
6 qmail

In diesem Kapitel lernen Sie

- ▶ die Architektur von qmail kennen.
- ▶ die Konfiguration von qmail kennen.
- ▶ SPAM-Mails zu blockieren.
- ▶ die Konfiguration wichtiger Praxisszenarios.

6.1 Einführung

Die lange währenden Sicherheitsprobleme von `sendmail` haben für ausreichend Anreiz gesorgt, dass sich viele Administratoren auf die Suche nach Alternativen gemacht haben. Es gab zwar schon einige Alternativsysteme wie `zmailer`, `MMDF`, `smail` und `exim`, diese konnten sich aber nie so recht durchsetzen. Um 1996 beschloss deshalb Daniel J. Bernstein²¹, ein Mailsystem komplett neu zu entwickeln, um sich nicht Probleme durch eine alte Code-Basis und durch unnötige Kompatibilitäten Einschränkungen einzuhandeln. Das Ergebnis ist `qmail`.

6.1.1 Ursachen für die Probleme mit `sendmail`

Die großen Probleme mit *Berkeley* `sendmail` entstanden nicht etwa durch fehlerhafte Entwicklung (kaum eine Software hat keine Fehler), sondern weil sich durch das Design von `sendmail` die Fehler besonders nachteilig ausgewirkt haben. Der Code-Wildwuchs, der durch schnell gebrauchte Features und Bugfixes verursacht wurde, hat die Situation leider nur noch verschlimmert.

Das Design-Manko bei `sendmail` ist seine monolithische Architektur. Ein Mailsystem hat eine Vielzahl von Aufgaben zu erledigen, die in vielen Fällen völlig unabhängig voneinander sind. Die Aufgaben sind z.B.:

- Am *SMTP*-Port horchen und eingehende *SMTP*-Verbindungen behandeln.
- Mails entsprechend Ihres Empfängers in Warteschlangen sortieren
- Mails an lokale Empfänger zustellen

²¹Professor für Mathematik an der Universität Chicago, hat für diverse Dienste neue Implementierung geschaffen, z.B. auch `djbdns`

qmail

- Mails an externe Mail-Server verschicken
- Adressen parsen und ggf. umschreiben
- ...

Einige dieser Aufgaben verlangen besondere Privilegien (i.A. *root*-Rechte). *sendmail* nun erledigt alle diese Aufgaben mit einem einzigen Executable, welches zu diesem Zweck das *setuid*-Bit für *root* gesetzt haben musste. Das bewirkt, dass *sendmail* immer mit *root*-Privilegien ausgeführt wird, auch wenn ein normaler Benutzer das *sendmail*-Programm aufruft. Fehler im *sendmail*-System haben sich daher besonders fatal auswirken können.

Fragen

1. Wodurch charakterisiert sich eine monolithische Architektur bei einer Software-Anwendung?
2. Was sind die Nachteile einer solchen Architektur?
3. Warum ist das bei *sendmail* besonders schlimm?

6.1.2 Der Ansatz von qmail

Genau diesen Fehler wollte D. J. Bernstein unbedingt vermeiden, und hat daher bei der Entwicklung von *qmail* auf zwei Dinge besonderen Wert gelegt: Sicherheit und Performance. D. J. Bernstein hat dabei die Aufgaben des Mailers klar getrennt und stellt für jede Aufgabe ein eigenes Executable bereit. Außerdem sind für den Betrieb von *qmail* sieben verschiedene Benutzerkennungen erforderlich, so dass jeder Teil der Mailbehandlung in einer eigenen *sandbox*²² ablaufen kann. Diese Maßnahme erscheint heute allerdings etwas übertrieben. Da auf Kompatibilität kein Wert gelegt wurde, ist *qmail* zunächst gänzlich anders als *sendmail* zu konfigurieren und zu betreiben. (Allerdings gibt es einige Zusatzpakete, die einem die Migration erleichtern.) Die Details werden nun im Folgenden beschrieben. Dieser radikale Ansatz hat nicht überall Zustimmung gefunden: Wietse Venema hat mit seinem Mail-System *postfix* (ehem. *vmailer*) gezeigt, dass man durchaus ein sicheres, schnelles Mailsystem von Grund auf neu entwickeln kann, ohne dass es völlig inkompatibel ist.

²²Eine Umgebung, die der Prozess nicht verlassen kann, und die so den potentielle Schaden durch einen Bug-(Exploit) begrenzt.

Fragen

1. Ist qmail ein System monolithischer Architektur?
2. Warum ist das so?

6.2 Installation von qmail

qmail ist frei verfügbare *Open Source*-Software und im Quellcode sowie für manche Systeme als vorkompiliertes Paket erhältlich. Das qmail-Paket lässt sich auf allen gängigen Unix-Systemen problemlos übersetzen und installieren. Die konkrete Vorgehensweise ist detailliert beschrieben und bereitet keine Schwierigkeiten. Allerdings ist das Standard-Verzeichnis, in welches qmail installiert wird, nicht unbedingt im Sinne des Admins: qmail installiert sich (wenn nicht anders angegeben) nach `/var/qmail`. Nun ist aber das `/var`-Dateisystem üblicherweise für Daten (mit variierendem Charakter) und weniger für Executables und Konfigurationsdateien vorgesehen. Viele Admins verlegen daher qmail lieber nach `/usr/local/qmail`, `/opt/qmail` oder `/home/qmail`. D.J. Bernstein gibt als Grund die Tendenz an, dass `/usr/local` et al. häufig von einem NFS-Server kommen (also als Netzlaufwerk von einem zentralen Fileserver), und die Benutzer-IDs/Gruppen-IDs der Dateien daher nicht mehr unbedingt übereinstimmen. Da aber die Komponenten von qmail fast nur von den entsprechenden qmail-Benutzerkonten verwendet werden können, ist eine konsistente Übereinstimmung dieser Konten und Gruppen auf den Systemen zwingend notwendig. Insofern ist dieser Einwand berechtigt. Man kann allerdings dagegen halten, dass ein vernünftiger Admin seinen Mailserver als möglichst unabhängiges System betreibt, das insbesondere nicht von Netzlaufwerken abhängt.

6.2.1 Zusätzlich erforderliche Pakete

tcpserver Um qmail zu betreiben, wird unbedingt das Paket `ucspi-tcp` (siehe Querverweise in 6.12) empfohlen. Um E-Mails zu empfangen, muss ein Serverdienst auf dem TCP-Port für SMTP (*Simple Mail Transfer Protokoll*), üblicherweise Port 25, Anfragen entgegennehmen. Die Komponente von qmail, die für das SMTP-Protokoll zuständig ist, also die SMTP-Kommandos bearbeitet und beantwortet, der **qmail-smtpd**, kann von sich aus nicht auf dem SMTP-Port „horchen“. Der Grund dafür ist, dass der SMTP-Port mit der Nummer 25 ein so genannter *privilegierter Port* ist (wie alle Ports kleiner 1024), und das System daher *root*-Privilegien verlangt. Der **qmail-smtpd** soll diese nicht haben, und erhält daher seine Daten durch die Standardeingabe. Es ist also noch ein Programm erforderlich, welches die Anfragen aus dem Netz entgegen nimmt und direkt an den **qmail-smtpd** weiterleitet. Frühere Versionen von qmail verwendeten den **inetd**, der allgemein auf

UNIX-Systemen genau solche Aufgaben übernehmen kann. Aus Performancegründen empfiehlt D.J.Bernstein aber das Programm **tcpserver**, welches Teil seines `ucspi-tcp`-Paketes ist. Mit Hilfe von **tcpserver** lassen sich weiterhin einfach Zugriffbeschränkungen implementieren (ähnlich `tcp-wrappers/tcpd/libwrap`).

Fragen

1. Wozu dient **tcpserver**?

dot-forward `qmail` kennt keine `.forward`-Dateien für benutzerdefinierte Weiterleitung, sondern nur `.qmail`-Dateien. Bei einer Migration von `sendmail` mit vielen Benutzern empfiehlt sich das `dot-forward`-Paket, welches die Weiterbenutzung von `.forward` ermöglicht.

fastforward Diese Paket ermöglicht die Verwendung eines binären Datenbankformats für systemweite *Aliases*. Das Quell-Format ist kompatibel zu `sendmail's /etc/aliases`, so dass bestehende `Sendmail-Alias-Datenbanken` (mit einigen Einschränkungen) übernommen werden können.

Fragen

1. Kann man bei der Migration von `sendmail` existierende `/etc/aliases` und `.forward` Dateien weiterverwenden?
2. Was ist dazu nötig?

6.3 Die Komponenten von `qmail`

Dieser Abschnitt beschreibt die Komponenten und Kommandos, sowie die Konfigurationsdateien.

6.3.1 `qmail`-Kommandos, mit denen man zu tun hat

Die folgenden Kommandos, die Teil von `qmail` sind, werden öfter gebraucht, bzw. mit ihnen hat man eher zu tun als mit einigen anderen (die danach erklärt werden).

qmail-smtpd Der **qmail-smtpd** ist für den *SMTP* (bzw. *ESMTP*)²³-Dialog mit einem entfernten Mail-System verantwortlich. Er nimmt Kommandos und Daten per *SMTP* entgegen und bearbeitet sie entsprechend. Nachrichten, für die das Mailsystem zuständig ist (entweder für lokale Benutzer, oder zulässige Weiterleitungs/Relay-Ziele), werden an die zentrale Warteschlange mit Hilfe von **qmail-queue** übergeben. Anfragen, für die das System nicht zuständig ist, werden abgewiesen. Der **qmail-smtpd** muss von einem Programm gestartet werden, das ihm die Daten von einer Netzverbindung übergeben kann. Empfohlen wird dafür, wie gesagt, **tcpserver**.

qmail-inject und **sendmail** **qmail-inject** ermöglicht das Einsortieren einer fertigen E-Mail in die zentrale Warteschlange zur Weiterverarbeitung. **qmail-inject** repräsentiert dabei den lokalen *MTA*, Mail Transport Agent. **qmail-inject** kann über das quasi-Standard-**sendmail**-Interface angesprochen werden, d.h. es gibt ein Programm namens **sendmail** im `qmail`-Paket, welches entsprechend dem *Berkeley*-**sendmail** aufgerufen werden kann, aber die Parameter entsprechend modifiziert an **qmail-inject** übergibt. So können alle E-Mail-Client-Programme (*MUAs*, Mail User Agents) weiterhin ein **sendmail**-Programm zum versenden von E-Mails verwenden. Siehe dazu auch 6.11.1.

qmail-qread und **qmail-qstat** Diese beiden Kommandos können dafür verwendet werden, den aktuellen Status der Warteschlange anzuzeigen. Dabei ist **qmail-qread** die ausführliche Variante, die zu jeder Nachricht in der Warteschlange einiges an Informationen anzeigt. **qmail-qread** wird auch durch das **sendmail**-Interface aufgerufen (**sendmail -bp** bzw. **mailq**). **qmail-qstat** ist die kurze Variante, die nur die Anzahl der Nachrichten in der Warteschlange ausgibt.

qmail-showctl Damit lässt sich die aktuelle `qmail`-Konfiguration analysieren und anzeigen. Auf etwaige Probleme wird hingewiesen.

qmail-start Das Kommando, mit dem der `qmail`-Auslieferungsdienst gestartet werden sollte. Die Dämonen **qmail-send**, **qmail-lspawn**, **qmail-rspawn** und **qmail-clean**, die dafür erforderlich sind, werden unter den erforderlichen Benutzerkennungen gestartet. Die Ausgaben von **qmail-send** werden dabei an **qmail-start** umgeleitet.

splogger Ein **syslog**-Interface von `qmail`. Mit Hilfe von **splogger** können diverse Ausgaben von z.B. **qmail-send**, bzw. **qmail-start**, an den **syslog**-

²³Das (Extended) Simple Mail Transport Protocol; das Protokoll zum Übertragen von E-Mails im Internet

Dienst übergeben werden. Die *Log-Facility* und *Priorität* kann dabei angegeben werden.

Fragen

1. Mit welchem Kommando kann man sich die Konfiguration anzeigen lassen?
2. Wie lassen sich die E-Mails in der Warteschlange anzeigen?
3. Wie wird eine neue E-Mail in die Warteschlange eingefügt?
4. Womit startet man den Auslieferungs- und Versende-Dienst?

6.3.2 Kommandos/Dienste, die nur implizit verwendet werden

qmail-queue **qmail-queue** liest eine Nachricht und die zugehörige *Envelope*-Information (der tatsächliche Absender und Empfänger, unabhängig davon, was in den *From:*, *To:*, *Cc:*, ... *Headern* steht) ein, und stellt die Nachricht dann in die entsprechende Warteschlange zur Weiterverarbeitung von **qmail-send**.

qmail-send und **qmail-clean** **qmail-send** bearbeitet die Nachrichten in der Warteschlange, Mails an lokale Empfänger werden an **qmail-lspawn**, an entfernte Empfänger an **qmail-rspawn** übergeben. Kann eine Nachricht nicht zugestellt werden, so verbleibt sie zunächst in der Warteschlange. Durch ein *ALRM*-Signal an den **qmail-send**-Prozess können alle Mails in der Warteschlange zur sofortigen Auslieferung markiert werden. **qmail-send** berücksichtigt alle Konfigurationsdateien, die mit der Auslieferung von E-Mails zu tun haben. **qmail-send** säubert die Warteschlange außerdem mit Hilfe von **qmail-clean**.

qmail-rspawn und **qmail-lspawn** Diese beiden Kommandos sind nur für die koordinierte Abwicklung der Auslieferung an lokale oder entfernte Empfänger verantwortlich. Sie erhalten die Auslieferungskommandos von **qmail-send**, und sorgen dann dafür, dass die Nachrichten von **qmail-remote**, bzw. **qmail-local** ausgeliefert werden.

qmail-remote nimmt eine E-Mail und schickt sie an einen oder mehrere Empfänger eines Zielsystems. Der Zielrechner ist dabei entweder direkt angegeben, oder wird über den *MX-Record*²⁴ im *DNS* der Zieldomain bestimmt. **qmail-remote** spricht

²⁴Dabei handelt es sich um den so genannten *Mail-Exchanger*, ein spezieller Eintrag im *Domain Name Service*, um Mail-Server für eine Domain oder Hosts zu kennzeichnen.

dabei mit dem Zielrechner über das *SMTP*-Protokoll, d.h. dass auf dem Zielsystem ein *SMTP*-Dämon laufen muss. `qmail-remote` berücksichtigt alle Konfigurationsdateien, die die Zielsysteme beeinflussen können, zum Beispiel, wenn alle Mails über ein zentrales *Mailrelay* (ein Mailserver, über den alle ausgehenden Mails verschickt werden müssen, da kein allgemeiner Zugang möglich ist) verschickt werden müssen. Dies steht dann in `smtproutes`.

`qmail-local` liefert eine E-Mail an einen lokalen Benutzer aus, dabei kann einiges den tatsächlichen Vorgang beeinflussen. Die Art der Auslieferung wird dabei indirekt über Parameter an `qmail-start` bestimmt.

`qmail-getpw` wird von `qmail-lspawn` verwendet, um den zugehörigen Benutzer zu einer E-Mail-Adresse zu ermitteln. Dabei stützt sich `qmail-getpw` unter anderem auf die `getpwnam()` Funktion, also auf die Bibliotheksfunktionen zum Zugriff auf die Benutzerinformation in der `/etc/passwd` und auf den `qmail-users`-Mechanismus (siehe 6.6.3), falls er konfiguriert ist.

Fragen

1. Welcher Dienst sortiert E-Mails in die Warteschlange ein?
2. Welches Programm stellt den lokalen Empfänger einer E-Mail fest?
3. Über welche Methoden kann `qmail-remote` das entfernte Zielsystem bestimmen?

6.3.3 Kommandos/Dämonen für besondere Aufgaben

Die folgenden Kommandos werden nicht zwingend benötigt, können aber in gewissen Konfigurationen hilfreich sein, bzw. für bestimmte Aufgaben erforderlich sein.

`qmail-newmrh` Die Kontrolldatei `morercpthosts` ermöglicht zusätzliche Rechner anzugeben, für die der Mailserver E-Mails empfangen soll (siehe 6.3.4). Damit neue Einträge auch benutzt werden muss `qmail-newmrh` aufgerufen werden.

`qmail-newu` Will man den erweiterten Alias-Mechanismus (`qmail-users`, siehe 6.6.3) verwenden, so müssen die Einträge initialisiert, bzw. in ein Binärformat konvertiert werden, damit sie verwendet werden können. `qmail-newu` erledigt diese Aufgabe, muss also jedes Mal aufgerufen werden, wenn sich an diesen Daten was geändert hat.

qmail

qmail-pw2u **qmail-pw2u** erzeugt eine initiale Adresse-Benutzer-Zuordnung für den erweiterten Alias-Mechanismus *qmail-users* (6.6.3) aus den Daten der `/etc/passwd`.

qmail-pop3d Dies ist der POP3-Server des *qmail*-Paketes. Er ermöglicht Benutzern den Zugriff auf Ihre Mailboxen über das POP3-Protokoll. Der **qmail-pop3d** wird von **qmail-popup** gestartet.

qmail-popup Damit wird die Benutzerauthentifizierung für POP3-Zugang geregelt. **qmail-popup** liest den Benutzernamen und das Passwort, und übergibt diese an ein externes Programm zur Prüfung. So ist maximale Flexibilität zur Authentifizierung (z.B. auch über *ldap*) gewährleistet. Ist die Authentifizierung erfolgreich, wird der **qmail-pop3d** gestartet und die Verbindung daran übergeben. Der **qmail-popup** wird normalerweise vom **(x)inetd** gestartet, muss also in die `/etc/(x)inetd.conf` eingetragen sein.

qmail-qmqpc und **qmail-qmqpd** Dies sind der Client und Server für das *Quick Mail Queueing Protocol - QMQP*. Das ist ein Protokoll, mit dem in einer Netzwerkinstallation mit einem Mail-Server die Clients mittels ganz simpler *MTAs* (Mail Transport Agents) ihre E-Mails über den Server verschicken zu können. Auf dem Server muss der **qmail-qmqpd** laufen, der E-Mails in Empfang nimmt und in die ausgehende Warteschlange (mittels **qmail-queue**) stellt.

Auf dem client kann dann eine so genannte Mini-*qmail*-Installation vorgenommen werden, bei der unter anderem das **qmail-queue**-Programm durch den **qmail-qmqpc** ersetzt wird. **qmail-qmqpc** hat die gleiche Schnittstelle wie **qmail-queue**, schickt die E-Mail aber mittels *QMQP* an den **qmail-qmqpd** auf dem Server.

qmail-qmtpd Ein Dämon für das *Quick Mail Transfer Protocol, QMTP*. *QMTP* ist ein, von D. J. Bernstein entwickelter Ersatz für das SMTP-Protokoll, welches unter bestimmten Umständen Vorteile gegenüber SMTP haben kann. Allerdings wird es nur von *qmail* unterstützt. Der **qmail-qmtpd** wird genauso verwendet wie der **qmail-smtpd**.

tcp-env Viele *qmail*-Server-Dienste benötigen Information über die Natur der übergebenen Netzwerkverbindung (*TCP Connection*). Bei der gewünschten Information handelt es sich um die reinen Verbindungsdaten wie entfernter Hostname, IP und Port, und entsprechend lokaler Hostname, IP und Port. Diese Information erwarten die Dienste über Umgebungsvariablen. Wenn der Dienst nun beispielsweise per **inetd**

gestartet wird, muss dafür gesorgt werden, dass die entsprechenden Umgebungsvariablen auch gesetzt sind. Dazu dient das Programm `tcp-env`. Es sitzt praktisch als Filter zwischen der Netzwerkverbindung, findet die relevante Information, setzt die Umgebungsvariablen und startet dann den entsprechenden Dienst. Wird `tcpserver` verwendet ist das nicht nötig, da `tcpserver` die entsprechenden Umgebungsvariablen bereits selbst bereitstellt.

Fragen

1. In welchen Fällen muss `qmail-newu` ausgeführt werden?
2. Wozu dient `qmail-popup`?
3. Wie kann man die Charakteristika einer eingehenden Netzwerkverbindung an einen `qmail`-Dienst weitergeben, z.B. an `qmail-smtpd`?

6.3.4 Konfiguration von `qmail`

`qmail` wird nicht (wie `sendmail`) durch eine einzige, möglicherweise sehr komplizierte Datei konfiguriert, sondern durch eine größere Anzahl von Konfigurationsdateien, wobei davon nur eine einzige zwingend erforderlich ist. Die Konfigurationsdateien liegen alle in:

```
`${QMAIL_HOME}/control/
```

Im Folgenden werden nun die wichtigsten Konfigurationsdateien vorgestellt und erläutert, wie sich welche Funktionalität erreichen lässt. Als Referenz wird ausdrücklich auf die Manual-Seiten (siehe Querverweise in 6.12) von `qmail` hingewiesen, welche die Konfiguration detailliert beschreiben, `qmail-control(5)` dient dabei als Übersicht.

me Dies ist die elementare Konfigurationsdatei. Sie ist die einzige, die zwingend vorhanden sein muss und enthält lediglich den voll-qualifizierten Hostnamen (FQHN) des Mail-Servers, z.B. `mail.mydomain.com`

Der Inhalt von `me` bestimmt im weiteren die Voreinstellungen einer ganzen Reihe weiterer Konfigurationsdateien.

badmailfrom E-Mails von Sendern in dieser Liste werden nicht angenommen. Dies ist ein primitives Mittel zur *SPAM*-Kontrolle. Siehe auch 6.4.2.

concurrencylocal und concurrencyremote Diese Parameter dienen der Performance-Optimierung. Sie geben die maximal gleichzeitig stattfindenden Auslieferungsversuche für lokale und entfernte Ziele an.

defaultdomain und defaulthost Der Inhalt von `defaultdomain` wird von `qmail-inject` an jeden Rechnernamen von zu verschickenden E-Mails angehängt, sofern die Rechnernamen keinen Punkt enthalten (also nicht *fully qualified* sind). Dies gilt auch für `defaulthost`, falls dieser keinen Punkt enthält.

`defaulthost` wird an jede E-Mail-Adresse ohne Rechner/Domain-Namen angehängt (also Adressen ohne „@“). Es muss sich dabei nicht um einen echten Rechnernamen handeln. Man kann auch seine Domain angeben, z.B. `mydomain.com`.

Üblicherweise betreffen diese Parameter nur Absenderadressen.

databytes Damit ist es möglich, die Größe eingehender E-Mails zu begrenzen. Die Voreinstellung ist 0, das heißt, es gibt keine Größenbegrenzung.

envnoathost Domainname für Empfängeradressen ohne „@“-Symbol. Als Voreinstellung gilt der Inhalt von `me`.

locals Liste aller Domains, für die der Mailserver E-Mails ausliefert. Eine E-Mail an `user@domain` wird als lokal betrachtet, wenn `domain` in `locals` enthalten ist.

idhost Rechnername, der für die Erzeugung der Message-ID einer E-Mail verwendet werden soll. Bei einer Mini-qmail-Installation ist das für die Client-Rechner nötig.

plusdomain Damit lassen sich Hostnamen mit dieser Domain ergänzen, falls sie auf ein Plus-Zeichen „+“ enden. Dies gilt ebenfalls für `defaulthost`.

qmcpserver Nur für einen Client einer Mini-qmail-Installation. Die Datei enthält eine Liste aller Server, die per QMQP E-Mails entgegennehmen können. Siehe Abschnitt 6.5.3.

rcpthosts und morercpthosts `rcpthosts` enthält eine Liste aller Domains, für die dieser Server E-Mails empfangen soll. Mails an Empfängerdomains, die nicht in dieser Liste aufgeführt sind, werden zurückgewiesen (falls die `rcpthosts` existiert). Für große Mailserver, die für viele Hosts/Domains Mails empfangen, empfiehlt es sich zusätzlich `morercpthosts` zu verwenden. Bei Änderungen daran kann mit Hilfe von `qmail-newmrh` die Liste für `qmail` dynamisch aktualisiert werden.

Achtung: Eine `rcpthosts` sollte auf alle Fälle angelegt werden, da der Mailserver sonst Mails für alle Domains akzeptiert und somit als offenes Relay arbeitet!

