QR:

Mark Martinec

Institut “Jožef Stefan”, Slovenia

amavis.org/Z1
(pdf)

amavis (amavisd-new)

Configuration and Management 2.7.0 update

Agenda

what it is, quick project history
some tuning hints
updates on a 2.7.0 release
pre-queue (proxy) filtering
configuration: policy banks, lookups, cc
monitoring

Amavis - what is it?

interface between MTA and

virus checkers and/or spam checkers

like *spamd* for SA, but speaks standard SMTP

checks for banned content and header syntax

quarantining/archiving

DKIM: signs and verifies signatures

monitoring: SNMP, SQL log, nanny

why is it popular?

reliable:

- checks status of every operation, internal asserts
- in case of a failure mail stays with MTA, not lost

adheres to standards (SMTP, MIME, DSN, ...)

reasonably fast, feature-rich

maintainable: logging for troubleshooting

security: perl, taint checks, can run *chroot*-ed

mature: 9+ years of steady development

OSS: GPL 2 license (+ BSD licensed tools)

AMaViS – A Mail Virus Scanner

shell program:
1997 Mogens Kjaer, Juergen Quade
1998 .. 2000 AMaViS Christian Bricart, Rainer Link, Chris Mason
(amavis.org)

Perl program:
2000-01 Amavis-perl Chris Mason
2003-03 Amavis-0.3.12 Lars Hecking

Perl daemon:
2001-01 .. 2003-03 amavisd Geoff Winkless, Lars Hecking

Perl, re-design
2002-03 .. 2003-03 amavis-ng Hilko Bengen

Amavis releases and events ...

2002-03-29 amavisd-new, pre-forked, Net::Server

2004-07-01 2.0policy banks, IPv6 address formats

2005-04-24 2.3.0 @decoders, per-recip banning rules

2006-04-02 2.4.0 DSN in SMTP, %*_by_ccat

2007-04-23 2.5.0 blocking cc, new SMTP client

2008-01-13 SpamAssassin Project Mgmt Committee

2008-04-23 2.6.0 DKIM, bounce killer, TLS

2009-06-25 2.6.4 SNMP monitoring

… Amavis releases and events

2010-04-25	2.7.0-pre4
2011-02-03	2.7.0-pre14
2011-03-07	moved ML from SF to amavis.org (hosted by Patrick Ben Koetter and Ralf Hildebrandt)
2011-04-07	2.6.5
2011-05-19	2.6.6
2011-05-18	2.7.0-rc1

9+ years of steady amavisd-new development

Did it grow too large?

29.000 lines of Perl code (with comments)
modules, loaded only what is needed
half of memory footprint is SpamAssassin
memory is not a limitation to mail size
grows linearly, hardware exponentially

Is it slow?

written in Perl
perform operations on large chunks of data
avoid line-by-line processing
avoid copying data

critical code paths are well optimized
sanity limits and suitable data struct & alg
the slow part is SpamAssassin, if enabled

SMTP read speedup example

2.7.0

by a factor of 3.9 (non-TLS) – 32.3 MiB/s
by a factor of 11 for TLS

The bottleneck was line-by-line reading due to SMTP dot-destuffing. Code reworked to operate on entire buffers, dealing with dot-stuffing intricacies when crossing buffer boundaries.

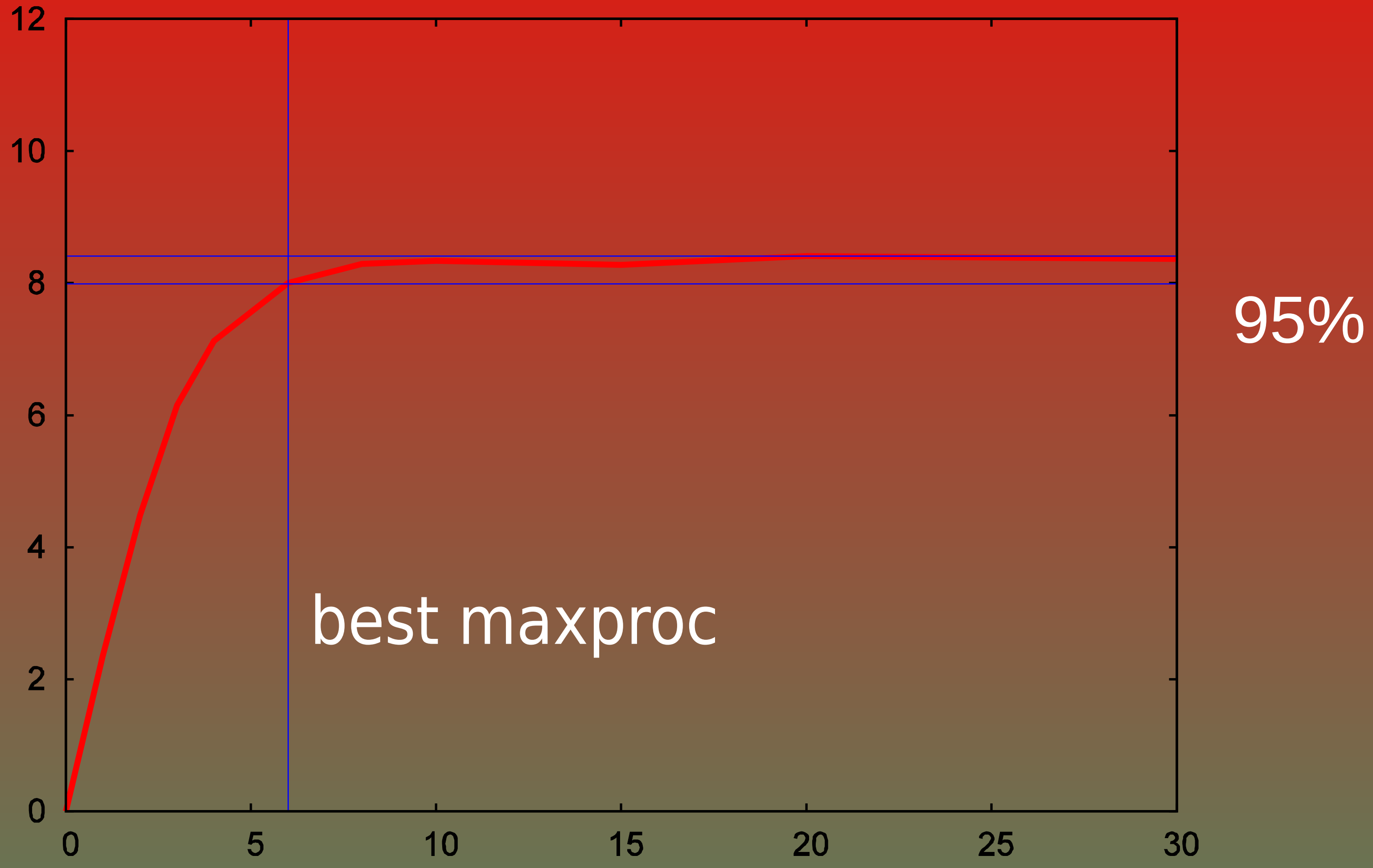
Network latency in SA a problem?

DNS black and white lists (RBL)
DCC, Razor, Pyzor network services

The bottleneck in SpamAssassin is CPU, idle wait times are compensated by running more processes, the only cost is memory.

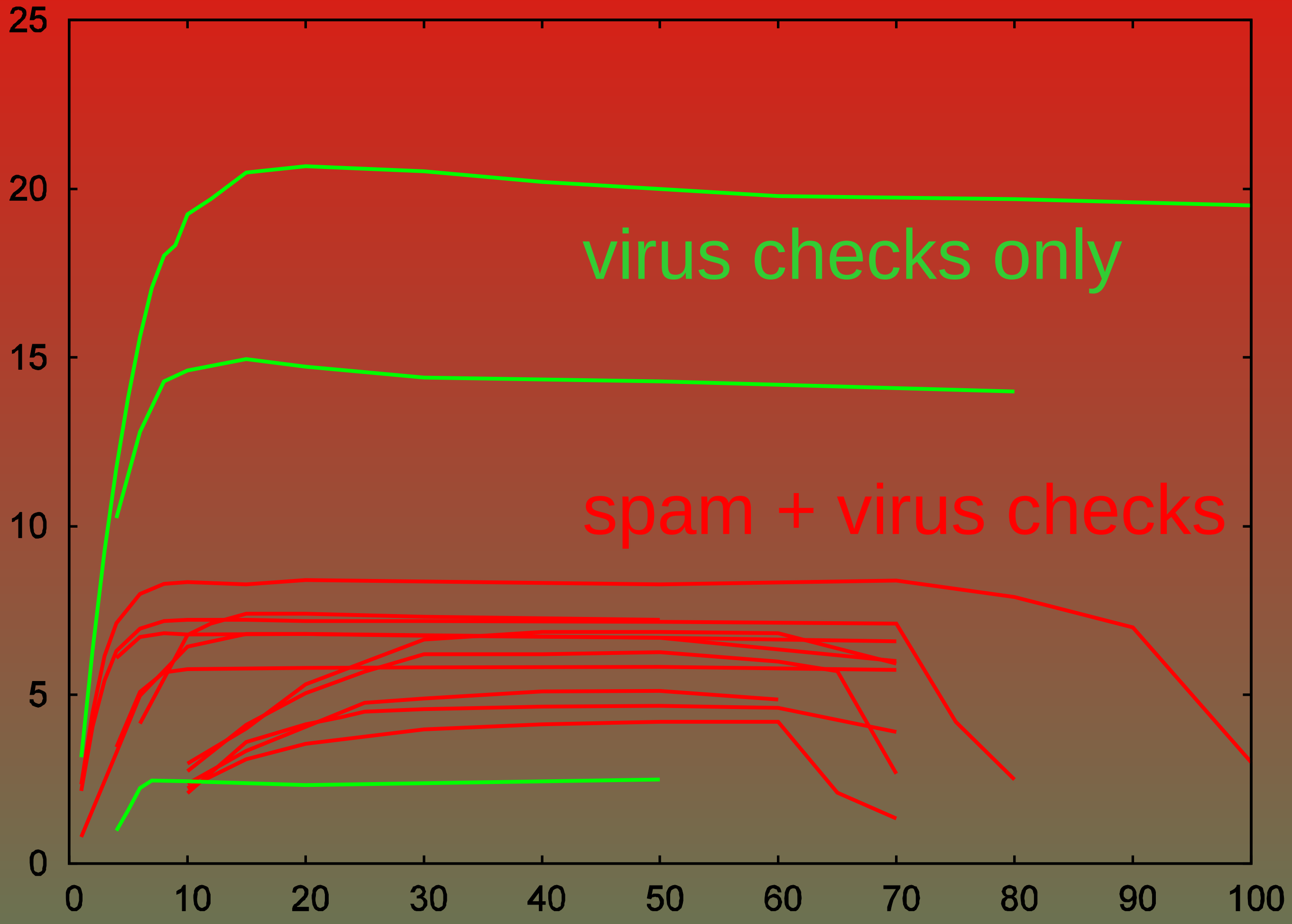
Performance – parallelism

msgs/s *vs.* maxproc



Performance: SpamAssassin

msgs/s *vs.* maxproc



Some tuning hints

choose number of processes to match CPU capacity

avoid slow command-line virus scanners

Linux syslogd: disable sync on MTA/amavisd logs

turn on *$quarantine_subdir_levels* = 1

examine timing reports at $log_level = 2

observe nanny, $nanny_details_level = 2

separate disks for MTA spool and amavisd tmp

separate MTA and amavisd hosts

split load through multiple MX records

SpamAssassin tuning ideas

use SQL for r/w Bayes and AWL databases
alternatively: r/o cdb, updated offline
compiled rules: sa-compile
limit mail size, truncate since 2.6.3 / SA 3.3.0
avoid slow regexp rules (HitFrequencies.pm)
reduce time limits on rbl, razor, pyzor
use local caching DNS server, mirrored RBL
examine SA timing reports at log level 2

New in 2.7.0 – at a glance

2.7.0

improved as a pre-queue proxy content filter
per-recipient SpamAssassin Bayes & user prefs
external DKIM signer
next hop failover
new macros, more informative logging
SMTP/LMTP receive speedup
Sophos-SSSP, Avira SAVAPI, clamd streaming
...

pre-queue filtering

2.7.0

Benefits:

can reject original SMTP session
(eliminates bounce backscatter to 3rd parties)
preferred to quarantine & discard or tag & deliver

Drawbacks:

tighter timing constraints
no. of content filters is more tightly coupled
to a number of concurrent SMTP sessions
must cope with peaks, instead of averages

pre-queue filter requirements

2.7.0

real-time nature
no. of filters = no. of sessions (almost)
SMTP end-of-data timeout at a mercy of client
minimize disruption caused by a filter restart

Stricter time limits

2.7.0

reworked sub-task time limiting
needs SpamAssassin 3.3.0 or later: *master_deadline*, results despite aborted tests

$child_timeout = 45   (good starting point)
the longest time most SMTP clients are willing to wait, less than *smtpd_proxy_timeout* (100 s)

Warm/flying reload

2.7.0

amavisd reload signals a HUP to a daemon

daemon clears FD_CLOEXEC on socket fd
and stores socket info to BOUND_SOCKETS

daemon restarts itself by *exec*(), passing
open socket descriptors to next incarnation

new instance attaches sockets to inherited fd

pre-queue filtering – Postfix

new option – since Postfix 2.7.0 (20091101) :

```
smtpd_proxy_options = speed_adjust
```

Postfix SMTP server receives an entire message before connecting to a before-queue (proxy) content filter

decouples slow SMTP clients from content filters

pre-queue filtering – Postfix

postscreen(8) is a new Postfix 2.8 feature, reducing the load on pre-queue content filters:

```
smtp    inet  n       -       n       -       1       postscreen

smtpd  pass  -       -       n       -      150   smtpd
    -o smtpd_proxy_filter=inet:[127.0.0.1]:10010
    -o smtpd_proxy_options=speed_adjust
```

External DKIM signer

2.7.0

amavisd calls Mail::DKIM to pre-process a message for signing

sends a prepared DKIM mail digest to an external signing daemon, along with a signing domain name and a selector (d, s)

receives a signed digest (p) and inserts a signature header field into a message

private keys can be kept hidden from amavisd

Penpals

old but often neglected feature
to reduce false positives

Our.Alice@here –> Some.Bob@example
Some.Bob@example –> Our.Alice@here

also: Message-ID <–> In-Reply-To, References

Bounce killer

another old but often neglected feature
to reduce foreign backscatter

If a message looks like a bounce and contains a header section of original mail, check if that came from our server. If decisively not, drop it.

Configuration – agenda

general
mail flow direction
logging, syslog
interfacing: input, output, milter
policy banks
lookups
content categories

Configuration – general

all config settings: amavisd.conf-default

directories, hostname, ...
user (uid)
destination, source
$max_servers
$nanny_details_level = 2; # verbosity: 0, 1, 2

Configuration – mail flow direction

origin: @mynetworks, $originating
destination: @local_domains_maps

originating (property of a message)	local-recipient (property of a recipient)	
0	0	... open relay
0	1	... inbound
1	0	... outbound
1	1	... internal-to-internal

Configuration – destination

list all your domains in @local_domains_maps
(local, virtual aliases, virtual mailbox, relay)

affects:

inserting header fields X-Spam-*,
X-Quarantine-ID, X-Amavis-OS-Fingerprint, ...

adding address extension (*plus addressing*)

recipient notifications

pen pals

defanging

statistics / SNMP

Configuration – origin (source)

origin: @mynetworks, $originating

affects:
DKIM signing
inserting disclaimers
bounce killer
pen pals
MYUSERS policy bank
statistics / SNMP

Configuration – origin (source)

setting the $originating flag:
implicitly: @mynetworks

explicitly, typically through a policy bank:

```
$inet_socket_port = [10024, 10026];
$interface_policy{'10026'} = 'ORIG';

$policy_bank{‘ORIG'} = {
      originating => 1,
};
```

Configuration – flow direction

2.7.0

2.7.0 new SQL fields:

msgs . originating
msgrcpt . is_local

see message flow direction from SQL log

Mail direction in SpamAssassin

2.7.0

internal_networks
trusted_networks
msa_networks

2.7.0: passes a value of the $originating flag to SpamAssassin 3.4.0, treated like msa_networks

Configuration – logging

SA	amavisd	syslog
	-3	LOG_CRIT
	-2	LOG_ERR
error	-1	LOG_WARNING
warn	0	LOG_NOTICE
info	1	LOG_INFO
	2	LOG_INFO
dbg	3	LOG_DEBUG
	4	LOG_DEBUG
	5	LOG_DEBUG

Configuration – syslog

```
$do_syslog = 1;  (pre-2.7.0:
$DO_SYSLOG)

$syslog_facility = 'user';

$log_level = 2;         # verbosity 0..5
```

Configuration – /etc/syslog.conf

```
user.err; mail.crit; ...   /var/log/messages
user.notice           /var/log/amavisd.log
user.info        /var/log/amavisd-info.log
user.debug            /var/log/amavisd-debug.log
```

Prepend ' – ' to a filename on Linux
to disable sync!

Configuration – log template

```
$log_templ = <<'EOD';
[?%#D|#|Passed #
[...]
[? %q ||, quarantine: %q]#
[? %Q ||, Queue-ID: %Q]#
[? %m ||, Message-ID: %m]#
[? %r ||, Resent-Message-ID: %r]#
, mail_id: %i#
, Hits: [:SCORE]#
, size: %z#
[...]
EOD
```

Configuration – log template

$log_templ
$log_recip_templ

macros: README.customize

From, Subject, Message-Id, User-Agent,
size, Hits, Tests, banning, DKIM id, ...

Configuration – log template

2.7.0

two pre-defined log templates:

```
$log_templ = $log_short_templ;   # default
$log_templ = $log_verbose_templ;
```

Configuration – log template

2.1.0

new macros:

client_helo, client_port, actions_performed, mime2utf8, rusage, ...

Configuration – log template

2.7.0

new macro: actions_performed

action:
Accepted, Relayed, RelayedTagged, Discarded, Rejected, Bounced, NoBounce, TempFailed

flow direction:
Inbound, Internal, Outbound, OpenRelay

Configuration – log template

2.7.0

new macro: actions_performed

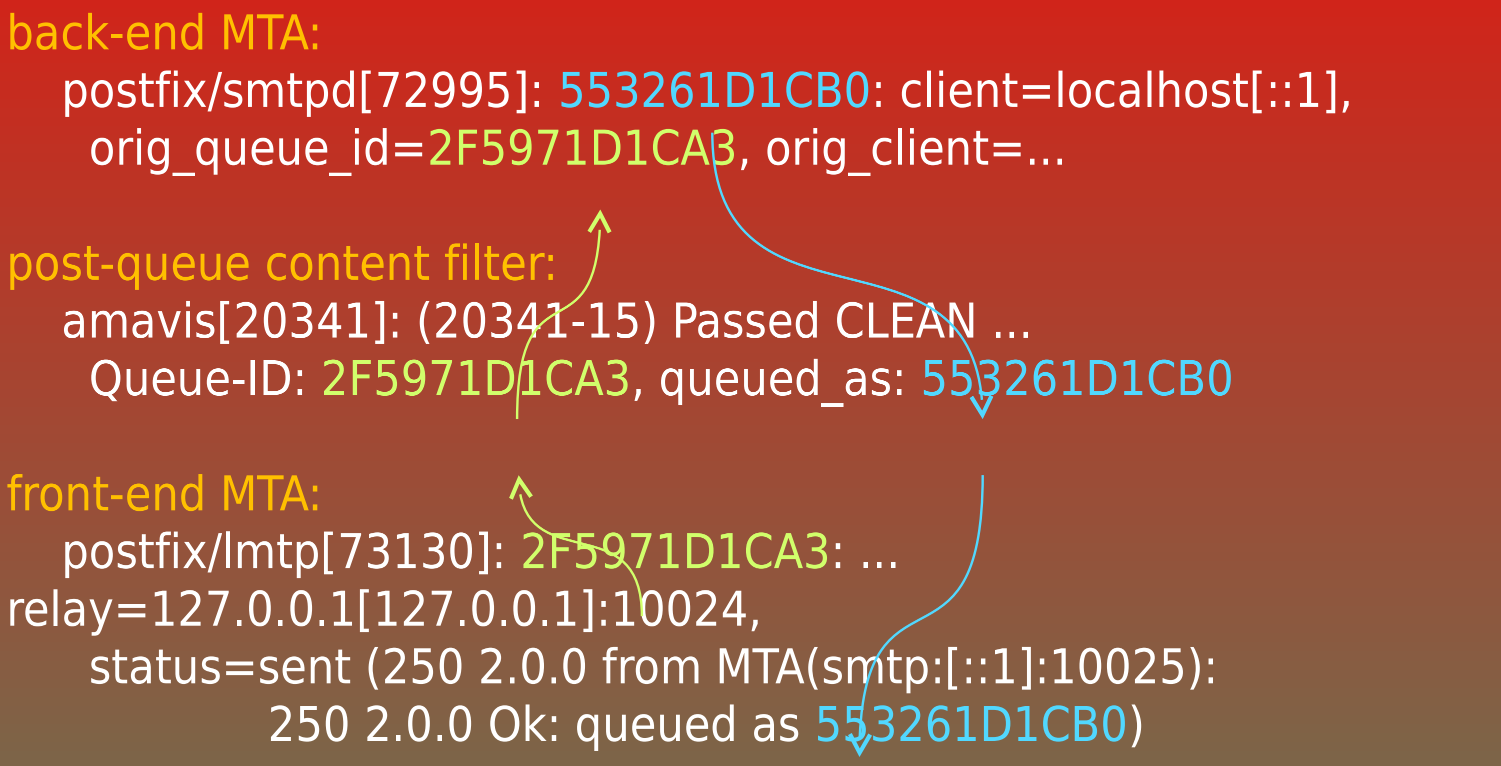
examples:

```
Passed CLEAN {RelayedOutbound}, ...
Passed CLEAN {RelayedInbound}, ...
Passed CLEAN {RelayedInternal,RelayedOutbound},
Passed SPAMMY {RelayedTaggedInbound}, ...
Blocked SPAM {RejectedInbound,Quarantined}, ...
Blocked INFECTED (Mal/BredoZp-B) {DiscardedInbound,Quarantined}, ...
```

Configuration – logging

2.7.0

2.7.0: passing queue-id end-to-end (XFORWARD IDENT Postfix 2.8.0)



back-end MTA:

```
    postfix/smtpd[72995]: 553261D1CB0: client=localhost[::1],
      orig_queue_id=2F5971D1CA3, orig_client=...
```

post-queue content filter:

```
    amavis[20341]: (20341-15) Passed CLEAN ...
      Queue-ID: 2F5971D1CA3, queued_as: 553261D1CB0
```

front-end MTA:

```
    postfix/lmtp[73130]: 2F5971D1CA3: ...
relay=127.0.0.1[127.0.0.1]:10024,
      status=sent (250 2.0.0 from MTA(smtp:[::1]:10025):
                  250 2.0.0 Ok: queued as 553261D1CB0)
```

Configuration – input interface

SMTP or LMTP or AM.PDP or AM.CL on input

```
$inet_socket_port = [10024, 10026, 10027];
    # TCP port numbers

@inet_acl = qw( 127.0.0.0/8 [::1] 192.168.1.1 );
    # access control

$inet_socket_bind = '127.0.0.1';
    # restrict to one interface

$unix_socketname = '/var/amavis/amavisd.sock';
    # quarantine release or milter
```

Configuration – input interface

2.7.0

2.7.0: a list @listen_sockets represents a unified configuration of listening sockets.

Combined: $unix_socketname, $inet_socket_bind, $inet_socket_port

```
@listen_sockets = (10024, '*:10026',
  '127.0.0.1:9998', '[::1]:9998', '192.0.2.0:10028',
  "$helpers_home/amavisd.sock" )
```

Configuration – output

SMTP or LMTP or pipe on output

```
$forward_method = 'smtp:[127.0.0.1]:10025';
$notify_method     = 'smtp:[127.0.0.1]:10025';

$forward_method = 'smtp:*:*';
$notify_method     = 'smtp:*:10587';
```

1st asterisk use SMTP client peer address
2nd asterisk incoming SMTP/LMTP session port no. plus one

```
$virus_quarantine_method,
$spam_quarantine_method, ...
```

Configuration – output

2.7.0

2.7.0: Failover or simpleminded load balancing
in SMTP and LMTP client – a list of next-hop destinations

Typical usage in $forward_method, $notify_method, $resend_method, $release_method, $requeue_method

```
$forward_method =
  [ 'smtp:[::1]:10025', 'smtp:[127.0.0.1]:10025', 'smtp:*:10025' ];

$notify_method =
  [ 'smtp:*:*', 'smtp:192.0.2.10:10025' ];
```

Configuration – output

2.7.0

by recipient:
    @forward_method_maps

by contents category
    %forward_method_maps_by_ccat

custom hook:
    $msginfo->delivery_method(...)

Configuration – milter setup

```
$unix_socketname =
   '/var/amavis/amavisd.sock';

$interface_policy{'SOCK'} = 'SOMEMILTER';

$policy_bank{'SOMEMILTER'} = {
   protocol => 'AM.PDP',
};

$forward_method = undef;
$notify_method = 'pipe: ... sendmail -Ac -i -odd
   -f ${sender} -- ${recipient}';
```

Policy banks

one global, currently in effect,
set of configuration variables

several replacement sets (groups)
of configuration variables,
prepared in advance and on stand-by,
quickly loadable

affects message as a whole (not per-recipient)

Policy banks

RED

```
$a = ”red”;
$b = 4;
$c = ”ABC”;
```

GREEN

```
$a = ”green”;
```

BLUE

```
$a = ”blue”;
$b = 99;
@d = (88);
```

current

```
$a = ”black”;
$b = 2;
$c = undef;
@d = (1, 2, 3);
```

Policy banks

RED

```
$a = ”red”;
$b = 4;
$c = ”ABC”;
```

GREEN

```
$a = ”green”;
```

BLUE

```
$a = ”blue”;
$b = 99;
@d = (88);
```

current

```
$a = ”blue”;
$b = 99;
$c = undef;
@d = (88);
```

Policy banks

RED

```
$a = ”red”;
$b = 4;
$c = ”ABC”;
```

GREEN

```
$a = ”green”;
```

BLUE

```
$a = ”blue”;
$b = 99;
@d = (88);
```

current

```
$a = ”green”;
$b = 99;
$c = undef;
@d = (88);
```

Policy banks

RED

```
$a = ”red”;
$b = 4;
$c = ”ABC”;
```

GREEN

```
$a = ”green”;
```

BLUE

```
$a = ”blue”;
$b = 99;
@d = (88);
```

current

```
$a = ”red”;
$b = 4;
$c = ”ABC”;
@d = (88);
```

Policy banks – Perl syntax

normal settings

variables, assignments

```
    $a = ”xyz”;
    @m = (1, 2, ”xyz”);
    %h = (a => 1, b => 2);
```

separator: semicolon
list: (1, 2, 3)
hash: (a => 1, b => 2)

within a policy bank

key / value pairs

```
    a => ”xyz”,
    m => [1, 2, ”xyz”],
    h => { a => 1, b => 2 },
```

separator: comma
list reference: [1, 2, 3]
hash ref: { a => 1, b => 2 }

Policy banks – examples

```
$policy_bank{'NOVIRUSCHECK'} = {
      bypass_decode_parts => 1,
      bypass_virus_checks_maps => [1],
      virus_lovers_maps => [1],
};

$policy_bank{'AM.PDP-SOCK'} = {
      protocol => 'AM.PDP',
      auth_required_release => 0,
      syslog_ident => 'amavis-release',
};
```

Policy banks – example

```
$policy_bank{'ALT'} = {
  originating    => 1,
  log_level          => 2,
  forward_method   => 'smtp:*:*',
  local_client_bind_address  => '193.2.4.6',
  localhost_name  => 'extra.example.com',
  final_spam_destiny   => D_PASS,
  spam_kill_level_maps  => 6.72,
};
```

Policy banks – activating by port no.

```
$inet_socket_port =
    [10024, 10026, 10028, 10030, 9998];

$interface_policy{'10026'} = 'ORIGINATING';
$interface_policy{'10028'} = 'NOCHECKS';
$interface_policy{'10030'} = 'CUSTOMER';
$interface_policy{'9998'} = 'AM.PDP-INET';
$interface_policy{'SOCK'} = 'AM.PDP-SOCK';
```

Policy banks – by client's IP address

```
my(@some_nets) = qw( 10.0.1.0/24 10.0.2.0/24 );

@client_ipaddr_policy = (
    [ '0.0.0.0/8',  '127.0.0.1/8',  '[::]',  '[::1]' ]
                        =>  'LOCALHOST',
    [qw( !172.16.1.0/24 172.16.0.0/12 192.168.0.0/16 )]
                        =>  'MYPRIVATENETS',
    [qw( 192.0.2.0/25 192.0.2.129 192.0.2.130 )]
                        =>  'PARTNERS',
    \@some_nets             =>  'OTHER',
    \@mynetworks            =>  'MYNETS',
);
```

Policy banks – implicitly MYNETS

```
@mynetworks = qw(
      0.0.0.0/8 127.0.0.0/8 [::1]
      10.0.0.0/8 172.16.0.0/12 192.168.0.0/16
      192.0.2.0/24 [2001:db8::/32]
);
```

implicitly loads policy bank MYNETS
if it exists

Policy banks – by DKIM signature

```
@author_to_policy_bank_maps = (
{     'uni-bremen.de'   => 'WHITELIST',
      'tu-graz.ac.at'      => 'WHITELIST',
      '.ebay.com'   => 'WHITELIST',
      '.paypal.com'      => 'WHITELIST',
      'amazon.com'      => 'WHITELIST',
      'cern.ch'       => 'SPECIAL',
      '.linkedin.com'     => 'MILD_WHITELIST',
      'dailyhoroscope@astrology.com'
                       => 'MILD_WHITELIST',
} );
```

Policy banks – by a virus name

2.7.0

```
@virus_name_to_policy_bank_maps = (
  new_RE(   # a regexp-type lookup
    [ qr'^(W32/MyDoom|W32/Netsky|Mal/BredoZp)'
            => 'REAL_INFECTION, MASS_VIRUS' ],
    [ qr'\bEICAR\b'i
            => 'EICAR_TEST' ],
  ),
);

$policy_bank{'MASS_VIRUS'} = {
  final_destiny_by_ccat  => { CC_VIRUS() => D_DISCARD },
  quarantine_method_by_ccat => { REPLACE => 1 },
};
```

Policy banks – by AM.PDP (milter)

AM.PDP protocol attribute:

policy_bank = AUTH, XYZ, ORIGINATING, ...

Policy banks – by custom hook

```
sub new {
    my($class, $conn, $msginfo) = @_;
    my($self) = bless {}, $class;
    if ( ... ) {
        Amavis::load_policy_bank(
                        'NOVIRUSCHECK' );
    }
    $self;
}
```

Policy banks – ACTION on load

2.7.0

```
$policy_bank{'TRUSTED_BOOKSHOPS'} = {
    bypass_spam_checks_maps => [1],
    spam_lovers_maps => [1],
    ACTION => sub { Amavis::Util::do_log(2,'Buying a book?');
                      Amavis::Util::snmp_count64('UserCounter2'); },
};

@author_to_policy_bank_maps = ({
    'amazon.com'   => 'TRUSTED_BOOKSHOPS',
    'amazon.co.uk' => 'TRUSTED_BOOKSHOPS',
    'amazon.de'     => 'TRUSTED_BOOKSHOPS',
});
```

Policy banks – Postfix side

```
# incoming mail MX
192.0.2.1:smtp inet  n  -  n  -  -  smtpd
   -o content_filter=amavisfeed:[127.0.0.1]:10040

# tcp port 587 for mail submission
submission inet  n  -  n  -  -  smtpd
   -o content_filter=amavisfeed:[127.0.0.1]:10042

# locally originating mail submitted on this host
pickup     fifo  n  -  n  60  1  pickup
   -o content_filter=amavisfeed:[127.0.0.1]:10043
```

Policy banks – Postfix side

```
content_filter = amavisfeed:[127.0.0.1]:10024

smtpd_sender_restrictions =
    check_client_access cidr:/etc/postfix/nets.cidr
    permit_mynetworks
    permit_sasl_authenticated
    check_sender_access
pcre:/etc/postfix/tag_as_inbound.pcre
```

overrides global *content_filter* setting */etc/postfix/nets.cidr* :

```
127.0.0.0/8 FILTER amavisfeed:[127.0.0.1]:10026
10.0.0.0/8   FILTER amavisfeed:[127.0.0.1]:10026
```

/etc/postfix/tag_as_inbound.pcre :

```
/^/       FILTER amavisfeed:[127.0.0.1]:10024
```

Lookup tables

many settings are lists of lookup tables

global assignment syntax:

```
@xxx_maps = ( ..., ..., ... );
```

syntax for policy banks (key / value):

```
xxx_maps => [ ..., ..., ... ],
```

Lookup tables

Static:
associative array (Perl hash)
a list (a.k.a. ACL) (Perl list)
list of regular expressions (object: list of *re*)
constant (Perl scalar)

Dynamic:
SQL, LDAP (Perl object)

Lookup tables – associative array

```
(   'me.ac.uk'   =>  1,
    '.ac.uk'     =>  0,
    '.uk'        =>  'indeed'  )
```

unordered set of key/value pairs
can provide any value (not just boolean)
predefined search order
lowercase search keys

read_hash('/etc/mydomains-hash')

Lookup tables – list (ACL)

```
    ( 'me.ac.uk',  '!.ac.uk',  '.uk' )
or:
qw(  me.ac.uk    !.ac.uk     .uk  )
```

sequential search, first match wins
can only provide booleans:
exclamation mark prefix: false

read_array('/etc/mydomains-list')

Lookup tables – regular expressions

```
new_RE(
   [ qr/ ^(noreply|offer) /i       => 0  ],
   [ qr/ [@.]example\.net$ /i  => 1  ],
     qr/ [@.]example\.net$ /i,  # shorthand
1
     qr/ [@.]example\.com$ /i,
)
```

sequential list, first match wins

can provide any value not just booleans

default rhs is a boolean true

Lookup tables – constant

trivial, always returns some constant (e.g. a string or a number) regardless of search key

useful as a final catchall

Lookup tables – SQL

```
CREATE TABLE users (
   id          SERIAL PRIMARY KEY,
   priority     integer,       -- 0 is low priority
   policy_id   integer  unsigned,
   email        varchar(255),
   local      char(1)
 );

CREATE TABLE policy (
   id          SERIAL PRIMARY KEY,
   spam_lover      char(1),
   virus_quarantine_to varchar(64),
    ...
);

SELECT  *,  users.id
  FROM users LEFT JOIN policy ON users.policy_id=policy.id
  WHERE users.email IN (?,?,?,...)
  ORDER BY users.priority DESC
```

Lookup tables – SQL

2.7.0

```
q_sql_s ('field-name')    ... string
q_sql_n ('field-name')  ... numeric
q_sql_b ('field-name')  ... boolean

@spam_kill_level_maps = (
  { ... },
  q_sql_n('spam_kill_level'),
  \$sa_kill_level_deflt,
);
```

Lookup tables – LDAP

2.7.0

```
q_ldap_s ('attribute-name')    ... string
q_ldap_n ('attribute-name')   ... numeric
q_ldap_b ('attribute-name')   ... boolean

@spam_kill_level_maps = (
  { ... },
  q_ldap_n('amavisSpamKillLevel'),
  \$sa_kill_level_deflt,
);
```

Lists of lookup tables: @*_maps

it became too awkward to have
one variable for each type of a
lookup table, and for each setting:

%local_domains # a hash
@local_domains_acl # a plain list
$local_domains_re # regexp list

solution:
a list of lookup tables of arbitrary
types

Lists of lookup tables: @*_maps

```
@local_domains_maps = (
   \%local_domains,
   \@local_domains_acl,
   \$local_domains_re,
);
```

actually: list of references to lookup tables

Lists of lookup tables: @*_maps

2.7.0

program only consults these *@_*maps* variables, no longer the individual old settings like *%local_domains*

2.7.0: explicit and/or implicit SQL or LDAP

```
@local_domains_maps = (
    [...list1...],  {...hash1...},  [...list2...],
    new_RE(...re1...),  read_hash('/etc/myfile'),
    %local_domains,  {...hash3...},  constant,
    q_sql_s('field'), q_ldap_s('attr'),
);
```

Remember:

policy banks affect message as a whole,
so can only depend on some common characteristic of a message, e.g. client's IP address, sender address / DKIM, TCP port number

lookups serve to implement
per-recipient settings
(and some other things)

Content categories

CC_VIRUS	
CC_BANNED	
CC_UNCHECKED	
CC_SPAM	above kill level
CC_SPAMMY	above tag2 level
CC_BADH	
CC_OVERSIZED	
CC_MTA	
CC_CLEAN	

Content categories

C_VIRUS	C_BANNED	C_UNCHECKED	C_SPAM	C_SPAMMY	C_BADH	C_OVERSIZED	C_MTA	C_CLEAN	
x			x	x	x			x	test results
x									lovers (mask)

main ccat

blocking ccat

Content categories

```
%subject_tag_maps_by_ccat = (
     CC_VIRUS,    [ '***INFECTED*** ' ],
     CC_BANNED,[ '***BANNED*** ' ],
     CC_UNCHECKED,
                    [$undecipherable_subject_tag],
     CC_SPAM,     undef,
     CC_SPAMMY,   \@spam_subject_tag2_maps,
     CC_CLEAN.',1',  \@spam_subject_tag_maps,
);
```

Monitoring health: amavisd-nanny

```
PID 28039: 28039-02      0:00:05 GSSSr
PID 28048: .             0:00:05 .....
PID 28174: 28174-01-10   0:00:02 VS
PID 28309: A             0:00:00
```

db key: PID
db data: timestamp of last event, status

status:

empty - idle child process

A   - just accepted a connection (post_accept_hook)

am_id - processing am_id task

.   - content checking done

$ amavisd-nanny -h

```
States legend:
  A  accepted a connection
  b  begin with a protocol for accepting a request
  m 'MAIL FROM' smtp command started a new
          transaction in the same session
  d  transferring data from MTA to amavisd
  =  content checking just started
  G  generating and verifying unique mail_id
  D  decoding of mail parts
  V  virus scanning
  S  spam scanning
  P  pen pals database lookup and updates
  r  preparing results
  Q  quarantining and preparing/sending notifications
  F  forwarding mail to MTA
  .  content checking just finished
  sp space indicates idle (elapsed bar showing dots)
```

Monitoring health: amavisd-nanny

normal

```
PID 27948: 27948-02-4    0:00:02 SF
PID 27987:               0:00:05 .....
PID 28039: 28039-02      0:00:05 DVSSS
PID 28048: .             0:00:05 .....
PID 28101: 28101-01-9    0:00:01 =
PID 28174: 28174-01-10   0:00:02 dV
PID 28187: 28187-01-5    0:00:12 VVSSSSSSSS:SS
PID 28245: 28245-01-4    0:00:07 GVSSSSS
PID 28309: A             0:00:00
```

Monitoring health: amavisd-nanny

mostly idle

```
PID 28187: 28187-02-8    0:00:02 SS
PID 28245:               0:01:16 .........:........>
PID 28309:               0:01:16 .........:........>
PID 28543: 28543-01-7    0:00:03 VSS
PID 28584: 28584-01-7    0:00:01 S
PID 28672:               0:00:24 .........:........
PID 28677:               0:01:06 .........:........>
PID 28678:               0:01:06 .........:........>
PID 28729:               0:00:56 .........:........>
```

Monitoring health: amavisd-nanny

trouble - crashed programs

```
PID 25408: 25408-01  went away  0:02:27 ==========:==>
PID 25496: 25496-01  went away  0:01:58 ==========:==>
PID 25728: 25728-01  went away  0:02:06 ==========:==>
```

process no longer exists, but is still registered in db
mail stays in MTA queue (temporary failure)

usual reasons:
bug in a library routine such as uulib, zlib, bdb
resources exceeded: *Lock table is out of available locker entries,*
    stack size, runaway regexp in custom rules

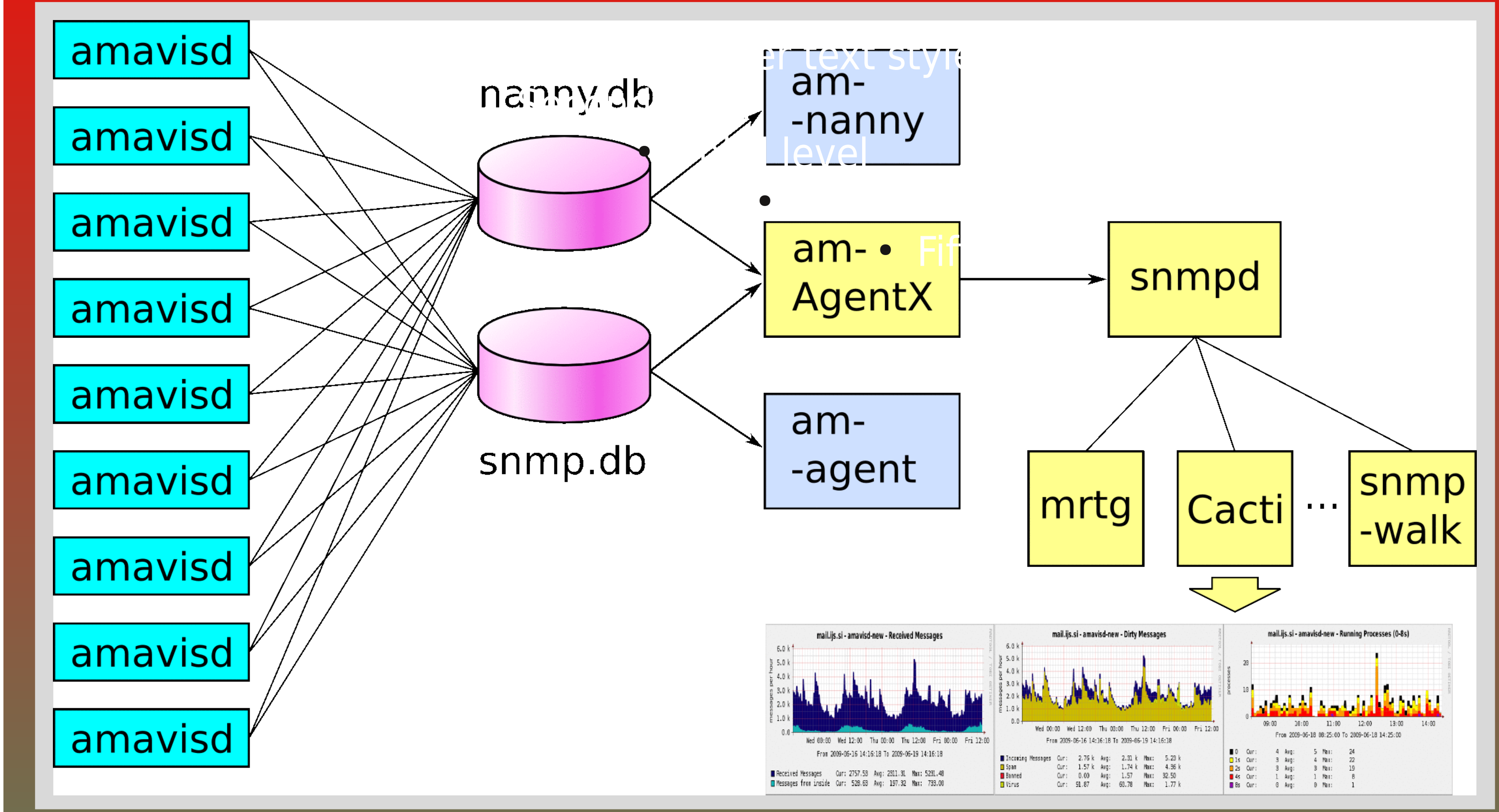
Monitoring health: amavisd-nanny trouble - looping or forgotten proc.

```
PID 25733: 25733-01  terminated 2:10:56 ==========:=>
```

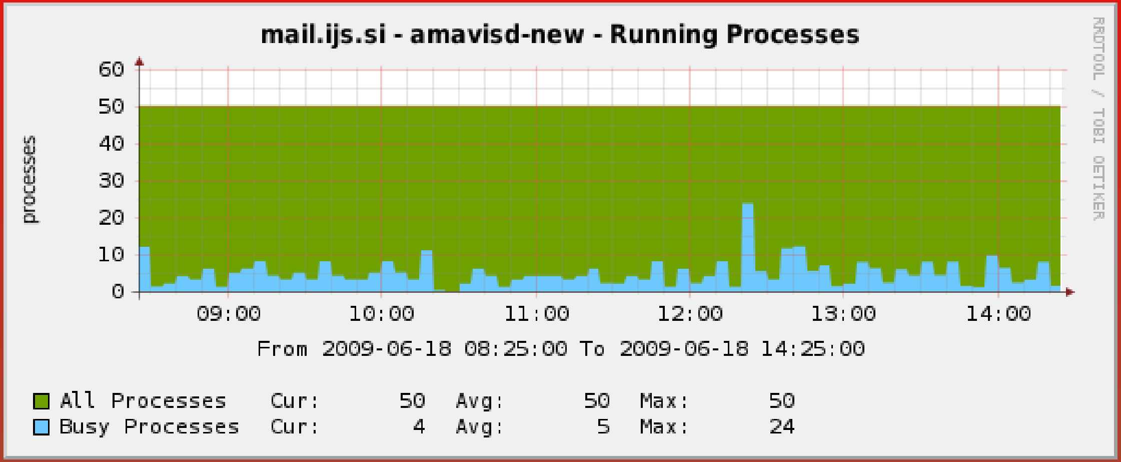
amavisd-nanny sends SIGTERM first
amavisd-nanny sends SIGKILL 30 seconds later if necessary

active ttl = 10 minutes   stuck active children
idle ttl   = 1 hour   unused idle process
(may be normal)

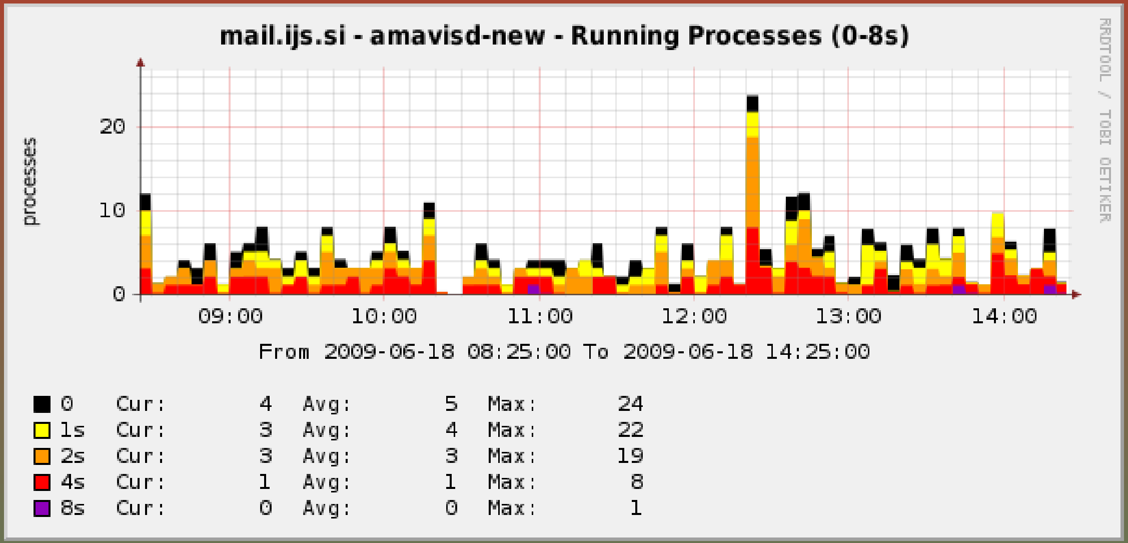
Monitoring – components


amavisd
amavisd
amavisd
amavisd
amavisd
amavisd
amavisd
amavisd
amavisd
nanny.db
snmp.db
er text style
am-
-nanny
level
am- •
Fif
AgentX
am-
-agent
snmpd
mrtg
Cacti
…
snmp
-walk
mail.ijs.si - amavisd-new - Received Messages
messages per hour
6.0 k
5.0 k
4.0 k
3.0 k
2.0 k
1.0 k
0.0
Wed 00:00
Wed 12:00
Thu 00:00
Thu 12:00
Fri 00:00
Fri 12:00
From 2009-06-16 14:16:18 To 2009-06-19 14:16:18
Received Messages Cur: 2757.53 Avg: 2311.31 Max: 5231.48
Messages from inside Cur: 528.63 Avg: 197.32 Max: 733.00
mail.ijs.si - amavisd-new - Dirty Messages
messages per hour
From 2009-06-16 14:16:18 To 2009-06-19 14:16:18
Incoming Messages Cur: 2.76 k Avg: 2.31 k Max: 5.23 k
Spam Cur: 1.57 k Avg: 1.74 k Max: 4.36 k
Banned Cur: 0.00 Avg: 1.57 Max: 32.50
Virus Cur: 91.87 Avg: 63.78 Max: 1.77 k
mail.ijs.si - amavisd-new - Running Processes (0-8s)
processes
20
10
0
09:00
10:00
11:00
12:00
13:00
14:00
From 2009-06-18 08:25:00 To 2009-06-18 14:25:00
0 Cur: 4 Avg: 5 Max: 24
1s Cur: 3 Avg: 4 Max: 22
2s Cur: 3 Avg: 3 Max: 19
4s Cur: 1 Avg: 1 Max: 8
8s Cur: 0 Avg: 0 Max: 1

SNMP: load, timing



🟩 All Processes	Cur:	50	Avg:	50	Max:	50
🟦 Busy Processes	Cur:	4	Avg:	5	Max:	24



■ 0	Cur:	4	Avg:	5	Max:	24
🟨 1s	Cur:	3	Avg:	4	Max:	22
🟧 2s	Cur:	3	Avg:	3	Max:	19
🟥 4s	Cur:	1	Avg:	1	Max:	8
🟪 8s	Cur:	0	Avg:	0	Max:	1

Statistics: amavisd-agent

```
sysUpTime              (0 days, 14:03:43.46)

InMsgs                14490   1030/h  100.0 % (InMsgs)
InMsgsRecips           27169   1932/h  187.5 % (InMsgs)

ContentCleanMsgs     6020    428/h   41.5 % (InMsgs)
ContentSpamMsgs       7807    555/h   53.9 % (InMsgs)
ContentVirusMsgs     567      40/h    3.9 % (InMsgs)

ContentBadHdrMsgs      91      6/h    0.6 % (InMsgs)
ContentBannedMsgs       5      0/h    0.0 % (InMsgs)
```

Statistics: amavisd-agent

```
OpsSpamCheck           12719    904/h   87.8 % (InMsgs)
OpsVirusCheck          13231    941/h   91.3 % (InMsgs)
OpsSqlSelect           50680   3604/h  186.5 % (InMsgsRc)

OutMsgs                 6248    444/h  100.0 % (OutMsgs)
OutMsgsDelivers         6248    444/h  100.0 % (OutMsgs)

OutForwMsgs             6155    438/h   98.5 % (OutMsgs)

OutDsnMsgs                35      2/h    0.6 % (OutMsgs)
OutDsnBannedMsgs           3      0/h    0.0 % (OutMsgs)
OutDsnSpamMsgs            32      2/h    0.5 % (OutMsgs)
```

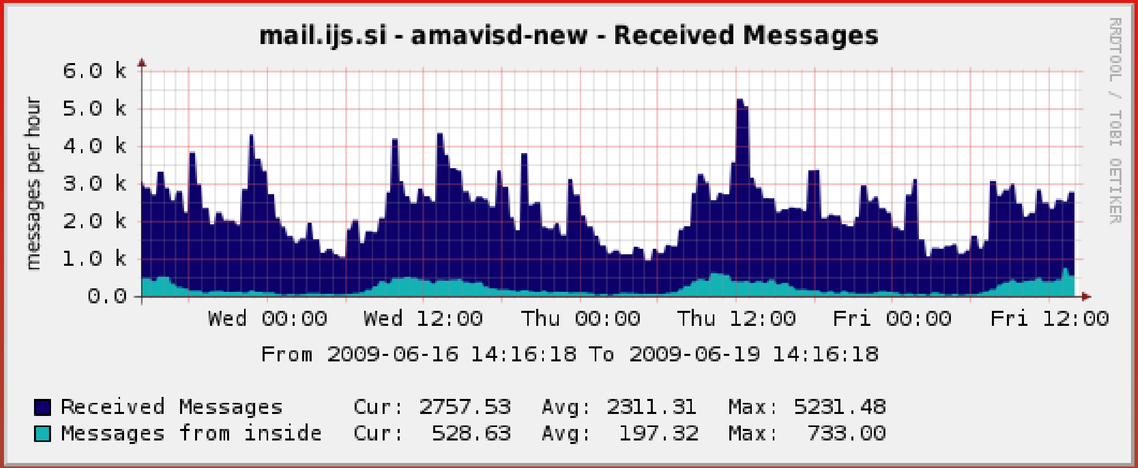
Statistics: amavisd-agent

```
QuarMsgs                2704    192/h  100.0 % (QuarMsgs)
QuarSpamMsgs            2100    149/h   77.7 % (QuarMsgs)
QuarVirusMsgs            567     40/h   21.0 % (QuarMsgs)
QuarBannedMsgs             5      0/h    0.2 % (QuarMsgs)
QuarOther                 32      2/h    1.2 % (QuarMsgs)
```

Statistics: amavisd-agent

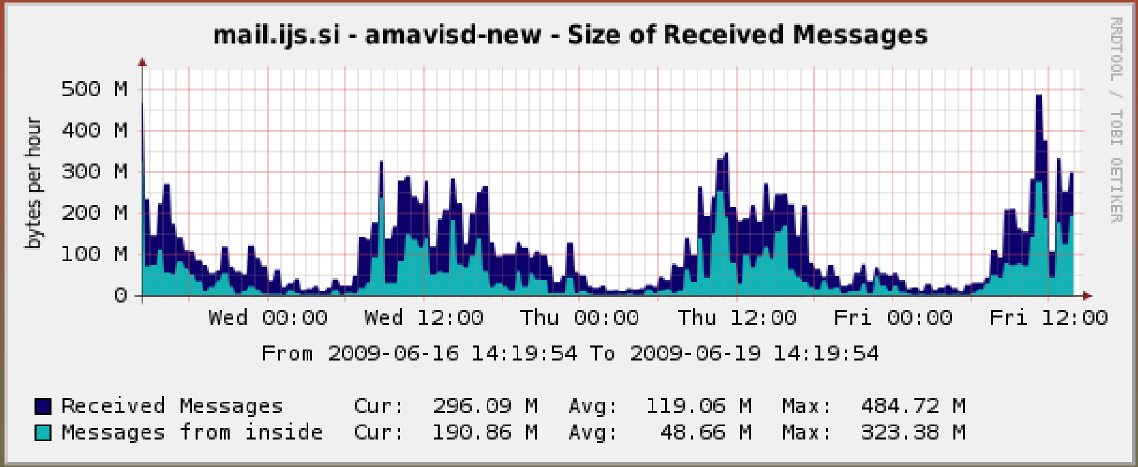
```
W32/Netsky-P          191  14/h
W32/Mytob-CA           59   4/h
W32/Netsky-D           25   2/h
W32/Lovgate-V          21   1/h
W32/Netsky-Q           21   1/h
W32/Bagle-AG           17   1/h
HTML.Phishing.Pay-1   18   1/h
HTML.Phishing.Bank-1  12   1/h
W32/Mytob-Z            11   1/h
W32/Wurmark-J          11   1/h
W32/Lovgate-X          11   1/h
```

SNMP: mail rate, size



From 2009-06-16 14:16:18 To 2009-06-19 14:16:18

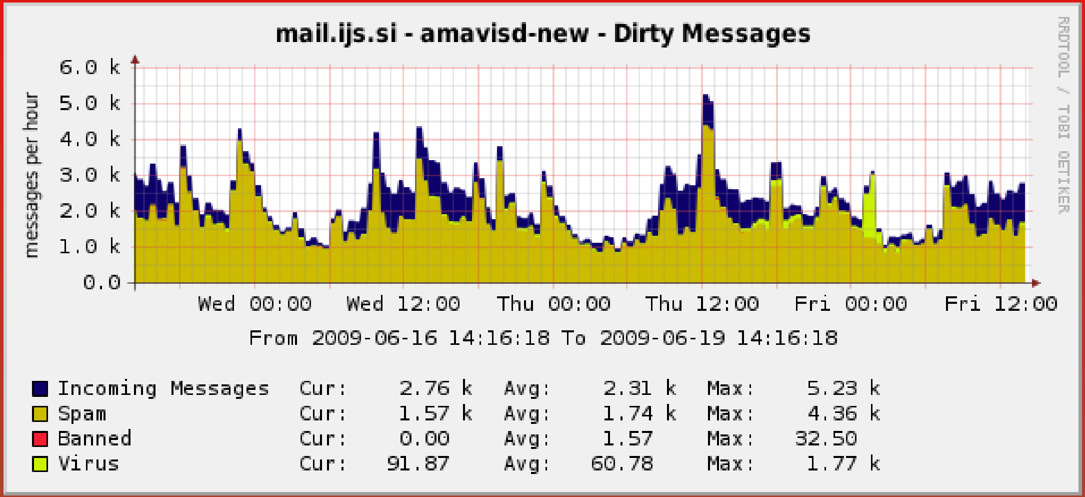
```
■ Received Messages      Cur: 2757.53   Avg: 2311.31   Max: 5231.48
■ Messages from inside   Cur:  528.63   Avg:  197.32   Max:  733.00
```



From 2009-06-16 14:19:54 To 2009-06-19 14:19:54

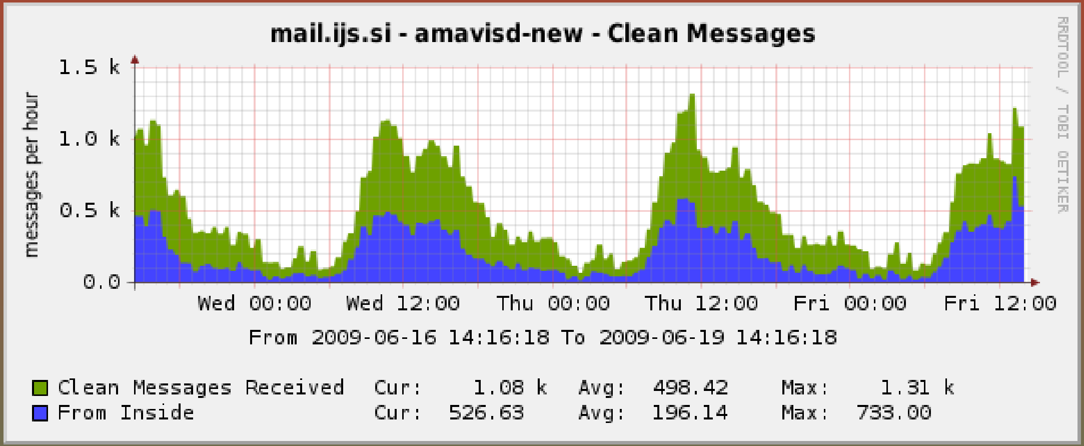
```
■ Received Messages      Cur: 296.09 M   Avg: 119.06 M   Max: 484.72 M
■ Messages from inside   Cur: 190.86 M   Avg:  48.66 M   Max: 323.38 M
```

SNMP: mail content



From 2009-06-16 14:16:18 To 2009-06-19 14:16:18

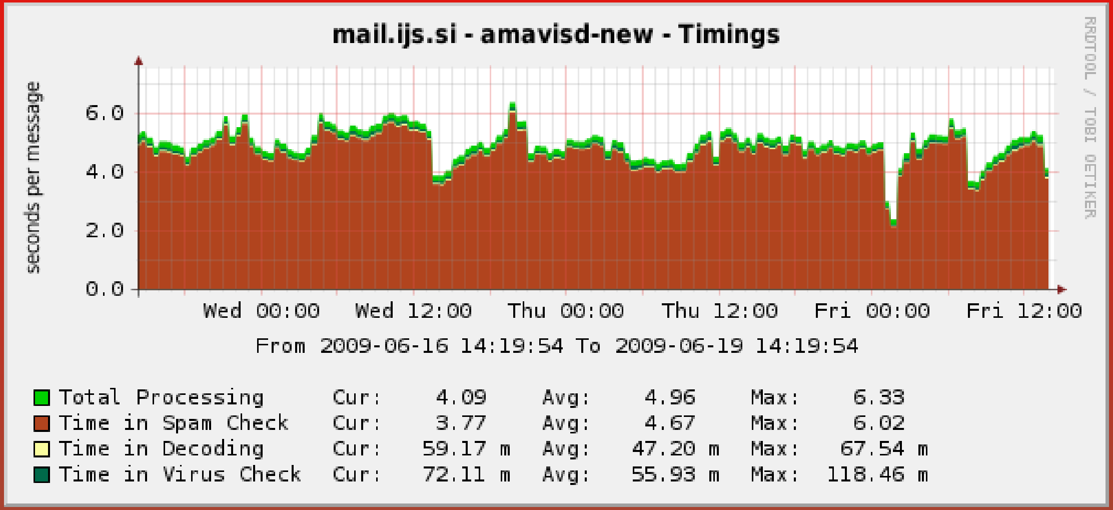
🟦 Incoming Messages	Cur: 2.76 k	Avg: 2.31 k	Max: 5.23 k
🟨 Spam	Cur: 1.57 k	Avg: 1.74 k	Max: 4.36 k
🟥 Banned	Cur: 0.00	Avg: 1.57	Max: 32.50
🟩 Virus	Cur: 91.87	Avg: 60.78	Max: 1.77 k

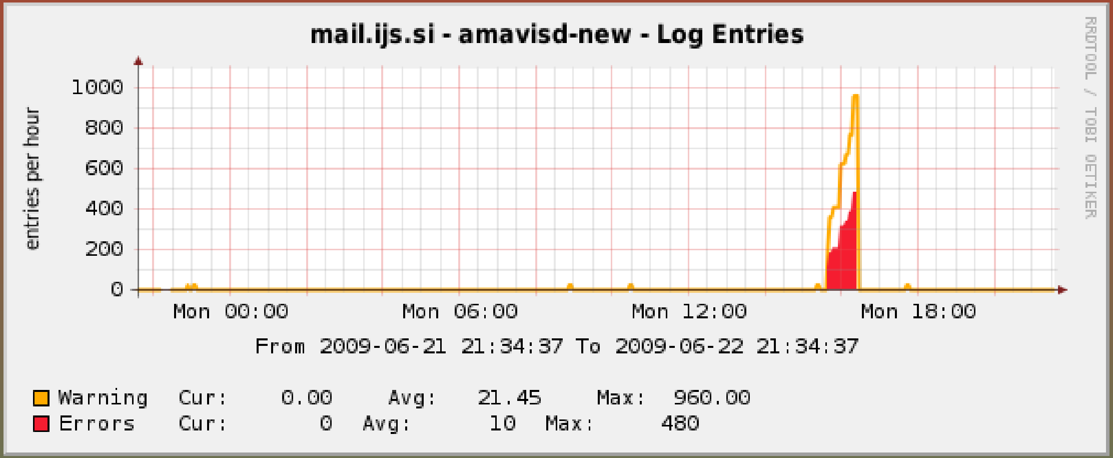


From 2009-06-16 14:16:18 To 2009-06-19 14:16:18

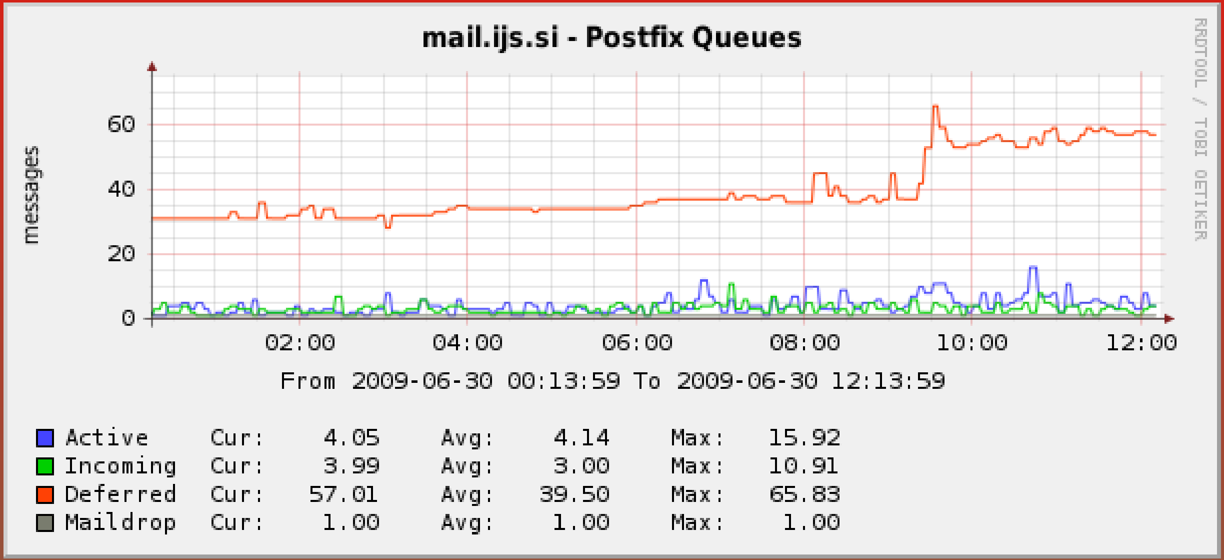
🟩 Clean Messages Received	Cur: 1.08 k	Avg: 498.42	Max: 1.31 k
🟦 From Inside	Cur: 526.63	Avg: 196.14	Max: 733.00

SNMP: elapsed time, errors





SNMP: Postfix queue entries



From 2009-06-30 00:13:59 To 2009-06-30 12:13:59

```
🟦 Active     Cur:    4.05    Avg:    4.14    Max:   15.92
🟩 Incoming   Cur:    3.99    Avg:    3.00    Max:   10.91
🟧 Deferred   Cur:   57.01    Avg:   39.50    Max:   65.83
⬛ Maildrop   Cur:    1.00    Avg:    1.00    Max:    1.00
```

Details in the log: timing report

```
TIMING [total 1725 ms] –
        lookup_sql: 6 (0%)0,
        SMTP pre-DATA-flush: 1 (0%)0, SMTP DATA: 88 (5%)6,
        body_hash: 1 (0%)6, sql-enter: 4 (0%)6,
        mime_decode: 6 (0%)6, get-file-type1: 23 (1%)7,
        parts_decode: 0 (0%)8,
        AV-scan-1: 7 (0%)8, AV-scan-2: 4 (0%)8, AV-scan-3: 5 (0%)8,
        AV-scan-4: 1 (0%)9, AV-scan-5: 1 (0%)9, AV-scan-6: 0 (0%)9,
        lookup_sql: 4 (0%)9, spam-wb-list: 3 (0%)9,
        SA msg read: 0 (0%)9, SA parse: 2 (0%)9,
        SA check: 1536 (89%)98,
        update_cache: 2 (0%)98, post-do_spam: 6 (0%)99,
        deal_with_mail_size: 0 (0%)99, main_log_entry: 18 (1%)100,
        sql-update: 4 (0%)100, update_snmp: 1 (0%)100,
        unlink-1-files: 1 (0%)100, rundown: 0 (0%)100
```

Details in the log: SpamAssassin 3.3 timing

```
TIMING-SA total 3491 ms -
     parse: 1.67 (0.0%), extract_message_metadata: 6
(0.2%),
     get_uri_detail_list: 0.49 (0.0%), tests_pri_-1000: 13
(0.4%),
     tests_pri_-950: 0.73 (0.0%), tests_pri_-900: 0.87
(0.0%),
     tests_pri_-400: 16 (0.5%), check_bayes: 15 (0.4%),
     tests_pri_0: 3106 (89.0%), check_dkim_adsp: 2 (0.1%),
     check_spf: 5 (0.2%), poll_dns_idle: 0.25 (0.0%),
     check_razor2: 1759 (50.4%), check_dcc: 1268 (36.3%),
     tests_pri_500: 7 (0.2%), tests_pri_899: 77 (2.2%),
check_crm114: 76 (2.2%), tests_pri_1000: 11 (0.3%),
     total_awl: 10 (0.3%), check_awl: 3 (0.1%),
     update_awl: 2 (0.1%), learn: 226 (6.5%),
     crm114_autolearn: 201 (5.7%), get_report: 1.15 (0.0%)
```

troubleshooting

amavisd-nanny
amavisd log and MTA log
increase log level if necessary
search log for am_id of a trouble message

strace -f amavisd foreground

troubleshooting

# *amavisd debug*
# *amavisd debug-sa*
# *amavisd foreground*
selective debug: *@debug_sender_maps*
selective debug: dedicated policy bank with elev. log
compare output of '*amavisd debug-sa*'
    to '*su vscan -c spamassassin -t -D*'

Regular maintenance tasks

run *amavisd-nanny* or SNMP, note any '*process went away*' reports, investigate and fix the problem if any

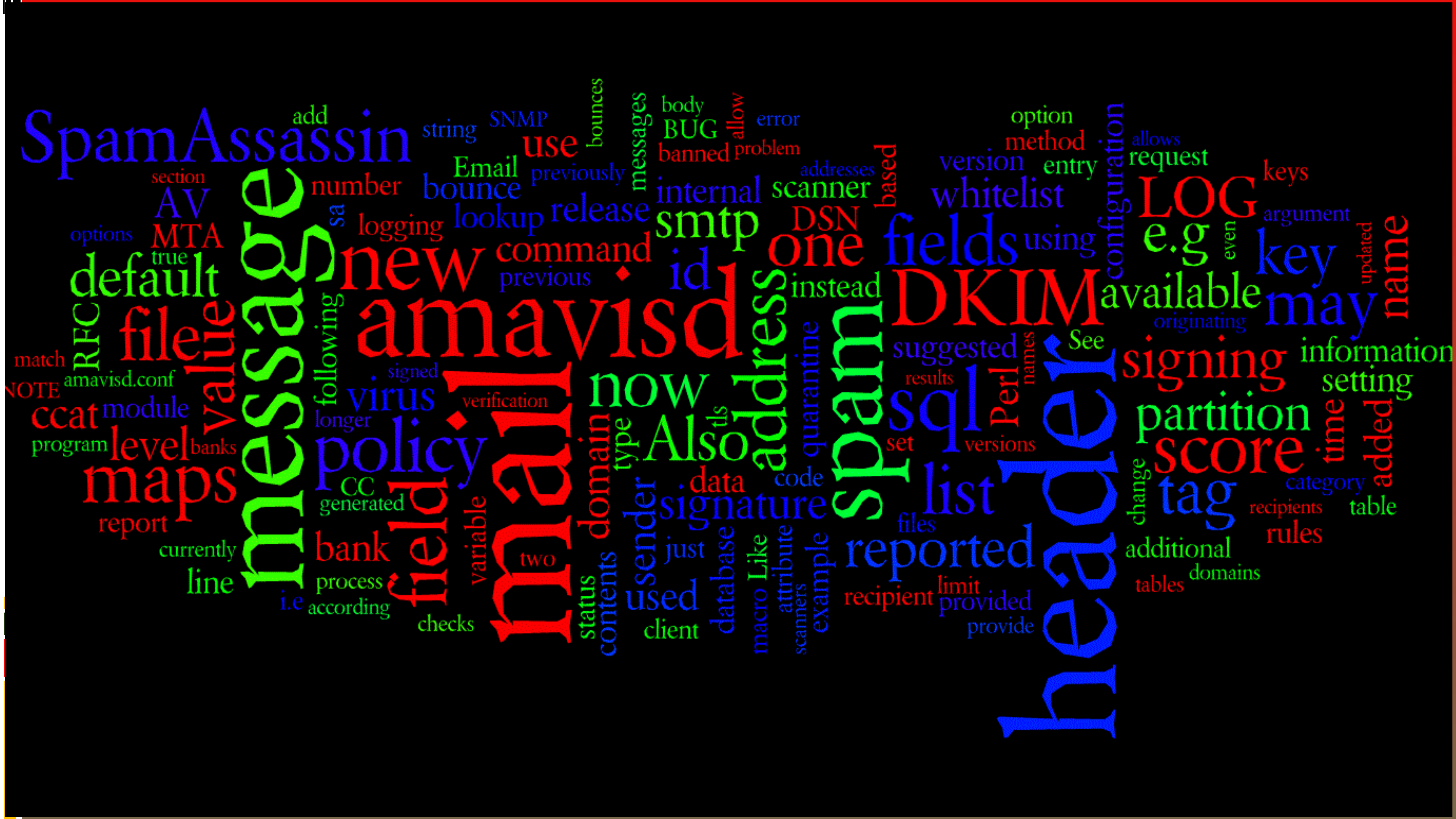
check *mailq* or *qshape* for stalled messages

check for preserved directories in */var/amavis/tmp*, search log for explanation, fix the problem and delete

remove old quarantine and SQL logs

Questions?

mailing list
hang around and ask
...


SpamAssassin
add
string
SNMP
bounces
messages
body
BUG
allow
error
option
method
allows
request
configuration
use
banned
problem
section
Email
previously
addresses
based
version
entry
keys
AV
number
bounce
internal
scanner
whitelist
LOG
argument
options
MTA
sa
logging
lookup
release
smtp
DSN
fields
using
e.g
even
updated
true
message
new
command
one
key
name
default
previous
id
instead
DKIM
available
may
file
RFC
amavisd
originating
value
following
suggested
names
See
signing
information
match
signed
address
quarantine
spam
results
Perl
setting
NOTE
amavisd.conf
virus
now
sql
ccat
module
verification
tls
partition
time
longer
mail
header
level
program
banks
policy
type
Also
set
versions
score
added
maps
domain
CC
data
code
list
change
tag
category
generated
signature
recipients
table
report
field
variable
sender
files
rules
currently
bank
two
just
reported
additional
line
process
database
Like
attribute
example
limit
domains
tables
i.e
according
status
macro
scanners
recipient
provided
checks
contents
used
client
provide