`guardian` – Unerwünschte Werbe-E-Mails erkennen und bekämpfen

Hendrik Weimer

10. März 2002

Inhaltsverzeichnis

1 Einführung

Der Siegeszug des Internets hat leider auch seine Schattenseiten. Da E-Mails heutzutage den Absender fast nichts kosten, gibt es ständig Leute, die auf die Idee kommen, ihr Produkt mittels Tausenden oder Millionen von E-Mails anzupreisen. Oftmals geschieht dies ohne die Zustimmung des Empfängers, bei dem dadurch Kosten und Zeitaufwand entstehen. Solche Aktionen werden im Netz allgemein als „Spamming“ bezeichnet.

1.1 Was versteht man unter Spam?

Als „Spam“ wird eine E-Mail bezeichnet, die an eine große Zahl von Empfängern versandt wurde, ohne dass diese dem zugestimmt haben. Der Begriff geht auf einen Sketch aus „Monty Python's Flying Circus“ zurück, in dem in einem Restaurant in jedem Gericht auf der Speisekarte Spam[1] enthalten ist, oft sogar mehrfach. Genauso wie es in diesem Restaurant kein Entrinnen vor Spam gibt, ergeht es vielen E-Mail-Accounts!

Die Definition für Spam unterscheidet bewusst nicht zwischen kommerziellen und nichtkommerziellen Inhalten. Wer z.B. die 213. Aufforderung erhält, für einen guten Zweck zu spenden, wird davon ähnlich genervt sein, wie wenn er die 213. Werbung für eine angeblich unschlagbare Anlagemöglichkeit zugeschickt bekommt.

1.2 Warum ist das ein Problem?

Manchmal bekommt man den Einwand zu hören, dass Spam doch gar kein Problem sei, weil man ihn ja einfach aus seinem E-Mail-Ordner löschen könne. Das ist aber in mehrfacher hinsicht ein Trugschluss.

Erstens sind die Kosten für die Übertragung bereits angefallen. Diese trägt auch immer der Empfänger mit, selbst wenn er einen Pauschaltarif (Flatrate) an seinen Provider zahlt. Denn die Leitungen, die der Provider nutzt, werden in der Regel volumenbasiert abgerechnet, und die dabei entstehenden Kosten werden natürlich an den Endkunden weitergegeben. Der Gesamtschaden des weltweiten Spam-Aufkommens wurde in einer von der EU-Komission beauftragten Studie mit 10 Milliarden Euro beziffert. [1] Diese Zahl bezieht sich allerdings auf die Infrastruktur (und damit auch auf die Kosten), wie sie in der EU vorherrscht. Ein User in einem Internet-Entwicklungsland wie z.B. Sudan zahlt für eine empfangene E-Mail wesentlich mehr als ein Europäer, nämlich etwa das dreifache [2], bei gleicher Übertragungsgeschwindigkeit. Wenn man dann das durchschnittliche Pro-Kopf-Einkommen hinzuzieht, fällt der Vergleich noch drastischer aus. Es ist zu befürchten, das bei einem weiteren Ansteigen des Spam-Aufkommens das Medium E-Mail für die Bewohner dieser Länder unbezahlbar wird.

Zweitens wird die Verschwendung von Arbeitszeit nicht berücksichtigt. Selbst wenn man annimmt, dass man zum Abfragen seines E-Mail-Accounts nur wenige Sekunden benötigt, kommen bei mehreren Spammails pro Tag auf das Jahr verteilt betrachtet ähnlich astronomische Summen zustande.

[1] Spiced Pork and hAM – zusammengepresstes Schweinefleisch in Dosen

Drittens würde sich durch eine passive Haltung die Situation weiter verschlechtern. Spammer würden nicht mehr Gefahr laufen, ihren Internet-Zugang zu verlieren, sondern eher dazu ermutigt, den Empfängerkreis weiter zu vergrößern. Auch werden dann bald seriöse Firmen das Medium E-Mail für ihre Zwecke entdecken und durch entsprechende Lobbyarbeit dafür sorgen, dass eine spammerfreundliche Rechtslage geschaffen wird.

Es geht schließlich darum, das Medium E-Mail für alle Menschen weltweit benutzbar zu halten. Spam stellt eine ernste Bedrohung dar und muss daher entschlossen bekämpft werden. Das von mir entwickelte Projekt versucht, dafür einen kleinen Beitrag zu leisten.

1.3 Was kann man dagegen unternehmen?

Niemand muss es sich gefallen lassen, wenn er zugespammt wird. Falls der Absender aus der E-Mail ersichtlich ist, kann man gegen ihn eine strafbewehrte Unterlassungserklärung erwirken. Allerdings wird wegen einer unerwünschten E-Mail kaum jemand einen Anwalt einschalten, und dies funktioniert auch nur, wenn der Absender in Deutschland sitzt. Einfacher, aber auch nur in diesem Spezialfall möglich, ist es, den Spammer auf das Bundesdatenschutzgesetz (BDSG) aufmerksam zu machen. Da E-Mail-Adressen personenbezogene Daten sind, dürfen sie nicht ohne Zustimmung des Besitzers gespeichert und verarbeitet werden. Das Gesetz verlangt auch, dass auf Antrag die Quelle und der Verwendungszweck der gespeicherten Daten offengelegt werden muss, was einen Spammer natürlich Zeit kostet, sodass er es sich zukünftig zweimal überlegen wird, ob er spammt. Es gibt im Internet bereits vorgefertigte Texte, die sich in der Praxis sehr gut bewährt haben. [3]

Falls der Absender aber gar nicht deutschem Recht unterliegt, oder eine ungültige Absenderadresse angegeben hat, gibt es dennoch eine effektive Methode, ihm das Handwerk zu legen: man beschwert sich bei dem Provider des Spammers. Dieser wird ihn in der Regel zumindestens verwarnen und bei Wiederholung ihm den Zugang sperren. Dies macht ein Provider aus reinem Selbstschutz, denn wenn ein Provider nichts gegen Spammer unternimmt, wird er sich sehr schnell in sogenannten „Blacklists“ (s. 2.1.2) wiederfinden, wodurch der gesamte E-Mail-Verkehr eingeschränkt wird. Deswegen haben die meisten Provider per AGB, oder einem vergleichbaren Schriftstück, Spamming untersagt.

Es ist allerdings nicht immer trivial, den Provider des Absenders ausfindig zu machen. Sowohl die Absender-Adresse in der E-Mail (From-Zeile) als auch die gegenüber dem Mailserver angegebene Adresse (Envelope-From) können beliebig gefälscht sein. Der einzige Anhaltspunkt, der sich bietet, sind die Received-Zeilen im Header einer E-Mail. Jeder Mailserver, über den die E-Mail gelaufen ist, schreibt eine eigene Received-Zeile in den Header, und macht den Weg der E-Mail damit für den Empfänger nachvollziehbar. Der typische Weg einer E-Mail sieht so aus:

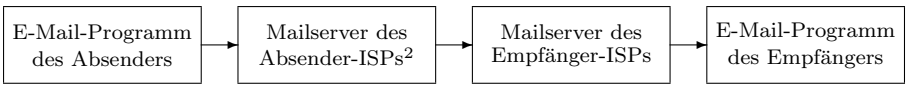


Abbildung 1: Weg einer E-Mail vom Sender zum Empfänger

Die dabei von den beiden Mailservern erzeugten Received-Zeilen könnten folgendermaßen aussehen:

```
Received: by mail.netic.de (Smail3.2.0.111/mail.s.netic.de)
    via LF.net GmbH Internet Services
    via remoteip 151.189.8.80
    via remotehost mx04.nexgo.de with esmtp for cygnus.s.netic.de
    id m15oKOn-001X3OC; Tue, 2 Oct 2001 09:55:29 +0200 (CEST)
Received: from internet (dsl-213-023-054-122.arcor-ip.net [213.23.54.122])
    by mx04.nexgo.de (Postfix) with SMTP id 220B737C52
    for <hendrik@enyo.de>; Tue, 2 Oct 2001 09:55:29 +0200 (CEST)
```

Received-Header sind immer von unten nach oben zu lesen. In diesem Fall wurde die E-Mail von dem Rechner mit der IP-Adresse `213.23.54.122` (einem Arcor-Einwahlknoten) vom Mailserver `mx04.nexgo.de` entgegengenommen (unterer Received-Header). Dieser nahm daraufhin Verbindung mit dem Rechner `mail.netic.de` auf und leitete an diesen die E-Mail weiter. Der zweite Mailserver fügte dann den oberen Received-Header ein und stellte die E-Mail dem Empfänger zu.

Wenn es sich in diesem Beispiel um Spam gehandelt hätte, wäre der Provider, auf den die IP-Adresse `213.23.54.122` registriert ist, der richtige Ansprechpartner. Eine Nameserver-Anfrage liefert `dsl-213-023-054-122.arcor-ip.net` als zugehörigen Hostname, die Angabe im Received-Header war also korrekt. Jetzt benötigt man nur noch einen Ansprechpartner auf Seiten des Providers. Dafür bietet das „Network Abuse Clearinghouse“ einen speziellen Dienst[3] an, dem man den Hostname eines Rechners übermittelt, worauf man die zuständige Beschwerdeadresse des Providers erhält. In unserem Fall liefert die Anfrage `abuse@arcor-ip.net` zurück, womit wir ein E-Mail-Adresse hätten, an die wir uns wenden könnten.
Die Beschwerdemail selbst muss drei Dinge enthalten:

- Den Grund der Beschwerde (also unerwünschtes Massenmailing)
- Den gesamten Header der Spammail
- Den kompletten Text der Spammail

Und schon hat man einem Spammer zumindest das Leben schwer gemacht, und vielleicht lässt er es in Zukunft ja ganz bleiben.

[2]Internet Service Provider
[3]whois.abuse.net

2 Automatisierung der Gegenmaßnahmen

Natürlich ist es ziemlich aufwändig, die zuständige Adresse stets von Hand zu recherchieren. Bei einer Spammail pro Woche mag das ja noch vertretbar sein, aber Spam im zweistelligen Bereich pro Tag ist für viele Accounts keine Seltenheit. Diese Arbeit kann einem der Computer aber relativ problemlos abnehmen. Allerdings wird damit das Problem der Kosten und verlorenen (Arbeits)zeit nicht gelöst. Daher ist es wichtig, dass Spam automatisch erkannt wird, und so in einem seperaten Mailordner gespeichert wird. Diesen Mailordner braucht man einerseits nie komplett herunterzuladen, und andererseits reicht es, alle paar Tage nachzuschauen, ob sich nicht vielleicht doch eine erwünschte E-Mail dahinverirrt hat.

Diese beiden Punkte werden von `guardian` abgedeckt. Das Projekt besteht dabei aus drei Komponenten:

- `guardian-check`: Dieses Filterprogramm erkennt mit sehr hoher Zuverlässigkeit, ob es sich bei einer E-Mail um Spam handelt.
- `guardian-complain`: Dieses Programm automatisiert den Versand der Beschwerde-E-Mails.
- `guardian-config`: Dieses Programm verwaltet die Einstellungen für die erstgenannten und ermöglicht eine flexible Anpassung des Spamfilters.

2.1 Automatische Erkennung

In diesem Abschnitt wird die Funktionsweise des Filterprogramms `guardian-check` beschrieben. Dieses Programm liest eine komplette E-Mail über die Standardeingabe ein und gibt sie mit ein paar zusätzlichen Headerzeilen versehen wieder auf der Standardausgabe aus. Somit ist es für den Einsatz in einer Pipeline prädestiniert. Am praktikabelsten ist es, das Programm direkt auf einem Mailserver laufen zu lassen, denn dadurch benötigt man keinen speziellen E-Mail-Client. Der Mailserver muss jedoch eine Möglichkeit bieten, in die lokale Zustellung der E-Mails eingreifen zu können, beispielsweise über eine `.forward`-Datei. Dies bieten zumindest alle gängigen UNIX-Mailserver wie `sendmail`, `qmail` oder `Exim`. Nach einer Untersuchung im September und Oktober 2001 laufen 65% aller Mailserver im Internet unter einem UNIX-Derivat. [4]

Das Programm selbst ist in Perl geschrieben und benötigt daher einen Perl-Interpreter. Dieser ist aber für alle gängigen Betriebsysteme verfügbar und oftmals bereits vorinstalliert. Desweiteren ist eine ständige Internetanbindung mit Zugriff auf einen Nameserver dringend zu empfehlen.

Um zu erkennen, ob es sich bei einer gerade eingetroffenen E-Mail um Spam handelt, habe ich ein spezielles Scoring-System entwickelt. Jede Mail erhält zu Beginn einen Score von 0. Sobald eins von zahlreichen Filterkriterien innerhalb der E-Mail erfüllt wurde, verändert sich dieser Score. Wenn der Score insgesamt -500 oder weniger beträgt, wird die E-Mail als Spam eingestuft. Die Filterkriterien sind als „Regular Expressions“ (auch

Regexps genannt) definiert. Diese werden vom (in Perl eingebauten) Regexp-Interpreter mit einer oder mehreren Zeilen aus der E-Mail verglichen. Fällt der Vergleich positiv aus, wird das Scoring um den der Filterregel zugehörigen Wert verändert.

Regexps unterstützen Wildcards, um beliebige Inhalte zu erfassen, „character classes“, um nur bestimmte Zeichen wie z.B. Satzzeichen, Ziffern oder Steuerzeichen zu finden. Es ist auch problemlos möglich, AND-, OR-, und NOT-Verknüpfungen einzubauen, um das Kriterium weiter zu verfeinern. Zur Veranschaulichung einige aus [5] entnommene Beispiele. Der Vergleich mit der Regexp

```
/Fred/
```

fällt positiv aus, wenn der zu untersuchende String an beliebiger Stelle die Zeichenfolge „Fred“ enthält. Nun ein etwas komplizierteres Beispiel:

```
/(Fred|Wilma|Pebbles) Flintstone/
```

Diese Regexp wird erfüllt, wenn der Vergleichsstring entweder „`Fred Flintstone`“, „`Wilma Flintstone`“ oder „`Pebbles Flintstone`“ enthält. Die Klammern dienen hierbei dazu, die Vornamen zu einer Gruppe zusammen zu fassen, der senkrechte Strich als ODER-Verknüpfung. Noch ein letztes Beispiel, diesmal ein aus dem Programm entnommenes:

```
/^From:.*[A-Za-z] [A-Za-z]/i
```

Der Zirkumflex zu Beginn dieser Regexp steht für den Anfang des zu untersuchenden Strings, und verlangt daher, dass dieser mit „`From:`“ beginnt. Danach kann aufgrund der `.*`-Wildcard jede beliebige Zeichenfolge auftreten. Irgendwo im restlichen Teil des Strings muss aber noch ein beliebiger Groß- oder Kleinbuchstabe, gefolgt von einem Leerzeichen, gefolgt von einem weiteren Groß- oder Kleinbuchstaben, auftauchen. In eckigen Klammern stehen die bereits erwähnten „character classes“ die in diesem Fall alle Groß- und Kleinbuchstaben enthalten. Das `i` am Ende der Regexp sorgt dafür, dass diese unabhängig von Groß- und Kleinschreibung zu interpretieren ist. Konkret prüft diese Regexp, ob in der From-Zeile einer E-Mail ein vollständiger Name enthalten ist.

2.1.1 Typische Erkennungsmerkmale

Glücklicherweise gibt es eine ganze Menge an Kriterien, an denen man Spam ziemlich sicher erkennen kann. Wenn in einer E-Mail der Satz „This is not spam.“ vorkommt, kann man mit an Sicherheit grenzender Wahrscheinlichkeit davon ausgehen, dass es sich doch um solchen handelt. Gleiches gilt für die immer wieder auftauchenden Verweise auf die „Bill 1618“, einem Gesetzesentwurf, der Spam in den USA legalisieren sollte, aber vom Kongress ins Gegenteil abgeändert wurde. [6] Aber zunächst empfiehlt es sich, auch den Header (Kopfzeilen) einer E-Mail genau anzusehen. Dort befinden sich bereits zahlreiche Hinweise, die auf Spam hindeuten. *Tabelle 1* enthält die Filter, die auf den Header der E-Mail angewendet werden.

Header	Filterkriterium	Erklärung	Score
From	nicht vorhanden	Ein nicht vorhandener From-Header findet sich normalerweise nur in Spam	-500
From	Absender von yahoo.com, aol.com, msn.com	Diese E-Mail-Provider werden häufig von Spammern genutzt, um Antworten abzufangen. Oft aber auch gefälscht.	-50
From	Absender enthält „sex“, „xxx“ oder „porn“	Enthält Werbung für eine Porno-Website	-500
From	kein Name	Spammer geben in der Regel keinen Namen an	-100
From	ungültige Domain	Mittels einer Nameserver-Anfrage wird überprüft, ob die vom Absender angegebene Domain überhaupt existiert	-500
From	Absender ist Abuse-Desk oder Mailer-Daemon	Vermutlich eine Antwort auf eine Beschwerde oder ein Bounce	+100000
To/Cc	nicht vorhanden	Das fehlen dieses Headers deutet auf Spam hin	-500
To/Cc	eigene Adresse fehlt	Da es für Spammer wesentlich teurer ist, jede Mail einzeln zuzustellen, wird die Zieladresse oft nur auf Mailserver-Ebene angegeben und fehlt damit im Header.	-500
Subject	enthält „http://“ oder „www. “	Werbung für eine Website	-250
Subject	unkodierte 8bit-Daten	Mehrere nicht korrekt kodierte 8bit-Zeichen deuten auf Spam aus Asien hin	-500
Subject	enthält $-Zeichen	In der Regel Werbung für dubiose Finanzgeschäfte	-100
Subject	beginnt mit „ADV: “	eindeutig Werbung	-500
References In-Reply-To	vorhanden	eine Antwort	+500
X-Mailer	enthält „bulk“, „mass“ oder „bomb“	Kennung eines Massen-E-Mail-Programms	-300
X-Mailer User-Agent	vorhanden	fehlt ansonsten in Spam	+100
Content-Type	text/html	reines HTML, oft Spam	-250
Received	Ungültige IP-Adresse	Gefälschter Received-Header	-10000
Received	nur ein Eintrag	Direkteinlieferung, wird oft von Spammern gemacht	-250
Received	IP-Adresse in Blacklist	Rechner, die in Blacklists (s. 2.1.2) stehen, sind Spammer oder spamfreundliche ISPs	-500

Tabelle 1: Filterkriterien im Header einer E-Mail

Die für den eigentlichen Text der E-Mail (Body) entwickelten Filter sind um einiges umfangreicher, da eine ganze Menge an Wörtern und Phrasen als spamtypisch definiert sind. Der Score wird dabei um 200 bis 500 Punkte nach unten korrigiert. Offensichtliches wie „porn“, „drugs“ oder „weight loss“ sind daher in *Tabelle 2* nicht aufgeführt.

Enthaltener String	**Erklärung**	**Score**
$$$	Drei Dollarzeichen sind ein sicheres Zeichen für „Make money fast“-Spam.	-500
multi level marketing	eine euphemistische Bezeichnung für Kettenbriefe	-500
Diverse „Remove“-Anleitungen	Spammer geben oft eine Anleitung an, wie man von ihrer Adressliste verschwinden kann. Eine beliebte Methode, um die bespammten Adressen zu verifizieren.	-500
>	Am Zeilenanfang ein Hinweis auf ein Zitat und damit auf eine Antwort	+10

Tabelle 2: Filterkriterien im Body einer E-Mail

Wenn die E-Mail komplett bearbeitet wurde, wird sie mit einigen zusätzlichen Headern versehen, die über den Score und damit über die Einstufung als Spam informieren. Diese Information kann dann z.B. vom E-Mail-Client ausgewertet werden, um die E-Mails entsprechend einzusortieren. Im Falle von Spam wird die E-Mail noch zusätzlich an einer vorgegebenen Stelle gespeichert, um sie für die automatische Beschwerde-E-Mail zur Verfügung zu stellen.

2.1.2 Realtime Blackhole Lists

Wie bereits erwähnt, nutzt `guardian-check` „Realtime Blackhole Lists“ (RBLs), um bekannte Spammer und Spammer-Provider zu erkennen. Diese Listen werden von unabhängigen Institutionen betrieben, ihre Benutzung ist in der Regel kostenfrei. Die Bedienung ist relativ einfach: man schickt eine Nameserver-Anfrage mit der abzufragenden IP-Adresse (die man aus dem Received-Header extrahiert hat) und der zu verwendenden Datenbank ab. Wenn auf die Anfrage ein Ergebnis zurückkommt, steht der in Frage stehende Rechner in der Datenbank und ist somit eine potentielle Spamquelle. Ein Beispiel für eine Anfrage an eine RBL (127.0.0.2 ist in fast allen RBLs für Testzwecke gelistet), die IP-Adresse muss hierbei umgekehrt in „reverse notation“ angegeben werden:

```
gienah:~/jufo/2002$ nslookup 2.0.0.127.inputs.orbz.org
Server: cygnus.enyo.de
Address: 192.168.1.1

Name: 2.0.0.127.inputs.orbz.org
Address: 127.0.0.2
```

Ein anderer Typ von RBLs listet nicht bekannte Spamquellen, sondern offene Mailrelays. Diese Rechner erlauben es jedem Internetnutzer auf der ganzen Welt, über ihr System E-Mails zu verschicken. Was auf den ersten Blick gar nicht schlecht klingt und früher auch als freundlicher Service galt, ist heute zu einer Anziehungsquelle für Spammer geworden. Diese nutzen offene Relays, um den Weg ihrer E-Mail schwerer nachvollziehbar zu machen und um Sanktionen ihres Providers zu entgehen. Offene Relays sind fast ausschließlich Rechner mit veralteter Mailserver-Software, die Standardkonfiguaration aktueller Software sorgt dafür, dass nicht jeder Internet-User über das System E-Mails verschicken kann.

Wenn eine E-Mail über ein offenes Relay versandt wurde, nimmt sie in der Regel folgenden Weg:

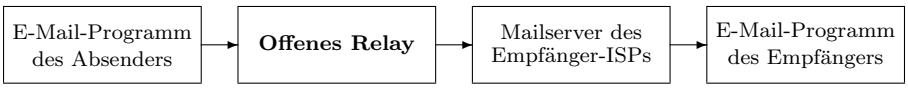


Abbildung 2: Weg einer E-Mail über ein offenes Relay

Es ist vergleichsweise einfach herauszufinden, ob ein Rechner ein offenes Relay ist: man versucht einfach, über diesen Rechner eine E-Mail an sich selbst zu schicken. Solch ein Relaytest wird von den Betreibern einer RBL gegen offene Relays stets durchgeführt, bevor der Server in die Blacklist aufgenommen wird.

Beide Arten von RBLs werden von `guardian-check` abgefragt, und die Ergebnisse fließen in das Scoring ein.

2.2 Automatische E-Mail an den Provider des Spammers

Dieser Abschnitt behandelt den zweiten Teil des `guardian`-Pakets, die automatische Beschwerdegenerierung mit `guardian-complain`. Das Programm ist ebenfalls in Perl geschrieben und lässt sich als CGI[4]-Script in nahezu jeden Webserver integrieren. Ein CGI-Script erzeugt die vom Benutzer angefordete Seite dynamisch, es kann also auf vorher übertragene Daten (z.B. aus einem Formular) reagieren. Damit ist es für den Benutzer problemlos möglich, mit jedem beliebigen Browser das Programm zu bedienen.

Zunächst muss sich der Benutzer gegenüber dem Webserver mit Usernamen und Passwort authentifizieren. Anhand der Kennung werden dem Benutzer die Nachrichten präsentiert, die von `guardian-check` als Spam eingestuft wurden. Dies erfordert natürlich, dass der Webserver zusammen mit dem Mailserver auf demselben Rechner läuft, oder dass ein Datenzugriff z.B. über NFS[5] möglich ist. Der Benutzer kann auch eine Textdatei mit einer Spam-E-Mail an den Webserver senden, falls diese E-Mail nicht korrekt herausgefiltert wurde. Der weitere Programmablauf lässt sich am einfachsten anhand des Flussdiagramms auf der folgenden Seite erklären.

[4]Common Gateway Interface
[5]Network File System

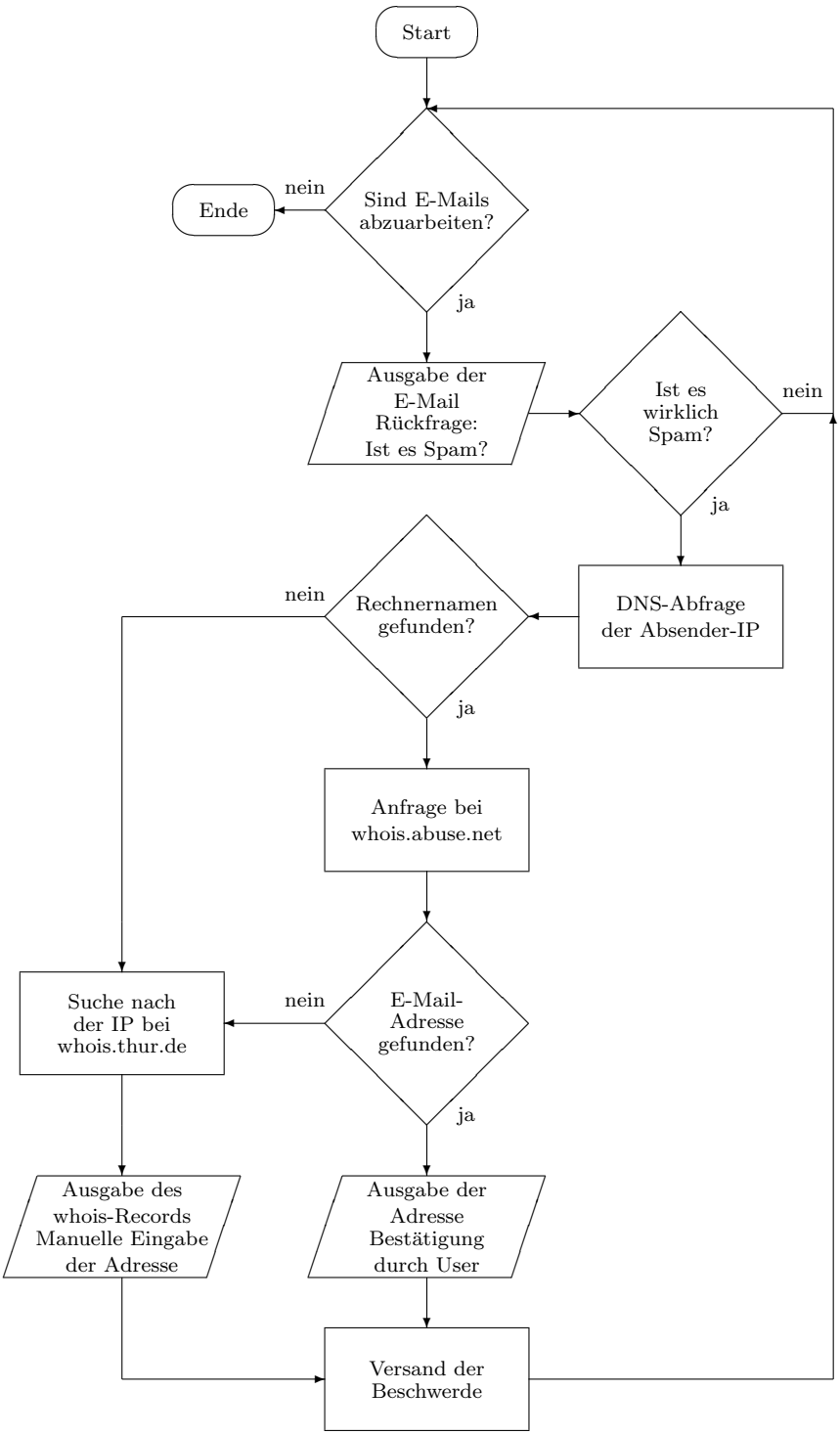


Abbildung 3: Programmablaufplan von `guardian-complain`

Jede E-Mail, die als Spam eingestuft wurde, wird einzeln abgearbeitet. Zunächst muss der Benutzer bestätigen, dass es sich bei dieser E-Mail wirklich um Spam handelt. Nichts ist ärgerlicher als der Beschwerde-Abteilung eines Providers Beschwerden über legitime E-Mails zu schicken, da dadurch die Bearbeitung von echten Missbrauchsfällen verzögert wird. Wird dieses bejaht, ermittelt das Programm die IP-Adresse des Absenders aus den Received-Zeilen der E-Mail und versucht durch eine Nameserver-Anfrage den Hostname zu bestimmen. Gelingt dies, greift das Programm auf den bereits erwähnten Dienst `whois.abuse.net` zu und sucht passende E-Mail-Adresse für Beschwerden. Diese wird dem Benutzer zur Bestätigung vorgelegt und die Beschwerde-E-Mail kann versandt werden.

Lässt sich kein Hostname für die IP-Adresse bestimmen oder findet `whois.abuse.net` keine zugehörige E-Mail-Adresse für Beschwerden, kann man mittels einer Anfrage bei `whois.thur.de` herausfinden, auf wen die IP-Adresse registriert ist. Dieser Dienst unterhält keine eigene Datenbank, sondern leitet die Anfrage automatisch an die für die Vergabe von IP-Adressen zuständige Institution (z.B. DENIC) weiter. Deren Antwort (whois-Record) wird dann dem Anfragenden geliefert. Allerdings ist diese Methode nur für versierte Benutzer zu empfehlen, da ein whois-Record mehrere E-Mail-Adressen enthält, von denen die meisten mit der Bearbeitung von Beschwerden wegen Spam nichts zu tun haben. Deswegen ist auch hier Vorsicht angesagt. Anhand des whois-Records kann dann der Benutzer die E-Mail-Adresse für die Beschwerde selbst eingeben, woraufhin die entsprechende E-Mail verschickt wird. Danach geht es mit der nächsten mutmaßlichen Spam-E-Mail weiter, bis alle angefallenen E-Mails abgearbeitet wurden.

2.3 Individuelle Anpassung

Das dritte Programm, `guardian-conf`, ist ebenfalls ein CGI-Script, das in einen Webserver eingebunden werden kann. Der Benutzer kann bequem mittels Browser auf die Einstellungen für `guardian-check` und `guardian-complain` zugreifen. Nach dem Login mit Benutzernamen und Passwort bekommt der Benutzer das nebenstehende Formular präsentiert.
Im obersten Feld gibt der Benutzer alle E-Mail-Adressen an, die von ihm verwendet werden. Für ganze Domains ist es möglich, einen Stern als Wildcard zu verwenden. Anhand der Domains der einzelnen E-Mail-Adressen ermittelt `guardian-conf` automatisch die IP-Adressen der für diese Domains zuständigen Mailserver, damit diese als provider-intern erkannt werden.

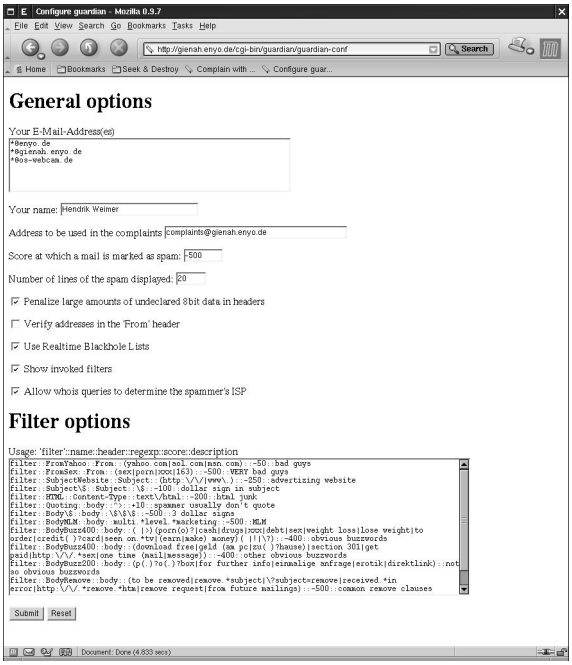


Abbildung 4: Konfiguration durch `guardian-conf`

In den beiden Feldern darunter kann man seinen Namen und eine E-Mail-Adresse für die Beschwerden angeben. Aus diesen Angaben erzeugt `guardian-complain` den From-Header der Beschwerde-E-Mail. Es ist ganz praktisch, für die Beschwerden eine eigene E-Mail-Adresse zu verwenden, da man so die Antworten der Provider in einen separaten Ordner sortieren kann.

Im vierten Feld kann man den Wert eintragen, ab dem eine E-Mail als Spam eingestuft wird. Das nächste Feld enthält die Information, wie viele Zeilen des Bodys der zu untersuchenden E-Mail `guardian-complain` ausgibt, damit der Benutzer entscheiden kann, ob es sich um Spam handelt oder nicht. Hier sollte kein zu hoher Wert gewählt werden, da man sich sonst den Spam nochmal herunterlädt. 20 Zeilen sind völlig ausreichend, um diese Entscheidung zu treffen.

Darunter befinden sich einige Checkboxen, mit denen sich Funktionen, die unter Umständen Nebenwirkungen zeigen können, abschalten lassen. So sind mehrere Nicht-ASCII-Zeichen (also > 127) im Subject ein deutliches Anzeichen für Spam, da diese nicht dem gültigen Standard für E-Mails entsprechen. [7] Allerdings dürfte es E-Mail-Clients geben, die sich in dieser Frage nicht standardkonform verhalten. Für Benutzer, die E-Mails mit Zeichensätzen, die weit über US-ASCII hinausgehen (für ISO-8859-1, der vor allem in Europa genutzt wird, ist das nicht der Fall) auf jeden Fall empfangen wollen, ist diese Option daher deaktivierbar.

Die Überprüfung der Absender-Adressen sowie die Abfrage von RBLs lassen sich abschalten, falls der Computer nicht über eine Internet-Anbindung mit Zugang zu einem Nameserver verfügt. Dies kann z.B. in einem durch eine Firewall geschützten Netz der Fall sein.

„Show invoked filters“ sorgt dafür, dass nach Überprüfung der E-Mail eine zusätzliche Headerzeile eingefügt wird, aus der ersichtlich wird, welche Filterkriterien erfüllt worden sind.

An dieser Stelle lassen sich auch die whois-Abfragen zu `whois.thur.de` an- und abschalten, je nachdem, ob der Benutzer mit der dort verfügbaren Information etwas anfangen kann.

Das unterste Textfeld dient dazu, eigene Filterregeln zu definieren, um `guardian-check` flexibel an die eigenen Wünsche anzupassen. Man muss nur den Headernamen (oder „Body“), die Regexp und die Auswirkung auf den Score angeben, und schon ist der Filter fertig gestellt. Schließlich ist kein Programm hundertprozentig perfekt, auch wenn viele Features von `guardian` dafür sorgen, dass dieser Idealzustand ein wenig näher rückt.

An dieser Stelle kann man auch eine Statistik aufrufen, die einem Auskunft erteilt, welche Filterregeln am häufigsten erfüllt wurden. Anhand der falsch erkannten Fälle erhält man eine erste Einschätzung, welche Filterkriterien möglicherweise mit einem zu hohen oder zu niedrigem Score belegt sind.

3 Ergebnisse

Der letzte Teil dieser Dokumentation widmet sich den Ergebnissen, die `guardian` im laufenden Betrieb erreicht. Einerseits ist natürlich die Erkennungsquote von Spam sehr wichtig, denn nur durch eine hohe Erkennungsrate bei einer sehr geringen Anzahl an „false positives“ kann das Programm erfolgreich sein. Andererseits muss man auch die Frage stellen, wie sich das Programm verhält, wenn es auf einem Mailserver eingesetzt wird, der mit mehreren tausend E-Mails pro Tag konfrontiert wird. Dabei darf `guardian-check` natürlich nicht zu einem Klotz am Bein werden, der das gesamte System merklich ausbremst.

Beide Punkte werden von `guardian` ohne Schwierigkeiten erfüllt. Die folgenden Zahlen verdeutlichen, dass das Programm sehr gut geeignet ist, um gegen die Geissel Spam anzugehen.

3.1 Erkennungsrate

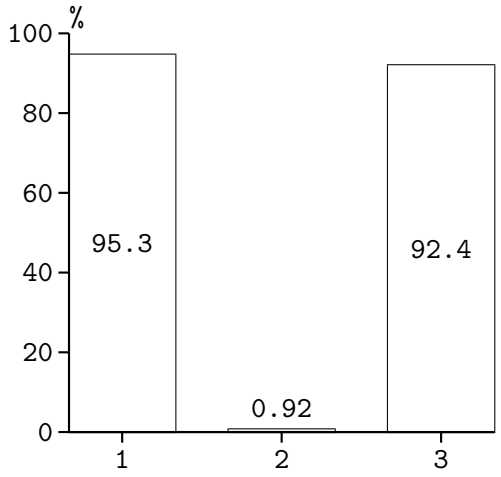


Abbildung 5: Gemeldeter Spam-Anteil

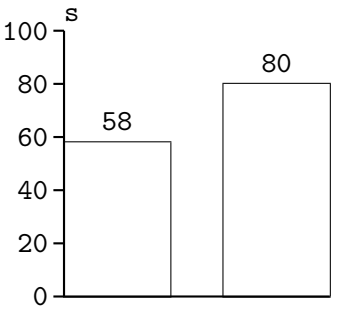
Der erste Test zeigt, wie gut das Programm im Erkennen von Spam ist. Dabei wurde in drei bereits vorsortierten E-Mail-Archiven jede E-Mail von `guardian-check` überprüft.
Archiv 1 ist mein eigenes Spam-Archiv, das ich zur Entwicklung der Filterregeln benutzt habe. In diesem Archiv wurden von 600 E-Mails 572 korrekt als Spam eingestuft. Bei Archiv 2 handelt es sich um meinen persönlichen E-Mail-Ordner von 2001, hier wurden nur 11 Nachrichten von 1201 irrtümlicherweise als Spam klassifiziert. Von diesen 11 Nachrichten waren allein 4 der Newsletter von dem Online-Auktionshaus ebay – da ebay-Mitglieder diesen nicht abbestellen können, stellt sich die Frage, ob es sich dabei nicht ohnehin um Spam handelt.

Da es sich bei Archiv 1 um das Entwicklungsarchiv handelt, müssen die Ergebnisse nicht zwangsläufig repräsentativ sein. Es wäre denkbar, dass die aufgestellten Filter nur für dieses Archiv und zu diesem Zeitpunkt eine so hohe Quote erreichen. Um diese Befürchtung auszuräumen, habe ich `guardian-check` ein zweites Spam-Archiv (Archiv 3), das mir freundlicherweise zur Verfügung gestellt wurde, überprüfen lassen, ohne das Programm und seine Filterregeln zu ändern. Das Programm hat von 1520 E-Mails 1401 richtig als Spam erkannt. Die sehr hohe Quote aus Archiv 1 ist folglich auch repräsentativ für andere Empfänger. Diese Ziffern belegen eindrucksvoll, dass `guardian-check` sehr effizient Spam erkennen kann.

3.2 Performance

Nun zur Frage, ob `guardian-check` den verwendeten Mailserver ausbremst. Im normalen Betrieb liessen sich auf allen getesteten Systemen keine Verzögerungen feststellen. Dies

liegt daran, dass das Programm nur wenig Prozessorleistung und Hauptspeicher benötigt und so gut wie keine Festplattenzugriffe durchführt. Erst bei einer starken Parallelisierung der E-Mail-Zustellung merkt man geringe Einbußen. Aber selbst unter extremen Bedingungen halten sich die Verzögerungen noch im Rahmen, wie die untenstehende Grafik zeigt. Bei diesem Test wurden in extrem kurzer Zeit über 10 Verbindungen gleichzeitig jeweils 100 E-Mails abgesetzt. Dies hat mit normalem E-Mail-Empfang nicht mehr viel zu tun, vielmehr stellt es bereits eine „Denial Of Service“-Attacke dar. Gemessen wurde die Dauer der gesamten Zustellung, also vom Beginn des Verbindungsaufbaus der ersten E-Mail bis zur endgültigen Zustellung der 1000. Das hierbei verwendete System war ein AMD K6-II mit 450 MHz und 128 MB RAM. Als Betriebssystem kam Linux 2.4.7, als Mailserver Exim 3.22 zum Einsatz.



Links der Wert für den Versand ohne `guardian-check`, rechts die Dauer bei eingeschaltetem Spamfilter. Man sieht, dass auch unter schwierigen Bedingungen sich die E-Mail-Zustellung nicht unermesslich verzögert. Auch größere Mailserver-Installationen können das Programm bedenkenlos einsetzen.

Abbildung 6: Zustelldauer für 1000 E-Mails

Literatur

[1] Serge Gauthonet, Étienne Drouard: Unerbetene kommerzielle Kommunikation und Datenschutz, Zusammenfassung der Schlussfolgerungen der Studie, Januar 2001, S.11

[2] Sudanet Dial-Up Access, `http://sudanet.net/dial_up.htm`

[3] Thomas Goerlich: Thoms Fassung von Framstags freundlichen Folterfragebogen (T5F), `http://www.schnappmatik.de/TFFFFF/`

[4] Dave J. Bernstein: Internet host SMTP server survey, 4. Oktober 2001, `http://cr.yp.to/surveys/smtpsoftware6.txt`

[5] Larry Wall, Tom Christensen, Randal L. Schwartz: Programming Perl, O'Reilly, S. 58f.

[6] Sebastian Koppehel: This is not Spam! oder: Die Wahrheit über Senate Bill S. 1618, `http://www.bastisoft.de/misc/s.1618.html`

[7] RFC 2822: Internet Message Format, 2.2 Header Fields: „A field body may be composed of any US-ASCII characters, except for CR and LF.“, `http://www.ietf.org/rfc/rfc2822.txt`