Email filtering with MIMEDefang

using MIMEDefang and perl

Jan-Pieter Cornet <johnpc@xs4all.nl>
http://www.xs4all.nl/~johnpc/

XS4ALL

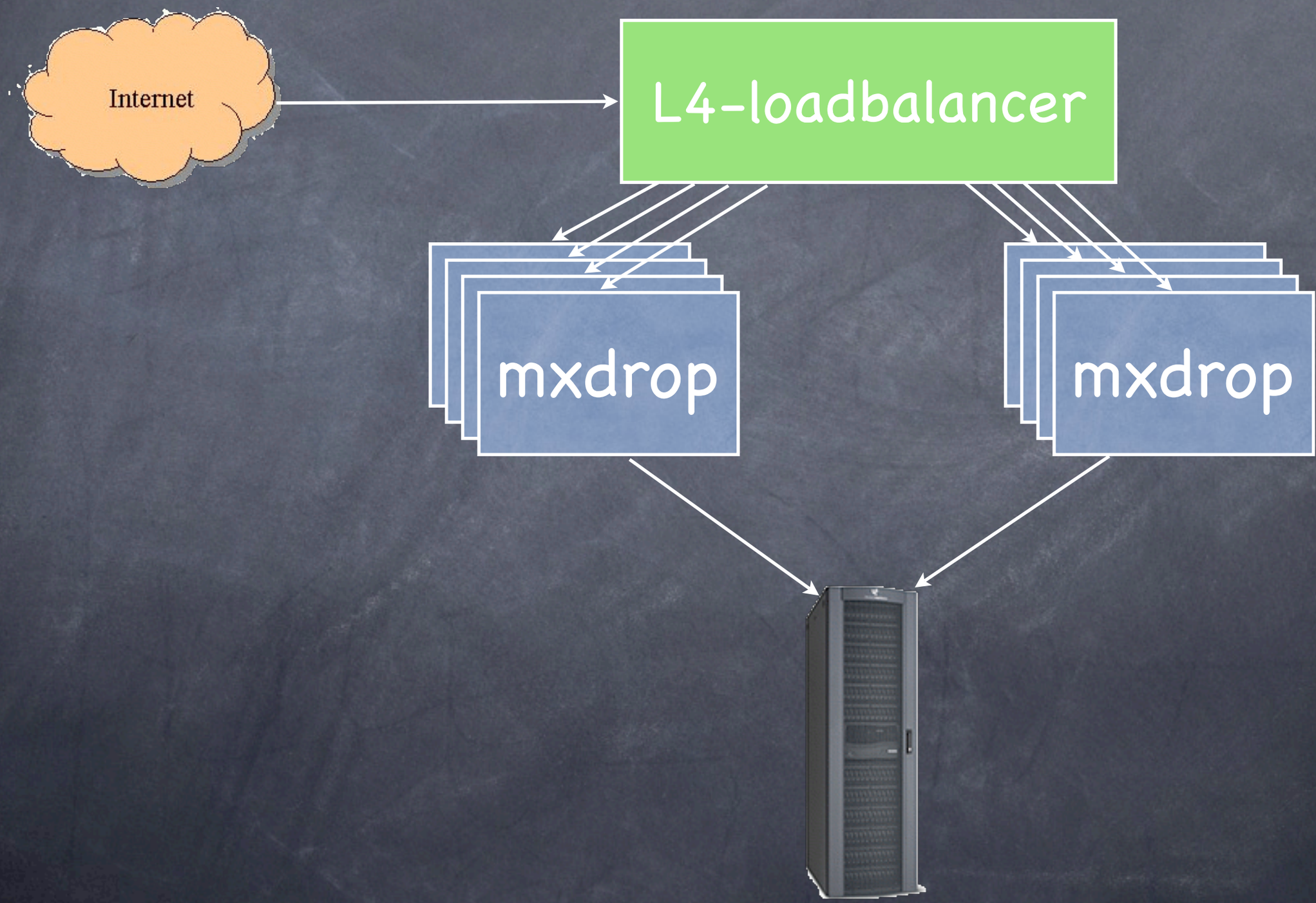
sendmail.org

MIMEDefang

Contents

- hardware
- designing/implementing software
- testing
- results
- future

hardware



Contents

- hardware
- designing/implementing software
- testing
- results
- future

new software design

- why sendmail?
- in-house experience
- configurable. It can handle our needs.
- serious competition not ready yet (then, 2003).

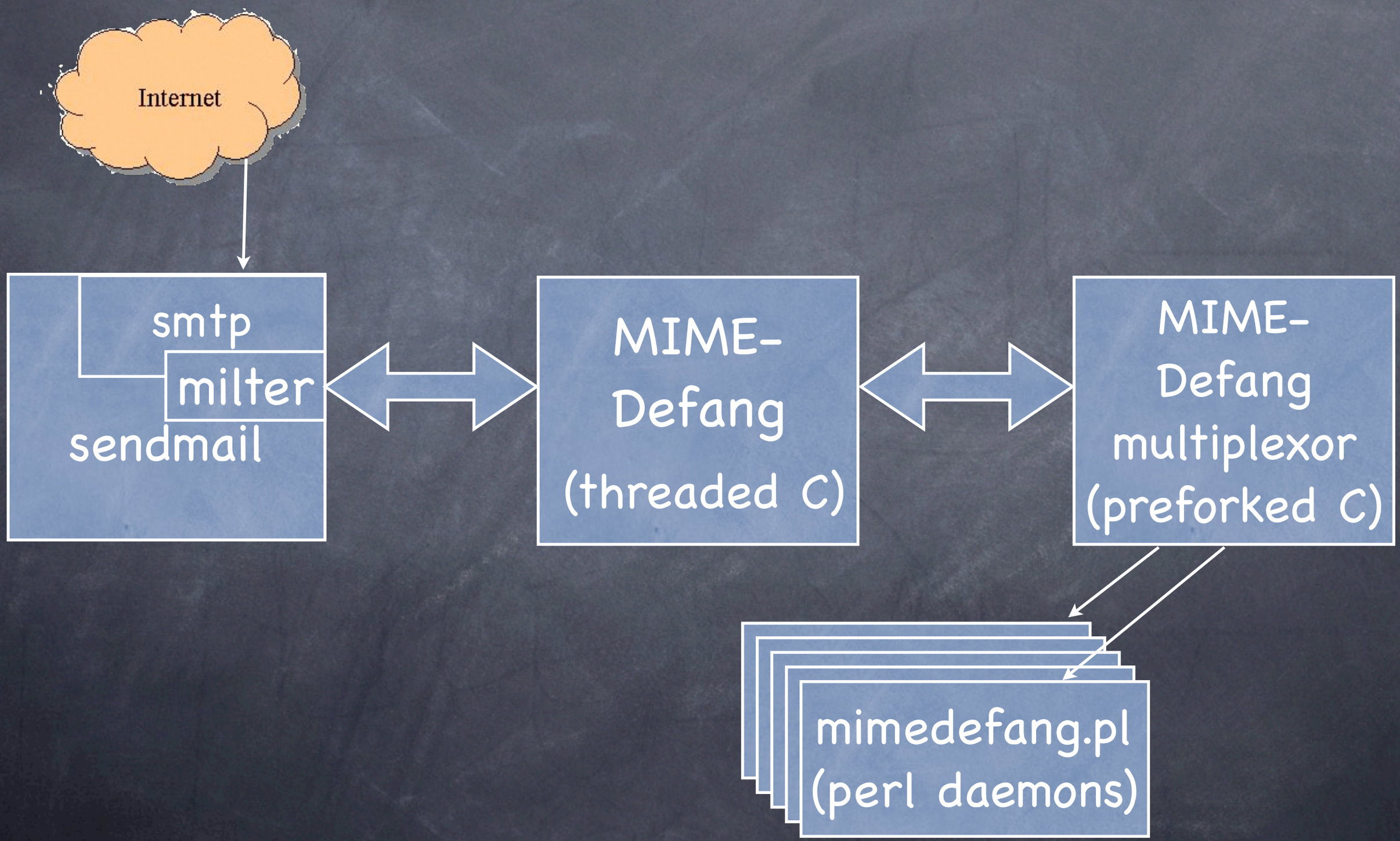
new software design

- why perl?
  - LOTS of in-house experience
  - fun! :)

new software design

- why MIMEDefang
  - allows interaction with SMTP protocol via Milter
  - filter written in perl
  - most complete/reliable perl interface to Milter.
  - Commercial product CanIt is based on MIMEDefang (made by same company)

MIMEDefang architecture



MIMEDefang architecture

- MIMEDefang calls your filter at various points in the SMTP conversation:
  connect, MAIL From, RCPT To, eom
- message can be rejected, discarded or even altered at each point.
- Can read sendmail macro values.

MIMEDefang filter

- standard MIMEDefang perl filter based on single file: suggested-minimum-filter-for-windows-clients
- hard to develop several independant filters.

MIMEDefang XS4ALL extension: modular filter

- replaces the default filter, and calls a configurable list of modules.
- Available at: http://www.xs4all.nl/~johnpc/mimedefang-modular/
- Each mimedefang filter call walks the list of modules until one module "takes action".

MIMEDefang filter fragments

- Methods that return a result code: filter_relay (on connect), filter_sender (on MAIL FROM), filter_recipient (on RCPT TO)
- Methods that call action_FOO() for results: filter, filter_end (on different MIME parts or the entire message).

MIMEDefang filter fragment example

```
sub filter_relay {
    my($ip, $name) = @_;
    if ( $name =~ /spammer.com/ ) {
        return(“REJECT”,
               “go away spammer!”);
    }
}
```

MIMEDefang modular filters, part 1

```
sub filter_relay {
    for my $c ( @coderefs ) {
        my($code, $msg) = $c->(@_);
        last if $code ne “OK”;
    }
    return ($code, $msg);
}
```

MIMEDefang filter fragment example #2

```
sub filter_begin {
    if ( $SuspiciousCharsInHeaders ) {
        action_bounce(“read RFC2822”);
    }
}
```

MIMEDefang modular filters, part 2

```
sub filter {
    for my $c ( @coderefs ) {
        $c->();
        last
         if filterchain_tookaction;
    }
}
```

modular MIMEDefang filter configuration

```
@FilterModules = qw(
    MailFilter::Virusscan
    MailFilter::BlockPartial
    MailFilter::BadFilename
    MailFilter::SpamAssassin
);
```

modular MIMEDefang filter configuration

```
@FilterModules = qw(
    MailFilter::ChangeRecipient
    MailFilter::FQ_RCPT
    MailFilter::BogusMX
    MailFilter::RelayOnMX
    MailFilter::Blacklists
    MailFilter::QuotaCheck
    MailFilter::Virusscan
    MailFilter::SuspiciousChars
    MailFilter::SpamAssassin
    MailFilter::ChangeRecipientEnd
);
```

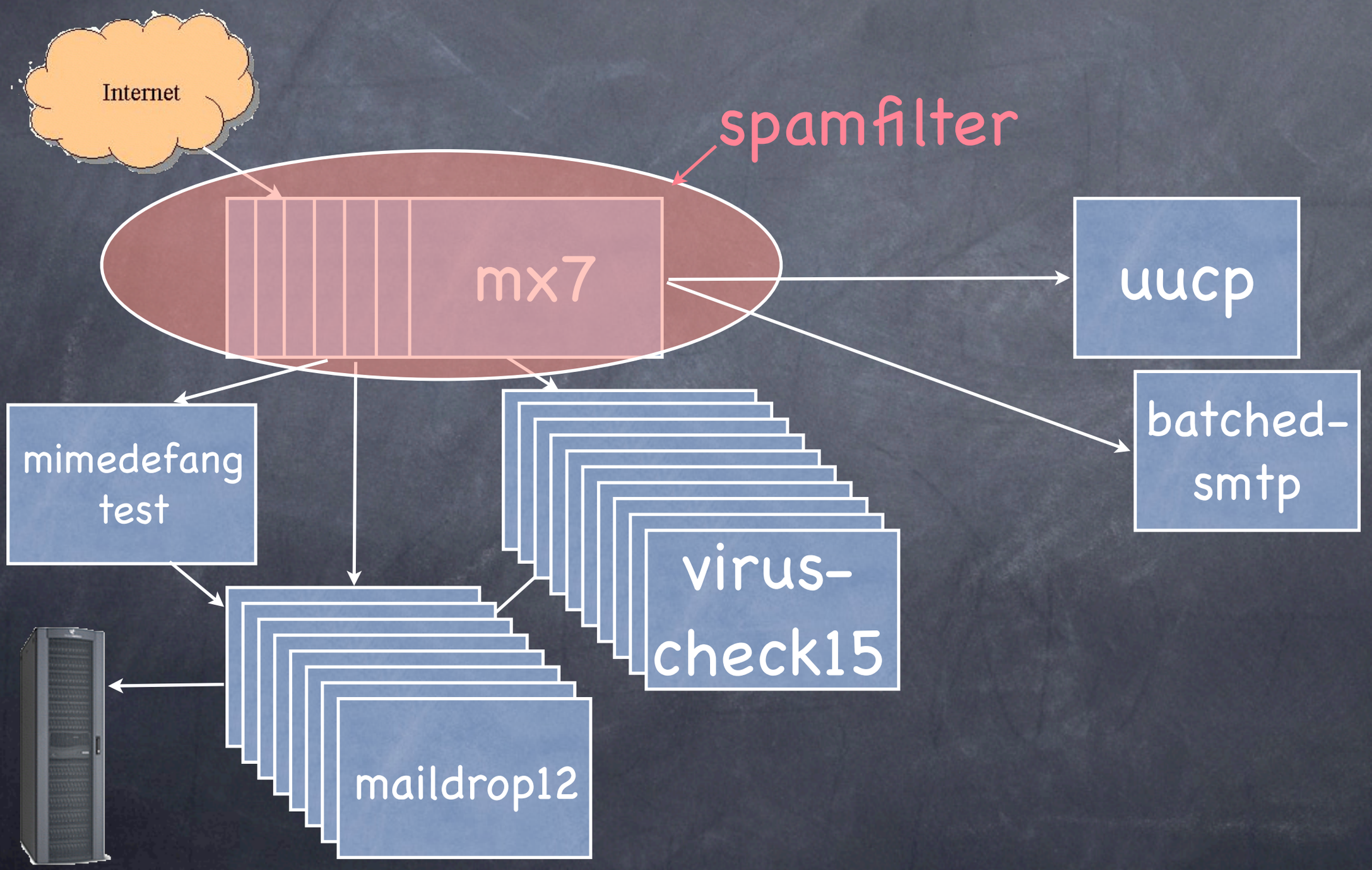
Contents

- hardware
- designing/implementing software
- testing
- results
- future

testing

- user community test: ask volunteers via usenet group xs4all.general
- use "accidental" source of pure spam as test input.
- final test: "mix in" the new software for a few minutes at a time.

previous email system: layout until 2004

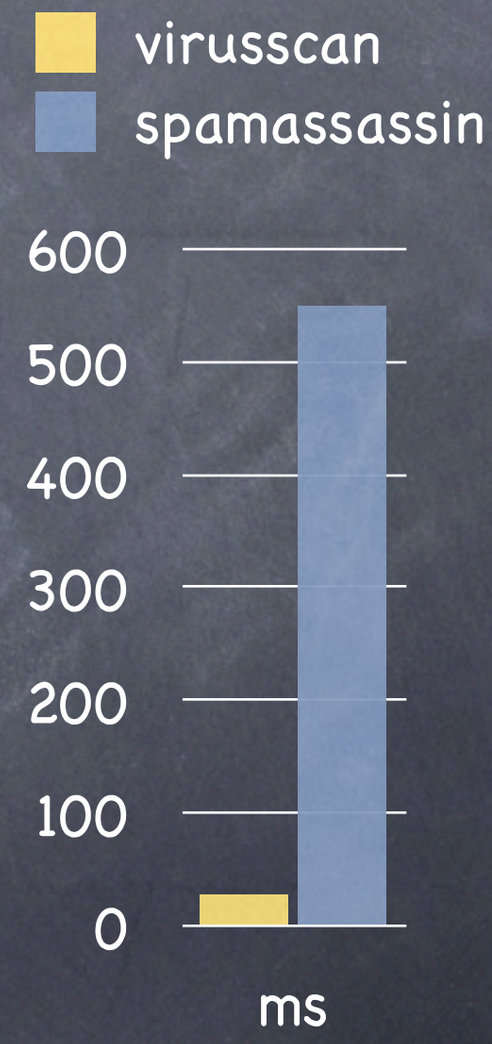


community test

- test performance of MIMEDefang
- test stability of MIMEDefang
- try spamassassin on larger audience

community test results

- performance: excellent.
  SpamAssassin is biggest CPU eater.
- stability: excellent, no unexpected crashes.
- spamassassin results are good (but test audience was mainly techies)



"live" testing with spam.

- live spam?
- We went from 7 IP numbers for mx*.xs4all.nl, to only 4 IP numbers.
- The three "abandoned" numbers were re-used for "mxdrop" clustered servers several days later.
- Lots of machines still connected to the old IP addresses, and thus to our new cluster!

"live" testing with spam

- This was a source of 99.99% spam, and 0.01% badly configured remote software.
- Since the "mxdrop" machines were running the new software, the spammers were testing our software for us. (Thanks, guys! :)
- Most email, uh, spam, was still delivered.
- This "source" of spam lasted at least a month.

"mix in" the new software

- Final test before going live.
- We added the L4 loadbalancer setup to the old mx*.xs4all.nl machines.
- The L4 loadbalancer allowed us to add one new "mxdrop" machine to the pool of old machines temporarily.

Contents

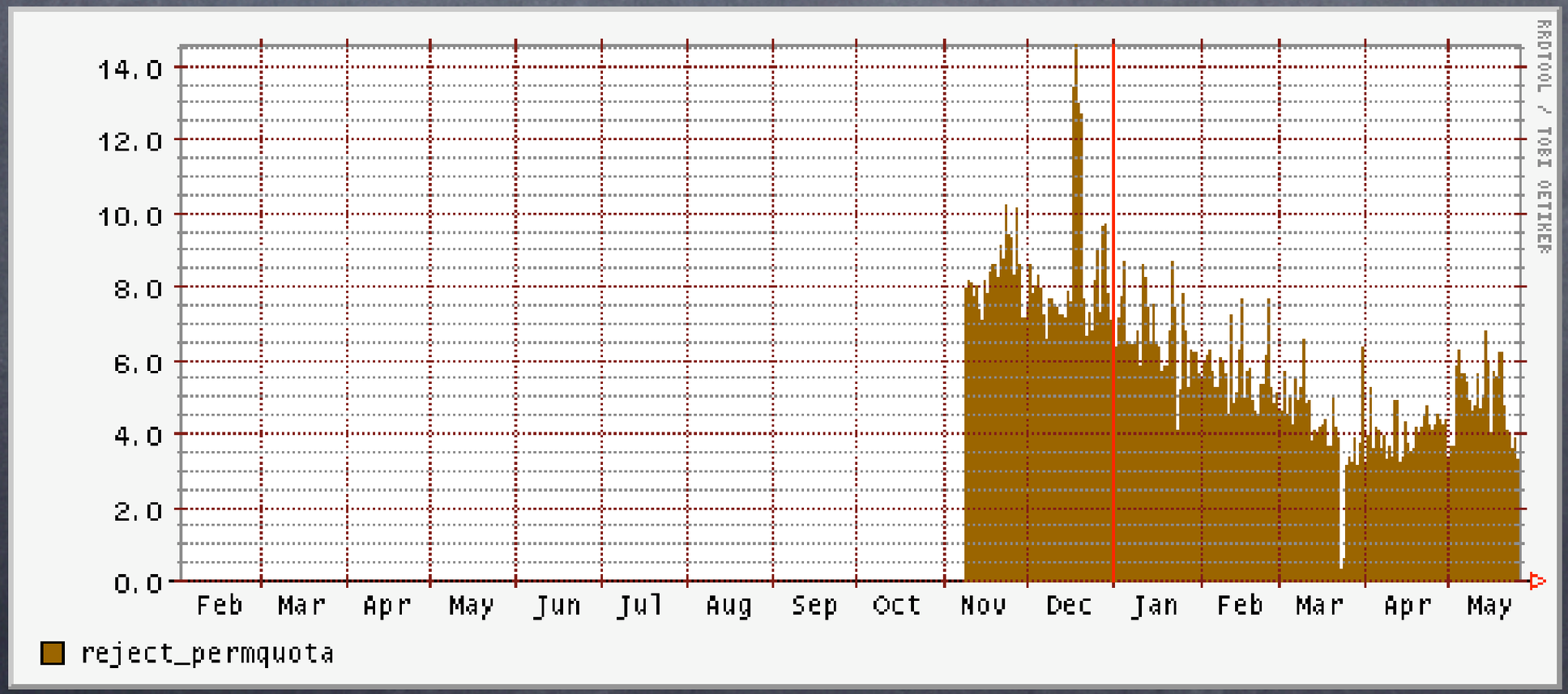
- hardware
- designing/implementing software
- testing
- results
- future

Results

- Introduction went very smooth due to rigorous testing.
- Some spam False Positives for some users. Spamassassin is no silver bullet.

Results

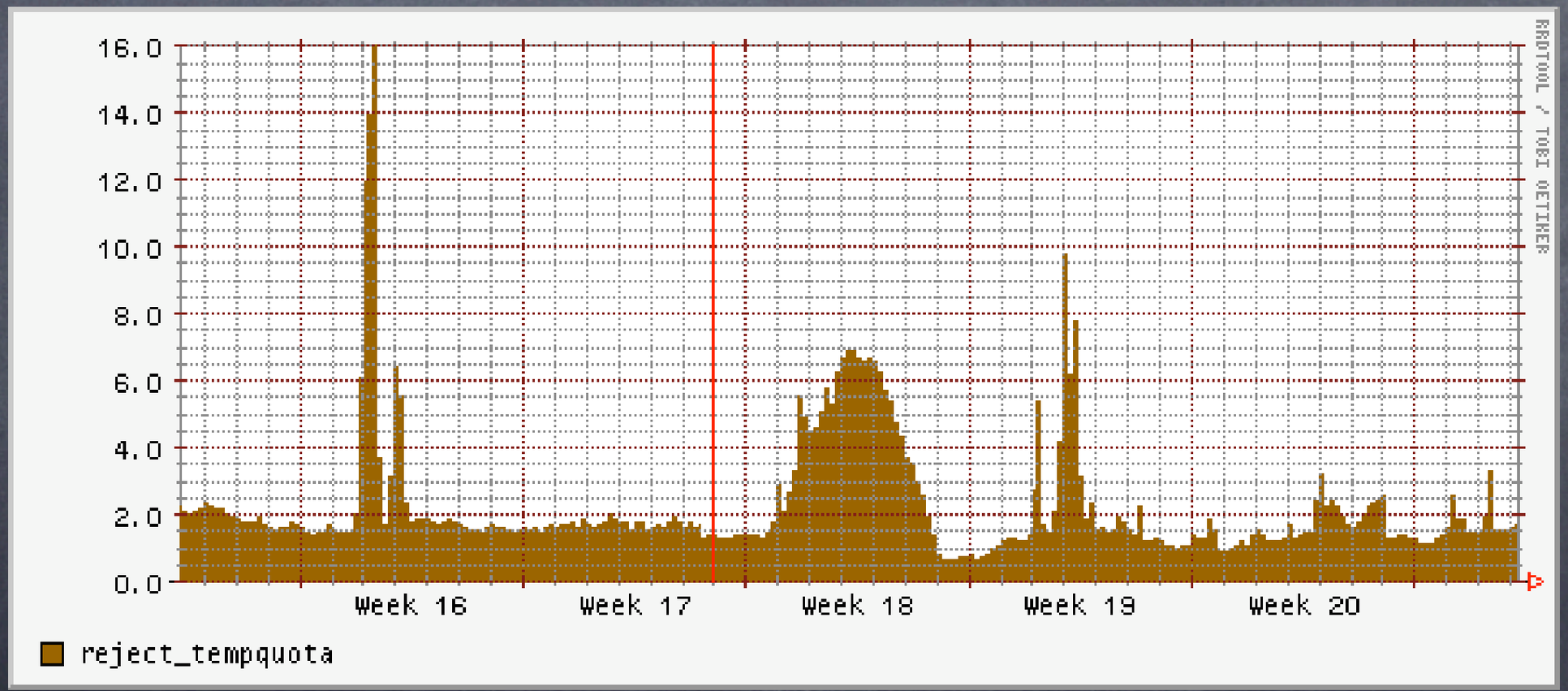
- Serious reduction in bounces



Rejects for quota exceeded per second

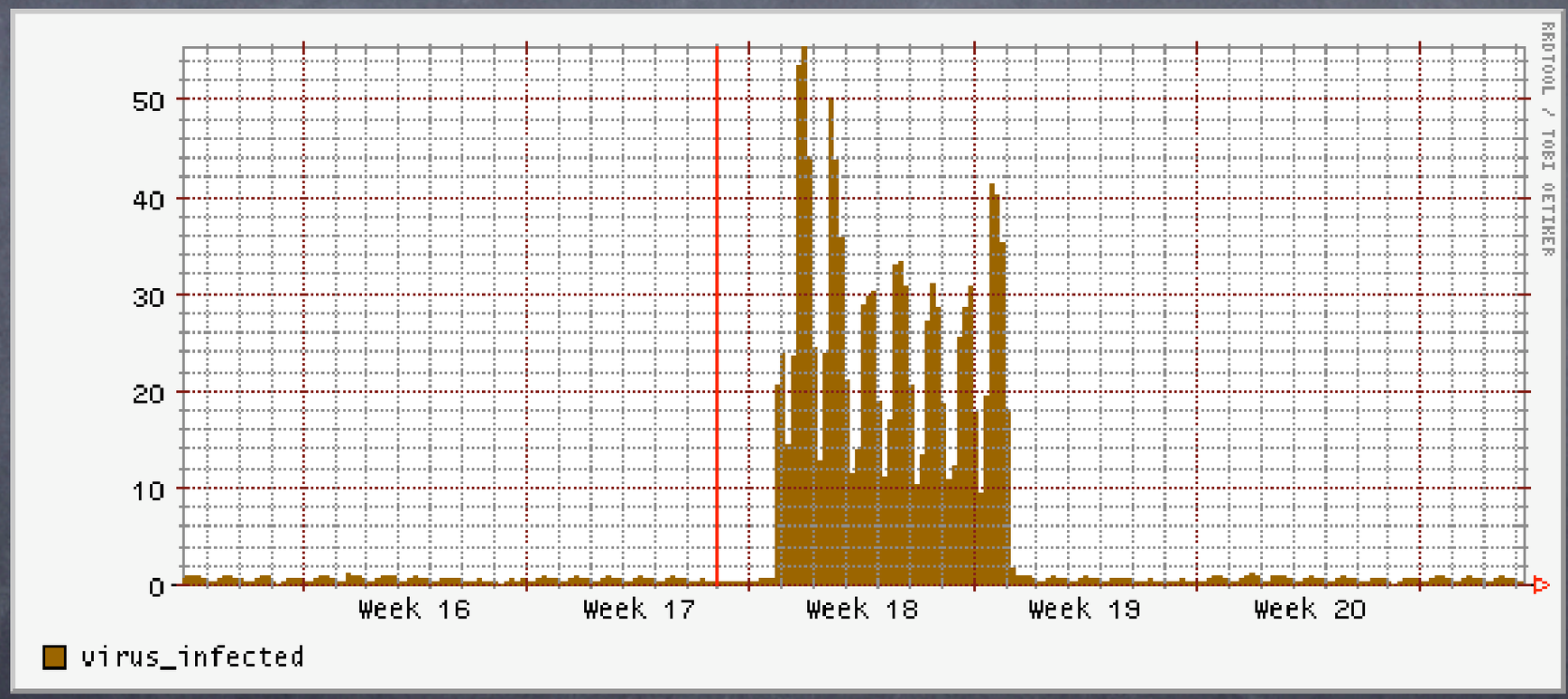
Results

- Serious reduction in bounces



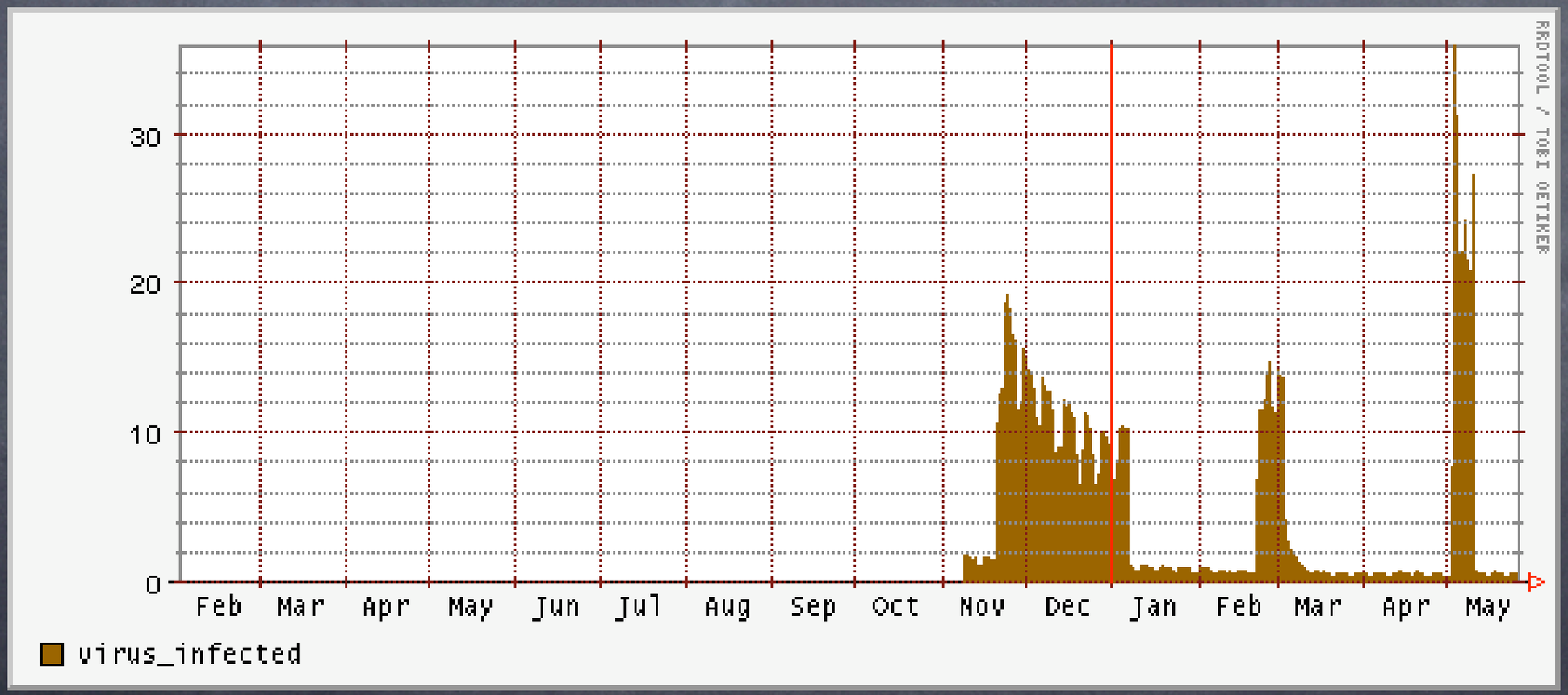
Tempfails for quota exceeded per second

results: nice pictures



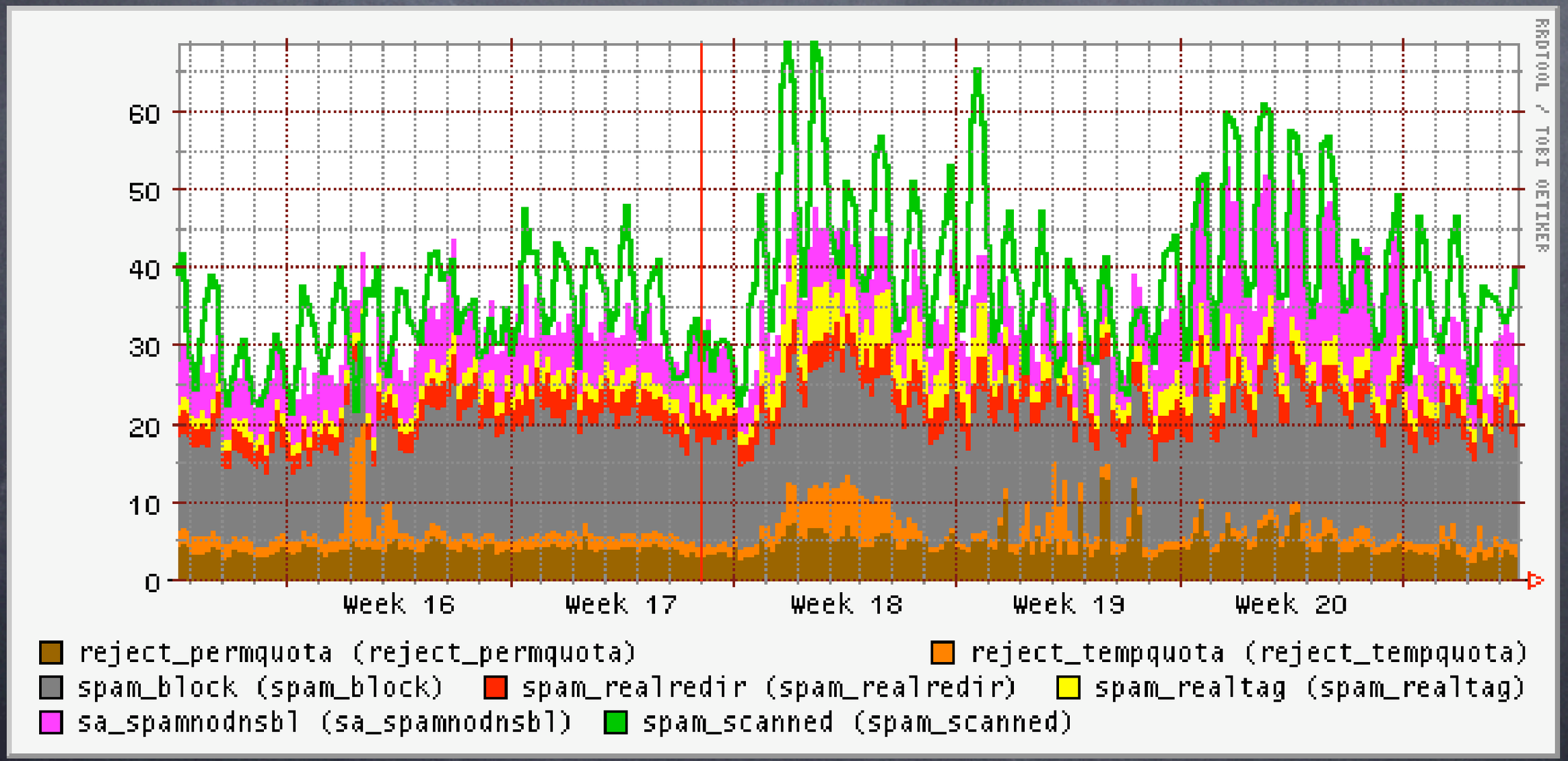
Viruses per second detected, around may

results: nice pictures

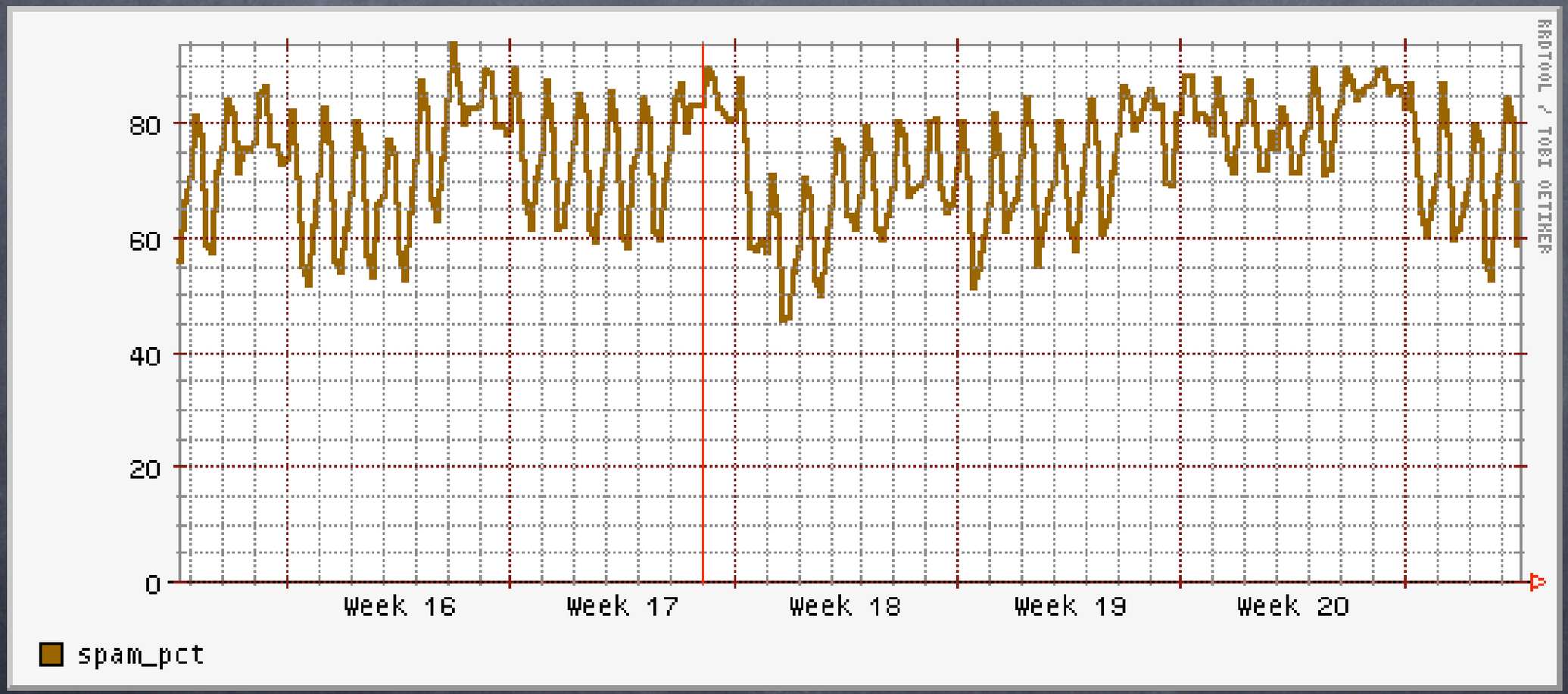


viruses per second, last year.

results: spam detect overview



results: spam percentages



Contents

- hardware
- designing/implementing software
- testing
- results
- future

Future developments

- User defined black-whitelists
- greylisting
- Bayes, system-wide and/or per-user?
- Blocking email based on spamassassin score.

blocking mail on spamassassin score

ESMTP conversation example:

<<< 220 ESMTP
>>> EHLO spammer.com
<<< 250 OK
>>> MAIL From:<badguy@spammer.com>
<<< 250 OK
>>> RCPT To:<recip1@example.com>
<<< 250 Recipient OK
>>> RCPT To:<recip2@example.com>
<<< 250 Recipient OK
>>> DATA
<<< 354 Send data, end with .
[email]
.
>>> ??? (recip1 wants mail, recip2 does not)

blocking mail on spamassassin score

- (E)SMTP protocol does not allow that!
- Possible solution: junk mail instead of rejecting.
  - Mail will get lost in case of false positive

blocking mail on spamassassin score

- Possible solution: only accept 1 recipient
  - legitimate mailinglists will get very slow, lots of email delays.
- Solution unknown, probably combination of methods.

?