procmail and SpamAssassin

UCLA Linux User Group, February 2004

Ben Clifford

benc@linux.ucla.edu

Outline

Two related themes:

- procmail
  Filtering mail
- SpamAssassin
  Detecting spam

What is procmail for?

Procmail filters incoming mail.

- Match messages based on rules
  - regexp on headers
  - return code of an executable
- Take actions
  - Save mail into a different folder
  - Forward mail to a different address
  - Pipe the mail through an arbitrary program

How does procmail get run?

Procmail can be integrated into your mail system in several different ways.
With recent RedHat installs, all mail runs through procmail automatically, so no extra effort needed to make it run.

Recipes

Procmail recipes go in `~.procmailrc`, separated by blank lines.
A recipe looks like:

```
:0 [flags] [ : [locallockfile] ]
<zero or more conditions>
<exactly one action line>
```

If *all* conditions in the recipe match, then the action is perfomed.

Example: sorting mailing lists

Send incoming postings to the linux kernel mailing list into `/home/benc/mail/lkml`

```
:0 :
* ^X-Mailing-List: .*linux-kernel@vger.kernel.org
/home/benc/mail/lkml
```

- IF there is an `X-Mailing-List:` header containing the string: `linux-kernel@vger.kernel.org`
- THEN save the message into `/home/benc/mail/lkml`

Note regexp syntax.

Processing several recipies in a row

`.procmailrc` can contain many recipies (Usually,) each recipe is processed in sequence until one is found where the conditions match (and so the action fires). Then processing stops. If reach the end of `.procmailrc` without any recipes matching, then deliver the message to usual inbox.

Example: A second rule

Add another rule: any email address of the form spam-*anything*@hawaga.org.uk gets filtered.

```
:0 :
* ^X-Mailing-List: .*linux-kernel@vger.kernel.org
/home/benc/mail/lkml

:0 :
* ^To:.*spam-.*@hawaga.*
/home/benc/mail/spam/addressed-to-spam
```

This filters out linux-kernel messages to the `lkml` folder. Then it filters out anything left that is addressed to `spam-*` and puts it in `spam/addressed-to-spam`

Quiz!

```
:0 :
* ^X-Mailing-List: .*linux-kernel@vger.kernel.org
/home/benc/mail/lkml

:0 :
* ^To:.*spam-.*@hawaga.*
/home/benc/mail/spam/addressed-to-spam
```

Q: What happens if a message matches both?

```
To: spam-example@hawaga.org.uk
X-Mailing-List: Linux Kernel List
                <linux-kernel@vger.kernel.org>
```

A: The first rule applies – the message goes to `lkml`, not to `addressed-to-spam`

Example: Piping mail to a program

New mail is announced through a speech synthesiser script

```
:0 c:
|/home/benc/bin/newmail
```

- No conditions: the recipe *always* matches.
- | means pipe message to specified executable
- *c* flag: The rule is *non-delivering*. Processing will continue after this rule instead of stopping.

formail

A tool that does various interesting things with mail.
Comes with procmail.
It can:

- keep track of of messages that have already been seen
- extract fields from headers
- other stuff. . .

Eliminating dupes

Problem: Some messages are crossposted to several lists but we only want to see the message once.

Solution: (two rules)

```
:0 wc: msgid.lock
| formail -D 8192 msgid.cache

:0 a:
mail/duplicates
```

Eliminating dupes – rule 1

```
:0 whc: msgid.lock
| formail -D 8192 msgid.cache
```

- No conditions – so we always perform action
- Pipe mail through formail
- `formail -D` – has formail seen this message-id before? Returns success if it has.
- `w` flag – Wait for filter to exit
  `c` flag – Processing continues after this rule
- Use `msgid.lock` as lock file

Eliminating dupes – rule 2

```
:0 a:
mail/duplicates
```

- `a` flag – Only perform the action if previous rule was successful. In this case, if formail has already seen the message-id.
- action is send the mail to `mail/duplicates` rather than deliver to inbox.

More procmailrc syntax

- Comments

```
# I like cheese.
```

- Environment variables
- Action can be a sub block of rules

Simple executable virus trapping

```
:0
* ^Content-Type: multipart/(alternative|mixed)
{
:0 $Lock
* B ?? ^Content-Type: \
(audio/x-|application).*;.*($.*)?name=.*\..*\.(scr|com|b
/home/benc/mail/virus
}
```

TODO: explain

End of procmail

For information:
Examples in `man procmailex`
Config syntax (much more than here) in `man procmailrc`

Questions!

SpamAssassin

SpamAssassin detects spam.

SpamAssassin

Key ideas:

- Lots of rules, each giving a small score. Total score is used to decide. Difficult to trick.
- SpamAssassin does not delete spam – it detects it and labels it. You need to use something else to filter/delete spam (eg. procmail)

How to run SpamAssassin

Like procmail, several ways:

- Some mail servers run every message through SpamAssassin. (eg. `xorn.linux.ucla.edu`)
- You might need to run it yourself, from .procmailrc (eg. a normal RedHat install)
  - standalone mode (`spamassassin` command)
  - client/server mode (`spamc` command, spamd run in background)

Integrating with procmail

Two procmail rules:

1. Feed all incoming mail through spamassassin. (unnecessary if all mail to goes through a systemwide SpamAssassing)
2. Take action based on SpamAssassin's opinion – it is a BAD idea to automatically delete mail. Better to move it into a separate folder.

Feed incoming through spamassassin

This rule filters every message through spamassassin and then continues processing the rest of .procmailrc. This gives spamassassin an opportunity to label spam.

```
:0fw :
| /usr/bin/spamc -d localhost
```

On some systems, this step is unnecessary as sysadmin has configured all mail to go through spamassassin.

Filter out spam

SpamAssassin adds some extra headers to every message it filters.
Example for non-spam:

```
X-Spam-Checker-Version: SpamAssassin 2.60 (1.212-2003-09-23-exp) on
        barbarella.hawaga.org.uk
X-Spam-Level:
X-Spam-Status: No, hits=-4.9 required=3.0 tests=BAYES_00
        autolearn=ham
        version=2.60
```

Example for spam:

```
X-Spam-Flag: YES
X-Spam-Checker-Version: SpamAssassin 2.60 (1.212-2003-09-23-exp)
        on barbarella.hawaga.org.uk
X-Spam-Level: ******
X-Spam-Status: Yes, hits=6.8 required=3.0 tests=AWL,BAYES_99,
        CLICK_BELOW, EARN_MONEY,HTML_50_60,HTML_FONTCOLOR_RED,
        HTML_FONTCOLOR_UNSAFE, HTML_MESSAGE,HTML_TAG_BALANCE_A
        autolearn=no version=2.60
```

Filter out spam

Use a procmail rule to filter out the spam that SpamAssassin has labelled.

```
:0:
* ^X-Spam-Status: Yes
mail/spamassassin
```

Matches `X-Spam-Status` header and sends to `mail/spamassassin` instead of delivering to inbox.

BAD IDEA

It is a bad idea to automatically delete mail based on SpamAssassin's opinion.
Bad bad bad
Do not delete mail automatically when SpamAssassin indicates the mail is spam.
It is much better to redirect it to a different folder, where you may peruse it for *false postitives* at your leisure.
This folder is also useful for Bayesian training (later...)

MS-Outlook with SpamAssassin

MS-Outlook can't filter on arbitrary headers, but we can match on the SpamAssassin spam level and rewrite the subject line if the mail is spam. (Outlook can filter on the subject line)

```
# make the spam mails sortable by Outlook
:0 fhw
* ^X-Spam-Level: \*\*\*\*\*
| sed -e '/^Subject:/s/Subject: \(.*\)/Subject: ****SPAM
```

SpamAssassin config file

`~/.spamassassin/user_prefs`

Some useful options:

`required_hits 3.0` – adjust the number of points needed to be considered spam

`score MICROSOFT_EXECUTABLE 2.50` – adjust the strength of a particular rule

SpamAssassin rules

The rules that SpamAssasin uses to detect spam fall into a few categories:

- Compare text against known patterns
- Check network-based blacklists
- AWL (Automatic White List)
- Bayesian filtering

Scores

Rules can output +ve score (the rule thinks the mail is spam) or -ve score (the rule thinks the mail is not spam) or 0 score (the rule doesn't have an opinion)
If message gets more than 5 points (by default), then the mail is spam.

This is adjustable in the required_hits parameter

Example scores for a piece of spam

```
 pts rule name              description
---- ---------------------- --------------------------------------
 4.3 MORTGAGE_RATES         BODY: Message talks about mortgage rates
 0.2 HTML_TAG_BALANCE_A     BODY: HTML has excess "a" close tags
 0.1 HTML_MESSAGE           BODY: HTML included in message
 0.1 HTML_FONTCOLOR_UNSAFE  BODY: HTML font color not in safe 6x6x6 pa
 5.4 BAYES_99               BODY: Bayesian spam probability is 99 to 1
                            [score: 1.0000]
 0.1 HTML_50_60             BODY: Message is 50% to 60% HTML
 0.1 HTML_FONTCOLOR_RED     BODY: HTML font color is red
 0.1 CLICK_BELOW            Asks you to click below
-0.3 AWL                    AWL: Auto-whitelist adjustment
```

These sum to `10.1` which is greater than the default `5.0` threshold. So this message is Spam!

Text comparison rules

These match again strings in the message text. For example, the rule `MORTGAGE_RATES` checks for the string 'Mortgage rates' anywhere in the body text, and assigns 4.3 points.

Network based blacklists

Various network based techniques

- Check if relays that have handled this message are known 'bad'. Some rules need a subscription/free contract. eg. `RCVD_IN_OSIRUSOFT_COM` and others.
- *Vipul's Razor* and *Distributed Checksum Clearinghouse* – internet-wide collaborative databases of spam. Detects if message looks similar to already reported spam. Needs extra clients installed. `RAZOR_CHECK` and `DCC_CHECK`

Automatic Whitelist (AWL)

Keeps track of average score for each sender. When mail is scanned, the average score is combined with the score for the new message. If someone sends you a message that looks spammy, but they normally send you ham, then the message is more likely to get through. Rule is `AWL`.

Bayesian filtering

Based on word analysis of *your* incoming e-mail.
Some words are common in non-spam, some words are common in spam.
Learns from the mail that *you* receive.
Words that are common in my ham mail are not necessarily common in yours, so it is difficult for spammers to guess 'good' words.

Bayesian - words

Based on word analysis of *my* incoming e-mail...

Five words that indicate spam:

```
FreeMovies.exe 1,000's Young jpg Federal
```

Five words that indicate ham (not spam):

```
Dick 2.6.0-test11 journalists silly config
```

Five words that don't mean anything either way:

```
got pretend illegal various Remember
```

Important to train the bayes filter often, otherwise it is not effective.

Training bayes: sa-learn

Need to regularly train bayes filter with *both spam and ham*.

Learn the good: `sa-learn --ham --mbox /var/spool/mail/benc`

Learn the bad: `sa-learn --spam --mbox /home/benc/mail/spam/manual-save`

Can do this in a daily/weekly cron job

What now?

```
apt-get install spamassassin
http://www.spamassassin.org
```

Questions?