NewElements

IT-Schulungen.com

Postfix

Dr.sc.nat. Michael J.M. Wagner*

Revision 1.14

*michael@wagnertech.de

Inhaltsverzeichnis

IT-Schulungen.com
Portfolio

IT-Schulungen.com ist eines der führenden, herstellerunabhängigen Seminarportale von Schulungen rund um die Informationstechnologie (IT) und das IT-Management. Seit über 15 Jahren ist IT-Schulungen.com eine anerkannte Anlaufstelle für viele Unternehmen und Behörden, wenn es um die Durchführung von DACH-weiten Schulungen geht.

- Applikationsserver / Middleware
- Business Intelligence
- Business-Skills und Führung
- Cloud
- CRM
- Datenbanken
- eBusiness
- ERP-Systeme
- IT Management
- IT-Recht / Lizenzierung
- ITIL
- Mobile
- Multimedia
- Office
- Open Source
- Portale
- SAP®
- Security
- Serversysteme
- Softwareentwicklung
- Systemmanagement

Zertifizierungen & Partnerschaften

1 Grundlagen

1.1 Einführung in Postfix

> Postfix ist ein sogenannter Message Transport Agent (MTA) oder kürzer: ein Mailserver. Mithilfe des Simple Mail Transfer Protocol (SMTP) transportiert er Nachrichten von einem Mail User Agent (MUA) – also von E-Mail-Programmen wie Mutt, Outlook oder Apple Mail – zu einem anderen Mailserver. Zu seinen Aufgaben zählt auch, eine Nachricht von einem entfernten Mailserver anzunehmen und diese lokal in der Mailbox eines Anwenders abzulegen oder sie an andere MTAs weiterzuleiten.[1]

Darüber hinaus können MTAs auch ausschließlich zur Weiterleitung verwendet werden. Wird Postfix in einer Ubuntu/Mint-Distribution installiert, so bietet die Installationsroutine folgende Nutzungsvarianten an:

- Internet-Site: Empfang/Senden über SMTP
- Internet mit Smarthost: Empfang über SMTP, Senden über Smarthost (=Relayhost)
- Satellitensystem: nur Weiterleitung über Smarthost (=Relayhost), keine lokalen Benutzer
- Nur Lokal: nur Empfang

Aufgabe:
Starten Sie die VM und installieren sie postfix als „Internet-Site“.

Dient der MTA als Endpunkt für Benutzer-Mails, so wird der MTA um einen POP- oder/und IMAP-Server ergänzt. Bei Postfix (MTA) ist dies Dovecot (POP/IMAP). POP/IMAP sind die Protokolle, die die Nachricht von der Mailbox zum MUA des Anwenders transportieren.

Für die Sicherheit von Servern werden heute Firewalls eingesetzt.

> Aber damit ist es bei E-Mails nicht getan! Firewalls kontrollieren zwar die Verbindungen zwischen Geräten, aber sie verfügen typischerweise nicht über die Fähigkeit, den Inhalt, der dabei ausgetauscht wird, zu kontrollieren. Im Regelfall erkennen sie lediglich die Hostrechner, die Ports und das Transport-Layer-Protokoll, das zur Kommunikation verwendet wird, aber nicht den Inhalt.
>
> Doch genau darauf kommt es bei Mailservern heute besonders an. Inhaltsanalyse ist eine komplexe Aufgabe und verlangt den Einsatz spezialisierter Software. Deren Aufgabe ist es, den transportierten Inhalt zu analysieren und zu bestimmen, ob er nützlich oder schädlich ist. Genau darin liegt die zentrale Aufgabe heutiger MTAs.[2]

[1] Hildebrandt (2008): S. 3.
[2] Hildebrandt (2008): S. 3f.

1.2 SMTP-Kommunikation im Überblick

Der Transport einer E-Mail findet immer zwischen einem Client (er versendet die E-Mail) und einem Server (er nimmt die E-Mail an) statt. Für das weitere Verständnis eines Mailservers ist es wichtig, sich vor Augen zu führen, dass auch ein Mailserver als Client agieren kann – nämlich dann, wenn er eine E-Mail zu einem anderen Server transportiert.

Wenn ein Client eine E-Mail an einen Server übergibt, dann werden immer die folgenden Schritte absolviert:

1. Client verbindet sich mit Server.
2. Server und Client stellen sich einander vor.
3. Client erzählt dem Server, was er veranlassen will.
4. Server prüft, ob der Client (oder Absender) autorisiert ist.
5. Server übernimmt E-Mail (oder weist sie ggf. auch ab).
6. Server bestimmt den Weg zum Empfänger.
7. Server transportiert die E-Mail näher zum Empfänger.

Je Nachdem, welches Protokoll – SMTP oder ESMTP – zur Anwendung kommt, werden dabei weniger (SMTP-Protokoll) oder mehr (ESMTP-Protokoll) Informationen in der Kommunikation zwischen Client und Server ausgetauscht.[3]

Aufgabe:

Klären Sie, welche IP zu welcher VM gehört und ergänzen Sie /etc/hosts entsprechend. Da Postfix Systemparameter zwischenspeichert, muss danach die VM neu gestartet werden. Führen sie im Terminal Ihrer Instanz folgende SMTP-Kommunikation durch:

```
$telnet localhost 25
HELO postfix2
MAIL FROM:<postfix@postfix2>
RCPT TO:<postfix@postfix1>
DATA
--- Some Text ---
.
QUIT
```

Stellen Sie auch Ihrem Kollegen* eine Nachricht zu.

[3] Hildebrandt (2008): S. 8.

1.3 Systemvorbereitung

Für den reibungslosen Betrieb eines Mailservers ist oft die richtige Serveridentität wichtig. Diese wir von folgenden Stellen beeinflusst:

- `/etc/hostname`: Überprüfbar mit: `hostname`
- `127.0.1.1`-Eintrag in `/etc/hosts`: Überprüfbar mit: `hostname -f`
- `myhostname` in `/etc/postfix/main.cf`: Überprüfbar mit: `postconf myhostname`
- Falls der Server für seine Benutzer Mails annehmen soll, muss der Hostname in `/etc/postfix/main.cf` auch in `mydestnation` eingetragen sein.
- `myorigin` wird bei der Paketinstallation mit dem Inhalt von `/etc/mailname` belegt, wird an alle Adressen ohne Serverangabe (@...) angehängt.

1.4 Mailserver für eine Domain einrichten

Standardmäßig leitet Postfix nur Mails, die aus der eigenen Domain stammen, weiter (*relay*).

Aufgabe:

Verschicken Sie eine Mail von postfix1 an user1@postfix2.

- Konfigurieren Sie dazu postfix1 so, dass er postfix2 als relayhost verwendet. Dazu muss der entsprechende Eintrag in `/etc/postfix/main.cf` gesetzt und postfix neu gestartet werden.
- Damit die Einträge von `/etc/hosts` von postfix überhaupt berücksichtigt werden, muss in `/etc/postfix/main.cf` der Eintrag `smtp_host_lookup = native,dns` ergänzt werden.
- Bei der Instanz postfix2 muss der Parameter `mynetwork` entsprechend erweitert werden (z.B. `192.168.1.0/24`)
- Verfolgen Sie die einzelnen Schritte in den jeweiligen Logdateien (`/var/log/mail.log`)
- • Betrachten Sie sich den Quelltext der empfangenen Mail in `/var/mail/user1` auf postfix2.

Anmerkung: Für einen Server lassen ich auch mehrere relay hosts definieren. Dazu gibt es die Konfigurationseinträge `relay_recipient_map` und `relay_domain`.

1.5 DNS-Administration

Für den Betrieb eines Mailservers sind korrekte DNS-Einträge nötig. Von Bedeutung sind:

- MX-Record: Er benennt den Server (FQDN), auf dem Postfix läuft.
  Beispiel: MX von `example.com` zeigt auf `mail.example.com`
- Ggf. SPF/DKIM-Einträge im TXT-Record (s. Kap. 3)

- A/AAAA-Record des Mailservers
- Reverse-DNS des Mailservers

2 Inhalte kontrollieren

2.1 Das Postmaster-E-Mail-Einmaleins

Abbildung 1 zeigt die Entsprechungen von physischer zu elektronischer Post.[4]

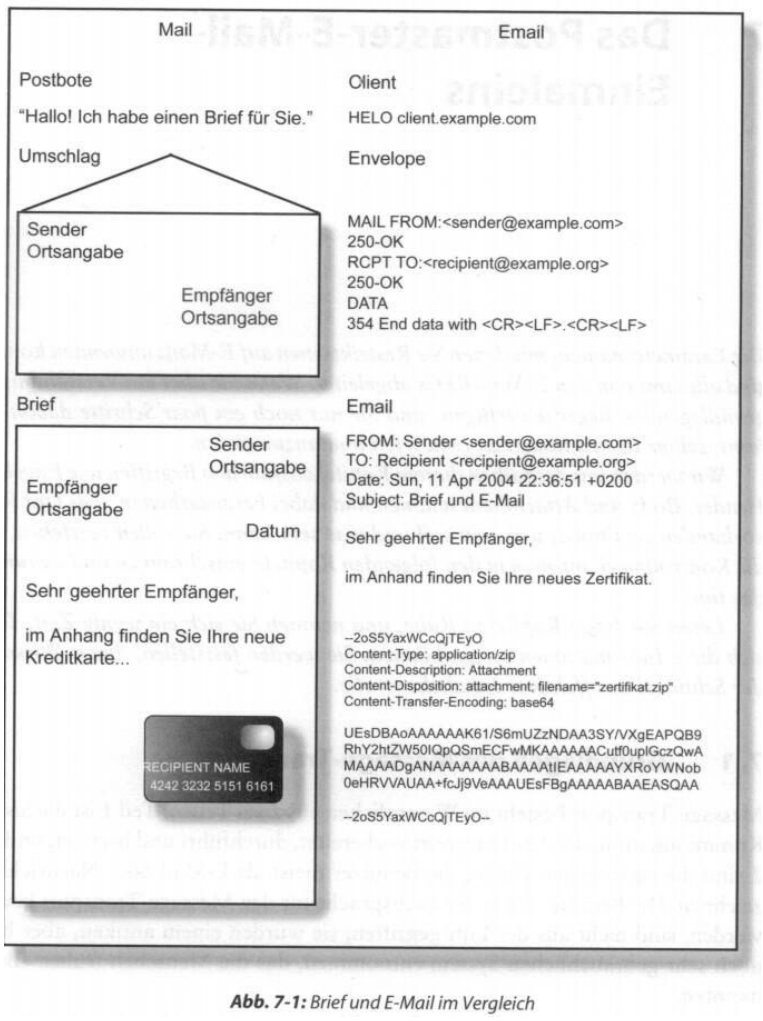

Abb. 7-1: *Brief und E-Mail im Vergleich*

Abbildung 1: Brief und E-Mail im Vergleich

Die Begriffe Client, Envelope, Header, Body, Attachment erscheinen wieder als Prüfpunkte in den entsprechenden für Restriktionen (`smtpd_..._restrictions`), Prüfungen (`smtpd_..._checks`) und Filter (`smtpd_..._filter`). Letztere sind Prüfungen, die an externe Programme delegiert werden. Schlägt eine Restriktion zu, so wird die Mail gar nicht angenommen.

Zur Beachtung: FROM/TO tritt zwei mal auf: Einmal im Umschlag, einmal im Kopf des Briefes.

[4] Hildebrand (2008): S. 130.

2.2 Wie interne Message Transfer Restrictions funktionieren

Abbildung 2 zeigt, wie Restriktionen zu den Prüfpunkten abgearbeitet werden.

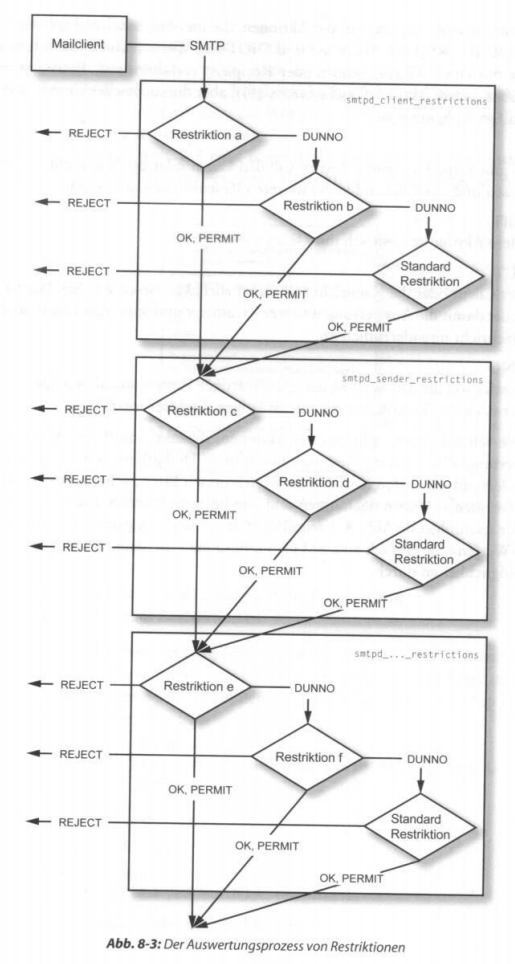

Abbildung 2: Der Auswerteprozeß von Restriktionen

Ist eine Prüfung innerhalb eines Prüfpunkts anwendbar (*reject* oder *permit*), so wird aus der Abarbeitung der entsprechenden Prüfpunkts ausgestiegen. Ist die Regel nicht anwendbar (*dunno = don't know*), wird die nächste Regel des Prüfpunkts abgearbeitet. Ist keine Regel innerhalb eines Prüfpunkts anwendbar, so wird die Standardregel des Prüfpunkts angewendet. Wird ein Prüfpunkt mit *reject* verlassen, so wird die Mail nicht angenommen, bei permit wird zum nächsten Prüfpunkt gesprungen.

Folgende Prüfpunkte sind im Postfix implementiert:

- `smtpd_client_restrictions` (default: empty)
- `smtpd_helo_restrictions` (default: empty)
- `smtpd_sender_restrictions` (default: empty)
- `smtpd_recipient_restrictions` (default: see postconf -d output)
- `smtpd_data_restrictions` (default: empty)

- `smtpd_end_of_data_restrictions` (default: empty)
- `smtpd_etrn_restrictions` (default: empty) (ein historisches „Abholprotokoll")
- `smtpd_relay_restrictions` (default: `permit_mynetworks, permit_sasl_authenticated`)

2.3 Interne Message Transfer Restrictions anwenden

Restriktionen lassen sich testweise auch simulieren. Dazu stellt man vor das `reject_...` ein `warn_if_reject`.

RFC-Konformität

Die RFC-Konformität lässt sich mit folgenden Maßnahmen sicherstellen:

- `HELO/EHLO` mit FQDN-Name
- Nur gültige Zeichen im Hostnamen
- Komplette Adresse im Sender
- Nur Mails von gültigen Domänen
- Nur Mails für gültige Domänen
- Mails an vorgeschriebene Empfänger müssen angenommen werden: `postmaster, abuse, hostmaster, wel`

Dies kann mit folgender Restriktion erreicht werden:

```
smtpd_recipient_restrictions =
 reject_non_fqdn_recipient
 reject_non_fqdn_sender
 reject_unknown_sender_domain
 reject_unknown_recipient_domain
 permit_mynetworks
 check_recipient_access hash:/etc/postfix/roleaccount_exceptions
 permit
```

Die Datei `roleaccount_exceptions` (sie muss vor der Verwendung mit `postmap` übersetzt werden) hat dabei folgenden Inhalt:

```
postmaster@    OK
abuse@         OK
hostmaster@    OK
webmaster@     OK
```

Die entsprechenden Accounts müssen zusätzlich in `/etc/aliases` angelegt sein.

Aufgabe:

- Ergänzen Sie diese Restriktionen. Damit werden aber die Mails von anderen Postfixinstenzen unseres Netzes abgelehnt.

- Ergänzen Sie `warn_if_reject`, damit in unserem Testnetzwerk wieder Mails von gesendet werden können.

Beispiele für Anti-Spam-Maßnahmen

> Spam ist deshalb so weit verbreitet, weil die Kosten für den E-Mail-Versand im Vergleich zum Postweg sehr gering sind, und gleichzeitig eine exponentiell größere Anzahl an Empfängern erreicht werden kann. Wenn Sie mit effektiven Maßnehmen das Versenden und Zustellen von Spam erschweren, werden weniger Empfänger erreicht, die Kosten steigen, und Spam wird dadurch weniger rentabel.[5]

Um Falschangaben im `HELO/EHLO`-Kommando entlarven, kann ein `check_helo_access` in die `smtpd_..._restricti` aufgenommen werden:

```
check_helo_access hash:/path/to/helo_checks
```

Für die Hashmap muss die entsprechende Hash-DB generiert werden. In der Datei `helo_checks` können Clients zurückgewiesen werden, die sich mit Ihren Serverdaten ausgeben:

```
my.host.name 550 Don't use my hostname
```

Folgende telnet-Kommunikation kann das Ergebnis verifizieren:

```
$ telnet MYHOSTNAME 25
HELO MYHOSTNAME
MAIL FROM: sender@MYHOSTNAME
RCPT TO: recipient@MYHOSTNAME
QUIT
```

Aufgabe:

Führen Sie diesen Check ein und prüfen Sie ihn über den entsprechenden `telnet`-Dialog.

Interne Message Transfer Restrictions können durch die Anweisung `FILTER` in einen externen Filter umgeleitet werden (s. Abschnitt 2.6).

Bounces (Rückweisungen) von fehlerhaften Mails müssen gemäß RFC an den Sender rückgesendet werden. Was ein Spamfilter dennoch leisten kann, ist Rückweisungen an mehrer Sender (die es ja nicht geben kann) abzuweisen.

Eine *Bounce*-Mail ist eine automatische Antwort über die Unzustellbarkeit einer Mail. Hat ein *Relay*-Server eine Mail angenommen, kann sie aber nicht weiter zustellen, sendet dieser eine Bounce-Mail an den Sender (s. Abb. 3.[6]

Bouncemails haben üblicherweise eine leere Sender-Angabe. Daher soll der Mailserver Mails mit leeren Sender-Angaben grundsätzlich akzeptieren. Dieses Soll-Verhalten nutzen gelegentlich

[5] Hildenbrand (2008): S. 150.
[6] `http://en.wikipedia.org/wiki/Bounce_message`

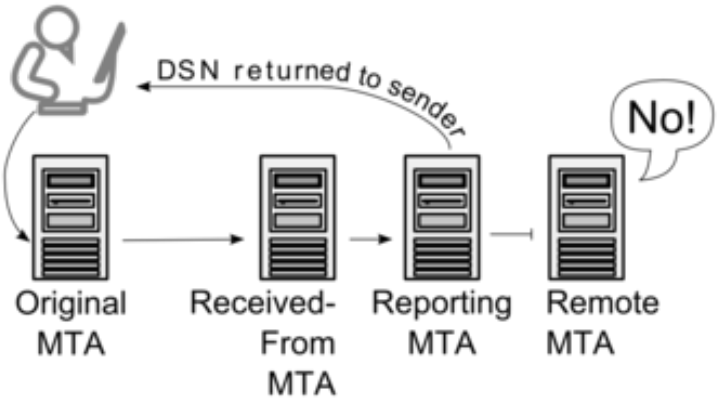

Abbildung 3: Der Auswerteprozeß von Restriktionen

Spammer aus um die Spamfilter zu umgehen. Hat eine solche Mail aber mehrere Empfänger, so kann das nicht wirklich ein echter Bounce sein und soll daher verworfen werden. Dies lässt sich mit der Restriktion `reject_multi_recipient_bounce` aktivieren. Diese Prüfung kann erst am DATA-Prüfpunkt erfolgen, da hier erst alle Empfänger übermittelt sind.

Folgende telnet-Kommunikation überprüft das Verhalten:

```
$ telnet MYHOSTNAME 25
HELO client.example.org
MAIL FROM: <>
RCPT TO: user1@MYHOSTNAME
RCPT TO: user2@MYHOSTNAME
DATA
QUIT
```

Aufgabe:

- Erzeugen Sie eine „normale" Bounce-Mail, indem Sie eine Mail von Ihrem Postfix als *relay* an eine andere Postfixinstanz mit unbekanntem Empfänger senden. Führen Sie den Vorgang mit gültigem und ungültigem Absender durch.
- Betrachten Sie die Bounce-Mails im Postfix-Log, im Postfach des gültigen Absenders und in der Mailqueue (`mailq`).
- Richten Sie die beschriebene Restriktion ein.
- Senden Sie eine fiktive Bounce-Mail per telnet an Ihren Benutzer.
- Erzeugen Sie oben beschriebene Ablehnung.

Eine weitere Möglichkeit zur Spamunterdrückung sind DNS-Blacklists.

> Eine DNS-basierte Blacklist ist ein Server, der Ihnen Ressourcen (IP-Adressen, envelope sender oder Domains) nennen kann, die wahrscheinlich nicht vertrauenswürdig sind. Blacklists sind sehr effizient in der Abwehr von Spam, aber falsch angewendet

oder mit einer falschen oder gar fehlerhaften Blacklist können Sie schnell viele gültige Nachrichten verlieren.[7]

Da diese Prüfungen stets Anfragen im Internet machen, können sie die Performance eines Mailservers stark beeinträchtigen. Eine verfügbare Blacklist ist spamhaus.org. Sie kann mit folgender Restriktion in Postfix integriert werden:

```
reject_rbl_client zen.spamhaus.org
```

Um diese Aufgabe aus Postfix autzlagern, kann Postscreen eingesetzt werden. Alle am Port 25 eingebenden Mails werden direkt vom Postscreen übernommen. Der Betrieb von Postscreen erfordert, dass die direkten Nutzer des Servers ihre Mails an einem anderen Port (typischerweise 587) anliefern. Postscreen prüft den anlieferenden Client und übergibt, falls die Prüfung das erlaubt, den geöffneten Stream an Postfix (pass in master.cf). Weitere Hinweise: https://blog.schaal-24.de/mail/postscreen-im-kampf-gegen-spam/

Aufgabe:

Installieren Sie postscreen gemäß dieser Anleitung.

2.4 Externe Message Transfer Restrictions

Externe Message Transfer Restrictions sind Prüfungen, für die Postfix zusätzliche Programme benutzt. Man spricht von Policy-Delegierung. Dabei wird ein Satz Metadaten (Client, Absender, Empfänger, etc.) an das externe Programm übergeben, das als Ergebnis die auszuführende Aktion an Postfix zurück gibt. Ein Beispiel hierfür ist postgrey. Postgrey wurde früher gerne zur Spamabwehr eingesetzt. Die SMTP-Spezifikation sagt: Kann ein Server grad mal keine Mail annehmen, antwortet er mit „bitte später nochmal“. Gemäß Spezifikation muss der Client dann einige Minuten später nochmal senden. Vielen Spammern ist dies aber zu mühsam, sie verzichten auf das erneute Senden. Allerdings klagen auch viele Mailanwender, wenn sie sich bei einem Internetportal registrieren, wenn die Bestätigungsmail erst verspätet ankommt.

Um eine *external message transfer restriction* zu aktivieren, muss dem passenden Prüfpunkt ein `check_policy_service`-Eintrag hinzugefügt werden. Zusätzlich muss der *policy service* in `master.cf` definiert werden. Ein Beispiel hierfür ist der SPF-Check, s. Abschnitt 3.4.

2.5 Interne Content-Filter

Die folgenden Prüfungen werden auf den Datenteil der Mail angewendet. Dieser besteht aus

- Header
- Mime-Header
- Nested-Header

[7] Hildebrandt (2008): S. 156.

- Body[8]

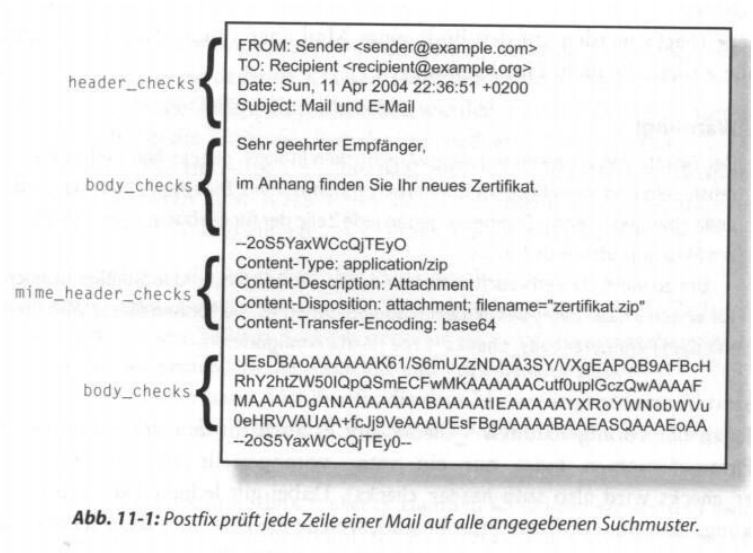

Abb. 11-1: *Postfix prüft jede Zeile einer Mail auf alle angegebenen Suchmuster.*

Postfix überprüft dabei jede Zeile innerhalb des entsprechenden Teils auf alle für diesen Teil angegebenen Suchmuster. Da dies beim Body sehr aufwändig sein kann, überprüft Postfix standardmäßig nur die ersten 51200 Bytes. Folgende Aktionen können die Prüfungen u.a. nach sich ziehen:

- `REJECT`: Abweisen der Mail
- `DEFER_IF_REJECT, DEFER_IF_PERMIT`: Temporäres Abweisen der Mail
- `IGNORE`: Entsprechende Zeile aus der Mail entfernen
- `WARN`: Eintrag ins Log
- `DISCARD`: Mail annehmen und verwerfen
- `REDIRECT`: Mail an einen anderen Empfänger weiterleiten
- `DUNNO`: Mit der nächsten Prüfung fortfahren
- `PREPEND`: Zusätzliche Headerzeile voranstellen

Um diese Checks zu aktivieren ist in der `main.cf` ein `..._checks`-Eintrag zu erstellen, der auf `regex:/path/to/..._check` verweist. In der Datei `..._check` sind dann Einträge in folgender Form zu erstellen:

```
/<Suchmuster>/ AKTION <Text>
```

Anmerkung: `postconf -m` zeigt die unterstützten Maps.

> Aufgabe:
>
> Implementieren und testen Sie folgende Prüfungen:
>
> - E-Mails an den Daemon selbst (`To: .*<>`) sollen abgewiesen werden

[8] Hildebrandt (2008): S. 182.

- `bat/exe`-Anhänge an den `postmaster` weiterleiten. In den Mime-Headern gibt es den Eintrag `name=".....exe"`.

Zur Beachtung: Es genügt `header_checks` zu aktivieren, da diese standardmäßig auch für die `mime_header` angewendet werden: `postconf -d mime_header_checks` (`-d`: *show default value*)

2.6 Externe Content-Filter

Die in Postfix eingebauten Filtermechanismen aus dem vorigen Kapitel sind nur für rudimentäre Aufgaben brauchbar; raffiniertere Anforderungen ans Filtern müssen durch Delegierung an externe Programme realisiert werden.

Postfix kann den Inhalt einer Mail vor oder nach dem Einstellen in die Queue inspizieren lassen. Wenn eine Mail gefiltert wird, bevor sie in die Queue gelangt, kann Postfix die Benachrichtigung über eine Nichtzustellung auf den Client abwälzen. Wenn die Mail allerdings erst in Queue akzeptiert wird, liegt diese Verantwortung bei Postfix.[9]

Da aber das Filtern gerne einige Zeit benötigt und währenddessen der Client-Timeout droht, wird üblicherweise folgendes Verfahren verwendet: Erst queuen, dann filtern. Dazu wird auf dem Server eine zweite Postfix-Instanz gestartet. PostfixA nimmt die Mails von außen an, leitet sie an das Filterprogramm weiter. Das Filterprogramm schickt die überprüften Mails an PostfixB, der wie weitere Zustellung übernimmt. Das Programm mMail[10] arbeitet als externer Filter um den Zugriff auf Mailverteiler zu prüfen. Mailverteiler können über Aliase einfach gepflegt werden. Nur hat per default dann einjeder Zugriff auf den Verteiler. mMail kennt dabei drei Zugriffsregeln:

- *all*: Einjeder hat Zugriff.
- *list*: Alle Listenmitglieder dürfen die Liste benutzen.
- Definition eines Nutzerkreises

Aufgabe:

- Installieren Sie das Paket `mmail`
- Aktivieren Sie die `mlist`-Funktionalität: `sudo mmail enable mlist`
- Überprüfen Sie den Vorgang: `sudo mmail list`
- Betrachten Sie die Einstellungen in `main.cf` und `master.cf`

[9] Hildebrandt (2008): S. 201.
[10] `http://wp.wagnertech.de/?page_id=388`

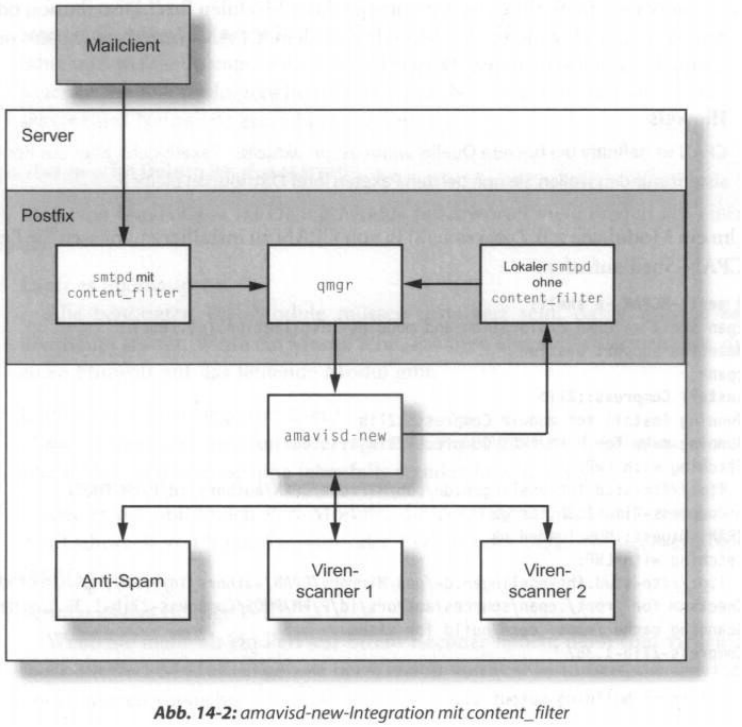

***Abb. 14-2:** amavisd-new-Integration mit content_filter*

Abbildung 4: amavis-new

Amavisd-new / clamav – Virenscanner

Abbildung 4 zeigt die beschriebene Architektur.[11]

Aufgabe:

Installieren Sie den Virenscanner. Dies ist auf zwei Weisen möglich:

- Über das `mmail`-Paket: `sudo mmail enable amavisd`
- In Einzelschritten.

Soll `amavisd` in Einzelschritten installiert werden, sind dazu folgende Schritte auszuführen:

- Installation der Pakete `amavisd-new` und `clamav`
- `Myhostname` in `/etc/amavis/conf.d/05-node_id` auf `wagnertech.de` ändern und `amavisd` neu starten
- Test, ob `amavisd` läuft: `telnet localhost 10024`
- `content_filter` für postfix definieren:
  - Eintrag in `main.cf: content_filter = amavisfeed:[127.0.0.1]:10024`
  - Eintrag in `master.cf: amavisfeed unix - - n - 2 smtp`
- Wiedereintritt ohne `content_filter`

[11] Hildebrandt (2008): S. 223.

- Mit Eintrag in `master.cf`:

```
127.0.0.1:10025 inet n  -       n       -       -       smtpd
  -o content_filter=
  -o smtpd_recipient_restrictions=
```

- Prüfung: `telnet localhost 10025`

- Mail schicken und log betrachten

Hinweis: Da ein Virenscan einen hohen Ressourcenbedarf hat, sind diese Konfigurationen noch optimierungsfähig: Art den Protokolls: `lmtp` statt `smtp`, Ausschalten weiterer Prüfungen, Ausschalten einer Verschlüsselung, ...

2.7 Zusammenfassung

Zur Kontrolle von Mails gibt es in Postfix folgende Möglichkeiten:

- Restriktionen (`smtpd_..._restrictions =`)
- interne Filter (`header_checks =`)
- externe Filter (`content_filter =`)

Bei den Restiktionen gibt es

- eingebaute Restriktionen (z.B. `reject_non_fqdn_recipient`)
- interne Restriktionen (z.B. Regelen in einer Hashmap oder über Regex-Ausdrücke)
- externe Restriktionen (Verweis auf einen externen Dienst)

3 Fortgeschrittene Funktionen

3.1 TLS Verschlüsselung

Transport Layer Security (TLS) beschreibt ein Verfahren, bei dem die Verbindung zwischen zwei Hosts verschlüsselt wird, bevor (vertrauliche) Informationen ausgetauscht werden. Diese Funktionalität implementiert Postfix sowohl client- als auch serverseitig. Darüber hinaus kann Postfix Clients anhand ihrer TLS-Zertifikate identifizieren und ihnen daraufhin das Relayen erlauben.[12]

[12] Hildebrandt (2008): S. 363.

TLS verschlüsselt die Verbindung zwischen Hosts, stellt aber keine Ende-zu-Ende-Verschüsselung sicher. In den Mailsqueues der Server liegen die Mails stets unverschlüsselt vor.

Bei der TLS-Kommunikation spielen Zertifikate eine wichtige Rolle. Der Server stellt dabei sein Zertifikat zur Verfügung, das der Client verifizieren kann. Dabei findet folgender Informationsaustausch statt:

1. Ein Client verbindet sich mit einem Server
2. Der Client beginnt eine ESMTP-Kommunikation ...
3. Der Server bietet STARTTLS als eine seiner Kommunikationsfähigkeiten ... an.
4. Beherrscht der Client seinerseits TLS, antwortet er STARTTLS an den Server.
5. Der Server signiert daraufhin sein öffentliches Zertifikat mit seinem privaten Schlüssel und sendet das öffentliche Zertifikat an den Client.
6. Der Client überprüft das Zertifikat, indem er die Signatur der CA ausliest und sie mit der des abgespeicherten CA-Zertifikats vergleicht.
7. Der Client überprüft, ob die Angaben im CN-Feld des Serverzertifikats mit dem FQDN-Hostnamen des Servers übereinstimmen.
8. Client und Server handeln einen Session-Key aus und verschlüsseln anschließend die Transportschicht.
9. Die SMTP-Kommunikation beginnt (verschlüsselt) erneut, und die Daten werden ausgetauscht.
10. Nach Beendigung des Datenaustauschs endet die SMTP-Kommunikation und damit auch die TLS-Session.

TLS ist bei Debian-Postfix mit einem selbst erstellten Zertifikat standardmäßig aktiviert. Um zu erkennen, dass der Transport auch verschlüsselt wird, muss der TLS-Loglevel wenigstens auf 1 stehen (`smtpd_tls_loglevel = 1`)

Aufgabe:

- Setzen Sie den TLS-Loglevel auf 1
- Senden Sie eine Testmail zu einem anderen Rechner mit diesem Kommando:

```
openssl s_client -starttls smtp -CApath /etc/ssl/certs/  \\
        -connect RECHNER:25
```

- Senden Sie nun eine Mail per mutt/mail zum selben Rechner.

Wenn Sie nun das Log betrachten, stellen Sie fest, dass im ersten Fall TLS verwendet wurde, im zweiten hingegen nicht. Postfix bietet zwar als Server standardmäßig TLS an, verwendet es als Client aber nicht. Dies kann mit dem Konfigurationseintrag `smtp_tls_security_level` eingestellt werden. Je nach Wert verwendet der Client TLS, wenn es geht, oder zwingend.

Aufgabe:

- Aktivieren Sie auch auf Senderseite TLS (`smtp_tls_security_level = may`)
- Senden Sie eine Mail per mutt/mail
- Betrachten Sie das Log

3.2 Bearbeitung von Mail-Listen

Das Paket `mmail` bietet die Möglichkeit einfache Mail-Listen zu administrieren. Über die Kommandos `mmail` und `mlist` lassen sich diese Funktionalität verwenden.

Aufgabe:

- Betrachten sie sich die *man pages* von `mmail` und `mlist`.
- Legen Sie eine Liste an, die auch einen Nutzer auf Ihrer Postfix-Instanz enthält und testen Sie diese.
- Verwenden Sie dabei verschiedene Einschränkungen für die erlaubten Absender.

3.3 GPG-Serververschlüsselung

Die TLS-Verschlüsselung greift nur bis zum nächsten Server. Was dort mit unseren Daten passiert, liegt außerhalb unseres Einflussbereichs. Vertrauliche Daten sollten daher stets Ende-zu-Ende-verschlüsselt werden. Dazu ist es aber notwendig, mit dem Empfänger Schlüssel auszutauschen. Beim Public-Key-Verfahren wird stets mit dem öffentlichen Schlüssel des Empfängers verschlüsselt.

Ein Problem stellen oft die privaten Mailaccounts der Mitarbeiter dar. Entweder wünschen sie eine Weiterleitung dorthin oder sie leiten selbst Mails auf solche Accounts weiter. Selbst wenn der Transport dorthin TLS-verschlüsselt wäre, wäre die Mail nun auf einem gmx-t-online-etc-Server für alle Geheimdienste, sowie alle Wirtschaftsspione lesbar. Abhilfe leistet eine serverseitige gpg-Verschlüsselung, die

- Mails an Accounts, deren öffentlicher Schlüssel vorliegt, verschlüsselt
- Mails an andere gmx-t-online-etc-Accounts abweist. Kein seriöser Geschäftspartner wird einen gmx-t-online-etc-Account haben.
- Software: `https://github.com/TheGreatGooo/gpg-mailgate`

Aufgabe:

Der Transport soll gpg-verschlüsselt werden.

- Erstellen Sie auf `RECHNER_A` ein Schlüsselpaar (`gpg -gen-key`): Das dauert seine Zeit. Ggf. bereits erzeugte Schlüssel verwenden.
- Öffentlichen Schlüssel exportieren (`gpg -armor -export "SCHLUSSEL_NAME" >DATEI`) und auf `RECHNER_B` kopieren.
- Die weitere Installation erfolgt gemäß der `INSTALL.md` – Datei
- Doppelte `content_filter`-Kette erstellen: Erst in den Virenscanner, zurück zu Postfix, dann in `gpg-mailgate` und nochmal zu Postfix

3.4 Sender Policy Framework (SPF)

Die Idee des SPF ist folgende: Wenn eine Mail xxx@sender.com als Absender hat, sollte sie auch von sender.com kommen. Bei großen Domains können aber mehrere Server für den Betrieb der Domain incl. Mailservice zuständig sein. Daher wird beim DNS-Eintrag der Domain hinterlegt (TXT-Record), welche Server für den Versand von Mails dieser Domain zuständig sind. Der Mailempfänger prüft also, ob die IP-Adresse (IP-Ebenene) mit dem Absender (Protokollebene) gemäß dem DNS-Eintrag der Domain zusammenpasst. Dazu ist es wichtig, dass der sendende Server den richtigen RDNS-Eintrag hat. Im SPF-Eintrag kann festgelegt werden, ob es eine harte Zurückweisung oder nur einen Eintrag für das Spamassign geben soll.

Das alles klingt soweit richtig und vernünftig, gäbe es keine e-mail-Weiterleitungen. Diese verhageln diese Prüfung. Wird nämliche eine Mail von `DOMAIN_A` an `DOMAIN_B` geschickt. Dort erfolgt eine Weiterleitung an `DOMAIN_C` ergibt sich dort folgende Situation: Absender der Mail ist (nach wie vor) ein Benutzer von `DOMAIN_A`, der übermittelnde Server ist aber von `DOMAIN_B`. Damit schlägt ein SPF-Check fehl. Domains wie `google.com` verlangen, dass entweder der SPF-Check erfolgreich ist oder der DKIM-Check (s. nächster Abschnitt).

Aufgabe:

Installieren Sie einen SPF-Check (externe Restriction) gemäß Debian Tutorial[13], erst ohne nachgelagertem Spamfilter.

Anmerkung: Die Domain `wagnertech.de` hat einen SPF-Eintrag. Wenn Sie diese Domain als Absender verwenden, sollte der Check reagieren.

3.5 DomainKeys Identified Mail (DKIM)

Die Idee von DomainKeys Identified Mail (DKIM) sieht so aus: Wenn ein DKIM-konfigurierter Mail-Server eine Mail versendet, verwendet er einen auf dem Mail-

[13] `https://www.debiantutorials.com/implementing-spf-checks-in-postfix/` (17.4.2023)

Server gespeicherten privaten Schlüssel und fügt der Mail eine Signatur hinzu.

Der Empfänger-Mail-Server kann nun zur Kontrolle den DNS-TXT-Eintrag des Senders auslesen. Dieser enthält den öffentlichen Teil des Schlüssels. Damit kann überprüft werden, ob die Mail und die Signatur zusammenpassen.

Mit DKIM kann also überprüft werden, ob die Mail tatsächlich von der in der Mail angegebenen Adresse stammt — und das ist schon viel wert. (Nur der legitime Mail Server verfügt über den privaten Schlüssel und kann damit eine korrekte Signatur durchführen.)

DKIM und SPF: DKIM wird oft zusammen mit SPF beschrieben. Das Sender Policy Framework (SPF) ist ein wesentlich einfacheres Verfahren, das das Fälschen der Absenderadresse vermeiden soll. Die Gemeinsamkeit zu DKIM besteht darin, dass auch SPF einen DNS-Eintrag nutzt. Anders als bei DKIM enthält dieser Eintrag aber nur die IP-Adressen der Mail-Server, die für einen bestimmten Domainnamen Mails versenden dürfen. ... Ähnlich minimal ist aber auch der Nutzen: Besonders ungnädig ist der Postfix-Experte Peer Heinlein, der SPF als Bullshit und Broken by Design bezeichnet (Quelle[14]).

DKIM/SPF als Spam-Schutz?

Niemand kann Spammer daran hindern, ebenfalls DKIM und SPF zu nutzen. Insofern sind DKIM und SPF kein eindeutiges Kriterium zur Spam-Erkennung. Welchen Grund gibt es dann, sich überhaupt damit auseinanderzusetzen?

Die Hauptmotivation besteht normalerweise darin, dass man als kleiner Mail-Server-Betreiber ständig das Problem hat, dass große Mail-Provider vom eigenen Mail-Server versandte Mails als Spam betrachten. Anders formuliert: Zum Ärger darüber, selbst ständig mit Spam überschüttet zu werden, kommt das Problem hinzu, dass Google, GMX, Microsoft, Yahoo etc. eigene Mails fälschlich als Spams identifizieren. Immer wieder muss man Kunden, Geschäftspartner etc. darauf hinweisen: »Werfen Sie bitte auch einen Blick in Ihren Spam-Ordner!« oder »Fügen Sie kofler.info zur Liste der vertrauenswürdigen Empfänger (= Whitelist) hinzu.«

Um die Schmach zu vermeiden, dass die Mails der eigenen Firma als unseriös betrachtet werden, versucht man seinen Mail-Server so regelkonform wie möglich zu betreiben und implementiert selbst Verfahren, von deren Wertlosigkeit man eigentlich überzeugt ist.[15]

Die Implementierung für das Erstellen der DKIM-Signatur sowie deren Überprüfung wird mit einem *milter (mail filter)* durchgeführt. Ein *milter* ist ein Zwischending zwischen *restriction* und *filter*. Wie eine Restriktion wird der *milter* vor dem Queuen durchgeführt, aber wie der Filter hat der *milter* Zugriff auf den kompletten Körper der Mail. *Milter* wurden in Postfix in der Version 2.2 speziell für DKIM eingeführt, haben sonst kaum Bedeutung. Eine Anleitung zu Installation findet sich hier[16].

[14] `https://www.heinlein-support.de/blog/news/gmx-de-und-web-de-haben-mail-rejects-durch-spf` (18.4.2023)
[15] `https://kofler.info/dkim-konfiguration-fuer-postfix/` (18.4.2023)
[16] `https://kofler.info/dkim-konfiguration-fuer-postfix/` (18.4.2023)

Aufgabe:

Implementieren Sie DKIM gemäß der genannten Anleitung. Ein Mail, die Sie nun mit der Domain Ihrer Postfixinstanz abschicken sollte nun mit einer DKIM-Signatur versehen werden. Zur überprüfung der Signatur brauchen wir einen DNS-Eintrag, den wir aber für unsere Testinstanzen nicht haben.

4 Quellen

Hildenbrand (2008)	Hildenbrandt, Ralph; Koettner, Patrick Ben: Postfix. Einrichtung, Betrieb und Wartung
gpg-mailgate	https://github.com/TheGreatGooo/gpg-mailgate