify --verbose if you want to see the gory details of what is going on.

You can now examine your spam folder and will see what spam was detected. You can change the SpamAssassin
threshold in your user_prefs file it created earlier.

isbg remembers which messages it has already seen, so that it doesn’t process them again every time it is run. If you
are testing and do want it to run again, then remove the trackfile (default $HOME/.cache/isbg/track*).

If you specified --savepw then isbg will remember your password the next time you run against the same server with
the same username. You should not specify --savepw in future runs unless you want to change the saved password.

You’ll probably want something to actually be done with the original spams in your inbox. By default nothing happens
to them, but you have two options available. If you specify --flag then spams will be flagged.

You can get the messages marked for deletion by specifying --delete. If you never want to see them in your inbox,
also specify the --expunge option after --delete and they will be removed when isbg logs out of the IMAP server.
10.5.1 SpamAssassin

If you have never used SpamAssassin before, you'll probably be quite nervous about it being too good and taking out legitimate email, or not taking out enough spam. It has an easily adjustable threshold to change how aggressive it is. Run the following command to create your preferences file ($HOME/.spamassassin/user_prefs):

```
$ spamassassin </dev/null >/dev/null
```

10.5.2 Your Folder Names

Each IMAP implementation names their folders differently, and most IMAP clients manage to hide most of this from you. If your IMAP server is Courier, then your folders are all below INBOX, and use dots to separate the components. The UWash server typically has the folders below Mail and uses slash (/) to separate components.

If you don't know how your IMAP folders are implemented, you can always use the --imaplist option to find out.

10.6 SEE ALSO

spamassassin(1), Mail::SpamAssassin::Conf(3).

The full documentation for isbg is maintained in https://isbg.readthedocs.io/

10.7 EXIT CODES

0  All went well.
10  There were errors in the command line arguments.
11  The IMAP server reported an error or error with the IMAP connection.
12  There was an error of communication between spamc or SpamAssassin.
20  The program was not launched in an interactive terminal.
30  Error with the lock file, is another instance of isbg must be running.
-1  Other errors.

With --exitcodes there are also:

1  There was at least one new message, and none of them were spam.
2  There was at least one new message, and all them were spam.
3  There were new messages, with at least one spam and one non-spam.

10.8 BUGS

You can report bugs on https://github.com/isbg/isbg/issues
11.1 SYNOPSIS

isbg_sa_unwrap [-from <FROM_FILE>] [-to <TO_FILE>]
isbg_sa_unwrap (-h | --help)
isbg_sa_unwrap --usage
isbg_sa_unwrap.py --version

11.2 DESCRIPTION

isbg_sa_unwrap unwrap a mail bundled by SpamAssassin.
it parses a rfc2822 email message and unwrap it if contains spam attached.

11.3 OPTIONS

-h, --help    Show the help screen
--usage       Show usage information
--version     Show version information

-f FILE, --from=FILE    Filename of the email to read and unwrap. If not informed, the stdin will be used
-t FILE, --to=FILE     Filename to write the unwrapped email. If not informed, the stdout will be used
11.4 SEE ALSO

spamassassin(1), Mail::SpamAssassin::Conf(3).

The full documentation for isbg is maintained in https://isbg.readthedocs.io/

11.5 BUGS

You can report bugs on https://github.com/isbg/isbg/issues
12.1 isbg

12.1.1 isbg package

Submodules

isbg.__main__ module

isbg.imaputils module

isbg.isbg module

isbg.sa_unwrap module

isbg.secrets module

isbg.spamproc module

isbg.utils module

Module contents

12.2 Main Objects and functions

`isbg.learn_mail`

`isbg.test_mail`

Continued on next page
12.3 Indices and tables

- genindex
- modindex
- search
isbg is a script and a python module 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. So this is the perfect tool to take good care of your ISP mailbox without having to leave it.

You can read the full documentation in Read the docs.

Table of Contents

- IMAP Spam Begone
  - Features
  - Installation
    - Dependencies
    - Install from source
    - install with PyPi
  - Usage
    - SpamAssassin
    - CLI Options
    - How does it work?
    - Multiple accounts
    - Saving your password
    - SSL
    - Read and Seen flags
13.1 Features

- Works with all common IMAP servers
- Works on Linux, MacOS X and Windows (even smartphones!)
- Can do IMAP over SSL
- Can remember your password
- Will work painlessly against multiple IMAP accounts and servers
- Is not involved in the mail delivery process, and so can run on any machine that can contact your IMAP server
- Highly configurable
- Sensible defaults so you don’t have to do any configuring :-)  
- Possibility to skip spam detection to stick only to the teach feature
- Don’t fail when meeting horrible and bad formed mail
- Lock file to prevent multiple instance to run at the same time (for cron jobs)

13.2 Installation

isbg was install a python package module and also a script to use it isbg, it also install another script to unwrap messages: isbg_sa_unwrap.

13.2.1 Dependencies

isbg is written in the Python language. Python is installed by default on most Linux systems. You can find more about Python at Python home page.

Make sure you have SpamAssassin installed. All the necessary information can be found on the SpamAssassin wiki. SpamAssassin should be on your $PATH (it installs in /usr/bin/ by default).

To run, isbg also depends on some python modules.

- docopt for command line options.
- cchardet or chardet for encoding detection.
- xdg to found the .cache directory. xdg is not required, if it’s not installed, isbg will try to found .cache.
13.2.2 Install from source

From the main directory where you have download isbg, run:

```
$ python setup.py install --record installed_files.txt
```

It will install under /usr/local/. In installed_files.txt there should be the list of files installed. To uninstall them, use:

```
$ tr \n \0 < installed_files.txt | xargs -0 rm -vf --
```

In windows systems, you can build a windows installer using:

```
$ python setup.py bdist_wininst
```

13.2.3 install with PyPi

You also can install isbg from Pypi:

```
$ pip install isbg
```

To see the files installed you can use:

```
$ pip show isbg --files
```

And to uninstall it:

```
$ pip uninstall isbg
```

13.3 Usage

13.3.1 SpamAssassin

If you have never used SpamAssassin before, you’ll probably be quite nervous about it being too good and taking out legitimate email, or not taking out enough spam. It has an easily adjustable threshold to change how aggressive it is. Run the following command to create your preferences file:

```
$ spamassassin </dev/null >/dev/null
```

Created user preferences file: /home/rogerb/.spamassassin/user_prefs

You can then edit $HOME/.spamassassin/user_prefs and change the thresholds.

You can also edit the system-wide settings in /etc/spamassassin/locals.cf.

If you want to use the --learnspambox or --learnhambox features, you'll have to configure your spamassassin.

**Configure your spamassassin**

If you want to use --learnspambox or --learnhambox features, you have to add this configuration:
Allow Tell

You have to start spamd with the --allow-tell option.

On Debian systems (Debian and Ubuntu), you have to edit /etc/default/spamassassin and replace:

```bash
OPTIONS="-D --create-prefs --max-children 5 --helper-home-dir"
```

by:

```bash
OPTIONS="-D --allow-tell --create-prefs --max-children 5 --helper-home-dir"
```

Don’t forget to restart your spamd server after that (sudo service spamassassin restart on Debian).

13.3.2 CLI Options

The default behavior of isbg is to not make any changes your Inbox unless you specify specific command line options. Consequently you can experiment without worry at the beginning.

Your first step is to create a new folder to receive suspected spam. I use one named ‘spam’.

Run isbg with the --help option to see what options are available or check its manual page with $ man isbg.

You can also unwrap SpamAssassin bundled emails with isbg_sa_unwrap, you can use the --help option to check the available options or $ man isbg_sa_unwrap to check its manual page.

13.3.3 How does it work?

IMAP assigns each message in a folder a unique id. isbg scans the folder for messages it hasn’t seen before, and for each one, downloads the message and feeds it to SpamAssassin. If SpamAssassin says the message is spam, then the SpamAssassin report is uploaded into your spam folder. Unless you specify the --noreport option, in which case the message is copied from your Inbox to the Spam folder (the copy happens on the IMAP server itself so this option is good if you are on a low bandwidth connection).

13.3.4 Multiple accounts

By default isbg saves the list of seen IMAP message unique IDs in a file in your home directory. It is named .isbg-trackXXXX where XXXX is a 16 byte identifier based on the IMAP host, username and port number. Consequently you can just run isbg against different servers/accounts and it will automatically keep the tracked UIDs separate. You can override the filename with --trackfile.

To run isbg for multiple accounts one after another, it is possible to use bash scripts like the ones in the folder “bash_scripts”. Since these scripts contain passwords and are thus sensitive data, make sure the file permissions are very restrictive.

13.3.5 Saving your password

If you don’t want isbg to prompt you for your password each time, you can specify the --savepw option. This will save the password in a file in your home directory. The file is named $HOME/.cache/isbg/.isbg-XXXX
where XXXX is a 16 byte identifier based on the IMAP host, username and port number (the same as for the multiple accounts description above). You can override the filename with --passwdfilename.

The password is obfuscated, so anyone just looking at the contents won’t be able to see what it is. However, if they study the code to isbg then they will be able to figure out how to de-obfuscate it, and recover the original password. (isbg needs the original password each time it is run as well).

Consequently you should regard this as providing minimal protection if someone can read the file.

### 13.3.6 SSL

isbg can do IMAP over SSL if your version of Python has been compiled with SSL support. Since Python 2.6, SSL comes built in with Python.

However you should be aware that the SSL support does NOT check the certificate name nor validate the issuer. If an attacker can intercept the connection and modify all the packets flowing by, then they will be able to pose as the IMAP server. Other than that, the connection will have the usual security features of SSL.

### 13.3.7 Read and Seen flags

There are two flags IMAP uses to mark messages, **Recent** and **Seen**. **Recent** is sent to the first IMAP client that connects after a new message is received. Other clients or subsequent connections won’t see that flag. The **Seen** flag is used to mark a message as read. IMAP clients explicitly set **Seen** when a message is being read.

Pine and some other mailers use the **Recent** flag to mark new mail. Unfortunately this means that if isbg or any other IMAP client has even looked at the Inbox, the messages won’t be shown as new. It really should be using **Seen**.

The IMAP specification does not permit clients to change the **Recent** flag.

### 13.3.8 Gmail Integration

Gmail has a few unique ways that they interact with a mail client. isbg must be considered to be a client due to interacting with the Gmail servers over IMAP, and thus, should conform to these special requirements for proper integration.

There are two types of deletion on a Gmail server.

- **Type 1**: Move a message to ‘[Gmail]/Trash’ folder.
  This “removes all labels” from the message. It will no longer appear in any folders and there will be a single copy located in the trash folder. Gmail will “empty the trash” after the received email message is 30 days old.
  You can also do a “Normal IMAP delete” on the message in the trash folder to cause it to be removed permanently without waiting 30 days.

- **Type 2**: Normal IMAP delete flag applied to a message.
  This will “remove a single label” from a message. It will no longer appear in the folder it was removed from but will remain in other folders and also in the “All Mail” folder.

Enable Gmail integration mode by passing --gmail in conjunction with --delete on the command line when invoking isbg. These are the features which are tweaked:

- The --delete command line switch will be modified so that it will result in a Type 1 delete.
- The --deletheigherthan command line switch will be modified so that it will results in a Type 1 delete.
– If `--learnspambox` is used along with the `--learnthendestroy` option, then a Type 1 delete occurs leaving only a copy of the spam in the Trash.

– If `--learnhambox` is used along with the `--learnthendestroy` option, then a Type 2 delete occurs, only removing the single label.

Reference information was taken from gmail IMAP usage.

### 13.3.9 Ignored emails

By default, `isbg` ignores emails that are bigger than 120,000 bytes since spam are not often that big. If you ever get emails with score of 0 on 5 (0.0/5.0), it is likely that SpamAssassin is skipping it due to size.

Default maximum size can be changed with the use of the `--maxsize` option.

### 13.3.10 Partial runs

By default, `isbg` scans 50 emails for operation: spam learn, ham learn and spam detection. If you want to change the default, you can use the `--partialrun` option specifying the number. `isbg` tries to read first the new messages and tracks the before seen to not reprocess them.

This is useful when your inbox has a lot of emails, since deletion and mail tracking are only performed at the end of the run and full scans can take too long.

If you want that `isbg` does track all the emails you can disable the `partialrun` with `--partialrun=0`.

### 13.4 Contact and about

Please join our `isbg` mailing list if you use `isbg` or contribute to it! The mailing list will be used to announce project news and to discuss the further development of `isbg`.

You can also hang out with us on IRC, at `#isbg` on Freenode.

See the CONTRIBUTORS file in the git repository for more information on who wrote and maintains this software.

### 13.5 License

This program is licensed under the GNU General Public License version 3.

This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.