KAPITEL 12

Procmail

Procmail ist eine E-Mail-Filterlösung für Unix-Systeme. Es ermöglicht, über eine eigene Filtersprache E-Mails zu filtern, zu sortieren und zu modifizieren. Auf vielen Unix-Systemen wird Procmail als Mail Delivery Agent (MDA) eingesetzt, der die eintreffenden E-Mails entgegennimmt und an die Benutzer ausliefert. Seine Filtersprache ist so flexibel, dass es sogar möglich ist, einen Mailinglisten-Server vollständig mittels Procmail zu schreiben.

In diesem Kapitel sollen allerdings die Möglichkeiten, Procmail zur Filterung von Spam und Viren einzusetzen, beleuchtet werden. Es wird eine Reihe von Codeteilen vorgestellt, die jeweils die Aufgabe haben, ein bestimmtes Problem zu lösen. Die einzelnen Fragmente sind also eher als unabhängige Rezepte statt als Gesamtlösung anzusehen. Am Ende des Kapitels werden noch einige allgemeinere in Procmail realisierte Lösungen vorgestellt.

Ein Hinweis vorweg: Dieses Kapitel lebt von seinen regulären Ausdrücken. Ohne zumindest rudimentäres Wissen über reguläre Ausdrücke sind die meisten Rezepte nicht zu verstehen; aus diesem Grund befindet sich in Anhang B, *Reguläre Ausdrücke* eine Einführung in reguläre Ausdrücke. Um das Thema zu vertiefen, wird das exzellente Buch »Reguläre Ausdrücke« von Jeffrey E. F. Friedl empfohlen (ISBN 3-8972-1349-4, erschienen im O'Reilly Verlag).

Einführung in Procmail

Dieser Abschnitt gibt eine Einführung in den Aufbau von Procmail-Filtern, um das Verständnis für die folgenden Abschnitte zu erhöhen. Fortgeschrittene Themen – wie die Gewichtung von Filtern – werden im Rahmen dieses Buchs nicht behandelt. Weitere Informationen zu Procmail findet man in den Manpages *procmail*, *procmailex* und *procmailrc*.

Procmail aufrufen

Normalerweise sollte Procmail so eingerichtet werden, dass es vom MTA aufgerufen wird. Im Folgenden wird beschrieben, wie dies eingerichtet werden kann. Oft sind die entsprechenden Einstellungen schon vorkonfiguriert oder müssen nur noch auskommentiert werden. Wenn die Einbindung in den MTA nicht möglich oder erwünscht ist, können einzelne Nutzer es auch aus ihrer *.forward*-Datei heraus aufrufen, was unten beschrieben wird.

Einbindung in Postfix

Um Procmail als MDA in Postfix zu verwenden, fügt man einfach die folgende Zeile in die Postfix-Konfigurationsdatei */etc/postfix/main.cf* ein:

```
mailbox_command = procmail -a "$EXTENSION"
```

Einbindung in Exim

Um Procmail in Exim einzubinden, geht man wie folgt vor. Man beachte, dass die entsprechenden Konfigurationen schon häufig voreingestellt sind oder zum einfachen Auskommentieren angeboten werden, oft mit weiteren Einstellungen, die für das System angebracht sind.

Zuerst definiert man folgenden Transport im Abschnitt `transports` der Konfigurationsdatei (Reihenfolge ist unerheblich):

```
procmail_pipe:
  driver = pipe
  command = /usr/bin/procmail
  return_path_add
  delivery_date_add
  envelope_to_add
```

Anschließend fügt man diesen Router im Abschnitt `routers` der Konfigurationsdatei ein:

```
procmail:
  driver = accept
  domains = +local_domains
  check_local_user
  transport = procmail_pipe
  no_verify
  no_expn
```

Die Reihenfolge der Router ist wichtig. Der `procmail`-Router muss auf jeden Fall vor dem Router `local_user` stehen und – falls vorhanden – auch vor `maildrop` und `mail4root`. Ansonsten sollte er hinter allen anderen Router-Einträgen stehen.

Einbindung in Sendmail

Selbstverständlich unterstützt auch Sendmail die Verwendung von Procmail. Dazu fügt man die folgende Zeile in die Sendmail-Konfigurationsdatei *sendmail.mc* ein:

```
FEATURE(`local_procmail', `/usr/bin/procmail')dnl
```

Diese Zeile muss vor der Zeile `MAILER(`local')` stehen. Nach der Änderung muss wie üblich die Konfigurationsdatei *sendmail.cf* mit `m4` erzeugt werden.

Einbindung über .forward-Dateien

Wenn Procmail nicht aus dem MTA heraus aufgerufen wird, kann man es auch aus der persönlichen *.forward*-Datei jedes Nutzers heraus aufrufen. Dazu trägt man folgende Zeile in die Datei *.forward* im jeweiligen Home-Verzeichnis ein:

```
"|IFS=' ' && exec /usr/bin/procmail -f- || exit 75 #benutzername"
```

Diese Zeile besteht aus mehreren Bestandteilen:

- Die Pipe am Anfang sagt dem MTA, der diese Datei liest, dass die E-Mail an einen Prozess weitergeleitet werden soll, anstatt direkt in ein Postfach ausgeliefert zu werden.
- `IFS=' '` setzt das interne Feldtrennzeichen der Shell auf ein Leerzeichen. Damit umgeht man einen alten Sendmail-Fehler, der das Trennzeichen auf andere Werte setzt.
- Das `&&` bewirkt, dass der folgende Befehl nur dann ausgeführt wird, wenn der vorangegangene Befehl erfolgreich ausgeführt wurde.
- Das `exec` ersetzt das aktuelle Skript durch den Procmail-Prozess, wenn er erfolgreich geladen werden konnte.
- Mit `-f-` wird Procmail angewiesen, dass fehlende From-Zeilen neu erzeugt werden sollen und die Zeitstempel von bereits existierenden aktualisiert werden sollen.
- Das `||` besagt, dass der folgende Befehl nur dann ausgeführt werden soll, wenn der vorherige Befehl fehlgeschlagen ist.
- `exit 75` gibt, wenn beim Aufruf von Procmail ein Fehler passiert, den Fehlercode 75 an das auslieferndе Programm zurück. Dieser Code entspricht dem Makro `EX_TEMPFAIL`, was zur Folge hat, dass durch dieses die E-Mail zurück in die Queue gestellt und die Auslieferung später noch einmal versucht wird.
- Der Kommentar `#benutzername` muss gesetzt werden, da einige Sendmail-Versionen versuchen, anhand des Dateinamens von Filtern zu erkennen, ob diese mehrfach vorkommen, und dieses Mehrfachvorhandensein dann wegoptimieren. Deshalb muss jede *.forward*-Datei ein einzigartiges Element haben.

Es gibt auch einfachere Varianten dieses Befehls. Man kann zum Beispiel auch einfach

```
|/usr/bin/procmail
```

schreiben, und es wird in der Regel funktionieren. Der oben gezeigte Eintrag ist aber die robusteste bekannte Variante.

Konfigurationsoptionen

Procmail sucht seine Einstellungen in der Datei *.procmailrc* im Home-Verzeichnis des jeweiligen Benutzers sowie globale Einstellungen in */etc/procmailrc*. Diese Dateien werden von Procmail bei jedem Aufruf gelesen und interpretiert.

Die Konfigurationsdatei enthält hauptsächlich Filterregeln, die im Anschluss behandelt werden, und kann außerdem einige Variablenzuweisungen enthalten, die Procmail konfigurieren. Diese sehen aus wie Variablenzuweisungen in der Shell, zum Beispiel:

```
VERBOSE=yes
```

Die Einstellungen haben im Normalfall sinnvolle Standardwerte, allerdings ist es zu Testzwecken unter Umständen nützlich, diese zu modifizieren. Nachfolgend werden die wichtigsten Einstellungen erläutert. Weitere Einstellungen findet man in der Manpage *procmailrc*.

`LOGFILE`
: Wenn diese Variable gesetzt ist, werden alle Log-Meldungen in die angegebene Datei geschrieben. Wenn keine Log-Datei gesetzt ist, werden Fehlermeldungen an den Absender einer Mail geschickt. Denkbar wäre zum Beispiel folgende Einstellung:

  ```
  LOGFILE=$HOME/.procmail.log
  ```

`VERBOSE`
: Wenn diese Variable auf `yes` oder `on` gesetzt ist, schreibt Procmail zusätzliche Informationen in die Log-Datei.

`INCLUDERC`
: Wenn diese Variable auf einen Dateinamen gesetzt ist, wird diese Datei eingelesen und von Procmail interpretiert. Damit kann man längere Regelsätze in separate Dateien auslagern, was die Übersicht bei großen Mengen von Regeln deutlich verbessern kann.

`MAILDIR`
: Setzt das Heimatverzeichnis von Procmail. Sämtliche Pfade und Dateien liegen relativ zu diesem Verzeichnis. Normalerweise ist es identisch mit dem von der Shell gesetzten `$HOME`.

Filterregeln

Filterregeln bestehen bei Procmail aus drei Komponenten: dem Kopf, einem Bedingungsblock und einer daraus resultierenden Aktion.

```
:0 [flags] [ : [lockdatei] ]
<Null oder mehr Bedingungen (eine pro Zeile)>
<eine Aktion>
```

Der Kopf

Der Kopf einer Procmail-Regel beginnt aus historischen Gründen immer mit `:0`. Danach können ein oder mehrere Flags gesetzt werden. Über diese Flags wird angegeben, auf welche Teile einer E-Mail sich die Bedingungen beziehen sollen.

`H`
: Die Bedingungen prüfen den Kopf (Header) der E-Mail. Dies ist die Standardeinstellung.

`B`
: Die Bedingungen prüfen den Körper (Body) der E-Mail.

`D`
: Alle Bedingungen beachten Groß-/Kleinschreibung. Im Normalfall wird die Groß- und Kleinschreibung ignoriert.

`A`
: Dieser Filter wird nur dann ausgeführt, wenn die Bedingungen des vorangegangenen Filters zutreffen.

`a`
: Dieser Filter wird nur dann ausgeführt, wenn die Aktionszeile des vorherigen Filters erfolgreich ausgeführt wurde.

`e`
: Dieser Filter wird nur dann ausgeführt, wenn die Aktionszeile des vorangegangenen Filters einen Fehler zurückgegeben hat.

`h`
: Wenn die Aktion eine Pipe ist, dann soll der Kopf der E-Mail an die Pipe weitergegeben werden. Wenn weder `h` noch `b` angegeben sind, wird die gesamte E-Mail weitergegeben (Header und Body).

`b`
: Wenn die Aktion eine Pipe ist, dann soll der Körper an die Pipe weitergegeben werden. Wenn weder `h` noch `b` angegeben sind, wird die gesamte E-Mail weitergegeben (Header und Body).

`c`
: Dieser Filter erstellt eine Kopie der E-Mail. Das ist zum Beispiel sehr nützlich, wenn man Kopien jeder E-Mail aufbewahren möchte.

`w`

Normalerweise ignoriert Procmail den Rückgabewert eines Programms, dem eine E-Mail übergeben wurde. Wenn das `w`-Flag gesetzt ist, wartet Procmail, bis das Programm, an das die E-Mail übergeben wurde, sich beendet hat, und analysiert dessen Rückgabewert. Im Fall eines Rückgabewerts größer null gilt der Filter als fehlgeschlagen.

`W`

Dieses Flag arbeitet genauso wie `w`, nur unterdrückt sie Fehlermeldungen des zurückgebenden Programms.

`i`

Wenn diese Variable gesetzt ist, ignoriert Procmail Fehler, die beim Schreiben in eine Pipe auftreten.

`r`

Normalerweise nimmt Procmail Optimierungen von E-Mails vor. Ein Beispiel dafür ist das Löschen von unnötigen Leerzeilen. Ist dieses Verhalten unerwünscht, muss das r-Flag gesetzt sein.

Wenn den Flags ein weiterer Doppelpunkt folgt, verwendet Procmail für diesen Filter eine lokale Lock-Datei. Das wird dann wichtig, wenn die Gefahr besteht, dass mehrere Programme gleichzeitig in eine Datei schreiben. Procmail wartet dann so lange, bis der Schreibvorgang des anderes Prozesses beendet ist. Lock-Dateien sind bei der Verwendung von Postfächern im Mbox-Format (alle E-Mails in einer Datei) wichtig, aber bei Postfächern im Maildir-Format (eine Datei pro E-Mail) unnötig. Welches Postfachformat verwendet wird, hängt vom MTA ab. Folgt auf den `:` noch ein Dateiname, wird dieser als Dateiname für die Lock-Datei verwendet.

Die Bedingung

Eine Bedingung beginnt mit einem * , darauf folgt ein regulärer Ausdruck. Leerzeichen direkt nach dem * werden ignoriert.

Zur Vereinfachung unterstützt Procmail vier Makros, die durch kompliziertere reguläre Ausdrücke ersetzt werden: `^TO` sucht nach einem Wort in der Empfängeradresse, zum Beispiel `^TOinfo` (einsetzbar an Stelle von `^(To|Cc):.*info` oder Ähnlichem). `^TO_` sucht nach einer E-Mail-Adresse in der Empfängerliste, zum Beispiel `^TO_info@example.net`. `^FROM_MAILER` passt auf E-Mails vom E-Mail-System selbst, also bei Fehlern und Ähnlichem. Und `^FROM_DAEMON` passt auf E-Mails von Daemon-Prozessen, zum Beispiel automatische E-Mails von Mailinglisten-Systemen und auch E-Mails vom E-Mail-System selbst.

Zusätzlich zum normalen regulären Ausdruck gibt es einige spezielle Möglichkeiten, Bedingungen zu schreiben. Dazu muss die Bedingung mit einem der folgenden Zeichen beginnen:

!

Negiert die Bedingung. Beispiel:

```
* !^FROM_DAEMON
```

Diese Bedingung würde auf alle E-Mails zutreffen, die nicht vom E-Mail-System kommen.

?

Wertet den Rückgabewert eines Programms aus. Beispiel:

```
* ? test ! -d mailbox
```

Diese Bedingung würde zutreffen, wenn das Verzeichnis `mailbox` nicht existiert.

<, >

Prüft, ob die Größe einer E-Mail größer (>) oder kleiner (<) als die angegebene Anzahl Bytes ist. Beispiel:

```
* < 1000
```

Diese Bedingung würde auf alle E-Mails zutreffen, die kleiner als 1000 Bytes sind.

Die Aktion

Eine Aktionzeile wird dann ausgeführt, wenn alle Bedingungen der Regel zutreffen. Procmail bietet verschiedene Möglichkeiten, E-Mails zu behandeln. Die einfachste besteht in der Speicherung von E-Mail in einem Postfach, entweder im Mbox-Format oder im Maildir-Format. Dazu gibt man als Aktionszeile einfach den Dateinamen für Mbox-Ordner oder das Verzeichnis mit abschließendem Schrägstrich für Maildir-Ordner an, zum Beispiel:

```
:0:
* ^Subject:.*test
test
```

Diese Regel speichert alle E-Mails mit dem Betreff »test« im Mbox-Ordner *test*.

Neben dem Speichern in einem Postfach kann man eine E-Mail auch an eine andere Adresse weiterleiten. Dazu stellt man der E-Mail Adresse ein `!` in der Aktionszeile voran, zum Beispiel:

```
:0
* ^Subject:.*test
! foo@test.domain
```

Diese Regel schickt alle E-Mails mit dem Betreff »test« an die E-Mail-Adresse *foo@test.domain*.

Wenn die E-Mail durch ein Programm weiterverarbeitet werden soll, stellt man dem Programmnamen ein | voran, zum Beispiel:

```
:0
* ^Subject:.*test
| /usr/local/bin/test.sh
```

Das bewirkt die Weiterleitung der E-Mail an das Programm *test.sh*.

Als letzte Möglichkeit kann man auf einen so genannten Block verweisen. Ein Block ist durch {} eingerahmt und ermöglicht es, Verschachtelungen oder verkettete Filter zu erstellen, zum Beispiel:

```
:0
* ^X-Mailinglist: linux
{
  :0 c:
  linux-liste

  :0
  ! hans@wwww1.com
}
```

In diesem Beispiel werden alle E-Mails, die den Header `X-Mailinglist: linux` haben, an das Postfach *linux-liste* und an *hans@wwww1.com* weitergeleitet. Wichtig ist hierbei das `:0 c:` bei der ersten Regel, damit die E-Mail nicht endgültig gespeichert wird, sondern nur eine Kopie der E-Mail gespeichert wird. Sonst wäre später keine E-Mail mehr zum Weiterleiten da.

Procmail als Mittel gegen Spam

Nachfolgend werden mehrere handliche kleine Rezepte aufgeführt und erläutert, mit denen man Spam-Probleme beseitigen kann. Sie sind alle völlig unabhängig voneinander und einzeln lauffähig. Während des Testens neuer Regeln empfiehlt es sich, am Anfang der Regeldatei eine Sicherheitskopie aller eingehenden E-Mails zu machen, etwa so:

```
:0c:
backup
```

Filtern nach Betreff

Die erste Aufgabe ist relativ leicht. Ziel ist es, alle E-Mails abzufangen, die das Wort »Viagra« im Betreff beinhalten. Die einfachste dieser Varianten ist sicherlich:

```
:0:
* ^Subject:.*viagra
spam
```

die alle E-Mails mit dem Wort Viagra in das Spam-Postfach schiebt.

Allerdings sind Spammer geschickter geworden und variieren die Schreibweise von Viagra, um einfache Wortfilter auszutricksen. Folgende Regel ist daher etwas effektiver:

```
:0:
^Subject:.*[Vv][1jl\|][aA\@][Gg][Rr][Aa\@]
spam
```

Diese Version deckt schon mal einige alternative Schreibvarianten des Worts Viagra ab. Da es aber nicht Aufgabe dieses Kapitels ist, einen möglichst weit reichenden regulären Ausdruck für Viagra zu entwickeln, soll dieses Beispiel ausreichend sein.

Blacklists in Procmail

Mit Procmail ist es relativ einfach, eine Blacklist von Adressen zu pflegen, von denen man keine E-Mail erhalten möchte. Dazu bedient man sich hier zusätzlich der externen Programme egrep und `formail`.

Beispiel:

```
BLACKLIST=/home/blacklist   # Diese Datei enthält die Blacklist.

:0:
* ? (formail -x From: -x Reply-To: -x Sender: -x From | egrep -q -f $BLACKLIST)
spam
```

Diese Filterregel extrahiert aus einer E-Mail alle Absenderadressen und vergleicht sie mit der Blacklist. Die Blacklist-Datei wird in der Variablen `$BLACKLIST` gespeichert und enthält reguläre Ausdrücke, einen pro Zeile. Hier ein simples Beispiel für den Inhalt einer solchen Blacklist:

```
spam.*@spammer.de
@*.tw
@*.jp
```

Der erste Eintrag blockt alles, was von der E-Mail-Domain *spammer.de* kommt und mit »spam« im Benutzerteil beginnt. Die letzten zwei Beispieleinträge blocken die Top-Level-Domains von Taiwan und Japan.

Kaputte Message-ID

Viele Spam-E-Mails haben eine kaputte Message-ID ohne das erforderliche @-Zeichen. Dieses Rezept filtert E-Mails, die eine solche defekte ID haben:

```
:0:
* ^Message-Id:.*<[^@]*>
spam
```

E-Mails mit vielen Empfängern

Häufig ist es ein schlechtes Zeichen, wenn eine E-Mail an unzählige Empfänger geht. Außerdem ist das eine beliebte Angriffsmethode der Spammer, um über Whitelist-Einträge an E-Mail-Filtern vorbeizukommen. Das folgende Rezept blockt alle E-Mails mit zwölf oder mehr Empfängern:

```
:0:
* ^(To|Cc):.*,.*,.*,.*,.*,.*,.*,.*,.*,.*,.*,.*,
spam
```

E-Mails ohne Empfänger

Spam-E-Mails werden oft ohne einen Empfänger im Header verschickt. Daher bietet es sich an, E-Mails, die sich ohne Empfänger präsentieren, als Spam einzustufen, zum Beispiel:

```
:0:
* ^To ?: *$
spam
```

E-Mails mit Procmail auf Viren prüfen

Dieses Rezept verwendet den Open Source-Virenscanner ClamAV, um E-Mails auf Viren zu prüfen. Ein angenehmer Nebeneffekt ist, dass je nach ClamAV-Konfiguration auch durch Passwort geschützte Archive geblockt werden, die gern zum Verschicken von Viren missbraucht werden.

```
:0:
* ? clamscan --quiet -
virus
```

Weitere Informationen zur Einbindung von ClamAV in Procmail und anderswo findet man in Kapitel 7, *Virenscanner*.

E-Mails von bekannten Spam-Programmen

Das folgende Rezept ist etwas praxisorientierter und filtert alle E-Mails von bestimmten E-Mail-Versenderprogrammen.

```
:0:
* ^X-Mailer:.*((MassE-Mail)|Extractor|Floodgate|(Emailer
Platinum)|JumboMail|(Advanced Mass Sender)|GreenRider|(FoxMail .*cn))
spam
```

Dabei prüft Procmail den Kopf der E-Mail auf die verwendete Software und sortiert somit Massenversender-Progamme von vornherein aus.

E-Mails mit unverständlichen Zeichensätzen

Wenn man nicht in der Lage ist, chinesische, japanische oder koreanische E-Mails zu lesen, bekommt man sicher auch keine legitimen E-Mails, die in diesen Sprachen geschrieben sind. Die meisten dieser E-Mails lassen sich über ihre Zeichensätze identifizieren; das versucht das folgende Rezept auszunutzen.

```
UNREADABLE='[^?"]*big5|iso-2022-jp|ISO-2022-KR|euc-kr|gb2312|ks_c_5601-1987'

:0:
* 1^0 $ ^Subject:.*=\?($UNREADABLE)
* 1^0 $ ^Content-Type:.*charset="?($UNREADABLE)
```

```
spam

:0:
* ^Content-Type:.*multipart
* B ?? $ ^Content-Type:.*^?.*charset="?($UNREADABLE)
spam
```

Zuerst wird eine Liste der Zeichensätze definiert, die man nicht lesen kann. Danach wird mit verschiedenen Methoden versucht, diese Zeichensätze zu erkennen. Weitere Zeichensätze kann man natürlich bei Bedarf hinzufügen. Aber man sollte auch bedenken, dass Anwender aus den betroffenen Ländern, die beispielsweise eine E-Mail auf Englisch verfassen, in der Regel trotzdem diese Zeichensätze verwenden werden, da so ziemlich jeder Zeichensatz der Welt zumindest die ASCII-Zeichen enthält.

E-Mails aus fast nur Nicht-ASCII-Zeichen

Etwas eleganter als die Filterung nach Zeichensatz ist diese Lösung: Eine E-Mail, die fast nur aus Zeichen besteht, die nicht im ASCII-Satz vorkommen (beispielsweise Japanisch oder Chinesisch, aber auch Russisch), ist mit hoher Wahrscheinlichkeit Spam. Wenn man also nicht gerade in diesen Sprachen kommuniziert, kann das folgende Rezept sehr nützlich sein. Es erkennt alle E-Mails, die mehr als fünf Prozent solcher Zeichen beinhalten. Dazu verwendet es das Procmail-Scoring-System, das in der Manpage *procmailsc* beschrieben wird. Vereinfacht gesagt, ist die Zahl vor dem ^ die Punktzahl, die für eine Übereinstimmung mit der dahinter stehenden Bedingung vergeben wird. Wenn die Punktzahl am Ende positiv ist, ist die Bedingung erfüllt.

```
:0BD
*  -1^1 .
*   2^1 =[0-9A-F][0-9A-F]
*  20^1 [¡¢£¤¥¦§¨©ª«¬®¯°±²³´µ¶¸¹º»¼½¾¿]
*  20^1 [ÀÁÂÃÄÅÆÇÈÉÊËÌÍÎÏÐÑÒÓÔÕ×ØÙÚÛÜÝÞ]
*  20^1 [àáâãäåæçèéêëìíîïðñòóôõ÷øùúûýþÿ]
*  20^1 =[A-F][0-9A-F]
* -20^1 =(E4|F6|FC|C4|D6|DC|DF)
spam
```

Filtern von bestimmten Anhängen

In Anhängen verbergen sich immer wieder Viren oder Trojaner. Aus diesem Grund ist man meistens gewillt, bestimmte Dateitypen von vornherein zu filtern. Dafür sorgt folgendes Rezept:

```
:0 B:
* name=.*\.(vbs\"|wsf\"|vbe\"|wsh\"|hta\"|scr\"|pif\"|shs\"|bat\"|bas\"|scr\"|dll\
")
blocked
```

Diese Liste ist leicht zu erweitern und an die eigenen Bedürfnisse anzupassen. Das \" stellt jeweils sicher, dass die angegebenen Buchstaben am Ende des Dateinamens stehen.

Filtern von bestimmten Inhalten

Bestimmte Inhalte beziehungsweise Wörter lassen mit ziemlicher Sicherheit auf Spam schließen. Mit diesem Rezept ist es möglich, eine Blockliste zu erstellen, mit der E-Mails gefiltert werden können, die bestimmte Wörter enthalten:

```
BADWORDS=badwords   # Diese Datei enthält die Badwords.

:0 B:
* ? egrep -q -f $BADWORDS
spam
```

In der Datei *badwords* befinden sich dann reguläre Ausdrücke, einer pro Zeile, die nacheinander von egrep durchgegangen werden. Hier ein kleines Beispiel für eine solche Datei:

```
foo
[Vv][1jl\|][aA\@][Gg][Rr][Aa\@]
```

Die erste Zeile blockt alle E-Mails, die das Wort »foo« enthalten. Die zweite Zeile enthält einen etwas komplexeren regulären Ausdruck, der das Wort Viagra erfassen soll.

DNS-Blackhole-Listen mit Procmail abfragen

Zum Ende dieser Rezeptsammlung ein kleines Schmankerl. Dieses Rezept erlaubt es mittels Procmail und dem Befehl host, eine DNS-Blackhole-Liste abzufragen und abhängig vom Ergebnis die E-Mail einzusortieren. Um das vollbringen zu können, benötigt das Rezept zwei Informationen: den Hostnamen des letzten Mailservers, der die E-Mail angenommen hat, und die Liste, die abgefragt werden soll. Dieses Beispiel verwendet dazu die NJABL-Blacklist (*http://www.njabl.org*). Die Idee des Rezepts stammt von der Webseite *http://www.benya.com/procmail/*. Das dortige Beispiel wurde hier noch etwas angepasst und optimiert.

```
# Hostname wird zur Erkennung der letzten Received-Zeile benötigt.
HOSTNAME=ned.snow-crash.org
# Welche DNSBL soll verwendet werden?
DNSBL=dnsbl.njabl.org

SENDERIP = 'formail -c -XReceived | grep "by $HOSTNAME" | grep -v "from $HOSTNAME"
| \
    sed "s/^Received: from .*\[\([0-9]*\.[0-9]*\.[0-9]*\.[0-9]*\)\].*by $HOSTNAME.
*$/\1/"'

:0
```

```
* ! SENDERIP ?? ^^[0-9]*\.[0-9]*\.[0-9]*\.[0-9]*^^
{
    # Löschen, wenn es keine IP-Adresse ist.
    SENDERIP =
}

:0
* ! SENDERIP ?? ^^^^
{
    # IP-Adresse umkehren.
    SENDER_REVERSED = 'expr "$SENDERIP" | \
        sed "s/\([0-9]*\)\.\([0-9]*\)\.\([0-9]*\)\.\([0-9]*\)/\4.\3.\2.\1/"'

    DNSBL_RESULT = 'host "$SENDER_REVERSED".$DNSBL | \
        sed "s/^.*\(127\.0\.0\.[0-9]*\)$/\1/"'

    :0
    * DNSBL_RESULT ?? ^^127.0.0.[0-9]*^^
    {
        LOG = "sender $SENDERIP is listed in $DNSBL
"
        :0:
        spam
    }
}
```

In der Variablen `HOSTNAME` muss der eigene Mailserver stehen, der die E-Mail annimmt, und in der Variablen `DNSBL` steht die verwendete DNS-Blackhole-Liste. Hier sieht man auch ein Beispiel zur Verwendung der Variablen `LOG`. Diese wird von Procmail dazu benutzt, um eigene Log-Einträge generieren zu können. Alles was dieser Variablen zugewiesen wird, erscheint später in der Log-Datei.

Fertige Anti-Spam-Lösungen für Procmail

Es gibt einige fertige Anti-Spam-Lösungen, die zum Gebrauch mit Procmail gedacht sind. Nachfolgend werden drei von ihnen vorgestellt, und ihre Features werden zusammengefasst. Für ihre Konfiguration und Installation wird auf die Websites der jeweiligen Entwickler verwiesen.

Spastic

Spastic ist eine Anti-Spam-Lösung für Procmail. Es unterstützt eine Reihe von Möglichkeiten zur Spam-Bekämpfung:

- Header- und Body-basierte Filterung
- Vordefinierte Filter, die ein schnelles Erstellen von eigenen Filtern ermöglichen
- Whitelists

- Prüfen von E-Mail-Adressen auf Gültigkeit
- Statistikfunktionen
- Einbindung von ClamAV als Antivirenlösung

Website: *http://spastic.sourceforge.net/*

SpamBouncer

SpamBouncer ist ein Satz fertiger Procmail-Regeln, die es ermöglichen, Spam auf Grund seines Inhalts zu markieren oder zu löschen. Folgende Features besitzt diese Anti-Spam-Lösung:

- Einbindung von externen Whitelists
- DNS-Blackhole-Lists
- Erkennung von japanischen, chinesischen und russischen Inhalten
- Erkennung von Windows-Exploits
- Header- und Body-basierte Filterung

Website: *http://www.spambouncer.org*

NiX Spam

NiX Spam wurde von der deutschen Computerzeitschrift iX als Anti-Spam-Lösung im Rahmen eines Artikels entwickelt und wird seit mehreren Jahren immer wieder aktualisiert. Es ist auf großen Durchsatz von E-Mails optimiert und verwendet Prüfsummenlisten und IP-Blacklists, um sowohl die Last als auch den Durchsatz möglichst optimal zu halten. Folgende Features sind enthalten:

- Prüfsummenlisten
- Header- und MIME-Analyse
- White- und Blacklists
- IP-Blocklisten
- Automatisch erzeugte Blacklists
- Body-Analyse

Website: *http://www.heise.de/ix/nixspam/*