masqmail

a mail transfer agent for workstations and small networks

markus schnalke <meillo@marmaro.de>

about me

markus schnalke

past:

Business Information Systems
at University of Applied Sciences, Ulm

Diploma Thesis about masqmail

future:

Master in Informatics
here at University, Ulm

I am interested in

Unix Philosophy

suckless software

Free Software

Debian

this talk

overview

1) prerequisites (5%)

2) about masqmail (25%)

3) the thesis (60%)

4) help wanted (10%)

email? MTA? . . .

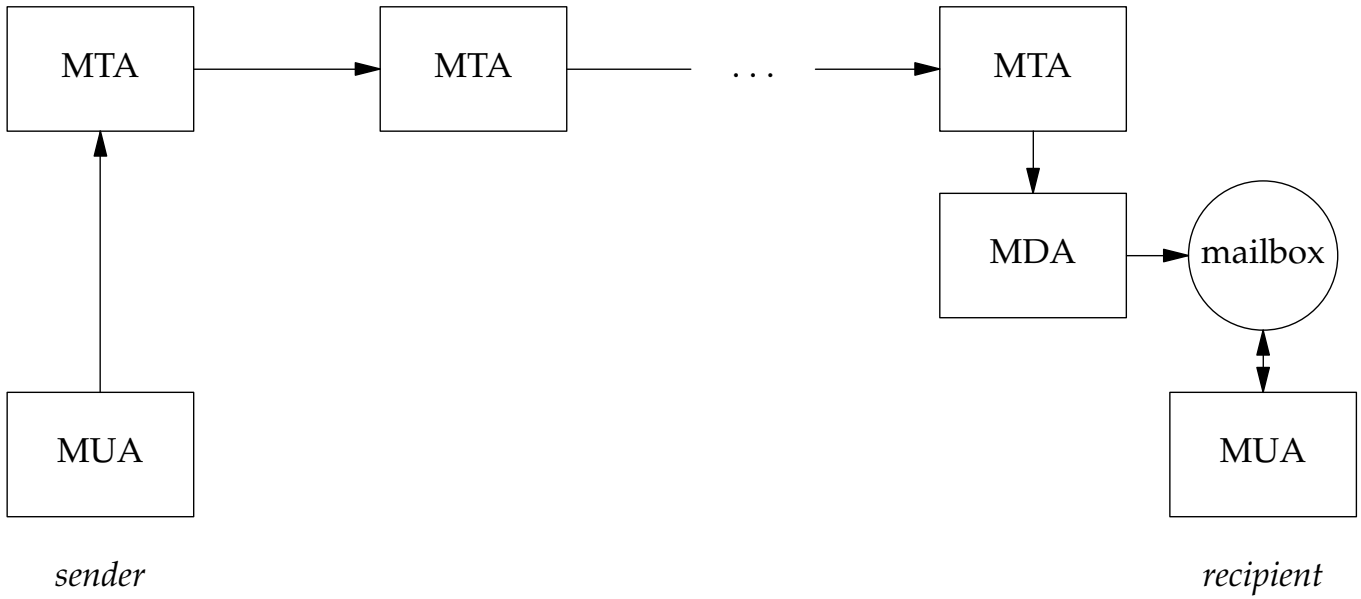
What are MTAs

MTA = Mail Transfer Agent

the post offices of email

MTAs transfer mail from senders to recipients

mail agents



questions?

about masqmail

what is masqmail

an MTA

a small one

one for special setups

masqmail's niche

non-permanent online connections

over multiple providers

especially for workstations and small networks

also notebooks

history

written by Oliver Kurth

1999 – 2003

GPL

about 9 000 lines of C

guessed

initially written to use masqdialer-controlled online connections

thus: source of the name (“masq. . .”) (not “. . . qmail”)

masqmail today

development stopped in 2003

not maintained

even its original website is down

but included in Debian, Ubuntu, Arch, Mandriva, FreeBSD

value of masqmail

covers a niche

has unique features

has users!

problems of masqmail

no further development

emailing has changed

unfixed bugs

questions?

when I met masqmail

Fall 2007

started to use masqmail

because of its small size

Spring 2008

problem: realized it is unmaintained in Debian

A situation to be changed ... but by whom?

. . . me? No!

Summer 2008

problem: find a topic for the diploma thesis

while lying in bed some night

the great idea: solve both problems at once

1) **use the diploma thesis to revive masqmail**

2) **use masqmail to do a great diploma thesis**

advising professor

professor Markus Schäffter

thanks!

logical decisions

motivation:

- for myself
- for the community

thus:

- write in English
- make the thesis free available

guiding principle

Do what you think is interesting,
do something that you think is fun and worthwhile,
because otherwise you won't do it well anyway.

—Brian W. Kernighan—

the thesis in short

ch01: introduction

general prerequisites

the masqmail project

value and problems of masqmail

$\Rightarrow$ **reasons to do this thesis**

ch02: market analysis

classification, life cycle, and trends of electronic communication

SWOT and trends for email

$\Rightarrow$ **show that email is expected to survive**

ch03: mail transfer agents

classification of MTAs

masqmail's competitors

short comparison

$\Rightarrow$ **provide knowledge for following chapters**

ch04: masqmail's present and future

the goal (revive masqmail)

requirements

- fulfilled requirements
- work to do

ways for further development

result

$\Rightarrow$ **the core of the thesis: a plan to revive masqmail**

ch05: improvement plans

more detailed descriptions of the work tasks

$\Rightarrow$ **recommendations how to do the proposed changes**

questions?

we're half way through :-)

in detail: ch03 MTAs

MTAs

relay-only MTAs (forwarders):

nullmailer, ssmtp

groupware:

Lotus Notes, Exchange, opengroupware.org

"real" MTAs:

sendmail, exim, qmail, postfix, masqmail

market shares

#	Bernstein	2001	O'ReillyNet	2007	MailRadar	?
1	**sendmail**	42 %	**sendmail**	12 %	**sendmail**	24 %
2	Microsoft Exchange	18 %	**postfix**	9 %	**postfix**	20 %
3	**qmail**	17 %	Microsoft Exchange	8 %	**qmail**	17 %
4	IMail	6 %	**qmail**	5 %	Microsoft Mail	15 %
5	**postfix**	2 %	**exim**	5 %	**exim**	13 %
6	**exim**	2 %	Cisco	3 %	IMail	2 %
7	NTMail	1 %	Barracuda	3 %	Microsoft Exchange	1 %
	mail security layers	4 %	*mail security layers*	22 %	*mail security layers*	2 %

comparison

MTA	first release	lines of C code	architecture	design goals
sendmail	1983	92k	monolithic	flexibility
exim	1995	85k	monolithic	generality, flexibility, extensive checking
qmail	1996	14k	modular	security
postfix	1999	87k	modular	performance and security
masqmail	1999	9k	monolithic	non-permanent Internet connections

why not sendmail

- too large
- too complex
- structurally insecure
- simply obsolete

why not exim

- too large

- designed for always online hosts

why not qmail

- outdated

- designed for always online hosts with fast connections

why not postfix

- too large
- too complex
- not designed for masqmail's scenario

in summary

- other MTAs are bad substitutes for masqmail

- though they seem to be generally able to be substitutes

comments and annotations?

in detail: ch04 present and future

functional requirements

Requirement	Importance	Pending work	Focus
RF 1: In/out channels	++	-	+
RF 2: Mail queuing	++	-	+
RF 3: Header sanitizing	0	-	-
RF 4: Aliasing	0	-	-
RF 5: Route management	+	-	0
RF 6: Authentication	++	+	+++
RF 7: Encryption	++	+	+++
RF 8: Spam handling	+	++	+++
RF 9: Malware handling	-	+	0
RF 10: Archiving	-	+	0

non-functional requirements

Requirement	Importance	Pending work	Focus
RG 1: Security	++	+	+++
RG 2: Reliability	++	+	+++
RG 3: Robustness	+	+	++
RG 4: Extendability	+	++	+++
RG 5: Maintainability	+	0	+
RG 6: Testability	0	0	0
RG 7: Performance	--	-	---
RG 8: Availability	-	-	--
RG 9: Portability	-	--	---
RG 10: Usability	+	--	-

work tasks

TODO 1: Encryption

TODO 2: Authentication

TODO 3: Security

TODO 4: Reliability

TODO 5: Spam handling

TODO 6: Extendability

further development

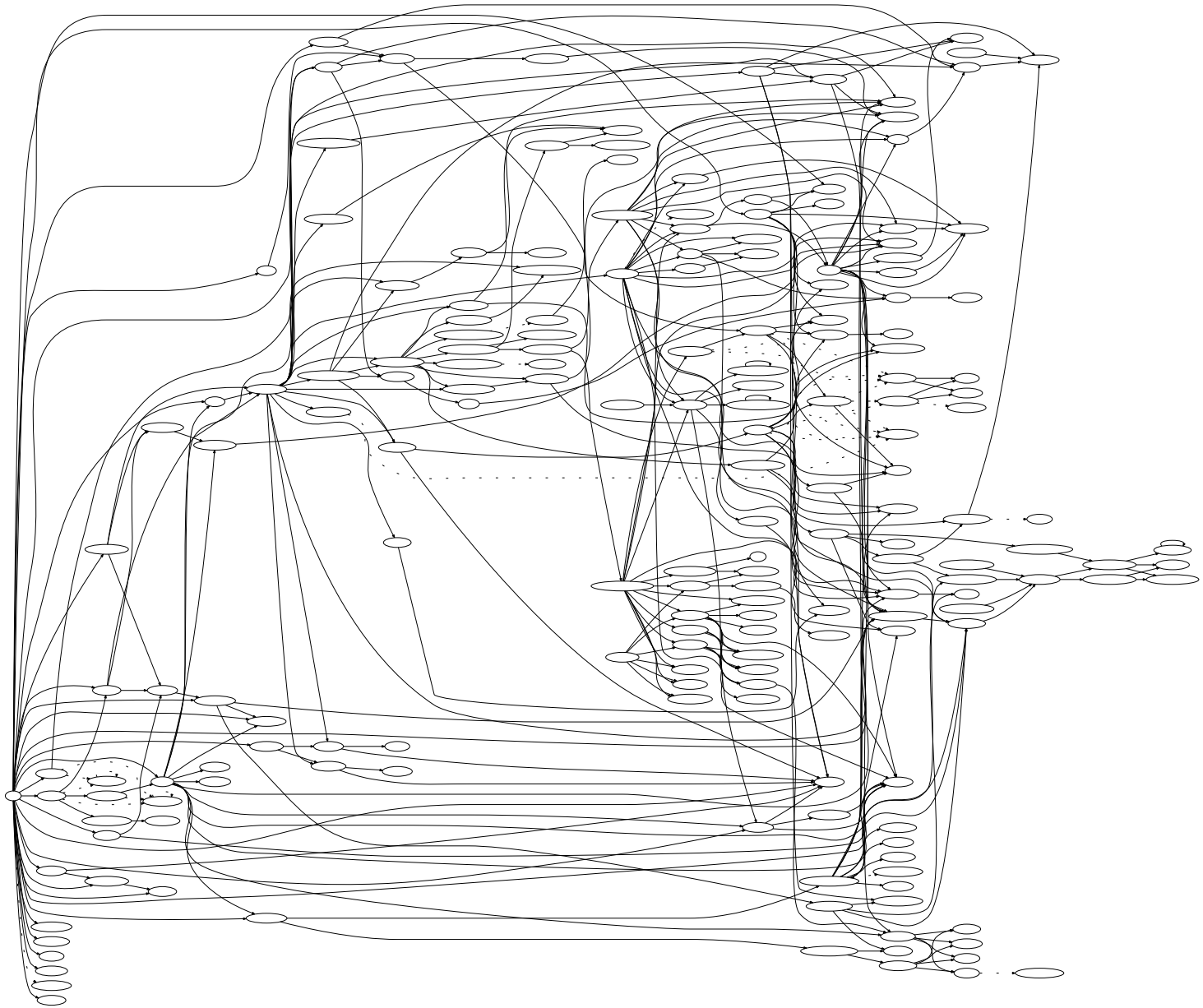
1) improve existing code

2) add wrappers and interposition filters

3) redesign and rewrite from scratch

$1) + 2) = 13$ vs. $3) = 17$

architecture



the result

"one fits all" is not possible

hence: different strategies for different goals

short-term goal: keep masqmail usable
$\longrightarrow$ improve existing code

long-term goal: make masqmail future-proof
$\longrightarrow$ recreate it from scratch

disadvantage: more work

questions?

in detail: ch05 (1) existing code

TODO 1: Encryption

STARTTLS

qmail patch as template

affected files: smtp_in.c, smtp_out.c, conf.c

TODO 2: Authentication

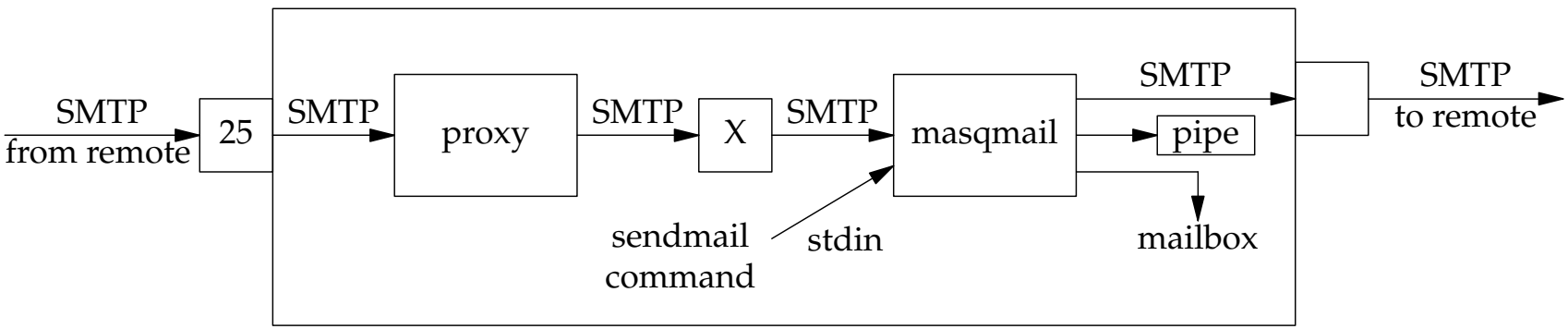
static: TCP Wrapper

dynamic: SMTP-AUTH with SASL

affected files: smtp_in.c, smtp_out.c, conf.c

TODO 3: Security

mail security layers



questions?

in detail: ch05 (2) new design

design goals

throughout compartmentalization

free internal system from in and out channels

single point for scanning

concentrate on mail transfer

keep it simple, clear, and general

incoming and outgoing

in/out:

- arbitrary protocol handlers (incoming/outgoing)
- one queue-in module; one queue-out module
- no own MDA, only a pipe module

route management:

- the thing special to masqmail
- done just before the outgoing modules

within the queue

mail sanitizing:

- qmail: “don’t parse” $\longrightarrow$ parse very carefully
- Jon Postel’s robustness principle

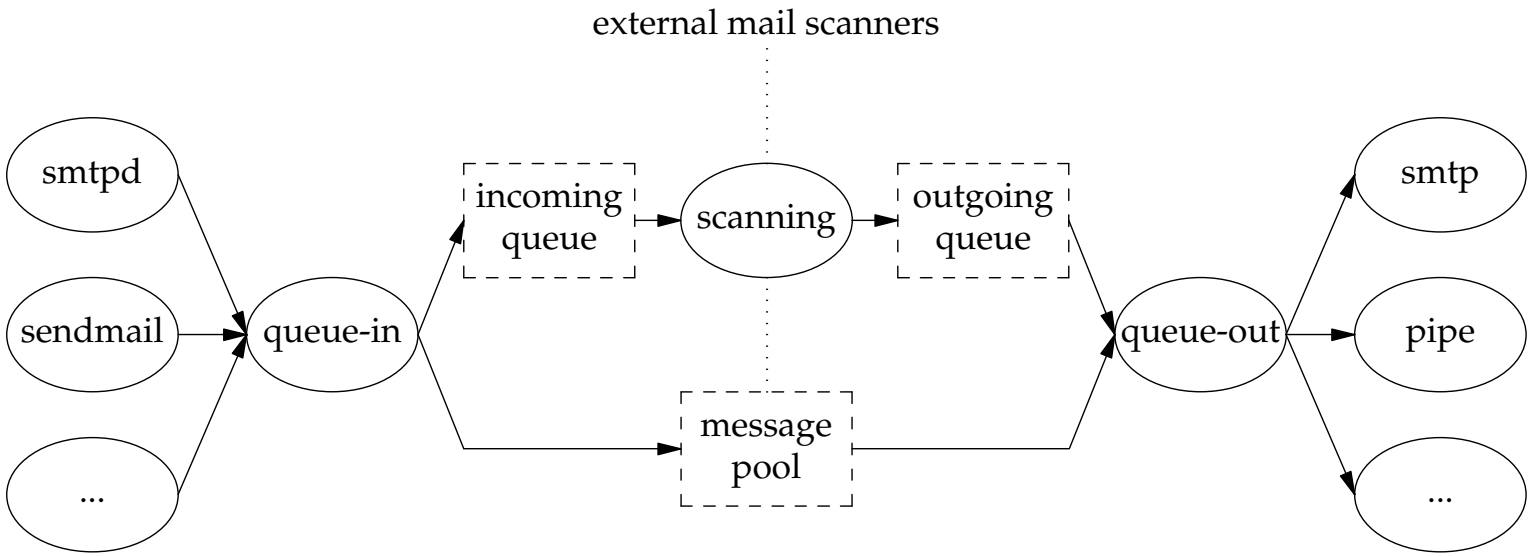
spool files:

- preferably in the system’s native format

auth, enc, spam

- auth and enc:
  - in receiving and sending modules (?)
  - with library code
- spam and malware:
  - receiving modules vs. a scanning module
  - in both places?
  - malware scanning is not the MTA's job

the proposed architecture



questions?

help wanted!

work to do

add encryption, authentication, security

fix bugs

documentation

user guides and “marketing”

create the new masqmail

current team

me

project site:

`http://prog.marmaro.de/masqmail`

mailing list:

masqmail@marmaro.de

Debian

I search for a DD to sponsor my masqmail work

please contact me!

you!

I welcome your help!

I appreciate your advice!

I am thankful for your comments!

help masqmail

Thanks for your attention

links

my diploma thesis

`http://marmaro.de/docs#diploma`

masqmail's website

`http://prog.marmaro.de/masqmail`

masqmail mailing list

`masqmail@marmaro.de`

`minimalist@marmaro.de?subject=subscribe%20masqmail`

suckless software

`http://www.suckless.org`

software used

Debian GNU/Linux

LaTeX with the beamer classes

Vim, latexmk, and Mercurial

diagrams: PIC and groff

call graph: Egypt and dot