

masqmail

a mail transfer agent for workstations and small networks

markus schnalke
meillo@marmaro.de

Abstract

This paper describes a plan to revive masqmail. It is a short overview of my diploma thesis [2].

1 Introduction

masqmail [1] is a small Mail Transfer Agent (MTA). It focuses on workstations and small networks that have non-permanent online connections. In these cases it is a small replacement for sendmail, exim, qmail, and postfix.

masqmail was written by Oliver Kurth between 1999 and 2003. Since then it was not developed further. In the meanwhile it is not even maintained and its original website is offline. Hence bugs have not been fixed as they appeared and the program did not evolve as emailing changed.

Nevertheless, the program has a remaining user base, mostly because no other MTA covers setups with non-permanent online connections over multiple providers equally well.

This discrepancy should get removed.

2 Market analysis

Before putting effort into some old software, it is important to know if the software is still important and if it will probably be important in future. If a software has no more relevance, effort is better put into something else. Fortunately, this is not the case for masqmail.

Email will likely stay important during the next decade. And if email itself becomes obsolete, it is pretty sure that a similar technology will supersede it. Thus MTAs, or programs pretty similar to them, will be still of need in the next years.

Get this document from <http://marmaro.de/docs>.

masqmail's kind of MTA was important some years ago when dial-up online connections were common. When they became rare, masqmail's decline began. The future will likely show new fields of operation for masqmail. Home servers become popular, thus small MTAs are needed again. Notebooks are frequently offline and they often move between different networks; they are perfect hosts for masqmail.

3 Mail transfer agents

masqmail's type of MTA can be called "real" MTAs—not simple forwarders nor groupware. The common substitutes for masqmail are: sendmail, exim, qmail, and postfix. These four programs are among the most popular MTAs and share between one third and half of the market.

sendmail and exim are both monolithic (like masqmail). They are about ten times larger than masqmail and thus no good substitutes.

qmail and postfix have a modular architecture which is seen to be more secure. postfix is as large as sendmail and tries to be full-featured and compatible to it. qmail is quite small, but it is outdated because its author stopped development and the community provides (for historical reasons) only patches.

Anyway, they all focus on permanent online connections.

4 masqmail's present and future

To make masqmail ready for the future, first the requirements need to be defined. It is common to split them into functional and non-functional ones. Functional requirements are for example support for special protocols, encryption, or authentication. Non-functional requirements include secu-

rity, robustness, and extendability.

The requirements are rated by their importance and by the pending work to achieve them. Large and important work tasks have a higher focus and should probably get addressed first. Low importance or few pending work lead to low focus.

Ordering the requirements by their focus results in a list of work tasks. The six topmost tasks are: encryption, authentication, security, reliability, spam handling, and extendability.

The next decision is about how to do this work. Three approaches are possible: 1) improve the existing code, 2) add wrappers and interposition filters, or 3) redesign and recreate the program. The first two can be combined well.

The pros and cons for each approach are manifold; they can not get evaluated sufficiently here. Only one view shall be given: architecture.

A program's architecture is probably its most influencing attribute. It determines many of the quality properties and influences the way further development goes. masqmail's architecture is monolithic and very interweaved. Figure 1 shows a call graph which depicts masqmail's internal structure.

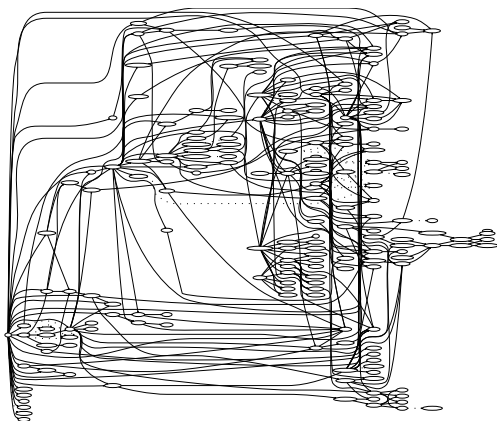


Figure 1: Call graph of masqmail

There is no single way for further development that satisfies all wishes. Thus a double-strategy is recommended. The short-term goal is to keep masqmail usable by solving the most important problems in the existing code. The long-term goal is to make masqmail ready for the future. To create a new and modern masqmail, a redesign is mandatory. The new version shall then supersede the current one.

5 Improvement plans

5.1 Based on current code

The first three work tasks (encryption, authentication, and security) can be added to current masqmail without much problems. But other work tasks require much effort this way, so they are delayed to the new design.

5.2 A new design

The new masqmail needs to be modular, for sure. Its core should be independent from the incoming and outgoing channels to allow easy adding of new protocol handlers, which will likely become important in future. Invoking external mail scanning software is important nowadays; this needs to be respected.

The proposed new design is shown in figure 2.

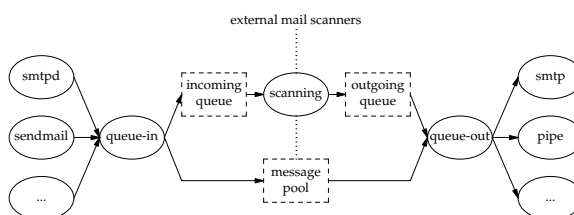


Figure 2: A new architecture for masqmail

6 Help wanted!

The here described plan to revive masqmail is only a first step. Now, volunteer developers are required to make it become reality. Programming needs to be done, documentation is important, also “marketing”, and even simply using masqmail.

I much welcome your help, advice, and comments.¹

Help masqmail!

References

- [1] Markus Schnalke. *Homepage of masqmail*. <http://prog.marmaro.de/masqmail>.
- [2] Markus Schnalke. *masqmail*. Diploma Thesis. University of Applied Sciences, Ulm, February 2009. Available online at <http://marmaro.de/docs#diploma>.

¹btw: I'm not looking for work-slaves, but for partners.