

Postfix, AMaViS und Co

Christian Rößner

Fachbereich Informatik
Bachelor Seminar Ausarbeitung
erstellt am
16. Oktober 2008

Inhaltsverzeichnis

I	Theoretische Grundlagen	4
1	Einführung	5
1.1	Theoretischer Hintergrund	5
1.1.1	Spam, UCE und UBE	5
1.1.2	Botnetze	6
1.1.2.1	Funktionsweise	6
1.1.2.2	Bekannte Botnetze	8
1.1.3	Offene Relays	8
1.1.4	Backscatter	9
2	Grundlagen des SMTP – Protokolls	10
2.1	Envelope MAIL FROM und Envelope RCPT TO	12
2.2	Mail-Header	13
3	Komponenten eines Mailserver-Setups	14
3.1	Kunden Datenhaltung	15
3.2	DNS – lokaler Nameserver	16
3.2.1	RBL – Realtime Black Listen	16
3.2.1.1	DNSBL – DNS Black Listen	16
3.2.1.2	RHSBL – Right Hand Side Black Listen	17
II	Konfiguration des Mailsystems	18
4	Postfix – Ein reiner MTA	19
4.1	Restriktionen	20
4.1.1	Policy Delegation Services	23
4.2	AMaViS als Pre-Queue Filter	23
4.2.1	Rechtlicher Hintergrund	24
5	AMaViS – A Mail Virus Scanner	26
5.1	Spamfilter mit SpamAssassin	27
5.2	Anti-Viren Komponeten	28

6 Fazit	29
III Anhang	30
A Verwandte RFCs	31
Glossar	31
Stichwortverzeichnis	31

Teil I

Theoretische Grundlagen

1 Einführung

Die Einrichtung eines sicheren Mailserver gehört mit zu den komplexesten Teilgebieten der Systemtechnik. Während in den frühen Anfängen des Internets der Versand von EMail völlig unproblematisch war, hat sich dies in den letzten Jahren durch steigende Internetkriminalität leider deutlich geändert.

Viele Firmen haben erkannt, dass das Medium EMail eine geeignete Möglichkeit zum Versand von Massenmail an potentielle Kunden ist. Die Vorteile dieser Methode gegenüber der klassischen Werbung liegen klar auf der Hand

- Kostenersparnis, da EMail in großen Mengen kostenlos verschickt werden können
- Bei Millionen von verschickten EMail genügt auch schon der kleinste Bruchteil gelesener EMail, um die illegale Werbekampagne als ein Erfolg zu werten

Aus diesen genannten Gründen müssen heutige Mailserver in Echtzeit jede eingehende EMail untersuchen und entscheiden, ob diese angenommen und zugestellt oder abgewiesen werden soll.

Das folgende Kapitel liefert daher zunächst einmal eine Zusammenfassung der wichtigsten theoretischen Grundlagen, damit die Hintergründe für eine erfolgreiche Installation gegeben sind.

1.1 Theoretischer Hintergrund

Als Leser dieses Dokuments sind Sie sicher schon öfters mit den Begriffen Spam, UCE und UBE in Berührung gekommen, haben aber womöglich keine genaue Definition dieser Begrifflichkeiten.

1.1.1 Spam, UCE und UBE

Die Bezeichnung Spam bedeutet übersetzt Dosenfleisch (*Spread Ham*). Die Herkunft dieses Begriffs geht vermutlich auf einen Monty Python Sketch zurück, in dem im Verlauf

jedes Wort durch Spam ersetzt wurde. Als Folge war schließlich jede Kommunikation unmöglich.

Auch eine etwas sachlichere Deutung lässt sich aus diesem Begriff ableiten: SPread Around Message.

In der Zusammenfassung bedeutet es also, dass durch Massenmail ein Mailserver in der letzten Konsequenz handlungsunfähig wird, da er durch Spam-Überflutung nicht mehr kommunizieren kann. Leidet also ein Mailserver unter zu hoher Spam-Last, so ist dies praktisch einer sog. Distributed Denial-of-Service (DDoS) Attacke gleichzusetzen.

UBE ist eine Abkürzung für Unsolicited Bulk EMail und bedeutet das unerwünschte Zusenden von Massenmails (auch nicht gewerbliche), wie z.B. Kettenbriefe, Phishing oder auch Volksverhetzung.

UCE ist die Abkürzung für Unsolicited Commercial EMail und gruppiert den Begriff der unerwünschten Werbe-EMails.

Historisch lässt sich feststellen, dass Sanford Wallace der erste sog. Spammer war, der die Idee der kostenlosen Werbemail erfand.

1.1.2 Botnetze

Eines der größten Probleme zur Bekämpfung von Spam stellen sog. Botnetze dar. Dabei handelt es sich um zentral gesteuerte Zombie-PCs, die nach belieben jede gewünschte Aktion ausführen.

Dies kann also der Versandt von Massenmails sein, aber auch jede andere denkbare Aktion (Zum Beispiel eine DDoS-Attacken auf Webserver, etc.)

1.1.2.1 Funktionsweise

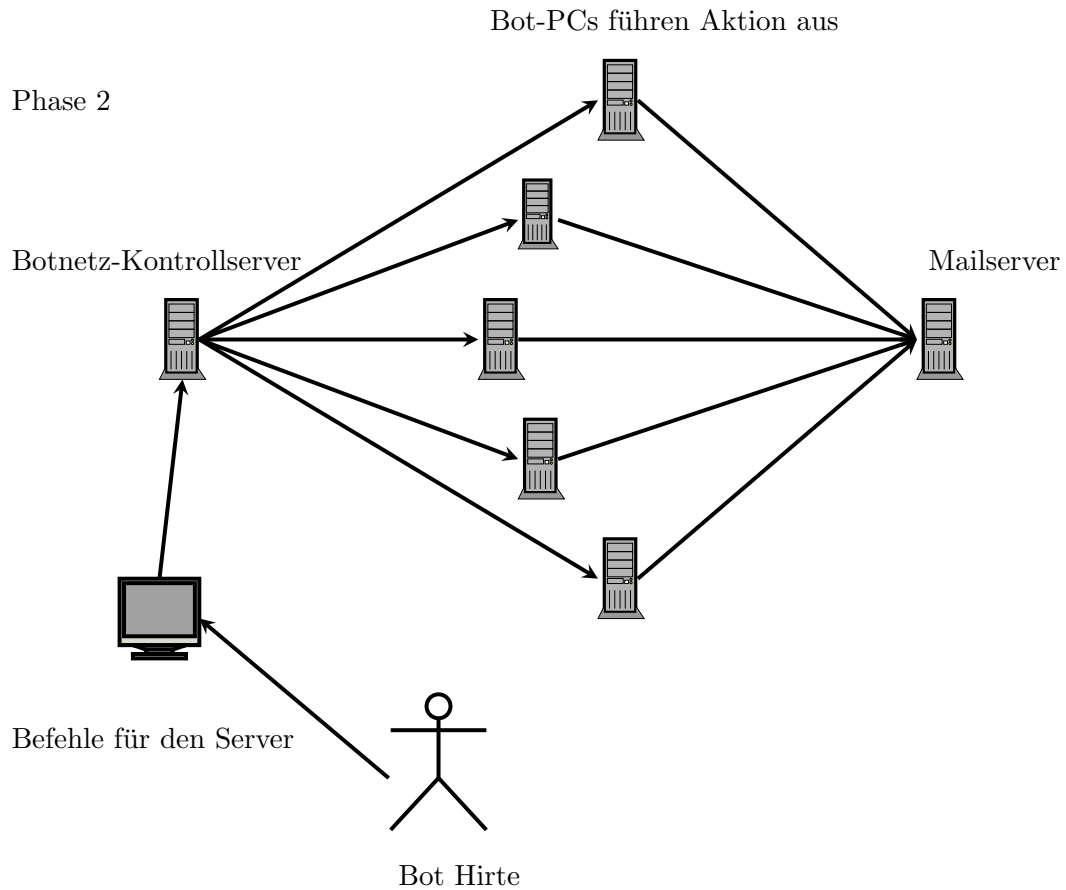
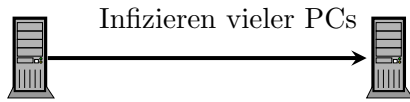
Wie entstehen nun Botnetze?

Botnetze entstehen in der Regel in zwei Phasen. In der ersten Phase wird eine Schad-Software durch bekannte Exploits auf vielen PCs installiert. Dies kann durch unachtsam geöffnete EMailen geschehen, bei denen Dateianhänge geöffnet werden. Aber auch das Benutzen von sog. Key-Generator-Programmen (Crack.exe) zum Erstellen illegaler Schlüssel-IDs für geklaute Programme führt oftmals zu Installationen solcher Software.

Ein solcher PC wird ab diesem Zeitpunkt als Zombie-PC oder Bot deklariert, da die Schad-Software unbemerkt Kontakt zu einem Netzwerk aufbaut (Dies kann z.B. ein

IRC-Netzwerk sein).

Phase 1



Eine wichtiges Kriterium zur Bekämpfung solcher Botnetze liegt in der Tatsache, dass fast alle infizierten Computer aus dynamischen Einwahlnetzen stammen (DSL, Modem, ISDN oder Kabelnetz). Diese dynamischen Netze lassen sich in einer sog. PCRE-Datei erfassen und konsequent von der Einlieferung von E-Mails ausschließen.

1.1.2.2 Bekannte Botnetze

Zu den derzeit bekanntesten Botnetzen gehören das Storm-Netz und das Srizbi Netz.

Das Storm-Botnetz¹ erhielt seine Bezeichnung durch den gleichnamigen Wurm Storm, ein sog. trojanisches Pferd. Im Jahr 2007 schätzte man die Größe dieses Netzes auf 1-10 Millionen infizierter Windows-PCs. Dies entsprach etwa 8% der damals bekannten Malware. Durch geeignete Updates seitens Microsoft konnte die Größe schon stark reduziert werden, da das Werkzeug MSRT den Schädling erkennen und löschen kann.

Das Srizbi²-Botnetz gilt derzeit als eines der größten Botnetze. Auch dieses Netz wurde nach dem gleichnamigen Trojaner benannt und hat das o.g. Netz bereits größtmäßig hinter sich gelassen. Fast 39% aller verschickten Spam-EMails entstammen diesem Netzwerk.

Zusammenfassend lässt sich sagen, dass Botnetze in von Kunden gemietet werden können. Der Preis ist hierbei von der Größe des Botnetzes abhängig, sowie von der Güte der gesammelten Empfänger-EMail-Adressen (Diese stammen i.d.R. aus Adressbüchern der befallenen PCs und haben daher eine hohe Trefferwahrscheinlichkeit).

Selbst Regierungen bedienen sich von Zeit zu Zeit solcher Netze, da ein solches Werkzeug eine gigantische Cyber-Armee darstellt.

1.1.3 Offene Relays

Als das SMTP-Protokoll entwickelt wurde (RFC[1] 821), gab es noch keine Authentifizierungsmechanismen. Jeder Computer konnte so beliebig EMails über ein nahe gelegenes Mail-Relay verschicken.

Erst durch Erkenntnis im praktischen Alltag wurde ein erweitertes Protokoll im RFC Standard 2821 vereinbart, der erstmals auch SMTP-AUTH als Kommando unterstützte. Man bezeichnet den neuen Standard daher auch als ESMTP (Der aktuelle Standard umfasst dabei natürlich weit mehr Neuerungen als nur SMTP-AUTH).

Als ein offenes Relay wird heutzutage ein Mailserver bezeichnet, der ungeprüft beliebigen Mail-Absendern das Verschicken von EMails gestattet. In Kombination mit Botnetzen kann dies fatale Auswirkungen haben.

Es empfiehlt sich daher nach jeder Konfigurationsänderung eine Testprüfung des eigenen Servers durchzuführen. Über die Internet-Seite <http://www.abuse.net/relay.html> kann

¹Quelle: http://de.wikipedia.org/wiki/Storm_Botnet

²Quelle: <http://de.wikipedia.org/wiki/Srizbi>

ein beliebiger Server untersucht werden.

1.1.4 Backscatter

Eine weitere Quelle des Spam-Versandts sind sog. Backscatter Mailserver. Hierbei empfangen falsch konfigurierte Mailserver – meist als Mail-Gateways – E-Mails für einen dahinter liegenden Mailserver, kennen aber die Liste der gültigen Empfänger-Adressen nicht. Versucht das Gateway nun z.B. seinem vertrauten eigenen Firmenmailserver die bereits empfangenen E-Mails durchzureichen, so werden alle E-Mails, für die es keinen realen Empfänger gibt, vom Firmenmailserver abgewiesen.

Dem Mail-Gateway bleibt nun nur noch eine verspätete Abweisung (Late Bounce) an den Ursprungsserver, der die E-Mail vormals einlieferte. Anhand der Absenderadresse wird nun der zuständige Mailserver ermittelt und diesem im Folgenden die Bounce-Mail zugestellt. Da es sich aber leider in aller Regel bei den verwendeten Absender-E-Mail-Adressen um Fälschungen handelt, bekommen nun völlig unbeteiligte Mailserver Bounce-Meldungen.

Ein anderes Szenario stellt ein Mailserver dar, der eingehende E-Mails vollständig angenommen hat, diese dann auf Spam prüft und im positiv-Fall als Late-Bounce abweist. Hier existieren gleich zwei fatale Fehler, zu denen ich später noch einmal kommen werde.

Ein Server, der ungeprüft anhand der Absender-E-Mail-Adresse Mails bouncet, wird Backscatter genannt.

2 Grundlagen des SMTP – Protokolls

Das heute gültige und verwendete Protokoll zur Übermittlung elektronischer Post geschieht nach RFC 2821. Es kann in etwa wie folgt skizziert werden:

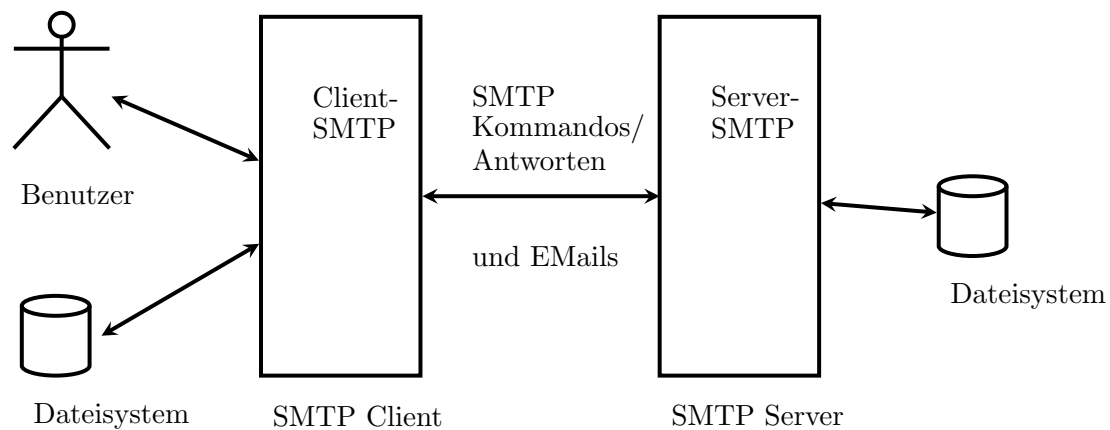


Abbildung nachgebildet aus dem RFC 2821

Das Protokoll arbeitet interaktiv. Der SMTP-Client (MUA¹) baut eine TCP/IP-Verbindung auf Port 25 zu einem SMTP-Server (MTA²) auf. Sobald die Verbindung erfolgreich zustande gekommen ist, sendet der Server ein sog. Greeting. Dieses besteht aus einer 3-stelligen Ziffer, einem Leerzeichen und einem Folgetext.

```
220 mx0.roessner-net.de ESMTP Postfix (2.5.5)
```

Der Server kennt hierbei 5 verschiedene Klassen von Statuscodes.

- 1xx – Positive Preliminary reply
- 2xx – Positive Completion reply
- 3xx – Positive Intermediate reply
- 4xx – Transient Negative Completion reply

¹Mail User Agent, kann ein EMail-Programm sein oder ein ausliefernder Mailserver; in beiden Fällen handelt es sich um einen SMTP-Client

²Mail Transfer Agent, entspricht einem annehmenden Mailserver; oder auch SMTP-Server

- 5xx – Permanent Negative Completion reply

Die genaue Übersetzung und Bedeutung ist im RFC³ genauestens definiert. Eine grobe Zusammenfassung könnte wie folgt interpretiert werden:

Codes beginnend mit 2xx zeigen in aller Regel eine positive Beantwortung des Servers an, 4xx bedeuten einen temporären Fehler, bei dem der MUA seinen Einlieferungsversuch zu einem späteren Zeitpunkt erneut durchführen soll und 5xx Codes bedeuten eine unwiderrufliche Annahmeverweigerung des MTAs.

Nachdem der Server das Greeting gesendet hat, tritt nun der MUA in Aktion. Er sendet ein EHLO-Kommando – Extended HELLO. Damit begrüßt er den Server. Dieser reagiert im Gegenzug mit seinem Serviceangebot, welches etwa wie folgt aussehen könnte:

```
250-mx0.roessner-net.de
250-PIPELINING
250-SIZE 31457280
250-ETRN
250-STARTTLS
250-AUTH CRAM-MD5 NTLM
250-AUTH=CRAM-MD5 NTLM
250-ENHANCEDSTATUSCODES
250-8BITMIME
250 DSN
```

Ab diesem Moment ist eine Fallunterscheidung notwendig.

Fall 1: Der SMTP-Client ist ein EMail-Programm eines Benutzers

Handelt es sich bei dem MUA um einen Benutzer, der gerade mit seinem EMail-Programm eine Nachricht verschicken möchte, so muss der MUA (also das EMail-Programm) eine Authentifizierung gegenüber dem MTA durchführen. Der Server bietet im hier gezeigten Fall für eine unverschlüsselte TCP/IP-Verbindung Challenge/Response-Verfahren an. Möchte der SMTP-Client hingegen ein anderes Authentifizierungsverfahren nutzen, so muss entweder eine SSL-Verbindung auf Port 465 aufgebaut werden (SMTPS), oder der Client schickt das STARTTLS-Kommando.

Letzterer Fall führt nun zu einer gesicherten Kommunikation zwischen Client und Server, bei der nun der Server erneut sein Angebot zur Verfügung stellt. Dieses Mal bietet er aber auch PLAIN und LOGIN als Authentifizierungsmechanismen an.

³RFC 2821, Sektion 4.2.1 Reply Code Severities and Theory

Fall 2: Der SMTP-Client ist ein einliefernder Mailserver

Handelt es sich nicht um einen Benutzer, sondern einen anderen Mailserver, der eine EMail einliefern möchte, so entfällt die Authentifizierung und auch die SSL-Verbindung ist nun rein optional und wird bei großen EMail-Providern (Anbietern) eher selten genutzt.

2.1 Envelope MAIL FROM und Envelope RCPT TO

Unabhängig, ob es sich um ein EMail-Programm als MUA oder einen anderen Mailserver handelt, muss generell eine Absender- und Empfängeradresse übermittelt werden. Dies kann man sich wie bei der herkömmlichen Papierpost vorstellen. Verschickt man einen Brief, so wird dieser in einen Briefumschlag (*engl. Envelope*) gesteckt. Dieser Umschlag erhält ebenfalls Absender- und Empfängeradresse.

Das Kommando zur Angabe der Absenderadresse lautet MAIL FROM; dass der Empfängeradresse RCPT TO. In beiden Fällen muss eine gültige EMailadresse angegeben werden. Diese setzt sich wie folgt zusammen:

```
<localpart@domain.tld>
```

Genauere Informationen befinden sich wieder im bereits genannten RFC 2821, Sektion 4.1.1.2 MAIL (MAIL) und 4.1.1.3 RECIPIENT (RCPT).

Hat der Server die Angaben angenommen und akzeptiert, so folgt die DATA-Phase, in der die eigentliche Nachricht (Vergleichbar mit dem Papierbogen im Umschlag) dem Server übergeben werden kann. Das Ende der DATA-Phase wird mit <CR><LF>.<CR><LF>⁴ ausgelöst. Der Server wird hierbei eine Bestätigung zurücksenden und der Mail-Client kann die Verbindung erfolgreich abbauen.

Das grobe Verständnis dieses Protokolls ist daher so wichtig, weil der Server an verschiedensten Stellen beginnend mit dem Aufbau der TCP/IP-Verbindung über jeden einzelnen Teilschritt der Kommunikation hinweg Prüfungen durchführen muss, um letztlich zu entscheiden, ob die EMail angenommen oder abgewiesen werden soll.

⁴CR-LF steht für Carriage Return, Line Feed. Es bedeutet einen vollständigen Zeilenumbruch.

2.2 Mail-Header

Der Mail-Header ist Bestandteil der DATA-Phase. Jeder an der EMail-Kommunikation beteiligte Mailserver (Dies können durchaus mehr als nur 2 Server sein) verewigt sich selbst jeweils an oberster Stelle mit sog. Received-Zeilen. Möchte man als Benutzer den Mail-Header anschauen, so kann man in aller Regel diesen in seinem EMail-Programm anzeigen lassen.

EMail-Programme filtern nur eine Absender- und Empfängeradresse, Betreff, Datum, Anhänge und den Nachrichtenkörper heraus und zeigen diesen an. In Wirklichkeit umfasst die empfangene EMail aber wesentlich mehr Informationen.

```
...
Received: from localhost (localhost [127.0.0.1])
  by mx0.roessner-net.de (Postfix) with ESMTP id 29AC25211F
  for <monit@roessner-net.com>; Sat, 13 Sep 2008 10:23:13 +0200 (CEST
)
...
Received: from mx0.roessner-net.de ([127.0.0.1])
  by localhost (amavis.internal.roessner-net.de [127.0.0.1]) (amavisd
-new, port 10024)
  with ESMTP id sOaYcKQ9EMqc for <monit@roessner-net.com>;
  Sat, 13 Sep 2008 10:23:09 +0200 (CEST)
Received: from localhost (localhost [127.0.0.1])
  by mx0.roessner-net.de (Postfix) with ESMTP
  for <monit@roessner-net.com>; Sat, 13 Sep 2008 10:23:09 +0200 (CEST
)
Date: Sat, 13 Sep 2008 10:23:09 +0200
To: monit@roessner-net.com
From: christian@roessner-net.com
Subject: test Sat, 13 Sep 2008 10:23:09 +0200
...
This is a test mailing: STARTTLS+SASL Port 587
```

Listing 2.1: Beispiel für einen Mailb-Header

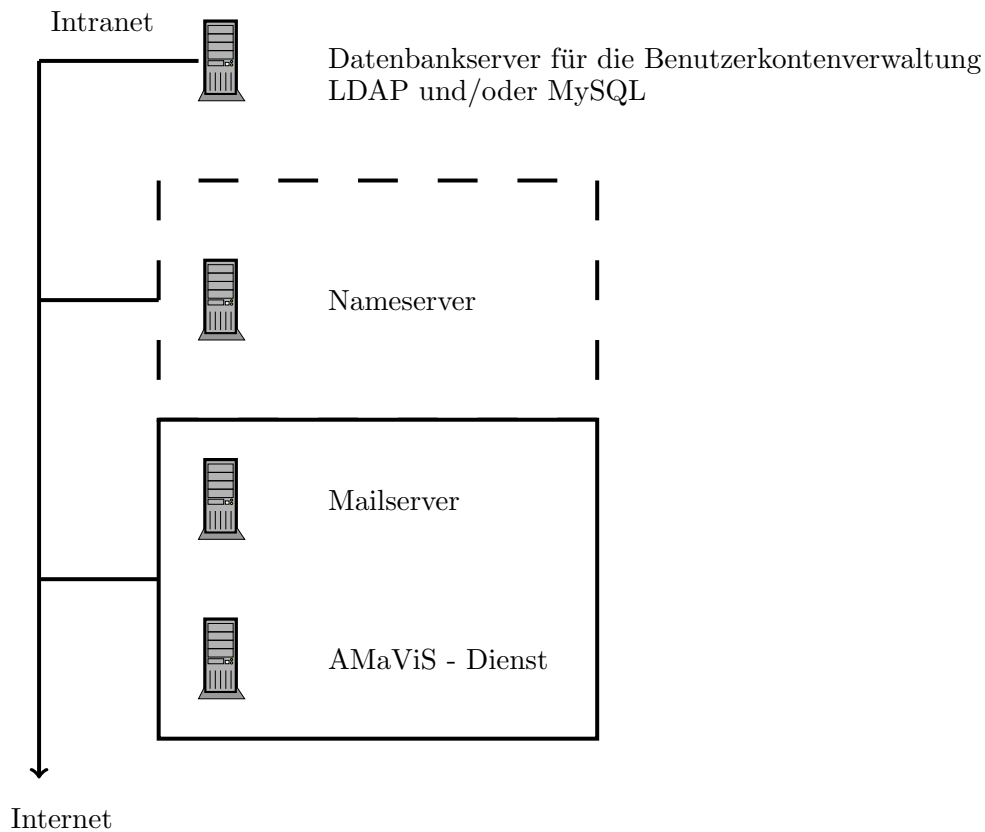
Möchte man das genaue Mail-Routing nachvollziehen, so müssen die Received-Blöcke also von unten nach oben ausgewertet werden. Spammer versuchen recht oft, diese Protokollinformationen zu fälschen. Ich empfehle daher, den ersten eigenen Received-Block zu finden. Der Folgeblock wird garantiert den echten realen einliefernden Mailserver offenbaren.

An dieser Stelle sei erwähnt, dass die Behandlung des SMTP-Protokolls und des Mail-Headers leider nicht in kompletter Detailtiefe erklärt werden kann, da dies den Rahmen dieses Dokuments überschreiten würde.

3 Komponenten eines Mailserver-Setups

Das folgende Kapitel dient lediglich als ein Anhaltspunkt, wie ein kleines Mailserver-Setup aussehen kann. Je nach Größe des Unternehmens oder auch des Standorts, kann eine Redundanz verschiedener Teilkomponenten essentiell sein.

Folgende Grafik zeigt exemplarisch die minimal benötigten Dienste, die für ein mittleres Mailserver-Setup einer Firma eingesetzt werden könnten.



Es empfiehlt sich, DNS, Datenbanken und Mailedienste auf getrennten Maschinen zu verwalten, da so eine erhöhte Sicherheit des Netzwerkes gegeben ist. Sollte einer der Dienste durch ein Sicherheitsloch kompromittiert werden, so hat dies dann nicht zwangsweise negative Auswirkungen auf andere.

Ein anderes, aber hier nicht weiter zu vertiefendes Thema ist die immer populärer werdende Virtualisierung von Servern. Diese kann als Kostenersparnis in die Planung eines Setups durchaus mit einbezogen werden.

3.1 Kunden Datenhaltung

Ein Mailserver arbeitet im Wesentlichen in drei Modi

- local – Der Mailserver verfügt über POSIX-Shadow-Konten und ist final zuständig für diese Empfängeradressen
- virtual – Über eine Anbindung an eine externe Datenbank, z.B. LDAP oder auch MySQL erfährt der Postfix-Mailserver, für welche Domains und Adressen er final zuständig ist.
- relay – Postfix kann EMail für gelistete Domains und Adressen annehmen, wird diese dann aber im späteren Mail- Routing an einen dafür final zuständigen Mailserver weiterreichen

In Multi-Domain-Umgebungen, wie dies häufig bei Internet-Service-Providern (ISP) der Fall ist, kommt *virtual* zur Anwendung. In Postfix selbst, werden über die Parameter `virtual_mailbox_maps`, `virtual_mailbox_domains` und `virtual_alias_maps` sog. Mappings übergeben. Eine Map ist eine Hash- oder Btree-Datenstruktur, die zweispaltig in Schlüssel und Wert getrennt vorliegt. In solchen Mappings können nun bis zu ca. 10.000 Datensätze problemlos untergebracht werden.

Typische Inhalte sind die Verwaltung der gehosteten Domains, der vorhandenen Mailkonten und evtl. Mailweiterleitungen. All diese Informationen sind für Postfix essentiell wichtig und werden über die genannten Parameter konfiguriert.

Eine besondere Form des Mappings sind Anbindungen an LDAP und/oder MySQL-Server. In sehr großen Netzwerken ist die zentrale Datenhaltung in einer Datenbank wesentlich effektiver. Zum einen besteht hierbei die Möglichkeit der Replikation (oder auch nur Teile davon), zum anderen sind Datenbanken oftmals durch geeignete Indizierung sehr effizient und schnell, was Lesezugriffe angeht.

Verwendet man beispielsweise LDAP, so sollte in Postfix das sog. Proxy-Mapping verwendet werden. `proxymap` ist ein Postfix-Modul, welches die eigentlich benötigten Datenbank-Verbindungen übernimmt und die Ressourcen dann an die einzelnen `smtpd`-Dienste verteilt. Welche Komponenten in Postfix vorhanden sind, folgt in einem späteren Abschnitt.

3.2 DNS – lokaler Nameserver

Eine weitere, wenn auch nicht auf den ersten Blick wichtige Komponente ist ein eigener Caching-Nameserver. Das Mailsystem muss für jede eingehende Verbindung unzählige DNS-Abfragen durchführen.

Würde man lediglich die Nameserver seines Rechenzentrums oder IPs anfragen, könnte ein Mailserver unter Last durchaus in Timing-Probleme geraten.

Der Nameserver kann entweder in unmittelbarer Nähe zum MTA stehen, oder falls nicht von anderen Diensten benötigt, sogar auf der MTA Maschine selbst. Aus Sicherheitsgründen empfehle ich dann allerdings den Betrieb in einer chroot()-Umgebung und die Verwendung einer kernelbasierenden Sicherheitsschicht (SELinux oder AppArmor, um hier nur zwei Beispiele zu nennen). Nameserver sind immer mal wieder in die Kritik von bekannt gewordenen Sicherheitslöchern geraten.

DNS-Dienste können aber auch für weitere Aufgaben im Mail-Setup genutzt werden. Dies zeigt der folgende Abschnitt.

3.2.1 RBL – Realtime Black Listen

Eine in den letzten Jahren etablierte Technik Spam zu bekämpfen besteht darin, bekannte offene Relays (*siehe 1.1.3, Seite 8*) – oder allgemeiner – schlecht konfigurierte Mailserver in sog. Realtime-Black-Listen einzutragen. Diese Listen werden in Nameservern verwaltet.

Hierbei existieren verschiedene Verfahren, von denen ich zwei vorstellen möchte:

3.2.1.1 DNSBL – DNS Black Listen

Häufen sich Spam-EMails von bekannten Mailservern, so kann man über den bereits weiter oben vorgestellten Mail-Header den Übeltäter ermitteln und dessen IP-Adresse im DNS-Server eintragen. Oft verwenden Spammer mehrere Server, doch sind diese schon nach wenigen Tagen auffällig geworden und lassen sich sukzessive über DNSBL sperren.

```
...
dnsbl                A          127.0.0.4
                    TXT        "Host-IP blocked"
$ ORIGIN dnsbl.rbl.roessner-net.de.
74.129.148.202      CNAME    dnsbl.rbl.roessner-net.de.
68.33.97.203       CNAME    dnsbl.rbl.roessner-net.de.
```

```

88.242.202.204      CNAME  dnsbl.rbl.roessner-net.de.
$ORIGIN 70.18.209.dnsbl.rbl.roessner-net.de.
114                CNAME  dnsbl.rbl.roessner-net.de.
24                 CNAME  dnsbl.rbl.roessner-net.de.
40                 CNAME  dnsbl.rbl.roessner-net.de.
$ORIGIN 175.226.209.dnsbl.rbl.roessner-net.de.
102                CNAME  dnsbl.rbl.roessner-net.de.
120                CNAME  dnsbl.rbl.roessner-net.de.
121                CNAME  dnsbl.rbl.roessner-net.de.
$ORIGIN dnsbl.rbl.roessner-net.de.
152.249.137.58    CNAME  dnsbl.rbl.roessner-net.de.
39.165.202.64     CNAME  dnsbl.rbl.roessner-net.de.
29.49.184.91      CNAME  dnsbl.rbl.roessner-net.de.
...

```

Listing 3.1: Auszug DNSBL einer Zonendatei

3.2.1.2 RHSBL – Right Hand Side Black Listen

Auch das Sperren von Mailservern anhand der Absender-Domain ist mit einem eigenen Nameserver möglich. Man sollte allerdings immer daran denken, dass die Absenderadresse womöglich eine Fälschung sein könnte. Leider wird viel Spam auch durch sehr große Mail-Provider in den Umlauf gebracht, so z.B. Google, Yahoo, AOL und Hotmail. Sperrt man hier nun beispielsweise die Domain @gmail.com, so werden alle Absender, die offiziell von Google stammen in Zukunft geblockt.

```

...
$ORIGIN rbl.roessner-net.de.
rhsbl      A      127.0.0.5
           TXT    "Address blocked"
$ORIGIN com.rhsbl.rbl.roessner-net.de.
centricsource CNAME rhsbl.rbl.roessner-net.de.
douglaspublications CNAME rhsbl.rbl.roessner-net.de.
$ORIGIN douglaspubs.com.rhsbl.rbl.roessner-net.de.
send1       CNAME rhsbl.rbl.roessner-net.de.
send2       CNAME rhsbl.rbl.roessner-net.de.
$ORIGIN com.rhsbl.rbl.roessner-net.de.
handbag     CNAME rhsbl.rbl.roessner-net.de.
wic004ef.exchange.server-login CNAME rhsbl.rbl.roessner-net.de.
$ORIGIN rhsbl.rbl.roessner-net.de.
1email.eu   CNAME rhsbl.rbl.roessner-net.de.
paradise.net.nz CNAME rhsbl.rbl.roessner-net.de.
...

```

Listing 3.2: Auszug RHSBL einer Zonendatei

Teil II

Konfiguration des Mailsystems

4 Postfix – Ein reiner MTA

Bei der nun im folgenden beschriebenen Einrichtung des Mailservers bildet die Open-Source-Software Postfix das Herzstück des Setups. Postfix ist ein reiner MTA, der in seiner Konzeption auf maximal mögliche Sicherheit ausgelegt wurde. Dies erreichte Wietse Venema – Autor des Programms und Sicherheitsexperte bei IBM – durch einen modularen Aufbau des Systems.

Postfix unterteilt sich dabei in einen Master-Prozess, der verschiedenste Unterprozesse startet, steuert und verwaltet. Jeder Unterprozess läuft dabei im System mit den minimal notwendigen Benutzerrechten. Zudem können annähernd alle Unterprozesse in einer chroot()-Umgebung ausgeführt werden.

Postfix umfasst mehr als 23 Teilkomponenten, daher können diese hier nicht alle vorgestellt und beschrieben werden. Der Hauptfokus bei der Einrichtung liegt im Normalfall auf zwei Komponenten. Eine Hauptaufgabe liegt in der Annahme von E-Mails, für die der Prozess `smtpd` zuständig ist. Bei der Auslieferung tritt `smtp` als Prozess in Aktion, also der MUA-Teil von Postfix.

Alle anderen Komponenten können zumeist mit ihren Standard-Werten direkt verwendet werden. Eine goldene Regel besagt, dass es schon triftiger Gründe bedarf, Standard-Einstellungen zu ändern, da dies impliziert, dass man besser als die Entwickler selbst über bestimmte Einstellungen bescheid weiß.

Postfix ist außerordentlich gut dokumentiert[2], trotzdem empfiehlt es sich Sekundärliteratur zu konsultieren, um ein umfassendes Verständnis zur Gesamthematik zu erlangen. Die folgenden Sektionen werden dabei nur einen sehr kleinen Einblick in die Einrichtung von Postfix geben.

Zwei Bücher haben sich auf dem Markt etabliert. Dies ist *Das Postfix-Buch*[3]; es ist **das** Standardwerk. Als gutes Referenzhandbuch hat sich das gleichnamige Buch *Postfix*[4] erwiesen.

Die Begrifflichkeit „reiner“ MTA erklärt sich damit, dass alle Aufgaben, die nichts explizit mit der SMTP-Kommunikation zu tun haben, an Hilfsanwendungen – zumeist in der Programmiersprache Perl geschrieben – delegiert werden.

4.1 Restriktionen

Die Konfigurationsdateien von Postfix befinden sich auf *unix-Systemen im Verzeichnis `/etc/postfix`. In der Datei `main.cf` werden alle Einstellungen für den Mailserver eingetragen. Die Datei `master.cf` ist die Hauptkonfigurationsdatei für den master-Prozess. Es ist eine tabellarisch aufgebaute Datei, in der alle Postfixmodule gelistet sind. Hier können Informationen wie die maximale Anzahl eines Prozesses, Parameter für einzelne Module oder auch Flags wie `chroot()` gesetzt werden.

Im Regelfall wird man in der `master.cf`-Datei relativ wenige Änderungen konfigurieren müssen. In einem späteren Abschnitt werde ich noch einmal darauf zurückkommen.

Im Folgenden betrachten wir einen kleinen Teilausschnitt aus der Datei `main.cf`. Es gibt weit über 550 Parameter, daher verweise ich auch hier wieder auf die vorhandene Fachliteratur.

```
1 smtpd_recipient_restrictions =
2     reject_non_fqdn_recipient
3     reject_non_fqdn_sender
4     reject_unknown_recipient_domain
5     reject_unknown_sender_domain
6     reject_unlisted_recipient
7     permit_mynetworks
8     permit_tls_clientcerts
9     permit_sasl_authenticated
10    reject_unauth_destination
11    ...
12    check_recipient_access btree:/etc/postfix/maps/
        roleaccount_exceptions
13    check_client_access pcre:/etc/postfix/maps/dynamic-ip.pcre
14    reject_invalid_helo_hostname
15    reject_non_fqdn_helo_hostname
16    reject_unknown_helo_hostname
17    reject_unknown_client_hostname
18    reject_unknown_reverse_client_hostname
19    check_policy_service inet:127.0.0.1:12525
20    ...
21    reject_rbl_client dnsbl.rbl.roessner-net.de
22    reject_rhsbl_client rhsbl.rbl.roessner-net.de
23    ...
24    check_client_access pcre:/etc/postfix/maps/greylist.pcre
```

Listing 4.1: Auszug aus Datei `main.cf`

Das abgebildete Codefragment bildet das Herzstück der Postfix-Konfiguration. Es beschreibt ein Regelwerk für den Prozess `smtpd`. Das kleine „d“ am Ende des Modulnamens

zeigt an, dass es sich um einen Daemon-Prozess handelt. Dieser wartet auf eingehende TCP/IP-Verbindungen (MTA).

Wie in vielen anderen Konfigurationsbeispielen aus der Informatik gilt auch hier das Prinzip first-match-wins. Eine Client SMTP-Verbindung durchläuft jede einzelne Zeile dieses Listings, bis eine Zeile eine Aktion auslöst.

Es gibt drei verschiedene Reaktionen, von denen immer genau eine für jede Zeile zutreffend ist (Dies ist nicht 100%-ig korrekt, ist aber ausreichend genau für die Folgeerklärungen).

- *permit* – Zustellung der EMail wird gestattet
- *reject* – Zustellung der EMail wird entgültig abgewiesen
- *dunno* – aktuelle Regel hat keine Auswirkung, Übergang zur nächsten Regel

Während bei *permit* und *reject* die Prüfungen sofort eingestellt werden, wird bei *dunno* die jeweils nächste Regel angefragt. Liegt keine weitere Regel zur Prüfung vor, so schließt die letzte Prüfung implizit mit *permit* ab.

Die Reihenfolge der Regeln ist ebenfalls essentiell wichtig. Postfix muss mit diesen sowohl vertraute als auch fremde Clients abfertigen. Die Zeilen 2-6 gelten dabei für alle Clients gleichermaßen. Sie bedeuten in etwa Folgendes:

Zeile 2, 3:

Für die angegebenen Evelope MAIL FROM und die Envelope RCPT TO Adressen (*siehe 2.1, Seite 12*) ist zwingende Voraussetzung, dass sie voll qualifiziert sind, also mindestens einen Domain- und einen Top-Level-Domain-Teil.

Zeile 4, 5:

Sind diese Voraussetzungen erfüllt, so prüft Postfix, ob die übermittelten Domains von einem DNS Server aufgelöst werden können. Dies stellt sicher, dass die Adressen zumindest real vorhanden sind, schützt aber nicht vor Fälschungen.

Zeile 6:

Hier werden unbekannte lokale Empfängeradressen sofort abgewiesen, wenn Postfix final für die angegebene Domain zuständig ist, aber kein passender Eintrag in der Liste der bekannten EMail-Adressen vorhanden ist.

Zeile 7-9:

Diese Zeilen stellen eine erste Weichenstellung in der Bahndlung der SMTP-Verbindung dar. Sie überprüfen, ob ein verbundener Client zum eigenen Netzwerk gehört, durch ein vertrautes SSL-Zertifikat legitimiert ist oder eine erfolgreiche Authentifizierung (*siehe 2, Seite 11*) durchgeführt wurde. Trifft dies zu, werden keine weiteren Prüfungen vollzogen und der einliefernden EMail wird volles Vertrauen geschenkt.

Zeile 10:

Der hier aufgeführte Parameter ist der absolut wichtigste Parameter innerhalb des Regelwerks. Wer zuvor keine Berechtigung (eine der permit-Anweisungen) durch Postfix erlangt hat, kann keine EMail verschicken, für die Postfix nicht selbst final zuständig ist. Der Versuch, Empfänger außerhalb Postfix' zu adressieren wird sofort abgewiesen. Dieser Parameter verhindert also, dass das Mailsystem zu einem offenen Relay (*siehe 1.1.3, Seite 8*) wird.

Zeile 12:

Ein Mailsystem unterliegt nicht nur dem bereits mehrfach genannten RFC 2181, sondern einer ganzen Reihe anderer ebenfalls. In RFC 5321 und RFC 2142 ist unter anderem definiert, dass für jede EMail-Domain, für die Postfix final zuständig ist, bestimmte EMail-Adressen existieren müssen. Ferner muss gewährleistet sein, dass Schreiben an diese Adressen ungehindert zu einem Sachbearbeiter zugestellt werden. Sie dienen dem Zweck, den Mailserververwalter (Postmaster) über Probleme bei der Kommunikation mit anderen Anbietern zu informieren. Ein besonders Konto ist das Abuse-Konto. Versendet das eigene System Massenmails, so können entfernte Mailserver-Administratoren über diese Adresse Kontakt suchen und über den Vorfall berichten.

Zeile 13:

Die folgende Zeile definiert eine PCRE-Map. Sie listet weltweit bekannte Einwahlnetze (*siehe 1.1.2.1, Seite 7*) und führt zur Beendigung der SMTP-Verbindung. Diese Liste wurde über Jahre von verschiedensten Administratoren zusammengestellt. Bei Interesse an dieser Datei kann man sich gerne an die Mailingliste¹ des Postfich-Buch Autor's wenden. Sie kann hier aufgrund des großen Umfangs nicht abgedruckt werden.

Zeile 14-16:

Wie bereits beim SMTP-Protokoll erwähnt, muss der SMTP-Client dem MTA mittels EHLO ein Greeting schicken. Dieses muss zum einen mit dem Servernamen des Mailsystems überstimmen; zum anderen muss der Name im DNS passend mit Vorwärts- und Rückwärtsauflösung (A- und PTR-Ressource-Records) eingetragen sein. Diese Restriktionen sind sehr hart und können bei großen Setups zu Problemen führen. Da hier keine eindeutige Vorschrift in den RFCs existiert, ist die Entscheidung über den Einsatz der Parameter individuell zu treffen.

Zeile 19 und 24 werden in der Folgesektion genauer beschrieben.

Zeile 21, 22:

Es besteht die Möglichkeit, Postfix an DNSBL und RHSBL-Listen anzubinden. Wird ein SMTP-Client mit seiner IP-Adresse in einer RBL-Liste gefunden, so wird die Verbindung abgewiesen. Die direkte Einbindung solcher Abfragen in Postfix empfiehlt sich nur für eigene RBL-Listen.

¹Mailingliste des Postfich-Buches: postfixbuch-users@listi.jpberlin.de

4.1.1 Policy Delegation Services

Bestimmte Prüfungen kann Postfix nicht selbst oder nicht optimal durchführen. Es besteht jedoch die Möglichkeit, diese an andere Programme zu delegieren. Die Anbindung erfolgt in der Regel entweder über TCP/IP oder Unix-Sockets. Man nennt diese Schnittstellen Policy-Service-Delegation. Der angesprochene Dienst liefert die bereits weiter oben genannten Stati zurück.

In Zeile 19 wird der Dienst Policyd-weight angesprochen. Er führt einige Validierungen zum übergebenen EHLO-Namen und der Client-IP-Adresse durch und fragt eine Liste von RBL-Servern ab. Der Vorteil hierbei liegt in der Arbeitsweise des Services. Wird ein Mail-Client positiv auf eine RBL-Liste getestet, so bekommt er einen Punktwert. Für jeden weiteren Verstoß werden Punkte addiert. Überschreitet ein Client eine minimale Grenze, so wird *reject* als Resultat an Postfix zurückgesendet und als Folge die Verbindung beendet.

Ein Nachteil der RBL-Listen ist es, dass gelegentlich auch namenhafte EMail-Anbieter in solchen Sperrlisten landen. Würde man nun diese Listen direkt in Postfix (Ausnahme siehe Zeile 21 und 22) einbinden, hätte man relativ schnell Abweisungen auch gültiger Absender.

Zeile 24 behandelt das sog. selektive Greylisting. Die Idee hinter Greylisting liegt in der Annahme, dass Malware auf infizierten Computern möglichst klein im Codeumfang sein sollte. Dies impliziert, dass solche Bots keine umfassende Fehlerabfangung implementiert haben. Über einen Dienst wie Postgrey kann Postfix nach bestimmten Regeln (daher selektiv) ein Greylisting für eingehende Verbindungen aktivieren. Postgrey selbst führt eine kleine Berkley-Datenbank. Kommt eine Anfrage über Postfix an, so wird die übergebene Client-IP-Adresse gegen diese Datenbank geprüft. Existiert noch kein Eintrag, so wird dieser neu angelegt, mit einem Zeitstempel von 300 Sekunden versehen und eine Art Störfall (*siehe 2, Seite 11*) mit Fehlercode 421 an Postfix übergeben. Als Resultat wird die Verbindung abgebaut. Durch die fehlende Fehlerabfangung seitens der Bots wird hierdurch oft eine erneute Zustellung unterbunden.

Ein regulärer SMTP-Client wird nach einer gewissen Zeitspanne eine erneute Zustellung versuchen (Maximal ca. 5 Tage lang). Ist die Frist von 300 Sekunden überschritten, so schaltet Postgrey die Verbindung frei. Es wird aber nur ein *dunno* zurückgeliefert.

4.2 AMaViS als Pre-Queue Filter

Heutzutage müssen eingehende EMail leider auf Malware untersucht werden. Diese Aufgabe übernimmt AMaViS (Im Folgenden Amavis genannt). Die Funktionsweise wird in einer Folgesektion genauer beschrieben. hier soll lediglich die Einbindung in das Postfix-

Setup erläutert werden.

In der Datei `master.cf` wird Amavis als Pre-Queue Filter wie folgt eingebunden:

```
...
smtp      inet  n       -       -       -       smtpd
          -o smtpd_proxy_filter=localhost:10024
...

```

Listing 4.2: Datei `master.cf`

Nachdem die Client-Server-Verbindung alle Restriktionen durchlaufen hat, folgt laut SMTP-Protokoll die DATA-Phase. Im Proxy-Betrieb baut nun Postfix in Echtzeit eine Verbindung zu Amavis auf und „reicht“ den Mail-Content direkt an Amavis durch. Dieser prüft auf Viren und Spam und liefert *reject* oder *permit* zurück. Erst mit dieser Rückmeldung quittiert Postfix die gehaltene Verbindung. 250 Ok im Idealfall, oder eine 5xx-Fehlermeldung (*siehe 2, Seite 11*) bei erkannter Schad-Software.

Diese Methode ist eine optimale Lösung, da der eigene Server nicht zum Backscatter wird. Abgewiesene EMail-Adressen landen unabhängig von gefälschten EMail-Adressen direkt wieder beim Verursacher.

Es gibt in der Praxis auch eine andere Möglichkeit, EMail-Adressen komplett anzunehmen und dann im positiv-Fall markiert an die Anwender weiterzureichen. Diese Lösung ist problematisch, da sie gezeigt hat, dass Benutzer mit Filtern in EMail-Programmen so eher zu Mailverlusten tendieren, als durch eine harte und direkte Abweisung. Sollte einmal fälschlicherweise eine gültige EMail (Man sagt hierzu False-Positive) abgewiesen worden sein, so erfährt dies der Absender unmittelbar und kann durch die mitgelieferte Fehlermeldung des Mailsystems die Probleme beseitigen.

4.2.1 Rechtlicher Hintergrund

Die Arbeitsweise von Postfix im Pre-Queue-Verfahren bringt noch einen weiteren elementaren Vorteil. Rechtlich gesehen darf eine anvertraute EMail nicht mehr unterdrückt werden. Würde man die EMail also komplett annehmen, die Verbindung vollständig abbauen und anschließend feststellen, dass die EMail Malware enthielt, so müsste sie trotz dieses Umstands an den betroffenen Empfänger zugestellt werden.

Da aber im hier gezeigten Setup die Verbindung offen gehalten wird, hält Postfix die Status-Rückmeldung bis zum abgeschlossenen Test seitens Amavis zurück. Eine EMail gilt nach dem SMTP-Protokoll dann als vollständig anvertraut, wenn ein 250 Ok Statuscode übergeben wurde.

Die Unterdrückung von anvertrauten Nachrichten ist strafbar nach §206 StGB, Verletzung des Post- oder Fernmeldegeheimnisses. Für rechtlich verbindliche Auskünfte bitte ich einen Rechtsanwalt zu konsultieren.

5 AMaViS – A Mail Virus Scanner

Eine sehr wichtige Teilkomponente in einem Mailsystem bildet Amavis. In den letzten Jahren gab es unterschiedliche Entwicklerzweige dieser Software, von denen sich aber nur einer durchsetzen konnte. Spreche ich hier von Amavis, so ist immer das Softwarepaket `amavisd-new`[5] gemeint.

Es wird aktiv in der Programmiersprache Perl entwickelt und stellt in Postfix-Umgebungen inzwischen quasi eine Standard-Komponente dar.

Amavis übernimmt als Pre-Queue-Filter vier Teilaufgaben.

- Bad Header – Überprüfung der Kopfzeilen im übergebenen Mail-Header
- Banned – Überprüfung der Mime-Anhänge auf illegale Dateianhänge. Die Liste der unerwünschten Erweiterungen ist hierbei mit regulären Ausdrücken frei definierbar.
- Virus – Prüfung mittels beliebig vieler Virens Scanner. Im Regelfall reichen ein bis zwei Scanner aus.
- Spam – Untersuchung des Mail-Bodys (man nennt den eigentlichen Nachrichtenteil „Body“) auf Spam

Es gibt 4 verschiedene Reaktionsmöglichkeiten, falls ein Test positiv ausgewertet wurde.

- D_BOUNCE – Es wird eine Bounce-Mail generiert und an den Absender verschickt. Achtung Backscatter-Gefahr
- D_DISCARD – Die EMail wird angenommen, intern aber kommentarlos verworfen
- D_REJECT – Die Annahme der EMail wird verweigert
- D_PASS – Amavis fügt bestimmte Mail-Header-Zeilen hinzu und leitet die EMail weiter

Für jede der genannten Teilaufgaben wird die gewünschte Reaktion definiert.

Die Praxis zeigt, dass das prüfen von „Bad Header“ keine gute Idee ist, da schon doppelte Header-Zeilen als Fehler gewertet werden. Gerade bei Mailing-Listen kann dies aber öfters geschehen. Der Reaktionstyp ist hier *D_PASS*

Unerwünschte Dateianhänge und Spam-EMails werden generell mit *D_REJECT* in der Annahme zurückgewiesen. Viren dagegen werden kommentarlos gelöscht – Rückgabewert *D_DISCARD*. Wenn eine Virensignatur gefunden wurde, so ist ein False-Positive ausgeschlossen.

Eine weitere sehr nützliche Funktion von Amavis sind die sog. Policy-Banks. Dies sind Konfigurationsmakros innerhalb der Konfigurationsdatei, die nur in bestimmten Fällen zur Anwendung kommen. In diesen Makros lassen sich global gültige Einstellungen überladen. Damit Amavis eigene Anwender des Systems ermitteln kann, nutzt es z.B. eine Policy-Bank mit dem Namen MYUSERS.

Die Optionsvielfalt von Amavis ist auch hier gigantisch, so dass ich hier auf die Online-Dokumentation verweisen möchte.

5.1 Spamfilter mit SpamAssassin

Für die Untersuchung einer übergebenen EMail an Amavis wird ein weiteres Software-Produkt verwendet. Das Projekt SpamAssassin[6] stellt einen gleichnamigen Spam-Scanner – geschrieben in der Programmiersprache Perl – zur Verfügung. Amavis kann hierbei direkt über die API der Software auf die Scanner-Funktionen zugreifen.

SpamAssassin prüft zum einen anhand zahlreicher Dateien, in denen reguläre Ausdrücke enthalten sind, den Nachrichteninhalt und berechnet ähnlich dem bereits vorgestellten Policyd-weight einen Gesamt-Score (Punktwert). Liegt dieser wieder über einer bestimmten Grenze, so wird die EMail als Spam erkannt (stark vereinfachte Darstellung). Die regulären Ausdrücke sind in Makros zusammengefasst. Eine Auswertungssoftware liefert eine tägliche Ausgabe (Beispiel):

SpamAssassin Rule Hits: Spam						
Rank	Hits	% Msgs	% Spam	% Ham	Score	Rule
1	3	5.36%	100.00%	0.00%	2.834	PYZOR.CHECK
2	3	5.36%	100.00%	0.00%	0.001	DIGEST_MULTIPLE
3	3	5.36%	100.00%	0.00%	0.5	RAZOR2.CF_RANGE.51.100
5	3	5.36%	100.00%	0.00%	0.5	RAZOR2.CHECK
6	2	3.57%	66.67%	0.00%	4.199	FORGED.MUA.OUTLOOK
...						

Listing 5.1: Logwatch - Spam-Auswertung

Wie bereits erwähnt, fügt Amavis u.U. Header-Zeilen hinzu. Diese können dann etwa wie folgt aussehen:

```
X-Virus-Scanned: Debian amavisd-new at mx0.roessner-net.de
X-Spam-Flag: YES
X-Spam-Status: Yes, score=5.336 required=4.3 tests=[DIGEST_MULTIPLE=0.001,
MISSING_MID=0.001, PYZOR_CHECK=2.834, RAZOR2_CF_RANGE_51_100=0.5,
RAZOR2_CF_RANGE_E4_51_100=1.5, RAZOR2_CHECK=0.5]
```

Listing 5.2: Eingefügte Amavis X-Header

Bei besonders großen Mail-Account-Kontingenten kann Amavis und SpamAssassin auch an einen LDAP und/oder MySQL-Server angebunden werden. Auf diese Weise können komfortabel benutzerdefinierte Wünsche umgesetzt werden.

5.2 Anti-Viren Komponenten

Eine letzte Komponente, die hier vorgestellt wird, ist die Einbindung von Virenschannern. Es gibt derzeit einen bekannten, aber noch recht jungen Virenschanner namens ClamAV[7] aus der Open-Source-Welt. Dieser liefert schon recht gute Ergebnisse, erkennt aber leider bei weitem nicht alle Viren, so dass auf mindestens einen weiteren Virenschanner zugegriffen werden sollte. Setzt man einen privaten EMail-Server ein, so besteht hier die Möglichkeit, die Personal-Edition von AntiVir¹[8] als Zweitschanner einzusetzen.

Die Kombination der beiden Scanner stellt eine solide Basis dar. Sobald die Virenschanner installiert und eingerichtet wurden, erkennt Amavis diese automatisch. Eine weitere Konfiguration ist hier nicht notwendig.

Bei der Verwendung von Antivir sei darauf hingewiesen, dass der Mailserver bei hohem EMail-Aufkommen eine deutlich stärkere Last produziert, als wenn nur ClamAV eingesetzt wird. Letzterer verfügt über einen Daemon, der einen Unix-Socket zur Verfügung stellt, mit dem Amavis kommunizieren kann. Bei Antivir muss jedesmal erneut der Programmcode in den Hauptspeicher geladen werden, da es leider nur als Konsole-Anwendung vorliegt.

¹Download-Version für Linux – Stand 20.10.2008

http://www.chip.de/downloads/AntiVir-Personal-Free-Antivirus-fuer-Linux_23188958.html

6 Fazit

Der Aufbau, die Konzeption und Verwaltung eines Mailservers ist im Detail sehr komplex und zeitaufwändig. Die hier dargelegte Theorie und die kurze Einführung in die wichtigsten Komponenten beschreibt sprichwörtlich nur die Spitze eines Eisbergs. Je nach Einsatzgebiet und Randbedingungen kann das Setup extrem variieren.

Da die genannten Komponenten fast ausschließlich aus dem Open-Source Umfeld stammen, existiert zum einen eine große Community (Gemeinschaft), zum anderen sind die Projekte alle im Internet mit teils hervorragender Dokumentation vorhanden.

Dieses Dokument soll Ihnen helfen, Gefahren und konzeptionelle Sicherheitsrisiken im Umgang mit E-Mails besser zu verstehen. Ein leichtfertiges Aufsetzen eines Mailservers kann fatale Folgen mit sich bringen. Sollten Sie ein weitergehendes Interesse an dieser Thematik gewonnen haben, so empfehle ich Ihnen als Einstieg „Das Postfix-Buch“ [3]. Dieses fasst fast alle genannten Themen zusammen. Der Autor ist Rechtsanwalt und LPIC-1 und LPIC-2 Absolvent und gilt als Experte in diesem Bereich. Das Buch liest sich zudem außerordentlich angenehm und ist nicht abstrakt verfasst. Mit 776 Seiten Umfang lässt die derzeit aktuelle Auflage 3 keine Wünsche offen.

Teil III
Anhang

A Verwandte RFCs

- RFC 2821 – SIMPLE MAIL TRANSFER PROTOCOL
- RFC 2822 – INTERNET MESSAGE FORMAT
- RFC 1652 – SMTP Service Extension for 8bit-MIMEtransport
- RFC 1845 – SMTP Service Extension for Checkpoint/Restart
- RFC 1870 – SMTP Service Extension for Message Size Declaration
- RFC 1894 – An Extensible Message Format for Delivery Status Notifications
- RFC 1985 – SMTP Service Extension for Remote Message Queue Starting – ETRN
- RFC 2034 – SMTP Service Extension for Returning Enhanced Error Codes
- RFC 2487 – SMTP Service Extension for Secure SMTP over TLS
- RFC 2505 – Anti-Spam Recommendations for SMTP MTAs
- RFC 2554 – SMTP Service Extension for Authentication
- RFC 2606 – Reserved Top Level DNS Names
- RFC 2852 – Deliver By SMTP Service Extension
- RFC 2920 – SMTP Service Extension for Command Pipelining
- RFC 3030 – SMTP Service Extensions for Transmission of Large and Binary MIME Messages
- RFC 3207 – SMTP Service Extension for Secure SMTP over Transport Layer Security
- RFC 3461 – SMTP Service Extension for Delivery Status Notifications (DSNs)
- RFC 3463 – Enhanced Status Codes for SMTP
- RFC 3464 – An Extensible Message Format for Delivery Status Notifications
- RFC 3700 – Internet Official Protocol Standards
- RFC 4409 – Message Submission for Mail
- RFC 5336 – SMTP Extension for Internationalized Email Addresses

Glossar

A-Ressource-Record

Eintrag in einem Nameserver. Er bildet ein Mapping von Rechnernamen zu IP-Adressen ab. 22

Backscatter

engl. Rückstreuung bezeichnet Server, die bereits empfangene E-Mails nachträglich ablehnen. Dabei begeht der handelnde Mailserver einen fatalen Fehler, denn er vertraut ungeprüft den angegebenen Absenderadressen, welche i.d.R. aber gefälscht sind. 9

chroot

chroot ist die Fähigkeit von *unix-basierten Betriebssystemen, Prozesse in einer Verzeichnisebene zu kapseln und damit den Zugriff auf Dateien außerhalb dieser zu sperren. 16

DDoS

Distributed Denial of Service. DDoS bezeichnet Angriffe von verteilten Computern, die zeitgleich einen Dienst eines Servers in Anspruch nehmen. Die Folge daraus ist, dass der Dienst überflutet wird und im schlimmsten Fall nicht mehr nutzbar ist. 6

DNS

Domain Name Service. Das DNS verwaltet IP-Adressen. Es ordnet Domain Names numerischen IP-Adressen zu. 15

ESMTP

Extended SMTP. Aktueller SMTP-Standard nach RFC 2821. 8

Exploit

Unter dem Begriff Exploit versteht man das Ausnutzen einer Sicherheitslücke im Betriebssystem, durch die eine Schad-Software unerlaubten Zugriff auf das System erlangt. 6

IRC

Internet Relay Chat. Das IRC stellt einen Service zum Chatten bereit. 6

Kernel

Als Kernel bezeichnet man denjenigen Teil des Betriebssystems, der ausgehend vom Bootloader geladen wird. Er ist der Hauptbestandteil des Betriebssystems. 16

LDAP

Lightweight Directory Access Protocol. Objektorientierte Datenbank, welche unter anderem besonders gut für personenbezogene Daten geeignet ist. 15

Mailing-Liste

Eine Mailing-Liste stellt einen EMail-Verteiler dar. Abonnenten einer Liste können an die Liste schreiben. Alle anderen Abonnenten erhalten dann diese. Und vice versa. 26

Malware

Malware ist ein englischer Begriff und bezeichnet Schad-Software. Hierzu gehören zum Beispiel Viren, Trojaner und Würmer. 8

MySQL

Bei MySQL handelt sich um einen relationalen Datenbankserver. 15

PCRE

Perl Compatible Regular Expressions. PCRE-Tabellen können für eine Mustererkennung innerhalb von Postfix genutzt werden. 7

Perl

Perl gilt als eine sehr sichere Skript-Programmiersprache, da sie häufige Fehlerquellen wie Bufferunderrun- oder Bufferoverflow-Fehler konzeptionell verhindert. 19

PTR-Resource-Record

Pointer-to-Record ist ein Feld im Nameserver. Es bildet eine IP-Adresse auf einen Rechnernamen ab. 22

Relay

Als Relay wird ein Mailserver bezeichnet, der E-Mails annimmt und an andere Mailserver im Internet weiterleitet. 8

SMTP

Simple Mail Transfer Protocol Standard-Protokoll bei der Übertragung von E-Mails. 8

STARTTLS

Start TLS, Transport Layer Security. Bezeichnet das nachträgliche Verschlüsseln einer zuvor ungesicherten TCP/IP-Verbindung. 11

Stichwortverzeichnis

- AMaViS, 23, 26
- Authentifizierung, 11
- Backscatter, 9
- Botnetze, 6
- chroot, 19
- Datenhaltung, 15
- Datenstruktur, 15
- DNSBL, *siehe* Nameserver
- dunno, 21
- Envelope
 - MAIL FROM, *siehe* SMTP
 - RCPT TO, *siehe* SMTP
- Exploit, 6
- False Positive, 24
- Greeting, 10
- IRC, 7
- Late Bounce, 9
- LDAP, 15
- local, 15
- Mail-Header, 13
- Mail-Routing, 13
- Map, 15
- master, 19
- MTA, 10
- MUA, 10
- MySQL, 15
- Nameserver, 16
- permit, 21
- Policy services, 23
- policyd-weight, 23
- postgrey, 23
- proxymap, 15
- RBL, 16
- Received, 13
- reject, 21
- Relay, 8
- relay, 15
- RHSBL, *siehe* Nameserver
- SMTP, 10
 - Protokoll, 10
- smtp, 19
- smtpd, 19
- smtpd_recipient_restrictions, 20
- SMTPS, 11
- Spam, 5
- SpamAssassin, 27
- STARTTLS, 11
- UBE, *siehe* Spam
- UCE, *siehe* Spam
- virtual, 15
- Zombie, 6

Literaturverzeichnis

- [1] RFCs unter <http://www.faqs.org/rfcs/>
- [2] Postfix-Projekt-Dokumentation: <http://www.postfix.org/documentation.html>
- [3] Peer Heinlein: *Das Postfix-Buch*, Sichere Mailserver mit Linux,
ISBN 978-3-937514-50-5
- [4] Ralf Hildebrandt, Patrick-Ben Koetter: *Postfix*, Einrichtung, Betrieb und Wartung,
ISBN 3-89864-350-6
- [5] AMaViS-Projekt: <http://www.ijs.si/software/amavid/>
- [6] SpamAssassin-Projekt: <http://spamassassin.apache.org/>
- [7] Clam-AV-Projekt: <http://www.clamav.net/>
- [8] Avira GmbH: <http://www.avira.com/de/pages/index.php>