7 Qmail im Einsatz

7.1 Qmail als Gateway

7.2 Unterdrückung von Spam E-Mails

Vor einiger Zeit hätte an dieser Stelle noch eine umfangreiche Erklärung gestanden, was Spam E-Mails sind und woher dieses Wort überhaupt stammt. Angesichts der realen Verhältnisse Anno 2003 mit einem Anteil von Spam E-Mails von 30% bis 50% braucht niemand mehr eine entsprechende Belehrung. Spam ist zum Alltag geworden; die Bekämpfung bzw. Unterdrückung von Spam E-Mails zu einem Bedürfnis.

Kommerzielle E-Mails

Dabei möchte ich nicht die Berechtigung des kommerziellen Einsatzes von E-Mails in Frage stellen und diese von Spam E-Mails abgrenzen. Um kommerzielle E-Mails zu erhalten, bedienen sich die Anbieter im wesentlichen folgender Methoden:

- **Opt-In** — Der Interessent muss sich z.B. über ein Webformular mit seiner E-Mail-Adresse eintragen und bestätigt somit den Empfang der Werbemails. Der Anbietet trägt ihn dann in eine E-Mail-Verteilerliste ein und verschickt eine Bestätigungs-E-Mail. Nachteil diese Verfahrens ist, dass natürlich auch Dritte unter falschen Namen und E-Mail-Adresse sich eintragen können (Adressen-Spoofing).
- Daher bieten alle Anbieter auch ein **Opt-Out** an, mit man sich von der Liste wieder löschen kann. Häufig wird auch bei Spam E-Mails eine Opt-Out-Möglichkeit angeboten; in aller Regel ist dies nicht seriös gemeint sollte daher nie genutzt werden (siehe unten).
- **Double Opt-In** — Hierbei trägt sich der Interessent zunächst wie beim Opt-In in eine E-Mail-Verteilerliste ein; diese verschickt aber dann nicht eine einfache Bestätigungs-E-Mail sondern ein sog. Cookie, dass der Absender bestätigen muss. Erst in diesem Falle wird er in die Verteilerliste aufgenommen. Vorteil hierbei ist, dass sowohl die Gültigkeit der E-Mail-Adresse des Interessenten überprüft, als auch ein mögliches Spoofing durch Dritte unterbunden wird.

7.2.1 Unsolicited Commercial E-Mails (UCE)

In Abgrenzung hierzu, haben wir es bei Spam um sog. Unsolicited Commerical E-Mails zu tun. Tatsache ist jedenfalls, dass der Umfang in den letzen Jahren/Monaten im allgemeinen stark zugenommen hat. Die individuelle

Belästigung durch Spam schwankt jedoch beträchtlich; und zwar sowohl hinsichtlich der emotionalen Betroffenheit als auch hinsichtlich der Anzahl der Spam E-Mails. Es gibt zwar einige Aussagen hinsichtlich des Umfangs von Spam bei der Internet E-Mail, aber die Schwankungsbreite dieser Zahlen ist beträchtlich und wird heute zwischen 30% und 60% abgeschätzt.

7.2.1.1 Ursachen

Spam E-Mails dienen in der Regel (> 99%) zur Anbiederung kommerzieller Dienste oder Produkte. Ganz vorne in der Statistik agieren Porno E-Mails (direkt oder indirekte "scharfe Fotos auf meiner Web-Site"), gefolgt von diversen pharmazeutisch-medizinischen Hilfsmitteln à la Viagra, Penis- und Brustvergrösserungen, der bekannten "Blue Pill" (nach einem Song der Bangels), in Deutschland 0190-Dialern, Anti-Spam Tools (!) und natürlich der "Nigeria-Connection".

Gemeinsam ist, dass trotz minimalster Trefferraten und Effizienz, die Gewinnmargen (besser gesagt: das Leimen der Betroffenen) so lukrativ ist, dass sich solche Massen-E-Mails lohnen. Die Trefferraten bewegen sich häufig im Promille-Bereich; wieviel der E-Mail-Anwender dann hierauf reagieren, ist unklar. In Anbetracht des wachsenden Verdrusses über Spam E-Mails sinkt dieser Anteil sicherlich (ein Alleinstellungsmerkmal fehlt), dem die Spammer durch höhere Trefferraten begegnen wollen ("Brut-Force-Attacs"), was das ganze System hochschaukelt.

7.2.1.2 Herkunft

Soweit bekannt ist, stammen die meisten Spam E-Mails aus USA, Brasilien, Korea und Deutschland. Verfolgen kann man den Weg der Spam E-Mails an der Reihenfolge der "Received:" Zeilen im Header der E-Mail. Hier ist besonders der erste (d.h. unterste) Eintrag von Belang, da dies die erste Station der Spam E-Mail darstellt. Ob allerdings in der Zukunft die Spammer nicht dazu übergehen, diese zu fälschen (siehe weiter unten), ist allerdings unklar.

Spammer nehmen die Erzeugung und den Versand ihrer Spam E-Mails in wachsendem Masse professionell vor:

- Für den Zweck des Spam-Versands werden eigne Rechner aufgebaut, deren "Baurezept" teilweise in den Spam E-Mails selbst offen beworben wird, und nach Ende der Spam-Aktion wieder abgebaut werden.
- Die Spam-Absender nutzen schnelle Internet-Anbindungen (z.B. DSL) über die eine grosse Anzahl von Spam E-Mails in kurzer Zeit versandt werden können. Wir reden hier von einem erzeugten E-Mail Aufkommen von Millionen/Stunde.
- Alternativ werden offene E-Mail-Server ("Open Relays") sowie offene Proxis genutzt, die vorher gescannt und auf ihre Relaying-Eigenschaft getestet

wurden. Besonders unrühmlich hervorgetan haben sich einige Windows-Produkte.

Die Herkunft der E-Mail — kenntlich durch die SMTP "Mail From:" Adresse — ist in aller Regel gefälscht, um möglichst unverdächtig zu erscheinen. Häufig werden Adressen der grossen E-Mail Internet-Anbieter wie "yahoo.com", "yahoo.de", "web.de", "hotmail.com" und "gmx.de" gefälscht, wobei dies davon abhängig gemacht wird, in welches Land (per Domain-Kennung) die Spam E-Mail versandt wird. Der Benutzername, d.h. der Teil der E-Mail-Adresse links vor dem "@" wird meist über eine Zufallsfolge von Buchstaben von Zahlen generiert; was durchaus auch bei normalen Anwendern nicht unüblich ist.

7.2.1.3 Adressen und Adresshandel

Den Umfang der Spam E-Mails sowohl nach Anzahl wie nach übertragenem Volumen abzuschätzen ist in aller Allgemeinheit nicht notwendig. An dieser Stelle wage ich zu unterschieden zwischen:

- Öffentlichen Institutionen, die einen Teil ihrer E-Mail Adressen veröffentlichen; sei es über das Web, über Newsgroups, LDAP, X.500 oder über andere Medien.
- Firmen, deren E-Mail Adressen in der Regel nur Kunden bekannt sind und die darüber hinaus nur "funktionale" Adressen öffentlich machen (Postmaster, Vertrieb, Hotline etc.).
- Anwender mit privaten E-Mail-Adressen bei unterschiedlichen ISPs und E-Mail-Providern.
- Anwender, mit E-Mail-Adressen bei den sog. Free-Mailern wie z.B. GMX, Web.de, Yahoo.de oder anderen.
- Anwender, die sich an öffentlichen Foren im Internet beteiligen und dort registriert sind.

Um überhaupt an brauchbare E-Mail-Adressen zu kommen, bedienen sich die Spammer mehrere Methoden[1]:

- **Adresshandel** — Der derzeitige Preis für eine E-Mail-Adresse beträgt etwa 0,002 Cent; also 5 Millionen Adressen für 99 $. Entsprechende Datensätze werden selbst als Spam beworben; teilweise mit entsprechender E-Mail Software dazu, um schnell Spam E-Mails versenden zu können. Inwieweit die kommerziellen Internet-Händler (wie Amazon) ihre Adressen an Dritte vermarkten, bleibt im Verborgenen. Einerseits garantieren diese natürlich einen Kundenschutz (bereits in ihrem eigenen Interesse); anderseits sind speziell ihre Adressen (mit den entsprechenden Kundenprofile) von einem

[1] vgl. http://www.rehbein-dortmund.de/spamtrap.html

hohen Interesse und daher sehr "wertvoll".

- **Harvesting** — Internet Web-Seiten sowie Internet-Foren wie Newsgroups werden über sog. E-Mail Spider auf verwertbare Adressen untersucht und diese gesammelt.
- **Fake FTP** — Bei der Verbindung auf eine Web-Seite wird versucht, den eingebauten FTP-Client des Web-Browsers zu starten; in der Regel versucht dieser ein sog. Anonymous-Login mit der E-Mail-Adresse des Anwenders als Passwort.
- **Rücklauf** — "Gold Plated" Adressen erhält der Spammer, falls tatsächlich einer der Adressaten von der "Opt-Out" Möglichkeit Gebrauch macht. Zur Erinnerung: Als "Opt-In" bezeichnet man die Möglichkeit, sich auf E-Mail Verteilern einzuschreiben; als "Opt-Out" sich entfernen zu lassen. EZMLM verwendet z.B. das "Double-Opt-In" Verfahren: Der potentielle Empfänger muss sich also zum erfolgreichen Einschreiben noch einmal durch eine Response-Mail auf Grundlage seiner eingegebenen E-Mail-Adresse legitimieren.

Neben der Werbung per Spam E-Mail ist also der Adresshandel im Internet zu einem lukrativen Geschäft geworden. Dies gilt im Besonderen in dem Kontext, dass speziell die E-Mail-Adressen der grossen Freemail-Provider schnell an Aktualität verlieren. Das ständige Katz-und-Maus Spiel erzeugt somit seine eigenen Regeln.

7.2.2 Arbeitsweise der Spammer

7.2.2.1 Temporäre Domains

Der einfachste und sicherste Weg für einen potentiellen Spammer ist es, eine oder gleich mehrere Domänen bei einem NIC anzumelden, einen Rechner mit E-Mail-Software aufzusetzen und über ein Skript die eingespeicherten E-Mail Adressen mit dem vorgegeben Text abzusetzen.

Innerhalb einer Stunde können so mehrere Millionen E-Mails abgesetzt werden. Nach dem Ende der Aktion werden Domains abgemeldet oder einfach "leer" gelassen und der Rechner von der (DSL-) Leitung abgeklemmt und abgeschaltet.

Beispiel für temporäre Domains

Laut einer E-Mail von Markus Stumpf (SpaceNet) wurden Anfang Juli 2003 u.a. folgende temporäre Domains zum Absenden von Spam E-Mails benutzt:

```
"   @brightbernies.us
    @brightdons.us
    @brighthenrys.us
    @brightsandys.us
    @dazzlingdans.us
    @fabulousozzies.us
```

```
    @famousmickeys.us
    @fantasticteds.us
    @friendlyhenrys.us
    @friendlyozzies.us
    @friendlysandys.us
    @friendlyvinnies.us
    @heathersholly.us
    @mightydans.us
    @robsrules.us
    @smartdons.us
    @smarthenrys.us
    @wittyozzies.us
    @wranglerron.us
    [ ... ] this list is by far incomplete

We got spam from these domains today and the last few days.
The spammers obviously registered these domain, they have DNS
servers for them and even MX and A records. The TTL of these
records is one hour. All of these are registered by

Domain Name:                    FRIENDLYSANDYS.US
Domain ID:                      D4426467-US
Sponsoring Registrar:           GO DADDY SOFTWARE, INC.
Domain Status:                  ok
Registrant ID:                  GODA-03439749
Registrant Name:                steven little
Registrant Organization:        Unknown
Registrant Address1:            321 s college
Registrant City:                seattle
Registrant State/Province:      Washington
Registrant Postal Code:         98144
Registrant Country:             United States
Registrant Country Code:        US
Registrant Phone Number:        +1.2063503057
Registrant Email:               stevenlittle206@yahoo.com
    [ ... ]
Name Server:                    NS1.ENCHANTINGIDEAS.NET
Name Server:                    NS2.ENCHANTINGIDEAS.NET
Created by Registrar:           GO DADDY SOFTWARE, INC.
Last Updated by Registrar:      GO DADDY SOFTWARE, INC.
Domain Registration Date:  Tue Jul 01 04:19:58 GMT 2003
Domain Expiration Date:    Wed Jun 30 23:59:59 GMT 2004
Domain Last Updated Date:  Tue Jul 01 04:29:52 GMT 2003

Get some millions throwaway accounts for as cheap as 5 USD."
```

7.2.2.2 Open Relays

Um eigene Ressourcen zu schonen und die Effizienz zu erhöhen, benutzen Spammer sog. offene Relays — also SMTP-Server, die ein unbeschränktes Weitersenden von E-Mails an beliebige Empfänger zulassen. Qmail, z.B. agiert als offenes Relay wenn weder die Datei `./rcpthosts` noch `./morercpthosts` vorhanden bzw. befüllt ist.

Jeder verantwortliche System-Administrator sichert natürlich sofort seinen E-Mail-Server gegen unbeabsichtigtes Relaying ab. Mit der Verbreiterung des Internets bis in den letzten Winkel der Welt, werden jedoch insbesondere Web- und E-Mail-Server aufgesetzt, wo dieses Paradigma nicht gilt. "Berüchtigt" in diesem Zusammenhang sind z.B. entsprechende Server in Korea, die in Schulen betrieben werden. Verfügt die Schule dann auch noch über einen schnellen Internet-Zugang, eignen sich diese Server als ausgezeichnete Relay-Systeme Spam E-Mails zu verschicken. Da für den mittelbar Betroffenen auch kein Schaden entsteht (ausser der Nutzung seiner Bandbreite) bleibt das Relaying auch folgenlos — entsprechend gering fällt die Verantwortlichkeit gegenüber der "Internet-Comunity" bzw. den unmittelbar Betroffenen aus.

7.2.2.3 Open Proxies

Im Zeitalter der DSL-Flat-Rate und der Home-LANs bauen sich viele Windows-Anwender permanent arbeitende SMTP-Proxies auf, über die E-Mails empfangen und versandet werden. PCs unter Windows XP Home Edition besitzen prinzipiell die gleichen TCP/IP-Fähigkeiten wie Unix-Workstations — auch hinsichtlich der Leistung.

Falsch oder ungenügend konfigurierte SMTP-Proxies lassen sich über einen IP/Port-Scanner ausfindig und für Spammer nutzbar machen. Besonders die Windows-Software "AnalogX" hat sich hier unrühmlich hervorgetan. Unter Unix stellen schlecht-konfigurierte Apache Web-Server ein grossen Risiko dar. Mit wenigen Ausnahmen besteht kein Grund darin, einen Web-Server als Proxy zu betreiben. In der Konfigurationsdatei `httpd.conf` ist (wie im Beispiel) folgende Passage auszukommentieren:

```
#
# Proxy Server directives. Uncomment the following lines to
# enable the proxy server:
#
#<IfModule mod_proxy.c>
#    ProxyRequests On
#    <Directory proxy:*>
#        Order deny,allow
#        Deny from all
#        Allow from .your-domain.com
#    </Directory>
```

```
    #
    # Enable/disable the handling of HTTP/1.1 "Via:" headers.
    # ("Full" adds the server version; "Block" removes all
outgoing Via: headers)
    # Set to one of: Off | On | Full | Block
    #
#    ProxyVia On
    #
    # To enable the cache as well, edit and uncomment the
following lines:
    # (no cacheing without CacheRoot)
    #
#    CacheRoot "/usr/local/www/proxy"
#    CacheSize 5
#    CacheGcInterval 4
#    CacheMaxExpire 24
#    CacheLastModifiedFactor 0.1
#    CacheDefaultExpire 1
#    NoCache a-domain.com another-domain.edu joes.garage-
sale.com
#</IfModule>
# End of proxy directives.
```

Ist einmal festgestellt, dass für eine IP-Adresse der Port 25 (SMTP) offen ist, kann sehr einfach getestet werden, ob über diesen Proxy E-Mails relayed werden können.

7.2.2.4 Trojaner, Malicious Code und Backdoors

Bei einigen Web-Servern (z.B. Apache 1.3.12) es möglich, über einen POST-Befehl und einem Buffer-Overflow (z.B. im PHP), Programme ablaufen zu lassen. Hierdurch können auch Spam E-Mails verschickt werden. Dazu kann entweder das Standard-Socket-Interface oder aber — über die Shell — ein Standard-Mail-Client wie **mail** oder **mailx** eingesetzt werden.

Gleiches ermöglichen Trojaner unter Windows, die sich als unverdächtige Programme tarnen und über das Web heruntergeladen werden können (z.B. als Dailer oder Programm zur angeblichen Performance-Optimierung). Die eigentliche Implementierung eines SMTP-Clients bedarf nur weniger Zeilen (Malicious) Code und reduziert sich auch hier auf einen geeigneten Socket-Aufruf.

Besonders problematisch ist hierbei, dass nun die E-Mails aus dem eigentlich geschützten Intranet-Bereicht abgesetzt werden. Dies kann dazu führen kann, dass die Mail-Server der Firma auf einer Open-Relay Liste im Internet auftauchen, obwohl sie nicht die eigentlichen Verursacher sind, sondern nur Vermittler.

7.2.3 Merkmale von Spam E-Mails

7.2.3.1 Aufbau von Spam E-Mails

Das eigentliche Merkmal von Spam E-Mails ist, das sie versucht keine identifizierbaren Merkmale aufzuweisen. SMTP-Absender, SMTP-Sender, Mail-Sender und die Angabe des "Subjects:" werden in einer Spam-Kampagne häufig gewechselt.

Bei mir flatterte im Juli 2003 folgende Spam E-Mail in den Postkasten:

A few moments of your time could save you thousands over the life of your loan! What are you waiting for?

Visit Now

pN2p

Hierbei dient "pN2p" als Serialisierungs-Information. Hinter der Angabe "Visit Now" versteckt sich die URL `http://mtggreat1.com/4/index.asp?RefID=588897`.

Der in X-HTML formulierte Text wurde mit einem erheblichen "Rauschen" aufbereitet, sodass es Systeme schwer haben, das Verhältnis zwischen Nutz- und Spam-Information zu bestimmen:

```
<x-html><HTML>
<!--16--><table align="center" bgcolor="ff2222" width="615"
border="2" cellspacing="12" cellpadding="5"><tr>
<td bgcolor="ffffff" align="center"><font size="4"
face="Arial, Helvetica, sans-serif"><b>A few m<!-AQ-->oments
of your ti<!--tI-->me cou<!--cA-->ld sav<!--Fz-->e y<!--rT--
>ou t<!--NK-->housa<!--DX-->nds o<!--QJ-->ver the l<!--w2--
>ife of your loa<!-dq-->n! W<!--Ju-->hat are y<!--v9-->ou
wai<!--88-->ting for? <!--Le--><br><br><!--Ep--><A
href="http://mtggreat1.com/4/index.asp?RefID=588897"> <font
size="5">Vis<!--Bw-->it N<!--U7-->ow</font></a></font><!--Xs--
> </td><!--54--></tr><br></table><br><br><!--pR--> <br> <HTML>
```

Abschickt wird die Spam E-Mail mit folgenden Merkmalen:

SMTP-Absender	SMTP-Sender (IP)	Mail-Sender	Subject:
elsey@tobe-online.com	61.78.232.181	kelsey <kelsey@tobe-online.com>	Congrats!

jeanine@crisp-mcs.ne	200.83.243.55	jeanine <jeanine@crisp-mcs.net>	Wanna know why?
jeanine@crisp-mcs.ne	61.255.105.123	jeanine <jeanine@crisp-mcs.net>	Don't pass this up

Tabelle 7.1-1: Einige Merkmale einer typischen Spam E-Mail.

Diese "harmlose" Spam E-Mail soll als Lehrbeispiel dienen, dass die Spam-Erzeuger/Sender es mittlerweile gelernt haben:

- Spam E-Mails von real existierenden (temporären) Domänen mit korrekter DNS-Infrastruktur zu versenden,
- die SMTP Envelope-Information gängigen RFCs zu entsprechen,
- den Aufbau des E-Mail-Headers strukturell untadelig zu gestalten,
- die Angabe des "Subjects:" möglichst unverdächtig abzufassen,
- mittels einer in den E-Mail Body eingebauten Serialisierungsinformation eine mögliche MD5-Checksum ins Leere laufen zu lassen und
- den Inhalt der E-Mail so zu optimieren, dass gängige Spam-Klassifizierungsmethoden (SpamAssassin) entweder getäuscht werden oder aber einen geringen "Score" ergeben.

Typischerweise würde trotzdem die E-Mail als Spam klassifiziert werden, da in ihr das Wort "loan" vorkommt — allerdings erst nachdem die HTML-Information entfernt wurde. Die Untersuchung des Textes mittels regulärer Ausdrücke auf den Begriff "loan" würde jedenfalls ins Leere laufen.

Es ist m.E. nur eine Frage der Zeit, bis sich die Spam-Absender an aktuelle (inhaltliche) Filterkriterien angepasst haben. Letztlich kann jede Spam E-Mail mittels eines Durchlaufs an die gerade gültigen SpamAssassin-Version so "getunt" werden, dass sie eine geringe Spam-Rating erzeugt: Hase-und-Igel-Spiel.

7.2.3.2 Fake Opt-In/Opt-Out

Ein Grossteil der Spam E-Mails lockt mit der Möglichkeit eines Opt-Out, d.h. Austragens aus der E-Mail Verteilerliste. Wir betrachten einmal einen Ausschnitt (in X-HTML formatiert) aus einer Spam E-Mail, die die Möglichkeit eines Opt-Out anbietet:

```
You are receiving this message as a member of the Opt-In
America List. To remove your email address please
click here
We honor all remove requests.
```

Die "Opt-Out" Möglichkeit lautet in Klartext:

> *<H4>You are receiving this message as a member of the Opt-In
*
> *America List. To remove your email address please
*
> *<A HREF="http://www.kgbfvkk29y.com@www.youngforever22.com">*
> *click here</A>
We honor all remove requests.</H4>*

Die Spam E-Mail gibt also vor, dass der Empfänger sich per Opt-In in der "America List" eingetragen hat und bietet ein Opt-Out. Die Webseite "www.youngforever22.com" ist tatsächlich aktiv und bietet dort eine "Remove" Möglichkeit, hinter der ein PHP-Skript steckt. Wahrscheinlich wurde die Spam E-Mail im Auftrag verschickt, da der SMTP-Absender `"Return-Path: <nl1jsmnp@msn.com>"` lautet. Die Hinweise auf das angebliche Opt-In und das angebotene Opt-Out sind für diese Spam E-Mail-Aktion offensichtlich "fake" und führen gewollt in die Irre.

Die Nutzung einer Opt-Out Möglichkeit kann daher zu folgenden Konsequenzen führen:

- Verbirgt sich hinter einem Opt-Out Link eine Web-Seite, können darüber zusätzlich Informationen über den "Besucher" ermittelt werden.
- Wird ein "mailto:" Link angeboten, liegt der Verdacht nahe, dass hierüber eine Adress-Harvesting (Verifikation der Empfänger-Adresse) vorgenommen wird.
- Im günstigsten Fall ist die Opt-Out Adresse ungültig (fake) und dient nur vergeblich zum Ausschreiben und um die Legitimität der Spam E-Mail zu wahren.

7.2.3.3 Joe-Jobs

Wie beschrieben, landen E-Mail Adressen (z.B. per Harvesting) häufig auf Spam-Listen, ohne dass der potentielle Empfänger dies verhindern kann. Dies ist unangenehm, aber gegen (passiven) den Empfang unerwünschter E-Mails kann man sich zumindest schützen.

Der umgekehrte Fall ist wesentlich unangenehmer: Die eigene E-Mail Adresse wird zum Versand von Spam E-Mails missbraucht und somit der (angebliche) Absender diskreditiert. Das sog. Faking von E-Mail Adressen in der Return-Path Angabe (`Mail From:`) ist trivial; genauso wie die gleiche Adresse im E-Mail Header als "From:" erscheinen zu lassen.

Der erste bekannte Fall dieses Spoofings betraf "Joes.com"[2] :

> *"The act of faking a Spam sot hat it appears to be from an innocent third*

[2] http://ww.spamfaq.net/terminology.shtml#joe_job

party, in order to damage their reputation and possibly to trick their provider into revoking their Internet access. Named after Joes.com, which was victimized in this way by a spammer some years ago."

Hiergegen gibt es kein probates Mittel. Eine Ausnahme besteht dann, wenn die Spam E-Mail mit der gespooften Absender-Adresse an Empfänger mit der gleichen Domain-Kennung verschickt wird. Mein SPAMCONTROL-Patch (siehe weiter unten) für Qmail bietet hierzu eine sog. Split-Horizon Überprüfung der SMTP Absender-Adressen.

7.2.3.4 Umfang

Die Frage, welchen Umfang Spam E-Mails mittlerweile ausmachen, muss von unterschiedlichen Perspektiven beleuchtet werden:

- Der Anwender, der ein E-Mail-Konto besitzt, wird die Frage dahingehend beantworten, welchen Anteil Spam E-Mails an seinem gesamten E-Mail-Aufkommen ausmachen.
- Für den Betreiber eines E-Mail-Gateways (= ISP) ist nicht nur die Zahl der zugestellten, sondern auch die die Zahl der unzustellbaren Spam E-Mails relevant. Letzere belasten den E-Mail-Server doppelt: Durch die Annahme der E-Mails und durch das Erstellen einer Bounce-Nachricht an den vermeintlichen Absender ("Return-Path:" im E-Mail Header). Im ungünstigsten Fall antwortet das (unschuldige) Zielsystem — was es nicht tun sollte — oder es wird eine (lokale) Double-Bounce-Nachricht an den Postmaster verschickt: Eine nichtzustellbaren Spam E-Mail generiert also in aller Regel zwei weitere. Im Normalbetrieb kein Problem — bei einer Spam-Attacke aber eine ernste Belastung für den E-Mail-Server, speziell, wenn dieser noch Viren-Checks vorzunehmen hat.

In jedem Fall kostet die Übertragung der Spam E-Mail Bandbreite — häufig zu Lasten des Betreibers des E-Mail-Servers. Glücklicherweise sind Spam E-Mails in der Regel relativ klein; was man bei den heute in HTML/X-HTML verfassten E-Mails häufig nicht behaupten kann. Wir bedenken, dass nicht nur im Weg des Empfangs von Spam E-Mail Bandbreite vergeudet wird (incoming), sondern auch durch die Erzeugung von Bounce-Nachrichten (outgoing).

Die Grössenverteilung für meine empfangenen Spam E-Mails findet sich in Abbildung 7.2-1; sie folgt im wesentlichen einer Poisson-Verteilung, wobei Spam E-Mails über 100 kByte selten sind und in der Regel darauf hindeuten, dass eine ganze Webseite verschickt wurde.

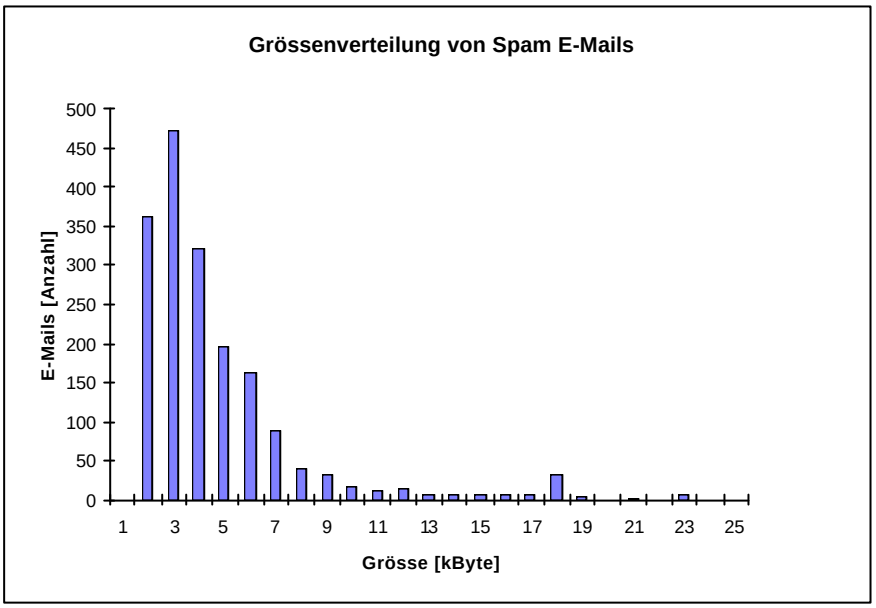

Abbildung 7.2-1: Grössenverteilung der bei mir eingelaufenen Spam E-Mails; der Mittelwert für die Grösse liegt bei etwa 5 kByte.

Zusammengefasst ergibt sich folgendes Bild:

- Die Gesamtzahl der einlaufenden *N* Spam E-Mails stellt volumenmässig einen Bruchteil *X* der Bandbreite für den (incoming) SMTP-Verkehr dar.
- Von diesen *N* Spam E-Mails erreicht ein (kleiner) Teil *n* die Mailbox eines Anwenders (*n/N = 0.1 ... 0.001*).
- *N-n* E-Mails werden als Bounces an den per Return-Path ("Mail From:") ermittelten Absender bzw. den MTA des ISP zurück geschickt. Hierdurch wird ein Bruchteil Teil *Y* der Bandbreite für den (outgoing) SMTP-Verkehr belegt.
- Da die Absenderkennung in der Regel gefälscht ist, wird die Annahme der Bounce-Nachricht von einem Teil der MTA mit einem SMTP-Protokollfehler abgelehnt ("No user/mailbox by that name"). Dies resultiert in *N-n-m* (*m* < *N-n*) Double-Bounces an den lokalen Postmaster.
- In ungünstigen Fällen (aufgrund nicht RFC-konformer MTA oder Autoresponder) ist es möglich, dass auch ein geringer Teil der Bounces zurückgeschickt werden: *N-n-o* (*o* << *N-n*).

Es muss strikt unterschieden werden, zwischen dem insgesamt empfangenen Spam E-Mails für eine Domain und den letztlich an eine Mailbox zugestellten (d.h.

wirksamen), da — je nach Spam-Aktion — der Spammer beliebige Empfänger in einer Domain adressiert, also quasi mit Schrot schiesst.

Insgesamt kann aber heutige (06/2003) das Spam-Aufkommen und zwar hinsichtlich der Anzahl als auch des Volumes zwischen 20% und 25% des gesamten E-Mail-Verkehrs abgeschätzt werden. Vergleichbare Werte ergeben sich auch aus einer Spam-Analyse von Cord Beermann, wie aus den Abbildungen 7.2-2 und 7.2-3 hervorgeht, und dem ich hiermit für sein Zahlenmaterial danken möchte.

Abhängig davon, wie bekannt (oder einfach erratbar) die E-Mail Adresse ist, bzw. auf welchen Adresslisten diese gehandelt ist, kann der "persönliche" Spam-Anteil bis 50% oder gar 80% ausmachen. Es ist klar, dass bei solchen Verhältnissen es keine Freude macht, per E-Mail zu kommunizieren.

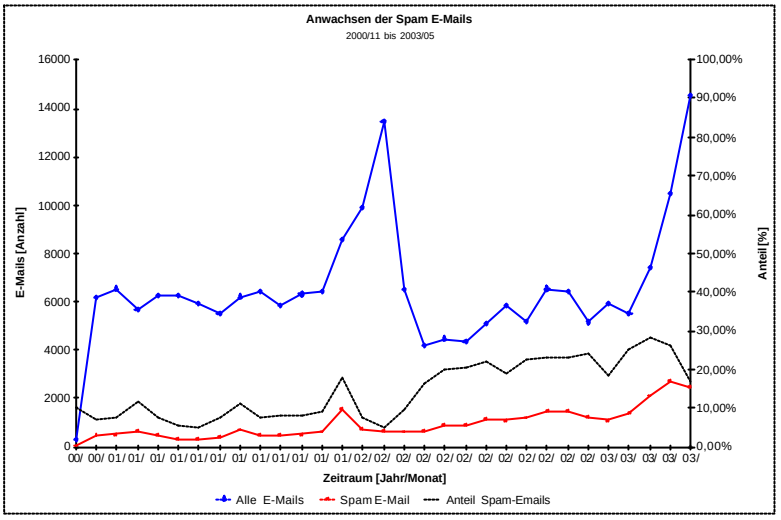

Abbildung 7.2-2: Wachstum der Anzahl der E-Mails und des Spams (Quelle: Cord Beermann)

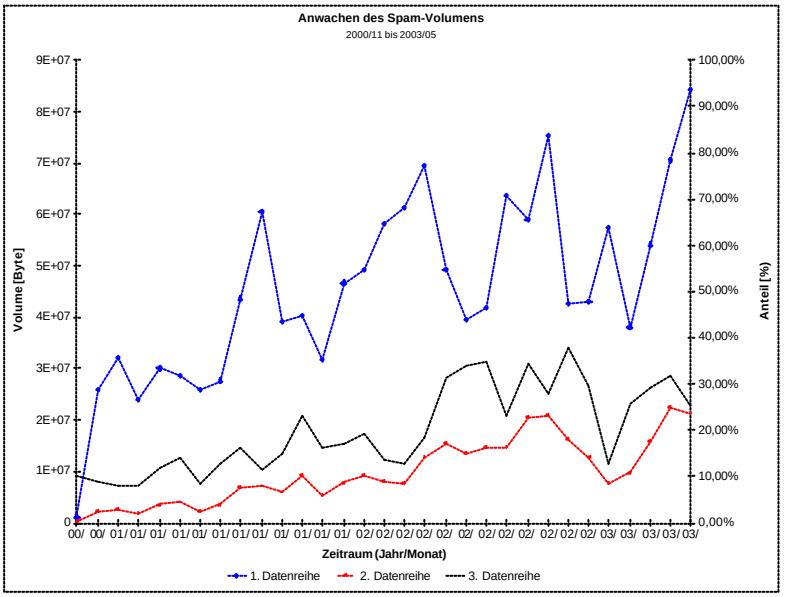

Abbildung 7.2-3: Wachstum des Volumes der E-Mails und des Spams (Quelle: Cord Beermann).

7.2.4 Gegenmassnahmen

Je nach Betroffenheitsgrad und Empfänger gibt es verschiedenste Arten der Reaktionen auf Spam E-Mails: Empörung, Ärger, Resignation, störrisches Wegklicken ...

Unbestritten ist, dass mitunter der erhebliche Spam-Anteil an den regulären E-Mails zur Beeinträchtigung des Arbeitsablaufs von Mitarbeitern wird, deren Aufgabe die Bearbeitung und Erledigung von E-Mails ist. Dies betrifft vor allem solche Mitarbeiter bzw. Abteilungen, deren E-Mail Adresse bekannt, oder relativ leicht erratbar ist. Im besonderen Masse ist dies für öffentliche Institutionen gegeben, deren Auftrag darin besteht, in der (Internet-) Öffentlichkeit Präsent zu sein.

7.2.4.1 Rechtliche Massnahmen

In der Presse sind immer wieder Artikel zu lesen, dass sich grosse Firmen (u.a. Microsoft) und die ISPs gegen Spam E-Mails wenden und Musterklagen gegen die Absender dieser E-Mails anstrengen. Der deutsche Gesetzgeber subsumiert Spam E-Mails unter dem allgemeinen Persönlichkeitsrecht und räumt spezielle Regelungen nur Wettbewerbern und Verbänden ein. Wirksam sind Klagen nur dann, wenn der Beklagte im Inland oder in einem über entsprechende bi- bzw. multilaterale Verträge rechtliche justiziablen Ausland befindet. Dies ist in der Regel nicht der Fall.

Hier nutzt auch das diese Tage (20.7.2003) von der Verbraucherministerin angekündigte Gesetzesvorhaben gegen Spam E-Mails nicht. Meines Erachtens gilt für das rechtliche Vorgehen gegen Spam E-Mails und deren Absender der bereits bei der Bewertung von Dienstaufsichtsbewerden zutreffende Spruch: Fristlos, Formlos, Folgenlos.

ISPs und Unterdrückung von Nachrichten

Internet Service Provider (ISP) müssen sich in der Behandlung der Spam E-Mails auch den Erfordernissen des TKG dar (Telekommunikations Gesetz) sowie des TDSG (Telekommunikation Daten Schutz Gesetz) entsprechen. Sobald Massnahmen des Providers zur Verminderung von Spam E-Mails greifen, macht er sich potentiell der Unterdrückung von Nachrichten schuldig. Ausnahmen hiervon bestehen m.E. jedoch darin, wenn festgestellt wird, dass die E-Mail nicht den gängigen Standards entspricht. Diese Situation ist vergleichbar, wenn ein Brief nicht ordentlich beschriftet oder frankiert ist. Bei Ablehnung erhält der Sender in jedem Fall eine entsprechende SMTP-Mitteilung über die Ursache und kann somit im Zweifelsfall seinen Fehler korrigieren bzw. über den (per E-Mail erreichbaren) Postmaster des Empfängers eine Änderung erwirken.

7.2.4.2 Organisatorische Massnahmen

Jedem von uns flattern mit der normalen Tagespost wöchentlich Dutzende von

Werbesendungen in den Briefkasten ein: Sei es der Pizzabäcker auf Rollschuhen um die Ecke, der Pfarrbrief der Gemeinde, Gewinnspiele, Werbung für Schlankheitskuren, Prospekte der im Umkreis befindlichen Supermärkte ....

Darüber regt sich niemand auf. Gelegentlich liest man beim ein oder anderen Briefkasten "Bitte keine Werbung einwerfen!" - zu einem öffentlichen Thema wird dies nicht, obwohl das Verhältnis von privater Post zu Werbesendungen nicht anders ist als bei Spam zu regulären E-Mails. Warum ist also die Betroffenheit so sehr viel grösser?

Hierfür gibt es m.E. zwei Gründe:

1. Glaube an die Unschuld des Mediums. Bislang wurde das Medium E-Mail als hinreichend sicher und "privat" betrachtet. Versetzen wir und mehr als 100 Jahre zurück, wo die Post noch per Pferdekutschen ausgetragen wurde, hätte auch niemand akzeptiert, dass hierüber Reklame verteilt würde. Wir haben also ein Problem des Umgangs mit dem neuen Medium.
2. Ein grosser Teil der Inhalt der E-Mails ist sexistischer Natur und teilweise so eindeutig, dass sich viele Menschen in ihrer moralischen Einstellung massiv belästigt und aufgewühlt fühlen — was durchaus beabsichtigtes Ziel der Werbung ist.

Betroffenheit

Wir haben es also mit einem menschlichen Betroffenheitsproblem zu tun. In grösseren Firmen/Organisationen mag es hilfreich sein, dem strukturiert zu entgegen:

- *Schulungen*: Interne Seminare über Spam E-Mails mit den in diesem Abschnitt besprochenen Inhalten. Ziel soll sein, statt Resignation zu verbreiten, möglichst die "Lacher" auf seiner Seite zu haben.
- *Abuse-Abteilung*: Schulung und Einweisung geeigneter Mitarbeiter zur Bearbeitung und Verfolgung besonders "gemeiner" Spam E-Mails. Dies ist kein einfacher Job, nimmt aber vielleicht den Druck von den Mitarbeitern, die hiervon besonders betroffen sind.
- *Firmeninterne Spam-Info Webseite*: Hier sollte auf das Problem allgemein eingegangen werden, ggf. auf den Spam E-Mail Account und auf die vom Unternehmen getroffenen organisatorisch/rechtlichen Massnahmen hingewiesen werden (u.a. auf die Abuse-Abteilung). Tröstlich für die betroffenen Mitarbeiter könnte auch ein periodisches Veröffentlichen der abgewiesenen Spam E-Mails sein. Ferner sollten noch einmal der "E-Mail-Codex" der Firma verbindlich vorgestellt werden.
- *Anti-Spam E-Mail-Account*: Einrichtung einer interen E-Mail Adresse, an denen die Mitarbeiter ihre Spam E-Mails weiterleiten können und damit sie diese alsdann löschen (z.B. über ein Regelwerk). Die eingesammelten Spam E-Mails können analysiert und ggf. über die Abuse- oder Rechtsabteilung

weiter verarbeitet werden.

- *Öffentliche Anti-Spam Webseite*: Bei aller Vorsicht und Sorgfalt kann es doch mitunter passieren, dass eine eigentlich unverdächtige und mitunter sogar wichtige E-Mail aufgrund der getroffenen Anti-Spam-Verfahren nicht angenommen wird und der Absender nach Möglichkeiten sucht, ggf. von einer Blocking List gestrichen zu werden. Hierzu ist es hilfreich (eine nicht verzinkte) Anti-Spam Webseite als URL zu referenzieren, die bei der Ablehnung auf IP- oder SMTP-Niveau dem vermeintlichen Spam-Sender mitgeteilt wird (vgl. **rbldns** und **tcpserver** weiter unten). Auf dieser Webseite sollten Kontaktinformationen veröffentlicht sowie die Eingabe von "Complaints" mittels HTML-Forms ermöglicht werden.

7.2.4.3 Vermeidung

Einige weitere wichtige Punkte ergibt sich hinsichtlich der Vermeidung von Spam E-Mails:

- Sofern sich die Mitarbeiter in öffentlichen Diskussionsformen engagieren, sollte darauf geachtet werden, nicht die firmen-eigene E-Mail Adresse dort zu nutzen bzw. bekanntzugeben. Es mag hilfreich sein, den Mitarbeitern nicht nur ein E-Mail Konto, sondern mehrere getrennte zur Verfügung zu stellen, z.B. auch eines zur Abwicklung der privaten Korrespondenz. Dies erlaubt in jedem Fall ein besseres Filtern von evtl. Spam E-Mails.
- Den Mitarbeiten sollte klar gemacht werden, dass auch sie nicht aktiv zur Weitergabe von E-Mail Adressen beitragen sollten. Hierzu zählt vor allem der Verzicht auf die Angabe mehrerer Empfänger von E-Mails in der "To:" oder "CC:" Zeile. Sollen dennoch mehrere Adressaten erreicht werden, lassen sich diese per "BCC:" (*Blind Carbon Copy*) einfügen.
- Bei E-Mail-Adressen, die per Web erreicht werden können, bietet es sich an, mit HTML-Forms zu arbeiten. Sollte aber dennoch eine E-Mail Adresse per "mailto:" bekannt gemacht werden, kann diese z.B. periodisch wechseln, beispielsweise per "adresse-YYYYMM", indem ein Zeitstempel einfügt wird. Qmail erlaubt durch die Vergabe von VERP (*Variable Envelope Response Path*) ein einfach steuerbares Filtern dieser Adressen; ohne dass Absender sofort eine Bounce-Nachricht erhält.

7.2.4.4 Technische Möglichkeiten

Im wesentlichen gibt zwei Ansätze, der Flut der Spam E-Mails entgegenzutreten:

- *Blockieren*, d.h. die E-Mails bereits zum Zeitpunkt der Verbindungsaufnahme entweder auf IP-Niveau oder auf SMTP-(=Anwendungs)-Ebene zu blockieren, sodass keine eigentlichen Nutzdaten übertragen wurden.

- *Filtern*, d.h. Empfang aller E-Mails und anschliessende Analyse der E-Mails, nach dem alten Verfahren: "Die Guten ins Töpfchen, die Schlechten ins Kröpfchen". Hierbei hat man die Wahl, auf die Schlechten — also als Spam identifizierten E-Mails — anschliessend zu reagieren. Soll die Spam E-Mail allerdings erhalten bleiben, setzt dieses Verfahren aber lokalen MUA mehrere adressierbare Mailboxen (oder Mailverzeichnisse) voraus, was beim entfernten Zugriff über IMAP4 realisiert werden kann.
- *Tagging* zeichnet die vorliegende Spam-Merkmale (bzw. den sog. Spam-Level) in einem eigenen E-Mail-Header auf und erlaubt es somit dem Anwender selbst mittels eigener Filterregeln auf die identifizieren Spam E-Mails zu reagieren.
- *Digesting* funktioniert in Abgrenzung zu den anderen Verfahren lediglich auf User-Basis. Der E-Mail-User führt hierbei normalerweise eine sog. Whitelist von Absenderadressen, von denen ohne Einschränkung E-Mails angenommen werden. Jeder andere Sender muss sich zunächst durch eine zusätzliche Reply-E-Mail in Form Message Digest qualifizieren, um auf diese Whitelist zu gelangen.

In der Praxis wird häufig eine Mixtur beider Verfahren angewandt, um ihre Stärken zu kombinieren. Die Problematik, die sich beim Einsatz jeder dieser Methode prinzipiell ergibt, lautet:

- *False Positives*: Reguläre E-Mails werden als Spam eingeschätzt und ggf. verworfen bzw. abgelehnt.
- *False Negatives*: Spam E-Mails werden als normale E-Mails betrachtet und gelangen trotz aller Massnahmen zum Adressaten.

Letzeres ist eine Frage der Effizienz; erstere eine der Schwellwerte und des Verfahrens.

7.2.4.5 Was sagen die RFC?

Obwohl Spam beileibe kein neues Phänomen ist, gibt es in der Sammlung der RFC kaum geeignete Hinweise in Form von "Best Current Practice (BCP)". Der RFC 2505 "Anti-Spam Recommendatations" gibt allerdings einige Empfehlungen zum Umgang mit Spam E-Mails und weist im besonderen folgende Anforderungen an MTAs aus:

1) MUST be able to restrict unauthorized use as Mail Relay.

2) MUST be able to provide "Received:" lines with enough information to make it possible to trace the mail path, despite spammers use forged host names in HELO statements etc.

3) MUST be able to provide local log information that makes it possible to trace the event afterwards.

4) SHOULD be able to log all occurrences of anti-relay/anti-spam actions.

5) SHOULD be able to refuse mail from a host or a group of hosts.

6a) MUST NOT refuse "MAIL From: <>".

6b) MUST NOT refuse "MAIL From: <user@my.local.dom.ain>".

7a) SHOULD be able to refuse mail from a specific "MAIL From:" user, <foo@domain.example>.

7b) SHOULD be able to refuse mail from an entire "MAIL From:" domain <.@domain.example>.*

8) SHOULD be able to limit ("Rate Control") mail flow.

9) SHOULD be able to verify "MAIL From:" domain (using DNS or other means).

10) SHOULD be able to verify <local-part> in outgoing mail.

11) SHOULD be able to control SMTP VRFY and EXPN.

12) SHOULD be able to control SMTP ETRN.

13) MUST be able to configure to provide different Return Codes for different rules (e.g. 451 Temp Fail vs 550 Fatal Error).

In Abschnitt 1.5 des RFC wird die Frage gestellt "Where to block spam, in SMTP, in RFC822 or in the UA", endet aber lediglich mit dem Verweis, dass Spam E-Mails während des SMTP-Dialogs abzuweisen sind.

Leider ist die Situation in der Praxis noch vertrackter:

- Aus den SMTP-RFC lassen sich keine Berechtigungen ableiten, E-Mails aufgrund irgendwelcher Kriterien zu blockieren. Dies gilt insbesondere für den Return-Path ("Mail From:") und die HELO/EHLO-Begrüssung des SMTP-Client. Dies gilt sowohl für die SMTP-Envelope, wie den E-Mail Header, als auch natürlich für den E-Mail-Body. Entsprechende Formulierungen in den RFC 2821/2822 lauten in der Regel "Should".
- Im Gegenzug, gibt es keine "Must" oder "Shall" Kriterien, auf die die E-Mail des Absenders überprüft werden muss. Der SMTP-Server muss so tolerant wie möglich sein. Qmail z.B. lehnt aus sich heraus nur E-Mails mit einem einfachen LF (statt CRLF) ab.
- Beim Einsatz "scharfer" Blockade-Massnahmen für Spam, kann es leicht passieren, dass — aufgrund der "False Negatives" — der Provider auf der

Liste der RFC-Ignorant[3] gesetzt wird; kein sonderliches Vergnügen und auf jedenfalls der Reputation schädlich.

- Die Blockade-Massnahmen bewegen sich somit in einer Grauzone; dies ist wohl mit ein Grund, warum einige grosse Freemail-Anbieter sich auf das Filtern von Spam E-Mails beschränken.

Insgesamt ist die Internet-Community ziemlich unschlüssig, wie mit Spam E-Mails umzugehen ist. Die Bandbreite schwankt zwischen grosser Toleranz beim Empfang selbst (SMTP-)syntaktische unzulänglicher E-Mails bis zu einer Ablehnung der Empfang von E-Mails von Dial-In SMTP-Clients. RFC 1123 "Requirements for Internet Hosts -- Application and Support" sagt z.B. hinsichtlich des HELO Parameters (5.2.6):

> *"The sender-SMTP MUST ensure that the <domain> parameter in a HELO command is a valid principal host domain name for the client host. As a result, the receiver-SMTP will not have to perform MX resolution on this name in order to validate the HELO parameter. The HELO receiver MAY verify that the HELO parameter really corresponds to the IP address of the sender. However, the receiver MUST NOT refuse to accept a message, even if the sender's HELO command fails verification.*
>
> *DISCUSSION:*
>
> *Verifying the HELO parameter requires a domain name lookup and may therefore take considerable time. An alternative tool for tracking bogus mail sources is suggested below (see "DATA Command"). Note also that the HELO argument is still required to have valid <domain> syntax, since it will appear in a Received: line; otherwise, a 501 error is to be sent."*

Dieses "MUST" Kriterium für SMTP-Clients ist unter den heutigen Bedingungen von vorwiegend temporären Internet-Dial-Up Verbindungen schlicht hinfällig.

7.2.5 Spam E-Mails im Verarbeitungszyklus

Zur Entscheidung ob eine E-Mail Spam ist oder nicht, bedarf natürlich hinreichender Informationen. Je nach Kenntnissstand kann die Spam E-Mail zu unterschiedlichen Verarbeitungszeitpunkten sehr differenziert behandelt werden, was nachfolgende Tabelle in einem Gesamtüberblick liefert:

Zeitpunkt	Schicht/(Schritt)	Informationen	Massnahmen

[3] www.rfc-ignorant.org

Zeitpunkt	Schicht/(Schritt)	Informationen	Massnahmen
Aufbau der TCP/IP-Verbindung	IP (Layer 3)	Sender-IP, FQDN des Senders, Blocking List	Blockieren [per MTA]
SMTP-Dialog	SMTP (Layer 7)	Return-Path (MAIL From:), Forwarding-Path (RCPT To:) MX der Return-Path Domain, HELO/EHLO Angabe	Blockieren (SMTP Fehlercode 5xx), Deferral (Verzögern, SMTP Fehlercode 4xx) [per MTA]
Einfügen in die Queue	(Verarbeitung)	E-Mail-Header, E-Mail-Body (Inhalt/Text), MIME-Struktur, Footprint (remote)	Bounce, Verwerfen, Taggen [per MTA]
Ausliefern in Empfänger-Mailbox	(lokale Zustellung)	E-Mail-Header, E-Mail-Body (Inhalt/Text), MIME-Struktur, Footprint (remote); evtl. Spam-Tags	Bounce, Verwerfen, Ablage "Junkmail" Ordner, Taggen [per User]
Abholung durch Remote Mail User Agent	(entfernte Zustellung)	E-Mail-Header, E-Mail-Body (Inhalt/Text), MIME-Struktur, Footprint (remote); evtl. Spam-Tags	Abholen, Nicht-Abholen, Löschen nach *N* Tagen, lokaler "Junkmail" Ordner [per User]

Tabelle 7.2-1: Informationen über E-Mails und Möglichkeiten zur Behandlung von Spam E-Mails im Verarbeitungszyklus.

Bei der Ablehnung von E-Mails auf IP-Schicht oder im Laufe des SMTP-Dialogs handelt es sich mithin um eine ***Blockierung***. Ist die E-Mail einmal in Empfang genommen, kann sie nach unterschiedlichen Kriterien einem oder mehreren ***Filtern*** unterworfen werden. Dies resultiert in einer — je nach Resultat — unterschiedlichen Verarbeitung der E-Mails (*aktives Filter*). Im Gegensatz hierzu wird beim ***Tagging*** lediglich das Filterergebnis beispielsweise in der zugefügten E-Mail Header-Zeile vermerkt (*passives Filter*) und die Aufgabe des aktiven Filterns den anschliessend ablaufenden Programmen überlassen.

Beachtenswert ist insbesondere auch, dass das Blockieren auf Grundlage MTA-

spezifischer Regeln erfolgt, währenddessen das Filtern sowohl "global", d.h. MTA- wie auch benutzerspezifisch vorgenommen werden kann.

Exemplarisch ist der Verarbeitungszyklus für Qmail in Abbildung 7.2-4 darstellt. Zusätzlich ist es möglich, die auf Schicht 3 (IP) und 7 (SMTP) ebenfalls in einem zusätzlich eingefügten E-Mail-Header Feld zu hinterlegen, und so für die spätere Auswertung verfügbar zu machen.

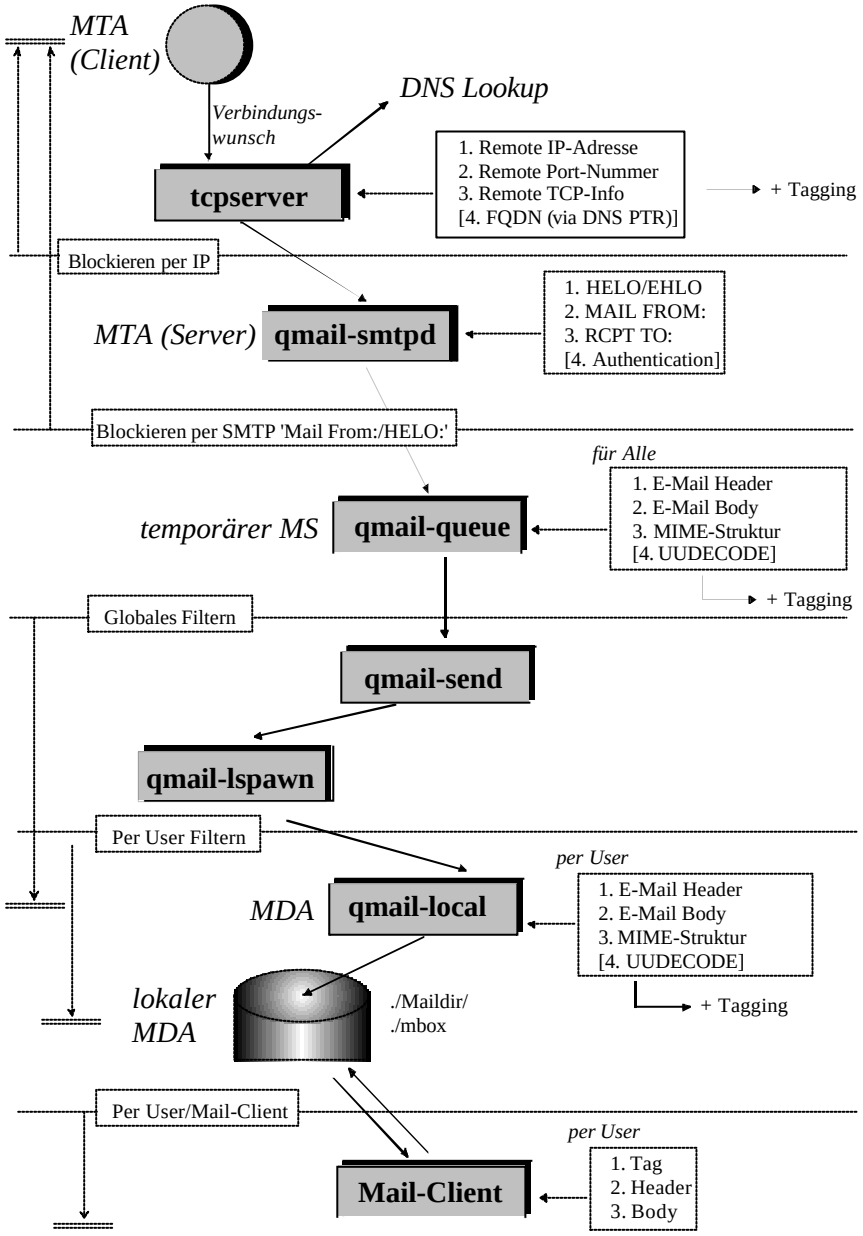

Abbildung 7.2-4: Mögliche Schritte der Verarbeitung von Spam E-Mails bei Qmail

7.2.5.1 Blockieren

Das Blockieren von E-Mails kann auf IP-Schicht und/oder auf SMTP-Schicht erfolgen; wem das OSI-Referenzmodell vertraut ist, wird die IP-Schicht mit dem Netzwerk (Schicht 3), SMTP mit der Anwendungsschicht (Sicht 7) identifizieren. Folgende Vorteile ergeben sich beim Blockieren von Spam E-Mails:

- Beim Blockieren einer E-Mail erhält der Absender in der Regel eine qualifizierte Mitteilung hinsichtlich der Ablehnung.
- Das Verfahren ist unabhängig davon, ob der Empfänger auch tatsächlich existiert, da es ausschliesslich Absender-spezfisch erfolgt.
- Da nur sehr wenig Protokollinformation zur Feststellung benötigt wird, erfolgt die Ablehnung zum frühest möglichen Zeitpunkt; er wird nicht die eigentliche Mitteilung übertragen und somit Bandbreite gespart.

Das Verfahren hat aber auch einige Nachteile:

- Das Blockieren arbeitet MTA- und nicht Benutzer- (=Empfänger-)spezifisch.
- Erfolgt die Ablehnung bereits auf IP-Niveau, können auch legitime Absender davon betroffen sein. So haben es z.B. einige Firmen zu eigen gemacht, keine E-Mails von Absendern mit Dial-In IP-Adressen in Empfang zu nehmen. Um zu diesen E-Mails zu übertragen, muss man — per SMPTROUTE — notgedrungen über das E-Mail-Relay des Providers gehen.
- Es kann nicht garantiert werden, dass der initiale Sender die Information über die Ablehnung erhält. Dies kann dann der Fall sein, wenn er über ein SMTP-Relay seine E-Mail verschickt.

tcpserver

Der effizienteste Weg zur Abwehr von Spam E-Mails ist deren Ablehnung auf IP-Niveau bei Verbindungsaufnahme:

Sender (IP; TCP-Port) ==> Empfänger (IP; TCP-Port=25)

Zudem wird häufig über einen DNS-Lookup (`tcpserver -h`) während des Verbindungsversuchs der FQDN des Sender ermittelt (ANAME). Es ist möglich, die Verbindungsaufnahme mittels **tcpserver** unter den folgenden Szenarien zu blockieren:

- Die IP-Adresse, bzw. der IP-Adressbereich wird blockiert:

```
61.255.105.123:deny   # deny, if from that IP
210.10.10.:deny       # deny, if form that net
```

- Mit dem **tcpserver** Flag "Paranoid" lässt sich ferner die Konsistenz von A- und RTP-RR überprüfen (`tcpserver -p`), indem zusätzlich über den FQDN und die Abfrage des RTP-Records erneut die IP-Adresse ermittelt wird. Bei einem Missmatch wird dann die Environment-Variable $TCPREMOTEHOST gelöscht (bei nicht feststellbaren FQDN stellt Qmail

dann den Namen als "unknown" dar). Das **tcpserver**-Regelwerk hierzu lautet:

```
=:allow  # accept, if $TCPREMOTEHOST is set
:deny    # deny all other connections
```

- Die IP-Adresse ist eine temporäre Dial-In Adresse; der Provider mappt diese häufig unter dem Namen "dial" oder mittels ihres Reverse-Namens. T-Online Dial-Up Sender und solche mit einem Reverse-Namen können über das **tcpserver**-Regelwerk wie folgt abgewiesen werden:

  ```
  =.in-addr.arpa:deny  # disable, if reverse IP name
  =.t-dialin-net:deny  # disable, if from t-online
  ```

  Mittels des Gleichheitszeichens "=" wird beim Aufbau bzw. Auswertung der `tcpserver` cdb kenntlich gemacht, dass nicht die IP-Adresse (via der Environment-Variablen `$TCPREMOTEIP`), sondern der FQDN (entsprechend `$TCPREMOTEHOST`) genommen werden soll, wobei im Fall der IP-Adresse die Auswertung von Rechts-nach-Links und bei den Namen von Links-nach-Rechts erfolgt.

- Die mögliche Ausgabe eines sog. *Banner* durch **tcpserver** bei der Verbindungsaufnahme (`tcpserver -B "No Spam"`) ist in der Regel nicht hilfreich, um dem SMTP-Client einen Hinweis auf seine Ablehnung zu geben; dies leistet aber — wie weiter unten dargestellt — **rblsmtpd** zusammen mit **tcpserver**.

Real-time Blocking List (RBL)

Soweit mir bekannt, stammt die Idee, die Entgegennahmen von E-Mails zu verweigern, die von "offenen" MTAs kommen, oder aktiv an der Weitergabe von Spam E-Mails beteiligt sind von Paul Vixie. Dieser rief Mitte der 90'er Jahre das Projekt *Mail Abuse Protection System* MAPS ins Leben, in dem per DNS solche MTAs aufgelistet sind und in Echtzeit — während Mailverarbeitung — ausgewertet werden können, was somit als *Real-time Blocking List* (RBL) bezeichnet wird.

MAPS ist zwischenzeitlich nicht mehr aktiv und Organisation wie ORBS, SpamCop, ORDB sowie viele andere realisieren heute vergleichbare Dienste. Der Name MAPS wird zwischenzeitlich vom Nachfolger *Mail Abuse Prevention System LLC* MAPS[SM][4] verwendet. Alle RBL-Anbieter gemeinsam ist, dass sie sich in einer Grauzone des Internet befinden. Niemand kann garantieren, dass die aufgenommen Server auch wirklich offene Relays sind, der Test vielleicht nicht sinnvoll ausfällt oder gar der Anbieter auf einen "Joe-Jobs" hereingefallen ist. Firmen — speziell Internet-Anbieter und FreeMailer — führen daher mitunter einen rechtlichen Kampf gegen die Blocking List-Betreiber, mit der Folge, dass die Dienste schnell verschwinden und genauso schnell unter neuer Bezeichnung und

[4] `www.rfc-ignorant.org`

Lokation ihren Dienst wieder aufnehmen.

Der "Scope" der Blocklisten-Betreiber hat sich mittlerweile enorm erweitert und betrifft nicht mehr ausschliesslich den SMTP-Datenverkehr. Der RBL-Dienst kann effektiv von beliebigen TCP/IP-Anwendungen genutzt sind. So finden sich in einigen Blocklisten ganze Netzwerke, die somit vom Internet-Datenverkehr "gebannt" werden können.

Dial Up User List (DUL)

Eine Spielart der Realtime Blocking List stellt die DUL (*Dial Up User List*) Datenbank dar. Während in der RBL "bekannte" MTAs mit festen Adressen auftauchen, landen in der DUL die von Providern dynamisch vergebenen IP-Adressen. Hinter den eingetragenen IP-Adressen finden sich nicht notwendigerweise Spam-Sender; der Nutzer des DUL-Dienstes sieht aber diese IP-Adressen als "unerwünscht" an, um eine SMTP-Verbindung mit seinem Server aufzunehmen.

Wie bei allen RBL-Diensten gilt es auch bei der Nutzung der DUL, den Anbieter sehr genau zu eruieren und sich mit dessen "Policy" vertraut zu mache; speziell wenn Problemfälle auftreten.

Blocking List-Server

Das Verfahren, nachdem die RBL aufgebaut sind, ist sehr effizient: Der Betreiber der Blocking List stellt einen authoritive Domain-Server für eine High-Level Domain zur Verfügung, auf dem ein spezieller Name-Server läuft (z.B. **rbldns** aus den DJBDNS-Package von Dan Bernstein).

- Jedes erkannte offene Relay wird über einen eigenen DNS TXT-Record (RFC 1035) eingetragen, indem die umgekehrte [a.b.c.d => d.c.b.a] IP-Adresse mit dem FQDN des Blocking List-Servers konkatiniert und unter diesem veröffentlicht wird. Inhalt des TXT-Records im positiven Fall üblicherweise ein erklärender Text sowie ein Verweis (URL) auf die Webseite des Blocking List-Betreibers; liegt kein entsprechender TXT-Record vor, bleibt die Antwort "leer".
- Wird nun ein DNS TXT-Lookup (z.B. mit dem Programm **dnstext** aus DJBDNS) auf diesen Namen vorgenommen, erhält man eine entsprechende Antwort; z.B. für den ORBD-Diensts mit dem FQDN `relays.ordb.org` und der Testaddresse 127.0.0.2:

```
dnstxt 2.0.0.127.relays.ordb.org
Listed by ORDB - for testing purposes only
```

Diese Vorgehen kann als ***Pre-Initial-Spam-Verfahren*** bezeichnet werden, wobei in jedem Fall ein spezielles Dienstprogramm wie **rblsmtpd** eingesetzt werden muss, auf das im Anschluss eingegangen wird.

Die Betreiber der Blocking List-Server[5] haben unterschiedliche Strategien:

- **Scanning**: Die im Internet per MX-Record eingetragenen Mail-Server werden daraufhin überprüft, ob sie sich ggf. wie ein offenes Relay Verhalten.
- **Testing**: Über eine Web-Interface des Betreibers ist es in der Regel möglich, selbst z.B. den eigenen MTA zu testen.
- **Aufnahme**: Über Complaints von Anwendern können Mailserver auf diese Liste gesetzt werden.
- **Löschen**: Nach dem Absichern eines MTA muss es natürlich möglich sein, den Eintrag kurzfristig zu entfernen.

rblsmtpd

Mittels des Programms **rblsmtp** aus dem UCSPI-Paket von Dan Bernstein kann ein Lookup mehrerer RBL-Listen vorgenommen werden. Ein typischer Aufruf unter **tcpserver** und beim Einsatz von Qmail lautet (vgl. Abbildung 7.2-5):

```
exec softlimit -m 2000000 \
       tcpserver -vRh -l $HOSTNAME \
        -x /var/qmail/etc/locals.cdb \
       -u $QMAILDUID -g $QMAILDGID 0 smtp \
       rblsmtpd -C -b -r relays.ordb.org \
       /var/qmail/bin/qmail-smtpd 2>&1
```

Mit dem Aufruf `rblsmtpd -C -B -r relays.ordb.org` wird folgendes erreicht:

- *-C*: Fail-Open Mode; falls der Dienst im DNS temporär nicht erreicht ist, wird die Verbindung nicht abgewiesen (ansonsten *-c*),
- *-b*: Ausgabe des SMTP-Fehler Codes '553' (permanente Ablehnung) statt wie per Default (*-B*) '451', d.h. lediglich deferral.
- *-r relays.ordb.org*: Angabe des FQDN der RBL-Quelle, wobei sich mehrere Quellen auflisten lassen (falls *-a FQDN* eintragen wird, handelt es sich nicht um eine Blocking List, sondern um eine Whitelist).

[5] vgl. http://www.geocities.com/spamresources/filter-dnsbl.htm

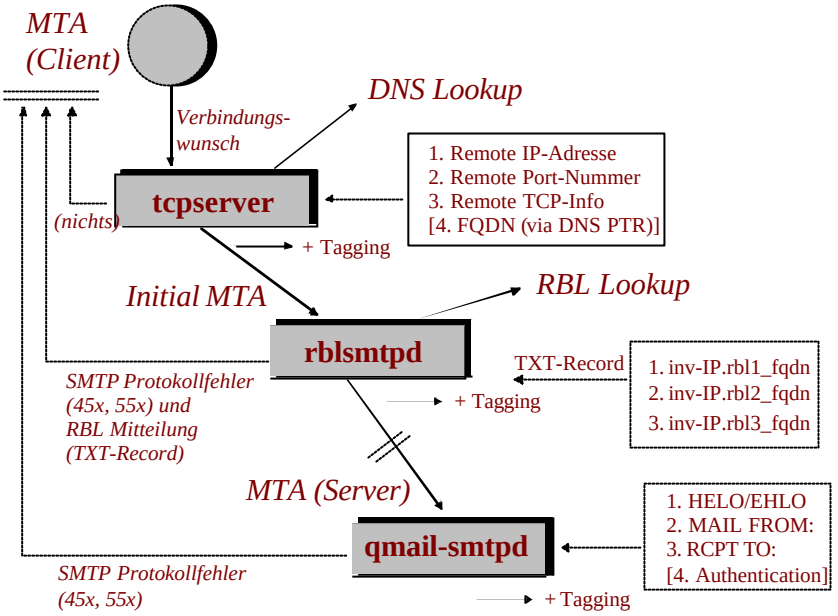

*Abbildung 7.2-5: Zusammenspiel von **tcpserver**, **rblsmtpd** und **qmail-smtpd***

rblsmtpd verhält sich wie ein "normaler" SMTP-Server, d.h. er führt den ersten Schritt des SMTP-Dialogs durch, indem er dem Client zunächst mit `220 rblsmtpd.local` antwortet und parallel hierzu einen DNS-Lookup in der angegebenen Blocking List vornimmt. Per Default, d.h. ohne Angabe einer expliziten RBL-Base, nimmt **rblsmtpd** einen Lookup bei `http://maps.vix.com/rbl/` vor; einer nicht mehr existenten RBL.

Abhängig vom Resultat des Lookups verhält sich **rblsmtpd** wie folgt:

- Die Environment-Variable `$RBLSMTPD` ist nicht gesetzt, die Antwort war also negativ, d.h. der Sender ist nicht in der RBL vorhanden, startet **rblsmtpd** den eigentlichen SMTP-Sever, z.B. **qmail-smtpd**.
- Ist die Antwort positiv, wird die Environment-Variable $RBLSMTP gesetzt, bzw. mit der Antwort der RBL befüllt. **rblsmtpd** führt anschliessend den SMTP-Dialog bis zum "RCPT TO: <Forwarding-Path>" fort und gibt erst dann einen SMTP-Fehlercode zurück. Hierbei wird die Information dem SMTP-Client zurückgegeben, die als Inhalt des DNS-TXT Records eingetragen ist bzw. nun in `$RBLSMTPD` steht. Beispiel:

```
220 rblsmtpd.local
helo me
```

```
250 rblsmtpd.local
mail from: <spamtest@example.com>
250 rblsmtpd.local
rcpt to: <erwin>
451 Blacklisted IP address 192.168.192.3; see
http://www.fehcom.de/qmail/spaminfo.html
```

- Ein Spezialfall liegt vor, wenn der Inhalt der Environment-Variable `$RBLSMTP` mit einem Bindestrich "`-`" beginnt. In diesem Fall gibt **rblsmtpd** immer eine '553' SMTP-Fehlermeldung an den Client zurück, sodass der Absender eine sofortige Bounce-Nachricht erhält.

Lokale RBL

Eine elegante Möglichkeit, sich der vermeintlichen Autorität von RBL-Betreibern zu entsagen, ist der Aufbau eines eigenen (unternehmensweiten) RBL-Servers. Notwendig sind folgende Instrumente:

1. Eine eigener lokaler RBL-Dienste, z.B. der bereits erwähnte **rbldns** von Dan Bernstein. Dieser muss als Nameserver im lokalen DNS den Clients (=MTAs, die auf diesen Dienst zugreifen) im allgemeinen DNS der Firma bekannt gemacht werden.
2. MTAs, die z.B. via **tcpserver** und **rblsmtpd** auf die Dienste dieses lokalen **rblsdns** lauschen.
3. Eine möglichst automatische Registrierung von Spam E-Mails mit der Möglichkeit, die IP-Adresse des Absenders in die **rbldns**-Datenbank unverzüglich einzustellen.
4. Gegebenenfalls kann es sich als nützlich erweisen, für das Unternehmen eine eigene "Abuse" Web-Seite zu veröffentlichen, die in der **rblsmtpd**-Mitteilung als URL referenziert wird.

In der Regel läuft der **rbldns** auf dem MTA, auf dem auch die Spam-Analyse durchgeführt wird, wodurch sich dieses Konstrukt auch für alleinstehende Server eignet. Die Arbeitsweise eines lokalen RBL-Dienstes ist verblüffend einfach:

Wird eine einlaufende Spam E-Mail als solche erkannt, erfolgt die sofortige Aufnahme der IP-Adresse des sendenden MTAs in die RBL, sodass alle weiteren E-Mails von diesem Absender (bis zum Zeitpunkt eines möglichen Entfernens - *Agings*) blockiert werden. Von den in einer typischen Kampagne einlaufenden mehreren tausend Spam E-Mails erreicht somit im Idealfall nur eine einzige ihr Ziel; alle anderen werden abgewiesen. Kombinieren kann man diese Methode mit speziell präparierten E-Mail-Adressen auf den öffentlichen Web-Seiten der Firma, die dann als funktionales Spam-Ziel fungiert.

Dieses ***Post-Initial-Spam-Verfahren*** ist effektiv und zuverlässig. Da **rblsmtpd** zudem eine sinnvolle Mitteilung an den sendenden MTA abgegeben kann, ist diese Methode auch ziemlich sicher hinsichtlich "False Negatives". Wird zudem ein Aging der IP-Adressen eingesetzt und als SMTP-Fehlercode — wie per

Default beim **rblsmtpd** vorgesehen — ein SMTP '451' Fehlercode ausgegeben, haben unschuldig betroffene Absender kein Nachsehen, da mittels der üblichen Queuing-Massnahmen die E-Mail zu einem späteren Zeitpunkt erfolgreich zugestellt werden kann, oder aber der betroffene Absender im schlechtesten Fall eine "böse" E-Mail an den Postmaster (z.B. per Abuse Web-Seite mitgeteilt) verschickt (dessen E-Mail-Adresse sich natürlich dann häufig ändern sollte).

SMTP

Das sicherlich gängigste Mittel zur Spam-Bekämpfung ist die Blockieren auf SMTP-Niveau durch eine der folgenden Massnahmen:

- Der SMTP-Absender ist auf einer Blocking Liste — bei Qmail `badmailfrom`.
- Der SMTP-Absender (genauer: die Return-Path Angabe im MAIL FROM: <Return-Path>) kann per DNS nicht aufgelöst werden; entweder weil der Domain-Teil der Adresse falsch geschrieben wurde, nicht-zulässige Zeichen enthält (z.B. Leerzeichen), die Domain nicht existiert oder für die Domain kein MX-Eintrag vorliegt, sodass ein evtl. Bounce nicht zustellbar sind.
- Im HELO/EHLO-Statement wird keine oder keine gültige FQDN-Adresse angegeben, bzw. diese kann via DNS-Lookup nicht ermittelt werden.
- Als Tarpitting wird eine zusätzliche Massnahme bezeichnet, die dann greift, wenn der Absender E-Mails an mehrere Absender "RCPT To:" absetzen will. Hierbei wird ein Zähler geführt der bei Überschreitung eine Verzögerung (sleep) in der Entgegennahme des nächsten "RCPT To:" einführt. Hierdurch wird der Absender gezwungen zu warten und es erscheint ihm, als wäre er in ein "Teerfass" gefallen.

Diese Verfahren funktionieren sehr gut, sofern die Spam-Erzeuger/Sender das SMTP-Protokoll nicht richtig verstanden haben. Häufig wird z.B. in Ermangelung einer geeigneten HELO/EHLO-Kennung die (einfach zu ermittelnde) IP-Adresse des Empfangssystems als SMTP-Begrüssung benutzt: Bingo.

Allerdings wird keines dieser praktikablen Verfahren von gängigen RFCs gedeckt; sie sind im Grunde nicht zulässig. Dies gilt umsomehr, wenn im Zuge eines auf diese Weise dem Spam-Sender zugetragenen SMTP '5xx' Fehlercodes zusätzlich die Verbindung gekappt wird.

SPAMCONTROL

Das von mir bereitgestellte SPAMCONTROL-Patch[6] für Qmail vereinigt die meisten der gängigen Anti-Spam-Verfahren auf SMTP-Niveau. Zudem ermöglicht es ein qualifiziertes Logins, falls eine Filtermassnahme greift und kann so sehr schnell zur Kontrolle der Spam-Aktivität bzw. der Effizienz eingesetzt werden:

- Filtern auf die "MAIL From:" (Return-Path) Adresse mit Wildcards.

[6] http:/www.fehcom.de/qmail/spamcontrol.html

- Filtern auf die "RCPT To:" (Forwarding-Path) Adresse mit Wildcards.
- Filtern auf die HELO/EHLO Angabe mit Wildcards.
- Überprüfung der Absenderangabe "MAIL From:" im DNS.
- Überprüfung der Hostangabe in der HELO/EHLO Mitteilung im DNS.
- Tarpitting mit variablen Grenzwerten.

Als weiteres nützliches Feature arbeiten die meisten Filter im *Split-Horizon* Verfahren. D.h. es wird eine Fallunterscheidung gemacht, ob ein Absender von einem vertrauenswürdigen Sende-Host die E-Mail verschickt oder nicht. Bei Qmail macht sich dies durch das Setzen der Environment-Variablen $RELAYCLIENT fest; ist diese gesetzt kann der Absender E-Mails beliebig verschicken, ansonsten nur an die per `rcpthosts/morercpthsots` deklarierten Domains. Hierdurch werden (intern wie extern) E-Mails mit gespooften Absenderadressen erkannt und ihre Weiterleitung unterbunden.

7.2.5.2 Strukturelle Analyse

Zur strukturellen Analyse von E-Mails müssen diese daher natürlich angenommen worden sein. Die Analyse kann zu folgenden Zeitpunkten stattfinden:

1. Unmittelbar nach dem Entgegennahme durch den SMTP-Server (**qmail-smtpd**) on-the-Fly beim Einstellen in die Queue.
2. Während der lokalen Zustellung durch den Mail Delivery Agent MDA (**qmail-local**).
3. Nach dem Herunterladen der E-Mail über das Protokoll POP3/IMAP4 als Plug-In für den lokalen Mail User Agent MUA (Outlook, The Bat, Eudora, Netscape ...).

Die ersten beiden Verfahren arbeiten also Empfangs-System basiert und sind stets dann notwendig, wenn der Empfänger seine E-Mail nicht ausschliesslich über einen lokalen MUA, sondern speziell über ein Web-Interface Zugriff erhält. Alle potentiellen E-Mail Empfänger werden identisch behandelt, d.h. die strukturelle Analyse erfolgt mittels einer einheitlichen Engine (MTA-basierend), wobei im zweiten Fall zusätzlich benutzerspezifische Kriterien einfliessen können.

Das letzte Verfahren ist natürlich am Ressourcen-schonendsten, da die Analyse nun auf den Clients stattfinden. Viele E-Mail Programme bieten Plug-Ins für Anti-Spam Programme an, wobei die meisten von kommerziellen Anbietern bereit gestellt, lizensiert und natürlich regelmässig ergänzt werden müssen.

Verfahren

Bei der strukturellen Analyse können zwei Verfahren unterschieden werden:

- *Formale Analyse*: Die E-Mail wird hinsichtlich des Aufbaus und Ausgestaltung des E-Header untersucht und dies mit dem vorliegenden

Mitteilungsinhalt — dem MIME-Content — korreliert. Die Erfahrung sagt, dass die meisten Spam-Erzeuger versuchen, die E-Mail wie eine "übliche" — z.B. von Outlook generierte — Nachricht aussehen zu lassen, aber dann doch Fehler einbauen.
Als weitere Kriterien lassen sich z.B. die verwendeten X-HTML-Attributen nutzen. Spam E-Mails wollen in der Regel so auffällig wie möglich sein. Ferner kann auch eine eingebettete URL auf eine Spam E-Mail deuten.

- *Inhaltliche Analyse*: Die inhaltliche Analyse besteht darin, die Bedeutung der verwendeten Worte zu erfassen (nachdem sie von eventuellen (X-HTML-) Datenmüll gesäubert wurden und gegen die gespeicherte typische Spam-Terminologie abzugleichen. Damit Worte wie "loan", "Viagra", "Erektion" nicht auch in E-Mails der damit beschäftigten Berufsgruppen als Spam identifiziert werden, bieten die meisten Systeme die Möglichkeit, spezielle Negativ- oder Positvlisten von Begriffen zu führen. Zudem wird in der Regel eine sog. Bayesean-Bewertung vorgenommen. Hierbei wird das Verhältnis von gefundenen Spam-Begriffen zu den unverdächtigen Begriffen gebildet und hiermit ein Rating durchgeführt.

Die Aufgabe der inhaltlichen Analyse ist daher sehr komplex. Ein entsprechendes System muss zum einem "Language-Aware" sein, als auch dafür Sorge tragen, dass sich die Entscheidungen nicht gegenseitig nihilieren. Hierfür gibt es Konzepte in Form sog. Entscheidungsbäume, die sicherlich in der nächsten Generation der Anti-Spam-Tools Einzug halten.

Ein weiteres Problem stellt sich, wenn in Zukunft die Aussagen der Spam E-Mails nicht in Form Text sondern z.B. einfach als eingebettete Graphiken erfolgt: Wer wollte einen Spam-Versender/Erzeuger daran hindern? Solange E-Mail Clients wie Outlook für ein eingebettetes Objekt einfach den dazu gehörigen Viewer öffnen, ist reichlich Spielraum für effektive Varianten geboten.

Alle ***Post-Receipt-Spam*** Verfahren haben folgende Nachteile:

- Eine komplexe und CPU-intensive Verarbeitung der empfangenen E-Mail ist notwendig.
- Die Verarbeitungskriterien müssen laufend dem Know-How der Spammer angepasst werden.
- Der Empfänger muss trotz alle dem gelegentlich in sein "Junk-Mail" Verzeichnis schauen, ob sich nicht etwa doch False-Positives hierin befinden.
- Der Spammer kann sich die (virtuellen) Hände reiben: Seine Spam E-Mail ist zunächst vom Zielsystem entgegengenommen worden; erstes Etappenziel erreicht.

Die Vorteile dieser Verfahren besteht allerdings darin, dass

- der Empfänger entscheiden kann, ob eine E-Mail Spam ist oder nicht,
- er in der Regel selbst Einfluss auf die Entscheidungskriterien hat,
- unter Verzicht auf zusätzliche Blockierungsmassnahmen jede E-Mail den Empfänger erreicht.

Abschliessend wollen wir uns eine weitere typische E-Mail betrachten, die vom Programm Spamnix[7] analysiert und als Spam klassifiziert wurde:

```
Delivered-To: erwin@localhost
From: "Lee Baxter" <ejxhot4kwjp3@copper.net>
To: <feh@fehcom.de>
Subject: You can order Anti-depressants, weight loss meds,and
pain relief meds online with NO PRESCRIPTION cw
Date: Sun, 15 Jun 03 22:53:18 GMT
X-Mailer: Microsoft Outlook Express 5.00.2615.200
X-MSMail-Priority: High
Content-Type: text/html;


 ATTENTION:

New Viagra Soft Tab Works In 15 Mins or Less!

 Compounded from FDA Approved Vi(a)graÿFFFFAE

 ORDER WITH NO PRESCRIPTION DISCREETLY ONLINE


 Do It For Her!


 Sildenafil Citrate soft tabs are compounded by licensed pharmacists using
FDA Approved Vi(a)graÿFFFFAE Tablets. These tablets are ground into a fine
powder and processed into a sublingual medication.*


 CLICK HERE FOR MORE INFO


 unsubscribe here
```

[7] http://www.spamnix.org

```
kqeh ec w umlfswb xsawd py abuckwtaawmek nvs ihf
SPAM: ------------------------ Spamnix Spam Report -----------
--------------

SPAM: Spamnix identified this message as spam.  This report
shows which
SPAM: rules matched the message and how many points each rule
contributed.
SPAM:
SPAM: Content analysis details:   (12.00 hits, 5 required)
SPAM: X_MSMAIL_PRIORITY_HIGH (0.4 points)  Sent with 'X-
msmail-Priority' set to high
SPAM: X_PRIORITY_HIGH    (1.9 points)  Sent with 'X-
Priority' set to high
SPAM: CLICK_BELOW_CAPS   (0.5 points)  BODY: Asks you to
click below (in capital letters)
SPAM: HTML_COMMENT_SAVED_URL (1.4 points)  BODY: HTML
message is a saved web page
SPAM: HTML_FONT_COLOR_UNSAFE (0.1 points)  BODY: HTML font
color not within safe 6x6x6 palette
SPAM: HTML_60_70         (0.5 points)  BODY: Message is 60%
to 70% HTML
SPAM: HTML_FONT_COLOR_RED (0.1 points)  BODY: HTML font
color is red
SPAM: HTML_MESSAGE       (0.2 points)  BODY: HTML included in
message
SPAM: HTML_LINK_CLICK_CAPS (1.1 points)  BODY: HTML link text
says "CLICK"
SPAM: HTML_FONT_BIG      (0.3 points)  BODY: FONT Size +2 and
up or 3 and up
SPAM: HTML_FONT_COLOR_BLUE (0.1 points)  BODY: HTML font color
is blue
SPAM: HTML_LINK_CLICK_HERE (0.1 points)  BODY: HTML link text
says "click here"
SPAM: HTML_SHOUTING4     (0.5 points)  BODY: HTML has very
strong "shouting" markup
SPAM: REMOVE_PAGE        (0.3 points)  URI: URL of page called
"remove"
SPAM: FORGED_MUA_OUTLOOK (3.9 points)  Forged mail pretending
to be from MS Outlook
SPAM: MIME_HTML_ONLY     (0.1 points)  Message only has
text/html MIME parts
SPAM: MISSING_MIMEOLE    (0.5 points)  Message has X-MSMail-
Priority, but no X-MimeOLE
SPAM:
SPAM: Spam level: ************
```

```
SPAM: -------------------- End of Spamnix Spam Report -------
-------------
```

SpamAssassin

Ein beliebtes und leistungsstarkes Anti-Spam-Tool ist der frei verfügbare SpamAssassin[8]. SpamAssassin lässt sich in nahezu jeden MTA integrieren, bietet aber auch die Möglichkeit, als Plug-In für MUAs unter Windows zu fungieren.

SpamAssassin kann über unterschiedliche Quellen bezogen werden:

1. Als vorkonfiguriertes Paket bzw. als tar-Archiv über die SpamAssassin Home-Page.
2. Als PERL-Module mittels des Aufrufs (als *root*):

```
perl -MCPAN -e shell
o conf prerequisites_policy ask
install Mail::SpamAssassin
quit
```

3. Als Bestandteil der lokalen Unix-Distribution (rpm, deb bzw. port).

SpamAssassin weist eine hohe Abhängigkeit von der installierten PERL-Version auf. Weitere vielfältige Systemabhängigkeiten und die zusätzlich vornehmenden Anpassungen, machen SpamAssassin nicht gerade einfach zu installieren bzw. erfolgreich aufzusetzen. Zudem existieren diffizile Unterschiede in der Konfiguration von SpamAssassin hinsichtlich der Major- aber auch Minor-Versionen.

Aufsetzen von SpamAssassin

Das ursprünglich in PERL geschriebenen SpamAssassin (CPAN: Mail::Mail-SpamAssassin) bietet mittlerweile eine Client/Server-Architektur, bei der die E-Mail zunächst vom (in C geschriebenen) Client entgegen genommen, dann aber von Memory-residente (PERL-)Serverprozess analysiert wird. Hierdurch wird erreicht, dass nicht jedesmal die umfangreich PERL-Bibliothek pro E-Mail geladen werden muss, sondern diese quasi im Hauptspeicher verbleibt.

Unter Qmail gibt es mehrere Varianten SpamAssassin einzusetzen[9]:

- Als Teil des Anti-Virus-Programms Qmail-Scanner[10].
- Als Ersatz des Programms `qmail-queue`. Bei Qmail lässt sich dies durch das QMAILQUEUE Patch[11] von Bruce Guenther sowie durch das **qmail-spamc** Programm John Peacock als Bestandteil von SpamAssassin erzielen:

---

[8] http://spamassassin.org/

[9] vgl. http://loghog.corc.uni.edu.ni/~jorge/spamassassin.html

[10] http://qmail-scanner.sourceforge.net/

[11] http://www.qcc.sk.ca/ bgueenter/

```
qmail-smtpd => qmail-spamc (<=> spamd) => qmail-queue
```

Hierbei ist es notwendig, im Startup-Skript von **qmail-smtpd** die Environment-Variable `QMAILQUEUE="qmail-spamc"` mit der korrekten Pfadangabe zu setzen.

- In der Qmail Default-Delivery mittels des Zusatzprogramms **maildir**. **maildir** ist Bestandteil der Programmsammlung **safecat**[12] von Len Budney. Folgendes **qmail-send** `run`-Skript ist hierzu in der Lage.

```
#!/bin/sh
exec env - PATH="/var/qmail/bin:$PATH" \
qmail-start '|/usr/local/bin/spamassassin -L | maildir
./Maildir/'
```

- Durch (individuelle) Einbettung in der `.qmail` Datei mittels des Zusatzprogramms **maildir**.

```
|/usr/local/bin/spamassassin -L | maildir ./Maildir/
```

In diesem Fall darf keine zusätzliche Angabe von `./Maildir/` in einer weiteren Zeile erfolgen, da die Zustellung der Nachricht nun mittels **maildir** über die Pipe erfolgt.

- Das Shell-Skript **ifspamh**[13] stellt durch seinen Aufruf innerhalb von `.qmail` Dateien () einen Wrapper für den **spamassassin** Aufruf dar. Allerdings sehe ich in meinen Qmail-Logs öfter folgenden Eintrag (unter Nutzung der **ksh**):

```
2003-05-19 22:59:26.426539500 delivery 1196: deferral:
/usr/local/sa/bin/ifspamh[83]:_printf:_Argument_list_to
o_long//usr/local/sa/bin/ifspamh[113]:_printf:_Argument
_list_too_long/spamc_returned_temporary_failure/
```

- Als Aufruf innerhalb von **procmail**[14]. Hierzu muss zunächst die (lokale) `.qmail` zum Aufruf von **procmail** angepasst werden:

```
# .qmail for procmail (customize .procmailrc!)
| preline /usr/bin/procmail
```

Die Einbettung von SpamAssassin erfolgt in der obligatorischen `.procmailrc` Datei:

```
# Sampel .procmailrc
:0fw
* < 265000
```

---

[12] `http://budney.homeunix.net:8080/users/budney/linux/software/safecat.html` bzw. `http://freshmeat.net/releases/124946/`

[13] `http://www.gbnet.net/~jrg/qmail/ifspamh/`

[14] `http://www.procmail.org/`

```
        | /usr/bin/spamc
        :0
```

spamd Daemon

Statt des monolithischen **spamassassin** PERL-Skripts kann auch die Kombination **spamc**/**spamd** eingesetzt werden. Der Client **spamc** (ein C-Programm) ersetzt hierbei den **spamassassin** Aufruf und verbindet sich über das Loopback-Interface und den TCP-Port 783 mit dem Daemon-Prozess **spamd** (ein PERL-Programm) — dem eigentlichen Arbeitspferd.

Hierzu muss natürlich zunächst der Daemon gestartet sein. Dies realisiert ein einfaches `run`-Skript unter den Daemontools:

```
#!/bin/sh
exec setuidgid root /usr/bin/spamd -x -L
```

Damit **spamd** eine Netzwerk-Ressource nutzen kann, muss es unter *root* laufen (via **setuidgid**); die Option besagt, dass es ohne lokalen User auskommt und mit dem Flag `-L` wird **spamd** nur lokale Informationen auszuwerten; insbesondere keinen DNS-Lookup vorzunehmen.

Leider war bei meiner SpamAssassin Version 2.54 und unter PERL 5.005_03 **spamd** durch ein kleines Bug nicht lauffähig: Der Aufruf von **mkdir** wollte aufgrund einer fehlenden Verzeichnis-Mode Angaben nicht gelingen und wurde mit einem Fehler quittiert ("Not enough arguments for mkdir at /usr/bin/spamd line 878, near "$spam_conf_dir"). Abhilfe schafft (etwa ab Zeile 875 in **spamd**):

```
if ( ! -d $spam_conf_dir )
    {
       if ( mkdir ($spam_conf_dir,0700) )
       {
          logmsg "info: created $spam_conf_dir for
$username.";
       }
```

Konfiguration

Die Installation von SpamAssassin über das tar-Archiv nutzt als Standardverzeichnis `/usr/share/spamassassin/`, die Binaries werden üblicherweise in `/usr/bin/` hinterlegt. Als zentrale Konfigurationsdatei fungiert `/etc/mails/spamassassin/local.cf`. Bein Einsatz des Daemon **spamd**, werden Änderung in `local.cf` erst nach dessen Neustart wirksam.

SpamAssassin kann (wie beschrieben) Systembezogen- als auch Userbezogen eingerichtet werden. Hierzu erhält jeder Benutzer ein eigenes `~/.spamassassin` Verzeichnis zur Bervorratung der Präferenzen. Zudem unterstützt SpamAssassin auch die unter Qmail möglichen "Virtuellen Benutzer".

Aufgrund dieser Komplexität und den Abhängigkeiten vom lokalen System (**procmail**, Qmail-Scanner), kommt der System-Administrator nicht um ein genaues Studium der SpamAssassin Dokumente (`perldoc`

`Mail::SpamAssassin::Conf`) sowie um einiges Experimentieren nicht herum.

Razor

SpamAssassin bietet optional die Möglichkeit auf eine verteilte Spam-Datenbank Razor[15] ("Rasierer") zuzugreifen, die von Vipul Ved Prakash ins Leben gerufen wurde und mittels Agenten (u.a. via eines Plug-Ins für SpamAssassin) befüllt und aktuell gehalten werden kann.

Im Gegensatz zu einer RBL werden hier "Footprints" der Spam E-Mails hinterlegt, die aufgrund statistischer Analysen der E-Mails und unter Eliminierung ihrer zufälligen Anteile erzeugt werden.

7.2.5.3 Message Digest

Das Message Digest Verfahren dreht das SMTP-Verfahren gewissermassen um. Heisst das Standard E-Mail-Verfahren:

- Akzeptiere alles — aber versuch' dich vor den "Bad Boys" zu schützen,

lautete es nun:

- Akzeptiere nichts — ausser du weisst ganz genau, dass es für dich bestimmt ist.

Beim Digesting wird somit der Absender gezwungen— wie bei einer Mailing-Liste — über einen Digest-Agent (Autoresponder) des Adressaten seine eigene E-Mail zu bestätigen. Das Message Digest Verfahren kann daher nur für individuelle Empfänger eingesetzt werden, wobei es zwei verbreitete Alternativen gibt:

1. Der Message Digest Agent verwaltet eine *Whitelist* von SMTP-Adressen.(*Sender-based Delivery Confirmation*). Nur E-Mails, die von einem registrierten Absender stammen werden sofort durchgelassen; alle anderen Absender werden durch eine Bestätigungs-E-Mail zunächst in die Whitelist aufgenommen. Dieses Verfahren verfolgt z.B. der bekannte Tagged Message Digest Agent TMDA[16]. Zusätzliche Anti-Spam-Filter sowie ein automatisches Aging machen den TMDA für den Anwender sehr bequem, der z.B. über ein dot-qmail Plug-In integriert werden kann.
2. Eine rigidere Strategie verfolgt z.B. das Programm **qsecretary** von Dan Bernstein, das seinen Dienst in der Qmail Mailing-Liste versieht: Hier muss jede einzelne E-Mail vom Absender bestätigt werden (*Message-based Delivery Confirmation*). Als Response auf die Original-Mail bekommt der Absender mittels des Qmail-VERP-Mechanismsu einen Cookie zugeschickt, den dieser beantworten muss. Spammer erhalten so in der Regel keine Chance;

[15] http://razor.sourceforge.net/

[16] http://www.tmda.sourceforge.net

ausser sie nutzen einen schnellen Autoresponder. Sicherlich ist dies einer der Gründe, warum Dan Bernstein den Quellcode des Programms bislang nicht veröffentlicht hat. Als Alternative bietet sich das Gerit Pape geschriebene **qconfirm**[17]an , das beide Betriebsmodi unterstützt.

So bequem und effektiv diese Werkzeuge für den Empfänger von E-Mails sind, so lästig sind sie für den Absender. Manche E-Mail hat auf diese Weise deshalb ihr Ziel nicht erreicht, weil sie einfach nicht — aus welchem Grund auch immer — nicht bestätigt wurde. Zudem verdreifacht der Message Digest Agent das E-Mail-Aufkommen prinzipiell (zumindest in der Zahl, nicht unbedingt im Volumen); keine Freude für den sparsamen E-Mail-Versender.

Andererseits gelten für MUAs, die mit einem Message Digest Agent ausgestattet sind, beim Versand der Bestätigungs-Mail im Falle von Spam vergleichbare Überlegungen wie bei Bounces: Wenige bis keine erreichen das Ziel; eine Double-Bounce Nachricht an den Postmaster ist die Regel.

Kommerzielle Anti-Spam-Dienste

Neben der freien Tools bieten mittlerweile auch kommerzielle Anbieter Anti-Spam-Dienste an. Hierbei wird von den Firmen (z.B. Brightmail) nicht nur an Spam-Filter angeboten, sondern auch aktive "Spam-Probes" im Internet genutzt. Empfängt ein Spam-Probe eine E-Mail und klassifiziert sie als Spam wird sie mit ihren typischen Merkmalen (über einen "Footprint") in einer zentrale Datenbank eingetragen und sie auf diese Weise klassifiziert und für Nutzer des Anti-Spam-Dienstes erkennbar und somit blockierbar gemacht.

Dieses Verfahren funktioniert daher ähnlich wie eine RBL, nur findet es ***Post-Receipt-Spam*** statt, da jede E-Mail zunächst in Emfang genommen werden muss, bevor sie klassifiziert und getagged bzw. verworfen werden kann.

7.2.5.4 Tagging

Als Tagging die Hinzufügung von Informationen in der E-Mail Verstanden, die es ermöglicht, die E-Mail als Spam zu identifzieren. In der Regel werden die Merkmale entweder

- als eigenständige E-Mail Headerzeile (X-Spam-Info:),
- (im Spam-Fall) im als Teil des "Subject:"-Angabe oder aber
- (in Spam-Fall) in den Message-Body

hineingefügt. Das heute übliche Verfahren (z.B. bei SpamAssassin) ist das Hinzufügung einer neuen Headerzeile. Bei älteren Version (bzw. bei Bedarf) wird hingegen die "Subject:"-Zeile modifizieren. Hintergrund hierfür ist die Tatsache, dass nicht alle E-Mail-Clients Filter auf beliebige Headerzeilen setzen können, sodass dann das Filtern ins Leere läuft.

---

[17] http://smarden.org/qconfirm

Problematisch beim Tagging ist, dass alle Spam-Informationen konsistent zusammengefügt werden müssen. Hierzu zählen z.B. Angaben, ob die Absender-IP auf einer Blocking List steht und ber RBL gefunden wurde, welchen Stellenwert die ausgewertete SMPT-Dialoginformation (Return-Path, Helo) geniest, inwieweit und in welchem Umfang die Ergebnisse der strukturellen und inhaltlichen Analyse der Nachricht eingeflossen sind.

Vorteilhaft beim Tagging ist, dass die potentiellen Spam-Informationen zentral gesammelt und verarbeitet werden können; jeder Anwender erhält somit die eine Spam-Analyse auf gleichem Niveau. Aufgabe des Mail User Agents ist es nun, mittels geeigneter Filterkriterien, das Einsortieren der Spam E-Mails vorzunehmen.

Environment-Variablen

Im Daemontools `run`-Skript des SMTP-Servers können sich Aufrufe von **tcpserver**, **rblsmtpd** und natürlich **qmail-smtpd** befinden. Dan Bernstein hat seine Programme so konstruiert, dass sie über Argumente aufgerufen werden: **exec** ruft **tcpserver**, **tcpserver** ruf **rblsmtpd**, **rblsmtpd** ruft **qmail-smtpd** und (falls entsprechend per Patch verfügbar), **qmail-smtpd** ruft ein PAM-Modul zur SMTP-Authentisierung auf. Während des Aufruf des `run`-Skriptes (oder ggf. des Run-Level-Skriptes unter `/etc/init.d/`) laufen alle Programme im selben Environment. Der Datenaustausch zwischen den Programmen kann somit effizient über Environment-Variablen erfolgen. Eine von diesen ist z.B. `$RBLSMTPD`.

Alle zum Tagging notwendigen Informationen müssen in geeignete Environment-Variablen gesteckt werden. Anders als bei z.B. UCSPI gibt es aber für Spam keine Konventionen und keine Vorgaben. Hierdurch müssen alle Programme entsprechend gepatcht werden, damit sie diese Informationen einerseits bereitstellen und andererseits verwerten können. Als letztes Glied der Kette schreibt **qmail-smtpd** einen qualifizierten E-Mail Header für die einlaufende E-Mail. An dieser Stelle (`received.c`) ist die Spam-Information einzufügen.

7.2.5.5 Zusammmenstellung der Methoden

Die bislang vorgenommenen Überlegungen hinsichtlich der Blockade, des Filterns und des Taggings, sowie des Digesting von E-Mails sollen nun systematisch zusammengestellt werden.

Wie wir gesehen haben, gibt es

1. *Pre-Initial-*,
2. *Post-Initial-* sowie
3. *Post-Receipt-Verfahren*.

Pre- und Post-Initial-Verfahren blockieren Spam E-Mails, während alle Post-Receipt-Verfahren lediglich zum Filtern, Tagging bzw. Digesting von E-Mails nützlich sind. Aus der Sicht des Systemadinistrators sind also die ersten beiden Verfahren zu bevorzugen, da sie sowohl die Last seiner Systeme als auch das

Spam-Volumen im Internet reduzieren. Für den Awender spielt dies nur eine untergeordnete Rolle. Er ist an eine an einer möglichst vollständigen Spam-Unterdrückung interessiert, die letzlich immer nur durch die Kombination mehrerer Verfahren zu realisieren ist — auch wenn diese CPU-Leistung kosten.

Für die Unterdrückung von Spam E-Mails sind zwei Grössen relevant:

- Wie *effektiv* ist eine Methode? Effektiv heisst, welchen Anteil am Spam-Aufkommen blockiert bzw. gefiltert werden kann.
- Wie *effizient* ist die Methode? Die Effizienz drückt sich durch den Anteil der False Positives sowie False Negatives aus.

Zudem haben alle Verfahren Auswirkungen auf die beteiligten Parteien:

- Den *Absender/Übermitter* der E-Mail: Bekommt er eine qualifizierte Antwort (in Falle der *False Postives*)?
- Dem *Empfänger* der E-Mail: Was hat er beim Empfang (von identifizierter Spam) E-Mail zu beachten? Besitzt er Möglichkeiten, die Klassifikation der E-Mails zu beinflussen (Trigger)?
- Dem *Systemadminstrator*? Welchen Aufwand hat er bei der Konfiguration und Administration der Anti-Spam-Verfahren?
- In grösseren Betrieben muss hierzu auch die *Abuse-Abteilung* gezählt werden, die sich mit dem Spam an die betroffenen Mitarbeitern aber vielleicht auch mit unberechtigten Blockaden (False Positives) herumschlagen muss.

Zwei weitere Fragen müssen bei den Verfahren gestellt werden hinsichtlich

- der notwendigen *Ressourcen* (CPU, Netz, Speicherplatz) und der hierdurch potentiell möglichen Denial-of-Service (DoS) Gefahren,
- der resultierenden *Bandbreite* des Spam-Aufkommens (einschliesslich Bounces und Double-Bounces).

Verfahren	**Pre-Initial**		**Post-Initial**		**Post-Receipt**		
Schicht	IP	SMTP	IP	SMTP	Anwen-dung (Queue)	Anwen-dung (MDA)	Anwen-dung (MUA)
Name	Remote RBL	SPAM-CON-TROL	IP-Aus-schluss-liste	Absender-Aus-schluss-liste	Filter	Filter	Filter
Beispiel	RBL-SMTPD	badmail-from/	Spam-trap[1])	RTS[2])	procmail	Spam-Assassin	TMDA

Verfahren	**Pre-Initial**		**Post-Initial**		**Post-Receipt**		
		badhelo					
Trigger	extern	intern, Konfigu-rations-datei	self /Absender	Absender/ Empfänger	extern/ Em-pfänger	System/ Em-pfänger	Absender
Risiko (False Negatives)	mittel	gering	sehr gering	sehr gering	gross	gering	gross
Effektivität	30%	10%-30%	50%	70%	80% -- [3])	90% + (Taining)	100%
Effizienz	100%	100%	90%	90%	80% --	90% + (Taining)	100%
Empfänger-Aktion	keine	keine	möglich	möglich	möglich	möglicht	Return-Mail
Absender (False Positives)	keine bzw. URL	SMTP Protokoll-fehler	URL	SMTP-Protokoll-fehler	keine	keine	keine
Admini-stration	gering	mittel	gering	mittel	gross	Em-pfänger	Em-pfänger
Bandbreite	gering/ DNS-RBL Lookup	sehr gering/ DNS-MX Lookup	gering	gering	hoch/ DNS-FootprintLookup	hoch/ DNS-Footprint Lookup	sehr gross

Tabelle 7.2-2: Zusammstellung der Blockade- und Filtermassnahmen für Spam E-Mails [1]) Spamtrap = ein Account zum „Fangen" von Spam E-Mails, [2]) RTS = Return-To-Sender, [3]) mit abnehmender Tendenz

Zusammengefasst kann man sagen, dass der Parameter-Raum zur Blockade/Unterdrückung von Spam E-Mails einerseits dargestellt werden kann durch ein

- **Benutzer-Dreieck**, das sich zwischen *Absender*, *Empfänger* sowie *System-Administrator* (bzw. Abuse-Abteilung) erstreckt, sowie andererseits durch ein
- **Aufwands- und Ressourcen-Viereck**, das mit den Grössen *Effektivität*,

Effizienz, Rechern-Ressourcen und *Bandbreite* aufgespannt werden kann.

Es ist letztlich die konkrete Situation, die analysiert und aus der Rückschlüsse gezogen werden sollten, welche "beste" Parametrierung dem derzeitigen Spam-Niveau angemessen ist. Hier lassen sich keine allgemeinen Ratschläge geben; wichtig ist in jedem Fall zunächt die genaue Kenntnis des aktuellen Spam-"Befalls".

7.2.6 Zusätzliche Anforderungen

7.2.6.1 Anforderung an den E-Mail-Provider

Egal ob blockiert oder gefiltert, für den E-Mail Provider stellt sich die Anforderung, nicht nur die durchgelassenen E-Mails zu protokollieren, sondern auch im besonderen Masse auch die blockierten bzw. gefilterten. Hierzu sagt RFC 2505 in §2.4:

> *"The MTA SHOULD be able to log all anti-relay/anti-spam actions. The log entries SHOULD contain at least:*
>
> *o Time information.*
>
> *o Refusal information, i.e. why the request was refused ("Mal From", "Relaying Denied", "Spam User", "Spam Host", etc).*
>
> *o "RCPT To:" addresses (domains).*
> *(If the connection was disallowed at an earlier stage, e.g. by checking the SMTP_Caller IP address, the "RCPT To:" address is unknown and therefore cannot be logged).*
>
> *o Offending host's IP address.*
>
> *o Offending host's FQDN hostname.*
>
> *o Other relevant information (e.g. given during the SMTP dialogue, before we decided to refuse the request)."*

Zur Kontrolle und Verständnis des Spam-Niveaus muss der Systemadministrator des MTA eine qualifizierte Analyse seiner Logfiles vornehmen. Mittels meines SPAMCONTROL-Patches für Qmail sowie dem Zusatzprogramm **newanalyse** lässt sich dies im Falle von Qmail sehr bequem realisieren. Voraussetzung ist allerdings, dass die Log-Aufzeichnung mittels **multilog** von Dan Berstein erfolgt.

Zusätzlich sind die Logfiles — zumindest für einen bestimmten Zeitraum — zu archivieren. Auch hier bietet **newanalyse** ein einfaches Verfahren an. Im Gegensatz hierzu rotieren zwar auch die per **syslogd** generieren Logfiles (in der Regel `/var/log/maillog`), doch werden üblicherweise nur die letzten 10 Tage aufgehoben. Hier muss der Systemadministrator händisch für die

Archivierung der Loginformation sorgen.

7.2.6.2 Anforderungen an eine künftige E-Mail Infrastruktur

Es gibt Pessismisten, die annehmen, dass SMTP-E-Mail in einigen Jahren aufgrund des Spam zusammenbrechen wird. Dies gilt unter der Voraussetzung, dass der Spam-Anteil stetig steigen wird. Natürlich kann dies kein Mensch voraussagen. Andererseits steigen sowohl die Anzahl der E-Mail-Teilnehmer, als auch die Kapazitäten zur Verarbeitung der E-Mails (Bandbreite, Leistung der MTAs etc.). Inwieweit sich die Fakturen kompensieren oder auch verstärken, kann nicht qualifiziert abgeschätzt werden. Fraglich ist auch, ob es nicht aufgrund des zu erwartenden "Sättigungsverhalten" hinsichtlich von Spam dazu kommt, dass die Effektivtät für die Spammer so sehr fällt, dass das Geschäft nicht mehr lukrativ ist.

Andererseits werden Stimmen lauter, dass eine neue E-Mail-Infrastruktur geschaffen werden müsste, die weniger Spam-Anfällig ist. Drei Ansätze wurden bislang gemacht:

1. Abkehr von der SMTP-E-Mail; hierzu gab/gibt es eine Initative von Dan Bernstein, "Internet Mail 2000" IM2000[18]. Hierbei reden wir immer von drei Bausteinen: 1. Dem E-Mail-Protokoll selbst, 2. dem Aufbau der E-Mail und 3. den Zulieferprotokollen à la POP3/IMAP4.
2. Ergänzung des (E-)SMTP-Protokolls; als Beispiel hierfür fungiert TORO[19]: "Trust of Reliable Origin" von Marc-Andre Pelletier.
3. Ergänzung des DNS-Protokolls zum Reverse-Lookup der IP-Adressen von MTAs, was als RMX[20] oder manchmal auch als XM bezeichnet wird; hierzu gibt es einen Draft von Hadmut Danisch.

Während Dan Bernstein's Vorstoss bislang weitgehend unbeachtet blieb, besitzen die letzteren beiden einen IETF Draft-Status. Kritisch muss an dieser Stelle beleuchtet werden:

Wir haben kein eigentliches Problem des SMTP-Protokolls hinsichtlich Spam. SMTP funktioniert so gut, wie es geplant ist: Als Übertragungs- und Nachrichten-Aufbau-Protokoll. Die aktuellen Versionen (RFC 2821/2822) vom Autor Klensin haben zum Verständnis und Weiterentwicklung nur wenig

[18] http://cr.yp.to/im2000.html

[19] http://www.ietf.org/internet-drafts/draft-smtp-trust-00.txt

[20] http://www.ietf.org/internet-drafts/draft-danisch-dns-rr-smtp-02.txt

beigetragen.

Ergänzungen, wie die SMTP-Authentication, haben relativ wenig gebracht, weil Spam kein Problem der *Authentisierung*, sondern eins der *Authorisierung* ist: Der SMTP-Sender kann in der Regel senden *was* und *wohin* er will; jeder SMTP-Empfänger muss akzeptieren, was er bekommt (nur nicht-authorisiertes Weiterleiten darf unterbunden werden). Die Situation ist also recht disparitätisch.

Der Vorschlag von M.A. Pelletier löst vielleicht die Frage der Authentisierung über Relays; nicht jedoch die der Authorisierung. Ferner vermischt der Draft die bislang beim SMTP-Protokoll sorgfältig vorgenomme Trennung zwischen Protokollelementen zur Übertragung (RFC 821) und solche zur Beschreibung des Nachrichten-Formats (insbesondere des E-Mail-Headers; RFC 822).

Die DNS-Ergänzung von H. Danish ist demgegenüber sogar noch kontraproduktiv: Sie bricht das Forwarding von E-Mails mit dem Original-SMTP-Abender über MTAs und sie erzeugt unerwünscht grosse DNS-Packete; ferner ist sie für ISPs nur schlecht administrierbar, weil diese ständig die IP-Adressen ihrer MTAs im DNS aktualisieren müssen. Die Authorisierung von Sendern lässt sich nicht über technische, sondern nur über administrative Massnahmen regeln. Ein Vorschlag besteht darin, für jede Domain mit einem SMTP-Sender, die möglichen MTAs mit ihrer jeweiligen Reverse-IP-Adresse über einen trivialen DNS TXT-Record kenntlich zu machen:

```
102.23.4.195.in-adr.arpa TXT MTA=yes
```

Hierüber kann sofort (Pre-Receipt) mittels der IP-Adresse überprüft werden, ob ein SMTP-Sender per DNS authorisiert ist. Dies ist unabhängig davon, *welche* E-Mail er verschickt und *welchen* SMTP-Absender diese besitzt. Hierfür müsste der SMTP-Server so modifiziert werden, dass per Default ein DNS TXT Lookup mit der Reverse IP-Adresse des sendenden SMTP-Clients vorgenommen wird. Bei **qmail-smtpd** kann für Clients, bei denen die Environment-Variable `$RELAYCLIENT` gesetzt ist, auf diesen Lookup verzichtet werden. Für E-Mail-Sender, die per Dial-Up (d.h. ständig wechselnden IP-Adress) über den MTA ihrer Firma senden möchten *(Roaming User)*, bietet sich an, SMTP-Authentication zu verwenden. Das Verfahren ist weiterhin kompatibel mit dem beliebten *POP-before-SMTP*-Verfahren. Die Tatsache, dass Protokolle wie POP3 und IMAP4 nur zum Abholen der E-Mail, nicht aber zu Verschicken geeignet sind, sondern letzteres über das SMTP-Protokoll erfolgt, ist sicherlich einer der Mitverursacher der heutigen Spam-Misere.

7.2.6.3 Organisationen und nützliche Links

Zum Abschluss möchte ich noch auf einige mehr-oder-weniger nützliche Quellen im Internet hinweisen, hinter denen sich Organisationen befinden, die sich der Spam-Abwehr bemühen. Diese Liste ist bei weitem nicht vollständig:

- Die "offizielle" Anti-Spam Research Group (ASRG)[21] des IETF.
- Das Internet Mail Consortium (imc)[22].
- Die deutsche Anti-Spam Webseite[23] von Florian Klein (alias "DocSnyder").
- http://www.spamfaq.net/ (nicht mehr Online).
- Der Anti-Spam-Dienst SpamCop[24].
- Der Anti-Spam-Dienst SPAMHAUS[25].
- Die Webseite des Mail Abuse Prevention System LLC (MAPSSM)[26].

Weitere interessante URLs finden sich unter dem jeweiligen Thema is diesem Abschnitt; inwieweit diese aber Bestand haben, ist natürlich ungewiss.

7.3 Viren und Würmer in E-Mails

Bei der Ausstattung eines PCs unter einem Windows-Betriebssystem gehört ein Viren-Scanner auf die "Muss"-Liste der Softwarebeschaffung, häufig begleitet von zusätzlichen Werkzeugen wie eine persönliche Firewall. Der PC ist mittlerweile sowohl zum Opfer als auch zum Täter für Angriffe aus dem Internet geworden. Einerseits sind die heutigen PCs — speziell unter dem Betriebssystem Windows XP — mit einem TCP/IP-Stack versehen, der sich hinsichtlich seiner Leistungsfähigkeiten mit Unix-Servern vergleichen lässt, andererseits macht es die DSL-Technologie für den Internet-Anschluss möglich, hohe Datenraten zu realisieren.

Ein ungeschützter PC — mit einer DSL-Flatrate am Internet angeschlossen — wird per Port-Scans spätestens nach 20 Minuten "online" erkannt (speziell bei Nutzung von T-DSL) und dient als Opfer umfangreicher Angriffe auf die offenen TCP-Ports. Gelingt es, auf diesem PC einen "Trojaner" zu installieren, kann dieser als Ausgangspunkt für Spam-, Virus- oder DoS-Attacken auf andere Systeme genutzt werden und wird somit zum Täter. Über ein solches System lassen sich ohne Probleme innerhalb kurzer Zeit Millionen von E-Mails versenden. Bedenkt man, dass sicherlich hunderttausende solcher Rechner potentiell erreichbar bzw. verwundbar und angreifbar sind, wird die Grössenordnung unseres Viren- aber auch Spam-Problems erkennbar:

Anti-Viren Paradigma

[21] http://www.irtf.org/charters/asrg.html
[22] http://www.imc.org/
[23] http://www.antispam.de/
[24] http://www.spamcop.net/
[25] http://spamhaus.org/
[26] http://mail-abuse.org/

Bei der Virenabwehr geht es heute nicht mehr um das Abfangen vereinzelt eintreffender E-Mails von versehentlich infizierten Rechnern. Vielmehr gilt es, die gezielte Infektion "schwacher" Systeme zu verhindern, die ihrerseits als Ausgangspunkt eines gesteuerten Massenangriff auf die E-Mail-Infrastruktur des Internet genutzt werden sollen. Zudem sind die E-Mail-Gateways so aufzusetzen, dass sie selbst bei einer Virus-Epedie nicht selbst Opfer werden und dadurch die übliche E-Mail-Kommunikation leidet bzw. zusammen bricht.

7.3.1 Kleine (Computer-)Virologie

Bevor mit der Diskussion der Virenabwehr begonnen werden soll, möchte ich eine kleine "Virenkunde" voranstellen, die das Problem in einen gewissermassen "historischen" Rahmen stellt. Doch zunächst die Definition der gebräuchlichen Begrifflichkeiten[27]:

- *Computer-Viren* fanden Verbreitung mit dem Aufkommen der Personal Computer; worunter diesmal ausnahmsweise nicht der IBM-kompatible PC gemeint ist, sondern auch z.B. der z80 und später der Atari ST. Der Computer war damals auch zum Gutteil Spielzeug, und so verwundert es nicht, dass sich die Vieren speziell über Computerspiele verbreiteten. Hier sind im besonderen die sog. Boot-Sektor-Viren gemeint, die beim Laden eines Spiele-Programms von Diskette greifen.
  Ein Virus ist immer Betriebssystem-spezifisch und enthält einen ausführbaren Code, der zunächst andere in den Speicher geladenen Programme infiziert und ggf. beim Zurückspeichern modifiziert. Die modifizierten (ausführbaren) Dateien richten dann ihrerseits die gewünschte Schadenswirkung an, z.B. Löschen von Dateien oder Zerstörung der Datenstrukturen bzw. des Dateisystems.
- Das *EICAR*[28]*-Virus* ist kein Virus im eigentlichen Sinne, sondern eine Virussignatur, die vom *Europen Institute for Computer Ant Virus Research* zur Überprüfung der Wirksamkeit von Anti-Viren-Programmen definiert wurde, ohne selbst Schaden zu verursachen. Die EICAR-Signatur sieht folgendermassen aus:
  `X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*`
- *Mutationen* bzw. *Varianten* liegen dann vor, wenn das Virus keine feste Checksumme (z.B. MD5 Hash der *.exe Datei) sondern einen sich ändernden Code mit u.U. leicht modifizierten Schadensfunktionen besitzt. Zur Unterscheidung der Mutationen bzw. Varianten vergeben die Anti-Viren-

[27] vgl. http://www.trojaner-und-sicherheit.de/

[28] http://www.eicar.org

Hersteller den Viren meist einen Buchstaben als Suffix (z.B. *W32/Sobig.F*).

- *Würmer* (*Worms*) sind Programme, die einen Selbstreplikations-Mechanismus aufweisen und sich über Kommunikationskanäle selbständig verbreitern. Als Kommunikationskanäle fungieren freigegebene Laufwerke (Shares), Peer-zu-Peer-Anwendungen sowie natürlich auch Internet E-Mail. Die Selbsreplikation kann entweder über die Nutzung von Betriebssystem-Funktionen erfolgen, oder aber durch eine eigenständige Client-Komponente, z.B. eine SMTP-Engine. Hierbei attackieren Würmer in erster Line die bekannten Schwachstellen diverser Betriebssysteme (und hier natürlich in erster Line die des Herstellers Microsoft) und Anwendungen wie den Microsoft IIS (Internet Information Server) aber auch den frei verfügbaren Web-Servers Apache.
- Als *Trojaner*, *Trojanische Pferde* (*Trojan Horses*) werden Programme bezeichnet die vorgeben, bestimmte harmlose (und gewünschte) Funktionen mitzubringen, aber das Ziel verfolgen, einen unberechtigten Zugriff auf den infizierten Rechner zu ermöglichen.
- Sie installieren somit auf dem Zielrechner eine *Backdoor*. Die Backdoor verhält sich zunächst passiv, d.h. wird erst bei Anforderung (z.B. über ein zusätzlich geöffnetes) TCP-Port aktiv.
- *Malicious Code* kann hingegen auch über sog. *Exploits* auf dem Zielrechner installiert werden. Bekannt geworden sind z.B. PHP-Exploits im Zusammenhang mit dem Apache Web-Server sowie vor allem der Mitte 2001 unter dem Namen *Code Red* bekannt gewordene Angriff[29] auf den Microsoft IIS. Bei den Web-Clients gelten Systeme unter Nutzung von Microsoft's ActiveX Skripten als besonders gefährdet.
- *Hoaxes*[30] sind hingegen (bewusst) falsche Viren-Warnungen, die in E-Mails mit besonders "reißerischer" Aufmachung versandt werden. Ziel ist des Hoaxes ist es, den Empfänger zu verunsichern und ihn dahingehend zu beeinflussen, diese E-Mail an Bekannte und Kollegen weiterzuschicken (Kettenbrief).

7.3.1.1 Begegnung der ersten Art ...

ChristmasTree Exec

Der erste mir bekannte Virus (besser Wurm) in E-Mails datiert auf das Jahr 1988 zurück. Damals war das Internet noch im Aufbau begriffen und die infizierte E-Mail erreichte uns (d.h. das MPI für Physik und Astrophysik in München, wo ich damals Doktorand war) über das *European Academic Research Net* (EARN). Zwar waren schon PCs und Apple Rechner im Einsatz, doch der Wurm zielte auf

[29] http://www.cert.org/advisories/CA-2001-19.html
[30] http://www.tu-berlin.de/www/software/hoax.shtml

Grossrechner, die unter dem Betriebssystem VM/CMS liefen, bzw. die hierunter verfügbare E-Mail Anwendung *Note*. Der Wurm produzierte beim Öffnen der E-Mail einen wunderschönen Tannenbaum (in EBCDIC Code) mit "Happy Christmas!" Wünschen. Gleichzeitig las er das lokale Adressbuch (All.Notebook.A0) aus und vervielfältigte sich auf diese Weise — ein typischer Wurm also.

Der Wurm wurde *ChristmasTree Exec* genannt, da er sich um ein EXEC2-Skript handelte, das diese Aktionen auslöste. Im Environment der Mail-Applikation wurde dieses Skript als solches erkannt und einfach ausgeführt; zu sehen war natürlich nur die Bildschirmausgabe — der Tannenbaum mit Schneegeriesel.

Eigentlich war dieser Wurm vollkommen harmlos und sicherlich auch so gedacht; wäre da nicht die unkalkulierte Tatsache, dass die damaligen "Weitverkehrsverbindungen" über Datex-P mit einer Bandbreite von 9.6 kbit/s abgewickelt (und nahezu ausschliesslich zum E-Mail Verkehr eingesetzt) wurden. Kurzum, der Wurm brachte die bescheidende E-Mail-Kommunikation zum Zusammenbruch, was eine teilweises Abschalten der Verbindungen erforderlich machte.

Melissa

Für die meisten Internet-Anwender brachte das Jahr 1999 die kopernikanische Wende in der E-Mail-Kommunikation. Am 26.3.1999 wurde das *Melissa*-Virus in Umlauf gebracht, das eine bis dahin unbekannte Epidemie von infizierten E-Mails auslöste. Das *Melissa*-Virus war — wie auch *der Christmas Tree Exec* — ein Makrovirus. Nur richtete er sich diesmal gegen die Windows-Betriebssyteme, auf denen die VBA (Visual Basis for Applications) zu finden ist; speziell die MS Word-Anwendung, die sich seit Word 95 (7.0) nicht mehr auf Word-Basic, sondern auf die einheitlichen VBA-Skriptsprache stützt. Auch hier analysierte das Virus das lokale Outlook Express Adressbuch und vervielfältigte sich über die ersten 50 hier gefundenen E-Mail-Adressen im Schneeballeffekt. *Melissa* fand viele Nachahmer, die nicht Word sondern Excel als Trägersystem missbrauchen. Es bleibt zu erwähnen, dass dieses Virus nicht nur Windows-Rechner sondern über das MacOS-Betriebssystem verbreiten kann, da dort ebenfalls Microsoft's Office im Einsatz sein kann.

Love-Letter-For-You

Erstaunlicherweise dauerte es mehr als ein Jahr (bis zum 4.5.2000) bis der nächste E-Mail-Wurm seine Runden machte: *ILOVEYOU*. Allerdings hatte das Internet zu diesem Zeitpunkt schon im Vergleich zu 1999 doppelt so viele Anwender, sodass diese simple Nachricht mit dem im Anhang befindlichen Visual Basic Skript grosse Verbreitung fand. Im Gegensatz zum Melissa-Virus hatte das "LoveLetter" jedoch auch eine schädigende Wirkung, indem es sich einerseits in der Windows-Registry festsetzte und andererseits Dateien vom Typ JPEG und MP3 überschrieb — (k)ein besonderer Liebesdienst. Schlimmer war jedoch die im eingegebene Eigenschaft, Passwörter auszuspionieren. Auch hier war Outlook und Outlook Express (unter eingeschalteter Option "Skripting aktivieren") das gewünschte E-

Mail-System zur Verbreiterung des Wurms.

7.3.1.2 Begegnung der zweiten Art ...

Nimda

Der *Nimda*-Wurm startet am 18.9.2001 seine Infektionswelle Er verbreitete sich zunächst als E-Mail in Form einer MIME "multipart/alternative" Nachricht, die aus dem (harmlosen und leeren) Teil eines MIME-Typ "text/html" und einem zweitem Teil vom MIME-Typ "audio/x-wav" bestand, die jedoch eine ausführbare Datei namens "readme.exe" enthielt.

Der *Nimda*-Wurm benutzte eine hybride Verbreitungsstrategie und war speziell auf die Schwachstellen der Microsoft Betriebsysteme und speziell des MS IIS[31] abgestimmt:

1. Erzeugung und Nutzung von Backdoors auf den befallenen Rechnern:. Freigabe lokaler Laufwerken (Shares) auf den befallenen PCs und Einrichtung eines Guest Account mit Administrator-Rechten. Der Wurm durchsucht alle erreichbaren Laufwerke, infiziert dort die .exe-Dateien und kopiert sich als riched20.dll in jene Laufwerke, die .doc oder .eml Dateien enthalten. Somit führen die Microsoft Anwendungen Word, Wordpad und Outlook die falsche riched20.dll aus, was wiederum zu weiteren Infektionen führte.
2. In die Verzeichnisse mit erkennbaren Web-Inhalten (HTML- oder ASP-Dateien) wird das Java-Skript
   ```
   <script language="JavaScript">
   window.open("readme.eml",    null,    "resizeable=no,
   top=6000,left=6000">
   </script>
   ```
   hinterlegt, sodass auch die Web-Clients infiziert werden, die sich diese Inhalte anschauen.
3. Das *Nimda*-Virus versucht, über den tftp-Port 69/UDP andere MS IIS zu erreichen und dort seinen *Malicious Code* unterzubringen. Zusätzlich werden auch Backdoors von älteren Viren (insbesondere von *Code Red* und von *sadmind/IIS*) gesucht und ausgenutzt.
4. Ferner scannt das Virus die lokalen Browser-Cache nach E-Mail-Adressen und versucht, zusätzliche Adressen über einen MAPI-Aufruf an den E-Mail-Client (Outlook) zu ergattern. An die so gefundenen Adressen werden im Abstand von 10 Tagen weitere infizierte E-Mail abgeschickt.

Der *Nimda*-Wurm stellt somit hinsichtlich seiner Verbreitungsstrategie und seiner Komplexität alle bislang "gebauten" Viren in den Schatten. Da er zugleich auch keine offensichtliche Schadensroutine beinhaltet, agierte er im Verborgenen und

[31] http://www.cert.org/advisories/CA-2001-26.html

infizierte in den nachfolgenden Tagen (und Monaten[32]), viele Internet-Server mit dem Microsoft IIS, die ihrerseits durch lange Antwortzeiten und die erhöhte Last im Datenverkehr auf die Anwesenheit des Virus aufmerksam machten.

Mit dem *Nimda*-Virus haben es die Viren-Hersteller gelernt, hocheffiziente Viren zu "bauen" und zugleich das Ziel verfolgt, eine Infrastruktur infizierter Rechner aufzubauen.

Sober & Co

Dass Visual Basic sich hervorragend zur Erzeugung neuer Viren nutzen lässt, gilt spätestens seit den Umtrieben des *Shakira*-Virus, der als Abkömmling des VBSWG Virus-Kits zu betrachten ist. Dieser auch *VBSWG.AQ* genannte Virus verbreitete sich ab Juni 2002 im Internet und köderte die Adressaten mit Bildern von Pop-Stars wie Shakira und Britney Spears oder dem Tennis-Idol Anna Kurnikowa, hinter denen aber ein VB-Skript steckte (z.B. ShakiraPics.jpg.vbs), das für die Weiterverbreitung mittels Outlook sorgte.

Als Weiterentwicklung der VBS-Würmer kann der *W32.Sober*-Virus angesehen werden, der am 24.10.2003 in Umlauf gebracht wurde. Dieser in MS Visual Basic programmierte Wurm beinhaltet allerdings seine eigene SMTP-Engine, die er nutzt, sowohl den textlichen Inhalt der Nachricht, als auch die Anhänge, sowie das "Subject:" der E-Mail zu variieren. Hierbei täuscht das Virus vor, selbst Opfer von z.B. Spam- oder Virus-Mails zu sein, oder von Anti-Virus-Organisationen zu stammen, indem er Inhalt und Absender entsprechend anpasst.. In Anhang der E-Mail findet sich die infizierte Datei mit der Endung .bat, .com, .exe, .pif, bzw. .scr.

Das Virus kopiert sich in das Windows-Stammverzeichnis als "Similare.exe" sowie zusätzlich in das Systemverzeichnis unter verschiedenen Namen. Durch einen Eintrag in der Windows-Registry

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
```

wird der Wurm bei jedem Windows-Neustart aktiviert und verschickt sodann weitere infizierte E-Mails an (die lokal ausspionierten) Empfängeradressen, die er in der Datei `%System%\Macromed\Help\Media.dll` hinterlegt.

Sircam

Der *W32/Sircam*-Wurm wurde zwar bereits am 25.7.2001 zum ersten mal registriert, findet aber in Wellen bis ins Jahr 2004 starke Verbreitung. Auch er tarnt sich mit unverdächtigem Inhalt. Der Anhang ist jedoch keine Graphikdatei, sondern — unter Ergänzung einer falschen Datei-Endung — eine ausführbare Datei. Hierbei macht er sich eine Eigenschaft der modernen Windows-Varianten zu nutzen, die die Ausgabe des vollen Namens einer Datei unterdrückt, falls die Endung "registriert" ist. Dem Anwender wird statt dessen ein (unverdächtiges) Icon im

[32] vgl. http://www.heise.de/newsticker/meldung/23259

Explorer bzw. Outlook dargeboten.

Der *W32/Sircam*-Wurm besitzt ebenfalls eine SMTP-Engine und verschickt weitere infizierte E-Mails an die Adressen, die er im Windows Address Book (*.wab) findet, bzw. die ihm über den Browser-Cache zugänglich sind.

Die Schadensroutinen des Wurm lauten "SCan32.exe" sowie "SirC32.exe", wobei die erste im Windows-Stammverzeichnis und die zweite im Windows Papierkorb abgelegt werden. Entsprechend wird der Wert der Windows Registry-Key Variablen `HKEY_CURRENT_USER\exefile\shell\open\command` durch `"[windows_drive]\recycled\SirC32.exe" "%1" "%*"` ersetzt und wird für die Registry-Variable `HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion/RunServices/Driver32` der Wert `%System%Scam32.exe` eingefügt.

Im Gegensatz zum *Sober*-Virus besitzt der *Sircam* jedoch eine Schadensfunktion: Im Papierkorb des Windows-Rechners wird eine Datei "sircam.sys" erzeugt, die den gesamten Speicherplatz dem (typischerweise) Laufwerk C: belegen kann und somit u.U. ein Abspeichern neuer Dateien unmöglich macht.

Sober und *Sircam* stellen in Abkehr zu den bisher bekannten Viren "psychologische" Würmer dar, indem sie den Adressaten dazu bringen, mit vermeintlich interessanten Informationen — basierend auf der Absenderadresse, des "Subjects:", des Inhalts und Aufmachung bzw. dem Anhang — den infizierten Code auszuführen.

7.3.1.3 Begegnung der dritten Art ...

Sobig.F

Die Attacke des Wurms *W32/Sobig.F* traf die Internet-Mail-Server am 20.8.2003. Auch hier sollte versucht werden, PCs unter den Windows-Betriebssystemen kompromittieren und ausführbare Dateien einzuschleusen. Im Falle von *Sobig.F* sind es `%System%winppr.exe` sowie `%System%winstt32.dat`[33]. Zusätzlich wird der Registry Key `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\Current\Run` mit dem Wert `%System%winppr.exe /sinc` versehen.

Sobig.F besitzt eine in "C" geschriebene SMTP-Engine, über die weitere infizierte E-Mails versandt werden. Neben dem Auslesen und Nutzung evtl. vorhandener E-Mail-Adressen öffnet das Virus über das Network Time Protocol (NTP) auf Port 8998/UDP eine Backdoor und versucht, sich mit zwanzig im Code hinterlegten IP-Adressen zu verbinden, um neue Instruktionen zu erhalten. Diese kontaktiert er Freitags und Sonntags zwischen 19:00 und 22:00 Uhr UTC. Es wird angenommen,

[33] http://www.cert.org/incident_notes/IN-2003-03.html

dass der Wurm seine Aktivität am 10.9.2003 eingestellt hat.

Wir betrachten einmal den Verlauf der Epidemie durch den *Sobig.F*-Wurm unter Nutzung der "virulator" Auswertungen meines Programmes QMVC (siehe übernächsten Abschnitt):

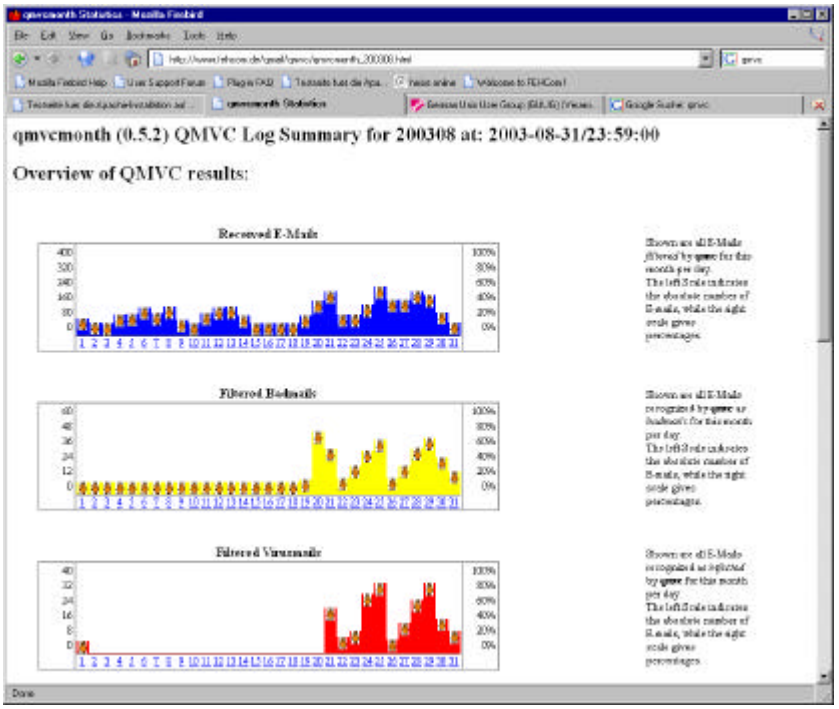

Abbildung 7.3-1: Verlauf des Befalls im August 2003 mit dem W32/Sobig.F Wurm.

Der infizierte Teil des Codes wurde als MIME-Attachment mit der Endung *.pif übertragen. Wir sehen im Diagramm "Filtered Badmails", dass am 20.8. QMVC anschlägt, indem zunächst 36 E-Mails mit verdächtigem MIME-Attachment festgestellt werden. Zu diesem Zeitpunkt sind die Virus-Patterns der eingesetzten AV-Scanner noch nicht aktualisiert, was dann am 21.8. erfolgt[34].

[34] vgl. http://www.heise.de/security/news/meldung/39648

Swen

Die *Sobig*-Epedime war noch nicht ganz vorbei, da begann der Angriff des Wurms *W32/Swen*. Beide Viren verhalten sich hinsichtlich des Windows-Betriebssystems sehr ähnlich; der Swen-Virus versucht jedoch (geschickt), bestehende Sicherheitsmassnahmen wie Virenscanner zu umgehen.

Wir betrachten zunächst den Verlauf der Epidemie, die am 19.9.2003 beginnt:

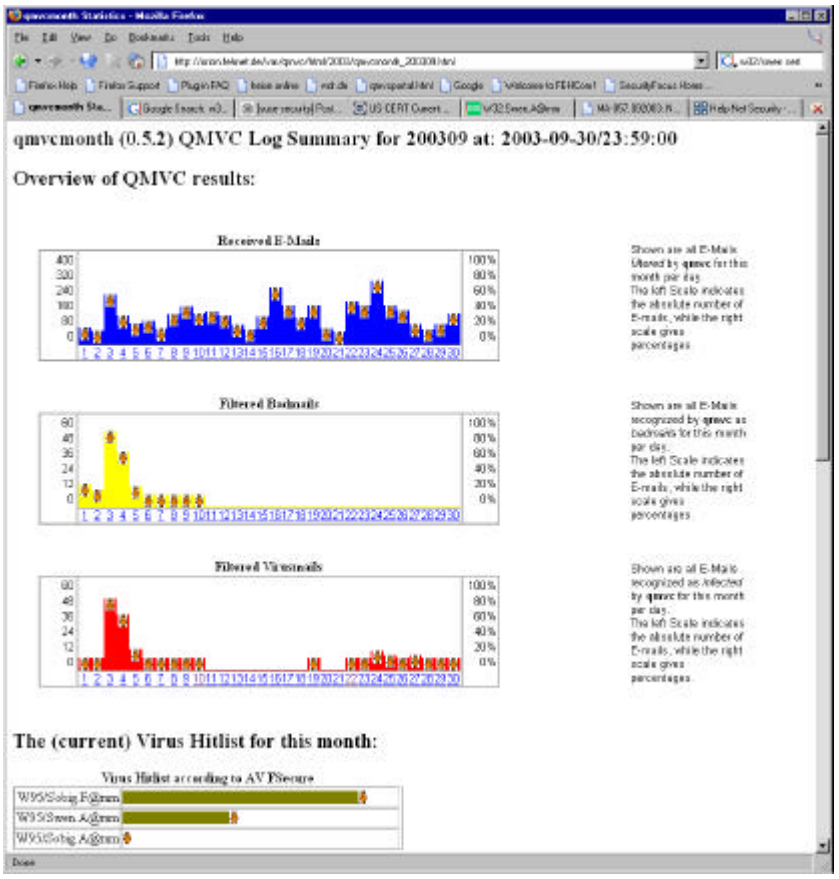

Abbildung 7.3-2: Die W32/Swen.A Epidemie im September 2003 — in der QMVC "virulator" Auswertung.

Der erste Virenscanner (F-Secure) schlägt am 19.8. zum ersten mal an. Erstaunlicherweise erkennt aber QMVC keine zugehörigen (und zu filternden) MIME-Attachments vom Typ "MS-DOS/MS-Windows executable".

Stealth Viren

Der Wurm *W32/Swen.A* ist der erste "Transport Tarnkappen" Virus! Nach der *Sircam*- und *Sober*-Attacke haben viele Administratoren auf ihren E-Mai-Systemen ein MIME-Filter aktiviert, dass die Flut von infizierten E-Mails "pro-

aktiv" filtern soll; wie dies z.B. Russell Nelson und Charles Cazabon für Qmail getan haben (siehe weiter unten).

Dem versucht der *W32/Swen.A*-Wurn zu entgehen, indem er

1. *aktiv* versucht, bestehende Anti-Viren-Scanner auf dem betroffenen PCs zu deaktivieren, und er
2. *passiv* statt einer einfachen MIME-Verkapselung nun den Anhang mit der ausführbaren Datei zunächst doppelt, später sogar dreifach MIME-kodiert. Hierdurch greifen Massnahmen zur binären Identifizierung des MIME-Types nicht mehr. Zugleich wird ein Outlook-Feature ausgenutzt, das diese mehrfach verpackten MIME-Anhänge dennoch korrekt umsetzt und ggf. ausführt.

MyDoom

Die *W32/Swen*-Epideme ebbte im Laufe des Dezember 2003 ab und wurde durch den *MyDoom*[35] (auch *W32/Novarg.A* bzw. *W32/Shimg* genannt) Virus abgelöst, der am 27.1.2003 losgetreten wurde. Das *MyDoom*-Virus verbreitet sich neben E-Mail-Übertragung vor allem über das Tausch-Netzwerk KaZaA.

Ist der Rechner einmal infiziert, führt das Virus folgende Schritte durch:

1. Infizierte E-Mails werden verschickt und der Empfänger versucht zu täuschen, indem sich der Inhalt als "technische" Mitteilung ausgibt. Der Anhang mit dem eigentlichen Virus-Programm wird teilweise in ZIP-Form übertragen. Der Name des Attachments sowie die SMTP-Absenderadresse ("Mail From:") wird beliebig gewählt.
2. Auf den TCP-Ports 3127 bis 3198 wird eine Backdoor geöffnet.
3. Durch Einträge in die *hosts*-Datei werden falsche IP-Adressen für die Server der bekannten Anti-Viren-Hersteller vorgenommen, sodass ein möglicher Pattern-Update der Virenscanner ins Leere laufen soll.
4. Das Virus kopiert sich in den Ordner `C:\Program Files\KaZaA\My Shared Folder\`, der für KaZaA-Benutzer zum Download freigegeben ist.
5. Es installiert sich selbst als "Taskmon.exe" im Windows-Systemverzeichnis und trägt einen entsprechenden Registry-Key `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run` mit dem Wert `"TaskMon" = %sysdir%\taskmon.exe` ein.

UPX-Tarnkappe

Als neue Tarnkappen-Variante, verpackt das *MyDoom*-Virus seinen infiziösen Code im Windows UPX-Format[36] in der Datei `%sysdir%\shimgapi.dll`. Zugleich werden diese Dateien per E-Mail übertragen, was dazu führt, dass ihre MIME-Type nicht mehr als "ausführbar" erkannt wird. Das UPX-Format verhält

[35] http://www.cert.org/incident_notes/IN-2004-01.html
[36] http://upx.sourceforge.net/

sich unter Windows vergleichbar einer "Selfextracting" Zip-Datei.

DDoS-Attacke gegen Microsoft und SCO

Die Variante *MyDoom.B* führte zusätzlich einen gezielten (Distributed) Denial-of-Service gegen die Sites der Firmen Microsoft und SCO durch. Einerseits dienten die E-Mail-Server dieser Firmen als direktes Ziel des Angriffs, andererseits waren die an beliebige Targets verschickten E-Mails z.B. mit dem SMTP-Absender "Mail From: <support@microsoft.com>" versehen. Einige Anti-Viren-Programme verschicken daraufhin prompt eine Benachrichtigung an den (gefälschten) Absender. Diese Notifikationen werden also selbst zum Teil den DDoS-Angriffs. Als Reaktion darauf, nahm die Firma SCO 1. Februar ihre Server vorübergehend vom Netz.

NetSky

Der *NetSky*-Wurm ist ursächlich für die bislang letzte grössere E-Mail-Epidemie (März 2004). *NetSky* nutzt praktisch alle bislang aufgeführten Verbreitungswege und zeichnete sich durch eine überraschende Vielfalt von Varianten aus (*NetSky.A* bis z.Z. *NetSky.R*). Ziel der *NetSky.Q*-Variante des Virus sind einige Tauschbörsen- und Cracker-Netzwerke (`www.kazaa.com`, `www.edonkey2000.com`, `www.cracks.st`, `www.cracks.am` und `www.emule-project.net`), die es zwischen dem 8. und 11. April 2004 mit E-Mails "versorgte".

Interessanterweise verbirgt sich im *NetSky.Q*-Virus folgende Nachricht an die Betroffenen[37]:

> *"We are the only SkyNet, we don't have any criminal inspirations. Due to many reports, we do not have any backdoors included for spam relaying. and we aren't children. Due to this, many reports are wrong. We don't use any virus creation toolkits, only the higher language Microsoft Visual C++ 6.0. We want to prevent hacking, sharing with illegal stuff and similar illegal content. Hey, big firms only want to make a lot of money. That is what we don't prefer. We want to solve and avoid it. Note: Users do not need a new av-upgrade, they need a better education! We will envelope... - Best regeards, the SkyNet Antivirus Team, Russia 05:11 P.M"*

Viren-Baukästen

Es erscheint naheliegend, dass die Virenhersteller sich eines immer komplexeren Codes bedienen und zudem genauestens die Schwachstellen der diversen Systeme ausnutzen, seien es die bekannten Bugs im Windows-Betriebssystem und dessen Netzanwendungen, aber auch die der Anti-Viren-Hersteller und der bekannt gewordenen Gegenmassnahmen.

Agobot und Phatbot

---

37 `http://www.sophos.com/virusinfo/analyses/w32netskyq.html`

Mittlerweile kursieren im Internet regelrechte "Viren-Baukästen", die unter den Namen *Agobot* und *Phatbot* bekannt geworden sind[38]. Offenkundig liegt nun das Ziel der Viren-Hersteller in der Störung der Internet-Kommunikation bzw. der gezielte Angriff gegen bestimmte Internet-Sites.

7.3.1.4 Die Zukunft: Viren- und Spam-Hersteller Hand-in-Hand

In der Ausgabe 5/2004 t die Zeitschrift ***c't*** nachgewiesen, dass Virenhersteller mit Spammern kooperieren. Der infizierte Rechner besitzt nicht nur eine geöffnete Backdoor, über die entsprechende Spam Nachrichten eingeschleust werden, sondern über die verfügbare SMTP-Engine des Viruses kann dieser PC auch mit grossem Erfolg und Wirkungsgrad zum Verschicken von Spam E-Mails genutzt werden (Spam-Bot). Einige Bespiele hierzu finden sich auch in meinen QMVC-Logfiles.

Für den Betroffen ändert sich beim Empfang solcher Spam E-Mails somit nicht nur klassischerweise der (SMTP-)Absender und — in gewissen Grenzen — der Inhalt der Nachricht, sondern man sieht sich (speziell als E-Mail-Provider) auch der Tatsache gegenüber stehend, dass die Herkunft nicht feststellbar ist. Hierdurch laufen auch übliche Methoden zur Spam-Bekämpfung — wie den Abruf einer *Realtime Blacklist* (RBL) sowie das Filtern auf die SMTP-Absenderadresse in "Mail From:" — ins Leere. Ging in der Vergangenheit bisher eine typische Spam E-Mail von *einem* sendenden MTA an *N* Zieladressen, so können nun die *M* infizierten PCs dazu gebracht werden, in einem Zug E-Mails an *P* Zieladressen zu verschicken. Die Zieladressen brauchen zudem nicht mehr zufällig generiert werden, sondern lassen sich aus dem Cache bzw. dem Adressbuch des infizierten Rechners entnehmen.

7.3.2 Anti-Virus-Tools

Mitte der neunziger Jahre fingen die ersten Hersteller von Anti-Viren-Produkten für DOS an (McAfee und Dr. Solomon), ihre Werkzeuge auch für Unix bzw. das damals noch recht neue Linux bereitzustellen. Viren-Scanner sind immer Dateien-Scanner. Im Grunde kann eine infizierte Datei durch (mindestens) eine der folgenden Kriterien festgestellt werden

- *Statischer Test*: Die Checksumme der Datei stimmt mit einer der im AV-Scanner bekannten Checksummen für infizierte Dateien überein.
- *Generischer Test*: Die Binär-Datei wird hinsichtlich eines oder mehrer bekannter Infektionsmuster untersucht.
- *Heuristischer Test*: Bei Skripten werden bestimmte Befehlspassagen erkannt, die im AV-Scanner hinterlegt sind.

---

[38] http://www.heise.de/newsticker/meldung/46634

Im jedem Fall muss der AV-Scanner eine Analyse-Routine (Scan-Engine) einer Datenbasis mit Viren-Informationen (Pattern-Database) aufweisen.

Attachments

Will man die Viren-Scanner für E-Mails benutzen, müssen diese so aufbereitet werden, dass die Anhänge einzeln vorliegen. Prinzipiell lassen sich drei Möglichkeiten zur Enkodierung der Anhänge einsetzen

- MIME-Format (RFC 2045) — dies ist das Standard-Verfahren und nutzt eine Base64 Kodierung. Zur Dekodierung eignen sich die Programme **reformime** aus dem Maildrop-Packet[39] oder das Programmpaket ripMIME[40] von Paul L. Daniels. Das ältere Programm **metamail**[41] ist nicht in der Lage, MIME-Attachments von Typ "multipart/alternative" richtig zu behandeln und sollte daher vermieden werden.
- BinHex-Enkodierung — wurde in der 7-bit SMTP-Zeit von (älteren) E-Mail Clients angewandt. Zum Dekodieren kann das Standard-Unix-Tool **uuencode** eingesetzt werden.
- MS-TNEF — ein proprietäres Microsoft Format, das im Zusammenspiel mit der MIME-Kodierung einsetzt wird; Anhänge werden in einer speziellen Datei "winmail.dat" transportiert. Mittels des Programms **tnef**[42] das auf dem Sourceforge-Server zu finden ist, können diese Attachments im Anschluss an die MIME-Dekodierung erfolgreich behandelt werden.

Zusätzlich kann die im Anhang befindliche Datei auch noch ein-"gepackt" sein. Hier sind alle Verfahren gängig, angefangen von den Windows-typischen Verfahren wie Zip, ARJ und Microsoft's CAB-Verpackung von installierbaren Dateien über UNIX-spezifische Methoden wie bzip, tar etc..

scanmails & Co

Die ersten verfügbaren AV-Scanner für Unix waren Ausgangspunkt für die Entwicklung von Anti-Virus-Tools für E-Mails. Mogens Kjaer vom Carlsberg Laboratory und Jürgen Quade (damals Softing GmbH) stellten das **scanmails** Skript zusammen, das in der Zeitschrift ***iX*** im Februar 1998 veröffentlicht wurde[43]. **scanmails** wurde entweder (systemweit) in die `sendmail.cf` eingeklinkt oder aber via **procmail** auf Benutzer-Basis aufgerufen.

scanmails wurde von Rainer Link (SuSE) zu Tool AMaViS (A Mail Virus Scanner) als PERL-Skript weiter entwickelt. Von Paul L. Daniels stammt das Derivat **inflex**, das auch als Ausgangspunkt meiner QMVC Entwicklung diente. An AMaViS angelehnt ist der spezifisch für Qmail angepasste Qmail-Scanner, der zunächst

---

[39] http://www.flounder.net/~mrsam/maildrop/

[40] http://www.pldaniels.com/ripmime/

[41] http://bmrc.berkeley.edu/~trey/emacs/metamail.html

[42] http://sourceforge.net/projects/tnef/

[43] http://www.heise.de/ix/artikel/1998/02/136/

unter dem Namen Scan4Virus bekannt wurde.

Eine von diesen Tools unabhängige Entwicklung hat Len Budney vorgenommen, dessen Qscanq komplett in C vorliegt und gemäss Dan Bernstein's Software-Paketierung-Empfehlungen zu installieren ist.

Virus++

Bedingt durch die starke Verbreiterung der Viren und Würmer über E-Mails in den letzten Jahren, kann von einem regelrechten Boom der Anti-Viren-Hersteller gesprochen werden. Mittlerweile tummelt sich eine (wachsende) Vielzahl von Anbietern auf dem Markt, der vielsprechende Wachstumsraten aufweist. Bei der Bewertung der AV-Scanner unter UNIX sind folgende Gesichtspunkte relevant:

- Wie effektiv ist die Scan-Engine des Herstellers ?
- Welche Eigenschaften bringt die Scan-Engine mit sich; kann sie z.B. gepackte Dateien zuerverlässig scannen ?
- Wie gross ist der Resourcen-Verbrauch der Scan-Engine; unterstützt sie z.B. einen Client/Server Mode ?
- Wie häufig sind Updates der Scan-Engine zu erwarten ?
- Wie schnell, zuverlässig und problemlos ist das Update der Viren-Pattern ?
- Welches Lizenzmodell liegt dem AV-Scanner zugrunde ?

Häufig wird es nicht als ausreichend betrachtet, lediglich einem AV-Scanner zu vertrauen, sondern es finden zwei oder drei parallel Einsatz. Das geht natürlich sowohl auf der Kosten der Resourcen als auf des Portemonnaies. Daher geniesst der Public Domain Virenscanner Clam AV[44], der von einer polnischen Entwickler-Crew rund um Tomasz Kojm entwickelt und gepflegt wird, mittlerweile grosse Beachtung. Im Gegensatz zu den kommerziellen AV-Produkten, deren Fokus zunächst die Windows-Version darstellt, zielt Clam AV vor allem auf den Unix-Anwender. Dies gilt nicht nur das Basis-Produkt (**clamscan**), sondern auch die Verfügbarkeit der Pattern-Updates.

Viren Pattern-Updates

Die Pattern-Updates müssen schliesslich nur erstellt, sondern auch Plattform-spezifisch verfügbar gemacht werden. Es nutzt dem Unix-Anwender nichts, wenn der Hersteller zwar einen exzellenten Support für die Windows-Variante bereitstellt, aber die Unix/Linux-Pattern erst Stunden oder Tage später verfügbar sind. Gleiches Kriterium gilt für die Erreichbarkeit der Server, über die der Download der Viren-Patterns erfolgt. In diesesm Zusammenhang ist es günstig, wenn nicht die Viren-Pattern in Gänze heruntergeladen werden müssen (in der Regel sind dies 5 bis 8 Mbyte), sondern ein differentieller Upgrade erfolgen kann. Dies beschleunigt nicht nur den Download, sondern erhöht auch die Verfügbarkeit der Download-Server, da diese in der Regel nur eine beschränkte

[44] http://www.clamav.net/

Anzahl simultaner FTP- bzw. HTTP-Downloads zulassen.

An diese Stelle erlaube ich mir, eine kritische Stimme bezüglich der kommerziellen Anti-Viren-Software-Anbieter zu zitieren[45]:

> *Anti-virus companies love to get people worked up into a lather about all the virus threats out there, because it helps them sell more product. So, it wasn't much of a surprise that, following the "big" virus and trojan horse problems in August [2003, eh.], the anti-virus "experts" started warning that this was just a prelude to something worse, and that we should expect even more virus problems as soon as the current viruses died out. While it is good to keep users vigilant about virus things, these announcements served more to make people ignore the real problem: the anti-virus companies failed. Of course, you don't hear them speaking up now about the fact that their original predictions of "the next wave" of viruses immediately following the last wave appears not to have come true. Especially with the SoBig virus, we were told that the next version was supposed to appear in September sometime, but that never happened. Of course, it's good when we don't have virus outbreaks - and I have no doubt that they will come again - but once again we have a situation where the anti-virus folks seemed to hype things up beyond necessary.*

7.3.2.1 Einsatz von Anti-Virus-Tools unter Qmail

Qmail bietet durch seinen modulare Aufbau günstige Bedingungen für den Einsatz beliebiger Anti-Viren-Scanner. Hierbei muss auch nicht unbedingt auf das QUEUE_EXTRA Patch von Bruce Guenter zurückgeriffen werden. Im Grunde bieten sich drei Möglichkeiten an:

- Alle E-Mails (ein- und auslaufende) sollen Vieren-gescannt werden. Hierzu muss statt des Programms **qmail-queue** ein Wrapper eingesetzt werden, der zunächst den Viren-Scanner aufruft, bevor die E-Mail (bei negativem Überprüfungsresultat) in die eigentliche Queue gesteckt wird.
- Lediglich die per SMTP einlaufenden E-Mails (in der Regel also aus dem Internet) sollen auf Viern überprüft werden. Dann ist die Nutzung des QUEUE_EXTRA Patches obligatorisch.
- Es werden nur die lokal zuzustellenden E-Mails überprüft. Dies geschieht mittels dedizierter `.qmail` Dateien im Heimat-Verzeichnis des E-Mail-Empfängers.

Wir betrachten für ersten beiden Fälle einen generischen Ansatz unter Einsatz des Public Domain AV-Scanners Clam AV. Fall Drei möchte ich später bei der

[45] http://www.stargeek.com/item/101432.html

Vorstellung meines QMVC diskutieren.

qmail-queue Wrapper

Es ist zunächst ein Wrapper-Skript für das Programm **qmail-queue** zu erstellen. Häufig wird hierzu ein PERL-Skript eingesetzt; doch ein einfaches aber cleveres Shell-Skript verrichtet mindestens die gleichen Dienste (Listing 7.3-1):

```
#!/bin/sh
# chmod 1755 qmail-queue.scan
#  mkdir /var/qmail/tmp; chown qmailq:qmail /var/qmail/tmp;
chmod 1777 /var/qmail/tmp
QMAIL_TMP="/var/qmail/tmp"
QMAIL_BIN="/var/qmail/bin"
cat > $QMAIL_TMP/msg.$$
if [ $? -ne 0 ]; then
        exit 53
fi
VIRUS=$(clamscan $QMAIL_TMP/msg.$$ --mbox --infected --no-
summary --tempdir="$QMAIL_TMP" --stdout)
case $? in
        0)      $QMAIL_BIN/qmail-queue < $QMAIL_TMP/msg.$$;
RC=$?;;
        1)      exec 1>&2; echo "Infected email not delivered
($VIRUS)"; RC=31;;
        *)      exec 1>&2; echo "clamscan internal error
($RC)"; RC=81;;
esac
rm $QMAIL_TMP/msg.$$
exit $RC
```

*Listing 7.3-1: Einfaches **qmail-queue** Wrapper Skript.*

Das Skript trägt zunächst den Namen **qmail-queue.scan** und ist unter dem User *qmailq* und der Gruppe *qmail* zu erzeugen. Wichtig sind die Dateirechte, die 4711 betragen müssen:

```
-rwxr-xr-t  1 qmailq  qmail    586 May  3 17:03 qmail-
queue.scan
```

Unter `/var/qmail/` ist ein anschliessend unter dem User *qmailq* und der Gruppe *qmail* ein temporäres Verzeichnis `tmp` einzurichten, das als "staging area" fungiert und sowohl zur Ablage der einlaufenden E-Mails ("seekable files") als auch für den Virenscanner **clamscan** gebraucht wird:

```
drwxrwxrwt   2 qmailq  qmail    1024 May  3 17:35 tmp
```

Scannen aller E-Mails

Will man alle E-Mails scannen, muss dieses Skript das eigentliche Programm **qmail-queue** ersetzen. Hierzu wird zunächst im Verzeichnis `/var/qmail/bin/`

die Datei **qmail-queue** nach **qmail-queue.org** umbenannt. Anschliessend wird ein Link erzeugt, der **qmail-queue.scan** nach **qmail-queue** dirigiert. Im Skript selbst ist dann der Name **qmail-queue** durch **qmail-queue.org** zu ersetzen.

Das Resultat sieht dann z.B. unter Verwendung der EICAR-Testmail von QMVC folgendermass aus:

```
# cat /var/qmvc/mails/testmail.body.plain |
/var/qmail/bin/mailsubj "qmail-queue Test" erwin
Infected email not delivered (/var/qmail/tmp/msg.37218: Eicar-
Test-Signature FOUND)
qmail-inject: fatal: mail server permanently rejected message
(#5.3.0)
# echo $?
100
```

Wurde das tmp-Verzeichnis mit zu eingeschränkten Schreibrechten ausgestattet, erhält man folgende Mitteilung:

```
$ cat /var/qmvc/mails/testmail.mime.uuencoded |
/var/qmail/bin/mailsubj "qmail-queue Test" erwin
bin/qmail-queue[7]: cannot create /var/qmail/tmp/msg.37296:
Permission denied
```

Scannen der per SMTP einlaufenden E-Mails

Um lediglich die vom Internet einlaufenden E-Mails auf Viren zu scannen, ist zunächst die Installation des QUEUE_EXTRA Patches von Bruce Guenter[46] für Qmail 1.03 notwendig. Das QUEUE_EXTRA Patch kann auch mittels meines SPAMCONTROLs aktiviert werden. Auch die *netqmail*-Distributionen bringt diese Ergänzung per Default mit.

Die typische Nutzung geschieht über die **tcpserver** Konfigurationsdatei (z.B. `tcp.smtp`):

```
:allow,QMAILQUEUE='/var/qmail/bin/qmail-queue.scan'
```

Hierdurch wird statt des Programms **qmail-queue** zunächst das oben beschriebene Wrapper-Skript ausgeführt. Läuft nun ein Virus via SMTP ein, macht er sich im **qmail-smtpd** Logfile bemerkbar. Dies soll in einem Auszug im Zusammenhang mit dem Logging meines SPAMCONTROL-Patches dargestellt werden:

```
2004-04-28 16:27:12.954615500 tcpserver: pid 35640 from
192.168.192.1
2004-04-28 16:27:12.956461500 tcpserver: ok 35640
qmailer.fehnet.de:192.168.192.2:25
orion.fehnet.de:192.168.192.1::4337
2004-04-28 16:27:12.961530500 Accept::RCPT::Recipients_Rcptto:
```

[46] http://qmail.org/qmailqueue-patch

```
MailFrom: <feh@fehcom.de> RcptTo: <erwin@qmailer.fehnet.de>
2004-04-28 16:27:15.677699500 Infected email not delivered
(/var/qmail/virus/msg.35641: W32.Yaha.g.dam FOUND)
2004-04-28 16:27:15.681289500 tcpserver: end 35640 status 0
```

Der SMTP-Sender erhält im Gegenzug die ablehnende SMTP-Fehlermeldung: "`554 mail server permanently rejected message (#5.3.0)`".

7.3.2.2 AMaViS

AMaViS hat bereits eine bewegte Geschichte hinter sich. Die "offizielle" Web-Seite[47] beschreibt die Version 0.2.1 des Tools, dessen Entwicklung allerdings bereits im Jahr 2000 eingestellt wurde. Hierbei war es notwendig, **qmail-remote** und **qmail-local** durch entsprechende AMaViS-Wrapper zu ersetzen; kein unbedingt sinnvolles Unterfangen. AMaViS wurde ursprünglich für den MTA *Sendmail* entwickelt und stützt sich auf die Skript-Sprache PERL.

Eine der zentralen Ideen von AMaViS ist, die Nachricht komplett auszupacken; d.h. nicht nur die MIME- bzw. BinHex-Attachments herauszulösen, sondern auch noch die Archive zu entpacken. Dies bedingt natürlich die Verfügbarkeit entsprechender (Ent-)Pack-Programme, die neben **reformime** und **file** auf der "Requirements" Liste stehen. Problematisch ist dieser Ansatz für mehrfach verschachtelte Archive und solche mit beliebiger Grösse (x42.zip). Einige Packer-Programme hinterlegen im Archiv-Header keine Angaben über die ursprüngliche Grösse der Datei, sodass aus zunächst wenigen hundert Byte im gepackten Archiv sich ausgepackt schnell einige Gigabyte ergeben können ...

Neben der Tatsache, dass AMaViS praktisch alle gängigen Virenscanner unterstützt, arbeitet es auch eng mit dem ebenfalls in PERL geschriebenen SpamAssassin zusammen, sodass dieser nicht noch separat aufgerufen werden muss.

AMaViS Derivate

Eine aktuelle Version (0.3 — AMaViS-ng[48]) findet sich bei Sourceforge. Dies ersetzt das ursprüngliche PERL-Module amavis-perl. Auch ein **amavisd** Daemon-Modul wird hier zum Download angeboten.

Zudem gibt es für Qmail unter FreeBSD ein in C geschriebenes AMaViS Derivat qmail-amavisd[49]. Eine weitere Entwicklung stellt amavisd-new[50] dar, die am J. Stefan Institut in Slowenien gepflegt wird

---

[47] http://www.amavis.og/

[48] http://sourceforge.net/projects/amavis

[49] http://freebsd.over.ru/qmail-amavisd.html

[50] http://www.ijs.si/software/amavisd/

7.3.2.3 Qmail-Scanner

Der Qmail-Scanner[51] kann ebenfalls über Sourceforge bezogen werden. Der Qmail-Scanner erlaubt eine genaue Analyse der E-Mail und ein präzises Filtern entsprechender E-Mails, falls entsprechende Kriterien vorliegen. Ferner bietet er auch Unterstützung für SpamAssassin. Auch der Qmail-Scanner ist in PERL geschrieben und verlangt zusätzliche PERL-Module sowie das Programm **reformime**.

Wie auch der oben beschrieben **qmail-queue** Wrapper, klinkt sich der Qmail-Scanner in die Queue ein, was typischerweise über das QUEUE_EXTRA Patch realisiert wird. Auch der Qmail-Scanner untersützt praktisch alle am Markt verfügbaren AV-Scanner für Unix/Linux.

PERL-Abhängigkeiten

Problematisch am Qmail-Scanner ist die Abhängigkeit von der PERL-Version sowie die hiermit verbundene Ressourcen-Nutzung. Schliesslich muss bei jeder einlaufenden E-Mail der gesamte PERL-Interpreter (plus die zugehörigen Module) in den Speicher geladen werden. Wird der **qmail-smtpd** Daemon per **softlimits** beschränkt (siehe Kapitel 5), ist der Wert für den Speicherbedarf entsprechend kräftig aufzustocken. Es gibt zwar mittels des Compilers **perlcc** die Möglichkeit, ein PERL-Skript zu übersetzen und als ausführbare Datei zu erzeugen; aufgrund der unterschiedlichen PERL-Versionen kann dies jedoch nicht immer garantiert werden.

7.3.2.4 QMVC

QMVC[52] ist eine Entwicklung, die bewusst auf die Eingriffe in die Qmail-Queue verzichtet und sich auf die lokale Zustellung mittels geeigneter dot-qmail Dateien beschränkt. Hierbei unterstützt QMVC sowohl den Qmail *virtualdomains*- als auch den **qmail-users**- Mechanismus.

Neben der Einbindung beliebiger (und bis zu vier simultan einsetzbarer) Anti-Virenprogramme, erlaubt QMVC auch das Filtern von E-Mails hinsichtlich des "Subjects", des Nachrichteninhalts und der Anhänge. MTAs, die mehrere (virtuelle) Domains hosten (z.B. mittels Vpopmail oder des Vmailmgr) können pro Domain spezifische QMVC-"Policies" vergeben, in denen definiert wird, welche Virenscanner und welche Filter (mit welchen Kriterien) aktiv sein sollen.

Loader-Type Filter

Zusätzlich zu den üblichen Dateiname und MIME-Type-Filter (letzere aufgrund des"magic"-Tests) beinhaltet QMVC in der Version 1.7 eine neue Filtertechnologie, die auf der Erkennung ladbaren Codes (Loader-Type) basiert. Ein ausführbares Programm (kein Skript!) muss eine betriebssystem-spezifische Loader-Anweisung beinhalten, um überhaupt ausgeführt werden zu können.

---

[51] http://qmail-scanner.sourceforge.net/

[52] http://www.fehcom.de/qmvc.html

Unter den heutigen Windows-Systemen ist das überlicherweise "Kernel32.dll". Erkennt **qmvc** in der Binärdatei dieses Statement handelt es sich mit Sicherheit um eine unter Windows ausführbare Datei, unabhängig von MIME-Type oder evtl. Dateiendungen. Das Filter lässt sich auch für andere Betriebssyteme, wie z.B. MacOS konfigurieren und bietet so einen zusätzlichen Schutz, der auch von "Stealth-Viren" nicht umgangen werden kann.

Zudem verfügt QMVC über ausgezeichnete Analyse-Werkzeuge. Hierdurch erhält der System- bzw. Mail-Administrator einen umfassenden — und bedarfsweise aktuellen — Überblick über die "Viren"- und "Badmail"-Situation.

Als Hilfsprogramme benötigt das Korn-Shell-Skript **qmvc** Dan Bernstein's mess882-Paket, das Programm **reformime** (aus dem Maildrop-Paket), sowie die Werkzeuge **uudecode** und **file**.

QMVC installiert sich im Verzeichnis `/var/qmvc/` und der Aufruf des Skriptes **qmvc** findet einfach per dot-qmail Datei statt:

```
  /.qmail:
|qmvc -n
./Maildir/
```

Das Flag "-n" steht für Nutzung der "noscantypes" Information. Die Idee ist, auf das (ressourcen-aufwändige) Virenscannen gänzlich zu verzichten, falls die E-Mail nur (identifizierte) "harmlose" Anhänge aufweist.

Für virtuelle Domains (deren Abwicklung bei Qmail über einen zugeordneten Benutzer-Account erfolgt) ergibt sich typischerweise folgender Eintrag in die zugehörigen dot-qmail Dateien unter Einsatz von Vpopmail:

```
# qmail + vpopmail + vdelivermail
|/usr/local/bin/qmvc -nvu
|/usr/local/vpopmail/bin/vdelivermail '' bounce-no-mailbox
```

Hierbei wird **qmvc** über das Flag "-v" informiert, den Account-Prepend bei evtl. Notifikationen an den Empfänger zu entfernen und das Flag "-u" sorgt dafür, dass **qmvc** seine Arbeit nicht in Bezug auf das Stammverzeichnis `/var/qmvc/` erledigt, sondern Benutzer- (User-)spezifisch im Verzeichnis `~/qmvc/` arbeitet. Hierdurch greifen nicht nur evtl. Quota-Regeln für den Benutzer, sondern es kann auch ein spezifisches QMVC-"Profil" (im Verzeichnis `~/qmvc/control/`) festgelegt werden, das Vorrang für den systemweiten Einstellungen hat. Ferner wird das gesamte Logging lokal vorgenommen; entsprechendes gilt auch für die spätere Auswertung (vgl. Abbildung 7.3-1 und 7.3-2).

7.3.2.5 Qscanq

Qscanq[53] ist eine relative neue Entwicklung von Len Budney. Wie auch Dan Bernstein, gibt er eine Security-Garantie für sein Produkt; allerdings erst ab der Version 1.0, von der allerdings die jetzige Version 0.42 noch ein Stück weit entfernt ist.

Die vorliegende Version von Qscanq nutzt das Tool **ripMIME** um die Anhänge in den E-Mails auseinanderzupflücken und startet dann (per Default) den AV-Scanner H+BEDV **antivir**. Auch andere Virenscanner lassen sich über ein Umbrella-Programm problemlos einbinden. Bei der Installation von Qscanq das Programm **qmail-queue** durch **qscanq** (ebenfalls ein C-Programm) ersetzt. **qscanq** kann natürlich auch über das QUEUE_EXTRA Patch für den Einsatz bei **qmail-smtpd** eingebunden werden.).

Dadurch, dass das ausführbare Programm **qscanq** nicht im gleichen Dateisystem voliegt wie Qmail, sondern typischerweise unter **/package/mail/qsanq/command/qscanq**, wird die Installation eindeutig verkompliziert. Es müssen eine Reihe von Soft- und Hardlinks erzeugt werden und das gesamte Queue-Verzeichnis wird nun nach `/var/qmail/bin/qscanq/` fest verlinkt (hard link).

Qscanq nimmt im Gegensatz zu den anderen AV-Tools keine Benachrichtigung an den Absender bzw. Empfänger der E-Mail vor; ein Verfahren, das sich mittlerweise eingbürgert hat und sinnlose Benachrichtigungen bei gefälschten Absenderadressen zu unterbinden. Auch Qscanq erlaubt die Einbindung von SpamAssassin.

7.3.3 Pro-Aktives Filtern von E-Mails

Die Diskussion der Anti-Virus-Tools im vorausgegangen Abschnitt hat deutlich gemacht:

- Das Scannen der E-Mails auf Viren ist in jedem Fall Ressourcen-aufwändig.
- Das Scannen der E-Mails kann fehlerträchtig sein; die Virenhersteller versuchen bewusst, die E-Mail teilweise so zu verstümmeln, das eine Analyse ins Leere läuft.
- Das Scannen der E-Mails kann zu DOS-Angriffen genutzt werden; dies gilt besonders dann, wenn ein **qmail-queue** Ersatz genutzt wird und die "staging area" im Qmail-Verzeichnis liegt.
- Beim Scannen der E-Mails gibt es nicht nur prinzipielle Abhängigkeiten zum Virusscanner sondern auch zu den Hilfprogrammen wie **reformime**, **file** und

[53] http://mysite.verizon.net/vze1ypud/software/qscanq/

evtl. auch der Skript-Sprache.

Hinsichtlich des letzten Punktes habe ich z.B. das Programm **qmvc** bewusst als KSH-Skript ausgelegt. Andere Programme wie der Qmail-Scanner und AMaViS nutzen PERL, was (meines Erachtens) keine gute Wahl ist, da hier zusätzlich eine starke Abhängigkeit von der PERL-Version besteht, die kaum überblickt werden kann. Daher besteht die generelle Tendenz, die Anti-Viruswerkzeuge als C-Programme zu implementieren.

Unabhängig wie immer die Bewertung der Risiken und Möglichkeiten der Virenabwehr aufgrund der vorgestellten Tools erfolgt, so erscheint doch ein pro-aktiver Filtern evtl. gefährlicher E-Mails bzw. Anhänge in E-Mails der geeignetere. In diesem Zusammenhang erscheint es logisch und folgerichtig, generell die Übertragung ausführbare Binärdateien (speziell für Windows) zu unterbinden. Dies aus naheliegenden Gründen:

- Mit Ausnahme von Software-Entwicklern braucht eigentlich niemand ausführbare Datein per E-Mail zu übertragen. Zum Dateientranfer bieten sich schlicht andere Möglichkeiten an, wie z.B. ein Upload-Server.
- Sollte dennoch einmal ausnahmsweise eine ausführbare Datei übertragen werden, bietet sich an, diese vorher zu packen, was zudem Bandbreite spart.
- Das SMTP-Protokoll bietet keinen Schutz gegen evtl. Übertragungsfehler bei den Anhängen. Ist eine Datei hingegen gepackt, kann in der Regel über die Auswertung der Checksummen die Integrität einer Datei sichergestellt werden.

An dieser Stelle setzt das sog. "Anti-Virus-Patch" von Russell Nelson und Charles Cazabon für Qmail an (das es aber auch in vergleichbarer Form für z.B. Postfix gibt), indem E-Mail mit speziell deklariertem MIME-Inhalt blockiert bzw. abgewiesen werden.

7.3.3.1 Russell Nelson's Anti-Virus-Patch

Aus Anlass der W32/Sircom Wellen, die seit 2001 über das Internet schwappten, hat sich der Administrator der Qmail-Web-Seite[54] Russell Nelson hingesetzt und eine Erweiterung des **qmail-smtpd** Dienstes vorgenommen, den er zunächst "qmail-smtpd-virusscan" nannte und sich fundamental von den üblichen Virenscanner unterscheidet. Folgende Eckpunkte der Überlegung lassen sich ausmachen:

1. Alle eventuelle Viren/Würmer sind in den E-Mail-(MIME-)Anhängen zu finden und unter den Windows-Betriebssystemen ausführbare Dateien.
2. Die Windows-Executables können mittels eines modifizierten "magic" Tests

[54] http://qmail.org/top.html

erkannt werden, der gegen die Base64 kodierten Anhänge auszuführen ist.

3. Wird ein solcher Anhang während der SMTP "Data"-Phase erkannt, wird postum die gesamte E-Mail abgewiesen und der Absender erhält einen SMTP 55x Return-Code übermittelt. Die eigentliche E-Mail wird somit erst gar nicht in die Queue eingefügt und braucht daher auch nicht auf Viren gescannt zu werden.

Im wesentlichen geht es darum, eine genaue Buchführung der MIME-Anhänge vorzunehmen und die zweils ersten Base64 kodierten Bytes des Anhangs auszuwerten und gegenüber den hinterlegten Zeichenketten von bekannten Windows-Signaturen zu vergleichen.

In Kollaboation mit Charles Cazabon wurde in der Version 1.1. des Patches zunächst die Environment-Variable `EXECUTABLEOK` eingeführt. Durch Setzen dieser Variablen (z.B. in der **tcpserver** Konfirgurationsdatei `tcp.smtp.cdb`) lässt sich die Überprüfung. MTA-Client spezifisch festelegen.

Base64-Signaturen

Ausgangspunkt des Checks sind die ersten Bytes des Base64-kodierten Anhangs. Dies ist in der Regel hinreichend genau; es zeigte sich aber die Windows eine überraschende Vielfalt von "ausführbaren" Dateien aufweist, je nachdem mit welchen Compiler sie erzeugt wurden. Beispiele sind TVqQAAMAA und TVpQAAIAA In der Version 1.2 des Patches wurde die Datenbasis in eine separate Qmail Kontrolldatei `signatures` ausgelagert, sodass der Anwender diese beliebig erweitern kann, z.B. um Signaturen von Zip-Dateien (UEsDBBQAA).

Ein anderes prinzipielles Problem, ist die Erkennung von Anhängen in beispielsweise Bounce-Nachrichten. Wie weiter oben dargestellt, nutzen einige Virenhersteller bewusst diese Möglichkeit, um den Empfänger zunächst mit einer harmlosen — weil scheinbar technischen — E-Mail zu konfrontieren. In der Version 1.3 des Anti-Virus-Patches wurde die strenge MIME-Konformität des Anhangs gelockert und somit auch dieser potentielle "Einbruchsweg" geschlossen.

7.3.3.2 Die Verteidungslinie: SPAMCONTROL und QMVC

Wir wollen nun die konkrete E-Mail-/Spam-/Virensituation der Monate Januar und Februar 2004 betrachten, wo wir den Ausbruch des *MyDoom*- bzw. *NetSky*-Virus beobachten. Unser System wird gesichert durch

1. die SPAMCONTROL[55]-Ergänzung (Version 2.2) für **qmail-smtpd**, die seit der Version 2.1 Russell Nelson's Patch in einer verbesserten Version beinhaltet, und

[55] `http://www.fehcom.de/qmail/spamcontrol.html`

2. QMVC in der Version 1.5.9, unter Einsatz des AV-Scanner **uvscan** von McAfee sowie zusätzlichen Filtern auf ausführbare Dateien und solche mit potentiell gefährlichen Endungen (.pif, .scr) im Namen.

Im Groben stellt sich das E-Mail-Volumen auf unserem E-Mail-Gateway für diese Monate entsprechend Tabelle 7.3-1 dar:

Kriterium/Zeitraum	**01/2004**	**02/2004**
Eingelaufene E-Mails	1.281.613	1.415.224
Abgewiesene Spam- und Relay-Versuche	470.561	555.361
Abgewiesene E-Mails mit Windows-Executables	61.712	92.710
Gefilterte E-Mails aufgrund von MIME-Type und Dateiname	1.096	1.563
Erkannte infizierte E-Mails (**uvscan**)	6.689	4.706

Tabelle 7.3-1: E-Mail-, Spam- und Virenstatistik für Januar und Februar 2004.

Es ist zu berücksichtigen, dass alle abgewiesenen E-Mails erst gar nicht zum Filtern und Virusscannen durchgelassen werden; der Sender solcher E-Mails hat hierfür bereits eine 55x SMTP-Meldung erhalten.

Die Tagesstatistiken (Abbildung 7.3-3a und 7.3-4.a) zeigen die Anzahl der eingelaufenen E-Mails (Received E-Mails), die der gefilterten (Filtered Badmails) und die der erkannten, Virus infizierten (Filtered Virusmails). Der Ausbruch des *NetSky*-Virus erfolgte am 27.1.2004 wo zunächst die Dateifilter anschlagen. Im Verlauf der Epedemie "mutierte" der *NetSky* und versuchte durch die üblichen Filter zu schlüpfen, wurde aber von McAfee AV-Scanner (aufgrund dessen guter Identifikationsheuristik) dennoch erkannt. Abbildungen 7.3-3b und 7.3-4b zeigen die Häufigkeitsverteilung der identifizierten Viren für den entsprechenden Monat.

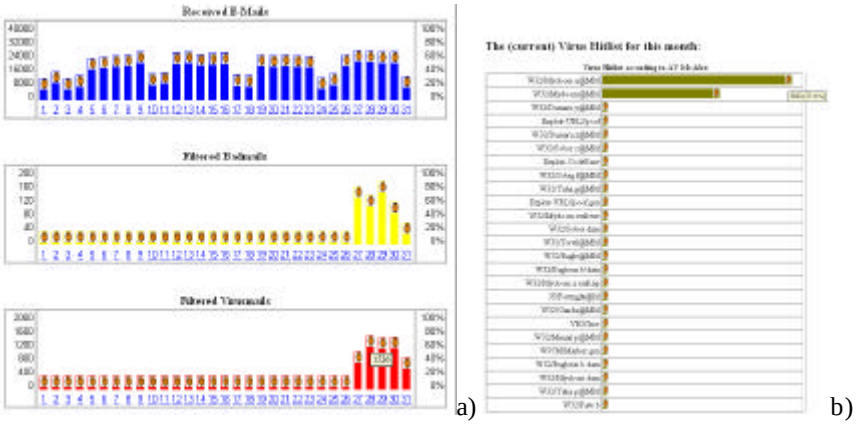

Abbildung 7.3-3: a) QMVC-Statistik der einlaufenden und gefilterten E-Mails, sowie b) die Häufigkeitsverteilungen der vom Virusscanner McAfee gefundenen Viren für den Monat Januar 2004.

Die Epidemie des MyDoom-Virus fand in den nächsten Tagen ihren Ausklang, wie in Abbildung 7.3-4a dargestellt ist.

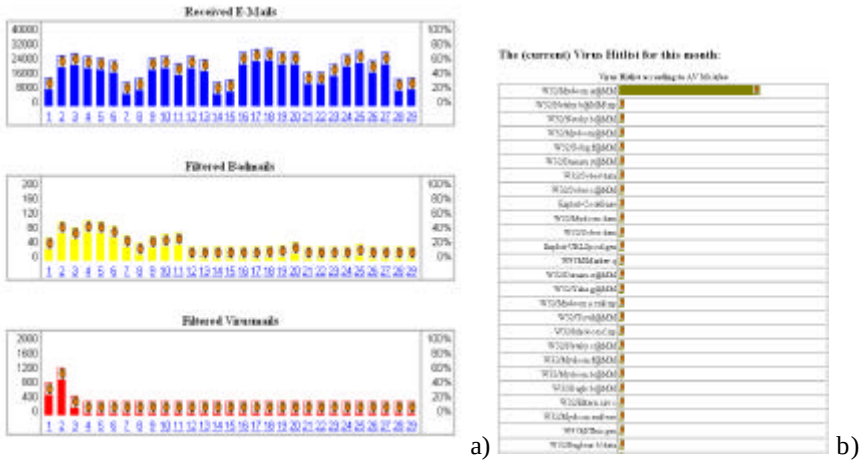

Abbildung 7.3-4: a) QMVC-Statistik der einlaufenden und gefilterten E-Mails, sowie b) die Häufigkeitsverteilungen der vom Virusscanner McAfee gefundenen Viren für den Monat Februar 2004.

7.3.3.3 Schlussfolgerungen

Die gezeigten Beispiele belegen, dass

- das Filtern auf Windows-Executables und ggf. Dateiname bzw. deren Endungen notwendig ist und unter den richtigen Voraussetzungen keine Einschränkung des E-Mail-Verkehrs darstellt,
- die frühzeitige Nutzung eines evtl. SMTP-Antwort-Codes 55x aufgrund von nicht-gewünschten MIME-Anhängen oder anderer Kriteren, eine erhebliche Entlastung (> 90%) der Virenscanner mit sich bringt, und somit
- die Arbeitsfähigkeit der E-Mail-Gateways auch "unter Stress" gewährleistet.

Sollte wirklich einmal ein ehrlicher Absender aufgrund der frühren Filterkriterien seine E-Mail nicht "los werden", so erhält er in jedem Fall — entweder aufgrund der direkten SMTP-Fehlermeldung, oder — mittels eines indirekten Bounces einen qualifizierten Hinweis auf das Übertragungsproblem.

Filterkriterien

1. Welche (erstrangigen) Filter sind nun für die SMTP-Datenübertragung effektiv und effizient ?
   - MIME-Type-Filter
   - Loader-Type-Filter
2. Welche (nachrangigen) Filter lassen sich noch während der SMTP-Datenübertragung effektiv einsetzen ?
   - Virenfilter
3. Welche (drittrangigen) Filter sollten ggf. dem Empfänger in Eigenverantwortung überlassen werden ?
   - Spamfilter
   - Dateinname-Filter
   - Inhaltliche Filter

Filtermöglichkeiten

Offen bleibt die Frage, welches Gefahrenpotential sich trotz optimal konfigurierter Filter ergeben kann. Nach heutigem Kenntnisstand sind folgende Probleme offen bzw. lassen sich nicht lösen:

- Exploits: Eine "an sich" harmlose E-Mail führt zu "Buffer-Overflows", die zu Exploits genutzt werden kann. Im Gegensatz zu "trojanischen Pferden" kann der Exploit aufgrund seines Code-Schemas nicht fest gemacht werden.
- Gepackte Dateien, z.B. im Zip-Format, die evtl. noch zusätzlich verschlüsselt werden. Hierdurch kann auch eine übliche Entpacker-Routine den Anhang nicht analysieren. Ohne aktives Zutun des Empfängers "ruht" aber das evtl. Virus und ist daher nicht weiter schädlich (solange Microsoft darauf verzichtet, einen automatischen "Unzipper" in seinen Betriebssystemen mitzuliefern).
- Makros in diversen Dateien, z.B. VBA-Makros eingebettet in den übertragenen Word- oder Exceldateien.

Es ist schwierig vorauszusagen, welche nächsten Schritte die Virenhersteller planen bzw. vornehmen. Im Grunde gibt es nur zwei Möglichkeiten dagegen zu halten:

- Ihnen die Aufgabe (technisch) so schwierig wie möglich zu machen.
- Eine tägliche Auswertung des E-Mail-Flusses vorzunehmen und auf Veränderungen zu achten. Werkzeuge wie QMVC oder newanalyse[56] können hierzu behilflich sein.

Es ist zu konstatieren, dass mittlerweile eine Infrastuktur infizierter PCs mit schlafenden Trojanern existiert[57], die bei Bedarf genutzt werden können. Ein neuer Worm braucht sich also nicht mehr die Mühe einer "Erstinfektion" zu machen, sondern kann sich gezielt auf dieses bekannte Potential stützen. Daher werden perspektivisch auch die typischen Antiviren-Programme und unwirksam: Bevor das erste Virus bei einem Antiviren-Hersteller antrifft, hat der Wurm bereits sein Ziel erreicht. Der einzige Ausweg bieten pro-aktive Filter auf den E-Mail-Gateways, die mit guter Effektivität und Effizienz bei Qmail vorliegen.

7.4 Domain- und User-Management

7.5 Lokale E-Mail-Anwender

7.6 Administration

7.7 Performance

7.8 Bearbeitung von E-Mail

7.9 Lisenmanager EZMLM

[56] http://ww.fehcom.de/qmail.html

[57] http://www.heise.de/newsticker/meldung/47134