mailpile

@MailpileTeam
Bjarni Rúnar Einarsson
FOSDEM 2014

Hi! I'm Bjarni

Icelandic: Bee Yarn Knee :-)

- B.Sc. Comp. Sci. from Uni. Iceland, 2000
- FOSS advocate since Linux 1.2

Full time FOSS developer since 2010

- Today: PageKite and Mailpile
- Previously: Google Site Reliability
- Even previouser: Frisk Software (fighting spam)

What is Mailpile?

An e-mail client: (not a mail server... yet)

- A web based user interface & API
- A powerful search engine
- An easy way to use PGP
- Free Software: AGPLv3 / Apache 2.0
- Python, HTML5, Javascript

A project to “take e-mail back”

Why?

Why another e-mail client?

The state of e-mail

Free e-mail is in bad shape

- RoundCube is “state of the art”
- Zimbra is not everyone's cup of tea
- Thunderbird is being retired!
- Where's the innovation?

Mass encryption is still a distant dream

The state of e-mail: Cloudy

Becoming centralized in the cloud

- GMail, Hotmail, Yahoo, Facebook, ...

- Spam filters: is this not censorship?
- Very good, cheap service
- Comes with spying!

Cloudy e-mail is scary

Edward Snowden said so!

- So did Eben Moglen, Richard Stallman, ...

Worse for freedom than closed source

- They have your data
- Lock-in and natural monopolies abound
- Risk of “embrace-extend-extinguish”
- Incompatible with encryption

Mailpile: The Plan

5 things!

1. Make software FOSS folks enjoy hacking on
2. Make software people want to use
3. Make e-mail encryption understandable
4. Make decentralization easy
   - Including an easy migration path!
5. Find better business models for e-mail
   - Without spying and data-mining

Timeline so far ...

2011

- Bjarni wrote an experimental search engine
- Able to search hundreds of thousands of e-mails on a crappy laptop in milliseconds!

... Bjarni went back to his day job (PageKite)

Timeline so far ...

2013

- Bjarni, Brennan & Smári met at the pool
- Bjarni, Brennan & Smári had coffee and beer
- BB&S came up with **The Plan**!
- Project “launched” in August, at OHM 2013
- Raised $163,000 USD on IndieGogo (+ 54 BTC)
- Work, work, work, work!

Timeline so far ...

2014

- Shipped perks: postcards, stickers, shirts, rocks
- First milestone announced at FOSDEM ...

Mailpile Alpha

(one day late)

Mailpile Alpha

Highlights

- Not Vaporware!
- Elegant HTML5-based user interface
- Fast search engine (... uh oh, not notmuch)
- Support for PGP encryption and signatures
- Bayesian spam filtering
- Translations in progress for over 30 languages

Mailpile Alpha

Lowlights

- It is an alpha, most things need work
- Hard to install and configure
- No IMAP / POP3 support
- No S/MIME support
- No PGP key management
- etc. etc. etc.

Mailpile Alpha

Source code on github:

git clone -b release/alpha \

https://github.com/pagekite/Mailpile.git

Live demos in 14 languages

https://www.mailpile.is/demos/

Demo... ?

How does it work?

Overall architecture

Guiding Principles

- Free software, open standards
- Decentralization, users should own their data
- Searching is critical
- Our primary UI is the web
- Encryption by default (when possible)
- Our end-users are not necessarily techies

Overall architecture

Python Core

- Configuration & contacts
- Search engine
- E-mail: read / write / send
- Crypto: encryption & signatures
- HTTP server & template engine
- Plugin architecture

Overall architecture

Python Plugins

- Multiple mailbox formats
- Importing contacts
- Setup and system integration
- Advanced searches (dates, sizes, ...)
- ...

Overall architecture

Web API

- Most URLs map directly to a Python method
- REST API, returns JSON by default
- Alternate result formats:
  - HTML
  - HTML embedded in JSON
  - Plain text
- HTML rendering via. Jinja2 templating engine

Overall architecture

Web interface

- HTML5 APIs, JQuery, some Bootstrap, LESS, ...
- Progressive enhancement:
  - Basic read-only site works without Javascript
  - Reponsive design for mobile, tablets, etc.

- Will be themable / skinnable

Overall architecture

Alternate interfaces:

- Command line
  - Used for debugging & development
- Python interface
  - For testing and automation
- Someday?
  - XML-RPC
  - ncurses

How does it work:

Mailpile Search

Mailpile Search

Why not notmuch?

- Started as an experiment
- Simple search engine is simple
  - Under 1600 lines of code
  - Easy to extend and modify
  - Takes <5 minutes to explain to any developer

... also, it works and is fast!

Mailpile Search

How does it work?

- Mailpile reads your mail:
  - Messages => IDs, metadata, keywords
- “Posting lists” map keywords to IDs
  - Smallish files on disk
- “Metadata index” maps IDs to metadata
  - <1 KB of data per message in RAM

Most search queries can be answered by reading one file from disk and looking metadata up in RAM!

Mailpile Search

Posting lists

```
get_msg_ids(keyword):
    filename = os.path.join(workdir, hash(keyword))
    return set(open(filename).read().split())
```

- Keyword grouping reduces file counts
- Adding things is fast, deleting is slow

Mailpile Search

Metadata index

```
metadata = {
    msg_id: [pointer, size, subject, to, from, tags, ...],
    ...
}
```

- All the data we need to generate result lists
- Stored GPG encrypted on disk
- Loaded into RAM on startup

Mailpile Search

Look, a 5 line search engine!

```
results = set(all the message IDs)
for kw in query:
    results &= get_msg_ids(kw)  # set intersection

for id in results:
    pretty_print(metadata.get(id))
```

Mailpile Search

Tags

- Keywords with editable lists of IDs
- Live in the metadata
- Used to implement common e-mail metaphors, like “unread”, “inbox”, “spam”, ...

Static filters and “autotagging” plugins can assign tags automatically.

Mailpile Search

Plugins

- New rules for generating keywords
  - Read attachments: pdf, odt, ...?
  - Understand the grammar of your language
- New magic keywords
  - Map “dates:2010..2014” to a list of keywords we can actually search for.

Plugins can also use the search engine to generate interesting views or analyze your mail.

Mailpile Search

The tricky bits

- Actually reading the mail
- Generating useful keywords
- Speed / RAM usage optimizations

Works in progress

- Better query language
- Deleting things from the index
- Better encryption of the posting lists

How does it work:

Mailpile Spam Filtering

Mailpile Spam Filters

How does it work?

- Statistical analysis of incoming mail
- Default engine is **spambayes**
- Spam is auto-tagged with a “spam” tag
- Messages tagged as “spam” are hidden from search results by default
- Training (ham/spam) based on user behavior

Mailpile Spam Filters

Statistical analysis

- “What content is spam to you?”
- Analyze the same keywords as are used by the search engine
- Classify mail into:
  - spam
  - maybe spam
  - ham
- Default engine is **spambayes**

Mailpile Spam Filters

Learning from the user

- When training, which mail is interesting?
  - Manually tagged mail
  - Messages the user bothers to read
  - Messages the user replies to or forwards
  - ... ?
- Mailpile tracks these actions, tags messages
- Tracking tags are used to select ham/spam for training the filter

Mailpile BACON Filters

Wait... bacon?

- There is nothing spam-specific about this
- We can use it to classify other things!
- Proposed UI:
  - Mark any tag as “autotagging”
  - Drop messages on tag
  - Over time Mailpile learns to tag for you

... still in testing, but awesome potential!

How does it work:

Mailpile Crypto

Mailpile Crypto

Where ~~do~~ should we do crypto?

- Encrypting data at rest (settings, index, ...)
- Reading, writing and sending e-mail
- HTTPS, incoming and outgoing
- Anonymizing network downloads (gravatar, ...)
- Proof-of-work (hash-cash) spam prevention?

Mailpile Crypto

Data at rest

- Use GnuPG or OpenSSL to encrypt:
  - Application settings
  - Contacts
  - Search index metadata
  - Search index posting lists
  - Plugin state
  - Drafts and downloaded mail
- GnuPG is slow, use that for the config file
- OpenSSL is fast, use for everything else!

Mailpile Crypto

Reading e-mail

- Parse PGP/MIME and call out to GnuPG
- Pluggable crypto-systems, so plugins can add support for S/MIME and other fun things
- Decrypting be done during indexing, so encrypted contents are searchable
- Encryption and signature state generate metadata (tags) which can be searched for and presented to the user interface

Mailpile Crypto

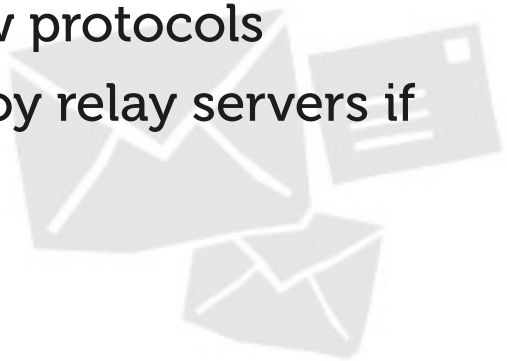
Writing e-mail

- PGP/MIME signatures and encryption
- Best effort security
  - Encrypt when we can
  - Sign by default
  - Unless the user overrides
- TOFU key verification – like ssh. No web of trust!
- Distribute keys by attaching to outgoing mail

Mailpile Crypto

Sending e-mail

- Use STARTTLS whenever possible
- We have this idea we call SMTorP...
  - Build a simple SMTP server into Mailpile
  - Ship Tor with Mailpile to end-users
  - Register the STMP server as a Tor hidden service
  - P2P encypted delivery: you@hrblhshddmnnmthng.onion
  - Closes the “meta-data” leak without new protocols
  - Building on SMTP makes it easy to deploy relay servers if your Mailpile is frequently offline



Mailpile Crypto

HTTPS Everywhere

- Add TLS support to the built-in web server
- Use HTTPS whenever possible, when downloading from the web (gravatar etc.)

Anonymize user traffic

- Ship with Tor
- Use Tor to anonymize downloads from the web
- Anonymize outgoing SMTP when possible

www.mailpile.is

Yay Alpha!