FAKULTÄT FÜR !NFORMATIK

Entwicklung eines Spam-Filter Gateways basierend auf Open-Source-Projekten

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

im Rahmen des Studiums

Informatik

eingereicht von

Thomas Rainer
Matrikelnummer 9526718

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung:
Betreuer: o.Univ.Prof. Dipl.-Ing. Dr.techn. A Min Tjoa
Mitwirkung: Univ.Ass. Dipl.-Ing. Dr.techn. Edgar Weippl

Wien, 15.09.2008

(Unterschrift Verfasser) (Unterschrift Betreuer)

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbständig und ohne fremde Hilfe verfasst, andere als die vorliegenden Quellen nicht benützt und die den benutzen Quellen wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

.....
Wien, am 15. September 2008

Zusammenfassung

Spam-Nachrichten stellen ein immer größeres Problem im E-Mail-Verkehr dar. E-Mail- Provider stehen zunehmend in der Pflicht aktiv gegen Spam vorzugehen. Zur Anwendung kommen neben statischen Verfahren, wie DNS-basierte Blacklists oder ähnliches, auch selbstlernende Filter, die sich in der Praxis als sehr effektiv erwiesen haben.
Um die eigene E-Mail-Adresse erst gar nicht in die Hände von Spammern gelangen zu lassen, existieren verschiedene Maßnahmen, die das Ausspähen von E-Mail-Adressen auf Websites verhindern. Ein geändertes Benutzerverhalten und die Verwendung von Wegwerf-Adressen erschweren die Arbeit der Spammer zusätzlich.
Diese Arbeit beschäftigt sich mit der Integration mehrerer Filtermethoden in eine einheitliche Benutzerumgebung. Hierfür werden sämtliche Benutzerdaten und Filterinformationen in einer MySQL-Datenbank abgelegt. Das Einsehen der Filteraktivitäten und das Bearbeiten der Einstellungen erfolgt mit Hilfe eines einfach zu bedienenden Web-Interface.
Durch realitätsnahe Tests mit unterschiedlichen Filter-Einstellungen lassen sich die verschiedenen Verfahren direkt miteinander vergleichen. Aufgrund der erzielten Ergebnisse kann das Spam-Filter Gateway durchaus auch in professionellen Umgebungen eingesetzt werden.

Inhaltsverzeichnis

1. Einleitung

1.1. Einführung

Die elektronische Post begann ihren Siegeszug Ende der 1980er Jahre. Wurde dieses Medium anfänglich nur auf universitärer Ebene genutzt, so ist es mittlerweile in Firmen und auch im Privatleben kaum mehr wegzudenken.
Bestand in den Anfangszeiten der E-Mail-Kommunikation noch keinerlei Handlungsbedarf gegen Spam vorzugehen, so würde heute ohne geeignete Gegenmaßnahmen das Medium E-Mail nahezu unbrauchbar werden. Hunderte unerwünschte Nachrichten pro Tag sind nun keine Seltenheit mehr. Laut der monatlichen Studie von Messagelabs [19] sind bereits über 76 % aller eMail-Nachrichten Spam. Andere Quellen [51] sprechen sogar von über 90 %. Diese Zahlen machen deutlich, dass ohne Maßnahmen gegen unerwünschte Nachrichten das Medium E-Mail in seiner Existens bedroht wäre. Entschieden sich Unternehmen vor Jahren aus Kostengründen ihre Kommunikation in elektronischer Form abzuwickeln, so sehen sich diese Firmen mit zusätzlichen Ausgaben konfrontiert, die sich durch die Spam- und Virenabwehr ergeben. Aber auch Privatanwender sehen sich zusehens mit diesem Problem konfrontiert.
Es wurden verschiedene Versuche unternommen, sowohl technischer als auch juridischer Art, um den Spammern das Leben zu erschweren. So setzt heute ein Großteil der Provider spezielle Filter ein, die der Spamflut Herr werden sollen. Immer neue Arten von Spam machten und machen aber eine ständige Adaptierung der Spam-Filter notwendig, da Spammer immer neue Methoden austüfteln, um die bestehenden Filter zu umgehen.
Trotz erheblicher Anstrengungen verschiedener Forschungsgruppen ist es bisher nicht gelungen, ein Patentrezept gegen unerwünschte Nachrichten zu finden, jedoch lassen sich durch die Kombination mehrerer Filtermethoden recht ansehnliche Resultate erzielen.

In dieser Arbeit werden keine neuen Filtertechnologien vorgestellt. Es wird vielmehr versucht, einen funktionsfähigen und frei verfügbaren Prototypen zu schaffen, der verschiedene Anti-Spam-Methoden miteinander verknüpft und dabei die Nachteile bereits existierender Filterlösungen ausmerzt. Diese betreffen weniger die Erkennungsraten, sondern vielmehr die fehlende Transparenz und schlechte Administrierbarkeit.
Im praktischen Teil dieser Arbeit wird ein besonderes Augenmerk auf eine einfache, leicht verständliche Benutzerschnittstelle gelegt. Hierfür wurde ein Web-Interface geschaffen, das jedem einzelnen User erlaubt, individuelle Filtereinstellungen zu treffen und sämtlichen Aktivitäten des Spam-Filters zu überwachen.

1.2. Gliederung

Nach einer technischen Einführung in die Funktionsweise von E-Mail-Systemen in Kapitel 2, wird in Kapitel 3 näher auf das Phänomen Spam eingegangen. Dies beinhaltet die Darlegung der allgemeinen Spam-Problematik und die Vorgehensweise von Spammern. Auch auf die Auswirkungen der unerwünschten Werbesendungen wird kurz eingegangen. In Kapitel 4 widmet sich diese Arbeit vorbeugenden Maßnahmen. Diese sollen die Spam-Problematik bereits im Ansatz bekämpfen und die Adressgewinnung für Spammer erschweren. Im darauf folgenden 5. Abschnitt werden einige technische Maßnahmen auf der Empfängerseite genauer durchleuchtet. Diese reichen von recht einfachen statischen Methoden bis hin zu selbstlernenden Filtern. Dabei wird jeweils auf die Vor- und Nachteile der Filtermethoden eingegangen. Abschnitt 7 behandelt die konkrete Implementierung eines voll funktionstüchtigen Spam-Filter Gateways samt einer Benutzerschnittstelle für den Endanwender. Eine detaillierte Installationsanleitung findet sich im Anhang dieser Arbeit. In Kapitel 8 folgt die Analyse der erzielten Ergebnisse anhand der Auswertung der Log-Files. Unterschiedliche Filter-Einstellungen sollen dabei helfen, die Effizienz der einzelnen Methoden zu vergleichen. Im abschließenden Kapitel 9 wird die Arbeit kurz zusammengefasst. Des weiteren werden mögliche Strategien und Verbesserungsmöglichkeiten aufgezeigt.

1.3. Danksagung

An dieser Stelle möchte ich mich vor allem bei meinem Betreuer Herrn Dr. Edgar Weippl bedanken, der mir bei der Erstellung des Prototyps weitgehend freie Hand ließ und mich so gut es ging unterstützte. Ein großes Dankeschön auch an Mag. David Huemer und Thomas Mandl, die durch ihre positiven Anregungen zum Gelingen dieser Arbeit beigetragen haben.
Bedanken möchte ich mich auch ganz herzlich bei meinen Eltern, die mir mir dieses Studium erst möglich gemacht haben.
Ein besonderer Dank geht an die Firma KONZEPT für die Zurverfügungstellung sämtlicher technischer Ressourcen, ohne die ein realitätsnaher Testbetrieb des Prototypen nicht möglich gewesen wäre.

2. E-Mail Grundlagen

Das Medium E-Mail hat sich in den letzten Jahren zu einem immer wichtigeren Instrument der Nachrichtenübermittlung entwickelt. Mit wenigen Klicks können weltweit blitzschnell und kostenlos, von den Kosten für den Internetzugang mal abgesehen, E-Mails versandt werden. Ein wichtiger Grund, warum sich die Kommunikation via E-Mail explosionsartig verbreitet hat, sind die einfachen, gut dokumentierten und seit vielen Jahren unveränderten Standards, die der Technik zu Grunde liegen. Es existieren eine Vielzahl von Programmen auf unterschiedlichen Plattformen mit sehr einfachen, leicht verständlichen Benutzeroberflächen, die den Versand von E-Mails zum Kinderspiel machen.

2.1. Aufbau einer E-Mail

Der Aufbau einer E-Mail ist im RFC 2822 [84] festgelegt. Eine E-Mail besteht aus einem „Envelope“ mit Informationen über den Absender und Empfänger und dem „Inhalt“, der wiederum aus einem Header und einem Body besteht [91]. Der E-Mail-Envelope lässt sich dabei durchaus mit dem Briefumschlag eines herkömmlichen Briefes vergleichen, der E-Mail-Body mit dem eigentlichen Briefinhalt.
Erlaubt sind nur Textzeichen (7 Bit ASCII-Zeichen). Dieser eingeschränkte Zeichensatz macht eine spezielle Kodierung von Datenanhängen (siehe E-Mail-Body) notwendig.

2.1.1. E-Mail-Envelope

Der E-Mail- oder SMTP-Envelope enthält Informationen, die für die korrekte Zustellung des Inhalts notwendig sind. Er wird als Serie von SMTP-Protokolleinheiten

übermittelt und muss zumindest den Absender (Envelope-Sender) und einen oder mehrere Empfänger (Envelope-To) enthalten. Der Envelope-Sender muss folgendermaßen aussehen:

```
MAIL FROM: <user@domain.com>
```

Nicht erlaubt ist z. B.:

```
MAIL FROM: user@domain.com
```

oder:

```
MAIL FROM: Joe User <user@domain.com>
```

Ein Absender ist deshalb vonnöten, um Berichte über Fehler oder Nichtzustellbarkeit zurücksenden zu können. Unzustellbarkeits-Nachrichten werden mit leerem Envelope-Sender „<>“ versandt, um Endloschleifen zu vermeiden. Optional können im E-Mail-Envelope noch Informationen über Protokollerweiterungen angegeben werden.
Generell ist der SMTP-Envelope für den Endanwender nicht einsehbar, jedoch werden einige Informationen, wie Absender und Empfänger in den Nachrichten-Header übernommen. Beim Versenden von elektronischen Nachrichten übernimmt der E-Mail-Client die Übermittlung des SMTP-Envelopes. Die notwendigen Informationen werden dabei dem E-Mail-Header entnommen [62].

2.1.2. E-Mail-Header

Der E-Mail-Header besteht aus mehreren Zeilen, die jeweils ein Schlüsselwort und den dazugehörigen Wert - getrennt durch einen Doppelpunkt - enthalten. Jede Zeile wird mit einem Zeilenumbruch abgeschlossen. Die Reihenfolge der Einträge ist dabei beliebig. Verpflichtend enthalten sein müssen der Absender („Date:“) und der Erstellzeitpunkt („Date:“) der E-Mail [75]. Der Header-Absender darf nicht mit dem Envelope-Sender verwechselt werden und muss nicht mit diesem übereinstimmen.
Weiters können noch eine Reihe optionaler Informationen im Header enthalten sein. Einige davon sind:

- „To:“ - Empfänger
- „CC:“ - weitere Empfänger
- „BCC:“ - Blind Carbon Copy (Blindkopie)
- „Subject:“ - Betreff

Es können auch nicht standardisierte Informationen in den Header einfügt werden. Einzige Voraussetzung hierfür ist, dass dem Schlüsselwort ein „X“ vorangestellt wird. Beispielsweise fügen einige Virenprogramme nach erfolgter Überprüfung eine entsprechende Zeile ein oder Spam-Filter markieren die Nachricht im Header als Spam. Der E-Mail-Header enthält keine für die technische Übermittlung notwendigen Informationen. Der technischer Absender und Empfänger werden vielmehr durch den Envelope-Sender und Envelope-To angegeben. So spielt es z. B. für die Zustellung keine Rolle, welche Adresse im Header unter „To:“ angegeben wird.
In der Regel wird der E-Mail-Header durch den E-Mail-Client erstellt und dann im Laufe der Übermittlung von Mail-Servern und Filterprogrammen ergänzt oder verändert. Header-Daten können genauso wie Envelope-Daten nicht als authentisch vorausgesetzt werden, da sie beliebig durch den Absender verändert werden können. Spammer z. B. benutzen in der Regel keine gültigen Angaben beim Mail-Versand, um die wirkliche Herkunft der Nachrichten zu verschleiern.

Beispielsweise könnte ein E-Mail-Header folgendermaßen aussehen:

```
Reply-To: <example@example.com>
From: <example@example.com>
To: <rcpt@test.com>
Return-Path: <example@example.com>
Delivered-To: rcpt@test.com
Received: (qmail); 7 Apr 2008 10:17:24 -0000
Received: from mail.example.com (HELO mail.example.com)
MIME-Version: 1.0
Subject: Test Nachricht
Date: Mon, 7 Apr 2008 12:15:44 +0200
Content-Type: text/plain; charset="iso-8859-1"
X-Mailer: pine
```

2.1.3. E-Mail-Body

Der E-Mail-Body folgt durch eine Leerzeile getrennt auf den E-Mail-Header und enthält die eigentliche Nachricht. Die Größe einer E-Mail und somit des E-Mail-Bodys ist grundsätzlich nicht begrenzt, jedoch erlauben die wenigsten Mail-Server den Versand von beliebig großen Nachrichten. Erlaubt sind nur Zeichen aus dem 7-Bit ASCII-Zeichensatz [84], wodurch die Codierung von Sonderzeichen bzw. Anhänge notwendig wird. Dies erledigt in der Regel der E-Mail-Client transparent für den Benutzer.

Multipurpose Internet Mail Extensions (MIME)

In den Anfangsstadien des Internet, bestanden E-Mails lediglich aus Text, geschrieben in Englisch mit dem 7-Bit US-ASCII Zeichensatz [30]. Die Verbreitung des Internets auf der ganzen Welt machten neue Standards notwendig, die einen weitreichenderen Zeichensatz unterstützen. Weiters wuchs das Bedürfnis nicht nur Text, sondern auch Bilder und Dokumente via E-Mail zu versenden. Die Mime Kodierung (beschrieben in den RFCs 2045 [72] bis 2049 [71]) macht das Versenden verschiedenartiger Inhalte möglich. Die Idee dahinter war, den verwendeten Zeichensatz unverändert zu lassen, um vorhandene Mail-Server und Protokolle weiterhin verwenden zu können. Lediglich die Client-Programme mussten erweitert und angepasst werden. Der Mime Standard definiert fünf weitere Schlüsselwörter für den E-Mail-Header:

- „MIME-Version:“ - Identifiziert die verwendete Mime-Version
- „Content-Description:“ - Beschreibung des Inhalts
- „Content-Id:“ - Eindeutige Kennung
- „Content-Transfer-Encoding:“ - Verwendete Kodierung
- „Content-Type:“ - Typ und Format des Inhalts

Interessant sind vor allem die Felder *Content-Type* und *Content-Transfer-Encoding*. Der Schlüssel *Content-Type* legt fest, um welche Art von Nachricht es sich handelt. Er besteht aus dem Tuel Type/Subtype, wobei *Type* die allgemeine Gruppe

definiert und *Subtype* das exakte Datenformat. Ursprünglich wurden sieben verschiedene Typen definiert. Mittlerweile sind eine Menge neuer dazugekommen. Einige davon sind z. B. *Text/Plain*, *Image/Jpeg* und *Multipart/Mixed*.

Der Schlüssel *Content-Transfer-Encoding* beschreibt die verwendete Kodierung. Für die Kodierung von Binärdaten verwendet man am Besten die base64 Kodierung. Da wie bereits erwähnt sämtliche E-Mail-Protokolle den 7-Bit ASCII Zeichensatz verwenden, muss aus 8 Bit Binärdaten eine Zeichenfolge geformt werden, die lediglich aus den 7-Bit ASCII Zeichensatz besteht. Dabei werden 24 Bit Blöcke in 6-Bit Einheiten aufgesplittet und als ASCII Zeichen übertragen. Die Kodierung lautet „A“ für 0, „B“ für 1 usw. gefolgt von den 26 Kleinbuchstaben, den zehn Zahlen und schlussendlich „+“ und „/“ für 62 und 63. Besteht die letzte Einheit nur aus 8 oder 16 Bits, so wird die Sequenz mit == oder = aufgefüllt [91]. Die Datenlänge vergrößert sich durch das base64 Verfahren um ca. ein Drittel.
Für Formate die fast ausschließlich aus ASCII-Zeichen bestehen, ist das base64 Verfahren höchst ineffizient. Vielmehr wird hier die *quoted-printable-Kodierung* angewandt. Zeichen, die im 7-Bit ASCII-Zeichensatz vorkommen, werden nicht kodiert. Alle anderen Zeichen werden durch ein „=“ gefolgt durch den Hexadezimalwert des Zeichens angegeben. Kodiert man z. B. das Zeichen „ü“ mittels quoted-printable unter Verwendung des Zeichensatzes ISO 8859-1, würde das die Zeichenfolge „=FC“ ergeben.

Beispiel für einen E-Mail-Body:

```
Subject: Test Nachricht
MIME-Version: 1.0
Content-type: multipart/mixed; boundary="example-1"

--example-1
Content-type: text/plain; charset=utf-8

Hier steht der Text dieser Beispielnachricht.

--example-1
Content-type: image/gif; name="bild.gif"
Content-Transfer-Encoding: base64
```

```
R0lGODlhIgFGAOYAAABmmYCruf///\\
zCIpa/S3QCZzECZtgCNvN/p7CB3lKDDzmCZrACGsxB2

--example-1--
```

Secure / Multipurpose Internet Mail Extensions(S/MIME)

S/MIME [82] ist ein Standard für die Verschlüsselung und Signierung von MIME-gekapselten Daten. Basierend auf dem weit verbreiteten MIME-Standard, bietet S/MIME die Möglichkeit zur Wahrung der Authentizität, Integrität und Vertraulichkeit im Umgang mit elektronischen Daten. S/MIME ist grundsätzlich nicht auf E-Mails beschränkt, es kann in jedem Transportmechanismus benutzt werden, das MIME Daten unterstützt, wie z. B. HTTP. Um Daten zu verschlüsseln bzw. zu signieren ist ein gültiges X.509-Zertifikat [52] notwendig.
S/MIME definiert zwei neue *Content-Types* für Mime. Das *Multipart/Signed*-Format zur Signierung einer Mail und das *Multipart/Encrypted*-Format zu deren Verschlüsselung.
Als Alternative zu S/MIME kann auch das OpenPGP [25] verfahren benutzt werden. Beide Verfahren sind jedoch nicht miteinander kompatibel.

2.2. Funktionsweise

Im vorigen Kapitel haben wir den Aufbau einer elektronischen Nachricht kennengelernt. Für das Senden und Empfangen sind zusätzliche Regeln und Protokolle vonnöten. Vom Computer des Senders und Empfängers einmal abgesehen, sind in der Übermittlung von Nachrichten mehrere Server involviert, die mit Hilfe von standardisierten Protokollen miteinander kommunizieren [91]. Der User benutzt in der Regel einen E-Mail-Client (Mail User Agent - MUA), um E-Mails zu versenden bzw. zu empfangen. Die Hauptaufgabe des MUA beim Versenden von E-Mails besteht darin, Nachrichten korrekt zu formatieren, um notwendige Informationen zu ergänzen und anschließend mittels standardisiertem Protokoll an den Postausgangsserver weiterzuleiten. Der Message Transfer Agent (MTA) seinerseits ist für die korrekte Übermittlung der Nachricht zuständig.

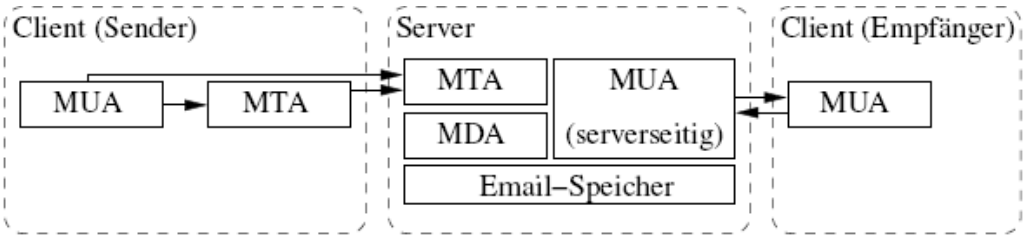

Abbildung 2.1.: Kontrollfluss vom Absender zum Empfänger

Der oder die Empfänger sind durch die angegebenen E-Mail-Adressen eindeutig spezifiziert. Mittels DNS [68] (MX-Record) wird der Rechner bestimmt, zu dem die Nachricht übertragen werden soll. Die Übermittlung erfolgt in der Regel nicht direkt sondern über Gateways. Sensible Daten wie z. B. Kreditkartennummern oder Passwörter sollte man deshalb niemals unverschlüsselt übertragen. Eine E-Mail ist etwa so gemein, wie eine Postkarte, die jeder Postbote lesen kann. Ist die Nachricht am Ziel-Rechner angekommen, kümmert sich der Message Delivery Agent (MDA) um das Ablegen der E-Mail im richtigen Postfach. Fragt nun der E-Mail-Client das Postfach ab, wird die gespeicherte Nachrichten an diesen übermittelt.
Nachfolgend wird kurz auf die häufigst verwendeten Protokolle eingegangen, ohne dabei allzu sehr in die Tiefe zu gehen. Ein grundlegendes Verständnis ist jedoch notwendig, um die Vorgehensweise von Spammern nachvollziehen zu können.

2.2.1. Simple Mail Transfer Protocol - SMTP

Das SMTP-Protokoll [62] ist ein einfaches ASCII Protokoll, das zur Übermittlung der Nachrichten verwendet wird. Da in der Regel der Client-Computer nicht ständig mit dem Internet verbunden ist, wird die Zustellung der Nachrichten von Servern übernommen. Der Client baut dabei eine SMTP Verbindung zum Postausgangsserver auf, authentifiziert sich gegebenenfalls und übermittelt die zu verschickende Nachricht. Auch wenn die Nachricht noch nicht endgültig zugestellt wurde, kann der Client die Verbindung mit dem SMTP-Server bereits abbrechen. Der Mail-Server kümmert sich im Hintergrund um die weitere Zustellung. Kann diese nicht erfolgreich abgeschlossen werden, so wird dem Absender eine entsprechende Meldung zugesandt. Nachfolgend ein Beispiel für ei-

ne SMTP-Konversation. Daten, die vom Client gesendet wurden, sind mit einem „C“ markiert, Daten vom Server mit einem „S“.

```
1  S: 220 mail.test.at SMTP server ready
2  C: HELO xyz.at.
3  S: 250 xyz.at., send us your mail but not your spam
4  C: MAIL From:<bob@xyz.at>
5  S: 250 <bob@xyz.at> Sender ok
6  C: RCPT To:<alice@test.at>
7  S: 250 <alice@test.at> Recipient ok
8  C: RCPT TO:<tom@test.at>
9  S: 250 <tom@test.at> Recipient ok
10 C: DATA
11 S: 354 Send mail, end with a "`."'
12 C: Hallo Alice, hallo Tom!
13 C: Beispiel für den Mail-Versand mit SMTP.
14 C: Bob
15 C: .
16 S: 250 message accepted
17 C: QUIT
18 S: 221 test.at closing connection
```

Beispiel für einen SMTP-Dialog

Im obigen Beispiel wird die Nachricht an zwei Empfänger gleichzeitig versandt.

2.2.2. Extented Simple Mail Transfer Protocol - ESMTP

In den Anfangszeiten des Internets war es noch nicht notwendig, Postausgangsserver mit Authentifizierung zu versehen. Jeder SMTP-Server war ein so genannter „Open Relay“, d. h. diese Server akzeptierten beliebige SMTP-Verbindungen und leiteten die E-Mails an die zuständigen Mail-Server weiter. Erst das Aufkommen von Spam-E-Mails machte zusätzliche Mechanismen notwendig.
1995 wurde das SMTP-Protokoll [62] mit mehreren Erweiterungen ausgestattet, unter anderem kamen folgende SMTP-Befehle hinzu: *STARTTLS* und *AUTH* [63]. Mit *STARTTLS* wurde ein Verfahren eingeführt, das die sichere Kommunikation beim E-Mail-Transport ermöglicht. *AUTH* erweitert das SMTP-Protokoll mit der Möglichkeit der Authentifizierung des Clients am Mail-Server.

```
1  S: 220 mail.test.at ESMTP server ready
```

```
2 C: EHLO client.at
3 S: 250 xyz.at., send us your mail but not your spam
4 S: 250 AUTH CRAM-MD5 DIGEST-MD5
5 C: AUTH CRAM-MD5
6 S: 334 PENCeUxFREJoNIKlHJHhNWitOMjNGNndAZWx3b29kLmlubm9zb2Z0LmNvbT4
     =
7 C: ZnJlZCA5ZTk1YWVlKNULnURNDBhZjJiODRhMGMyYjNiYmFlNzg2ZQ==
8 S: 235 Authentication successful.
```

SMTP-Dialog mit Authentifizierung

2.2.3. Post Office Protocol - POP

Das Post Office Protokoll in der Version drei ist im RFC 1939 definiert. Dieses weit verbreitete Protokoll ist sehr einfach gehalten und benötigt keine ständige Verbindung zum Mail-Server. Es ist daher ideal für User, die nicht ständig mit dem Internet verbunden sind. Es beschränkt sich auf das schlichte Auflisten, Abholen und Löschen der Nachrichten vom E-Mail-Server. Das Versenden von Nachrichten ist in diesem Protokoll nicht vorgesehen. Es ist wie fast alle Mail-Protokolle ASCII basierend und unterstützt die Authentifizierung des Benutzers über Benutzernamen und Passwort. Wurde ursprünglich nur eine unverschlüsselte Anmeldung unterstützt, so existieren nun diverse Verfahren, die eine sicherere Authentifizierung gewährleisten.

2.2.4. Internet Message Access Protocol - IMAP

Ermöglicht das POP Protokoll lediglich das Auflisten, Abholen und Löschen von Nachrichten auf dem Server, so bietet das IMAP Protokoll weitreichende Möglichkeiten, um E-Mails auf dem Server zu verwalten [29]. Die Funktionalität entspricht der einer lokalen Mailbox. Im Gegensatz zum POP-Protokoll verbleiben die Nachrichten in der Regel auf dem Server. Es besteht zwar auch beim Zugang über POP die Möglichkeit, die Nachrichten auf dem Server zu belassen, jedoch kann nicht festgestellt werden, ob die Nachricht schon gelesen, bearbeitet oder beantwortet wurde. Mittels IMAP besteht auch die Möglichkeit, eine Ordnerstruktur auf dem Server anzulegen, so wie es der User von einer lokalen Mailbox gewohnt ist.

Nicht alle Internet-Provider stellen dem User das IMAP-Protokoll zur Verfügung, da zum einen die technische Bereitstellung weitaus anspruchsvoller ist, zum anderen wird viel mehr Speicherplatz für die Speicherung der E-Mails benötigt. Ein weiterer Grund, warum das IMAP-Protokoll nicht so eine große Verbreitung erfahren hat, liegt in der schlechten oder nicht vorhandenen Implementierung des Protokolls in diversen E-Mail-Clients.

2.2.5. Simple Mail Access Protocol - SMAP

Eine Erweiterung des IMAP-Protokolls stellt das SMAP-Protokoll [34] dar. Es wurde von Double Precision, Inc im Rahmen ihres populären Mail-Servers entwickelt und soll einige Schwachstellen von IMAP beseitigen. So ist zum Beispiel das direkte Herunterladen von Binär-Anhängen möglich, wodurch ca. 25 % an Bandbreite eingespart werden kann. Eine Kodierung der Anhänge mit base64 ist somit nicht mehr notwendig. Weiters ist das gleichzeitige Versenden und Ablegen von Nachrichten im „Gesendet“ Ordner in einem Schritt möglich.
Die bisherige Verbreitung des Protokolls scheiterte an den spärlich verfügbaren Mail-Servern bzw. Client Programmen.

3. Spam Grundlagen

Unter Spam oder Junk versteht man im Allgemeinen unerwünschte elektronische Nachrichten, die meist in großen Mengen versendet werden. Die Definition von Spam gestaltet sich dabei sehr schwierig, eine offiziell gültige Begriffsbestimmung gibt es bis dato nicht.
Bevor Spammer überhaupt Nachrichten verschicken können, müssen sie zuerst an möglichst viele Adressaten gelangen. Dabei kommen hauptsächlich automatisierte Verfahren zur Anwendung, die mit Hilfe des Internets völlig selbstständig die Sammlung von Adressen durchführen.
Zum Versenden werden schon lange nicht mehr nur eigene Rechner verwendet. Falsch konfigurierte Mail-Server oder durch Schadsoftware übernommene Computer erlauben es Spammern Millionen von Nachrichten innerhalb kürzester Zeit zu versenden. Die dadurch eingesparten Kosten machen das Phänomen Spam noch profitabler.
Auf der anderen Seite verursacht Spam jährlich einen Schaden in Milliardenhöhe. Nur wenige Unternehmen können von Spam profitieren.

3.1. Was ist Spam?

Der Begriff „Spam“ steht ursprünglich für „Spiced Pork And Meat“ und bezeichnet ein von der US-Firma Hormel Foods hergestelltes Dosenfleisch. Die heutige Bedeutung verdankt der Begriff wohl der britischen Komikergruppe Monty Python, die in einem Sketch das Dosenfleisch auf die Schippe nimmt.
Ein Gast fragt in einem Restaurant, wo es ausschließlich Speisen mit Spam gibt, nach einem Gericht ohne Spam. Die Kellnerin offeriert dem Gast aber weiterhin nur Speisen mit Spam. Der Dialog zwischen Gast und Kellnerin wird permanent durch Spam-Loblieder eines Wikinger-Chors unterbrochen, der schließlich jede weitere Unterhaltung unmöglich macht [21].

Tabelle 3.1.: Was wird als Spam empfunden? [42]

Unverlangte E-Mail-Werbung von unbekannten Absendern	92 %
Unverlangte E-Mail-Werbung von Absendern mit Geschäftsbeziehung	32 %
Unverlangte E-Mail von einer Non-Profit Organisation	65 %
Unverlangte E-Mail mit Erwachsenen-Inhalt	92 %
Unverlangte E-Mail mit politischem Inhalt	76 %
Unverlangte E-Mail mit religiösem Inhalt	76 %
Unverlangte E-Mail mit Aktientipps oder Kreditangeboten	89 %

Genauso wie die Loblieder auf Spam jegliche Unterhaltung unmöglich machten, so erschwert die Spamflut die E-Mail-Kommunikation.
Die Definition des Begriffs „Spam“ ist nicht so einfach, wie es auf dem ersten Blick erscheinen mag. Einer der Gründe liegt darin, dass die Wahrnehmung von Spam grundsätzlich subjektiv ist. Als Beobachter lässt sich nicht immer feststellen, ob die Nachricht nun erwünscht ist oder nicht. Ist für den einen eine E-Mail mit den neuesten Kochrezepten kein Spam, kann es für den anderen durchaus als Belästigung empfunden werden. Die in der Tabelle 3.1 zusammengefasste Umfrage macht die Unterschiede in der Wahrnehmung deutlich. So werden z. B. Produktangebote von unbekannten Absendern mit 92 % als Spam eingestuft. Wird dieselbe Nachricht jedoch von Unternehmen versandt, mit denen bereits eine Geschäftsbeziehung besteht, so wird diese nicht als Spam empfunden. Es scheint also eine Rolle zu spielen, wer der Absender der Nachricht ist.

3.1.1. Definition

Grundsätzlich werden unter Spam jene E-Mails verstanden, die unerwünscht im Postfach eintreffen. Bislang wurden aber wenige Versuche unternommen, um eine einheitliche Begriffsbestimmung zu finden. Eine international gültige Definition des Begriffs wäre sehr nützlich, um z. B. die Effizienz von Spam-Filter untereinander vergleichen zu können und koordiniert Aktionen gegen Spammer zu starten. Im Internet kursieren diverse Definitionen von Spam. Spamhaus z. B. definiert Spam folgendermaßen [79]:

*«Spam: **U**nsolicited **B**ulk **E**mail (UBE)»*

Die Definition des Online Lexikons Merriam-Webster beschreibt Spam etwas restriktiver [67]:

*«Spam: **U**nsolicited **C**ommercial **E**-Mail sent to a large number of addresses (UCE)»*

Die Beschränkung auf den rein kommerziellen Inhalt würde aber politische oder religiöse Nachrichten vom Begriff „Spam“ ausschließen. E-Mails von diesem Typ muss man aber sehr wohl auch als Spam bezeichnen. Treffender die Definition von Paul Graham [47]:

«Spam: Unsolicited Automated email»

Einige rechtliche Definitionen in verschiedenen Ländern nehmen elektronische Nachrichten ausdrücklich aus, die von Unternehmen stammen, mit denen bereits eine Geschäftsbeziehung besteht [47]. Aber was berechtigt z. B. einen Online-Shop, unverlangt E-Mails zu versenden, nur weil dort irgendwann ein Einkauf getätigt worden ist? Wir wollen uns der Definition von Paul Graham anschließen und auch solche Mitteilungen als Spam einstufen.
Während der Begriff „Spamming“ das Phänomen an sich beschreibt, bezeichnet man Personen die Spam in Umlauf bringen als Spammer.

3.1.2. Geschichtliches

Bereits 1975 - lange bevor der Begriff „Spam“ in aller Munde war - reflektierte Jon Postel über die Probleme, die unerwünschte Nachrichten hervorrufen könnten. Im damaligen ARPANET [73] waren keine Mechanismen vorgesehen, die einen Missbrauch des Mail-Systems verhindern konnten. Postel führt z. B. die Möglichkeit einer „Denial of service“ Attacke an, bei der ein fehlerhaft konfigurierter Host unzählige Nachrichten verschickt und den Zielhost nicht mehr erreichbar macht.
Um so erstaunlicher scheint es, dass im später definierten SMTP-Protokoll keine Verfahren vorgesehen wurden, die die missbräuchliche Verwendung des E-Mail-Systems verhindern würden.
Der erste bekannte kommerzielle Internet-Spam wurde 1978 von Gary Thuerk - einem Vetreter von DEC Computern - versendet [24]. Er verschickte Einladungen

zur Vorstellung eines neuen DEC Rechner-Modells in Kalifornien. Weil es eine eingebaute Netzwerk-Software besaß, beschloss Thuerk alle Benutzer des Netzes anzusprechen, die an der Westküste der USA wohnten.
Bereits sieben Jahre vorher versandte Peter Bos - damals Systemadminstrator beim „Massachusetts Institute of Technology“ (MIT) - eine Antikriegsbotschaft mit Hilfe des am MIT entwickelten „Compatible Time-Sharing System“ (CTSS) [24]. Aufgrund der Netzwerktopologie beschränkte sich die Ausbreitung aber auf das Universitätsnetzwerk.

3.1.3. Arten von Spam

Spam-E-Mails lassen sich grundsätzlich in zwei Gruppen einteilen, zum einen in solche mit Anhang, und zum anderen in solche ohne Anhang .
Betrachtet man diese zwei Gruppen etwas näher, so ergibt sich folgendes Bild: Spams ohne Anhang sind meistens Text-Nachrichten mit URLs oder anklickbaren Links zu Webseiten. Mails mit reinem Textinhalt sind meistens Betrug-Mails und richten sich an eine beschränkte Anzahl von Empfängern. Die Größe beträgt in der Regel nicht mehr als 2 bis 3 Kilobyte.
Spams mit Anhang machen mehr als 50 % des Spamaufkommens aus und können in vier weitere Kategorieren unterteilt werden:

- Mails mit Bildanhang und Text
- Mails mit Bildanhang und Link/URL
- Mails mit Bildanhang, Text und Link/URL
- Mails mit Viren, Würmer oder Trojaner

Spams mit Anhang weisen normalerweise eine Größe von 35 bis 140 Kilobyte auf.

3.1.4. Spam Kategorien

Die Definition des Begriffs „Spam“ ist weitreichend und nimmt keinen Bezug auf deren Inhalt. Nachfolgend sei ein kurzer Blick darauf geworfen. Der von der Firma Symantec Inc. monatlich veröffentlichte Bericht über das derzeitige Spam-Aufkommen gibt dabei einen guten Überblick. [90].
In Abbildung 3.1 erfolgt die Aufschlüsselung der Inhalte von Spam-Mails. Der größte Teil davon enthält demnach Produktwerbung aus verschiedenen Lebensbereichen. Angefangen von Internet-Produkten, wie Webhosting, Webdesign, bis hin zu Gesundheitsprodukten, ist fast die gesamte Produktpalette vertreten. Mit 6 % ist der Anteil von Spams mit Erwachseneninhalten relativ gering.

Beschreibung der Spam-Kategorien:

Produkte: E-Mails, die allgemeine Produkte und Dienstleistungen bewerben z. B. Elektrogeräte, Kleidung.

Erwachseneninhalt: E-Mails, die Produkte anpreisen, die für Personen ab 18 Jahren bestimmt sind, z. B. Partnervermittlungen, pornografische Produkte.

Betrug: E-Mails, die gutgläubigen Personen versuchen, ihr Geld aus der Tasche zu ziehen z. B. mit Pyramidenspielen oder Kettenbriefen.

Gesundheit: E-Mails, die Medikamente oder Gesundheitsprodukte anbieten oder bewerben wie z. B. Viagra.

Phishing: Mit diesen E-Mails wird versucht, an fremde Zugangsdaten, Passwörter oder Kreditkartennummern zu kommen (Identitätsdiebstahl).

Freizeit: E-Mails, die Freizeitprodukte bewerben oder verkaufen z. B. verbilligte Reisen, Online-Casinos, Spiele.

Internet: E-Mails, die computerbezogene Produkte und Dienstleistungen bewerben z. B. Webhosting, Webdesign, Software.

Finanz: E-Mails mit finanzbezogenem Inhalt wie z. B. Aktien-Tipps und günstige Kredite.

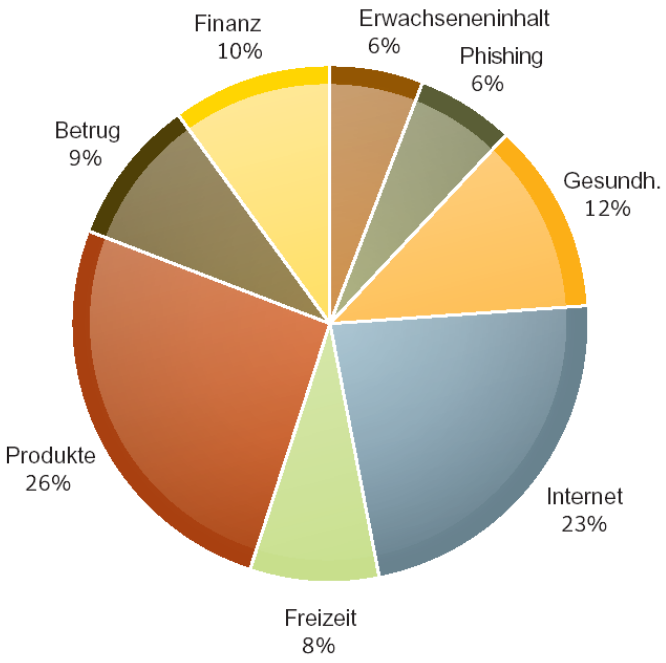

Abbildung 3.1.: Spam-Kategorien

3.2. Adressengewinnung

Um als Spammer erfolgreich agieren zu können, wird vor allem eines benötigt: eine ganze Menge E-Mail-Adressen. In Anlehnung an verschiedene Studien [26], [38] benutzen Spammer nach wie vor das Internet, um an E-Mail-Adressen zu gelangen. Sobald man seine E-Mail-Adresse im Internet veröffentlicht, kann man mit ziemlicher Sicherheit davon ausgehen, dass die Adresse in den Händen von Spammern landet. Diese tauschen Adressen häufig untereinander aus, womit sich das Problem gleich multipliziert.
Die Quellen der Adressen sind vielfältig. Nachfolgend seien die wichtigsten Quellen erwähnt [83]:

Webseiten: Ähnlich den Suchmaschinen-Agenten verwenden Spammer häufig automatisierte Programme, die jeden Link auf einer Website verfolgen und aus jeder gefundenen Seite die E-Mail-Adressen extrahieren. Dabei reicht manchmal ein Eintrag in einem Gästebuch schon aus, um eine Spam-Flut auszulösen. In [26] wurde ein direkter Zusammenhang zwischen Besucheranzahl einer Website mit veröffentlichter E-Mail-Adresse und dem danach

folgenden Spamaufkommen festgestellt. Interessant ist auch, dass nach der Entfernung der E-Mail-Adresse das Spam-Aufkommen wieder abnahm.

Usenet: Aber nicht nur Websites werden durchforstet, das Usenet bietet ebenfalls einen guten Nährboden für Spammer. Beiträge in Newsgroups werden analog wie Webseiten mittels automatisierten Programmen durchleuchtet. Auch das künstliche Verunstalten von E-Mail-Adressen z. B. das Ersetzen des „@“ Zeichens durch „[at“] scheint nichts zu nützen, da einschlägige Spam-Tools bereits längst damit fertig werden.

Mailinglisten: Mailinglisten stellen ebenfalls gute Adressquellen für Spammer dar, da diese in der Regel einen sehr hohen Anteil an gültigen Einträgen enthalten.

Kettenbriefe: Bei den oben beschriebenen Verfahren lassen sich E-Mail-Adressen vollautomatisch extrahieren. Es gibt bereits unzählige Tools im Internet, z. B. „Atomic Harvester 2000“, das bereits für unter 150 $ zu haben ist, mit dem selbst Laien ohne besondere technische Kenntnisse Adressen sammeln können. Ein anderer, manueller Weg, um an viele E-Mail-Adressen zu gelangen, besteht z. B. darin, Kettenbriefe zu versenden. Durch versprochene Gewinne werden Benutzer animiert ihre E-Mail-Adresse preiszugeben.

Webbrowser: Einige Websites benutzen verschiedene Tricks, um die im Webbrowser eingegebene E-Mail-Adresse auszuforschen. Eine Möglichkeit besteht zum Beispiel darin, 1x1 Pixel Graphiken in die Seite einzubinden. Diese Graphik wird aber nicht über HTTP geladen, sondern über FTP:

```
<IMG SRC="ftp://ftp.server.at/spam/grafik.gif"
WIDTH="1" HEIGHT="1">
```

Bei FTP-Zugriffen über „Anonymous FTP“ wird die E-Mail-Adresse des Benutzers als Kennwort übertragen, in diesem Fall, die im Browser eingestellte E-Mail-Adresse. Von dieser Übertragung bekommt der User in der Regel überhaupt nichts mit [45].

Aber auch Personen, die ihre E-Mail-Adresse nicht im Internet publizieren, können Spam erhalten. Der Grund dafür liegt darin, dass Spammer häufig mittels Wörterbuchattacken alle möglichen Namenskombinationen ausprobieren und auf

gut Glück Spam versenden. Durch Auswertung der Server-Fehlermeldungen kann darauf geschlossen werden, ob eine Adresse aktiv ist oder nicht.
Da diese Methode jedoch sehr viel Datenverkehr erzeugt und vom Provider leicht bemerkt wird, gehen Spammer dazu über, bereits während des SMTP-Dialogs Adressvalidierungen vorzunehmen. Korrekt konfigurierte Mail-Server antworten nämlich bereits nach einem `RCPT TO:` Befehl mit einer entsprechenden Fehlermeldung, sollte die Empfänger-Adresse nicht existieren. Das machen sich Spammer zu Nutze und senden während der SMTP-Sitzung unzählige `RCPT TO:` Befehle. Nicht abgelehnte Adressen werden dann als gut befunden und in die Adress-Datenbank aufgenommen.

```
1  telnet mail.mydomain.net 25
2  server: 220 mail.mydomain.net Mail Server
3  client: HELO localhost
4  server: 250 mail.mydomain.net
5  client: MAIL FROM: from@spammer.net
6  server: 250 Ok
7  client: RCPT TO: max@mydomain.com
8  server: 550 Sorry, max@mydomain.com is not a valid recipient
9  client: RCPT TO: moritz@mydomain.com
10 server: 550 Sorry, moritz@mydomain.com is not a valid recipient
11 client: RCPT TO: fritz@mydomain.com
12 server: 250 Ok             --> gültige Adresse gefunden
13 client: DATA
14 server: 354 End data with <CR><LF>.<CR><LF>
15 client: From: from@spammer.net
16     To: irgendwas@mydomain.com
17     Subject: Spam
18     Hallo,
19     Spam Inhalt
20     .
21 server 250 Ok: queued as A0BFC1A6586
22 client QUIT
23 server 221 Bye
```

Glaubt man Insider-Quellen [64], so bekommt man eine Million E-Mail-Adressen bereits zu einem Preis von 15 $ bis 25 $. Dabei kann man auswählen, ob man generelle Adressen, AOL-Adressen oder Erwachsenen-Adressen haben möchte.

3.3. Versand

Die Techniken des Spamversands sind in den letzten Jahren immer ausgereifter geworden. Waren es zu Beginn meist Hobbyisten, die teilweise stümperhaft agierten, so gibt es mittlerweile professionelle, leicht zu bedienende Software[1], die nicht mehr als 100 $ kostet. Eigenschaften, wie die zufällige Generierung der Absender-Adresse, wechselnder Inhalt und Statusverfolgung, gehören zur Standardausstattung solcher Programme. Die Nachrichten werden meist nicht über den eigenen Rechner versendet, sondern über offene *Mailrelays* und/oder *Open Proxies* [93]. Bei *Mailrelays* handelt es sich in erster Linie um falsch oder schlecht konfigurierte Mail-Server, die das Versenden von E-Mails ohne Authentifizierung erlauben. *Open Proxies*, im weiteren Sinne ebenfalls *Mailrelays*, sind in erster Linie durch Hacker übernommene PCs, über die ohne Wissen des Besitzers Spam versendet wird. Dabei handelt es sich nicht um einzelne Rechner, sondern um ganze Netzwerke mit nicht selten über 10.000 durch Schadsoftware übernommene PCs. Ende November 2007 etwa wurde in Neuseeland ein 18-jähriger Hacker festgenommen, der nicht weniger als 1,3 Millionen PCs unter seiner Gewalt gehabt haben soll [51].
Daraus lässt sich ableiten, dass heutzutage Spammer und Hacker eng zusammenarbeiten. Es ist sogar ein regelrechter Handel mit E-Mail-Adressen, übernommenen Rechnern und Spam-Dienstleistungen entstanden. So kann man beispielsweise einen Zombierechner für 3 $ im Monat mieten oder den kompletten Spam-Versand outsourcen [46]. So muss man für das Versenden von 20 Millionen Spams lediglich 350 $ hinlegen [94]. Die Graphik 3.2 gibt einen Überblick, über welche Länder zwischen Januar und März 2008 am meisten Spam versandt wurde [88]. Die Vereinigten Staaten von Amerika sind dabei nach wie vor einsamer Spitzenreiter.

3.4. Realisierung des Gewinns durch Spam

Genauso wie im gewöhnlichen Geschäftsleben handeln Spammer in der Regel nicht aus Überzeugung, sondern um Geld zu verdienen. Die Schritte, die dafür

[1] siehe z. B. http://www.send-safe.com, http://www.bestextractor.com

Position	Land	Prozentualer Anteil des verursachten Spams
1	Vereinigte Staaten	15,4%
2	Russland	7,4%
3	Türkei	5,9%
4	China (inkl. Hong Kong)	5,5%
5	Brasilien	4,3%
6	Südkorea	4,0%
7	Polen	3,8%
8	Italien	3,6%
9=	Deutschland	3,4%
9=	Vereinigtes Königreich	3,4%
10	Spanien	3,3%
11	Frankreich	3,2%
Sonstige		36,8%

Abbildung 3.2.: Spam - Das dreckige Dutzend - Quelle sophos.de

notwendig sind, unterscheiden sich nicht von einem herkömmlichen Business Plan:

1. Finden von potenziellen Kunden
2. Produkt oder Dienstleistung anbieten
3. Geschäft abschließen

Der Erfolg von Spammern fußt darauf, dass die Aktivitäten 1 und 2 äußerst günstig durchzuführen sind, d. h. auch eine sehr geringe Erfolgsrate erweist sich bereits als profitabel. Jeremy Jaynes - in den USA bereits verurteilt - hat in seiner kurzen Laufbahn als Spammer sage und schreibe 24 Millionen Dollar verdient [1]. Ein erfahrener Spammer kann heutzutage durchaus 800.000 $ und mehr pro Monat verdienen [76].
Die Tabelle 3.2 vergleicht die Kosten herkömmlicher Werbeaussendungen mit denen von Spam. Sind herkömmliche Aussendungen per Post erst ab einer Antwort-Rate von 2 % profitabel [43], so erreicht eine Spamaussendung den Break-Even bereits bei einer Rate von 0,001 %, d. h. eine Briefaussendug benötigt mindestens ein 2000-mal höheres Feedback als eine Spam-Kampagne. Ein Spammer kann so z. B. 500.000 Nachrichten verschicken und ist bereits ab fünf Kaufabschlüsse profitabel.
Viel höher sind die Rücklaufquoten bei Phishing-E-Mails, die bis zu 5 % betragen [43]. Der Grund der hohen Antwortraten liegt primär im professionellen

Tabelle 3.2.: Die Kosten pro Empfänger diverser Marketinginstrumente [76]

	Gesamtkosten	**Anzahl Empf.**	**Kosten pro Empf.**
Herkömmlicher Brief	9.700 $	7.000	1,39 $
Telemarketing	160 $	240	0,66 $
Print Werbung	30.000 $	442.000	0,067 $
Fax Werbung	30 $	600	0,05 $
Online Werbung	35 $	1.000	0,035 $
Spam	250 $	500.000	0,0005 $

Auftreten der Betrüger. Angefangen vom E-Mail bis hin zur nachgebauten Seite erscheint alles täuschend echt. Im Gegensatz zu gewöhnlichen Spams, die eine Menge unleserlicher oder nicht zusammenhängender Wörter enthalten, um die Spam-Filter zu umgehen, erscheinen Phishing-Mails äußerst professionell. Leider gibt es zu viele Personen, die ohne eine Sekunden nachzudenken, sensible Daten in Online-Formulare eingeben.

Tabelle 3.3.: Die häufigsten Ziele von Phishing-Attacken [27]

Unternehmen	**Anteil in %**
CitiBank	54.16
Smith Barney	13.48
SunTrust	10.02
Paypal	7.57
Wells Fargo	5.42
HSBC	5.07
eBay	4.15
USBank	0.11
CitizensBank	0.014

Gesondert betrachtet werden müssen Aktien-Spams. Diese Nachrichten bewerben kein Produkt im eigentlich Sinn, sondern versuchen durch vermeintliche gute Tipps den Aktienkurs - meist von Penny-Stocks - in die Höhe zu treiben. Der Versender gibt sich meist als Insider aus und verspricht durch angeblich in Kürze veröffentlichte „Gute Nachrichten“ einen schnellen Gewinn. Nachfolgend wollen wir kurz untersuchen, welche Auswirkungen solche Nachrichten auf den Aktienkurs haben. Man muss leider feststellen, dass das aktuelle Business-Modell von Aktien-Spammern funktioniert [81]. In den Tagen, an denen solche Aktien beworben werden, war das Handelsvolumen überdurchschnittlich hoch. Dabei scheint es einen direkten Zusammenhang zwischen der Anzahl von Spamnachrichten

und gehandeltem Volumen zu geben, d. h. je mehr Spam-Mails im Umlauf waren, desto mehr Aktien wurden gehandelt.

Tabelle 3.4.: Intraday Kursentwicklung [81]

	Aktienspam an diesem Tag im Umlauf	
Intra-day Veränderung	Nein	Ja
Eröffnungskurs > Schlusskurz	27,8 %	51,9 %
Eröffnungskurs = Schlusskurz	47,1 %	24,3 %
Eröffnungskurs < Schlusskurz	25,1 %	23,8 %

Da solche Penny-Stocks an normalen Tagen kaum oder gar kein Handelsvolumen erzeugen, können bereits kleinere Transaktionen eine Kursveränderung zur Folge haben. Die Veränderungen im Handelsvolumen und Aktienkurs sind vor allem auf folgende drei Gruppen zurückzuführen:

1. Spammer, die die selber beworbenen Aktien kaufen
2. Naive Empfänger, die Aktien kaufen, im Glauben eine gute Investition zu tätigen
3. Empfänger, die das Spiel von Spammern durchschauen und selber versuchen, vom Kursverlauf zu profitieren.

Beobachtet man die Aktienkurse jedoch langfristig, so ist man gut beraten, nicht in solche Papiere zu investieren. Nur wenige Papiere sind nach einigen Monaten mehr Wert als zum Kaufzeitpunkt [31].

3.5. Profiteure von Spam

Am meisten profitieren natürlich die Spammer selbst und die Unternehmen, deren Produkte beworben werden. Mittlerweile sind jedoch rund um die Spam-Problematik weitere Geschäftsfelder entstanden. In erster Linie sind dabei Hersteller von Anti-Spam-Lösungen zu nennen, deren Geschäftsgrundlage direkt auf Spam aufbaut. Unmittelbar profitieren auch Hersteller von Sicherheitslösungen, wie Anti-Virus- oder Firewall-Hersteller, da die Bedrohung, die von Phishingattacken oder Spyware ausgeht, ständig steigt.

Indirekte Profiteure von Spam sind hingegen Internet-Provider, die den Spammern den Internet-Zugang zur Verfügung stellen und vom entstandenen Datenverkehr erheblich profitieren. Nicht selten löhnen Spammer pro Monat einige Zehntausend Dollar für ihren Internet-Anschluss. Es ist daher durchaus verständlich, dass einige Internet-Provider ein Auge zudrücken und nicht alles Notwendige dagegen unternehmen [89]. Auch Betreiber von Rechenzentren erfreuen sich wachsender Gewinne, da auch hier ein Großteil des Datenverkehrs auf Spam zurückzuführen ist.

3.6. Auswirkungen von Spam

Haben wir im vorherigen Kapitel die Methoden und Adressquellen von Spammern kennengelernt, so wollen wir nun einen Blick auf die Auswirkungen von Spam werfen.

3.6.1. Volkswirtschaftlicher Schaden

Die Kosten, die unerwünschte Nachrichten verursachen, sind vielfältig und lassen sich nur durch Schätzungen in Zahlen ausdrücken. Diese Zahlen geben aber Anlass zur Sorge. Einer Studie von Ferris Research aus dem Jahr 2005 zufolge, verursacht Spam weltweit geschätzte 50 Milliarden US-Dollar an Kosten. Für Großbritannien allein schätzt Ferris Research die Gesamtkosten auf 2,48 Milliarden US-Dollar, für Japan auf rund 5,2 Milliarden US-Dollar [44].
Direkte Kosten für Unternehmen entstehen z. B. durch den Produktivitätsverlust am Arbeitsplatz, da Spamnachrichten vielfach händisch aussortiert werden müssen. Spam-Nachrichten enthalten nicht nur Werbung, sondern auch vielfach Schadsoftware, die sich für Unternehmen als kostspielig erweisen. Besonders rufschädigend kann dabei sein, wenn von übernommenen Rechnern aus über das eigene Firmennetzwerk Spam-Nachrichen versendet werden. Schlimmstenfalls landet die öffentliche IP-Adresse des eigenen Mail-Servers auf schwarzen Listen, sodass viele Empfänger nicht mehr erreicht werden können.
Weiters wird in vielen Fällen mehr Bandbreite und Speicherkapazität benötigt, um die vielen Nachrichten überhaupt empfangen und speichern zu können. Die

IT-Abteilungen müssen viel Geld und Zeit in Lösungen investieren, die versuchen, die Spamflut einzudämmen.
Indirekte Kosten werden durch Falsch-Klassifizierungen (Falsche-Positive) von eintreffenden Nachrichten produziert. Diese werden entweder direkt geblockt oder landen im Spam-Ordner, sodass E-Mails in vielen Fällen übersehen und von niemandem gelesen werden.
Einer der bedenklichsten Folgen von Spam liegt jedoch im zunehmend sinkenden Verbrauchervertrauen, das die Grundlage für den Erfolg des elektronischen Handels darstellt. So kann man sich nicht mehr sicher sein, ob ein E-Mail den Empfänger nun erreicht hat, von einem Spam-Filter fälschlicherweise geblockt oder vom Empfänger - aufgrund der vielen Spams in seinem Postkasten - einfach übersehen wurde.

3.6.2. Auswirkungen auf den E-Mail-Nutzer

Wie im vorhergehenden Abschnitt beschrieben, verursacht Spam erhebliche Kosten für Unternehmen. Hier sei nochmals näher darauf eingegangen, was Spam für den einzelnen E-Mail-Nutzer bedeutet. Neben den bereits erwähnten Kosten für Firmen, die großteils auch auf jeden E-Mail-Nutzer zutreffen, ergeben sich immer mehr Einschränkungen für den Endverbraucher.
Konnten anfangs über jeden Postausgangsserver Nachrichten verschickt werden, so ist dies in der Regel nicht mehr möglich. Einige Provider akzeptieren ausgehende Verbindungen nur mehr sehr eingeschränkt. Auch eine erfolgreiche SMTP-Authentifizierung bedeutet noch lange nicht, dass sämtliche Nachrichten versandt werden können. In vielen Fällen sind nur Absenderadressen erlaubt, die von eigenen Domains stammen bzw. muss sich die Client IP-Adresse im Adressbereich des Providers befinden. Einige Internet-Provider gehen sogar noch einen Schritt weiter und sperren den SMTP Standard-Port komplett. Die Benützung eines E-Mail-Clients ist somit für den Normalverbraucher nicht mehr möglich. Als einziger Ausweg bleibt das meist umständlichere Web-Interface.
Besonders ärgerlich sind Phishingmails, die auf einen Identitätsdiebstahl abzielen. Täuschend echte E-Mails im Namen renommierter Banken oder Handelsplattformen sollen den unerfahrenen Nutzer ermutigen, persönliche Daten wie Kreditkartennummern, Bankcodes und dergleichen preiszugeben. Laut einer Studie der Federal Trade Commission (FTC) haben Phishingattacken in den USA

allein im Jahr 2002 einen Gesamtschaden von 37 Milliarden Dollar verursacht. Nicht weniger als 3,3 Millionen Amerikaner waren davon betroffen [87].
Um dem Spam-Problem schon von vornherein aus dem Weg zu gehen, vermeiden viele die Weitergabe der persönlichen E-Mail-Adresse bzw. die Publizierung im Internet [42].

3.6.3. Folgen für den E-Mail-Provider

E-Mail-Provider stehen in den letzten Jahren immer mehr in der Pflicht, geeignete Maßnahmen zu treffen, die den Endverbraucher vor unerwünschten Nachrichten bewahren. Kaum ein Provider kann es sich heute leisten, Postfächer nicht vor Spam und Viren zu schützen. Dabei greifen Provider auf Eigenentwicklungen oder auf Lösungen von Drittanbietern zurück. Egal, für welche Lösung sich Provider letztlich entscheiden, sie kosten auf jeden Fall Geld. Diese Kosten können aus verschiedenen Gründen nicht immer an den Endverbraucher weitergegeben werden.
Aber der Schutz gegen eingehende Spams reicht längst nicht mehr aus. Auch ausgehende Nachrichten, also Nachrichten, die vom Kunden über die Server des Providers verschickt werden, müssen auf Viren und Spams überprüft werden. Geschieht dies nicht ausreichend genug, so landen die IP-Adressen der Mail-Server sehr schnell auf „Schwarzen Listen“, mit der Folge, dass viele ausgehende Nachrichten nicht mehr zugestellt werden können. Der Ruf des Providers wird dadurch stark geschädigt.
Wie andere Unternehmen auch, trifft der erhöhte Bandbreiten- und Speicherbedarf Provider besonders hart. Weiters sind die technischen Maßnahmen gegen Spam meist sehr ressourcenhungrig, d. h. ein Großteil der Rechenleistung wird für die Spamabwehr benötigt. Weiters müssen sich Provider gegen gezielte „Denial of Service“ Attacken rüsten, um eine hohe Verfügbarkeit seiner Dienstleistungen gewährleisten zu können.
Neben den rein technischen Maßnahmen gegen Spam können sich unter Umständen auch rechtliche Probleme für den Provider ergeben. Durch Falsch-Klassifizierungen kann es durchaus vorkommen, dass wichtige Nachrichten den Empfänger nicht erreichen. Für solche Szenarien sollte sich der Provider wappnen, um nicht mit Schadenersatzforderungen konfrontiert zu werden.

Auch wenn ein qualitativ hochwertiger Spam-Filter einen erheblichen finanzieller Aufwand darstellen mag, so erhält der Provider dadurch auch ein Marketinginstrument, um sich von den Mitbewerbern abzuheben. Einige Provider haben sich gänzlich darauf spezialisiert und betreiben Filterlösungen für Webhosting Unternehmen.

4. Vorbeugende Maßnahmen

4.1. Schutz der E-Mail-Adresse

Wie bereits im Abschnitt „Adressengewinnung“ beschrieben, verwenden Spammer das Internet, um an potenzielle Empfänger zu gelangen. Designer und Programmierer von Webseiten sollten daher technische Möglichkeiten einsetzen, die eine solche Adressausspähung unmöglich machen oder zumindest erschweren. Grundsätzlich sollte man seine E-Mail-Adresse nur veröffentlichen, wenn es unbedingt notwendig ist. Das Verwenden eines Kontaktformulars auf Websiten ist der Veröffentlichung einer E-Mail-Adresse auf jeden Fall vorzuziehen [92]. Das Webformular muss man wiederum auch gegen Missbrauch - z. B. durch den Einsatz von Capchats - schützen [85].
Am einfachsten macht man es Spammern, wenn man die E-Mail-Adresse als Plain-Text in die Website einfügt. Versieht man diese Adresse zusätzlich noch mit einem „mailto“ Link, so ist man leichtes Fressen für Spam-Bots.
Einfache Codierungsverfahren, wie das Ersetzen des „@“-Zeichens durch das ASCII-Äquivalent (Hexadezimalwert @), führen bereits dazu, dass einige Spider die E-Mail-Adressen nicht mehr aufspüren. Besser als nur das „@“-Zeichen zu verschlüsseln, ist das Codieren der gesamten Adresse samt dem „mailto“ Präfix. Der Link „mailto:max@mustermann.at“ würde dann im Quelltext der Internetseite folgendermaßen aussehen:

```
&#109;&#97;&#105;&#108;&#116;&#111;&#58;&#109;&#97;&#120;&#64;
&#109;&#117;&#115;&#116;&#101;&#114;&#109;&#97;&#110;&#110;
&#46;&#97;&#116;
```

Anstelle der ASCII-Codierung kann man auch die URL-Codierung anwenden.

```
&#109;&#97;&#105;&#108;&#116;&#111;&#58;%6d%61%78%40%6d%75%73%
74%65%72%6d%61%6e%6e%2e%61%74
```

Ausgeklügelte Spider lassen sich durch solche einfache Methoden zwar nicht zur Gänze abschrecken, ein Großteil der sich im Umlauf befindlichen Programme kann aber mit solchen Kodierungen nicht umgehen. Entfernt man noch zusätzlich den „mailto“ Link, so ist die Wahrscheinlichkeit sehr gering, dass die Adresse ausgespäht wird [92].
Effektivere Methoden bedienen sich Javascripts oder gar Flash-Plugins [41]. Javascript beherrschen mittlerweile alle gängigen Browser, jedoch hat nicht jeder ein Flash-Plugin installiert, deshalb sollte man mit der Auswahl dieser Verschleierungsmethode Vorsicht walten lassen.

```
1 <script type="text/javascript">
2 //<![CDATA[
3 var email = "max"
4 var domain = "mustermann.at"
5 document.write("<a href=" + "mail" + "to:" + email + "@"
6 + domain + "?subject=Anfrage" + ">" + email + "@"
7 + domain + "</a>")
8 //]]>
9 </script>
```

Adress-Verschleierung mittels Javascript

Ein anderer Ansatz besteht darin anstatt dem Text ein Bild anzuzeigen, das die E-Mail-Adresse enthält. Spam Bots ignorieren Bilder in den meisten Fällen, da die Decodierung mittels OCR Software sehr schwierig und zu ressourcenintensiv wäre.
Bei bereits erstellten Webseiten kann es unter Umständen sehr zeitaufwendig sein, alle E-Mail-Adressen nachträglich zu kodieren. Ein guter Ansatz stellt das in [92] beschriebene Verfahren dar, das die Kodierung der Adressen bereits auf Webserver-Ebene - mittels Perl Module für Apache - vornimmt.

4.2. Wegwerfadressen

Als Wegwerfadressen werden solche E-Mail-Adressen bezeichnet, die nur eine beschränkte Gültigkeit aufweisen. Diese Beschränkung kann sich auf den Zeitraum oder auf die Häufigkeit der eingehenden Nachrichten beziehen. Erfolgt eine weitere Zustellung außerhalb des Gültigkeitsbereichs, so werden diese Nachrichten automatisch verworfen.

Wegwerfadressen sind für den Umgang mit Institutionen gedacht, bei denen man sich nicht sicher sein kann, ob diese sorgfältig mit E-Mail-Adressen umgehen. Dies kann z. B. ein Online-Shop sein, der trotz Widerruf regelmäßige Newsletter aussendet, oder eine Portal, dessen Seriösität nicht auf den ersten Blick ersichtlich ist.
Ein bekannter Anbieter für Wegwerfadressen ist z. B. SpamGourmet [12]. Für die Benutzung dieses Dienstes ist zunächst die Angabe einer gültigen Weiterleitungsadresse notwendig, an die alle an SpamGourment gesandten E-Mails weitergeleitet werden. Die Wegwerfadressen haben bei SpamGourment folgendes Format:

```
irgendwas.x.benutzername@spamgourmet.com
```

„Irgendwas“ ist dabei ein beliebiges Wort und „x“ die Anzahl der E-Mails, die man über diese Adresse erhalten möchte. Wählt man für „Irgendwas“ einen vernünftigen Wert, beispielsweise den Namen des Webshops, bei dem man sich gerade registriert, so kann man später auch nachverfolgen, welches Unternehmen sich nicht an die Spielregeln gehalten hat.
Mittlerweile bieten auch große Free-Mail-Provider die Möglichkeit der Erstellung von Wegwerf-Adressen an [14].

4.3. Weitere Maßnahmen

Wie bereits erörtert wurde, beruht das Business-Modell von Spammern darauf, dass der Versand massenhafter E-Mails sehr schnell abläuft und nur sehr wenig kostet. Einige schlagen deshalb vor, dass man ähnlich wie bei der herkömmlichen Post, eine virtuelle Briefmarke einführt. Diese Briefmarke muss nicht unbedingt einem realen Geldwert entsprechen, sondern kann auch mit Rechenleistung abgegolten werden. Das Konzept dahinter beruht darauf, dass jedem ausgehenden E-Mail ein eindeutiger Hash-Wert angehängt werden muss (im Mail-Header), dessen Erstellung sehr rechenintensiv ist. Der empfangende Host nimmt dieselbe Berechnung vor und überprüft die Übereinstimmung des Hashwertes.
Bei normalem E-Mail-Aufkommen würde die in Anspruch genommene Rechenzeit nicht ins Gewicht fallen, Versender von Massenmails würden es aber unmöglich schaffen, täglich Millionen von Nachrichten zu versenden [20][60].

Dieses Verfahren lässt sich jedoch sehr schwer in die Praxis umsetzen, da alle Mail-Systeme umgestellt werden müssten. Da die Rechenleistung der Computersysteme ständig steigt, wäre auch eine ständige Anpassung des Hash-Algorithmus notwendig.
Ein virtuelles Porto, in welcher Form auch immer, würde auch nicht nur Spammer treffen, sondern auch legitime Versender von Massenmails [50].

5. Technische Maßnahmen gegen Spam

Wurden im vorhergehenden zwei Kapiteln Maßnahmen erläutert, die Spam bereits an der Wurzel den Garaus machen sollen, so kommen nachfolgende Möglichkeiten erst zur Anwendung, nachdem Spam bereits verschickt wurde, d. h. die Spam-Filterung erfolgt auf der Empfängerseite. Grundsätzlich kann zwischen statischen und bayesschen Verfahren unterschieden werden. Im Gegensatz zu statischen Verfahren können bayessche Filter auf veränderte Rahmenbedingungen reagieren und stellen somit ein höchst effizientes Instrument in der Spam Bekämpfung dar.

5.1. Falsche Positive, falsche Negative

Bis dato wurde noch kein Filter erfunden, der alle Spams richtig erkennt und zudem alle regulären Nachrichten normal zustellt. Jede Art von technischem Filter stellt somit immer einen Kompromiss dar. Soll auf der einen Seite die Erkennungsrate sehr hoch sein, so steigt andererseits immer das Risiko einer Falschklassifizierung.
Falsch-Positive (englisch: false positives) sind Nachrichten, die vom Filter fälschlicherweise als Spam eingestuft wurden. Da Falsch-Positive in der Regel einen höheren Schaden anrichten, als ein zugestelltes Spam-E-Mail, sollte bei Design und Implementierung von technischen Filtern die Priorität auf möglichst wenige Falsch-Positive gelegt werden. Gänzlich vermeiden lassen sich Falsch-Positive jedoch nicht.
Falsche-Negative (englisch: false negatives) sind E-Mails, die vom Filter nicht als Spam erkannt worden sind. Natürlich sind falsche Negative - noch viel mehr als

Falsch-Positive - nicht zu verhindern. Durch die nicht immer gleich lautende Definition des Begriffs Spam lässt sich manchmal objektiv auch nicht feststellen, ob es sich nun um eine Falsch-Positive handelt oder nicht.
Kommerzielle Spam-Filter-Lösungen bewerben Ihre Lösungen mit Falsch-Positiv-Raten von unter einem Prozent und Erkennungsraten jenseits der 95 %. Aus den bereits erwähnten Gründen ist eine objektive Messung solcher Filter nur sehr schwer möglich.

5.2. Server- und clientbasierte Ansätze

Bis eine E-Mail beim Empfänger ankommt, durchlauft diese in der Regel mehrere Stationen, normalerweise vier: Den Rechner des Absenders, den Mailserver des Absenders, den Mailserver des Empfänger und den Arbeitsplatzrechner des Empfängers. Die Mail-Server werden dabei häufig von einem Internet Service Provider gestellt. Größere Unternehmen betreiben aber meist eigene Mail-Server. Die technischen Maßnahmen auf der Empfängerseite können nun entweder auf dem Mail-Server oder am Client installiert werden. Für die Installation am Posteingangsserver spricht zum einen die Möglichkeit der zentralen Verwaltung, zum anderen gelangen Spam-Nachrichten gar nicht erst zum Arbeitsplatz. Einige Filtermethoden, wie Greylisting oder DNS basiernde Blacklists, lassen sich nur sinnvoll am Mailserver einsetzen.
Dagegen spricht, dass serverbasierende Lösungen meist schwieriger an die Wünsche und Bedürfnisse der einzelnen Benutzer angepasst werden können. Meistens haben einzelne Personen keine Möglichkeit individuelle Filter-Einstellungen vorzunehmen. Diese werden in der Regel vom Administrator vorgenommen und gelten dann für alle Benutzer. Vor allem die Erkennungsrate von selbst lernenden Filtern leidet darunter, wenn ein zentrales Spam-Corpus darüber entscheidet, was nun Spam ist und was nicht. In vielen Fällen kann diese Entscheidung nur der Empfänger selbst treffen [77].

5.3. Behandlung der als Spam erkannten E-Mails

Nach erfolgter Spam-Klassifizierung muss festgelegt werden, was mit diesen Nachrichten passieren soll. Mögliche Verfahrensweisen sind:

1. **Die E-Mail wird abgelehnt:** Wenn bereits während des laufenden SMTP-Dialogs eine Spam-Erkennung stattfindet, kann bereits zu diesem Zeitpunkt die E-Mail abgelehnt werden. Der absendende Mail-Server erhält den entsprechenden Fehlercode retourniert und verständigt im Normalfall den Absender über die Nichtzustellbarkeit. Wir wollen den SMTP-Dialog etwas genauer durchleuchten:

```
1  telnet mail.mydomain.net 25
2  server: 220 mail.mydomain.net Mail Server
3  client: HELO localhost
4  server: 250 mail.mydomain.net
5  client: MAIL FROM: from@spammer.net
6  server: 250 Ok
7  client: RCPT TO: rctp@mydomain.com
8  server: 250 Ok
9  client: RCPT TO: rctp2@mydomain.com
10 server: 550 Sorry, bad from address
11 client: DATA
12 server: 354 End data with <CR><LF>.<CR><LF>
13 client: From: from@spammer.net
14         To: irgendwas@mydomain.com
15         Subject: Spam
16         Hallo,
17         Spam Inhalt
18         .
19 server 250 Ok: queued as A0BFC1A6586
20 client QUIT
21 server 221 Bye
```

   Besonders interessant is hierbei das Verhalten bei mehreren Empfängern. Im obigen Beispiel wird die E-Mail für <rctp@mydomain.com> angenommen, für <rctp2@mydomain.com> hingegen endet der „RCPT TO:“ Befehl mit dem Fehlercode 550. Im SMTP-Dialog kann also entschieden werden, für welche Empfänger man die Nachricht annehmen möchte. Diese Pro-

tokolleigenschaft kann man sich zu Nutze machen und die Konfiguration benutzerbasierend gestalten.
Wie nach dem Senden des DATA-Befehls die E-Mail abgewiesen, so betrifft das sämtliche im SMTP-Dialog angegeben Empfänger.

2. **Die E-Mail wird angenommen, gelöscht, und eine Nachricht wird an den Absender geschickt:** Die Spam-Klassifizierung während des SMTP-Dialoges ist aus Performance-Gründen nicht immer möglich. Daher werden bei diesem Verfahren E-Mails bedingungslos angenommen und erst anschließend spam-technisch untersucht. Wird die Nachricht als Spam eingestuft, so kann der Absender nur mehr durch eine Antwort-Mail darüber informiert werden, da zu diesem Zeitpunkt die SMTP-Verbindung zum sendenden Host bereits abgebrochen wurde.
   Das stellt zugleich auch das größte Problem bei diesem Verfahren dar. Spammer benutzen in der Regel keine gültigen oder gefälschte Absenderadressen. Eine Antwort würde mit ziemlicher Sicherheit nicht ankommen oder den Falschen erreichen.
   Deshalb sollte man auf keinen Fall den Mail-Server so konfigurieren, dass dieser als Reaktion auf alle E-Mails, die als Spam oder Virus eingestuft wurden, eine E-Mail an den angeblichen Absender dieser E-Mail sendet [77].

3. **Die E-Mail wird verworfen:** Hier wird dem Absender, wie beim Verfahren zuvor, eine erfolgreiche Zustellung suggeriert. Die Nachricht wird aber nicht dem Empfänger übergeben, sondern gelöscht. Dieses Verfahren stellt also streng genommen eine Protokollverletzung dar und sollte nur dann zur Anwendung kommen, wenn man mit hundertprozentiger Sicherheit davon ausgehen kann, dass es sich um Spam handelt [77].

4. **Die E-Mail wird markiert weitergeleitet:** Bei diesem Verfahren wird die E-Mail ebenfalls bedingungslos angenommen und anschließend markiert weitergeleitet. Dies ist dann sinnvoll, wenn man dem User überlassen möchte, wie er mit den erkannten Mails verfährt, oder wenn es nicht möglich ist, dass der Benutzer eigene Filter-Einstellungen vornehmen kann.
   Die Markierung erfolgt in der Regel durch die Erweiterung des Betreffs oder durch das Hinzufügen weiterer Header-Zeilen. Mittels einer Regel im E-Mail-Client kann dann über das weitere Vorgehen entschieden werden.

5. **Die E-Mail wird in einen dedizierten Ordner verschoben:** Häufig werden erkannte Spam Nachrichten auch in separaten Ordnern auf dem Mail-Server abgelegt. Man spricht hier auch von einem Quarantäne-Ordner. Diese Vorgehensweise ist nur dann sinnvoll, wenn der User über die Möglichkeit verfügt, auf diesen Ordner zuzugreifen. Das in E-Mail-Programmen am häufigsten verwendete Protokoll POP kann mit serverbasierten Ordnern aber leider nicht umgehen. Da das IMAP-Protokoll nicht sehr weit verbreitet ist, bleibt in vielen Fällen nur mehr der Weg über ein Web-Interface. Dies ist meist sehr umständlich und somit besteht die Gefahr, dass solche Ordner nie oder sehr selten auf Falsch-Positive hin untersucht werden.
6. **Die E-Mail wird an eine andere Adresse umgeleitet:** Dies kann eine sekundäre Adresse des Empfängers, die Adresse eines Administrators oder eine speziell hierfür eingerichtete E-Mail-Adresse sein. Letzteres vermeidet beispielsweise, dass jeder Benutzer den Quarantäne Ordner eigenständig überprüfen muss.

Es kann nicht generell gesagt werden, was nun die beste Vorgehensweise ist. Diese Entscheidung liegt in der Regel beim Netzwerk-Administrator und muss mit Bedacht gefällt werden. Häufig werden oben beschriebene Verfahren auch in Kombination eingesetzt. So könnten beispielsweise Nachrichten mit sehr hoher Spam-Wahrscheinlichkeit bereits während des SMTP-Dialoges abgewiesen und die restlichen E-Mails als Spam markiert zugestellt werden. [77].

5.4. Statische Verfahren

Unter statische Verfahren fallen Methoden, deren Filterregeln sich nicht automatisch an veränderte Gegebenheiten anpassen, d. h. die Regeln bleiben immer dieselben. Filter mit statischen Regeln lassen sich relativ leicht umgehen, sobald der Spammer über die Funktionsweise dieser Filter Bescheid weiß.
Aufgrund ihrer relativ einfachen Implementierung sind statische Verfahren weit verbreitet und bieten in Kombination ansprechende Erkennungsraten [77].

5.4.1. Schwarze Listen (Blacklists)

Schwarze Listen sind Listen, die Absender, Wörter oder ganze Textpassagen enthalten, die von Spammern häufig gebraucht werden. Die einfachste Form einer Blacklist ist die Absender-Blacklist. Diese Form einer schwarzen Liste blockiert lediglich E-Mails von den in der Liste enthaltenen Absendern. Dabei wird entweder nur das SMTP „Mail From“ Feld, nur der Absender im Header oder beides überprüft.
Diese sehr primitive Form der Spamfilterung ist nur dann sinnvoll, wenn man sicher sein kann, dass sich diese Merkmale nicht verändern. Es ist z. B. meist weniger zeitaufwendig, eine Maillinglist-Adresse in die Blacklist einzutragen, als sich von dieser abzumelden.
Gewöhnlichen Spam kann man mit Blacklist nur sehr begrenzt erkennen, da Spammer selten den gleichen Absender, Betreff oder Inhalt verwenden.

5.4.2. Weiße Listen (Whitelists)

Das Gegenstück zu den schwarzen Listen stellen die weißen Listen dar. Weiße Listen werden geführt, um Falsch-Positive zu vermeiden, d. h. die Spamfilter-Maßnahmen werden außer Kraft gesetzt, wenn sich der Absender auf einer weißen Liste befindet. Man könnte z. B. Absender von wichtigen Geschäftspartnern auf diese Liste setzen, um sicherzustellen, dass diese E-Mails nicht verloren gehen. Bei clientbasierenden Lösungen werden häufig die Einträge im Adressbuch automatisch auf die Whitelist gesetzt.
Einige Spam-Filter-Lösungen bieten die Möglichkeit, Adressen von ausgehenden E-Mail automatisch auf die Whitelist zu setzen.
Analog zu den schwarzen Listen wird dabei entweder das SMTP „Mail From“, der Absender im Header oder beides überprüft. Da diese Angaben beliebig gesetzt werden können, kann es unter Umständen passieren, dass Spammer zufällig einen Absender aus der Whitelist wählen und so alle Spamfilter-Maßnahmen nutzlos machen.

5.4.3. Channels

Das Konzept von Channels wird bereits seit langem von Instant-Messaging Programmen verwendet, um unverlangte Nachrichten zu vermeiden. Diese Programme lassen in der Regel nur Nachrichten von Personen zu, die sich bereits in der Kontaktliste befinden [53].
Diesen Ansatz kann man auch auf die E-Mail-Kommunikation übertragen, indem Nachrichten von nicht bekannten Absendern abgewiesen werden. Diese sehr restriktive Vorgangsweise funktioniert bei Instant-Messaging Lösungen sehr gut, da der Erstkontakt meistens bereits anderweitig erfolgt ist.
Um diese Methode auch für E-Mails praktikabel zu machen, wurden Mechanismen entwickelt, die das Hinzufügen zu den weißen Listen automatisieren. Dabei wird beim Erstkontakt eine automatische Antwort-Mail generiert und an die im Header angegeben Antwort-Adresse gesandt. In dieser Nachricht wird der Absender gebeten, sich als „echter“ Absender zu validieren. Dies geschieht entweder durch das Antworten auf diese Nachricht oder durch das Eingeben eines bestimmten Codes auf einer bestimmten Website. Nach erfolgter Bestätigung wird die Absender E-Mail-Adresse automatisch in die Whitelist aufgenommen.
Dieses Verfahren birgt jedoch einige gravierende Fallstricke. Generell ist es keine gute Idee auf eintreffende Nachrichten automatisch zu antworten, da die Absender-Adressen von Spam-E-Mails in der Regel gefälscht sind und die generierten Antworten den Falschen erreichen und Unschuldige mit Tausenden von Nachrichten bombardieren würden. Die dabei zusätzlich notwendige Bandbreite könnte außerdem zusätzliche Kosten verursachen.
Ebenfalls problematisch ist die Anmeldung bei Mailinglisten, da die Absender- und Antwort-Adressen vielfach unterschiedlich sind. Im schlimmsten Fall endet eine Anmeldung bei einem Newsletter-Dienst mit einer gefährlichen Endlosschleife, bei der sich der Spam-Filter und der Newsletter-Daemon endlos autogenerierte Nachrichten zusenden. Dasselbe würde passieren, wenn beide Seiten Channels einsetzen.
Des Weiteren werden häufig autogenerierte Nachrichten versandt, auf die gar keine Antwort erwünscht wird, beispielsweise Anmeldebestätigungen bei Webshops oder Backup Statusmeldungen. E-Mails, die an die angegebene Antwort-Adresse gesandt werden, gehen meistens verloren und werden von niemanden gelesen. Die geforderte Validierung der Absender-Adresse ist somit nicht mög-

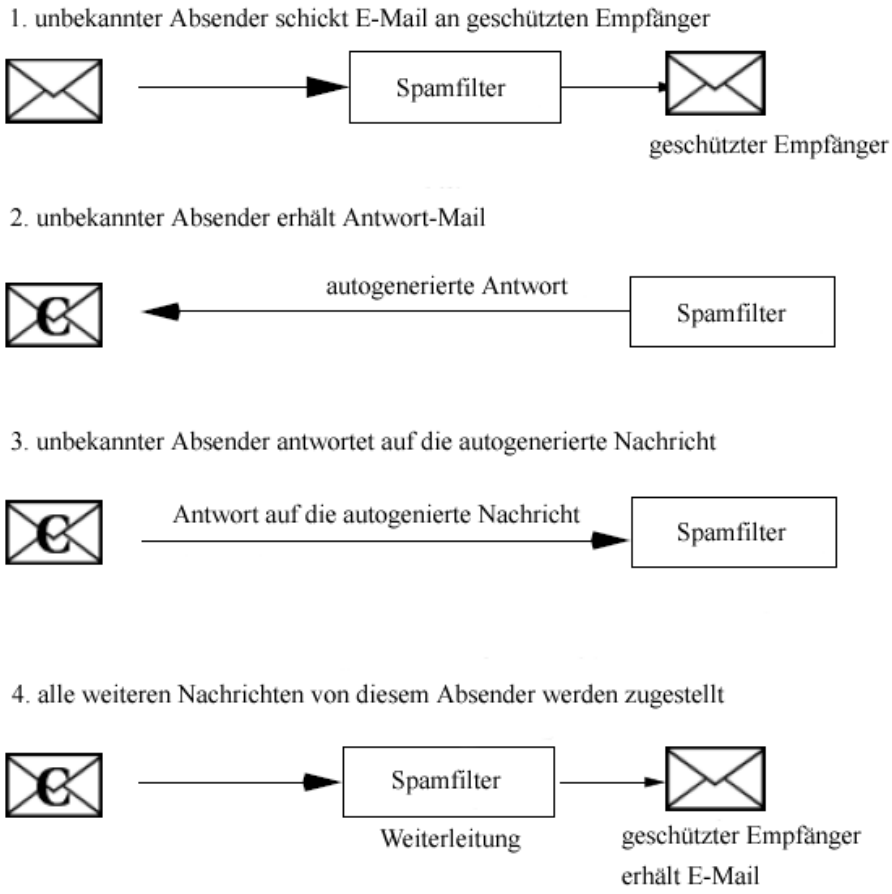

Abbildung 5.1.: Funktionsweise von Channels

lich.
Die Verwendung von Channels ist eigentlich nur in Ausnahmefällen zu empfehlen, da die dadurch verursachten Probleme meist den Nutzen übersteigen [49].

5.4.4. Regelbasierte Filter

Wie der Begriff schon verrät, geschieht die Erkennung von Spams anhand vordefinierter Regeln. Meistens geschieht dies mit Hilfe von regulären Ausdrücken. Diese sollen die Charakteristik von Spam möglichst genau beschreiben. Jede Regel erhält eine bestimmte Gewichtung. Bei Überschreiten eines bestimmten Grenzwertes wird die Nachricht folglich als Spam klassifiziert.
Beispielsweise könnte eine Regel dahingehend definiert werden, alle E-Mails zu erkennen, die ausschließlich Großbuchstaben im Betreff verwenden oder Nach-

richten, die Java-Script enthalten.
Regelbasierte Filter stellen eine gute Ergänzung zu bestehenden Spam-Filtern dar, aufgrund ihrer Starrheit ist die Erkennungsrate von nur auf Regeln aufbauender Filterlösungen jedoch zu gering.

5.4.5. Verteilter Prüfsummenfilter

Streng genommen stellen verteilte Prüfsummenfilter (engl. Distributed Checksum Clearinghouse - DCC) keine Spamfilter dar, sondern erkennen allgemein in Massen versandte E-Mails. Für jede empfangene E-Mail wird eine spezielle Prüfsumme berechnet, welche anschließend an eine zentrale Prüfstelle (Clearinghouse) weitergeleitet wird. Als Antwort erhält man die Anzahl der bereits versandten Nachrichten, anhand derer entschieden werden kann, ob es sich um Spam handelt oder nicht. Bei der Prüfstelle kann es sich entweder um einen lokalen Rechner handeln oder um einen der unzähligen Server im Internet.
Das besondere bei dieser Methode ist die Berechnung der Prüfsumme. Kleine Änderung in der Nachricht wie Personalisierungen wirken sich nicht auf die berechnete Checksumme aus. Man spricht hier von „fuzzy checksums“ (unscharfe Prüfsummen).
Herstellerangaben zufolge bewältigen die derzeit 300 DCC-Rechner über 300 Millionen E-Mails pro Tag. Dabei werden ca. 70 % der Nachrichten als Massenmails identifiziert. Die große Beliebheit dieser Filtertechnik liegt nicht nur in der guten Erkennungsrate, sondern auch im geringen Bandbreitenverbrauch. So umfasst eine UDP Abfrage lediglich 150 bytes. So wird erst ab einer Nachrichtenanzahl von täglich über 100.000 empfohlen, den Prüfsummen-Server im lokalen Netzwerk zu betreiben [16].
Der größte Nachteil dieser Filtermethode liegt darin, dass reguläre Massenaussendungen wie Newsletter ebenfalls als Spam eingestuft werden können [50].

5.4.6. Realtime Blackhole Lists (RBLs)

Realtime Blackhole Lists - oft auch DNS Blacklist genannt - sind öffentlich zugängliche schwarze Listen von IP-Adressen, die benutzt werden können, um eingehende E-Mails als Spam zu klassifizieren. Diese Listen können durch eine

Tabelle 5.1.: Erkennungsrate von Spamhaus und Spamcop [57]

Blacklist	Spam hits	Erkennungsr. %	Ham hits	Fehlerquote %
Spamcop	156.194	49,37	0	0,00
Spamhaus	255.521	80,77	5	0,10
Spamc. + Spamh.	267.795	84,65	5	0,10

Zeitraum:	ca. 74 Tage
Gesamt Spam:	316.348
Gesamt Ham:	4.999

einfache DNS Abfrage kontaktiert werden. Durch die einfache Implementierung und den geringen Bandbreitenverbrauch kommt dieses Verfahren sehr häufig zur Anwendung.
Für gewöhnlich sind in DNS-Blacklists IP-Adressen aufgeführt, über die in der Vergangenheit häufig Spam versendet wurde. Einige Listen führen auch Einträge von so genannten „Open Relays“ (oft auch Open Proxies genannt), also von Mail-Servern, die nur unzureichend gegen Missbrauch geschützt sind. Darunter fallen auch von Schadsoftware übernommene Rechner. Die Listen von SORBS [17] gehen noch einen Schritt weiter und nehmen zusätzlich noch ganze Adressbereiche von dynamischen vergebenen IP-Adressen auf.
Realtime Blackhole Lists sind auch deshalb sehr beliebt, da die Spam-Abwehr bereits zu einem Zeitpunkt möglich ist, wo die Nachricht noch gar nicht empfangen wurde. Sobald versucht wird, eine SMTP-Verbindung aufzubauen, kann die Client IP-Adresse benutzt werden, um eine oder mehrere schwarze Listen zu befragen. Befindet sich die IP-Adresse in einer der Listen, kann man den SMTP-Dialog bereits mit einer Fehlermeldung beenden.
Das Institut für Internet-Sicherheit an der Fachhochschule Gelsenkirchen untersuchte das Abfrageverhalten von Blacklists. Dabei wurde unter anderem ermittelt, dass in der Mehrheit aller Fälle dieselbe IP-Adresse nur über einen sehr kurzen Zeitraum verwendet wird. Gut 75 % der von Spammern verwendeten IP-Adressen wurden lediglich innerhalb eines Tages benutzt, nur 4 % länger als über einen Zeitraum von drei Tagen. Eine Blacklist kann also nur dann effektiv arbeiten, wenn die eingetragenen IP-Adressen sehr aktuell sind [33].
IP-basierte schwarze Listen kamen in der Vergangenheit immer wieder in den Schlagzeilen vor, da sie es mit der Spam-Bekämpfung etwas zu gut meinten. So landete z. B. Mitte 2007 der gesamte IP-Adressbereich von nic.at auf den schwar-

zen Listen von Spamhaus, nur weil nic.at sich weigerte, einige für Phishing-Attacken missbrauchte Rechner vom Netz zu nehmen. Da die Listen von Spamhaus häufig eingesetzt werden, waren sämtliche Mail-Server praktisch von der Außenwelt abgeschnitten. Man sollte also die zur Anwendung kommenden Listen mit Bedacht auswählen und vorher überprüfen, wer hinter solchen Listen steckt und wie die Aufnahmekriterien lauten.
Eine Möglichkeit zur Verhinderung von Falsch-Positiven bestünde z. B. darin, die Resultate von mehreren qualitativ hochwertigen Blacklists miteinander zu verknüpfen und nur IP-Adressen zu blocken, die nicht nur in einer Listen vorkommen. Von der Verwendung von Listen, die ganze dynamische Adressbereiche ohne vorherige Prüfung aufnehmen, ist generell abzuraten, da zu viele legitime IP-Adressen fälschlicherweise gelistet werden. Spam-Experten empfehlen auch die zusätzlichen Verwendung von Positivlisten wie dnswl.org, um die Falsch-Positiv-Rate noch weiter zu senken [51].

5.4.7. Realtime URI Blacklists (URIBL)

Im Gegensatz zu Realtime Blackhole Lists beinhalten diese Listen keine IP-Adressen, sondern URLs von Websites, die in Spam-Nachrichten auftreten. Um diese Methode anwenden zu können, muss der Nachrichteninhalt zuerst vollständig empfangen werden. Das Verfahren kann natürlich nur Spam erkennen, sofern im Nachrichten-Body auch eine URL vorkommt. Andererseits funktioniert dieses Verfahren aber unabhängig davon, von welcher IP-Adresse die E-Mail kommt und erreicht Erkennungsraten von 80 % bis 90 % bei Falsch-Positiv Raten von 0,001 % bis 0,05 % [18]. Analog zu den IP basierten Blacklists kann diese Methode nur gut funktionieren, wenn die Listeneinträge sehr aktuell sind.
Beispielsweise enthält eine E-Mail folgenden Text:

```
HIGH QUALITY REPLICA WATCHES:
http://rsc.zwindependent.com
```

Um nun eine URI-Blacklist Abfrage (z. B. auf multi.surbl.org) durchführen zu können, muss zuallererst der Domainname aus dem Nachrichten-Body extrahiert werden. Ist dies geschehen, so kann mit Hilfe des kombinierten Domainnamens - in diesem Fall zwindependent.com.multi.surbl.org - versucht werden, diesen

mittels DNS-Abfrage aufzulösen. Wird ein gültiger DNS A-Record zurückgeliefert, ist die entsprechende URL gelistet ansonsten nicht.
Spammer versuchen solche Filter zu umgehen, indem sie unzählige Weiterleitungs-URLs einrichten, die sich häufig ändern. Für obiges Beispiel würde das z. B. bedeuten, dass sich der eigentliche Inhalt der Spam-Seite nicht unter http://rsc.zwindependent.com befindet, sondern die Anfrage lediglich an die Zieldomäne weitergeleitet wird.
Weitere Möglichkeiten solche Filter zu verwirren, bestehen darin, Spam-Mails auch mit legitimen URLs zu versehen oder URLs durch IP-Adressen zu ersetzen [59].

5.4.8. Greylisting

Bei Greylisting handelt es sich um ein Verfahren, das erstmalige Zustellversuche von unbekannten Absendern temporär abweist und erst eine spätere Zustellung akzeptiert. Korrekt implementiere Mailserver können mit einer temporären Abweisung problemlos umgehen und unternehmen auf jeden Fall mehrere Zustellversuche. Spam-Systeme verfügen in der Regel nicht über derartige Mechanismen, da das Betreiben von Warteschlangen einen enormen Ressourcenverbrauch darstellen würde. Dies bedeutet, dass die temporäre Abweisung (SMTP-Returncode 451) in diesem Fall einer permanenten Abweisung gleichkommt. Theoretisch sollte diese Verfahren keine Falsch-Positive erzeugen. Es gibt aber leider doch falsch konfigurierte Mail-Server, die keinen weiteren Zustellversuch unternehmen.
Ein weiterer Zustellversuch wird nur dann akzeptiert, wenn dieser innerhalb eines gewissen Zeitfensters passiert. Typische Werte sind z. B. 30 Minuten bis 24 Stunden, d. h. ein weiterer Versuch ist nur gültig, wenn dieser nach frühestens 30 Minuten und spätestens innerhalb von 24 Stunden durchgeführt wird. Wurde ein gültiger Zustellversuch registriert, wird die E-Mail dem Emfpänger zugestellt und der Absender auf die Whitelist gesetzt, d. h. zukünftige E-Mails von diesem Absender werden nicht mehr verzögert. Treffen von diesem Absender über längere Zeit keine E-Mails mehr ein, wird der Eintrag wieder aus der weißen Liste entfernt.

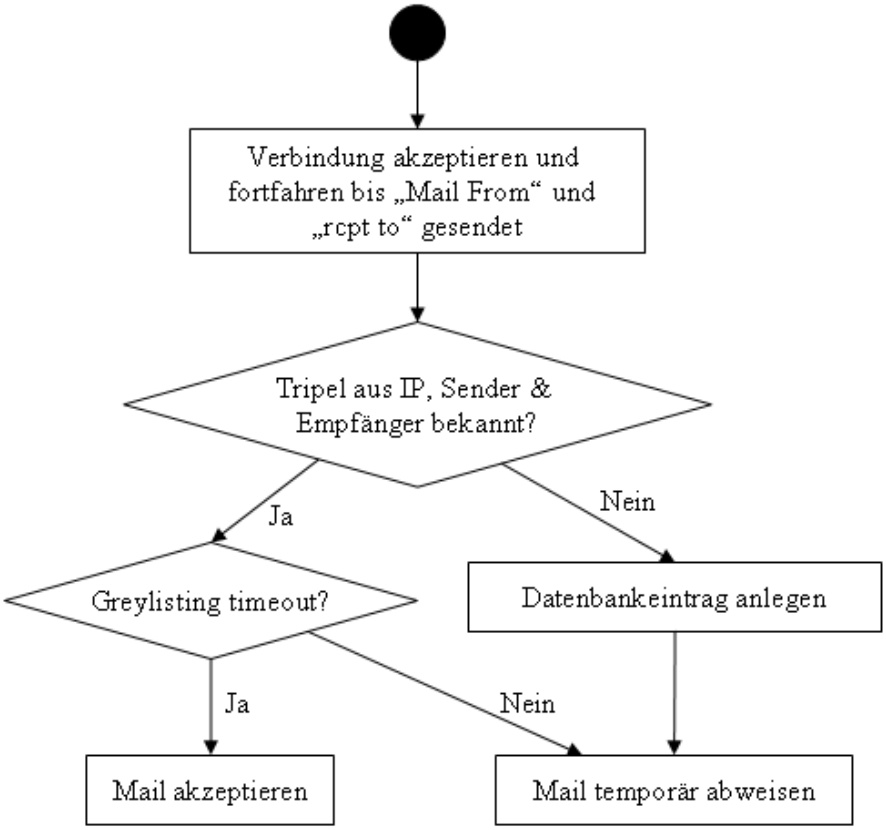

Abbildung 5.2.: Funktionsweise von Greylisting

Greylisting identifiziert einen Absender typischerweise anhand folgendem Tripel:

1. IP-Adresse des absendenden Mail-Servers
2. E-Mail-Adresse des Senders (MAIL FROM:)
3. E-Mail-Adresse des Empfängers (RCPT TO:)

Einige E-Mail-Provider z. B. Google verfügen über große Rechenzentren, in denen der neuerliche Versand jeweils durch Loadbalancing angestoßen wird und somit ist die IP-Adresse des absendenden Mailserver unter Umständen immer verschieden. Es kann dadurch passieren, dass sich der Mail-Versand über Stunden hinzieht. Um diese negativen Folgen zu minimieren, kann man anstelle der vollständigen IP-Adresse den gesamten Netzblock heranziehen.
Nachfolgend sei ein erfolgreicher Zustellversuch (grey time = 5 Minuten) von chello.at angeführt. Auffallend dabei die äußert kurzen Intervalle zwischen den einzelnen Zustellversuchen:

```
11:19:00 213.46.255.22 temporär abgewiesen
11:19:30 213.46.255.22 temporär abgewiesen
11:19:32 213.46.255.22 temporär abgewiesen
11:19:34 62.179.121.45 temporär abgewiesen
11:19:36 62.179.121.46 temporär abgewiesen
11:19:52 62.179.121.31 temporär abgewiesen
11:20:59 213.46.255.22 temporär abgewiesen
11:21:01 213.46.255.22 temporär abgewiesen
11:36:39 62.179.121.45 angenommen
```

Weitere Schwierigkeiten bereitet Greylisting Mailinglisten, die VERP (Variable Envelope Return Paths) [22] verwenden und den SMTP-Absender ständig variieren. Das hat zur Folge, dass es für Nachrichten von solchen Mailinglisten nie ein gültiges Tripel gibt und diese jedes mal verzögert eintreffen. Dasselbe Problem haben Methoden wie BATV (siehe eigenen Abschnitt), bei denen die Absender-Adresse ebenfalls häufig wechselt [66].

```
1 telnet mail.mydomain.net 25
2 server: 220 mail.mydomain.net Mail Server
3 client: HELO spammer.domain.com
4 server: 250 mail.mydomain.net
5 client: MAIL FROM: from@spammer.net
6 server: 451 <from@spammer.net>: Recipient address rejected
7     Greylisting in action.
```

SMTP-Dialog bei aktivem Greylisting

Die Vorteile von Greylisting liegen nicht nur in der Bandbreitenersparnis (die temporäre Abweisung erfolgt bereits, bevor die eigentliche Nachricht auf dem Mail-Server eingelangt ist), sondern auch in der recht einfachen Implementierung. Auf der anderen Seite zahlt man durch die verspätete Zustellungen einen relativ hohen Preis. Verzögerungen in der Zustellung, die manchmal einige Stunden betragen können, sind in vielen Fällen nicht akzeptabel. Eine Möglichkeit, die negative Wirkung von Greylisting zu vermindern, wäre z. B. diese Methode nur auf nicht bekannte IP-Adressen anzuwenden und die bereits als „gutmütig“ eingestuften Mail-Server von Greylisting auszunehmen.
Bei gehackten Mailservern, die über eine eigenen Warteschlange verfügen, zeigt

dieses Verfahren keine Wirkung, da auf jeden Fall mehrere Zustellversuche unternommen werden [78].

5.4.9. Teergruben

Das Geschäftsmodell von Spammern funktioniert nur, wenn diese sehr viele Nachrichten fast kostenlos versenden können. Teergruben versuchen das Empfangen von Nachrichten künstlich zu verlangsamen und den unerwünschten Kommunikationspartner so lange wie möglich zu blockieren. Dieses Verhalten irritiert einen korrekt konfigurierten Mail-Server kaum. Spammer jedoch müssen in kürzester Zeit möglichst viele Nachrichten verschicken. Eine Teergrube, die den sendenden Mail-Server für Sekunden oder gar Minuten „gefangen“ hält, kann den Spammer dadurch empfindlich in seiner Arbeit stören.

```
1  telnet mail.mydomain.net 25
2  S: 220 mail.test.at SMTP server ready
3  C: HELO xyz.at.
4  S: 250-xyz.at., send us your mail but not your spam
5  S: 250-This server understands these commands.
6  S: 250-PIPELINING
7  S: 250-SIZE 10240000
8  S: 250-VRFY
9  S: 250-ETRN
10 S: 250-HELP  (Dieses Spiel kann man beliebig lange fortsetzen)
11 S: 250 Use HELP command to get specific help
```

SMTP-Dialog in einer Teergrube

Modifiziert man das SMTP-Protokoll dahingehend, dass in Abständen von einigen Sekunden Fortsetzungszeichen gesendet werden, so wird die Verbindung quälend langsam, ohne dass dabei ein Timeout eintritt. Ein gewöhnliches SMTP-Commando besteht normalerweise aus dem Fehlercode, einem Leerzeichen und einem für den Menschen lesbaren Text. Fügt man zwischen Fehlercode und Text ein Minuszeichen ein, so wird dem Client signalisiert, dass der Server mit der Antwort noch nicht fertig ist und die Verbindung nicht abgebrochen werden darf. Inzwischen haben sich Spammer jedoch auf Teergruben eingestellt und beenden die Verbindungen bei Verzögerungen lieber gleich [39]. Die Teergrube verliert so

zwar den ursprünglichen Zweck, als Schutz für den eigenen Mail-Server eignet sich eine Teergrube aber allemal [80].
Setzt man eine Teergrube zur Spambekämpfung ein, so muss man darauf achten, dass die maximale Anzahl von gleichzeitigen SMTP-Verbindungen nicht zu niedrig eingestellt wird, da durch die verlangsamten SMTP-Dialoge sehr schnell das maximale Sitzungslimit erreicht werden kann. Die negative Wirkung von Teergruben lässt sich minimieren, indem man bereits bekannte gutmütige Hosts an der Teergrube vorbeischleust [40].

5.4.10. Bounce Address Tag Validation (BATV)

Bounce Address Tag Validation ist kein Spamfilter im eigentlichen Sinne, sondern stellt einen Mechanismus zur Verfügung, der es ermöglicht, selbst ausgelöste Fehlermeldungen von solchen zu unterscheiden, die durch Adressfälschungen hervorgerufen wurden.
Wie bereits mehrfach erwähnt, benutzen Spammer keine echten Absender-Adressen. Entweder werden diese zufällig erzeugt oder Spammer verwenden Absender-Adressen von unschuldigen Benutzern. Da ein Großteil der Spam-Nachrichten nicht zugestellt werden kann, endet dies mit einer Fehlermeldung. Wird die unerwünschte Nachricht bereits während des SMTP-Dialogs abgelehnt, passiert in der Regel nichts, da der absendende Spam-Mail-Server keine korrekte Fehlerbehandlung durchführt und somit keine Bounce-Meldung generiert. Nimmt der Mail-Server die Nachricht jedoch an, so muss eine etwaige Fehlermeldung laut RFC [69] an die während des SMTP-Dialoges angegebe „Mail From“ Adresse zurückgesandt werden. In nicht wenigen Fällen werden dadurch Unschuldige in kürzester Zeit mit Tausenden Fehlermeldungen bombardiert.
Kommt das BATV Verfahren zur Anwendung, so werden in das „Mail From“ Feld weitere Informationen kodiert, die eine spätere Bounce-Validierung ermöglichen. Nachfolgend die Syntax einer BATV-kodierten Absender-Adresse:

```
batv-mail-from =  <local-part> "@" <orig. domain-part>
local-part     =  "prvs=" <tag-val> "=" <orig. local part>
tag-val        =  K DDD SSSSSS
K              =  1 Ziffer
                  ; Schlüssel Nummer, um eine
                  ; Schlüsselrotation zu ermöglichen
```

```
DDD             =  3 Ziffern
                   ; Letzten drei Stellen der verstrichenen
                   ; Tage seit 1970 an dem die Adresse
                   ; ihre Gültigkeit verliert
SSSSSS          =  6 HEXDIG
                   ; Hexadezimaler Wert der ersten drei
                   ; Bytes des Hashwertes SHA-1 HMAC aus
                   ; <hash-source> und eines geheimen Schlüssels
hash-source     =  K DDD <orig. Mail From>
```

Beispielsweise könnte die BATV-codierte E-Mail-Adresse für thomas.rainer@test.it folgendermaßen aussehen:

```
<prvs=0014aaec00=thomas.rainer@test.it>
```

Werden ausgehende und eingehende Mails nicht über gleichen Mail-Provider geroutet, so kann diese Methode nicht angewandt werden.
Da BATV nicht sehr weit verbreitet ist, kann es zu Problemen mit einigen Mailinglisten kommen. Nicht selten identifizieren die Listen ihre registrierten Benutzer anhand der „Mail From“ Adresse, die das BATV-Verfahren immer wieder anders erzeugt. Weitere Probleme können falsch konfigurierte Mail-Server verursachen, die etwaige Fehlermeldungen nicht an die SMTP „Mail From“ Adresse senden, sondern an die angegebene From-Adresse im E-Mail-Header. Einige Mail-Server verhalten sich auch nicht RFC konform und senden Fehler- oder Abwesenheitsmeldungen nicht an die „Mail From“ Adresse sondern an die Adresse im E-Mail-Header [65][58].

5.4.11. Sender Policy Framework - SPF

Da das SMTP-Protokoll sehr schwerwiegende Designschwächen besitzt, wurden verschiedene Techniken entwickelt, die dessen Mankos beheben sollen. Darunter fällt z. B. das Sender Policy Framework (ursprünglich „Sender Permitted From“ genannt), durch das Adressfälschungen auf SMTP-Ebene erschwert werden sollen. Absender-Informationen, die im Mail-Header abgelegt werden, sind von dieser Prüfung ausgenommen.

Diese Verfahren greifen nur, wenn sowohl der sendende Host als auch der Mail-Server auf der Empfängerseite dieses Verfahren implementiert haben. Der Mail-Sender muss sicherstellen, dass im DNS der Absender-Domaine ein Resource Record vom Typ SPF oder TXT hinterlegt wurde, in dem die IP-Adressen aller Mail-Server aufgelistet sind, über die der Versand erfolgen darf. Nachfolgend der DNS TXT Eintrag von GMX [1]:

```
gmx.net. TXT "v=spf1 ip4:213.165.64.0/23 ip4:74.208.5.64/26 -all"
```

Der empfangene Mail-Server muss nun durch eine DNS-Abfrage am „Mail From“ Host überprüfen, ob die eingehende Verbindung auch von einem legitimierten Server stammt [11].
Analog zu BATV muss das Senden der Nachrichten über den Provider erfolgen, der für die entsprechende Domain zuständig ist.
Probleme verursachen kann SPF bei Weiterleitungen, bei denen das SMTP „Mail From“ vom weiterleitenden Server nicht neu gesetzt, sondern vom ursprünglichen Mail-Server übernommen wird. Nachfolgend ein Beispiel:

```
1 MAIL FROM: <alice@domain1>
2 RCPT TO: <bob@domain2>
3 DATA
4 From: Alice <alice@domain1>
5 To: Bob <bob@domain2>
6 Subject: SPF vs. Sender ID
7 [...]
```

Der Mailserver von domain2 nimmt die E-Mail von alice@domain1 entgegen. Bob hat aber eine Weiterleitung auf bob@domain2 eingerichtet, an die der Mailserver die Nachricht weiterleitet. Nun hat der Benutzer bob allerdings eine Weiterleitung an bob@domain3 eingerichtet. Der weiterleitende Server benutzt aber die ursprüngliche „MAIL FROM“ Identität:

```
1 MAIL FROM: <alice@domain1>
2 RCPT TO: <bob@domain3>
3 DATA
4 From: Alice <alice@domain1>
5 To: Bob <bob@domain2>
6 Subject: SPF vs. Sender ID
7 [...]
```

[1] http://www.gmx.net

Der Mailserver von domain3 hat SPF implementiert und nimmt während des SMTP-Dialogs die Überprüfung vor, indem er eine DNS-Anfrage an domain1 abstößt. Dort sind allerdings nur die Mailserver von domain1 hinterlegt. Von domain2 dürfte also keine E-Mail mit dieser „Mail From“ Identität eintreffen und somit wird der Absender als gefälscht eingestuft.
Eine ähnliche auf dem SPF-Verfahren aufbauende Methode wurde von Microsoft entwickelt und nennt sich Sender-ID. Im Gegensatz zu SPF überprüft das Sender-ID Verfahren nicht nur die SMTP Absender-Daten sondern auch die Informationen im Header [10].
Beide Verfahren erschweren zwar grundsätzlich Adressfälschungen, man kann aber keinen Spammer daran hindern, korrekte SPF-Einträge für die eigenen Domains vorzunehmen.

5.4.12. DomainKeys Identified Mail (DKIM)

DKIM stellt ein System zur Verifizierung des Absenders und der Nachrichtenintegrität dar. Es wurde 2004 von Yahoo entwickelt und soll Absenderfälschungen durch Spammer erschweren. Basierend auf asymmetrischer Verschlüsselung (RSA) wird jede ausgehende E-Mail mit einer digitalen Signatur versehen, die der empfangene MTA überprüfen kann.
Der sendende Mailserver erzeugt die digitale Signatur mittels des SHA-1-Hash der Nachricht und anschließender Verschlüsselung mit seinem privaten Key. Der so erzeugte Schlüssel wird im Feld „DomainKey-Signature“ des Headers abgelegt. Der empfangende Host seinerseits entschlüsselt die digitale Signatur mittels des im DNS der Absender-Domäne hinterlegten öffentlichen Keys und kalkuliert den SHA-1-Hash der Nachricht neu. Stimmen beide Werte überein, kann man davon ausgehen, dass die Absender-Adresse echt und die Nachricht während des Transports nicht verändert wurde.
Das DomainKeys Verfahren verwendet zur Verifizierung nur die im Mail-Header vorkommende E-Mail-Adresse. Es kann also unabhängig vom SMTP-Protokoll eingesetzt werden und erzeugt keine Probleme bei E-Mail-Weiterleitungen. Ähnlich dem Sender Policy Framework wird es aber noch sehr lange dauern, bis sämtliche Mail-Server solche Verfahren implementiert haben [3][37].

5.5. Bayessche Filter

Um dem immer größeren Spamaufkommen Einhalt zu gebieten, mussten und müssen starre, regelbasierte Filter immer wieder angepasst werden. Spammer wurden immer trickreicher und versuchten, solche Filter mit Textverstümmelung und ständig wechselndem Inhalt zu überlisten. Paul Graham erläuterte erstmal 2002 die Möglichkeit der Spamfilterung mittels eines statistischen Verfahrens basierend auf dem bayesschen Wahrscheinlichkeitsbegriff [47]. Anhand der Auftrittswahrscheinlichkeit einzelner Wörter in bereits gelernten Spam- und Ham-Nachrichten, kann zukünftig eine Entscheidung getroffen werden, ob es sich um Spam handelt oder nicht. Dabei werden nicht alle Wörter zur Berechnung herangezogen, in Regel sind es lediglich 15 oder 27 [96].
Im Gegensatz zu anderen Filtern, muss man Bayessche Filter zuerst „einlernen“, d. h. man muss diese mit einer gewissen Anzahl vorhandener Spam- und Ham-Nachrichten trainieren, um zukünftige Nachrichten vernünftig klassifizieren zu können. Sinken die Erkennungsraten oder macht der Filter Fehler, so reicht es, wenn man den Filter mit neuen Spam und Ham-Nachrichten füttert.
Bayessche Filter funktionieren am besten, wenn sich jeder Benutzer eine eigene Lerndatenbank erzeugt, die individuell auf diesen abgestimmt ist. Da ein serverbasiertes, individuelles Lernen nur sehr schwer zu implementieren und resourcenhungrig ist, werden solche Filter meist auf der Benutzer Seite, d. h. im E-Mail-Client umgesetzt. Mittlerweile gibt es für sämtliche E-Mail-Programme kostenlose Plugins [2] [77].

5.5.1. Berechnung der Wortwahrscheinlichkeiten

Bevor die Berechnung der Spam-Wahrscheinlichkeit erfolgen kann, müssen die Nachrichten auf ihre Merkmale hin untersucht werden. Dies erfolgt durch die Aufteilung der Nachricht in Tokens. Im einfachsten Fall entspricht ein Token einem einzelnen Wort. Im Abschnitt „Tokenizing“ werden erweiterte lexikalische Analyseverfahren beschrieben.
Jedem Token wird basierend auf dessen historischem Auftreten die Spamwahr-

[2] http://spambayes.sourceforge.net

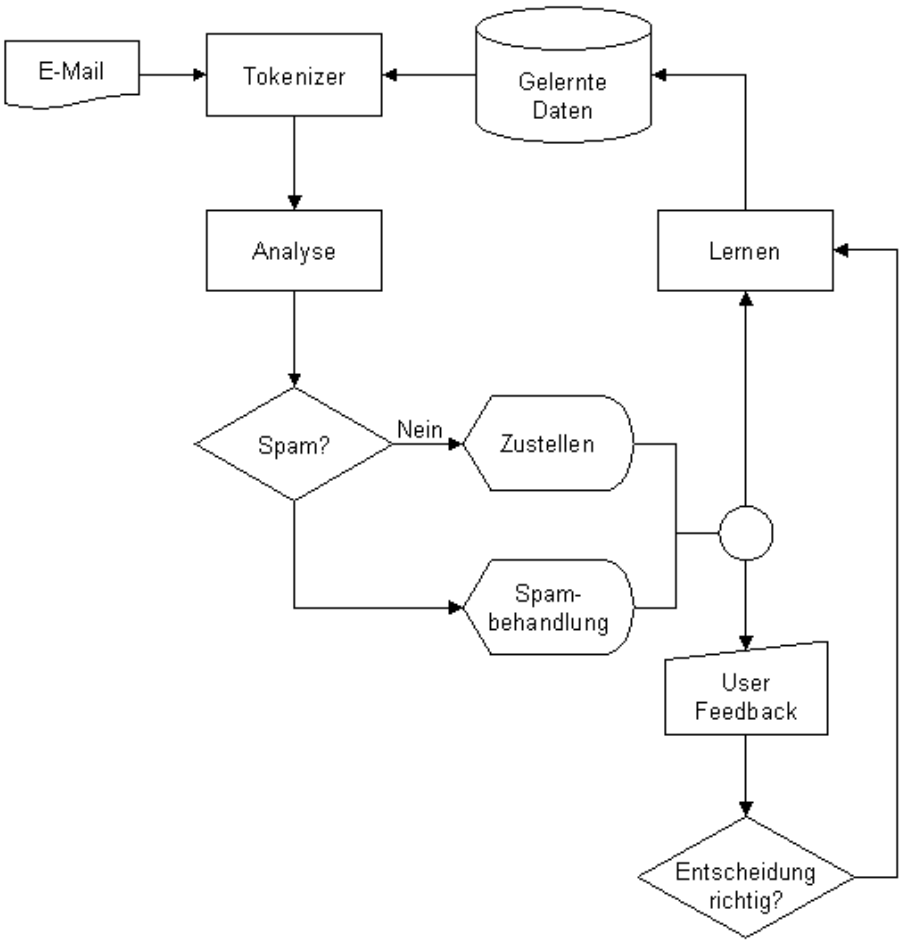

Abbildung 5.3.: Schematische Darstellung eines lernenden Filters

scheinlichkeit zugeordnet. Die von Paul Graham entwickelte Formel lautet:

$$P = \frac{SH/TS}{(SH/TS) + (IH/TI)}$$

SH .... Vorkommen in Spam E-Mails
IH .... Vorkommen in legitimen E-Mails
TS .... Anzahl der Spam E-Mails
TI .... Anzahl der legitimen E-Mails

Das Resultat dieser Form liegt zwischen 0 und 1, wobei Token mit einem Wert größer als 0,5 als „schlecht“ und Token mit einem Wert kleiner als 0,5 als „gutmütig“ angesehen werden.
Zum Beispiel hat unsere historische Datenbank 224 Spam-Nachrichten und 112

legitime Nachrichten gelernt. Das Wort „the“ kam 96 Mal in Spam-E-Mails und 48 Mal in Ham-E-Mails vor. Für diesen Token ergäbe sich eine Wortwahrscheinlickkeit von 0,5, was „neutral“ bedeuten würde.

5.5.2. Kombinieren der Wortwahrscheinlichkeiten

Wie man anhand des vorherigen Beispiels gut sehen kann, eignen sich nicht alle Token gleich gut, um auf die Spamwahrscheinlichkeit einer E-Mail zu schließen.

Graham Paul Graham verwendet in seinem Algorithmus die 15 aussagekräftigsten Token. Das sind die Token, deren Wortwahrscheinlichkeit am weitesten vom neutralen Wert 0,5 entfernt ist. Diese 15 Werte werden dann anschließend mit dem Bayesschen Theorem miteinander kombiniert:

$$P = \frac{ABC...N}{ABC...N + (1-A)(1-B)(1-C)...(1-N)}$$

A,B,C,...,N .... Wortwahrscheinlickkeiten

Das Resultat dieser Berechnung ist meist ein Wert der entweder sehr Nahe an 0,0 oder sehr Nahe an 1,0 liegt. Dieser Algorithmus trifft also meist eine sehr klare Entscheidung [96].

Burton Der Autor der Software SpamProbe [23] analysierte den statistischen Ansatz von Graham und führte einige Verbesserungen durch. Zum einen wurde die Anzahl der zur Berechnung herangezogenen Tokens auf 27 erhöht. Zum anderen kann ein Token in der Liste der wichtigsten Token auch zweimal vorkommen. Burton erkannte nämlich das grundlegende Problem des Graham Algorithmus: Kommen zu viele stark gewichtete Tokens vor, so erfolgt die Entscheidung für Spam oder Nicht-Spam eher zufällig. Dasselbe Problem tritt auf, wenn die Nachricht zu wenig Tokens enthält. Dies hat zur Folge, dass viele unwichtige Wörter in die Berechnung miteinfließen. Kommen Token in der Nachricht öfter als einmal vor, können diese zweimal für die Berechnung verwendet werden und erhalten dadurch ein höheres Gewicht. Die Praxis hat ergeben, dass der Burton

Algorithmus eine bessere Erkennungsrate liefert als das Verfahren von Graham [96].

5.5.3. Lexikalische Analyse - Tokenizing

Der Einfachheit halber wurde in den vorhergehenden Abschnitten davon ausgegangen, dass ein Token einem einzelnen Wort entspricht. Um der Bayesschen Formel bessere Daten zuführen zu können, wurden weitere Methoden der Text-Analyse entwickelt.

uniGram - einzelnes Wort In der einfachsten Form der Analyse wird jedes einzelne Wort als Token interpretiert und unabhängig vom restlichen Inhalt betrachtet. Diese sehr einfache Methode der Text-Analyse wurde bereits 2002 von Paul Graham angewandt. Da keinerlei Zusammenhang zwischen den einzelnen Wörtern berücksichtigt wird, spricht man auch vom naiven Bayes-Klassifikator. Aufgrund der einfachen und schnellen Analyse ist diese Art von Klassifizierung nach wie vor sehr beliebt und erzielt ansprechende Erkennungsraten [47].

biGram - chained tokens Anstatt nur einzelne Wörter zu betrachten, werden mit den angrenzenden Wörtern zwei zusätzliche Tokens erzeugt. Dieses Verfahren ist also kein Ersatz der simplen Wort-Analyse, sondern kann als Ergänzung eingesetzt werden.

Tabelle 5.2.: Beispiel für die Spamwahrscheinlichkeit einzelner Wörter und eines Wortpaares

color	0.3282537758
#000000	0.5794490576
color * #000000	0.9684159160

Jedes Attribut separat betrachtet hat sehr wenig Aussagekraft über die Spamwahrscheinlichkeit, da diese sowohl in Spam als auch in Nicht-Spam E-Mails häufig vorkommen. Beide zusammen sind jedoch fast nur in Spam-Nachrichten zu finden [74].

Sparse Binary Polynomial Hashing Bei dieser Methode werden ebenfalls nicht nur einzelne Wörter betrachtet sondern ganze Text-Phrasen. Dabei wird für jede erdenkliche Kombination die Spam-Wahrscheinlichkeit berechnet und in die Lern-Datenbank aufgenommen. Aus Performance-Gründen wird in der Praxis die Länge der betrachteten Sequenzen limitiert. CRM114 [95] z. B. verwendet standardmäßig eine Sequenzlänge von max. fünf Wörtern. Beispielsweise würden für den Satz „Houston, we've got a problem" folgende Phrasen gebildet:

```
Houston
Houston we've
Houston <skip> got
Houston we've got
Houston <skip> <skip> a
Houston we've <skip> a
Houston <skip> got a
Houston we've got a
Houston <skip> <skip> <skip> problem
Houston we've <skip> <skip> problem
Houston <skip> got <skip> problem
Houston we've got <skip> problem
Houston <skip> <skip> a problem
Houston we've <skip> a problem
Houston <skip> got a problem
Houston we've got a problem
```

Die Erkennunsrate ist in der Praxis etwas besser als bei der einfachen Wort-Analyse, geht aber zu Lasten der Skalierbarkeit und Geschwindigkeit [74].

5.5.4. Lernmethoden

Der wohl wichtigste Aspekt bei lernfähigen Filter liegt im richtigen Lernen. Gute Erkennungsraten können nur erzielt werden, wenn die Informationen in der Lern-Datenbank qualitativ hochwertig sind. Dabei ist auch wichtig, dass sowohl

Spam-Mails als auch legitime Nachrichten trainiert werden. Anhand dieser Nachrichten werden Statistiken erzeugt, auf deren Basis zukünftige Entscheidungen getroffen werden.

Train Every Thing Bei der Methode „Train Every Thing“ (TET) wird jede Nachricht dem Filter zwecks Training zugeführt. Dies hat den Vorteil, das sich der lernende Filter sofort auf neue Gegebenheiten einstellen kann. Andererseits reagiert der Filter dadurch auf jede kleinste Veränderung, sodass eine gewisse Volatilität zu erwarten ist. Ist der Anteil an gelernten Spam-Nachrichten viel größer als von legitimen Nachrichten, kann TET auch vermehrt zu Falsch-Positiven führen.

Train only Error Wird die Methode „Train only Error“ (TOE) angewandt, erfolgt eine Anpassung der zugrunde liegenden Statistiken nur, sobald ein Fehler aufgetreten ist. Die Erkennungs- und Fehlerrate ist in den meisten Fällen besser als bei TET. Automatisches Lernen ist aber nur sehr schwer möglich, da Fehler meist nur vom Benutzer erkannt werden können.

Train Until Mature Die Methode „Train Until Mature“ (TUM) verknüpft die zwei vorher genannten Verfahren. Bis die gelernte Datenbank einen gewissen Umfang erreicht hat, wird jede Nachricht „gelernt“, nachher nur mehr dann, wenn ein Fehler aufgetreten ist [96].

5.5.5. Umgehungsversuche

Spammer versuchen auf verschiedenste Weise Bayessche Filter zu umgehen. Beispielsweise sollen die Filter durch Maskierung der Wörter oder Einfügen von Leerzeichen zwischen den einzelnen Buchstaben verwirrt werden.
Weiters versuchen Spammer unerwünschte Nachrichten mit unwichtigen Sequenzen aufzufüllen. In HTML-Nachrichten sind diese Sequenzen für den Benutzer meist unsichtbar, da sie entweder als Kommentar eingefügt, sehr klein geschrieben (Schriftgröße 1) oder als weißer Text auf weißem Hintergrund getarnt werden. Durch diesen zusätzlichen Text kann die Spam-Wahrscheinlichkeit sinken, da Bayessche Filter die Auftrittswahrscheinlichkeit der Wörter im gesamten Text

betrachten. Für die Entscheidung Spam oder Nicht-Spam wird meist jedoch nur eine begrenzte Anzahl von Tokens herangezogen.
Eine weitere Möglichkeit lernfähige Filter zu umgehen, besteht darin, den Text in Bilder einzubetten. Sofern keine OCR Erkennung stattfindet, kann der Inhalt nicht weiter untersucht werden. Man muss sich dann mit den wenigen Informationen im Nachrichten Body und Header begnügen.

5.6. Kombination aus mehreren Filtern

Wir haben nun eine Reihe technischer Maßnahmen kennengelernt mit all ihren Vor- und Nachteilen. Hersteller von Anti-Spam Filtern bewerben ihre Produkte mit Filterraten von über 99 % [3],[4]. Herstellerangaben sind zwar immer mit Vorsicht zu genießen, da diese den Begriff Spam nach eigenem Gutdünken definieren. Raten über 90 % lassen sich in der Regel aber nur erreichen, wenn verschiedene Filtermethoden kombiniert werden. Man könnte nun beispielsweise verschiedene Filter in Serie schalten, um die Erkennungsraten zu steigern. Wird also eine E-Mail von einem Filter als Spam erkannt, erfolgt keine weitere Validierung und die Nachricht wird nach den gewählten Spam-Richtlinien behandelt. Diese sehr starre Methode überzeugt wahrscheinlich durch sehr hohe Treffenquoten, andererseits ist die Wahrscheinlichkeit für Falsch-Positive erheblich größer.
Hohe Erkennungsraten bei gleichzeitig wenigen Falsch-Positiven kann man z. B. erreichen, indem man die Ergebnisse der einzelnen Filter unterschiedlich gewichtet und bei Überschreiten eines bestimmten Schwellwertes die E-Mail aussortiert. Das wohl populärste Beispiel für einen solchen Filter stellt das Opensource Projekt Spamassassin dar [15]. Eine fälschlicherweise angeführte IP-Adresse in einer der DNS basierten Blacklists hat dann nicht gleich zur Folge, dass die entsprechenden Nachrichten aussortiert werden.
Vor allem Klassifizierungen von Spam-Quellen, die nicht durch eigene Tests validiert wurden, sollte man mit einer gewissen Skepsis begegnen. So muss man Verbindungen, die von einem gelisteten Rechner stammen, nicht gleich gänzlich abweisen, sondern kann erst einmal Greylisting oder eine Teergrube anwenden.
Ergebnisse aus mehreren Filtern eignen sich aber nicht nur, um Falsch-Positive zu

[3] http://www.cleanmail.ch
[4] http://www.ikarus.at

verringern, man kann sie auch hervorragend dazu benutzen, um eigene schwarze oder weiße Listen aufzubauen. Trifft z. B. eine Spam-Nachricht von einem Server ein, der auf einer IP basierten Blacklist aufscheint, so kann man den Inhalt der E-Mail verwenden, um z. B. URL Blacklists oder bayesische Filter damit zu füttern. Um nicht Falsch-Positive anderen Filtern weiter zu vererben und somit die Qualität der eigenen schwarzen Listen zu vermindern, sollte man sich dabei nicht nur auf eine Blacklist verlassen, sondern diese Technik erst anwenden, wenn sich die IP-Adresse des absendenden Hosts auf mehreren Listen wiederfindet. Füttert man bayesische Filter damit, so haben falsch gelernte E-Mails nur geringe Auswirkungen auf die Erkennungsraten, sofern diese einen gewissen Schwellwert nicht überschreiten.

5.7. Ausgehende E-Mails

In den vorhergehenden Abschnitten wurde eingehend auf die verschiedenen Filtermethoden für eingehende Nachrichten eingegangen. Die Betreiber von Mail-Servern stehen aber zunehmend in der Pflicht, etwas gegen das Versenden von Spam über das eigene Netzwerk zu unternehmen. Zumindest über eine Benutzer-Authentifizierung sollte jeder ausgehende Mail-Server verfügen.
Spam, der über die Mail-Server von Internet Service Providern (ISP) versendet wird, verursacht folgende Probleme:

1. Jede Nachricht kostet den Providern Geld durch den erhöhten Bandbreiten- und Resourcenverbrauch.
2. Es besteht ein erhöhtes Risiko, dass die Mail-Server des Providers auf diversen schwarzen Listen landen. Dies hat meist verheerende Folgen für das Unternehmen, da ihre Server de facto von der Außenwelt abgeschnitten werden.
3. Durch die durch Spam erzeugten Fehlermeldungen wird der ISP selbst zum Spam-Versender.
4. Free-Mail-Provider haben zusätzlich das Problem, dass ihr werbefinanziertes Konzept untergraben wird und die Kosten für die Erkennung und Schließung von Spam-Accounts immer größer werden.

5.7.1. Probleme für „Gratis“ E-Mail-Provider

Große Free-Mail-Provider, bei denen binnen Minuten anonym ein kostenloser Zugang erstellt werden kann, sind ein beliebtes Versandinstrument für Spammer. Um Missbrauch zu verhindern, wurden Methoden entwickelt, die eine automatische Erstellung von solchen Accounts erschweren. Beispielsweise benutzen Yahoo und Hotmail sogenannte „reverse Turing“ Tests, die sicherstellen sollen, dass der Zugang durch einen Menschen und nicht durch ein automatisiertes Computerprogramm erstellt worden ist. Einige Provider wie z. B. GMX gehen noch einen Schritt weiter und lassen keine anonyme Accounterstellungen mehr zu [61].
Eine weitere Möglichkeit den Spamversand zu erschweren bestünde darin, in regelmäßigen Abständen, z. B. alle 500 Nachrichten, die Identität des Inhabers zu verifizieren. Einige ISP gehen auch dazu über, die Anzahl der Nachrichten pro Stunde oder Tag zu limitieren. Drastischer die Maßnahmen anderer Internet-Provider, die keine Verbindungen über den Port 25 mehr zulassen und ihren Kunden einen anderen SMTP-Port vorschreiben (Tele2 benutzt z. B. Port 587 [5]). Solche Maßnahmen stärken aber keineswegs das Verbrauchervertrauen und stellen keine geeignete Lösung des Spam-Problems dar [86].

5.7.2. Technische Maßnahmen

Viele der Verfahren, die auf eingehende Post angewandt werden, kann man ebenfalls auf ausgehende Post anwenden. IP basierende Blacklist sind aber gänzlich ungeeignet, da der Spamversand nicht über die Client-Rechner sondern über die eigenen Mail-Server erfolgt. Des weiteren hat auch Greylisting keinerlei Wirkung: Gewöhnliche Benutzer verfügen ja nicht über einen Mail-Server, sondern versenden die elektronischen Nachrichten in der Regel mit Hilfe eines SMTP-fähigen Mailprogramms.
Inhaltsbasierte Filter wie z. B. eine URL Blacklist oder heuristische Verfahren eignen sich aber sehr gut zur Spamerkennung.
Diese Maßnahmen sind aber in der Regel eher aufwendig zu implementieren. Eine einfachere und kostengünstigere Methode ausgehenden Spam zu erkennen, besteht in der Analyse der Log-Dateien. Wie bereits festgestellt wurde, ist eines

[5] http://www.tele2.at

der Merkmale von Spam die Aussendung in Massen. Da viele der dazu verwendeten Adressen entweder nicht mehr existieren oder falsch sind, hat das Versenden der Nachrichten eine Unzahl von Fehlermeldungen zur Folge. Aber auch Mailinglisten mit vielen fehlerhaften Adressen können ein ähnliches Verhalten an den Tag legen, weshalb man gegebenenfalls die Fehlermeldungen noch genauer unter die Lupe nehmen muss.
Des weiteren benutzen Spammer häufig wechselnde Absender und HELO-Werte, anhand derer man auf Spamversand schließen kann [28].
Grundsätzlich unternehmen ISPs noch zu wenig gegen Spammer, die ihr Netzwerk missbrauchen. Würden die Techniken, die für den Posteingang schon sehr gut arbeiten, auch auf den Postausgang übertragen, so ließe sich ein Großteil des Übels bereits vor der Aussendung verhindern.

6. Entwurf des Gateways

6.1. Ziele

Das Ziel dieser Implementierung ist eine kostengünstige und einfach zu installierende bzw. konfigurierende Lösung bereitzustellen. Um dies zu bewerkstelligen, soll durchwegs frei verfügbare Software zum Einsatz kommen, ohne dabei auf gute Erkennungsraten verzichten zu müssen.
Die wesentlichen Zielstellungen sind: Offenheit, Integration, Erweiterbarkeit, Verfügbarkeit und Skalierbarkeit

Offenheit Die zu entwickelnde Filterlösung soll mit allen gängigen Mail-Servern, die für Linux erhältlich sind, kompatibel sein und ohne größere Anpassungen mit diesen verwendet werden können.

Integration Da eine Filtermethode allein nicht für ausreichend gute Ergebnisse sorgt, sollen mehrere Verfahren miteinander kombiniert werden. Die Konfiguration der einzelnen Module soll über eine einheitliche Benutzerschnittstelle erfolgen.

Erweiterbarkeit Spamfilter haben sich in den letzten Jahren einem ständigen Wandel unterzogen, um auf geänderte Rahmenbedingungen reagieren zu können. Man kann davon ausgehen, dass dies auch in Zukunft notwendig sein wird. Aus diesem Grund müssen Möglichkeiten geschaffen werden, um neue Funktionalitäten einfach und rasch einbinden zu können.

Verfügbarkeit Die E-Mail-Kommunikation wird in vielen Unternehmen zunehmend als kritische Anwendung betrachtet. Möchte man eine hohe Verfügbarkeit erreichen, so sollten pro Domain zumindest zwei zuständige Mail-Server vorhanden sein.
Um Spammern nicht ein Einfallstor über die Mail-Server mit geringerer Priorität zu bieten, müssen diese in gleichem Maße vor Spam geschützt werden. Deshalb ist es notwendig, dass Benutzerkonten und individuelle Einstellung automatisch auf sekundäre Mail-Server übertragen werden können.

Skalierbarkeit Das System soll in kleinen Umgebungen genauso einsetzbar sein wie in großen Server-Farmen mit einigen Tausend Postfächern. Ebenfalls soll die Möglichkeit existieren, den Spam-Filter nicht nur als Gateway zu betreiben, sondern auch als eigenständigen Mail-Server. Idealerweise sollen diese beiden Verfahren pro Domain konfigurierbar sein, d. h. für die Domain X soll der Filter so konfigurierbar sein, dass eingehende, gefilterte E-Mails an einen weiteren Mail-Server weitergeleitet werden, für eine weitere Domain Y sollen die Einstellungen so vorgenommen werden können, dass sich die Mail-Postfächer in derselben Serverumgebung befinden. Das bedeutet auch, dass die Möglichkeit geschaffen werden muss, dass ausgehende Nachrichten über dieses System verschickt werden können.

6.2. Architektur

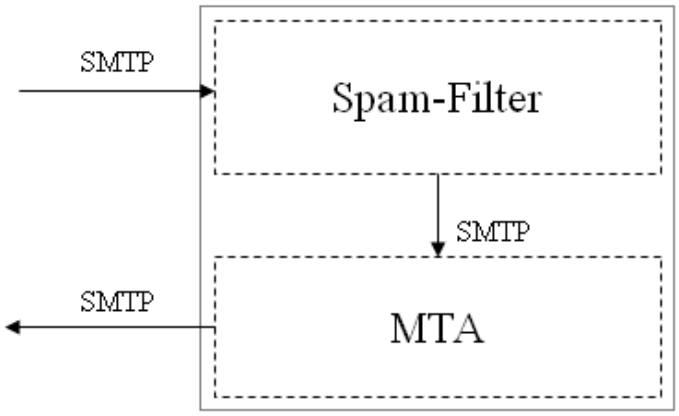

Abbildung 6.1.: Konzept bei reinem Gateway-Betrieb

Um der geforderten Offenheit Genüge zu tun, wird der Spam-Filter dem eigentlichen Mail-Server vorgeschaltet. Der Spam-Filter horcht dabei den SMTP Standard Port 25 ab, bei Verwendung von verschlüsselten Verbindungen noch zusätzlich den SSL-Port 465. Der Mail-Server muss lediglich dahingehend umkonfiguriert werden, dass er einen anderen Port für eingehende Verbindungen verwendet z. B. den Port 2525. Passiert nun eine E-Mail den Spam-Filter, so wird diese mittels SMTP-Verbindung an den „Message Transfer Agent“ weitergeleitet. Der MTA muss sich dabei nicht zwangsweise auf demselben Rechner befinden.
Die gewählte Architektur hat zum einen den Vorteil, dass der Filter getrennt vom Mail-Server betrieben und gewartet werden kann, zum anderen bleibt die Lösung gut skalierbar.
Wird die Filter-Lösung als reines Gateway betrieben, so werden die Nachrichten

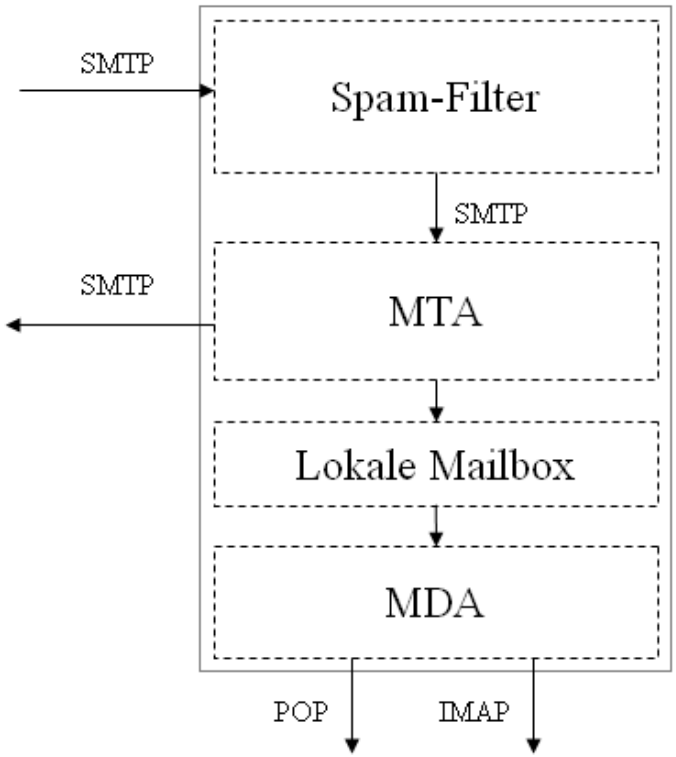

Abbildung 6.2.: Konzept bei Betrieb mit MDA

vom MTA dem endgültigen Mail-Server zugestellt. Sollen auch POP oder IMAP Postfächer verfügbar sein, so ist es zusätzlich notwendig, dass ein „Message Delivery Agent“ betrieben wird. Der notwendige Speicherplatzbedarf ist dann natürlich ungleich höher.

7. Implementierung des Spam-Filter Gateways

7.1. Verwendete Software

Im Folgenden wollen wir einen genaueren Blick auf die eingesetzte Software werfen. Wie in den geforderten Eigenschaften festgelegt, ist sämtliche verwendete Software als Opensource verfügbar.

7.1.1. QMail - Mail Server

Qmail ist ein in der Progammiersprache C geschriebener SMTP Message Transfer Agent, der von D. J. Bernstein entwickelt wurde und bereits 1997 in einer ersten Version vorlag. Die zur Zeit aktuelle Version 1.03 wurde bereits 1998 fertig gestellt [4]. Das auf Sicherheit und Zuverlässigkeit aufgebaute Konzept machte bis heute keine neue Version notwendig. Im Gegensatz zum monolithischen Konzept von Sendmail teilte Bernstein den Mail-Server in kleine eigenständige Programme auf, die großteils mit eingeschränkten Benutzerrechten laufen. So hat eine eventuelle Kompromittierung einer Komponente keinerlei Auswirkungen auf die restlichen Module des Mail-Servers.
Obwohl das Aufteilen in kleine Programme häufig zu Lasten der Geschwindigkeit geht, schaffte es Bernstein einen schnellen, sicheren und ressourcenschonenden Mail-Server zu implementieren. Umfasste die aktuelle Version von Sendmail über 70.000 Zeilen an Source-Code, so findet QMail mit gerade mal etwas mehr als 16.600 Zeilen ein Auslangen und das, obwohl QMail komplett auf die Standard Unix Bibliotheken verzichtet, da diese in den Augen von Bernstein eine erhöhte Gefahr für Pufferüberläufe darstellen [70].

Abbildung 7.1 veranschaulicht die Funktionsweise von QMail. Die fett umrandeten Programme sind Dämon-Prozesse, alle anderen Programme werden bei Bedarf aufgerufen. In Klammern ist jeweils der System-Benutzer angeführt, unter dem die einzelnen Programme laufen. Bis heute sind keine schwerwiegenden

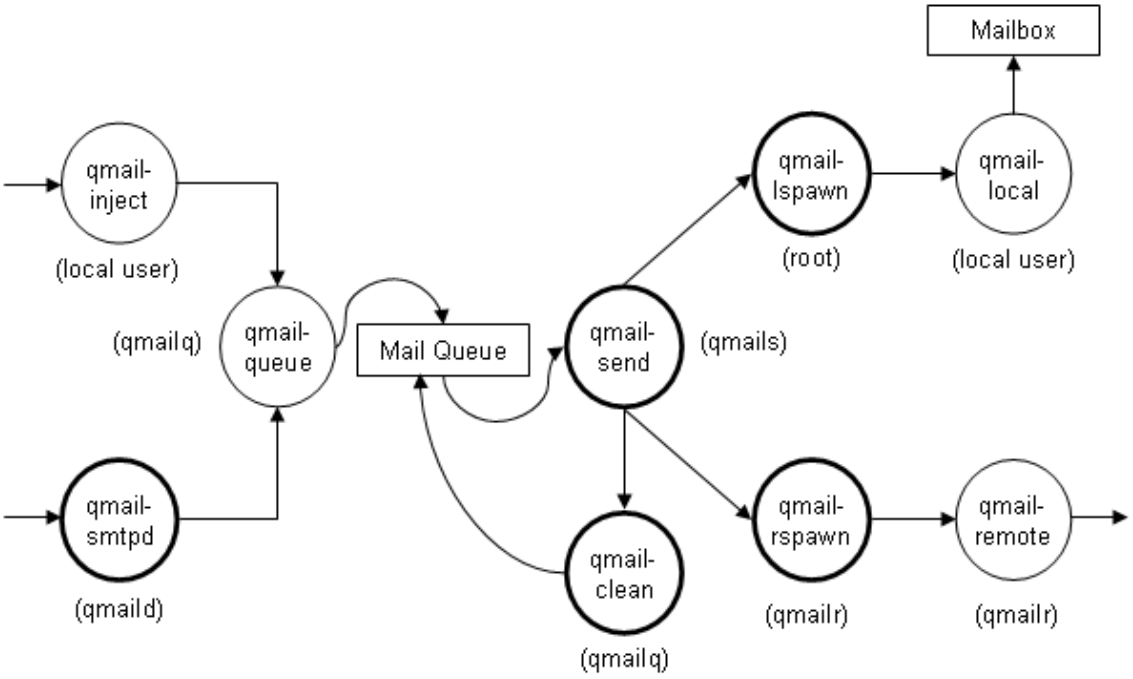

Abbildung 7.1.: Aufbau von QMail

Fehler aufgetaucht, die eine neue Version rechtfertigen würden. Dan Bernstein ging sogar soweit, dass er dem ersten, der eine Sicherheitslücke in Qmail findet, eine Prämie von 500 $ ausbezahlt. Bis heute wurden keine sicherheitsrelevanten Fehler entdeckt, lediglich Georgi Guninski [48] merkte an, dass auf einem 64-Bit System die 32-Bit Implementierung von QMail unter Umständen die Ausführung von schadhaftem Code ermöglichen würde (in der Annahme, dass qmail-smtpd unbeschränkt Speicher zu Verfügung steht und eine Menge von virtuellem Speicher vorhanden ist). Dan Bernstein erkannte diese Lücke nicht als „echte Lücke“ an, da eine solche Konstellation in der Praxis nicht vorkommen würde [7]. Zusammenfassend lässt sich sagen, QMail ist schnell, sicher und ressourcenschonend und eignet sich hervorragend für unsere Implementierung.
Da die aktuelle Version 1.03 bereits 1998 fertiggestellt wurde und zu dieser Zeit noch kaum Spam kursierte, enthält QMail keinerlei Mechanismen zur Spam-Abwehr. Vor allem die Möglichkeit, Empfänger während der SMTP-Sitzung ab-

zuweisen, macht QMail in der Standardversion sehr eingeschränkt empfehlenswert. Unter qmail.org existieren aber eine ganze Reihe von Patches, durch die zusätzliche Funktionen bereitgestellt werden.
Bei der Implementierung des Anti-Spam Gateways kam netqmail[1] in der Standardausführung zum Einsatz, das die QMail Version 1.03 samt einigen Fehlerbereinigungen enthält. Alle zusätzlichen Funktionen und Filter wurden im vorgeschalteten SMTP-Dämon integriert. So bleibt der eigentliche Mail-Server von zukünftigen Funktionen unberührt, lediglich die Implementierung von BATV machte einen kleinen QMail Patch notwendig.

DNS-Patch

DNS Antwort-Pakete waren ursprünglich auf 512 Bytes beschränkt. Einige größere Provider verfügen mittlerweile aber über eine ganze Reihe an Mail-Servern, sodass die DNS-Antwort unter Umständen länger als 512 Bytes sein kann. Während des Testlaufes kam es konkret mit E-Mails an eBay zu diesen Problemen und QMail beendete die Zustellung mit „CNAME lookup failed temporarily“. Mittlerweile existieren einige Patches, die diese Problem beheben [32].

7.1.2. vpopmail

QMail besitzt zwar von Haus aus die Möglichkeit, virtuelle Domains zu verwalten, jedoch ist es sehr umständlich und mühsam diese Einstellung händisch in Textfiles vorzunehmen. Das Programmpaket vpopmail [55] bietet die Möglichkeit, sämtliche Userinformationen in einer Datenbank, in unserem Fall MySQL, abzulegen. Durch eine Reihe nützlicher Befehle lassen sich sämtliche Operationen via Command-Line vornehmen.

7.1.3. vqadmin

vqadmin [56] ist eine webbasierte Schnittstelle für Administratoren, die es ermöglicht, vpopmail-Befehle auszuführen, die normalerweise root-Rechte benötigen, z. B. das Anlegen oder Löschen von virtuellen Domains. Über die eingebaute

[1] http://www.qmail.org/netqmail

Rechteverwaltung lassen sich bestimmte Menüpunkte individuell aktivieren.

7.1.4. qmailAdmin

qmailAdmin [54] ist ein frei erhältliches Software-Paket, das ein Web-Frontend für Enduser zur Verfügung stellt. So kann jeder User sein Kennwort ändern, eine Weiterleitung einrichten oder eine Abwesenheitsnachricht aktivieren. Dem Domain-Administrator stehen noch eine Reihe weiterer Befehle zur Verfügung, die es ermöglichen, neue E-Mail-Postfächer oder Aliase anzulegen. Sofern in vqadmin aktiviert, lassen sich damit auch Mailinglisten anlegen und verwalten.

7.1.5. qpsmtpd

Das Herzstück unseres Spam-Filters stellt qpsmtpd [8] dar. Dieser in Perl geschriebene SMTP-Dämon bietet die notwendige Flexibilität, um neue Spam-Filter Module einfach und rasch einzubinden. qpsmtpd wurde ursprünglich geschrieben, um qmail-smtpd zu ersetzen, mittlerweile kann dieses Programm aber mit fast jedem Mail-Server verwendet werden. Wir wollen nun einen genaueren Blick auf den Aufbau von qpsmtpd werfen.
Besonders interessant für unsere Implementierung wird qpsmtpd durch sehr flexible Plugin-Architektur. Die Kern-Implementierung enthält lediglich die Implementierung des SMTP Protokolls. Ohne das Zuschalten von Plugins würden keinerlei Filteraktivitäten durchgeführt, d. h. sämtliche zusätzliche Funktionalität ist als Plugin realisiert und kann mit dem Einfügen oder Entfernen einer Zeile in der entsprechenden Konfigurationsdatei bequem ein- oder ausgeschaltet werden. Durch diesen modularen Aufbau bleibt die Filterlösung auch nach der Integration vieler Plugins noch sehr übersichtlich.
Plugins können sich zu verschieden Zeitpunkten des SMTP-Verkehrs einhaken. Hierfür reicht es, wenn man in der Plugin-Datei eine Funktion mit dem entsprechenden Namen erstellt. Man spricht hier von sogenannten „hooks“. Je nach dem, welche Werte die Funktionen zurückliefern, wird mit der weiteren Verarbeitung der Nachricht fortgefahren oder eine Fehlermeldung an den E-Mail-Client zurückgeliefert.

Nachfolgend werden die wichtigsten „hooks“ kurz erklärt.

`hook_connect`: Wird aufgerufen, sobald sich ein Client zum Server verbindet, und zwar noch bevor eine event. Begrüßung gesendet wird.

`hook_helo / hook_ehlo`: Wird aufgerufen, sobald der Client den „HELO“ oder „EHLO“ Befehlt absetzt. Hier kann z. B. die Überprüfung des „HELO“ Wertes erfolgen.

`hook_mail`: Wird aufgerufen, sobald der Client das SMTP Kommando „Mail From“ sendet. Dieser „hook“ eignet sich beispielsweise, um bestimmte Absender schon während der SMTP-Sitzung abzuweisen.

`hook_rcpt`: Diese Plugin wird aufgerufen, wenn vom Client „rcpt to“ aufgerufen wird. Da das SMTP-Protokoll mehrere Empfänger pro Sitzung zulässt, wird dieses Plugin unter Umständen öfters aufgerufen.

`hook_data_post`: Wird aufgerufen, sobald der Inhalt der Nachricht gesendet wurde. Hier lässt sich z. B. eine Virensoftware einsetzen, die die Nachricht auf Schadsoftware überprüft.

`hook_queue`: hook_queue wird normalerweise nach allen anderen „hooks“ aufgerufen. Hier wird die Nachricht in der Regel im entsprechenden Mailbox-Format abgespeichert. In unserer Umsetzung wird die Nachricht an einen weiteren Mail-Server weitergeleitet. Dieser nimmt dann die endgültige Speicherung der E-Mail im Benutzerpostfach vor.

Des weiteren existieren noch weitere „hooks“ die verschiedene Möglichkeiten der Authentifizierung zur Verfügung stellen. Für eine detaillierte Beschreibung sei das qpsmtpd-Wiki [9] empfohlen.
Obwohl nicht in C geschrieben, eignet sich qpsmtpd durchaus auch für größere Installationen. Beispielsweise verarbeitet der qpsmtpd-Dämon für apache.org rund zwei Millionen Nachrichten pro Tag, was ca. dem Mailaufkommen eines kleinen Providers entspricht.
In dieser Implementierung wurde versucht, sämtliche Daten in einer MySQL-

Datenbank abzulegen, einerseits aus Performance-Gründen, andererseits um für Provider oder größere Umgebungen eine relativ einfache Möglichkeit der Echtzeit-Replikation zu schaffen.

7.1.6. ClamAV

Als Viren-Scanner kommt in dieser Implementierung lediglich ClamAV [2] zum Einsatz. Natürlich wäre es zu empfehlen, zusätzliche Viren-Scanner zu verwenden, jedoch ist ClamAV zur Zeit die einzige brauchbare Opensource Viren-Software. Auf http://www.av-test.org werden immer wieder die weitestverbreiteten Antivirus-Produkte getestet, wobei ClamAV meist nicht mit guten Erkennungsraten glänzte. Man sollte ClamAV deshalb lediglich als Ergänzung bei der Viruserkenung einsetzen. Für qpsmtpd existieren bereits Plugins für weitere Viren-Programme, jedoch sind diese meist nicht mehr kostenlos erhältlich.

7.1.7. DSPAM

DSPAM [6] ist ein inhaltsbasierender selbstlernender Spam Filter, der auf dem Bayes Theorem basiert. Laut Herstellerangaben liegen die Erkennungsraten zwischen 99,5 % und 99,9 %, d. h. eine Falschklassifizierung alle 200 - 2000 Nachrichten. Ein Hauptaugenmerk wurde dabei auf die Skalierung und Geschwindigkeit gelegt. So umfasst die größte bekannte Implementierung von DSPAM ca. 350.000 Postfächer. Durch die einfache Installation macht die Anwendung aber auch bei viel kleineren Umgebungen Sinn.
Im Gegensatz zu SpamAssassin, das die Bewertung auf Grund hunderter verschiedener Regeln vornimmt, ist DSPAM ein reiner Bayes Filter. Der Wartungsaufwand minimiert sich dadurch erheblich, da das manuelle Adaptieren der statischen Regeln entfällt. Diese statischen Regeln können - zumindest was SpamAssassin betrifft - nicht individuell für jeden Benutzer angepasst werden, somit erschien SpamAssassin für diese Implementierung als nicht geeignet.
In der Praxis hat sich auch die mangelnde Performance und der erhöhte Resourcenverbrauch von SpamAssassin als großer Nachteil erwiesen. Während DSPAM in der Programmsprache C geschrieben wurde, beruht SpamAssassin auf der Interpretersprache Perl. Das mag sich für kleinere Installationen nicht besonders

negativ auswirken, bei einem Mailaufkommen von mehreren Tausend E-Mails pro Stunde fallen die längere Ausführungszeit und der zusätzliche Speicherbedarf aber sehr wohl ins Gewicht.

7.1.8. Courier IMAP

Möchte man das Gateway auch als Server nutzen, so benötigt man natürlich einen Mail-Server, der die gängigen Protokolle POP und IMAP beherrscht, um den Inhalt eines Postfaches einem E-Mail-Client zur Verfügung zu stellen. Auf dem Markt sind eine Reihe POP fähiger Server für Linux erhältlich. Ausgereifte IMAP-Implementierungen sind jedoch nach wie vor eher rar. Schlussendlich fiel die Wahl auf die ausgereifte Implementierung Courier IMAP [36]. Courier IMAP verwendet nicht das UNIX E-Mail-Format mbox, sondern setzt auf das wesentlich robustere und wartungsfreundlichere Maildir-Format. Diese Robustheit wird vor allem dadurch erreicht, dass jede Nachricht in einem eigenen File abgelegt wird. Dies hat nicht nur den Vorteil, dass sich Maildir-Postfächer praktisch nicht zerstören lassen, man vermeidet auch die Notwendigkeit eines Schreibschutzes (file locking), sobald ein E-Mail-Client darauf zugreift.
Das Maildir-Format wurde ursprünglich von D. J. Bernstein im Rahmen seiner QMail-Implementierung entwickelt und sorgt für ausreichende Performance. Bei großen Systemumgebungen ist vor allem darauf zu achten, dass die richtige Hardware zum Einsatz kommt. Beim Maildir Format wird nämlich der Status der Nachricht im Dateinamen abspeichert, sodass sehr viele Schreib- und Lesezyklen notwendig werden. Schnelle SCSI Festplatten sind praktisch ein Muss.
Zum Authentifizieren seiner Benutzer verwendet Courier IMAP ein eigenes Modul namens „Courier Authentication Library“. Für unsere Implementierung ist es ebenfalls sehr gut geeignet, da sich das Modul dahingehend konfigurieren lässt, dass die Benutzerinformationen aus einer MySQL Datenbank entnommen werden.

7.1.9. Squirrelmail

Die bisher angeführte Software bleibt in der Regel für den Endbenutzer verborgen und kommt mit ihm auch nicht direkt in Berührung.
Um auf das E-Mail-Postfach bequem zugreifen zu können, bedarf es einer einfachen Web-Schnittstelle, die intuitiv und ohne vorherige Einschulung verwendet werden kann. Das in PHP geschriebene Webmail-Frontend Squirrelmail [13] erfüllt diese Voraussetzungen. Durch die inzwischen sehr große Entwickler-Community existieren eine Reihe von Plugins, mit der sich die Funktionalität fast beliebig erweitern lässt.

7.2. Implementierte Filter

Die nachfolgend beschriebenen Filter-Methoden wurden allesamt als qpsmtpd-Plugin implementiert. Einige davon wurden lediglich den eigenen Anforderungen angepasst, andere wiederum wurden von Grund auf neu erstellt.

earlytalker Dieses Plugin überprüft, ob der sich verbindende E-Mail-Client nicht schon vorher beginnt zu senden, noch bevor vom Mail-Server eine 2xx Begrüßung gesendet wurde. Dies stellt eine Protokoll-Verletzung dar und wird mit einer temporären Abweisung geahndet.

count_unrecognized_commands Zählt die Anzahl der vom Mail-Server nicht verstandenen SMTP-Kommandos. Wird die Anzahl von vier überschritten, endet dies mit einem Verbindungsabbruch.

auth_vpopmail_sql Dieses Plugin authentifiziert Benutzer anhand der in der vpopmail-Datenbank gespeicherten Benutzerdaten. Dieses Modul ist nur notwendig, sofern ausgehende Nachrichten ebenfalls über denselben Server verschickt werden sollen.

check_valid_vpopuser QMail führt während der SMTP-Sitzung keinerlei Überprüfungen durch, ob die Empfängeradresse existiert oder nicht. Dieses Plugin übernimmt diese Aufgabe und überprüft die Existenz der Empfängeradresse in der vpopmail-Datenbank.

require_resolvable_fromhost Dieses Plugin überprüft den DNS-Eintrag des Absender Hosts. Existiert für den Domain-Teil der Absender-Adresse kein MX- oder A-Eintrag, so wird die Verbindung mit einer temporären Fehlermeldung beendet.

db_whitelist / db_blacklist Selten, aber doch, besteht die Notwendigkeit Absender manuell auf die Whitelist zu setzen. db_whitelist sorgt dafür, dass die entsprechende Variable gesetzt wird, damit nachfolgende Überprüfungen übersprungen werden. Empfänger von ausgehenden Nachrichten landen automatisch auf der Whitelist, wobei diese bei Nichtverwendung nach 60 Tagen wieder davon entfernt werden.
Befindet sich ein Absender auf der „Schwarzen Liste“, wird die Zustellung mit sofortiger Wirkung abgebrochen und der Absender mittels Fehlermeldung informiert.

geo_blacklist_whitelist Dieses Modul ermöglicht das Blocken von eingehenden Verbindungen, die aus bestimmten Ländern stammen. Andererseits kann dieses Plugin so konfiguriert werden, sodass nur Nachrichten aus erlaubten Ländern angenommen werden.

greylisting Implementiert den Greylisting Algorithmus. Da eine Zustellverzögerung von einer Stunde und länger in den meisten Fällen nicht hinnehmbar ist, werden Zustellversuche maximal fünf Minuten blockiert.

dnsbl Ermöglicht die Verwendung verschiedener IP-basierter „Schwarzen Listen“, die mittels einer DNS-Anfrage abgefragt werden.

sender_permitted_from Setzt das „Sender Policy Framework“ Verfahren um.

uribl Überprüft, ob in eingehenden Nachrichten URIs enthalten sind, die sich auf schwarzen Listen wiederfinden. Diese Listen werden analog zu IP-basierten Blacklists mittels DNS-Abfragen konsultiert.

clamdscan Dieses Plugin ermöglicht das Scannen von ein- und ausgehenden Nachrichten durch den ClamAV-Dämon. Infizierte E-Mails werden nicht in einen Quarantäne Ordner verschoben, sondern mit eine entsprechenden Fehlermeldung abgewiesen.

dspam Mit Hilfe dieses Plugins werden die entsprechenden DSpam-Routinen aufgerufen. Um eine zuverlässige Lerndatenbank zu erhalten, werden ausgehende Nachrichten automatisch dem Ham-Corpus zugeführt.

batv Für das BATV Verfahren wurde kein eigenes Plugin erstellt, sondern die notwendigen Schritte in verschiedene Module integriert.

7.3. User-Interface

Genauso wichtig wie das gute Funktionieren des Filters, ist ein umfangreiches Dokumentieren der Filteraktiväten. Des weiteren wurde durch ein zusätzliches Squirrelmail-Plugin die Möglichkeit geschaffen, die Einstellungen des Spam-Filters individuell anzupassen. Administratoren können zusätzlich die Log-Einträge der gesamten Domäne einsehen und die Präferenzen der einzelnen Benutzer verändern.

7.3.1. Logging

Jedem Benutzer mit aktiven Konto steht das Logging-Fenster zur Verfügung, in dem sämtliche Aktivitäten des Spam-Filters dokumentiert werden. Um auf den ersten Blick erkennen zu können, ob eine Nachricht abgewiesen worden ist oder

nicht, wird jede eintreffende SMTP-Verbindung farblich durch einen Punkt markiert:

- ROT: E-Mail wurde abgewiesen
- GELB: Temporäre Abweisung (bei Greylisting)
- GRÜN: E-Mail wurde erfolgreich zugestellt.

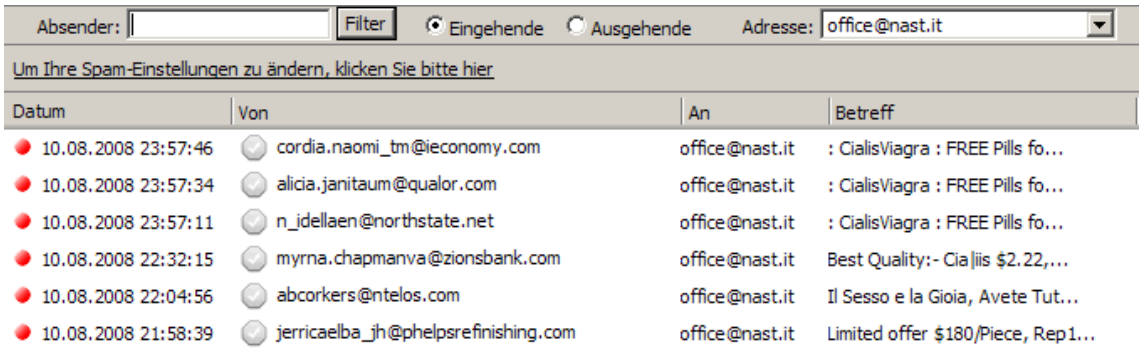

Abbildung 7.2.: Logging Übersicht

Weiters ersichtlich sind der Betreff, das Land des absendenden Mail-Servers und der eventuelle Grund der Abweisung.
Um etwaige Absender freischalten oder blockieren zu können, reicht das Klicken mit der rechten Maustaste auf die entsprechende Zeile. Mit Hilfe des sich öffnenden Kontextmenüs, kann die Absender-Adresse auf die White- oder Blacklist gesetzt werden.

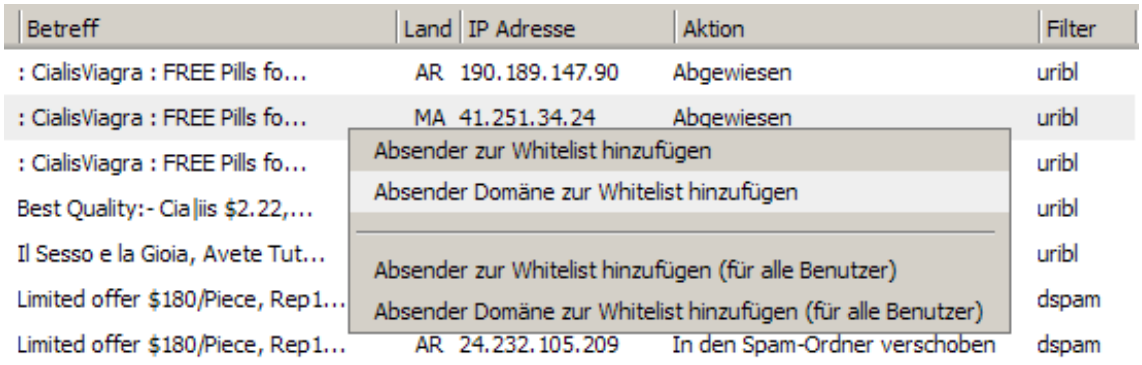

Abbildung 7.3.: Logging - Absender auf die Whitelist setzen

7.3.2. Spam-Filter Einstellungen

Beim Einstellungsfenster wurde darauf geachtet, dieses so einfach wie möglich zu gestalten. Es ist nicht notwendig, komplizierte Regeln zu erstellen. Das Aktivieren bzw. Deaktivieren geschieht durch das Setzen eines einfachen Häkchens. Dem Domain-Postmaster ist zudem erlaubt, die Einstellungen seiner Benutzer zu überschreiben. Für einige Filter kann festgelegt, wie mit den erkannten Nach-

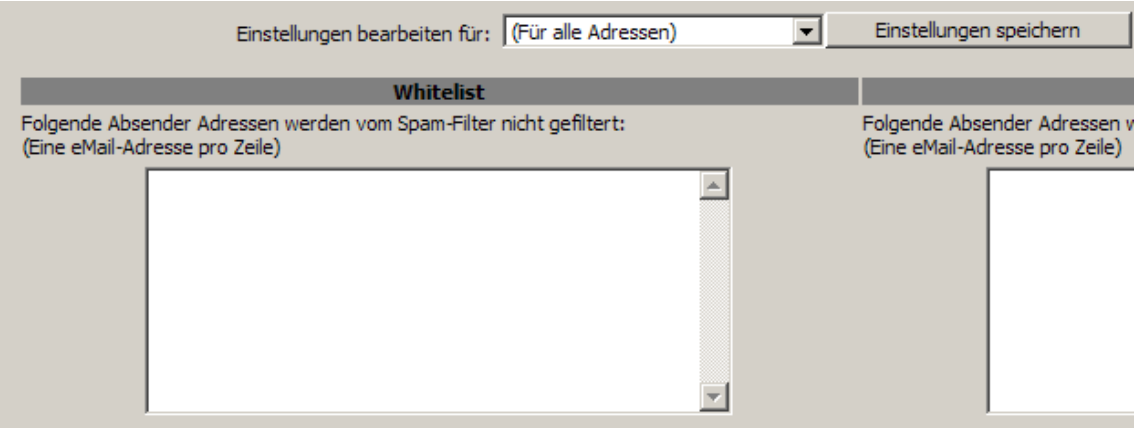

Abbildung 7.4.: Spam-Filter Einstellungen

richten umgegangen werden soll. Dabei hat der Benutzer folgende Wahlmöglichkeit:

- Abweisen: Der SMTP-Dialog wird beendet und die E-Mail abgewiesen
- Betreff mit dem „[Spam]“ erweitern: Die Nachricht wird normal zugestellt. Durch die Markierung im Betreff kann eine weitere Filterung im E-Mail-Client erfolgen.
- In den Spam-Ordner (Junk) verschieben: Wird diese Option gewählt, landen erkannte Spam-Nachrichten im eigens dafür vorgesehenen Ordner. Dieser wird bei Bedarf automatisch erstellt.
- An eine E-Mail-Adresse weiterleiten: Diese Option ist z. B. sinnvoll, wenn im Unternehmen ein Benutzer existiert, der die als Spam gekennzeichneten Nachrichten auf Fehler überprüft. Aus Sicherheitsgründen darf sich diese Weiterleitungsadresse nur in der eigenen Domain befinden.

Beim Punkt „DNS-Blacklists“ lassen sich zusätzlich zu den Filteroptionen die verschiedenen Abfrage-Listen einzeln aus- oder einschalten.

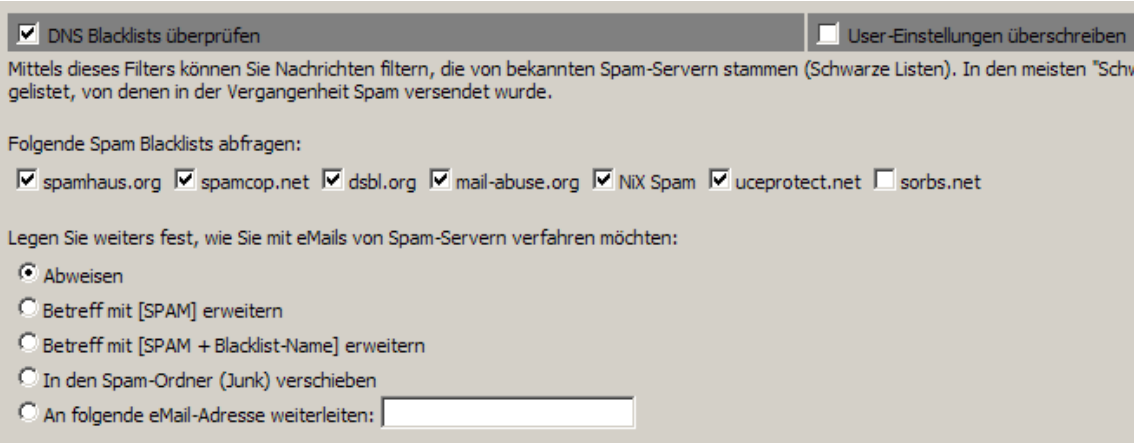

Abbildung 7.5.: IP-Blacklist Einstellungen

8. Auswertung der erzielten Ergebnisse

8.1. Verwendetes Testsystem

Als Testrechner kam ein Server mit einem AMD Opteron Prozessor mit 2 Gigahertz, 1024 Megabyte RAM und 2 x 250 Gigabyte Festplatten im Raid1 Verbund zum Einsatz. Der geschätzte Neupreis des eingesetzten Testsystems belief sich zum Testzeitpunkt auf ca. 800 Euro. Auf dem Rechner war die Linux-Distribution Debian in der Version 4 (etch) installiert. Des weiteren standen folgende Softwarepakte zur Verfügung:

- apache 2.2.3
- php 5.2.0
- mysql 5.0.32

Auf dem Testrechner waren über 700 E-Mail-Posfächer aktiv, wobei täglich ca. 26.000 SMTP-Anfragen verarbeitet wurden. Über den Zeitraum von einem Monat betrachtet, waren es ca. 800.000.

8.2. Einschränkungen

Durch die Auswertung der Log-Files wurde versucht, ein bestmögliches Gesamtbild der Filterleistung zu liefern. Aus Privacy-Gründen konnten aber keine in Verwendung befindlichen E-Mail-Adressen überprüft werden, stattdessen beschränkt

sich die Auswertung auf stillgelegte Postfächer. Deshalb finden in den Statistiken lediglich die Erkennungsraten ihren Niederschlag. Die Anzahl von Falsch-Positiven konnte aus den genannten Gründen nicht ermittelt werden, was ohne die Einsicht in die jeweiligen Postfächer auch nicht möglich war.
Um das Testergebnis nicht zu verzerren, wurden mehrere Verbindungen von demselben Mail-Server nur einfach gezählt.

8.3. Ergebnisse

8.3.1. Gesamtergebnis

Die folgende Auswertung präsentiert die Erkennungsraten des Filter mit folgenden aktiven Modulen (in dieser Reihenfolge):

- Greylisting
- Bounce Address Tag Validation
- DNS Blacklists
- URI Blacklists
- Sender Policy Framework

Da selbstlernende Filter auch mit legitimen E-Mails gefüttert werden müssen, konnte dieses Modul aus Datenschutzgründen keinem aussagekräftigen Test unterzogen werden und wurde beim Gesamtergebnis nicht berücksichtigt. Wie in

Tabelle 8.1.: Gesamtergebnis

	Anzahl	**%**
Greylisting	39865	99,66
BATV	59	0,15
DNS Blacklists	61	0,15
URI Blacklists	6	0,01
Sender Policy Framework	0	0,00
Nicht erkannt	12	0,03
Gesamt	40003	100,00

Tabelle 8.1 ersichtlich, wird der Großteil der unerwünschten Nachrichten bereits von Greylisting geblockt, sodass die anderen Filtermethoden sehr selten zum Zug kommen. Mit einer Erkennungsrate von 99,97 % braucht unser Filter auch den Vergleich mit kommerziellen Anwendungen nicht zu scheuen.

8.3.2. Greylisting

Greylisting wurde mit folgenden Einstellungen getestet:

- Black Timeout von 5 Minuten
- Grey Timeout von 24 Stunden

Das bedeutet, dass ein erfolgreicher Zustellversuch von unbekannten Tripeln (IP-Adresse / Absender / Empfänger) frühestens nach 5 Minuten und spätestens 24 Stunden nach dem Erstkontakt erfolgen kann. Um die negativen Auswirkungen bei Zustellungen von Server-Farmen weitmöglichst zu minimieren, wurde nicht die vollständige IP-Adresse des absendenden Mail-Servers verglichen sondern lediglich die ersten drei Byte. Obwohl der praktische Test des Greylisting-

Tabelle 8.2.: Greylisting im Test

	Anzahl	**%**
Spam durch Greylisting abgewiesen	20630	98,51
nicht abgewiesen	312	1,49
Gesamt	20942	100,00

Verfahrens mit sehr aufgeweichten Einstellungen vorgenommen wurde, brachte die Auswertung doch eine kleine Überraschung zu Tage. Nach wie vor ist Greylisting ein einfaches aber sehr effizientes Verfahren, um den Großteil der unerwünschten Nachrichten vom eigenen Mail-Server fernzuhalten. Während des Testlaufs ließ sich aber beobachten, dass einige Spam-Kampagnen mehrere Zustellversuche unternahmen. Egal wie hoch man den Zeitraum für die temporäre Abweisung auch schraubte, alle 10 Minuten erfolgte konsequent ein Zustellversuch. Nach ein paar Tagen war dieses Phänomen aber genauso schnell verschwunden, wie es aufgetaucht war.
Offensichtlich dürfte es für Spammer nach wie vor sehr ressourcenaufwendig

sein, für großangelegte Spam-Kampagnen ein geeignetes Warteschlangenmanagement zu implementieren.
Im Testbetrieb verursachte Greylisting keine nennenswerten Probleme. Traten dennoch Probleme auf, so lag der Grund meist in nicht RFC-konformen Mail-Servern. In der Regel handelte es sich hierbei um kleine firmeneigene Umgebungen, die falsch konfiguriert waren. Aber auch große Portale, wie z. B. Facebook.com, scheinen sich nicht immer an Standards zu halten. In diesen Fällen half nur ein händisches Freischalten mittels Whitelist.
Greylisting ist aber nicht allen Fällen eine praktikable Lösung, zahlt man ja durch die verzögerte Zustellung einen relativ hohen Preis.

8.3.3. DNS Blacklist

Im Test kamen folgende DNS-Blacklists zum Einsatz:

- rbl.mail-abuse.org
- zen.spamhaus.org
- bl.spamcop.net
- list.dsbl.org
- ix.dnsbl.manitu.net
- dnsbl.sorbs.net
- dnsbl-2.uceprotect.net

Tabelle 8.3.: DNS Blacklists im Test

	Anzahl	**%**
Spam erkannt	20630	98,51
nicht erkannt	312	1,49
Gesamt	20942	100,00

Das Ergebnis macht deutlich, warum DNS-Blacklists so beliebt sind. Man erreicht mit einfachsten Mitteln sehr gute Erkennungsraten. Einige Blacklist-Betreiber leiden aber nach wie vor unter einem schlechten Ruf, zum einen aufgrund zweifelhafter Aufnahmekriterien, zum anderen ist eine sofortige Entfernung fälsch-

lich gelisteter IP-Adressen teilweise nur gegen Bezahlung möglich. Der Blacklist-Betreiber uceprotect.net beispielsweise verlangt 50 Euro für das umgehende Löschen aus ihren Listen. Bei Nichtbezahlung wir der Eintrag erst nach sieben Tagen entfernt, was in vielen Fällen einen unerträglich langen Zeitraum darstellt.
Grundsätzlich lässt sich aber feststellen, dass IP-basierte Blacklists in den letzten Monaten einen Qualitätssprung erfahren haben. Neue Spam-Quellen werden sehr rasch gelistet, wobei die Anzahl der False-Positives ständig zurückgeht [57].

8.3.4. URI Blacklist

Im Test kamen folgende URI-Blacklists zum Einsatz:

- multi.surbl.org
- multi.uribl.com

Die Erkennungsraten sahen dabei folgendermaßen aus: Berücksichtigt man, dass

Tabelle 8.4.: URI Blacklists im Test

	Anzahl	**%**
Spam erkannt	5299	87,29
nicht erkannt	771	12,71
Gesamt	6070	100,00

nicht alle Spam-Nachrichten URIs enthalten, so sind die Erkennungsraten doch recht beachtlich.

8.3.5. Sender Policy Framework

Das „Sender Policy Framework“ wurde, wie bereits erwähnt, geschaffen, um Adressfälschungen auf SMTP-Ebene zu erschweren. Absenderfälschungen wurden nur mit einer Abweisung geahndet, wenn dies im TXT-Feld des DNS-Eintrags ausdrücklich festgelegt wurde. Das Auktionshaus eBay z. B. verwendet nach wie vor die „~all“ Direktive, die eigentlich nur für Testzwecke gedacht ist und keine Abweisung der Nachrichten von nicht autorisierten Mail-Servern zur Folge hat.

Tabelle 8.5.: Das Sender Policy Framework im Test

	Anzahl	**%**
Absenderfälschung festgestellt	112	2,00
Keine Absenderfälschung festgestellt	5489	98,00
Gesamt	5601	100,00

Das Ergebnis des Tests ist sehr ernüchternd. Gerade einmal 2 % der Spam-Nachrichten wurden auf Grund der Absenderfälschung aussortiert. Das ist auch nicht weiter verwunderlich, haben große Provider wie T-Online, Web.de oder Yahoo bis heute keinen SPF-Record im DNS veröffentlicht. Aber auch Banken oder Versicherungen gehen sehr zögerlich mit der Einführung von SPF um.

8.3.6. BATV

„Bounce Adress Tag Validation“ stellte sich im Test als wirksames Mittel heraus, um die Folgen von Adress-Missbrauch zu mindern. Voraussetzung für die Verwendung von BATV ist natürlich, dass eingehende und ausgehende Nachrichten vom selben Server-Verbund verarbeitet werden. Im praktischen Test kam es dabei zu folgenden Problemen:

- Einige Server sendeten Fehlerberichte bzw. Unzustellbarkeitsnachrichten nicht mit leerem Absender (Mail From) an den Server zurück. Dies ist allerdings ein klarer Verstoß der allgemein gültigen Richtlinien.
- Abwesenheitsnachrichten wurden manchmal an die Absender-Adresse im SMTP-Envelope gesandt und nicht an die Absender-Adresse des Mail-Headers. Diese Nachrichten wurden vom BATV-Filter ebenfalls abgefangen.

Zusammenfassend lässt sich feststellen, dass Probleme nur dann auftraten, sobald man es mit nicht RFC-konformen Mail-Servern zu tun hatte.

9. Fazit

Ungeachtet aller Sicherheitsmaßnahmen und des langsamen Abnehmens der Open Relay Mailserver hat sich das Spamaufkommen in den letzten Jahren mehr als verdoppelt. Es scheint also für Spammer trotz erschwerter Voraussetzungen (und teilweise empfindlicher Geld- und Haftstrafen) immer noch rentabel zu sein, Millionen von Werbebotschaften in die Welt zu setzen und lassen so die Postfächer der Internetnutzer überquellen. Dabei bedienen sich Spammer zusehends fremder Rechner, die zuvor durch Schadsoftware übernommen wurden.
Ein verantwortungsbewusster Umgang mit der eigenen E-Mail-Adresse würde die Adressgewinnung dabei erheblich erschweren, zumal verschiedene Verfahren existieren, die das Ausforschen von auf Websites veröffentlichten E-Mail-Adressen nahezu unmöglich machen.
Bei der Auswahl von Blacklists, die nicht im eigenen Verantwortungsbereich liegen, muss mit besonderer Sorgfalt vorgegangen werden, eine genaue Studie der Aufnahmekriterien ist unumgänglich.
Selbstlernende Filter haben sich in der Praxis bewährt, jedoch erzielen diese nur ansprechende Ergebnisse, wenn der Lerndatenbank ausreichend Spam und Ham Nachrichten zugeführt wurden. Dabei besteht vor allem die Schwierigkeit, ein gewisses Gleichgewicht zwischen den beiden Arten von Nachrichten zu erhalten. Eine individuelle Lerndatenbank für jeden User würde das Ergebnis zwar verbessern, der Ressourcenverbrauch erhöht sich jedoch dramatisch.
Greylisting ist entgegen anders lautenden Meldungen nach wie vor ein effizientes Mittel, um einen Großteil unerwünschter Nachrichten abzuweisen, wenn gleich die Effizienz des Verfahrens langsam abzunehmen scheint. In Verbindung mit DNS-Blacklists lassen sich ohne großen Aufwand ansprechende Resultate erzielen.
Die verschiedenen Versuche, Absender-Adressen zu verifizieren, scheitern an der schwierigen Umsetzung und der Zerstrittenheit der einzelnen Initiativen. Zumal diese Methoden allesamt auf dem unsicheren DNS-Protokoll aufbauen, darf an

deren Erfolg gezweifelt werden.
Noch bessere Erkennungraten ließen sich erreichen, wenn die Ergebnisse der einzelnen Methoden dafür genutzt würden, um andere Module mit neuen Daten zu füttern. So könnten z. B. alle URIs aus Spam-Nachrichten - erkannt durch IP-Blacklists - extrahiert und einer eigenen URI-Datenbank zugeführt werden. Besonders Bayes basierte Filter könnten automatisch mit anderweitig erkannten Spam-Nachrichten gefüttert werden.
Grundsätzlich sind moderne Spam-Filter mittlerweile in der Lage weit über 95 % des Spams auszusortieren, bei einer false-positives-Rate im Promillebereich - es gibt also günstige, frei verfügbare Lösungen, um sich effektiv vor Spam zu schützen.
Das Aufrüsten der Mail-Provider gegen die Spamflut birgt aber auch einige Gefahren für die Zukunft. Um trotz verschärfter Anstrengungen vonseiten der E-Mail-Provider genügend Nachrichten zustellen zu können, müssen Spammer immer mehr Nachrichten versenden. Da die Spamfilterung erst auf der Empfängerseite erfolgen kann, müssen Provider ständig die Bandbreite und die Rechenleistung ihrer Server-Farmen erhöhen. Große Provider wie „1 & 1“ verzeichnen täglich bereits zwischen 100 Millionen und einer Milliarde Zustellversuche [51]. Kleinere Provider können mit dem technischen Ausbau oft nicht mithalten und laufen Gefahr, bei diesem Wettrüsten auf der Strecke zu bleiben.
Egal, ob man nun über einen Spam-Filter verfügt oder nicht, so scheint die Verlässigkeit des Medium E-Mail ständig abzunehmen. Verzichtet man auf einen Filter, so wird über kurz oder lang das Postfach mit Werbemüll überquellen. Setzt man auf technische Filtermaßnahmen, so bleibt bei jedem noch so effizienten Verfahren immer der üble Beigeschmack einer möglichen Falsch-Klassifizierung.
Die gänzliche Eindämmung der Spam-Flut ließe sich nur erreichen, wenn die seit eh und je existierende Design-Schwäche des SMTP-Protokolls geschlossen würde. Dieses historisch gewachsene Protokoll ist jedoch zu sehr verbreitet, sodass eine rasche Ablöse mehr als unwahrscheinlich erscheint.

A. Anhang

A.1. Installationsanleitung

Nachfolgend werden alle notwendigen Schritte erklärt, die zu einer erfolgreichen Installation des Spam-Filters notwendig sind. Je nach vorhandenen Linux-Kenntnissen lässt sich die gesamte Installation in wenigen Stunden durchführen.

A.1.1. Systemvoraussetzungen

Unser Spam-Filter-System kann auf jedem handelsüblichen System, auf dem Linux mit Kernel 2.6 läuft, installiert werden. Je nach geforderter Größenordnung können einzelne Module auf zusätzliche Rechner ausgelagert werden. Für kleinere Umgebungen (<100.000 E-Mails pro Tag) ist dies jedoch nicht notwendig. Wie bei jeder Datenbankanwendung sollte aber auf die richtige Auswahl der Speichermedien geachtet werden. Schnelle SCSI Platten im Raid-Verbund sind günstigeren IDE-Platten auf jeden Fall vorzuziehen. Arbeitsspeicher sollte mindestens 1GB vorhanden sein, je mehr, desto besser.
Weiters müssen vor dem Installationsstart folgende Softwarepakte installiert werden:

- MySQL
- Apache
- php5 mit MySQL Unterstützung
- Perl
- Berkeley DB

- ezmlm (wenn Mailing-Listen zur Verfügung stehen sollen)

A.1.2. QMail

Zur Anwendung kommt Netqmail [5], eine leicht gepachte Version der QMail Version 1.03. Unter http://www.lifewithqmail.org befindet sich eine recht ausführliche Installationsanleitung, an die man sich ohne Einschränkung halten kann. Einzig und allein das qmail-smtpd run-Script muss dahingehend abgeändert werden, dass eingehende SMTP-Verbindungen auf dem Port 2525 akzeptiert werden. Unser Run-Script sieht nun folgendermaßen aus:

```
#!/bin/sh

QMAILDUID=`id -u qmaild`
NOFILESGID=`id -g qmaild`
MAXSMTPD=`cat /var/qmail/control/concurrencyincoming`
LOCAL=`head -1 /var/qmail/control/me`

if [ -z "$QMAILDUID" -o -z "$NOFILESGID" -o \
 -z "$MAXSMTPD" -o -z "$LOCAL" ]; then
    echo QMAILDUID, NOFILESGID, MAXSMTPD, or LOCAL is unset in
    echo /var/qmail/supervise/qmail-smtpd/run
    exit 1
fi

if [ ! -f /var/qmail/control/rcpthosts ]; then
    echo "No /var/qmail/control/rcpthosts!"
    echo "Refusing to start SMTP listener \
    because it'll create an open relay"
    exit 1
fi

exec /usr/local/bin/softlimit -m 4000000 \
    /usr/local/bin/tcpserver -v -R -l "$LOCAL" \
    -x /etc/tcp.smtp.cdb -c "$MAXSMTPD" \
    -u "$QMAILDUID" -g "$NOFILESGID" 0 2525 /var/qmail/
```

Möchte man auch BATV implementieren, so ist an dieser Stelle ein guter Zeitpunkt, den QMail BATV-Patch anzuwenden (siehe Anhang).

A.1.3. vpopmail

Nachdem die Installation von QMail erfolgreich abgeschlossen wurde, kann als nächstes vpopmail Version 5.4.25 installiert werden. Für vpopmail ist ein eigener System-Benutzer und eine eigene System-Gruppe vorgesehen:

```
# groupadd -g 89 vchkpw
# useradd -g vchkpw -u 89 vpopmail
```

Damit vpopmail sich mit dem richtigen Benutzernamen und Passwort beim MySQL-Server anmeldet, ist es notwendig, dass wir im „etc“ Verzeichnis eine Datei namens vpopmail.mysql erstellen:

```
# echo "localhost|0|vpopmailuser|vpoppasswd|vpopmail"
                      > ~vpopmail/etc/vpopmail.mysql
# chown vpopmail.vchkpw ~vpopmail/etc/vpopmail.mysql
# chmod 640 ~vpopmail/etc/vpopmail.mysql
```

Bevor vpopmail installiert werden kann, muss noch eine entsprechende Datenbank für vpopmail erstellt werden:

```
mysql> CREATE DATABASE vpopmail;
mysql> GRANT select,insert,update,delete,create,drop ON vpopmail.*
       TO vpopmailuser@localhost IDENTIFIED BY 'vpoppasswd';
```

Nun kann man ins Source-Verzeichnis wechseln und das Konfigurationsscript mit folgenden Parametern aufrufen:

```
# ./configure --disable-roaming-users \
  --enable-logging=p \
  --disable-passwd \
  --enable-clear-passwd \
  --disable-domain-quotas \
  --enable-auth-module=mysql \
```

```
  --enable-auth-logging \
  --enable-sql-logging \
  --enable-valias \
  --disable-mysql-limits \
  --enable-qmail-ext=y
```

Kompilieren und Installieren lässt sich das Paket mit folgenden Befehlen:

```
# make
# make install-strip
```

Jetzt noch `~vpopmail/etc/vlimits.default` an die eigene Bedürfnisse anpassen, in `~vpopmail/etc/defaultdomain` die Domainnamen des Rechners eintragen und fertig ist die vpopmail-Installation.

A.1.4. vQadmin

Mit der Installation von vpopmail steht dem Systemadministrator eigentlich bereits ein Programmpaket zur Verfügung, das uns erlaubt, alle notwendigen Operationen bzgl. virtueller Domains durchzuführen. Jedoch können Befehle nur über Shell-Kommandos abgesetzt werden. Um die Administration angenehmer zu gestalten, wurde vQadmin entwickelt. Es lässt sich mit folgenden Schritten installieren:

```
# ./configure --enable-cgibindir=/usr/lib/cgi-bin/
# make
# make install-strip
```

Um vQadmin zum Laufen zu bringen, müssen wir im Apache Konfigurationsfile für das vQadmin-Verzeichnis eigene Rechte festlegen. Das sieht dann folgendermaßen aus:

```
<Directory "/usr/lib/cgi-bin/vqadmin">
    deny from all
    Options ExecCGI
    AllowOverride AuthConfig
    Order deny,allow
</Directory>
```

Da natürlich nur ausgewählte Personen Zugriff auf die Domain-Verwaltung haben dürfen, muss eine geeignete Authentifizierung durchgeführt werden. Am einfachsten geschieht dies durch die Verwendung einer Apache .htaccess Datei. In unserem Fall sieht diese Datei folgendermaßen aus:

```
AuthType Basic
AuthName vqadmin
AuthUserFile /etc/apache2/vqadmin.passwd
Require valid-user
Satisfy any
```

Nach Anlegen dieser Datei folgt als nächstes die Erstellung eines Benutzers und das Festlegen des dazugehörigen Passworts.

```
# chown apache .htaccess
# chmod 600 .htaccess
# htpasswd -bc /etc/apache2/vqadmin.passwd admin adminpass
```

Legt man in der Datei vqadmin.acl noch die Zugriffsrechte fest, so steht einem Apache Neustart nichts mehr im Wege.

A.1.5. qmailadmin

Die Installation des Webinterface zum Bearbeiten der E-Mail-Einstellungen durch den Endbenutzer kann wie folgt durchgeführt werden (Installation mit root-Rechten durchführen):

```
# ./configure \
  --enable-htmldir=/var/www/htdocs \
  --enable-cgibindir=/usr/lib/cgi-bin
# make
# make install-strip
```

Sofern das cgi-bin Verzeichnis richtig angegeben wurde, kann qmailadmin im Browser durch die Eingabe von

```
http://yourdomain/cgi-bin/qmailadmin
```

aufgerufen werden.

A.1.6. Courier authentication library

Dieses Modul ist nur notwendig, sofern der Rechner auch als Mail-Server mit POP bzw. IMAP Support agieren soll. Für den reinen Gateway-Betrieb kann dieser Abschnitt übersprungen werden.
Courier IMAP führt die Authentifizierung nicht selber durch, sondern hat diese Funktionalität in eine eigene Bibliothek ausgelagert. Courier IMAP lässt sich nicht installieren, bevor nicht die Installation dieses Pakets erfolgte. Die Courier authentication library [35] ist ebenfalls sehr flexibel und authentifiziert Benutzer aus passwd-Files bis hin zu MySQL- oder PostgreSQL Datenbanken. Da wir bereits vpopmail so konfiguriert haben, dass Benutzerinformationen in einer MySQL Datenbank abgelegt werden, müssen wir dem Authentifizierungsmodul ebenfalls den Zugang auf diese Daten ermöglichen. Ursprünglich enthielt die Bibliothek bereits ein vpopmail Modul. Dieses wurde aber in der neuesten Release entfernt, sodass wir auf das MySQL Modul zurückgreifen müssen.
Nach Download des Source-Codes (Version 0.61) können wir das Konfigurations-Script mit folgenden Parametern aufrufen:

```
# ./configure --without-authpam \
   --without-authshadow \
   --without-authpipe \
   --without-authpgsql \
   --without-authuserdb \
   --without-authpwd \
   --with-authmysql
# make
# make install
# make install-configure
```

Nach erfolgter Installation muss die MySQL Konfigurationsdatei dahingehend abgeändert werden, damit die Authentifizierungsinformationen aus der vpopmail-Datenbank entnommen werden. Standardmäßig befindet sich das Konfigurationsfile unter `/usr/local/etc/authlib/authmysqlrc`. Folgende Einstellungen müssen eingetragen werden:

```
MYSQL_SERVER            localhost
MYSQL_USERNAME          vpopmailuser
MYSQL_PASSWORD          vpoppasswd
```

```
MYSQL_PORT              3306
MYSQL_OPT               0
MYSQL_DATABASE          vpopmail

MYSQL_SELECT_CLAUSE     SELECT CONCAT(pw_name, '@', pw_domain), \
         pw_passwd,     \
         '',            \
         89,            \
         89,            \
         pw_dir,        \
         '',            \
         '',            \
         '',            \
         CONCAT("disableimap=",pw_gid&8>0,",disablepop3=",    \
                pw_gid&2>0,",disablewebmail=",pw_gid&4>0)    \
         FROM vpopmail   \
         WHERE pw_name = '$(local_part)' \
         AND pw_domain = '$(domain)'
```

Abschließend kann der Authentifizierungsdämon gestart werden:

```
/etc/init.d/courier-authlib start
```

A.1.7. Courier IMAP

Dieses Modul ist nur notwendig, sofern der Rechner auch als Mail-Server mit POP bzw. IMAP Support agieren soll. Für den reinen Gateway-Betrieb kann dieser Abschnitt übersprungen werden. (Wichtig: Das Entpacken des Tarballs und das Erstellen der ausführbaren Dateien darf nicht unter root erfolgen):

```
# ./configure
# make
# make check
# su root
# make install
# make install-configure
```

Die Konfigurationsdateien befinden sich unter `/usr/lib/courier-imap/etc` und müssen noch den eigenen Bedürfnissen angepasst werden. Vor allem die Variablen `IMAPDSTART` in `imapd` und `POP3DSTART` in `pop3` müssen auf „YES“ gesetzt werden, damit die entsprechenden Dienste mit dem Aufruf von

```
/etc/init.d/courier-imap start
```

gestartet werden.

A.1.8. dspam

Für unsere Implementierung kommt dspam Version 3.8.0 zur Anwendung. Nach Entpacken des Quellcodes kann unser selbstlernender Filter wie folgt installiert werden:

```
./configure --with-dspam-home=/home/dspam \
  --prefix=/home/dspam \
  --enable-domain-scale \
  --with-delivery-agent=/usr/local/bin/maildrop \
  --with-storage-driver=mysql_drv \
  --with-mysql-includes=/usr/include/mysql \
  --with-mysql-libraries=/usr/lib/mysql \
  --enable-virtual-users \
  --with-dspam-home-owner=vpopmail \
  --with-dspam-home-group=vchkpw \
  --with-dspam-owner=vpopmail \
  --with-dspam-group=vchkpw \
  --enable-preferences-extension
```

Jetzt müssen die Konfigurationsparameter an die eigene Systemumgebung angepasst werden. Wichtig hierbei ist, dass „owner“ und „group“ auf die entsprechenden vpopmail Werte gesetzt werden. Nach Aufruf von

```
make && make install
```

befindet sich im Subdirectory „etc“ des gewählten Home-Verzeichnises nun das zentrale Konfigurationsfile `dspam.conf`.
DSpam unterstützt mehrere Arten der lexikalischen Analyse, von simpler Word-Analyse bis hin zum komplexeren „Sparse Binary Polynomial Hashing“. „Chained Token“ ist dabei meistens die beste Wahl.
Die Berechnung der Spam-Wahrscheinlichkeit kann ebenfalls auf die persönlichen Bedürfnisse angepasst werden. Dabei können auch mehrere Algorithmen zur Anwendung kommen. In dieser Implementierung wird die Methode von Graham in Verbindung mit dem Burton-Verfahren eingesetzt.
Weiter im Konfigurationsfile befinden sich noch die Verbindungseinstellungen für MySQL. Diese müssen auf folgende Werte gesetzt werden:

```
MySQLServer     <path to mysql socket>/mysqld.sock
MySQLUser       vpopmailuser
MySQLPass       vpoppasswd
MySQLDb         vpopmail
MySQLCompress   true
```

Um das Lernen via E-Mail zu ermöglichen, muss auch die Option „signatureLocation“ auf „message“ gesetzt werden. Dies hat zur Folge, dass eintreffenden Nachrichten eine spezielle DSPAM-Signatur im E-Mail-Body angehängt wird. DSpam kann auch so konfiguriert werden, dass diese Signatur im Nachrichten-Header gespeichert wird, jedoch müssten die Nachrichten samt Header weitergeleitet werden (Weiterleiten als Anhang), was längst nicht alle gängigen E-Mail-Clients beherrschen.
Um für den User die Möglichkeit zu schaffen, selber falsch klassifizierte Nachrichten dem selbstlernenden Filter zuzuführen, müssen dafür zwei spezielle E-Mail-Adressen angelegt werden. Beispielsweise könnten die Adressen spam@<mydomain> und notspam@<mydomain> lauten.
Bei Falsch-Klassifizierungen reicht es, diese Nachrichten an die weiter unten erwähnten E-Mail-Adressen zu senden. Anhand der speziellen Signatur kann DSPAM auf die ursprüngliche Nachricht schließen und die Tokens neu lernen.
Jetzt fehlen noch die entsprechenden MySQL Tabellen in der vpopmail Datenbank. DSpam liefert für MySQL bereits fertige Scripts (befinden sich im Source-Verzeichnis unter `/src/tools.mysql_drv`), mit Hilfe deren diese in zwei einfachen Schritten erstellt werden können:

```
mysql -p vpopmail < mysql_objects-4.1.sql
```

```
mysql -p vpopmail < virtual_users.sql
```

A.1.9. qpsmtpd

Um unsere qpsmtpd Version erfolgreich installieren zu können, müssen folgende Perl Module vorhanden sein:

- Net::DNS
- MIME::Base64
- Mail::Header
- Geo::IP
- DBI

Kommt eine Perl Version älter als 5.8.0 zum Einsatz, benötigt man noch zusätzlich folgende Module

- Data::Dumper
- File::Temp
- Time::HiRes

Die weitere Installation gestaltet sich dann relativ simpel. Nach erfolgtem Download des Sourcecodes in ein beliebiges Verzeichnis, kann mit dem Einrichten des Dämons begonnen werden. Das Starten des Dämons kann auf mehrere Weisen erfolgen. In dieser Umsetzung wird der Prozess mittels des während der QMail-Installation kennengelernten daemontools Kit gestartet und überwacht. Bevor dies geschieht, muss sichergestellt werden, dass sämtliche Files im qpsmtpd Source-Verzeichnis als Besitzer vpopmail und als Gruppe vchkpw aufweisen. Unser run-Script sieht folgendermaßen aus:

```
#!/bin/sh
exec 2>&1 \
sh -c '
  exec \
    /usr/local/bin/softlimit -m 50000000 \
    ${PERL-perl} -T ./qpsmtpd-forkserver \
    --listen-address 0 \
```

```
    --port 25 \
    --limit-connections 100 \
    --max-from-ip 5 \
    --user vpopmail
'
```

Da einige Plugins direkten Zugriff auf die User-Postfächer brauchen, ist es notwendig, dass vpopmail und qpsmtpd unter demselben Benutzer laufen. Diese Implementierung hebelt teilweise das von QMail umgesetzte Sicherheitskonzept aus, d. h. im Falle einer auftretenden Sicherheitslücke ist unter Umständen der Zugriff auf einzelne Mail-Boxen möglich.
Im obigen run-Script sind maximal 100 gleichzeitige SMTP-Verbindungen und maximal fünf von derselben IP-Adresse möglich. Durch Anlegen eines symbolischen Links vom daemontools Serviceverzeichnis wird unser Dämon nun umgehend gestart:

```
cd /service
ln -s <qpsmtpd source directory> qpsmtpd
```

Mit `ps -AH|grep perl` kann man nun überprüfen, ob der Start des Dämons erfolgreich war. Für jede neue eingehende Verbindung wird ein neuer Child-Prozess erzeugt, die qpsmtpd Einstellungen werden jedoch nur beim Starten des Elternprozesses geladen.

A.1.10. SquirrelMail

Nach erfolgtem Download von SquirrelMail [13] dieses in beliebiges Web-Verzeichnis entpacken. Die Installationsanleitung auf der offiziellen Website ist recht umfangreich, weswegen hier auf eine weitere Beschreibung verzichtet wird.
Für unser Spam-Filter Gateway wurden speziell zwei Plugins entwickelt, zum einen das Logging-Plugin und zum anderen das Plugin zum Bearbeiten der Spam-Filter Einstellungen. Da diese Plugins direkten Zugriff auf die vpopmail-Datenbank benötigen, muss die Konstante `BOOKMARKS_DSN` jeweils mit den richtigen Zugangsdaten gefüttert werden. Weitere Adaptierungen sind nicht notwendig.
Um diese Plugins in SquirrelMail verwenden zu können, diese ins dafür vorgesehene Plugin-Verzeichnis kopieren und mit dem Konfigurationsskript aktivieren.

A.2. SMTP Return Codes

```
200 (nonstandard success response, see rfc876)
211 System status, or system help reply
214 Help message
220 [domain] Service ready
221 [domain] Service closing transmission channel
250 Requested mail action okay, completed
251 User not local; will forward to
354 Start mail input; end with [CRLF].[CRLF]
421 [domain] Service not available, closing transmission channel
450 Requested mail action not taken: mailbox unavailable
451 Requested action aborted: local error in processing
452 Requested action not taken: insufficient system storage
500 Syntax error, command unrecognised
501 Syntax error in parameters or arguments
502 Command not implemented
503 Bad sequence of commands
504 Command parameter not implemented
521 [domain] does not accept mail (see rfc1846)
530 Access denied (???a Sendmailism)
535 SMTP Authentication unsuccessful/Bad username or password
550 Requested action not taken: mailbox unavailable
551 User not local; please try
552 Requested mail action aborted: exceeded storage allocation
553 Requested action not taken: mailbox name not allowed
554 Transaction failed
```

A.3. DSpam Lernscripts

Folgendes Script muss aufgerufen werden, sobald eine E-Mail an die vorher definierte Adresse notspam@<mydomain> gesendet wird. In QMail geschieht das einfach durch das Erstellen einer .qmail Datei.

```
1 #!/bin/sh
2 INPUT="/dev/stdin"
3
```

```
4  ADDRESS=`echo $SENDER|/bin/sed -n \
5                 's/^prvs=\(.*\)\/.*@\(.*\)/\1@\2/ip;T;q'`
6  /home/dspam/bin/dspam --user $ADDRESS --mode=tum \
7                 --feature=noise,whitelist \
8                 --source=error --class=innocent < $INPUT
9
10 DOMAIN=`echo $SENDER|/bin/sed -n \
11                      's/^prvs=\(.*\)\/.*@\(.*\)/\2/ip;T;q'`
12 /home/dspam/bin/dspam --user $DOMAIN --mode=tum \
13                --feature=noise,whitelist \
14                --source=corpus --class=innocent < $INPUT
```

Folgendes Script muss aufgerufen werden, sobald eine E-Mail an die vorher definierte Adresse spam@<mydomain> gesendet wird.

```
1  #!/bin/sh
2
3  INPUT="/dev/stdin"
4
5  ADDRESS=`echo $SENDER|/bin/sed -n \
6                 's/^prvs=\(.*\)\/.*@\(.*\)/\1@\2/ip;T;q'`
7  /home/dspam/bin/dspam --user $ADDRESS --mode=tum \
8                 --feature=noise,whitelist \
9                 --source=error --class=spam < $INPUT
10
11 DOMAIN=`echo $SENDER|/bin/sed -n \
12                 's/^prvs=\(.*\)\/.*@\(.*\)/\2/ip;T;q'`
13 /home/dspam/bin/dspam --user $DOMAIN --mode=tum \
14                --feature=noise,whitelist \
15                --source=corpus --class=spam < $INPUT
```

A.4. Perl Outputfilter für Apache 2.0

```
1  # Perl Output Filter for Apache 2.0:
2  # Automatically conceals email addresses to
3  # prevent harvesters from fetching them.
4
5  package MyApache::ObMail;
6
7  use strict;
8  use warnings;
```

```
9  use Apache::Filter ();
10 use Apache::RequestRec ();
11 use APR::Table ();
12 use Apache::Const -compile => qw(OK DECLINED);
13
14 use constant BUFF_LEN => 10240;
15 # Store apache output data
16
17 sub obfuscate
18 {
19     # Invocation: obfuscate(data)
20     # Conceal all mail addresses
21
22     my $line = shift;
23     my $mail_regexp = '[A-Za-z_0-9.-]+@'.
24         '([A-Za-z_0-9-]+'.
25         '.)+[A-Za-z]{2,6}';
26     my $adr = undef;
27
28     while ($line =~ /($mail_regexp)/g)
29     {
30         # Split address into single characters
31         # and reassemble with spaces in between
32         $mail = $1;
33         $obfus = join(' ',split(//,$mail));
34
35         # Replace all occurences
36         $line =~ s/$mail/$obfus/gi;
37     }
38     return $line;
39 }
40
41 sub handler
42 {
43     # Called by Apache. Works through the
44     # blocks of data delivered by the httpd.
45
46     my $f = shift;
47     unless ($f->ctx)
48     {
49         # Test content-type on first invocation
50         unless ($f->r->content_type =~m!text/(html|plain)!i )
```

```
51          {
52              # Only modify text/html and text/plain
53              return Apache::DECLINED;
54          }
55          # Reset Content-Length calculated by the
56          # server. We'll change the amount of data
57          $f->r->headers_out->unset('Content-Length');
58      }
59
60      my $leftover = $f->ctx;
61      while ($f->read(my $buffer, BUFF_LEN))
62      {
63          $buffer = $leftover.$buffer
64          if defined $leftover;
65          if (length($buffer) > (2*BUFF_LEN))
66          {
67              # Don't wait forever for whitespace
68              $f->print(obfuscate($buffer));
69              $buffer = $leftover = "";
70          }
71          else
72          {
73              # Keep the last beginning of a word
74              # in leftover to work only on full
75              # addresses and not on fragments.
76              $buffer =~ /(.*)(sS*)z/gs;
77              $leftover = $2;
78              $f->print(obfuscate($1));
79          }
80      }
81      if ($f->seen_eos)
82      {
83          # End of data-stream in sight.
84          if (defined $leftover)
85          {
86              $leftover=obfuscate($leftover);
87              $f->print(scalar $leftover);
88          }
89      }
90      else
91      {
92          # Pass remaining data to next invocation
```

```
93          $f->ctx($leftover) if defined $leftover;
94      }
95      return Apache::OK;
96  }
97
98  1;
```

Perl Outputfilter für Apache 2.0 [92]

A.5. BATV Qmail Patch

```
1  --- qmail-send.c.org    2008-06-06 20:00:29.000000000 +0200
2  +++ qmail-send.c    2008-06-07 01:18:18.000000000 +0200
3  @@ -661,6 +661,9 @@
4    static stralloc quoted = {0};
5    datetime_sec birth;
6    unsigned long qp;
7  + /* BATV patch */
8  + int i;
9  + int sig_len;
10
11   if (!getinfo(&sender,&birth,id)) return 0; /* XXX: print warning
        */
12
13 @@ -671,7 +674,24 @@
14        sender.len -= 4;
15        sender.s[sender.len - 1] = 0;
16       }
17 -
18 + /* BATV qmail patch */
19 + /* prvs=0919d765b7=example@bitwise.it */
20 + /* remove BATV signature */
21 + if (sender.len >= 5)
22 + {
23 +    if (!str_diffn(sender.s,"prvs=",5))
24 +    {
25 +        i = str_chr(sender.s + 5, '=');
26 +        if ( sender.s[i] )
27 +        {
28 +            sig_len = i + 6;
29 +            byte_copy(sender.s, sender.len - sig_len, sender.s +
       sig_len );
```

```
30 +             sender.len = sender.len - sig_len;
31 +             sender.s[sender.len] = '\0';
32 +         }
33 +     }
34 + }
35 + /* End BATV handling */
36   fnmake2_bounce(id);
37   fnmake_mess(id);
38   if (stat(fn2.s,&st) == -1)
```

Ins Qmail Source Verzeichnis wechseln und Patch anwenden mit:

```
patch -p0 <patch_file
make qmail-send
```

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[1] *Jeremy Jaynes Prozessunterlagen*.

[2] *ClamAV*. http://www.clamav.net, Besucht im Juni 2008.

[3] *DomainKeys Identified Mail*. http://www.dkim.org, Besucht im Juni 2008.

[4] *Life with QMail*. http://www.lifewithqmail.org/lwq.html#history, Besucht im Juni 2008.

[5] *netqmail*. http://www.qmail.org/netqmail, Besucht im Juni 2008.

[6] *Nuclear Elephant: The DSPAM Project*. http://dspam.nuclearelephant.com, Besucht im Juni 2008.

[7] *The qmail security guarantee*. http://cr.yp.to/qmail/guarantee.html, Besucht im Juni 2008.

[8] *qpsmtpd*. http://smtpd.develooper.com, Besucht im Juni 2008.

[9] *qpsmtpd Wiki*. http://wiki.qpsmtpd.org, Besucht im Juni 2008.

[10] *Sender ID Homepage - Microsoft*. http://www.microsoft.com/mscorp/safety/technologies/senderid/default.mspx, Besucht im Juni 2008.

[11] *Sender Policy Framework - Projekt Page*. http://www.openspf.org, Besucht im Juni 2008.

[12] *Spamgourmet - Wegwerf-E-Mail-Adressen*. http://www.spamgourmet.com, Besucht im Juni 2008.

[13] *SquirrelMail - Webmail for Nuts!* http://www.squirrelmail.org, Besucht im Juni 2008.

[14] *Wegwerfadressen bei Yahoo! Mail*. http://de.docs.yahoo.com/benefits/deatour, Besucht im Juni 2008.

[15] *The Apache SpamAssassin Project*. http://spamassassin.apache.org, Besucht im Mai 2008.

[16] *Distributed Checksum Clearinghouse*. http://www.dcc-servers.net/dcc, Besucht im Mai 2008.

[17] *SORBS*. http://www.sorbs.net, Besucht im Mai 2008.

[18] *Spam URI Blacklists*. http://www.surbl.org, Besucht im Mai 2008.

[19] *MessageLabs Intelligence monthly report*. http://www.messagelabs.com, Februar 2008.

[20] Adam Back. *Hashcash - A Denial of Service Counter-Measure*. http://www.hashcash.org/papers/hashcash.pdf, August 2002.

[21] BBC. *Monty Python's Flying Circus*. http://www.bbc.co.uk/comedy/montypython, Besucht m Mai 2008.

[22] D. J. Bernstein. *Variable Envelope Return Paths*. http://cr.yp.to/proto/verp.txt, 1997.

[23] Brian Burton. *SpamProbe - Bayesian Spam Filtering Tweaks*. http://spamprobe.sourceforge.net/paper.html, Besucht im Mai 2008.

[24] H. Martin-Jung C. Schrader. *„Sie haben Müll"*. Süddeutsche Zeitung - Online Ausgabe, http://www.sueddeutsche.de/computer/artikel/968/166491, 1. April 2008.

[25] J. Callas. *OpenPGP Message Format. RFC 4880*. RFC Editor, http://www.ietf.org/rfc/rfc4880.txt, November 2007.

[26] Center for Democracy and Technology. *Why am I getting all this spam?* web, http://www.cdt.org/speech/spam/030319spamreport.pdf, 2003.

[27] Ciphertrust. *Spam statistics*. http://www.ciphertrust.com/resources/statistics/index.php, Besucht im Mai 2008.

[28] Richard Clayton. *Stopping Spam by Extrusion Detection*. http://www.cl.cam.ac.uk/ rnc1/extrusion.pdf, 2004.

[29] M. Crispin. *Internet Message Access Protocol, Version 4. RFC 3501*. RFC Editor, http://www.ietf.org/rfc/rfc3501.txt, März 2003.

[30] David H. Crocker. *Standard For The Format Of Arpa Internet Text Messages. RFC 822*. RFC Editor, http://www.ietf.org/rfc/rfc0822.txt, August 1982.

[31] Joshua Cyr. *Spam Stock Tracker*. http://www.spamstocktracker.com, Besucht im Mai 2008.

[32] Christopher K. Davis. *QMail DNS Patch*. http://www.ckdhr.com/ckd/qmail-103.patch, Besucht im Mai 2008.

[33] Christian Dietrich. Abfrageverhalten von blacklists. Jänner 2008.

[34] Inc Double Precision. *Simple Mail Access Protocol, Version 1*. http://www.courier-mta.org/cone/smap1.html, Besucht im Februar 2008.

[35] Double Precision, Inc. *Courier Authentication Library*. http://www.courier-mta.org/authlib, Besucht im Juni 2008.

[36] Double Precision, Inc. *Courier-IMAP*. http://www.courier-mta.org/imap, Besucht im Juni 2008.

[37] M. Delany M. Libbey J. Fenton M. Thomas E. Allman, J. Callas. *Domain-Keys Identified Mail (DKIM) Signatures. RFC 4871*. http://www.ietf.org/rfc/rfc4871.txt, Mai 2007.

[38] Tobias Eggendorfer. *Methoden der präventiven Spambekämpfung im Internet*. Diplomarbeit, Fern-Universität in Hagen, 2005.

[39] Tobias Eggendorfer. Reducing spam by using a proxy simulating a smtp tar pit on a bridge. *Conference on Communications, Internet and Information Technology, St. Thomas, US Virgin Islands*, 2006.

[40] Tobias Eggendorfer. Her mit dem abfall. *Linux-Magazin Deutschland*, Jänner 2007.

[41] Enigma SEO Consulting. Combating email harvester robots - email obfuscation. Besucht im Mai 2008.

[42] Deborah Fallows. *How It Is Hurting Email and Degrading Life on the Internet*. Pew Internet & American Life Project, http://www.pewinternet.org/pdfs/PIP_spam_Report.pdf, 2003.

[43] Jack Ferreri. *Entrepreneur Magazine's Knock-Out Marketing: Powerful Strategies to Punch Up Your Sales*. Entrepreneur Press, 1999.

[44] Ferris Research. The global economic impact of spam. http://www.ferris.com/2005/02/24/the-global-economic-impact-of-spam-2005, 2005.

[45] Fries Websolutions. *Wie bekommen Spammer meine eMail Adresse?* web, http://www.htmlopen.de/webdesign/email/spam_adressen.html, Besucht im April 2008.

[46] Joshua Goodman Geoff Hulten. *Tutorial on Junk Mail Filtering*. Microsoft Research, 2003.

[47] Paul Graham. *A Plan for Spam*. http://www.paulgraham.com/spam.html, August 2002.

[48] Georgi Guninski. *64 bit qmail fun*. http://www.guninski.com/where_do_you_want_billg_to_go_today_4.html, 2005.

[49] Robert J. Hall. How to avoid unwanted email. *Communications of the ACM*, pages 88–95, März 1998.

[50] Shane Hird. *Technical Solutions for Controlling Spam*. Distributed Systems Technology Centre, 2002.

[51] Mirko Dölle Holger Bleich. Spam-golem. *Magazin für Computer und Technik, c't*, page 118, 2/2008.

[52] R. Housley. *Internet X.509 Public Key Infrastructure. RFC 3280*. RFC Editor, http://www.ietf.org/rfc/rfc3280.txt, April 2002.

[53] ICQ Inc. *Security and Privacy - Anti Spam*. http://www.icq.com/support/security/spam.html, Besucht im Mai 2008.

[54] Inter7 Internet Technologies, Inc. *qmailadmin*. www.inter7.com/qmailadmin, Besucht im Juni 2008.

[55] Inter7 Internet Technologies, Inc. *vpopmail*. www.inter7.com/vpopmail, Besucht im Juni 2008.

[56] Inter7 Internet Technologies, Inc. *vqadmin*. www.inter7.com/vqadmin, Besucht im Juni 2008.

[57] Al Iverson. *Blacklist Statistics Center*. http://stats.dnsbl.com, Besucht im Mai 2008.

[58] S. Silberman T. Finch J. Levine, D. Crocker. *Internet-Draft, Bounce Address Tag Validation - (BATV)*. http://tools.ietf.org/html/draft-levine-smtp-batv-00, Jänner 2007.

[59] Gihyun Jung2 Jangbok Kim, Kyunghee Choi. Spam filtering with dynamically updated url statistics. *IEEE Security u. Privacy*, pages 33–39, 4/2007.

[60] Eric S. Johansson. *Campaign for real mail*. http://freshmeat.net/projects/camram, Besucht im Mai 2008.

[61] Robert Rounthwaite Joshua T. Goodman. Stopping outgoing spam. *Proceedings of the 5th ACM conference on Electronic commerce*, pages 30–39.

[62] J. Klensin, editor. *Simple Mail Transfer Protocol. RFC 2821*. RFC Editor, http://www.ietf.org/rfc/rfc2821.txt, April 2001.

[63] J. Klensin. *SMTP Service Extensions. RFC 1869*. RFC Editor, http://www.ietf.org/rfc/rfc1869.txt, November 1995.

[64] Matthias Leisi. *Spam Biz*. web, http://matthias.leisi.net/archives/80_Spam_Biz.html, September 2004.

[65] John Levine. *Bounce Address Tag Validation*. http://www.mipassoc.org/batv/index.html, Besucht im Mai 2008.

[66] John R. Levine. *Experiences with Greylisting*. 2005.

[67] Merriam-Webster. *Spam Definition*. http://www.merriam-webster.com/dictionary/spam, 1994.

[68] P. Mockapetris. *Domain Names - Concepts and Facilities. RFC 1034*. RFC Editor, http://www.ietf.org/rfc/rfc1034.txt, November 1987.

[69] K. Moore. *Recommendations for Automatic Responses to Electronic Mail*. http://tools.ietf.org/html/rfc3834, August 2004.

[70] Raja Afandi Munawar Hafiz, Ralph E Johnson. The security architecture of qmail. *University of Illinois at Urbana-Champaign, Department of Computer Science*, 2004.

[71] N. Borenstein N. Freed. *Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples. RFC 2049*. RFC Editor, http://www.ietf.org/rfc/rfc2049.txt, November 1996.

[72] N. Borenstein N. Freed. *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies. RFC 2045*. RFC Editor, http://www.ietf.org/rfc/rfc2045.txt, November 1996.

[73] John Naughton. *A Brief History of the Future, The Origins of the Internet*. Phoenix, 2000.

[74] Nuclear Elephant. *Advanced Language Classification using Chained Tokens*. http://dspam.nuclearelephant.com/papers/chained.html, Februar 2004.

[75] J. Palme. *Common Internet Message Headers. RFC 2076*. RFC Editor, http://www.ietf.org/rfc/rfc2076.txt, February 1997.

[76] Weilai Yang Paul Judge, Dmitri Alperovitch. *Understanding and Reversing the Profit Model of Spam*. CipherTrust Inc., 2005.

[77] Alexander Wirt Peter Eisentraut. *Mit Open Source-Tools Spam u. Viren bekämpfen*. O'Reilly Verlag, Juli 2005.

[78] Michael Stini Martin Mauve Peter Lieven, Björn Scheuermann. Filtering spam email based on retry patterns. *International Conference on Communications, IEEE*, pages 1515–1520, 2007.

[79] The Spamhaus Project. *The Definition of Spam*. http://www.spamhaus.org/definition.html, Besucht im Mai 2008.

[80] David Purdue. *Adventures in the tar pit*. http://www.openbsd.org/spamd, Besucht im Mai 2008.

[81] Thorsten Holz Rainer Böhme. The effect of stock spam on financial markets. *Working Paper Series*, April 2006.

[82] B. Ramsdell, editor. *S/MIME Version 3 Message Specification. RFC 2633*. RFC Editor, http://www.ietf.org/rfc/rfc2633.txt, Juni 1999.

[83] Uri Raz. *How do spammers harvest email addresses?* web, http://www.private.org.il/harvest.html, Besucht im April 2008.

[84] P. Resnick, editor. *Internet Message Format. RFC 2822*. http://www.ietf.org/rfc/rfc2822.txt, April 2001.

[85] Christiane Rütten. Testbilder - der kampf zwischen captcha-entwicklern und spam-bots. *Magazin für Computer Technik, c't, 9/2008*, pages 188 – 190, 2008.

[86] Jonathan E. Schmidt. Dynamic port 25 blocking to control spam zombies. *CEAS Papers*, 2006.

[87] silicon.de. Identity-diebstahl zieht weite kreise. http://www.silicon.de/sicherheit/management/0,39039020,39167085,00/identity_diebstahl+zieht+weite+kreise.htm, Besucht im April 2008.

[88] Sophos Deutschland. *Das 'Dreckige Dutzend': Immer mehr Spam-Mails mit integrierten Weblinks im Umlauf*. http://www.sophos.de/pressoffice/news/articles/2008/04.html, April 2008.

[89] Bob Sullivan. *Who profits from spam? Surprise*. MSNBC, http://www.msnbc.msn.com/id/3078642, 2003.

[90] Inc. Symantec. *The State of Spam, A Monthly Report*. http://eval.symantec.com/mktginfo/enterprise/other_resources/SpamReport_March08.pdf, März 2008.

[91] A. S. Tanenbaum. *Computer Networks*. Prentice Hall Inc., 4. edition, 2003.

[92] Jörg Keller Tobias Eggendorfer. Preventing spam by dynamically obfuscating emailaddresses. *From Proceeding. Communication, Network, and Information Security*, 2005.

[93] Jochen Topf. Aktuelle Entwicklungen bei der Spam-Bekämpfung. 2005.

[94] Thorsten Urbanski. *Das große Geschäft mit dem E-Müll*. GData AG, http://www.gdata.de/unternehmen/DE/articleview/3920/1/160, Oktober 2007.

[95] W. S. Yerazunis. *The CRM114 Discriminator Revealed! or How I Learned to Stop Worrying and Love My Automatic Monitoring Systems*. http://crm114.sourceforge.net/docs/CRM114_Revealed_20061010.pdf, 2005.

[96] Jonathan A. Zdziarski. *Ending Spam*. No Starch, 2005.