Postfix

- Das Handbuch -

Modul

Mehrplatzbetriebssysteme

Schule

Berufsbildende Schule Neustadt

Klasse

FSI-00

Abgabetermin

2004-03-30

Autoren

Philipp Rung
Simone Schäfer
Jochen Weinheimer

Inhaltsverzeichnis

Vorwort

Dieses Handbuch wurde in Form einer Projektarbeit im Fach „Mehrplatzbetriebssysteme“ an der Berufsbildenen Schule in Neustadt, zur Weiterbildung zum „Staatlich geprüften Betriebswirt, Fachrichtung Informationsmanagement und Informationsverarbeitung“ erstellt.
Bei dieser Projektarbeit wurde uns von der Lehrerschaft auferlegt, einen virensicheren Postfix-eMail-Server unter dem Betriebsystem SuSE Linux 9.0 sowie entsprechenden Clients aus vorgegebenen Mitteln wie Rechnern und Software selbstständig aufzubauen und einzurichten.

Kapitel 1 Postfix

1.1 Die Geschichte

Postfix wurde von Wietse Venema, der auch die in der Unix-Welt bekannten Programme „tcp_wrappers“ und „Satan“ geschrieben hat, entwickelt.

Postfix fing sein Leben unter der Bezeichnung „Vmailer“ an. Wietse Venema hat die Software unter dem IBM GPL (General Public License) herausgegeben. Da jedoch Rechtsanwälte der Firma IBM entdeckten, dass der Name „Vmailer“ einer vorhandenen Schutzmarke zu ähnlich war, musste der Name „VMailer“ geändert werden.
Von da an wurden von Wietse Venema und seinem Team viele Monate investiert um dem Projekt einen neuen Namen zu geben, denn jede vorgeschlagene Bezeichnung wurde von der IBM abgelehnt, worauf zu guter letzt eine neue Taktik angewandt werden musste. Daraus resultierte, dass das Programm nun unter zwei Namen bekannt ist: „IBM Secure Mailer“ UND „Postfix“.

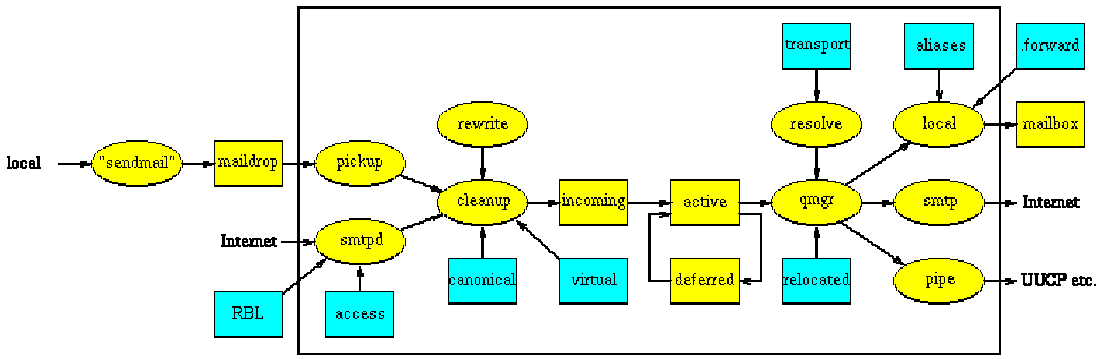
1.2 Was ist Postfix?

Postfix ist ein sogenannter MTA, ein Mail Transfer Agent. Er kümmert sich um den Versand und Empfang von eMails. Er nimmt die eMails von den Clients im Netzwerk an, verteilt sie an die auf ihm angelegten Mailboxen oder übergibt sie an einen anderen MTA. Des Weiteren kann er auch eMails von anderen MTAs außerhalb des eigenen Netzes annehmen, weiterleiten und sie in seinen lokalen Mailboxen ablegen.

1.3 Der Aufbau

Wietse's Ziel bei der Entwicklung von Postfix war ein schnelles, einfach zu administrierendes und sicheres Programm(paket) zu entwickeln, das so weit wie möglich zu Sendmail kompatibel sein soll. Das Interessanteste an Postfix ist sein innerer Aufbau: Es besteht aus mehreren kleinen Programmen, die über sogenannte LINUX-Domain-Sockets kommunizieren. Auf diese Weise ist es viel einfacher Probleme, Fehler oder Sicherheitsmängel in den Griff zu bekommen.

Ebenfalls aus Sicherheitsgründen arbeitet Postfix mit vier verschiedenen Queues (Warteschlangen): "maildrop", "incoming", "active" und "deferred". Lokal gesendete Mails landen in "maildrop" und werden von dort in die "incoming"-Queue kopiert, nachdem sie regelbasiert auf Größe, Inhalt und anderes überprüft wurden. In der "active" Queue landen die Mails, die der Queue-Manager gerade bearbeitet und ausliefert (lokal oder remote). Nachrichten, die Postfix nicht ausliefern kann (Dienst des Zielmailservers reagiert nicht, keine Route, keine Netzverbindung, ...), landen in der "deferred" Queue. Da Postfix immer nur eine Mail gleichzeitig bearbeitet und die "active" Queue klein hält, ist es unempfindlich gegen Ressourcenknappheit. Das Bearbeiten/Ausliefern von Mails kann also in keinem Fall, beispielsweise wegen eines vollen Dateisystems, blockiert werden.



Die Grafik zeigt den modularen Aufbau von Postfix. Hierbei bedeuten:

- gelbe Ellipsen Programme
- gelbe Kästen Mail-Queues oder -Dateien
- blaue Kästen (Nachschlage-) Tabellen
- Programme in der umrandeten Box laufen unter der Kontrolle des Postfix *master* Daemons, Dateien in diesem Kasten gehören dem Postfix-Mail-System.

1.4 Installation

Postfix muss in dem Sinne nicht installiert werden, da er als fester Bestandteil der Grundinstallation von SuSE Linux 9.0 automatisch mitinstalliert wird.

1.5 Konfiguration

1.5.1 Die Konfigurationsdatei main.cf

Die Datei main.cf, zu finden im Pfad „/etc/postfix/“, ist die Hauptkonfigurations-Datei von Postfix. In ihr sind alle wichtigen und notwendigen Einstellungen vorzunehmen. Die Datei erklärt in englischer Sprache direkt den Großteil der verschiedenen Einstellungsmöglichkeiten von Postfix und zeigt durch Beispieleinstellungen die richtige Syntax der einzelnen Parameter.

Insgesamt gibt es über 300 verschiedene Parameter die in der Datei main.cf vorgenommen werden könnten, um den Server zu konfigurieren. Da allerdings nicht alle Einstellungsmöglichkeiten für unseren Server konfiguriert werden müssen, da wir diese entweder nicht kennen, nicht benötigen oder diese schon voreingestellt sind, sind hier nur jene Parameter erläutert die für unseren Server von Bedeutung sind oder die einer Erklärung würdig erscheinen.

```
queue_directory = /var/spool/postfix
```

dient der Einstellung des Verzeichnisses in dem die zu sendenden Mails zwischengelagert werden.

```
command_directory = /usr/sbin
```

dient zur Einstellung des Verzeichnisses in welchem sich die Kommandos zur Steuerung des Postfix-Servers befinden (postXXX-Kommandos)
dient der Einstellung des Pfades in dem sich alle Postfix-Programme

```
daemon_directory = /usr/lib/postfix
```

befinden

```
mail_owner = postfix
```

legt den Besitzer der Warteschlangen (queues) und der meisten Postfix-Programme fest.
Hier darf auf keinen Fall als Besitzer der User "root" angegeben werden. Besser ist es einen neuen Benutzer anzulegen der nur für Postfix zuständig ist und sich selbst nicht einloggen kann.
Daher wird standardmäßig bei der Installation von Postfix der User "postfix" automatisch angelegt.

```
unknown_local_recipient_reject_code = 450
```

gibt den SMTP-Server-Antwort-Code an, nach dem im Falle einer nicht-zustellbaren Mail gehandelt werden soll.
Hier gibt es zwei Möglichkeiten:
code = 550 (reject mail --> Mail direkt an Absender zurücksenden)
code = 450 (try again later --> Mail zurückstellen und später die Zustellung versuchen).

Original-Postfix-Konfigurationsdatei „main.cf“

```
mynetworks  = 172.16.0.0/16, 127.0.0.0/8, 127.0.0.1/8
```

legt fest, welchen Netzen Postfix vertrauen soll und von denen Postfix demnach Mails annehmen darf. Unser Netz ist das Klasse B-Netz „172.16.“ Die 127. Ist hierbei der interne Loop-Back des eMail-Servers selbst und muss mit eingetragen bleiben.

```
mail_spool_directory = /var/mail
```

gibt an wo sich die Mailboxen im LINUX-Stil befinden.

```
myhostname = linux.bbsnw.de
```

legt den Namen des Postfix-Servers fest, in unserem Falle „linux.bbsnw.de“.

```
program_directory = /usr/lib/postfix
```

gibt an in welchem Verzeichnis sich das Postfix-Programm befindet.

```
mydestination = bbsnw.de
```

legt fest für welche Domänen der Server die eMails annehmen soll. Für uns ist dies die Domäne „bbsnw.de“.

```
relayhost = [mail.bbsnw.de]
```

gibt an, an welchen RelayHost die eMails gesendet werden sollen die nicht lokal zugestellt werden können. Wir haben an dieser Stelle den Server „mail.bbsnw.de“ eingetragen, falls es diesen Server einmal zur Weiterleitung der eMails unseres Servers geben wird.

```
alias_maps = hash:/etc/aliases
```

gibt an wo sich die Alias-Datenbank für die Zuordnung von eMail-Adressen zu lokal angelegten Benutzern befindet

```
mailbox_size_limit = 0
```

dient der Einschränkung der Mailbox-Grösse. Wird als Wert "0" eingetragen, so unterliegen die Mailboxen keiner Einschränkung

```
message_size_limit = 10240000
```

legt die maximale Grösse von eMails in Bytes fest.

```
inet_interfaces = all
```

gibt die Adressen von Netzwerk-Schnittstellen an auf denen der Postfix-Server eMails empfängt.

Original-Postfix-Konfigurationsdatei „main.cf“

1.5.2 Die Konfigurationsdatei master.cf

Die Datei master.cf ist die Haupt-Prozess-Konfigurationsdatei von Postfix. In ihr wird beschrieben wie die Postfix-Dämonprogramme laufen sollen, wie viele Ressourcen Postfix verbrauchen darf, wie viele Prozesse maximal gleichzeitig laufen dürfen, usw.

Standardmäßig sind in dieser Datei alle notwendigen Konfigurationen für einen „normalen“ Betrieb des Servers bereits vorgenommen. Sollten weitere Programme, wie z.B. AMaViS (Virenscanner), dem Postfix-Server angekoppelt werden, so werden die vorzunehmenden Eintragungen, die zur Kommunikation zwischen den verschiedenen Programmen mit dem Postfix-Server vorzunehmen sind, normalerweise von den Anbietern der Programme mitgeliefert.

```
# ==========================================================================
# service  type    private unpriv   chroot  wakeup  maxproc  command + args
#                  (yes)   (yes)    (yes)   (never)  (100)
# ==========================================================================
smtp       inet    n       -        n       -       -        smtpd
pickup     fifo    n       -        n       60      1        pickup
cleanup    linux   n       -        n       -       0        cleanup
qmgr       fifo    n       -        n       300     1        qmgr
```

Erläuterung der Syntax:

<u>service:</u> Hier wird irgend ein Name eingetragen der für den im nächsten Schritt beschriebenen Transport-Typ gültig ist

<u>type:</u> Transport-Typ. Folgende Varianten sind möglich:
- „inet“ für Internet-Sockets
- „linux“ für LINUX-domain-sockets
- „fifo“ für benannte Pipes

<u>private:</u> Hier wird angegeben, ob der Zugang auf das Mail-System beschränkt ist oder nicht.

<u>unprivileged:</u> Hier wird angegeben, ob der Dienst mit „root“-Privilegien laufen soll oder als Besitzer des Postfix-Systems.

<u>chroot:</u> Beschreibt, ob das Programm mit „chrooted“ auf die Mail-Warteschlange laufen soll.

<u>wakeup:</u> Veranlasst das Programm automatisch nach der angegebenen Zeit in Sekunden zu starten.

<u>maxproc:</u> Gibt die maximale Anzahl von Prozessen an die das angegebene Programm gleichzeitig ausführen dürfen

<u>command + args:</u> Hier wird das auszuführende Programm mit eventuellen Zusatzparametern angegeben.

Original-Postfix-Konfigurationsdatei „master.cf“

1.5.3 Konfiguration der Alias-Datenbank

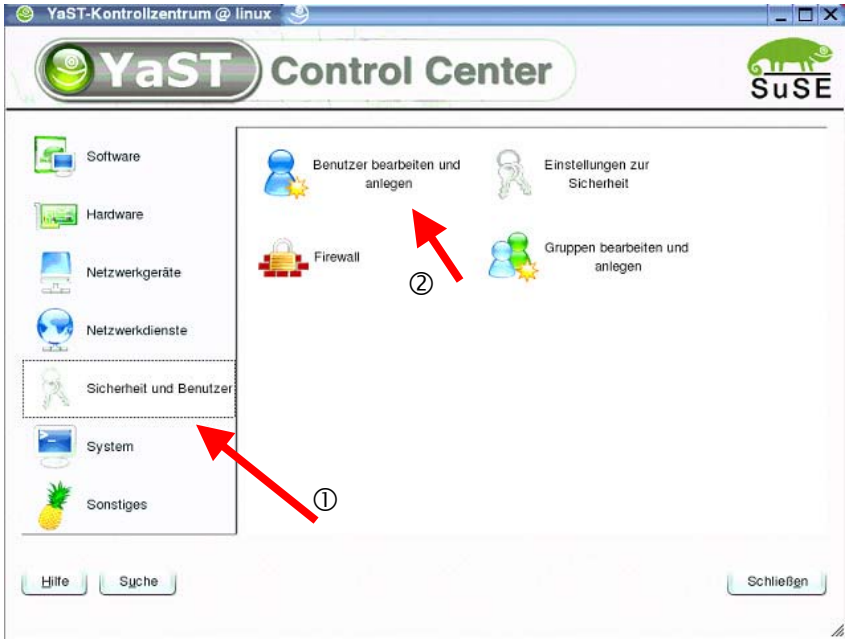
Damit Postfix seinen Zweck erfüllen kann und eMails, die an virtuelle Empfänger gesendet werden, an reell existierende und somit lokal angelegten eMail-Empfänger verteilen kann, müssen die virtuellen Empfängeradressen in einer sogenannten Alias-Datenbank lokalen Empfängern zugeordnet werden. Die Zuordnung eben dieser Adressen geschieht in der Datei „virtual“ im Verzeichnis „/etc/postfix“.

Damit die virtuellen Adressen zu lokalen Empfängern zugeordnet werden können müssen diese zuerst angelegt werden.

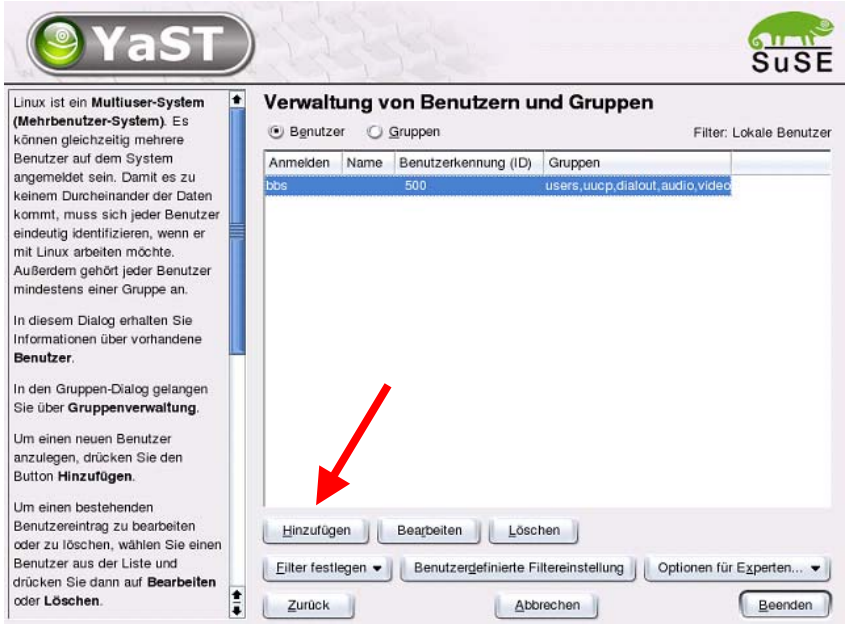
1.5.3.1 Anlegen neuer Benutzer

Um einen lokalen Empfänger und somit einen lokalen User einzurichten sind folgende Schritte nötig:

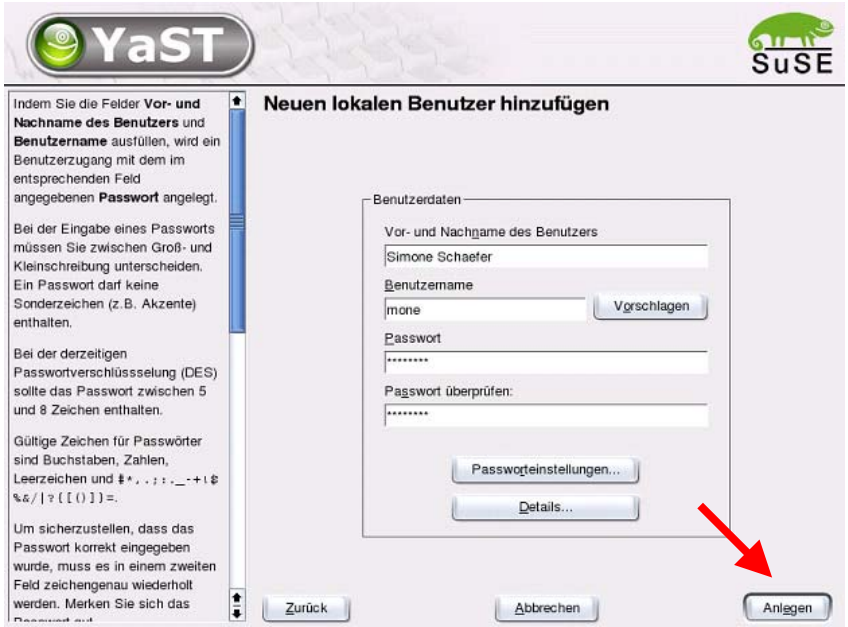
Zuerst muss das Konfigurationsprogramm von Linux, das YaST2 („Yet another Setup Tool 2“) gestartet werden. Klickt man darin auf den Punkt „Sicherheit und Benutzer“ im linken Menü, so wird folgende Anzeige dargestellt:



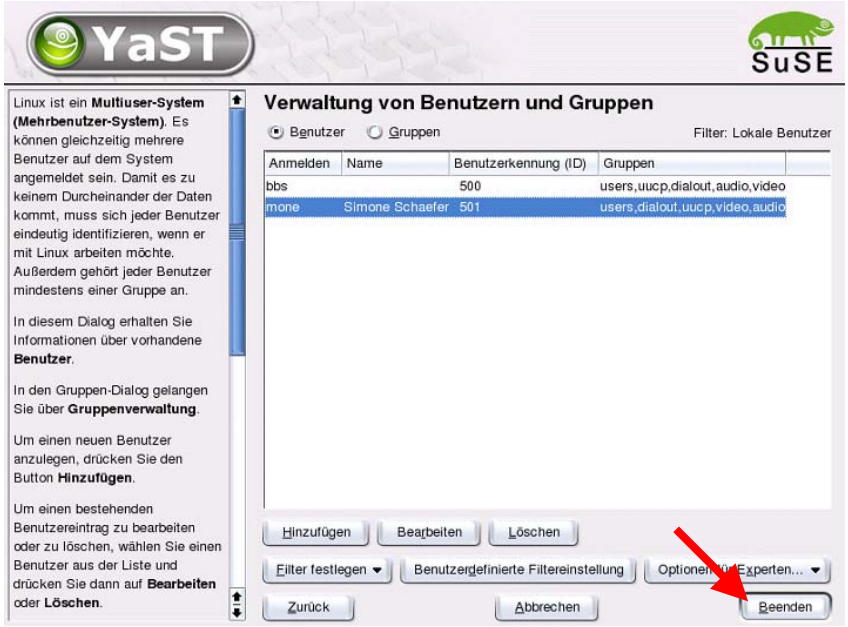
Durch klick auf den Punkt „Benutzer bearbeiten und anlegen“ im Auswahlmenü werden die bereits angelegten Benutzer angezeigt:



Durck Klick auf den Punkt „Hinzufügen“ erscheint die Maske zur Erfassung eines neuen Benutzers:



Hier müssen nun zur Erfassung eines neuen Benutzers auf dem Server zumindest der Benutzername und ein passendes Passwort (inklusive Passwortbestätigung) in die dafür vorgesehenen Felder eingegeben werden. Die Eingabe des Vor- und Nachnamens ist sicher sinnvoll, wird aber von Linux nicht zwingend gefordert. Hat man alle notwendigen Daten angegeben und klickt auf den Button „Anlegen“, so wird der Benutzer im Linux-System erstellt und in der Benutzerliste aufgeführt.



Nachdem nun alle Benutzer auf diese Art und Weise erfasst wurden, kann das die Benutzerverwaltung durch klick auf den Button „Beenden“ abgeschlossen werden.

1.5.3.2 Zuordnung lokaler Benutzer zu virtuellen eMail-Empfängern

Zur Zuordnung der verschiedenen virtuellen eMail-Empfänger zu den lokal angelegten Benutzern gibt es verschiedene Möglichkeiten, welche in der Datei „virtual“ selbst in englischer Schrift erklärt sind.

Am simpelsten ist das Standardvorgehen, bei dem einfach die virtuelle Adresse, getrennt durch einen oder mehrere Tabstops einem lokalen Konto, zugeordnet sind.

Beispiel:

```
# User Fips
fips@bbsnw.de      fips
```

Dem lokalen User „fips“ wird die virtuelle eMail-Empfängeradresse fips@bbsnw.de zugeordnet.

Da es nun heutzutage oft der Fall ist, dass ein User mehrere eMail-Adressen braucht, beispielsweise eine für private und eine für geschäftlichen Kontakt nach Außen, ist die Zuordnung mehrerer eMail-Adressen wie im folgenden Beispiel beschrieben möglich:

```
# User Mone
mone@bbsnw.de      mone
simmy@bbsnw.de     mone@bbsnw.de
chester@bbsnw.de   mone@bbsnw.de
```

Dem lokalen User „mone“ werden die virtuellen eMail-Adressen mone@bbsnw.de, simmy@bbsnw.de und chester@bbsnw.de zugeordnet,

wobei die eMails, welche an simmy@bbsnw.de und chester@bbsnw.de einfach an die virtuelle eMail-Adresse mone@bbsnw.de verwiesen werden.

Damit Postfix nun mit der erstellten bzw. geänderten Alias-Datenbank arbeiten kann, muss sie noch in ein Format konvertiert werden mit dem Postfix arbeiten kann. Hierzu ist das Kommando „postmap /etc/postfix/virtual“ vom Hauptbenutzer „root“ auf der Konsole zu starten. Das hierdurch gestartete Programm „postmap“ konvertiert die Datei „virtual“ in eben eine für Postfix interpretierbare Datei namens „virtual.db“, welche im Postfix-Konfigurationsverzeichnis „/etc/postfix“ abgelegt wird.

1.6 Start/Stop/Neustart

Der Postfix-Server wird mit dem Befehl

```
linux:/ # /etc/init.d/postfix start
```

gestartet.
Sollten eventuell Fehler in der Konfigurationsdatei vorhanden sein oder diverse Rechte auf Dateien falsch gesetzt sein, weist Postfix mit entsprechenden Kommentaren auf dem Bildschirm darauf hin. Schreibt der Server als Antwort auf den Startbefehl

```
Starting mail service (Postfix)                    done
linux:/ #
```

so ist der Start des Servers geglückt und das Mailsystem ist einsatzbereit.

Will man den Postfix-Server beenden, so lautet der Befehl hierzu

```
linux:/ # /etc/init.d/postfix stop
```

Auf ein geglücktes Beenden des Servers reagiert Postfix mit

```
Shutting down mail service (Postfix)               done
linux:/ #
```

Wurden nun zum Beispiel Änderungen an den Konfigurationsdateien des Servers bzw. der Alias-Datenbank vorgenommen, so muß der Postfix-Server neu gestartet werden damit die geänderten Dateien neu eingelesen werden.

Das Kommando hierzu lautet

```
linux:/ # /etc/init.d/postfix restart
```

Sind bei den Änderungen keine weiteren Fehler gemacht worden, sieht ein fehlerfrei vollzogener Neustart von Postfix wie folgt aus :

```
Shutting down mail service (Postfix)               done
Starting mail service (Postfix)                    done
linux:/ #
```

1.7 Test

Sollten nach dem Start von Postfix keine Beanstandungen des Programms bezüglich Fehler in der Konfiguration gemeldet worden sein, kann man ganz einfach mit dem Programm telnet überprüfen ob der MTA für den Empfang von eMails bereit ist.
Hierzu ist telnet mit der IP-Adresse des Servers und des Ports 25, dem SMTP-Port, aufzurufen. Unter SMTP versteht man das Simple Mail Transport Protocol. SMTP ist ein Server-zu-Server-Protokoll, welches für den Versand von Mails verantwortlich ist.

Hier ein Beispiel einer telnet-MTA-Testverbindung im Falle unseres Servers:

```
linux  # telnet 172.16.111.107 25
Trying 172.16.111.107...
Connected to 172.16.111.107.
Escape character is '^]'.
220 linux.bbsnw.de ESMTP Postfix
```

Antwortet der Server wie in unserem Beispiel mit “220 … ESMTP Postfix”, so ist der MTA für den Empfang bzw. die Entgegennahme von eMails bereit.

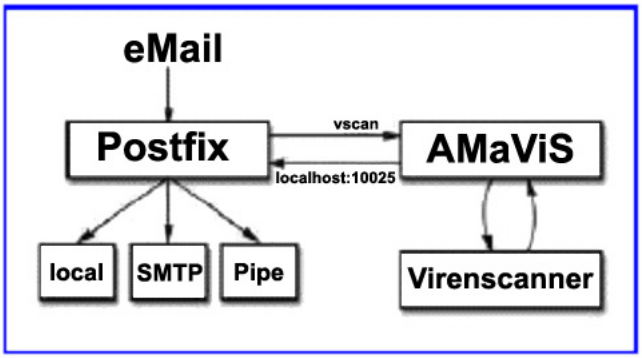
Kapitel 2 AMaViS - eMail-Virenscanner

In der heutigen Zeit, in der jeder der eine eMail-Adresse besitzt, ein potentielles Opfer von Virenattaken via eMail darstellt, ist es äußerst sinnvoll einen passenden Schutz gegen Viren und deren Verwandten auf einem Mailserver zu integrieren. Solch einen Schutz bietet „AMaViS“, „A Mail Virus Scanner“. AMaViS stellt hierbei eine Schnittstelle zwischen dem Postfix-MTA und einem der vielen existierenden Virenscanner dar.

2.1 Funktion

AMaViS nimmt eMails vom Mailserver, in userem Falle Postfix, entgegen, extrahiert die zu überprüfenden Daten und übergibt diese an einen von AMaViS unterstützten Virenscanner weiter.
Meldet der Virenscanner eine Infektion, stoppt AMaViS die Auslieferung, stellt die infizierte Datei in einen Quarantäne-Bereich und sendet je nach Konfiguration eine entsprechende Warnmeldung an den Versender und/oder den Empfänger der eMail.

Bildlich dargestellt sieht das Ganze folgendermaßen aus:



Bevor AMaViS selbst installiert wird bietet es sich an, zuerst einen Virenscanner zu installieren. AMaViS ist hierbei mit folgenden Virenscannern kompatibel und koppelbar:

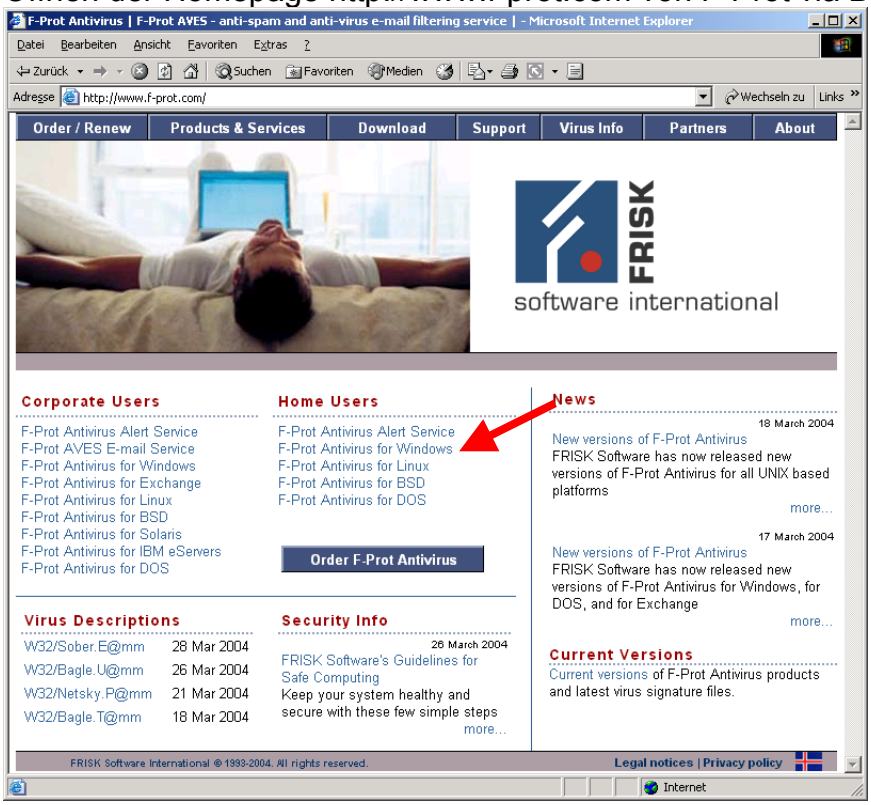
2.2 Virenscanner

Im Zuge unseres Projektes haben wir uns für den Virenscanner F-Prot von Frisk entschieden. F-Prot kann, zumindest in der „Home-User“-Variante, kostenlos im Internet unter www.f-prot.com als RPM-Datei heruntergeladen werden. Diese Möglichkeit haben wir zur Realisierung unseres Projektes in Anspruch genommen.

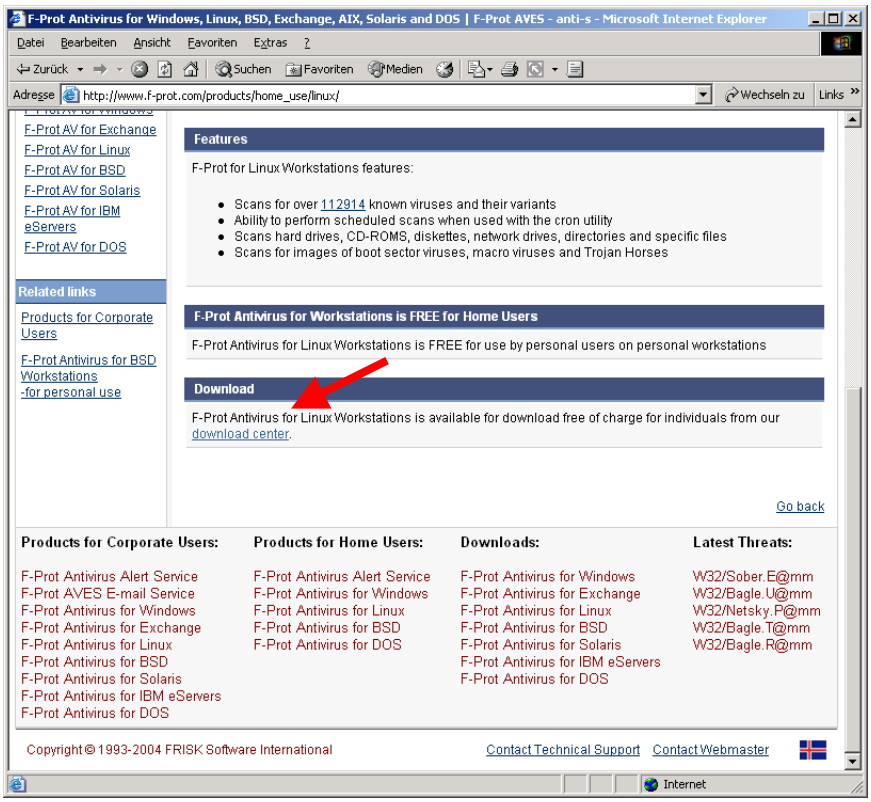
2.2.1 Download

Folgende Schritte müssen zum Download durchlaufen werden:

Öffnen der Homepage http://www.f-prot.com von F-Prot via Browser:

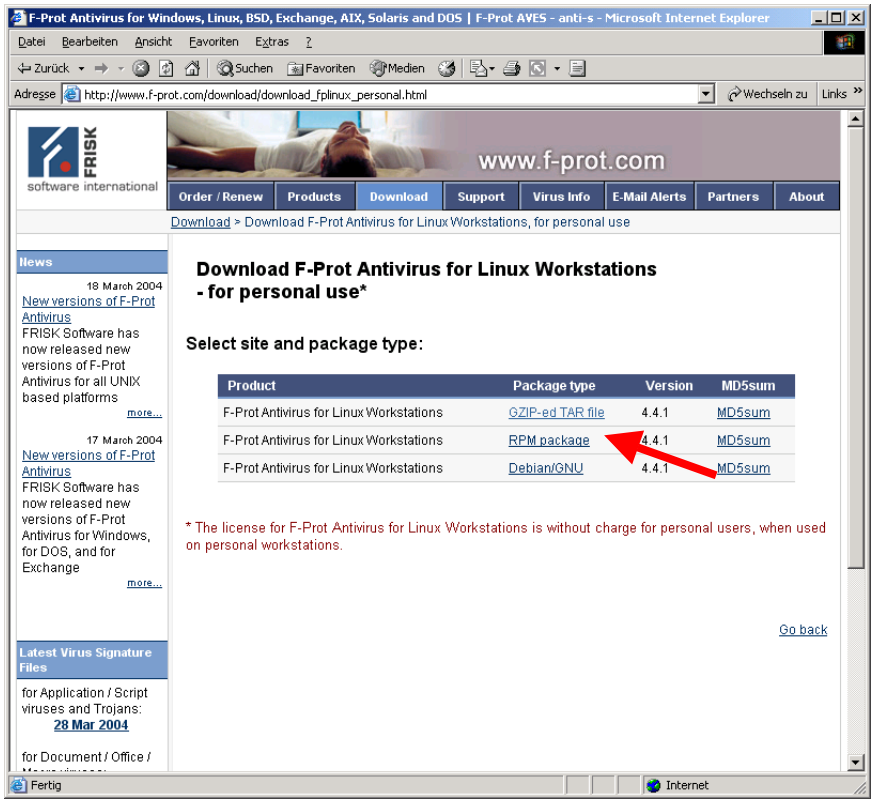


Um an das begehrte Installationspaket zu gelangen muss zunächst der Punkt „F-Prot Antivirus for Linux“ angewählt werden. Folgende Bildschirmanzeige erscheint:

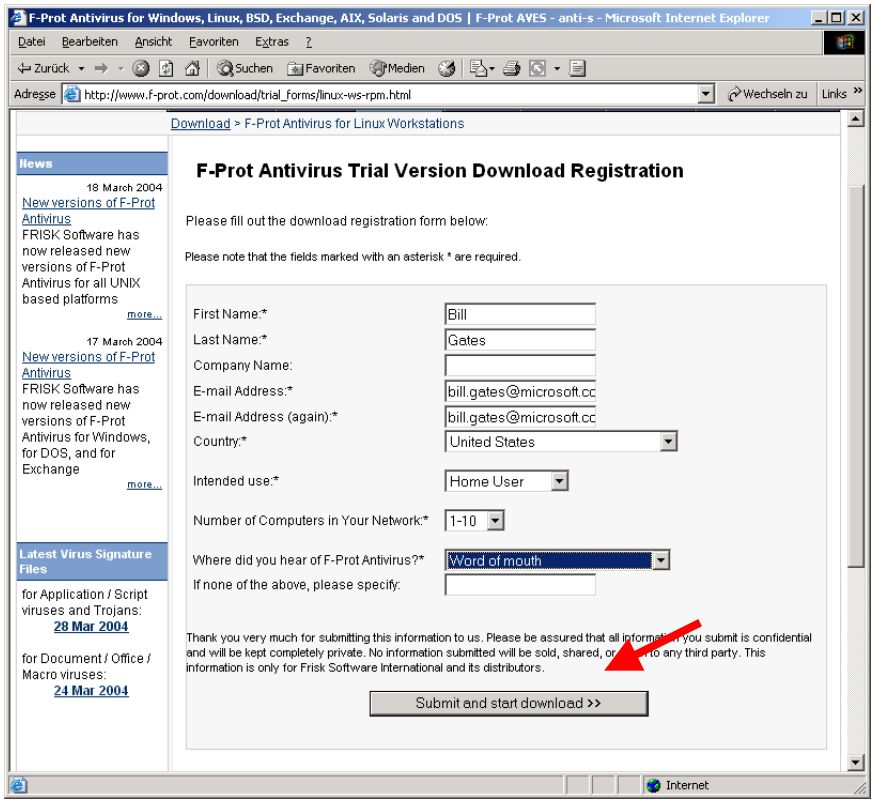


Hier werden nun einige Informationen über das herunterzuladende Paket dargestellt. Durch klick auf den Punkt „download center“ werden die

verschiedenen, downloadbaren Pakete angezeigt. Wir entscheiden uns an diesem Punkt der Einfachheit halber zum Download des RPM-Paketes.



Bevor F-Prot nun endgültig heruntergeladen werden kann muss noch das Registrierformular von F-Prot ausgefüllt werden.



Wurden alle von F-Prot gewünschten Angaben erfasst beginnt der Download nach einem Klick auf den Button „Submit and start download >>“. Nun muss der Download-Routine nur noch ein Pfad angegeben werden in welchem die RPM-Datei abgelegt werden soll.

2.2.2 Installation

Die Installationsdatei von F-Prot liegt nun in einem Verzeichnis. Um sie zu installieren muss der Befehl

```
rpm –Uhv *
```

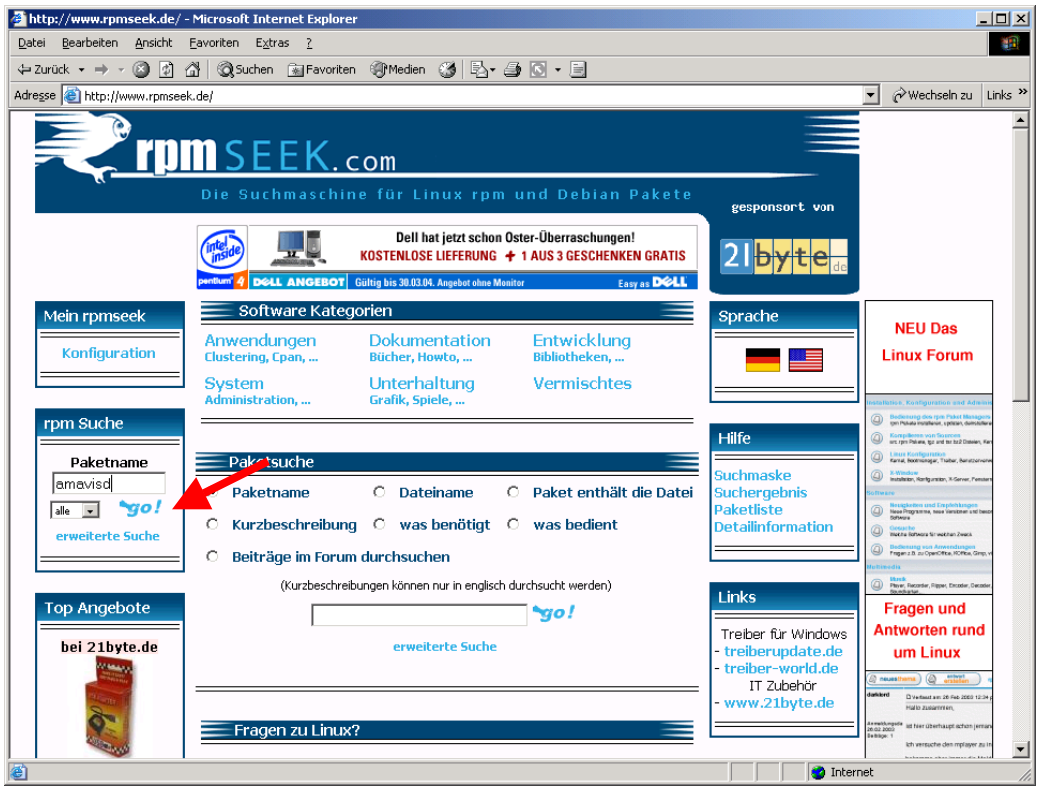
in dem Verzeichnis ausgeführt werden, in dem die Datei abgelegt ist. In unserem Falle hier als Beispiel:

```
linux:/home/bbs/F-Prot # rpm -Uhv *
```

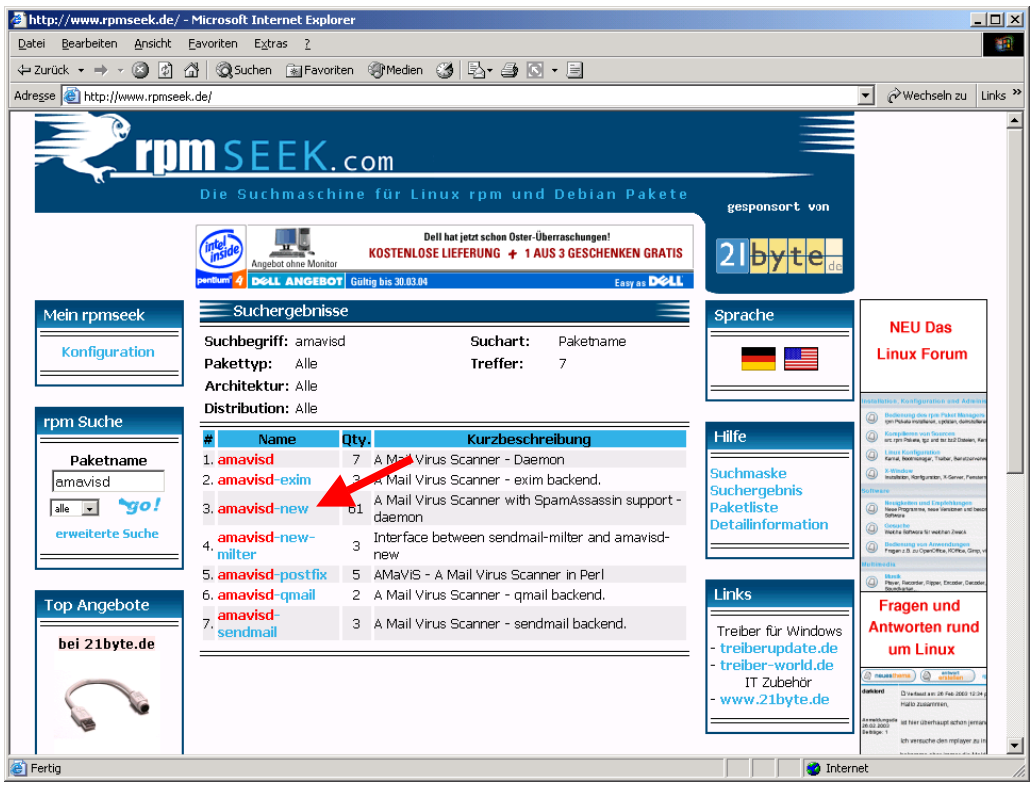
2.3 Download

Da AMaViS nicht im Paket von SuSE Linux 9.0 beinhaltet ist, muss er zuerst aus dem Internet heruntergeladen werden. Da es sich hierbei, wie bei den meissten Programmen für Linux, um ein Open-Source-Produkt handelt ist es kostenlos. Eine Site auf der AMaViS downgeloaded werden kann ist beispielsweise www.rpm-seek.de. Natürlich gibt es noch viele weitere Sites die rpm-Installationspakete anbieten. Daher haben wir hier nur als Beispiel die Schritte aufgeführt um AMaViS von www.rpm-seek.de zu erhalten:

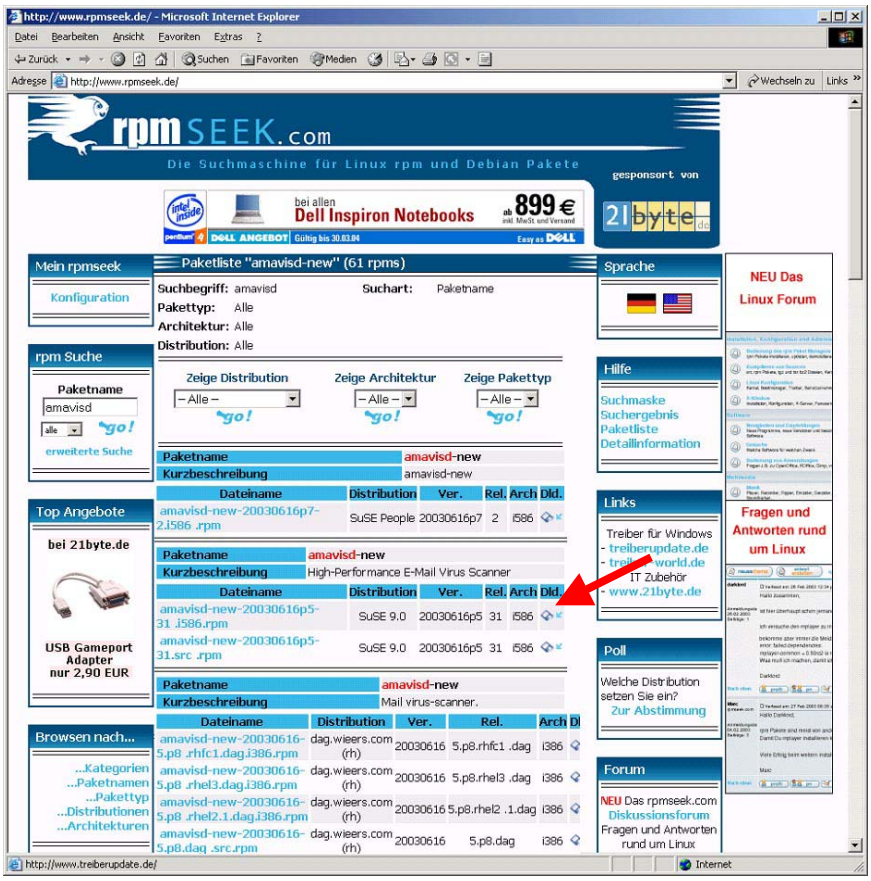
Auf www.rpm-seek.de muss zuerst nach dem Paket gesucht werden. Hierzu ist der Paketname in das dafür vorgesehene Feld einzugeben und auf den Button „*go!“ zu drücken.



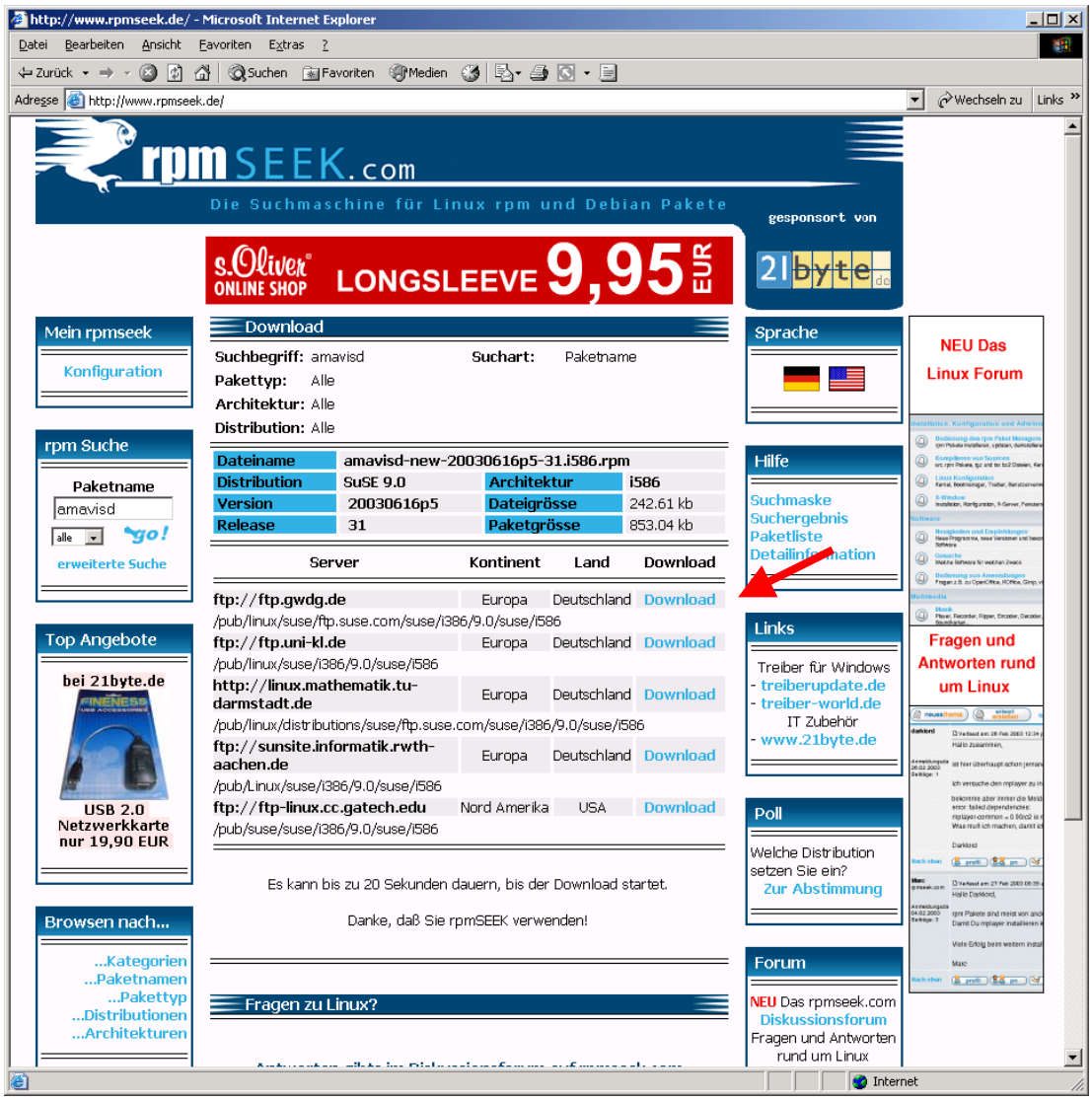
Hierauf erhält man eine Liste mit den verfügbaren Paketen. Das Paket das heruntergeladen werden soll ist „amavisd-new“, die neueste Version von AMAVIS.



Mit einem Klick auf den Namen des herunterzuladenden Pakets erscheint folgende Liste:



In dieser Liste sind nun die verschiedenen Plattformen aufgelistet für die das Paket erstellt wurde. Da wir SuSE Linux 9.0 im Einsatz haben, werden wir auch das dafür vorgesehene Paket durch klick auf den Dateinamen des Paketes wählen. Zu beachten ist allerdings, dass lediglich Dateien mit der Endung „.i586.rpm“ herunterzuladen sind, da es sich hierbei um die bereits kompilierten und voll einsatzfähigen Pakete handelt. Die Pakete mit der Endung „.src.rpm“ beinhalten nur den Source-Code, also den noch zu kompilierenden Quellcode der entsprechenden Pakete.



Zu guter letzt bekommt man noch eine Auswahl der verschiedenen Server angeboten, auf denen das gewünschte Paket abgelegt ist. Durch klick auf „Download“ hinter dem Server, von welchem man das Paket herunterladen möchte, beginnt die Download-Sequenz. Es ist hier nur noch der Pfad anzugeben, in dem das Paket gespeichert werden soll. Wir haben für eigene Zwecke ein neues Verzeichnis mit Namen „AMaViS“ im Pfad „/home/bbs“ erstellt und die Datei darin abgelegt.

2.4 Installation

Die Installationsdatei von AMaViS liegt nun in einem Verzeichnis. Um sie zu installieren muss der Befehl

```
rpm –Uhv *
```

in dem Verzeichnis ausgeführt werden, in dem die Datei liegt. In unserem Falle hier als Beispiel:

```
linux:/home/bbs/AMaVisD # rpm -Uhv *
```

Da AMaViS um seine Funktion erfüllen zu können selbst auf einige andere Linux-Programme zurückgreift, die eventuell nicht im Paket von Linux oder in unserem Falle im Paket von SuSE Linux 9.0 beinhaltet sind, reagiert die Installationsroutine mit dem oben angegebenen Befehl oft mit einer Fehlermeldung ähnlich der folgenden:

```
linux:/home/bbs/AMaVisD # rpm -Uhv *
Fehler: Failed dependencies:
        arc is needed by amavisd-new-20030616p5-31
        lha is needed by amavisd-new-20030616p5-31
        zoo is needed by amavisd-new-20030616p5-31
        perl-Convert-UUlib is needed by amavisd-new-20030616p5-31
        perl-IO-stringy is needed by amavisd-new-20030616p5-31
        perl-MIME-tools is needed by amavisd-new-20030616p5-31
        perl-MailTools is needed by amavisd-new-20030616p5-31
        perl-Archive-Tar is needed by amavisd-new-20030616p5-31
        perl-Linux-Syslog is needed by amavisd-new-20030616p5-31
        perl-Convert-TNEF is needed by amavisd-new-20030616p5-31
        perl-Archive-Zip is needed by amavisd-new-20030616p5-31
        perl-Net-Server is needed by amavisd-new-20030616p5-31
        perl-spamassassin is needed by amavisd-new-20030616p5-31
linux:/home/bbs/AmavisD #
```

Hierbei besagt der Fehler “Failed dependencies”, dass die Installation von AMaViS durch nicht erfüllte Abhängigkeiten zu den darunter aufgeführten Programmen nicht durchgeführt werden kann. Um die fehlenden Programme auf dem schnellsten Wege nachzuinstallieren bietet es sich an, diese auch auf Site www.rpm-seek.de oder eben einer anderen rpm-Site herunterzuladen. Um die ganze Sache zu beschleunigen bietet es sich weiterhin an, die fehlenden Programmdateien in das gleiche Verzeichnis zu speichern in dem schon die Installationsdatei von AMaViS liegt, da durch den Befehl „rpm –Uhv *“ alle in dem Verzeichnis liegenden Installationspakete installiert werden und somit weitere Abhängigkeiten zu eventuellen weiteren fehlenden Programmen angezeigt werden.

2.5 Konfiguration

Sind nun alle Programme installiert und die Installation von AMaViS ist geglückt, dann befindet sich im Verzeichnis /etc/ eine Datei namens „amavisd.conf“, die Konfigurationsdatei von AMaViS. In ihr sind nun alle notwendigen Einstellungen vorzunehmen um den reibungslosen Transfer zwischen AMaViS und dem Virenscanner sowie verschiedener Methoden einzustellen. Diese Datei ist genauso wie sämtliche Konfigurationsdateien und Tabellen von Postfix zum Teil schon vorkonfiguriert und direkt in englischer Sprache mit Beispielen dokumentiert. Aus diesem Grund und wegen der Tatsache, dass AMaViS weitaus mehr Möglichkeiten besitzt die wir für unser Projekt zu nutzen nicht in der Lage sind, beschränken wir uns nur auf die Erläuterung jener Einstellungen, welche wir an der Konfigurationsdatei vorgenommen haben:

```
$mydomain = 'bbsnw.de';
```

Hier wird die Domäne der eMails angegeben für die AMaViS die eMails prüfen lassen soll, in unserem Falle für die Domäne „bbsnw.de“.

```
$forward_method = 'smtp:127.0.0.1:10025';
```

Mittels dieses Eintrages werden emails von AMaViS an den Postfix-MTA über die Local-Host-Adresse bzw. den internen Netzwerk-LoopBack des Systems auf dem Port 10025 zurückgesandt.

```
$final_virus_destiny      = D_BOUNCE;
```

Durch diesen Eintrag wird AMaViS veranlasst, die email, sofern sie mit einem Virus behaftet ist, nicht an ihren Empfänger zu senden. Vielmehr wird eine Informationsmail von „amavisd-new“ an den Absender geschickt. Hierbei gibt es noch drei andere Möglichkeiten für den Tag „$final_virus_destiny = „, und zwar:

D_PASS: eMail wird im ohne Zusatzaktion an den Empfänger zugestellt.

D_DISCARD: eMail wird in den Quarantäne-Bereich verschoben, es erhalten jedoch weder Absender noch Empfänger eine entsprechende Benachrichtigung.

D_REJECT: eMail wird zuerst an den Absender zurückgesandt und dann in den Quarantäne-Bereich verschoben.

```
$warnvirussender = 1;
```

Durch diesen Eintrag wird dem Sender einer infizierten eMail eine nicht-Zustellbarkeitsmeldung zurückgesandt.

Original-AMaViS-Konfigurationsdatei „amavisd.conf“

```
$warnvirusrecip = 1;
```

Dieser Eintrag bewirkt eine Benachrichtigung des vermeintlichen Empfängers einer verseuchten eMail dahingehend, dass er eine infizierte Datei bekommen sollte.

```
### http://www.f-prot.com/
 ['FRISK F-Prot Daemon',
   \&ask_daemon,
   ["GET {}/*?-dumb%20-archive HTTP/1.0\r\n\r\n",
     ['127.0.0.1:10200','127.0.0.1:10201','127.0.0.1:10202',
      '127.0.0.1:10203','127.0.0.1:10204'] ],
   qr/(?i)<summary[^>]*>clean<\/summary>/,
   qr/(?i)<summary[^>]*>infected<\/summary>/,
   qr/(?i)<name>(.+)<\/name>/ ],
```

Dies sind die Eintragungen die vorgenommen werden müssen, um den von uns installieren F-Prot-Virenscanner in AMaViS einzubinden. In der „amavisd.conf“ sind standardmäßig alle Eintragungen für sämtliche von AMaViS unterstützten Virenscanner bereits beinhaltet und müssen gegebenenfalls nur durch entfernen des „#“-Zeichens entkommentiert werden.

2.6 Anbindung an Postfix

Nachdem nun AMaViS mit den gewünschten Möglichkeiten konfiguriert worden ist, muss er dem Postfix-MTA noch bekannt gemacht werden. Hierzu sind folgende Eintragungen in die Postfix-Konfigurationsdateien vorzunehmen:

Postfix-Konfigurationsdatei master.cf:

```
localhost:10025 inet n - n - -
  smtpd -o content_filter=
```

Dieser Eintrag legt einen Transport mit Namen *vscan* an, über den die E-Mail zur Überprüfung an AMaViS weitergeleitet wird.

```
vscan     linux  -       n       n       -       -       pipe
  user=vscan argv=/usr/sbin/amavis ${sender} ${recipient}
```

Damit die Mail nach der Bearbeitung wieder an Postfix zurückgegeben werden kann, startet dieser Eintrag einen zusätzlichen *smtpd*. Dieser Prozess ist an Port 10025 gebunden und läuft mit deaktivierter Filterfunktion (*smtpd -o content_filter=*). So wird verhindert, dass jede E-Mail in einer Endlosschleife immer wieder überprüft wird.

Postfix-Konfigurationsdatei main.cf:

```
content_filter = vscan:
```

Hierdurch wird der in der Datei master.cf bekannt gemachte *vscan*-Transport als zuständiger Filterprozess angelegt.

2.7 Start/Stop/Neustart

Die Befehle selbst und die eventuellen Reaktionen des AMaViS sind ähnlich der des Postfix-MTA und unterliegen daher der gleichen Definition.

Gestartet wird AMaViS mit dem Befehl

```
linux:/ # /etc/init.d/amavis start
```

Soll AMaViS gestoppt werden, so ist folgendes Kommando auszuführen:

```
linux:/ # /etc/init.d/amavis stop
```

Zum Neustart von AMaViS nach Änderungen an der Konfiguration ist folgender Befehl auszuführen:

```
linux:/ # /etc/init.d/amavis restart
```

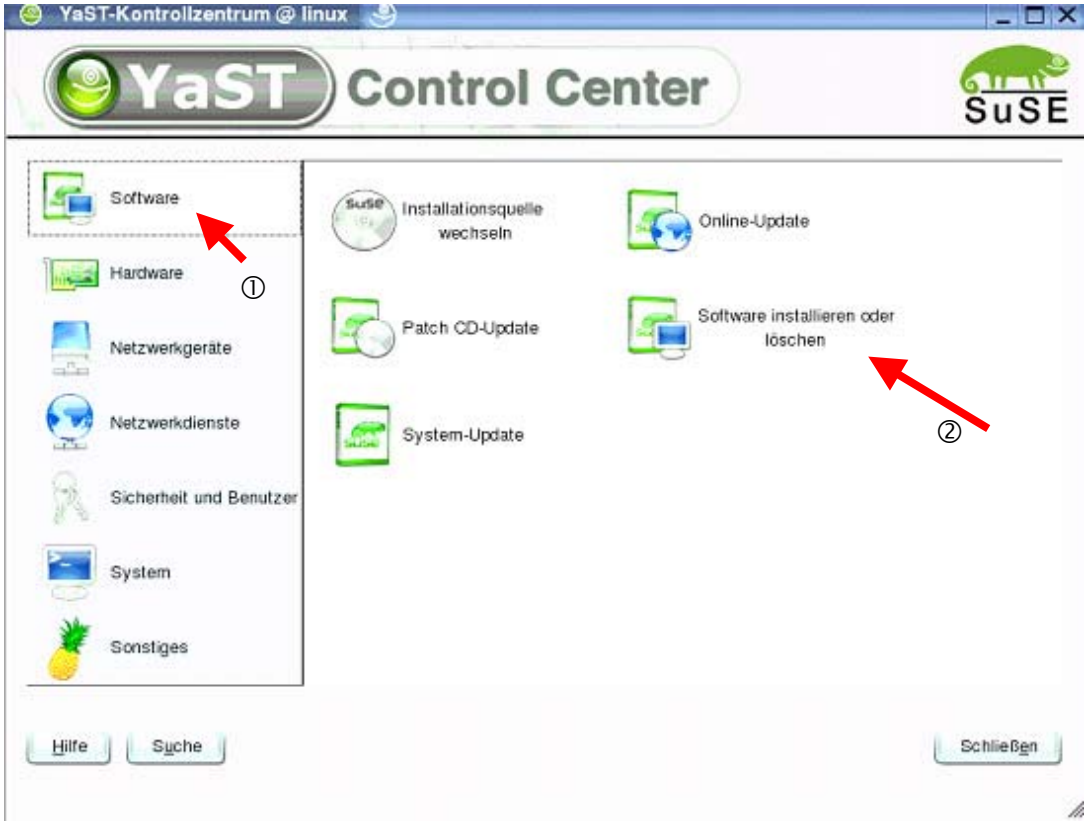
Kapitel 3 qpopper – POP3 Mail Deamon

Wie in Kapitel 1 – Postfix erwähnt ist der Postfix-Server in der Lage, eMails von verschiedenen Stellen aus anzunehmen, weiterzuleiten oder in seine lokal eingerichteten Mailboxen abzulegen. Da nun nicht jeder User seine Post auf dem Server selbst lesen kann muss eine Schnittstelle zwischen dem Server den Client-Computern und ihren Client-eMail-Programmen geschaffen werden. Diese Schnittstelle bietet unter SuSE Linux 9.0 das Programm „qpopper“, ein POP3-Mail-Dämon. qpopper stellt dabei einen POP3-Server dar, welcher das POP3- bzw. das Post Office Protocol in der Version 3 benutzt um einem Client-Computer zu erlauben, elektronische Post von dem POP3-Server, welcher eben durch qpopper gebildet wird, zurückzuholen.

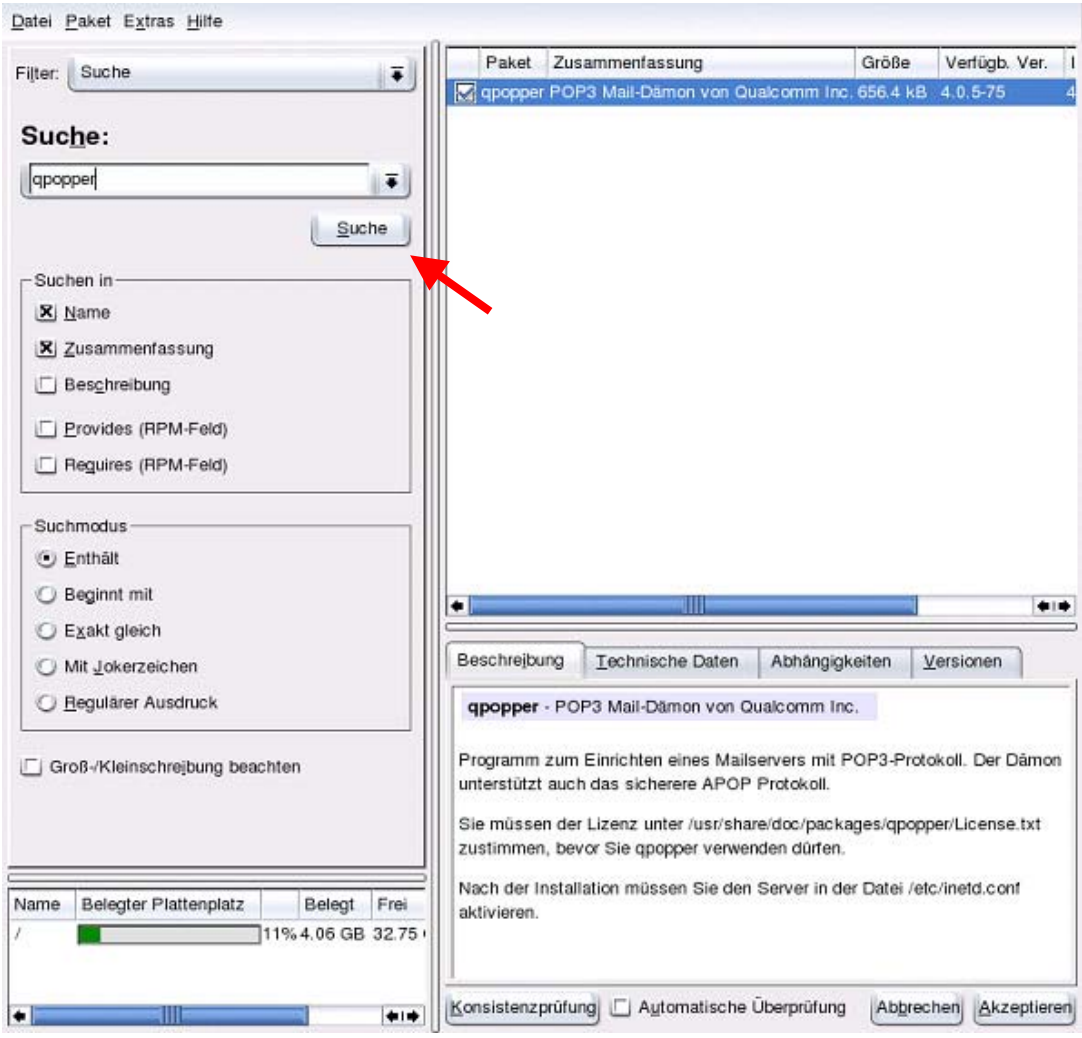
3.1 Installation

Standardmäßig ist qpopper nicht in der Grundinstallation von SuSE Linux 9.0 enthalten, jedoch im umfangreichen Installationspaket beinhaltet. Um qpopper nachzuinstallieren sind folgende Schritte nötig:

Zuerst muss das Konfigurationsprogramm von SuSE Linux, das YaST2 (Yet another Setup Tool 2) gestartet werden.



Dann klickt man unter dem Hauptmenüpunkt „Software“ auf den Untermenüpunkt „Software installieren oder löschen“. Hierauf öffnet sich folgendes Dialogfeld:



Um qpopper nun auf dem schnellsten Weg zu installieren setzt man den Filter links oben auf „Suche“, gibt dann in das daraufhin angezeigte Feld „Suche:“ „qpopper“ ein, klickt auf den Button „Suche“ und erhält kurz darauf das zu installierende Paket im rechten Teil des Auswahlbildschirms angezeigt. Hier muss nun nur noch das Programm zur Installation angeklickt bzw. ein Häkchen in die dafür vorgesehene Checkbox geklickt werden. Zur endgültigen Installation des qpopper muss die Auswahl nun nur noch mit klick auf den Button „Akzeptieren“ gestartet werden. Wurde dies getan, so ist es möglich, dass das Installationsprogramm darauf hinweist, dass noch weitere Programme in Verbindung mit qpopper installiert werden müssen. Dies geschieht ganz automatisch. Zur Bestätigung der Installation aller zugehörigen Programme ist die Aufforderung einfach mit Klick auf den Button „Akzeptieren“ zu beantworten.

3.2 Konfiguration

qpopper ist nach der Installation fast einsatzbereit. Um ihn zu aktivieren ist lediglich die Konfigurationsdatei „qpopper“ im Verzeichnis „/etc/xinetd.d“ zu editieren und der Eintrag „disabled = yes“ auf „disabled = no“ abzuändern.

```
#
# qpopper - pop3 mail daemon
#
service pop3
{
 disable       = no
 socket_type   = stream
 protocol      = tcp
 wait          = no
 user          = root
 server        = /usr/sbin/popper
 server_args   = -s
 flags         = IPv4
}
```

3.3 Start/Stop/Neustart

Da qpopper ein Teil der INET-Dämonen bzw. –Dienste von Linux darstellt, deren Startsequenzen alle in der Datei „xinetd“ im Verzeichnis „/etc/init.d“ zusammengefasst sind, ist er nach der Installation und Konfiguration durch das Kommando „/etc/inet.d/xinetd restart“ zu starten. Wird das System später neu gestartet, so wird er mit den anderen INET-Dämonen automatisch durch das System selbst gestartet.

3.4 Test

Um schnellstmöglich testen zu können ob qpopper einwandfrei arbeitet, kann hier wieder, wie beim Test des Postfix-Servers auf das Programm „telnet“ zurückgegriffen werden. Hierbei wird telnet jedoch nicht wie bei Postfix mit dem Port 25, dem SMTP-Port, sondern mit Port 110 gestartet, welcher den POP3-Port eines jeden Mailsystems bildet. Am Beispiel unseres Servers sieht der Aufruf von telnet wie folgt aus:

```
linux:/ # telnet 172.16.111.107 110
Trying 172.16.111.107...
Connected to 172.16.111.107.
Escape character is '^]'.
+OK ready  <2215.1080320906@linux.bbsnw.de>
```

Antwortet der Server wie oben mit “+OK ready …”, so ist der Server für den Versand an bzw. das Abholen von eMails an die Client-Computer bereit.

Kapitel 4 Abwesenheitsnotizen

Oft ist es notwendig Abwesenheitsnotizen, sogenannte „vacation-Mails“ einzurichten, um eMail-Absender darüber zu informieren, dass der angeschriebene eMail-Empfänger aus irgendwelchen Gründen die eMail zur Zeit nicht beantworten kann, etc.

Um solche vacation-Mails einzurichten bietet SuSE-Linux eine sehr simple Methode, das Programm „vacation“.

4.1 Voreinstellungen

Bevor das Programm „vacation“ gestartet werden kann ist mit einem beliebigen Text-Editor eine Datei namens „.vacation.msg“ im jeweiligen Home-Verzeichnis eines Users anzulegen, für den eine Abwesenheitsnotiz bei eingehender eMail versandt werden soll.

.vacation.msg

Diese Datei muss in der ersten Zeile mit dem Wort „Subject: „ und dem Betreff beginnen, den das Mailsystem dem vermeintlichen Empfänger zurücksenden soll. Hierauf ist eine Leerzeile einzufügen, gefolgt von der Nachricht, welche der Empfänger der Abwesenheitsnotiz erhalten soll.

Hier ein Beispiel einer möglichen .vacation.msg:

```
Subject: away from my mail

I will not be reading my mail for a while.
Your mail will be read when I'm back.
```

.forward

Nachdem die Datei .vacation.msg erstellt wurde muss mit einem Editor noch eine zweite Datei mit namen „.forward“ im Home-Verzeichnis des jeweiligen Users angelegt werden, welche folgenden Inhalt aufweist:

```
\username, "|/usr/bin/vacation username"
```

username ist in diesem Fall der Benutzer- bzw. Loginname des Users für den die „.vacation.msg“-Datei erstellt wurde.

Allmann , Eric P.: Vacation User Manual

4.2 Start des Abwesenheitsnotizmechanismus

Wurden eine „.vacation.msg“- und eine „.forward“-Datei im jeweiligen Home-Verzeichnis eines Users erstellt, so ist das Kommando

```
vacation -I
```

auf der Konsole aus dem jeweiligen Home-Verzeichnis des entsprechenden Users, für den der Abwesenheitsnotiz-Automatismus geschaltet werden soll, einzugeben.

Hierdurch wird die Datei „.vacation.db“, die Abwesenheitsnotiz-Datenbank initialisiert und die Abwesenheitsnotizfunktion gestartet.

4.3 Stop der Abwesenheitsnotizmechanismus

Um die Abwesenheitsnotizfunktion für einen jeweiligen User zu stoppen, ist lediglich die Datei „.forward“ im entsprechenden Home-Verzeichnis des Users zu entfernen oder umzubennenen.

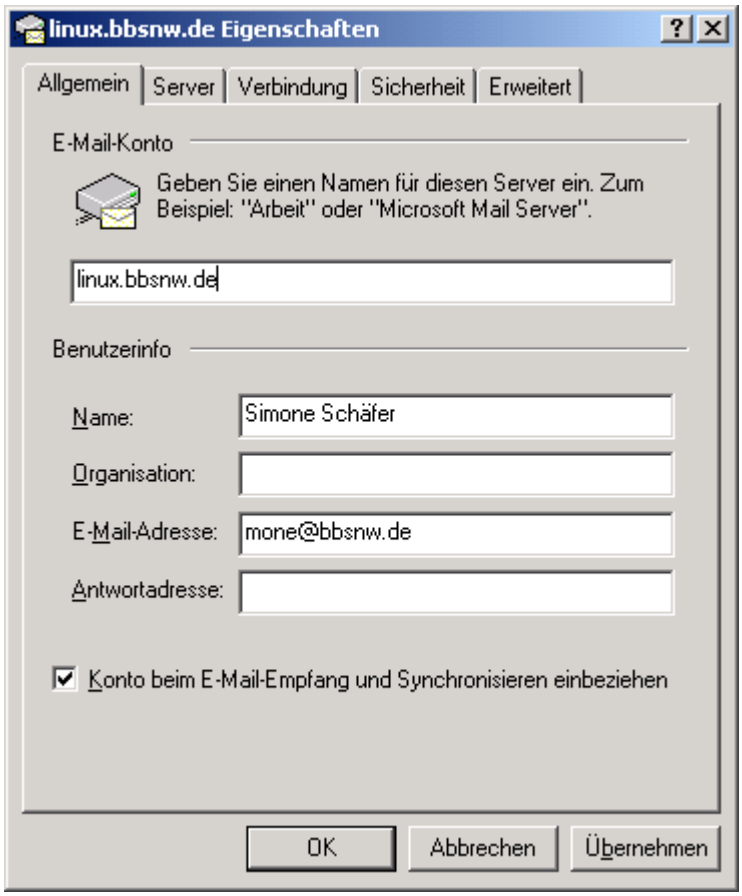
Kapitel 5 eMail-Clientprogramme

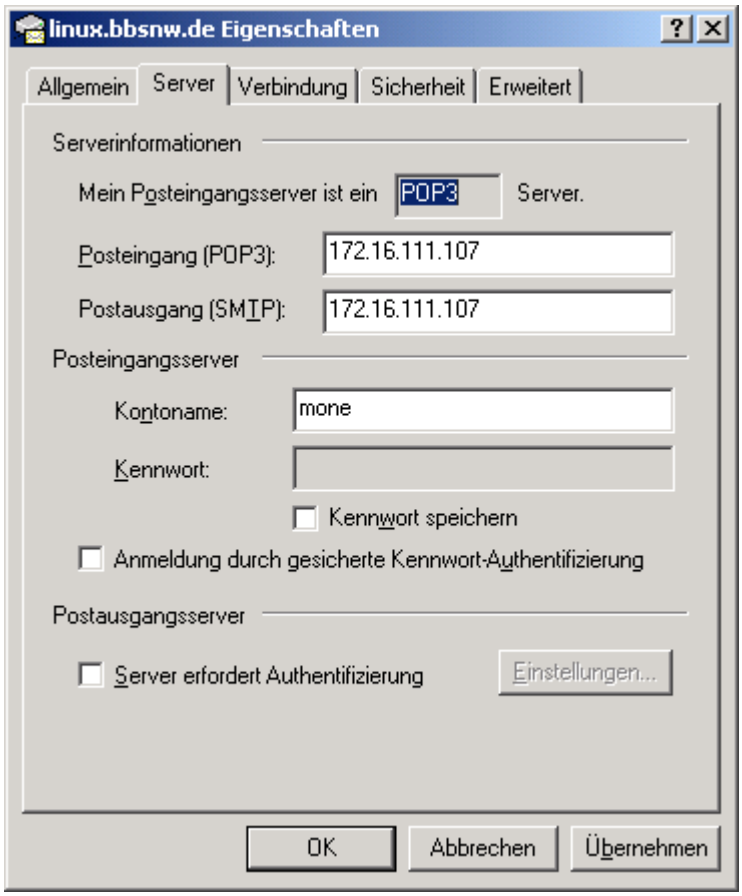
Da der Postfix-Server gekoppelt mit dem Programm qpopper nun in der Lage ist, eMails via POP3-Protocol an Client-Rechner im Netzwerk zu senden und über das SMTP-Protocol zu empfangen bietet es sich an, auf den Client-Rechnern eMail-Client-Programme zu installieren, damit die dahinter sitzenden User ihre eMails nicht am Server selbst anschauen, bearbeiten und senden müssen. Voraussetzung hierfür ist, dass auf den Client-Computern das TCP-IP-Protokoll installiert ist.

Um ein eMail-Client-Programm einzurichten bedarf es nur weniger Handgriffe. Es ist hier vorwegzunehmen, dass sowohl der Postfix-Server als auch der qpopper plattformunabhänig arbeiten, das heißt sie versenden und empfangen eMails von und an sämtliche eMail-Client-Programme, auf verschiedenen Betriebssystemen. Die Einstellungen, welche an den verschiedenen eMail-Programmen vorzunehmen sind, sind immer die gleichen:

POP3 bzw. POP-Server: IP-Adresse des Servers auf dem die Post abzuholen ist.
SMTP bzw. SMTP-Server: IP-Adresse des Servers auf dem Post versand werden kann.
Konto- bzw. Benutzername: der lokale Benutzername, der dem User auf dem Postfix-Server für den User angelegt wurde
Name: Der reelle Name des Mail-Users (wird in gesendeten eMails angezeigt).
eMail-Adresse: virtuelle eMail-Adresse des Users (dient als Rücksendeadresse für gesendete eMails und wird auch in gesendeten eMails angezeigt).

Zur Demonstration, dass ein Postfix- und qpopper-Server unter Linux auch z.B. mit einem Microsoft-Windows-System und einem darauf installieren eMail-Client von Microsoft, dem Programm „Outlook Express“ zusammenarbeitet sind, hier beispielhaft die Einstellungen auf eben dieser Konfiguration zu sehen:


linux.bbsnw.de Eigenschaften
Allgemein
Server
Verbindung
Sicherheit
Erweitert
E-Mail-Konto
Geben Sie einen Namen für diesen Server ein. Zum Beispiel: "Arbeit" oder "Microsoft Mail Server".
linux.bbsnw.de
Benutzerinfo
Name:
Simone Schäfer
Organisation:
E-Mail-Adresse:
mone@bbsnw.de
Antwortadresse:
Konto beim E-Mail-Empfang und Synchronisieren einbeziehen
OK
Abbrechen
Übernehmen


linux.bbsnw.de Eigenschaften
Allgemein
Server
Verbindung
Sicherheit
Erweitert
Serverinformationen
Mein Posteingangsserver ist ein
POP3
Server.
Posteingang (POP3):
172.16.111.107
Postausgang (SMTP):
172.16.111.107
Posteingangsserver
Kontoname:
mone
Kennwort:
Kennwort speichern
Anmeldung durch gesicherte Kennwort-Authentifizierung
Postausgangsserver
Server erfordert Authentifizierung
Einstellungen...
OK
Abbrechen
Übernehmen

Quellen

Markus Ungermann: Email-Server: Postfix.
<http://www.tuxhausen.de/software_postfix.html>.
In: Homepage von Tuxhausen. <http://www.tuxhausen.de>.
Download vom 2004-02-03

Wiesner, Michael: Antiviren-Mailrelay mit Amavis.
<http://www.linux-magazin.de/Artikel/ausgabe/2001/06/Amavis/amavis.html>.
In: Homepage von Linux-Magazin. <http://www.linux-magazin.de>.
Download vom 2004-03-22

Allmann , Eric P.: Vacation User Manual.
Einzusehen via „man vacation“ auf Konsole unter SuSE Linux 9.0

Original-Postfix-Konfigurationsdatei „main.cf“
Einzusehen in Verzeichnis „/etc/postfix/“ unter SuSE Linux 9.0

Original-Postfix-Konfigurationsdatei „master.cf“
Einzusehen in Verzeichnis „/etc/postfix/“ unter SuSE Linux 9.0

Original-AMaViS-Konfigurationsdatei „amavisd.conf“
Einzusehen in Verzeichnis „/etc/“ unter SuSE Linux 9.0 nach Installation des RPM-Paketes von AmaViS.

Konfigurationsdatei “main.cf”

(abgelegt unter “/etc/postfix/main.cf”)

```
#
# ----------------------------------------------------------------------
# NOTE: Many parameters have already been added to the end of this file
#       by SuSEconfig.postfix. So take care that you don't uncomment
#       and set a parameter without checking whether it has been added
#       to the end of this file.
# ----------------------------------------------------------------------
#
# Global Postfix configuration file. This file lists only a subset
# of all 300+ parameters. See the sample-xxx.cf files for a full list.
#
# The general format is lines with parameter = value pairs. Lines
# that begin with whitespace continue the previous line. A value can
# contain references to other $names or ${name}s.
#
# NOTE - CHANGE NO MORE THAN 2-3 PARAMETERS AT A TIME, AND TEST IF
# POSTFIX STILL WORKS AFTER EVERY CHANGE.

# SOFT BOUNCE
#
# The soft_bounce parameter provides a limited safety net for
# testing.  When soft_bounce is enabled, mail will remain queued that
# would otherwise bounce. This parameter disables locally-generated
# bounces, and prevents the SMTP server from rejecting mail permanently
# (by changing 5xx replies into 4xx replies). However, soft_bounce
# is no cure for address rewriting mistakes or mail routing mistakes.
#
#soft_bounce = no

# LOCAL PATHNAME INFORMATION
#
# The queue_directory specifies the location of the Postfix queue.
# This is also the root directory of Postfix daemons that run chrooted.
# See the files in examples/chroot-setup for setting up Postfix chroot
# environments on different UNIX systems.
#

# The command_directory parameter specifies the location of all
# postXXX commands.
#
command_directory = /usr/sbin

# The daemon_directory parameter specifies the location of all Postfix
# daemon programs (i.e. programs listed in the master.cf file). This
# directory must be owned by root.
#
daemon_directory = /usr/lib/postfix

# QUEUE AND PROCESS OWNERSHIP
#
# The mail_owner parameter specifies the owner of the Postfix queue
# and of most Postfix daemon processes.  Specify the name of a user
# account THAT DOES NOT SHARE ITS USER OR GROUP ID WITH OTHER ACCOUNTS
```

```
# AND THAT OWNS NO OTHER FILES OR PROCESSES ON THE SYSTEM.  In
# particular, don't specify nobody or daemon. PLEASE USE A DEDICATED
# USER.
#

# The default_privs parameter specifies the default rights used by
# the local delivery agent for delivery to external file or command.
# These rights are used in the absence of a recipient user context.
# DO NOT SPECIFY A PRIVILEGED USER OR THE POSTFIX OWNER.
#
#default_privs = nobody

# INTERNET HOST AND DOMAIN NAMES
#
# The myhostname parameter specifies the internet hostname of this
# mail system. The default is to use the fully-qualified domain name
# from gethostname(). $myhostname is used as a default value for many
# other configuration parameters.
#
#myhostname = host.domain.tld
#myhostname = virtual.domain.tld

# The mydomain parameter specifies the local internet domain name.
# The default is to use $myhostname minus the first component.
# $mydomain is used as a default value for many other configuration
# parameters.
#
#mydomain = domain.tld

# SENDING MAIL
#
# The myorigin parameter specifies the domain that locally-posted
# mail appears to come from. The default is to append $myhostname,
# which is fine for small sites.  If you run a domain with multiple
# machines, you should (1) change this to $mydomain and (2) set up
# a domain-wide alias database that aliases each user to
# user@that.users.mailhost.
#
# For the sake of consistency between sender and recipient addresses,
# myorigin also specifies the default domain name that is appended
# to recipient addresses that have no @domain part.
#
#myorigin = $myhostname
#myorigin = $mydomain

# RECEIVING MAIL

# The inet_interfaces parameter specifies the network interface
# addresses that this mail system receives mail on.  By default,
# the software claims all active interfaces on the machine. The
# parameter also controls delivery of mail to user@[ip.address].
#
# See also the proxy_interfaces parameter, for network addresses that
# are forwarded to us via a proxy or network address translator.
#
# Note: you need to stop/start Postfix when this parameter changes.
#
#inet_interfaces = all
```

```
#inet_interfaces = $myhostname
#inet_interfaces = $myhostname, localhost

# The proxy_interfaces parameter specifies the network interface
# addresses that this mail system receives mail on by way of a
# proxy or network address translation unit. This setting extends
# the address list specified with the inet_interfaces parameter.
#
# You must specify your proxy/NAT addresses when your system is a
# backup MX host for other domains, otherwise mail delivery loops
# will happen when the primary MX host is down.
#
#proxy_interfaces =
#proxy_interfaces = 1.2.3.4

# The mydestination parameter specifies the list of domains that this
# machine considers itself the final destination for.
#
# These domains are routed to the delivery agent specified with the
# local_transport parameter setting. By default, that is the UNIX
# compatible delivery agent that lookups all recipients in /etc/passwd
# and /etc/aliases or their equivalent.
#
# The default is $myhostname + localhost.$mydomain.  On a mail domain
# gateway, you should also include $mydomain.
#
# Do not specify the names of virtual domains - those domains are
# specified elsewhere (see sample-virtual.cf).
#
# Do not specify the names of domains that this machine is backup MX
# host for. Specify those names via the relay_domains settings for
# the SMTP server, or use permit_mx_backup if you are lazy (see
# sample-smtpd.cf).
#
# The local machine is always the final destination for mail addressed
# to user@[the.net.work.address] of an interface that the mail system
# receives mail on (see the inet_interfaces parameter).
#
# Specify a list of host or domain names, /file/name or type:table
# patterns, separated by commas and/or whitespace. A /file/name
# pattern is replaced by its contents; a type:table is matched when
# a name matches a lookup key (the right-hand side is ignored).
# Continue long lines by starting the next line with whitespace.
#
# See also below, section "REJECTING MAIL FOR UNKNOWN LOCAL USERS".
#
#mydestination = $myhostname, localhost.$mydomain
#mydestination = $myhostname, localhost.$mydomain $mydomain
#mydestination = $myhostname, localhost.$mydomain, $mydomain,
#     mail.$mydomain, www.$mydomain, ftp.$mydomain

# REJECTING MAIL FOR UNKNOWN LOCAL USERS
#
# The local_recipient_maps parameter specifies optional lookup tables
# with all names or addresses of users that are local with respect
# to $mydestination and $inet_interfaces.
#
# If this parameter is defined, then the SMTP server will reject
```

```
# mail for unknown local users. This parameter is defined by default.
#
# To turn off local recipient checking in the SMTP server, specify
# local_recipient_maps = (i.e. empty).
#
# The default setting assumes that you use the default Postfix local
# delivery agent for local delivery. You need to update the
# local_recipient_maps setting if:
#
# - You define $mydestination domain recipients in files other than
#   /etc/passwd, /etc/aliases, or the $virtual_alias_maps files.
#   For example, you define $mydestination domain recipients in
#   the $virtual_mailbox_maps files.
#
# - You redefine the local delivery agent in master.cf.
#
# - You redefine the "local_transport" setting in main.cf.
#
# - You use the "luser_relay", "mailbox_transport", or "fallback_transport"
#   feature of the Postfix local delivery agent (see sample-local.cf).
#
# Details are described in the LOCAL_RECIPIENT_README file.
#
# Beware: if the Postfix SMTP server runs chrooted, you probably have
# to access the passwd file via the proxymap service, in order to
# overcome chroot restrictions. The alternative, having a copy of
# the system passwd file in the chroot jail is just not practical.
#
# The right-hand side of the lookup tables is conveniently ignored.
# In the left-hand side, specify a bare username, an @domain.tld
# wild-card, or specify a user@domain.tld address.
#
#local_recipient_maps = unix:passwd.byname $alias_maps
#local_recipient_maps = proxy:unix:passwd.byname $alias_maps
#local_recipient_maps =

# The unknown_local_recipient_reject_code specifies the SMTP server
# response code when a recipient domain matches $mydestination or
# $inet_interfaces, while $local_recipient_maps is non-empty and the
# recipient address or address local-part is not found.
#
# The default setting is 550 (reject mail) but it is safer to start
# with 450 (try again later) until you are certain that your
# local_recipient_maps settings are OK.
#
#unknown_local_recipient_reject_code = 550
unknown_local_recipient_reject_code = 450

# TRUST AND RELAY CONTROL

# The mynetworks parameter specifies the list of "trusted" SMTP
# clients that have more privileges than "strangers".
#
# In particular, "trusted" SMTP clients are allowed to relay mail
# through Postfix.  See the smtpd_recipient_restrictions parameter
# in file sample-smtpd.cf.
#
# You can specify the list of "trusted" network addresses by hand
```

```
# or you can let Postfix do it for you (which is the default).
#
# By default (mynetworks_style = subnet), Postfix "trusts" SMTP
# clients in the same IP subnetworks as the local machine.
# On Linux, this does works correctly only with interfaces specified
# with the "ifconfig" command.
#
# Specify "mynetworks_style = class" when Postfix should "trust" SMTP
# clients in the same IP class A/B/C networks as the local machine.
# Don't do this with a dialup site - it would cause Postfix to "trust"
# your entire provider's network.  Instead, specify an explicit
# mynetworks list by hand, as described below.
#
# Specify "mynetworks_style = host" when Postfix should "trust"
# only the local machine.
#
#mynetworks_style = class
#mynetworks_style = subnet
#mynetworks_style = host

# Alternatively, you can specify the mynetworks list by hand, in
# which case Postfix ignores the mynetworks_style setting.
#
# Specify an explicit list of network/netmask patterns, where the
# mask specifies the number of bits in the network part of a host
# address.
#
# You can also specify the absolute pathname of a pattern file instead
# of listing the patterns here. Specify type:table for table-based lookups
# (the value on the table right-hand side is not used).
#
#mynetworks = 168.100.189.0/28, 127.0.0.0/8
#mynetworks = $config_directory/mynetworks
#mynetworks = hash:/etc/postfix/network_table
mynetworks  = 172.16.0.0/16, 127.0.0.0/8, 127.0.0.1/8

# The relay_domains parameter restricts what destinations this system will
# relay mail to.  See the smtpd_recipient_restrictions restriction in the
# file sample-smtpd.cf for detailed information.
#
# By default, Postfix relays mail
# - from "trusted" clients (IP address matches $mynetworks) to any
destination,
# - from "untrusted" clients to destinations that match $relay_domains or
#   subdomains thereof, except addresses with sender-specified routing.
# The default relay_domains value is $mydestination.
#
# In addition to the above, the Postfix SMTP server by default accepts mail
# that Postfix is final destination for:
# - destinations that match $inet_interfaces,
# - destinations that match $mydestination
# - destinations that match $virtual_alias_domains,
# - destinations that match $virtual_mailbox_domains.
# These destinations do not need to be listed in $relay_domains.
#
# Specify a list of hosts or domains, /file/name patterns or type:name
# lookup tables, separated by commas and/or whitespace.  Continue
# long lines by starting the next line with whitespace. A file name
```

```
# is replaced by its contents; a type:name table is matched when a
# (parent) domain appears as lookup key.
#
# NOTE: Postfix will not automatically forward mail for domains that
# list this system as their primary or backup MX host. See the
# permit_mx_backup restriction in the file sample-smtpd.cf.
#
#relay_domains = $mydestination

# INTERNET OR INTRANET

# The relayhost parameter specifies the default host to send mail to
# when no entry is matched in the optional transport(5) table. When
# no relayhost is given, mail is routed directly to the destination.
#
# On an intranet, specify the organizational domain name. If your
# internal DNS uses no MX records, specify the name of the intranet
# gateway host instead.
#
# In the case of SMTP, specify a domain, host, host:port, [host]:port,
# [address] or [address]:port; the form [host] turns off MX lookups.
#
# If you're connected via UUCP, see also the default_transport parameter.
#
#relayhost = $mydomain
#relayhost = gateway.my.domain
#relayhost = uucphost
#relayhost = [an.ip.add.ress]

# REJECTING UNKNOWN RELAY USERS
#
# The relay_recipient_maps parameter specifies optional lookup tables
# with all addresses in the domains that match $relay_domains.
#
# If this parameter is defined, then the SMTP server will reject
# mail for unknown relay users. This feature is off by default.
#
# The right-hand side of the lookup tables is conveniently ignored.
# In the left-hand side, specify an @domain.tld wild-card, or specify
# a user@domain.tld address.
#
#relay_recipient_maps = hash:/etc/postfix/relay_recipients

# INPUT RATE CONTROL
#
# The in_flow_delay configuration parameter implements mail input
# flow control. This feature is turned on by default, although it
# still needs further development (it's disabled on SCO UNIX due
# to an SCO bug).
#
# A Postfix process will pause for $in_flow_delay seconds before
# accepting a new message, when the message arrival rate exceeds the
# message delivery rate. With the default 100 SMTP server process
# limit, this limits the mail inflow to 100 messages a second more
# than the number of messages delivered per second.
#
# Specify 0 to disable the feature. Valid delays are 0..10.
#
```

```
#in_flow_delay = 1s

# ADDRESS REWRITING
#
# Insert text from sample-rewrite.cf if you need to do address
# masquerading.
#
# Insert text from sample-canonical.cf if you need to do address
# rewriting, or if you need username->Firstname.Lastname mapping.

# ADDRESS REDIRECTION (VIRTUAL DOMAIN)
#
# Insert text from sample-virtual.cf if you need virtual domain support.

# "USER HAS MOVED" BOUNCE MESSAGES
#
# Insert text from sample-relocated.cf if you need "user has moved"
# style bounce messages. Alternatively, you can bounce recipients
# with an SMTP server access table. See sample-smtpd.cf.

# TRANSPORT MAP
#
# Insert text from sample-transport.cf if you need explicit routing.

# ALIAS DATABASE
#
# The alias_maps parameter specifies the list of alias databases used
# by the local delivery agent. The default list is system dependent.
#
# On systems with NIS, the default is to search the local alias
# database, then the NIS alias database. See aliases(5) for syntax
# details.
#
# If you change the alias database, run "postalias /etc/aliases" (or
# wherever your system stores the mail alias file), or simply run
# "newaliases" to build the necessary DBM or DB file.
#
# It will take a minute or so before changes become visible.  Use
# "postfix reload" to eliminate the delay.
#
#alias_maps = dbm:/etc/aliases
#alias_maps = hash:/etc/aliases
#alias_maps = hash:/etc/aliases, nis:mail.aliases
#alias_maps = netinfo:/aliases

# The alias_database parameter specifies the alias database(s) that
# are built with "newaliases" or "sendmail -bi".  This is a separate
# configuration parameter, because alias_maps (see above) may specify
# tables that are not necessarily all under control by Postfix.
#
#alias_database = dbm:/etc/aliases
#alias_database = dbm:/etc/mail/aliases
#alias_database = hash:/etc/aliases
#alias_database = hash:/etc/aliases, hash:/opt/majordomo/aliases

# ADDRESS EXTENSIONS (e.g., user+foo)
#
# The recipient_delimiter parameter specifies the separator between
```

```
# user names and address extensions (user+foo). See canonical(5),
# local(8), relocated(5) and virtual(5) for the effects this has on
# aliases, canonical, virtual, relocated and .forward file lookups.
# Basically, the software tries user+foo and .forward+foo before
# trying user and .forward.
#
#recipient_delimiter = +

# DELIVERY TO MAILBOX
#
# The home_mailbox parameter specifies the optional pathname of a
# mailbox file relative to a user's home directory. The default
# mailbox file is /var/spool/mail/user or /var/mail/user.  Specify
# "Maildir/" for qmail-style delivery (the / is required).
#
#home_mailbox = Mailbox
#home_mailbox = Maildir/

# The mail_spool_directory parameter specifies the directory where
# UNIX-style mailboxes are kept. The default setting depends on the
# system type.
#
#mail_spool_directory = /var/mail
#mail_spool_directory = /var/spool/mail

# The mailbox_command parameter specifies the optional external
# command to use instead of mailbox delivery. The command is run as
# the recipient with proper HOME, SHELL and LOGNAME environment settings.
# Exception:  delivery for root is done as $default_user.
#
# Other environment variables of interest: USER (recipient username),
# EXTENSION (address extension), DOMAIN (domain part of address),
# and LOCAL (the address localpart).
#
# Unlike other Postfix configuration parameters, the mailbox_command
# parameter is not subjected to $parameter substitutions. This is to
# make it easier to specify shell syntax (see example below).
#
# Avoid shell meta characters because they will force Postfix to run
# an expensive shell process. Procmail alone is expensive enough.
#
# IF YOU USE THIS TO DELIVER MAIL SYSTEM-WIDE, YOU MUST SET UP AN
# ALIAS THAT FORWARDS MAIL FOR ROOT TO A REAL USER.
#
#mailbox_command = /some/where/procmail
#mailbox_command = /some/where/procmail -a "$EXTENSION"

# The mailbox_transport specifies the optional transport in master.cf
# to use after processing aliases and .forward files. This parameter
# has precedence over the mailbox_command, fallback_transport and
# luser_relay parameters.
#
# Specify a string of the form transport:nexthop, where transport is
# the name of a mail delivery transport defined in master.cf.  The
# :nexthop part is optional. For more details see the sample transport
# configuration file.
#
# NOTE: if you use this feature for accounts not in the UNIX password
```

```
# file, then you must update the "local_recipient_maps" setting in
# the main.cf file, otherwise the SMTP server will reject mail for
# non-UNIX accounts with "User unknown in local recipient table".
#
#mailbox_transport = lmtp:unix:/file/name
#mailbox_transport = cyrus

# The fallback_transport specifies the optional transport in master.cf
# to use for recipients that are not found in the UNIX passwd database.
# This parameter has precedence over the luser_relay parameter.
#
# Specify a string of the form transport:nexthop, where transport is
# the name of a mail delivery transport defined in master.cf.  The
# :nexthop part is optional. For more details see the sample transport
# configuration file.
#
# NOTE: if you use this feature for accounts not in the UNIX password
# file, then you must update the "local_recipient_maps" setting in
# the main.cf file, otherwise the SMTP server will reject mail for
# non-UNIX accounts with "User unknown in local recipient table".
#
#fallback_transport = lmtp:unix:/file/name
#fallback_transport = cyrus
#fallback_transport =

# The luser_relay parameter specifies an optional destination address
# for unknown recipients.  By default, mail for unknown@$mydestination
# and unknown@[$inet_interfaces] is returned as undeliverable.
#
# The following expansions are done on luser_relay: $user (recipient
# username), $shell (recipient shell), $home (recipient home directory),
# $recipient (full recipient address), $extension (recipient address
# extension), $domain (recipient domain), $local (entire recipient
# localpart), $recipient_delimiter. Specify ${name?value} or
# ${name:value} to expand value only when $name does (does not) exist.
#
# luser_relay works only for the default Postfix local delivery agent.
#
# NOTE: if you use this feature for accounts not in the UNIX password
# file, then you must specify "local_recipient_maps =" (i.e. empty) in
# the main.cf file, otherwise the SMTP server will reject mail for
# non-UNIX accounts with "User unknown in local recipient table".
#
#luser_relay = $user@other.host
#luser_relay = $local@other.host
#luser_relay = admin+$local

# JUNK MAIL CONTROLS
#
# The controls listed here are only a very small subset. See the file
# sample-smtpd.cf for an elaborate list of anti-UCE controls.

# The header_checks parameter specifies an optional table with patterns
# that each logical message header is matched against, including
# headers that span multiple physical lines.
#
# By default, these patterns also apply to MIME headers and to the
# headers of attached messages. With older Postfix versions, MIME and
```

```
# attached message headers were treated as body text.
#
# For details, see the sample-filter.cf file.
#
#header_checks = regexp:/etc/postfix/header_checks

# FAST ETRN SERVICE
#
# Postfix maintains per-destination logfiles with information about
# deferred mail, so that mail can be flushed quickly with the SMTP
# "ETRN domain.tld" command, or by executing "sendmail -qRdomain.tld".
#
# By default, Postfix maintains deferred mail logfile information
# only for destinations that Postfix is willing to relay to (as
# specified in the relay_domains parameter). For other destinations,
# Postfix attempts to deliver ALL queued mail after receiving the
# SMTP "ETRN domain.tld" command, or after execution of "sendmail
# -qRdomain.tld". This can be slow when a lot of mail is queued.
#
# The fast_flush_domains parameter controls what destinations are
# eligible for this "fast ETRN/sendmail -qR" service.
#
#fast_flush_domains = $relay_domains
#fast_flush_domains =

# SHOW SOFTWARE VERSION OR NOT
#
# The smtpd_banner parameter specifies the text that follows the 220
# code in the SMTP server's greeting banner. Some people like to see
# the mail version advertised. By default, Postfix shows no version.
#
# You MUST specify $myhostname at the start of the text. That is an
# RFC requirement. Postfix itself does not care.
#
#smtpd_banner = $myhostname ESMTP $mail_name
#smtpd_banner = $myhostname ESMTP $mail_name ($mail_version)

# PARALLEL DELIVERY TO THE SAME DESTINATION
#
# How many parallel deliveries to the same user or domain? With local
# delivery, it does not make sense to do massively parallel delivery
# to the same user, because mailbox updates must happen sequentially,
# and expensive pipelines in .forward files can cause disasters when
# too many are run at the same time. With SMTP deliveries, 10
# simultaneous connections to the same domain could be sufficient to
# raise eyebrows.
#
# Each message delivery transport has its XXX_destination_concurrency_limit
# parameter.  The default is $default_destination_concurrency_limit for
# most delivery transports. For the local delivery agent the default is 2.

#local_destination_concurrency_limit = 2
#default_destination_concurrency_limit = 20

# DEBUGGING CONTROL
#
# The debug_peer_level parameter specifies the increment in verbose
# logging level when an SMTP client or server host name or address
```

```
# matches a pattern in the debug_peer_list parameter.
#
debug_peer_level = 2

# The debug_peer_list parameter specifies an optional list of domain
# or network patterns, /file/name patterns or type:name tables. When
# an SMTP client or server host name or address matches a pattern,
# increase the verbose logging level by the amount specified in the
# debug_peer_level parameter.
#
#debug_peer_list = 127.0.0.1
#debug_peer_list = some.domain

# The debugger_command specifies the external command that is executed
# when a Postfix daemon program is run with the -D option.
#
# Use "command .. & sleep 5" so that the debugger can attach before
# the process marches on. If you use an X-based debugger, be sure to
# set up your XAUTHORITY environment variable before starting Postfix.
#
debugger_command =
       PATH=/bin:/usr/bin:/usr/local/bin:/usr/X11R6/bin
       xxgdb $daemon_directory/$process_name $process_id & sleep 5

# If you don't have X installed on the Postfix machine, try:
# debugger_command =
#     PATH=/bin:/usr/bin:/usr/local/bin; export PATH; (echo cont;
#     echo where) | gdb $daemon_directory/$process_name $process_id 2>&1
#     >$config_directory/$process_name.$process_id.log & sleep 5

# INSTALL-TIME CONFIGURATION INFORMATION
#
# The following parameters are used when installing a new Postfix version.
#
# sendmail_path: The full pathname of the Postfix sendmail command.
# This is the Sendmail-compatible mail posting interface.
#
sendmail_path = /usr/sbin/sendmail

# newaliases_path: The full pathname of the Postfix newaliases command.
# This is the Sendmail-compatible command to build alias databases.
#
newaliases_path = /usr/bin/newaliases

# mailq_path: The full pathname of the Postfix mailq command.  This
# is the Sendmail-compatible mail queue listing command.
#
mailq_path = /usr/bin/mailq

# setgid_group: The group for mail submission and queue management
# commands.  This must be a group name with a numerical group ID that
# is not shared with other accounts, not even with the Postfix account.
#
setgid_group = maildrop

# manpage_directory: The location of the Postfix on-line manual pages.
#
manpage_directory = /usr/share/man
```

```
# sample_directory: The location of the Postfix sample configuration files.
#
sample_directory = /usr/share/doc/packages/postfix/samples

# readme_directory: The location of the Postfix README files.
#
readme_directory = /usr/share/doc/packages/postfix/README_FILES
mail_spool_directory = /var/mail
canonical_maps = hash:/etc/postfix/canonical
virtual_maps = hash:/etc/postfix/virtual
relocated_maps = hash:/etc/postfix/relocated
transport_maps = hash:/etc/postfix/transport
sender_canonical_maps = hash:/etc/postfix/sender_canonical
masquerade_exceptions = root
masquerade_classes = envelope_sender, header_sender, header_recipient
myhostname = linux.bbsnw.de
program_directory = /usr/lib/postfix
masquerade_domains = bbsnw.de
defer_transports = smtp
disable_dns_lookups = yes
relayhost = [mail.bbsnw.de]
mailbox_command =
mailbox_transport =
smtpd_sender_restrictions = hash:/etc/postfix/access
smtpd_client_restrictions =
smtpd_helo_required = no
smtpd_helo_restrictions =
strict_rfc821_envelopes = no
smtpd_recipient_restrictions = permit_mynetworks,reject_unauth_destination
smtp_sasl_auth_enable = no
smtpd_sasl_auth_enable = no
smtpd_use_tls = no
smtp_use_tls = no
alias_maps = hash:/etc/aliases
mailbox_size_limit = 0
message_size_limit = 10240000
myorigin = bbsnw.de
mydestination = bbsnw.de

content_filter = vscan
inet_interfaces = all
#soft_bounce = yes
```

Konfigurationsdatei „master.cf“

(abgelegt unter “/etc/postfix/master.cf”)

```
#
# Postfix master process configuration file.  Each logical line
# describes how a Postfix daemon program should be run.
#
# A logical line starts with non-whitespace, non-comment text.
# Empty lines and whitespace-only lines are ignored, as are comment
# lines whose first non-whitespace character is a `#'.
# A line that starts with whitespace continues a logical line.
#
# The fields that make up each line are described below. A "-" field
# value requests that a default value be used for that field.
#
# Service: any name that is valid for the specified transport type
# (the next field).  With INET transports, a service is specified as
# host:port.  The host part (and colon) may be omitted. Either host
# or port may be given in symbolic form or in numeric form. Examples
# for the SMTP server:  localhost:smtp receives mail via the loopback
# interface only; 10025 receives mail on port 10025.
#
# Transport type: "inet" for Internet sockets, "unix" for UNIX-domain
# sockets, "fifo" for named pipes.
#
# Private: whether or not access is restricted to the mail system.
# Default is private service.  Internet (inet) sockets can't be private.
#
# Unprivileged: whether the service runs with root privileges or as
# the owner of the Postfix system (the owner name is controlled by the
# mail_owner configuration variable in the main.cf file). Only the
# pipe, virtual and local delivery daemons require privileges.
#
# Chroot: whether or not the service runs chrooted to the mail queue
# directory (pathname is controlled by the queue_directory configuration
# variable in the main.cf file). Presently, all Postfix daemons can run
# chrooted, except for the pipe, virtual and local delivery daemons.
# The proxymap server can run chrooted, but doing so defeats most of
# the purpose of having that service in the first place.
# The files in the examples/chroot-setup subdirectory describe how
# to set up a Postfix chroot environment for your type of machine.
#
# Wakeup time: automatically wake up the named service after the
# specified number of seconds. A ? at the end of the wakeup time
# field requests that wake up events be sent only to services that
# are actually being used.  Specify 0 for no wakeup. Presently, only
# the pickup, queue manager and flush daemons need a wakeup timer.
#
# Max procs: the maximum number of processes that may execute this
# service simultaneously. Default is to use a globally configurable
# limit (the default_process_limit configuration parameter in main.cf).
# Specify 0 for no process count limit.
#
# Command + args: the command to be executed. The command name is
# relative to the Postfix program directory (pathname is controlled by
# the daemon_directory configuration variable). Adding one or more
```

```
# -v options turns on verbose logging for that service; adding a -D
# option enables symbolic debugging (see the debugger_command variable
# in the main.cf configuration file). See individual command man pages
# for specific command-line options, if any.
#
# In order to use the "uucp" message tranport below, set up entries
# in the transport table.
#
# In order to use the "cyrus" message transport below, configure it
# in main.cf as the mailbox_transport.
#
# SPECIFY ONLY PROGRAMS THAT ARE WRITTEN TO RUN AS POSTFIX DAEMONS.
# ALL DAEMONS SPECIFIED HERE MUST SPEAK A POSTFIX-INTERNAL PROTOCOL.
#
# DO NOT SHARE THE POSTFIX QUEUE BETWEEN MULTIPLE POSTFIX INSTANCES.
#
# ==========================================================================
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)
# ==========================================================================
smtp    inet      n     -     y     -     -     smtpd
#smtps  inet      n     -     n     -     -     smtpd -o
smtpd_tls_wrappermode=yes
#  -o smtpd_tls_wrappermode=yes -o smtpd_sasl_auth_enable=yes
#submission inet  n     -     n     -     -     smtpd
#  -o smtpd_enforce_tls=yes -o smtpd_sasl_auth_enable=yes
#628    inet      n     -     n     -     -     qmqpd
pickup  fifo      n     -     y     60    1     pickup
cleanup unix      n     -     y     -     0     cleanup
qmgr    fifo      n     -     y     300   1     qmgr
#qmgr   fifo      n     -     n     300   1     nqmgr
#tlsmgr fifo  -       -       n       300     1       tlsmgr
rewrite unix      -     -     y     -     -     trivial-rewrite
bounce  unix      -     -     y     -     0     bounce
defer   unix      -     -     y     -     0     bounce
flush   unix      n     -     n     1000? 0     flush
proxymap unix     -     -     n     -     -     proxymap
smtp    unix      -     -     y     -     -     smtp
relay   unix      -     -     n     -     -     smtp
#     -o smtp_helo_timeout=5 -o smtp_connect_timeout=5
showq   unix      n     -     y     -     -     showq
error   unix      -     -     y     -     -     error
local   unix      -     n     n     -     -     local
virtual unix      -     n     y     -     -     virtual
lmtp    unix      -     -     y     -     -     lmtp


localhost:10025 inet    n     -     n     -     -     smtpd -o content_filter=
```

```
#
# Interfaces to non-Postfix software. Be sure to examine the manual
# pages of the non-Postfix software to find out what options it wants.
#
# maildrop. See the Postfix MAILDROP_README file for details.
#
maildrop  unix  -       n       n       -       -       pipe
  flags=DRhu user=vmail argv=/usr/local/bin/maildrop -d ${recipient}
cyrus   unix      -     n     n     -     -     pipe
  user=cyrus argv=/usr/lib/cyrus/bin/deliver -e -r ${sender} -m ${extension}
${user}
uucp    unix      -     n     n     -     -     pipe
  flags=Fqhu user=uucp argv=uux -r -n -z -a$sender - $nexthop!rmail
($recipient)
ifmail    unix  -       n       n       -       -       pipe
  flags=F user=ftn argv=/usr/lib/ifmail/ifmail -r $nexthop ($recipient)
bsmtp     unix  -       n       n       -       -       pipe
  flags=Fq. user=foo argv=/usr/local/sbin/bsmtp -f $sender $nexthop $recipient
vscan     unix  -       n       n       -       -       pipe
  user=vscan argv=/usr/sbin/amavis ${sender} ${recipient}
procmail  unix  -       n       n       -       -       pipe
  flags=R user=nobody argv=/usr/bin/procmail -t -m /etc/procmailrc ${sender}
${recipient}
```

Konfigurationsdatei „virtual“
(abgelegt unter “/etc/postfix/virtual”)

```
# VIRTUAL(5)                                                VIRTUAL(5)
#
# NAME
#       virtual - format of Postfix virtual alias table
#
# SYNOPSIS
#       postmap /etc/postfix/virtual
#
#       postmap -q "string" /etc/postfix/virtual
#
#       postmap -q - /etc/postfix/virtual <inputfile
#
# DESCRIPTION
#       The  optional virtual alias table specifies address alias-
#       ing for arbitrary local or non-local recipient  addresses.
#       Virtual  aliasing is recursive, and is done by the Postfix
#       cleanup(8) daemon.
#
#       The main applications of virtual aliasing are:
#
#       o      To redirect mail for one address  to  one  or  more
#              addresses.
#
#       o      To   implement  virtual  alias  domains  where  all
#              addresses  are  aliased  to  addresses   in   other
#              domains.
#
#              Virtual  alias  domains are not to be confused with
#              the virtual mailbox domains  that  are  implemented
#              with  the  Postfix  virtual(8) mail delivery agent.
#              With  virtual  mailbox  domains,   each   recipient
#              address can have its own mailbox.
#
#       Virtual  aliasing  is  applied  only to recipient envelope
#       addresses, and does not  affect  message  headers.   Think
#       Sendmail  rule  set S0, if you like. Use canonical(5) map-
#       ping to rewrite header and envelope addresses in  general.
#
#       Normally,  the  virtual alias table is specified as a text
#       file that serves as input to the postmap(1) command.   The
#       result,  an  indexed file in dbm or db format, is used for
#       fast searching by the mail  system.  Execute  the  command
#       postmap  /etc/postfix/virtual  in  order  to  rebuild  the
#       indexed file after changing the text file.
#
#       When the table is provided via other means  such  as  NIS,
#       LDAP  or  SQL,  the  same lookups are done as for ordinary
#       indexed files.
#
#       Alternatively, the table can be  provided  as  a  regular-
#       expression map where patterns are given as regular expres-
#       sions. In that case, the lookups are done  in  a  slightly
#       different way as described below.
```

```
#
# TABLE FORMAT
#        The input format for the postmap(1) command is as follows:
#
#        pattern result
#               When pattern matches a mail address, replace it  by
#               the corresponding result.
#
#        blank lines and comments
#               Empty  lines and whitespace-only lines are ignored,
#               as are lines whose first  non-whitespace  character
#               is a `#'.
#
#        multi-line text
#               A  logical  line starts with non-whitespace text. A
#               line that starts with whitespace continues a  logi-
#               cal line.
#
#        With lookups from indexed files such as DB or DBM, or from
#        networked tables such as NIS, LDAP or  SQL,  patterns  are
#        tried in the order as listed below:
#
#        user@domain address, address, ...
#               Mail  for  user@domain  is  redirected  to address.
#               This form has the highest precedence.
#
#        user address, address, ...
#               Mail for user@site is redirected  to  address  when
#               site  is equal to $myorigin, when site is listed in
#               $mydestination,   or   when   it   is   listed   in
#               $inet_interfaces.
#
#               This  functionality  overlaps with functionality of
#               the local aliases(5) database.  The  difference  is
#               that  virtual  mapping  can be applied to non-local
#               addresses.
#
#        @domain address, address, ...
#               Mail for  any  user  in  domain  is  redirected  to
#               address.  This form has the lowest precedence.
#
#        In  all the above forms, when address has the form @other-
#        domain, the result is the same user in otherdomain.   This
#        works for the first address in the expansion only.
#
# ADDRESS EXTENSION
#        When a mail address localpart contains the optional recip-
#        ient delimiter (e.g., user+foo@domain), the  lookup  order
#        becomes: user+foo@domain, user@domain, user+foo, user, and
#        @domain.  An unmatched address extension (+foo) is  propa-
#        gated to the result of table lookup.
#
# VIRTUAL ALIAS DOMAINS
#        Besides  virtual aliases, the virtual alias table can also
#        be used to implement virtual alias domains. With a virtual
#        alias  domain,  all  recipient  addresses  are  aliased to
#        addresses in other domains.
#
```

```
#        Virtual alias domains are not to be confused with the vir-
#        tual mailbox domains that are implemented with the Postfix
#        virtual(8)  mail  delivery  agent.  With  virtual  mailbox
#        domains,  each recipient address can have its own mailbox.
#
#        With a virtual alias domain, the virtual  domain  has  its
#        own  user  name  space. Local (i.e. non-virtual) usernames
#        are not visible in a virtual alias domain. In  particular,
#        local  aliases(5)  and local mailing lists are not visible
#        as localname@virtual-alias.domain.
#
#        Support for a virtual alias domain looks like:
#
#        /etc/postfix/main.cf:
#            virtual_alias_maps = hash:/etc/postfix/virtual
#
#            Note: some systems use dbm databases instead of  hash.
#            See the output from postconf -m for available database
#            types.
#
#        /etc/postfix/virtual:
#            virtual-alias.domain anything (right-hand content does not
matter)
#            postmaster@virtual-alias.domain      postmaster
#            user1@virtual-alias.domain   address1
#            user2@virtual-alias.domain   address2, address3
#
#        The virtual-alias.domain anything entry is required for  a
#        virtual alias domain. Without this entry, mail is rejected
#        with "relay access denied", or bounces  with  "mail  loops
#        back to myself".
#
#        Do  not  specify virtual alias domain names in the main.cf
#        mydestination or relay_domains configuration parameters.
#
#        With a virtual  alias  domain,  the  Postfix  SMTP  server
#        accepts   mail  for  known-user@virtual-alias.domain,  and
#        rejects  mail  for  unknown-user@virtual-alias.domain   as
#        undeliverable.
#
#        Instead  of  specifying  the virtual alias domain name via
#        the virtual_alias_maps table, you may also specify it  via
#        the main.cf virtual_alias_domains configuration parameter.
#        This latter parameter uses the same syntax as the  main.cf
#        mydestination configuration parameter.
#
# REGULAR EXPRESSION TABLES
#        This  section  describes how the table lookups change when
#        the table is given in the form of regular expressions. For
#        a  description  of regular expression lookup table syntax,
#        see regexp_table(5) or pcre_table(5).
#
#        Each pattern is a regular expression that  is  applied  to
#        the entire address being looked up. Thus, user@domain mail
#        addresses are not broken up into their  user  and  @domain
#        constituent parts, nor is user+foo broken up into user and
#        foo.
#
```

```
#        Patterns are applied in the  order  as  specified  in  the
#        table,  until  a  pattern is found that matches the search
#        string.
#
#        Results are the same as with indexed  file  lookups,  with
#        the  additional feature that parenthesized substrings from
#        the pattern can be interpolated as $1, $2 and so on.
#
# BUGS
#        The table format does not understand quoting  conventions.
#
# CONFIGURATION PARAMETERS
#        The  following  main.cf parameters are especially relevant
#        to this topic. See the Postfix  main.cf  file  for  syntax
#        details  and  for  default  values. Use the postfix reload
#        command after a configuration change.
#
#        virtual_alias_maps
#               List of virtual aliasing tables.
#
#        virtual_alias_domains
#               List of virtual alias domains. This uses  the  same
#               syntax as the mydestination parameter.
#
#        Other parameters of interest:
#
#        inet_interfaces
#               The  network  interface  addresses that this system
#               receives mail on.  You need to stop and start Post-
#               fix when this parameter changes.
#
#        mydestination
#               List  of  domains  that  this mail system considers
#               local.
#
#        myorigin
#               The domain that is appended  to  any  address  that
#               does not have a domain.
#
#        owner_request_special
#               Give special treatment to owner-xxx and xxx-request
#               addresses.
#
# SEE ALSO
#        cleanup(8) canonicalize and enqueue mail
#        postmap(1) create mapping table
#        regexp_table(5) POSIX regular expression table format
#        pcre_table(5) Perl Compatible Regular Expression table format
#
# LICENSE
#        The Secure Mailer license must be  distributed  with  this
#        software.
#
# AUTHOR(S)
#        Wietse Venema
#        IBM T.J. Watson Research
#        P.O. Box 704
#        Yorktown Heights, NY 10598, USA
```

```
#
#                                                          VIRTUAL(5)
# User Test
test@bbsnw.de           test

# User Mone
mone@bbsnw.de           mone
simmy@bbsnw.de          mone@bbsnw.de
chester@bbsnw.de        mone@bbsnw.de

# User Fips
fips@bbsnw.de           fips

# User Jogy
jogy@bbsnw.de           jogy

# User Chris
chris@bbsnw.de          cj
cjakob@bbsnw.de         chriss@bbsnw.de
cj@bbsnw.de             chriss@bbsnw.de
```

Konfigurationsdatei „amavisd.conf“
(abgelegt unter “/etc/amavisd.conf”)

```
use strict;

# Configuration file for amavisd-new
#
# This software is licensed under the GNU General Public License (GPL).
# See comments at the start of amavisd-new for the whole license text.

#Sections:
# Section I    - Essential daemon and MTA settings
# Section II   - MTA specific
# Section III  - Logging
# Section IV   - Notifications/DSN, BOUNCE/REJECT/DROP/PASS destiny,
quarantine
# Section V    - Per-recipient and per-sender handling, whitelisting, etc.
# Section VI   - Resource limits
# Section VII  - External programs, virus scanners, SpamAssassin
# Section VIII - Debugging

#GENERAL NOTES:
#  This file is a normal Perl code, interpreted by Perl itself.
#  - make sure this file (or directory where it resides) is NOT WRITABLE
#    by mere mortals, otherwise it represents a severe security risk!
#  - for values which are interpreted as booleans, it is recommended
#    to use 1 for true, and 0 or undef or '' for false.
#    THIS IS DIFFERENT FROM OLDER AMAVIS VERSIONS where "no" also meant false,
#    now it means true, like any nonempty string does!
#  - Perl syntax applies. Most notably: strings in "" may include variables
#    (which start with $ or @); to include characters @ and $ in double
#    quoted strings, precede them by a backslash; in single-quoted strings
#    the $ and @ lose their special meaning, so it is usually easier to use
#    single quoted strings. Still, in both cases a backslash need to be
doubled
#  - variables with names starting with a '@' are lists, the values assigned
#    to them should be lists as well, e.g. ('one@foo', $mydomain, "three");
#    note the comma-separation and parenthesis. If strings in the list
#    do not contain spaces nor variables, a Perl operator qw() may be used
#    as a shorthand to split its argument on whitespace and produce a list
#    of strings, e.g. qw( one@foo example.com three );  Note that the argument
#    to qw is quoted implicitly and no variable interpretation is done within
#    (no '$' variable evaluations). The #-initiated comments can not be used
#    within the string. In other words, $ and # lose their special meaning
#    withing a qw argument, just like within '...' strings.
#  - all e-mail addresses in this file and as used internally by the daemon
#    are in their raw (rfc2821-unquoted and nonbracketed) form, i.e.
#    Bob "Funny" Dude@example.com, not: "Bob \"Funny\" Dude"@example.com
#    and not <"Bob \"Funny\" Dude"@example.com>; also: '' and not '<>'.


#
# Section I - Essential daemon and MTA settings
#

# $MYHOME serves as a quick default for some other configuration settings.
```

```
# More refined control is available with each individual setting further down.
# $MYHOME is not used directly by the program. No trailing slash!
$MYHOME = '/var/spool/amavis';

# $mydomain serves as a quick default for some other configuration settings.
# More refined control is available with each individual setting further down.
# $mydomain is never used directly by the program.
$mydomain = 'bbsnw.de';      # (no useful default)

# Set the user and group to which the daemon will change if started as root
# (otherwise just keep the UID unchanged, and these settings have no effect):
$daemon_user = 'vscan';
$daemon_group = 'vscan';

# Runtime working directory (cwd), and a place where
# temporary directories for unpacking mail are created.
# (no trailing slash, may be a scratch file system)
$TEMPBASE = $MYHOME;            # (must be set if other config vars use is)
#$TEMPBASE = "$MYHOME/tmp";      # prefer to keep home dir /var/amavis clean?

# $helpers_home sets environment variable HOME, and is passed as option
# 'home_dir_for_helpers' to Mail::SpamAssassin::new. It should be a directory
# on a normal persistent file system, not a scratch or temporary file system
#$helpers_home = $MYHOME;     # (defaults to $MYHOME)

#$daemon_chroot_dir = $MYHOME;  # (default is undef, meaning: do not chroot)

#$pid_file  = "$MYHOME/amavisd.pid";  # (default is "$MYHOME/amavisd.pid")
#$lock_file = "$MYHOME/amavisd.lock"; # (default is "$MYHOME/amavisd.lock")

# set environment variables if you want (no defaults):
$ENV{TMPDIR} = $TEMPBASE;       # wise, but usually not necessary
#...

# MTA SETTINGS, UNCOMMENT AS APPROPRIATE,
# both $forward_method and $notify_method default to 'smtp:127.0.0.1:10025'

# POSTFIX, or SENDMAIL in dual-MTA setup, or EXIM V4
# (set host and port number as required; host can be specified
# as IP address or DNS name (A or CNAME, but MX is ignored)
$forward_method = 'smtp:127.0.0.1:10025';  # where to forward checked mail
$notify_method = $forward_method;          # where to submit notifications

# NOTE: The defaults (above) are good for Postfix or dual-sendmail. You MUST
#       uncomment the approprate settings below if using other setups!

# SENDMAIL MILTER, using amavis-milter.c helper program:
#$forward_method = undef;  # no explicit forwarding, sendmail does it by
itself
# milter; option -odd is needed to avoid deadlocks
#$notify_method = 'pipe:flags=q argv=/usr/sbin/sendmail -Ac -i -odd -f
${sender} -- ${recipient}';
# just a thought: can we use use -Am instead of -odd ?

# SENDMAIL (old non-milter setup, as relay):
#$forward_method = 'pipe:flags=q argv=/usr/sbin/sendmail -
C/etc/sendmail.orig.cf -i -f ${sender} -- ${recipient}';
#$notify_method = $forward_method;
```

```
# SENDMAIL (old non-milter setup, amavis.c calls local delivery agent):
#$forward_method = undef;  # no explicit forwarding, amavis.c will call LDA
#$notify_method = 'pipe:flags=q argv=/usr/sbin/sendmail -Ac -i -f ${sender} --
${recipient}';

# EXIM v3 (not recommended with v4 or later, which can use SMTP setup
instead):
#$forward_method = 'pipe:flags=q argv=/usr/sbin/exim -oMr scanned-ok -i -f
${sender} -- ${recipient}';
#$notify_method = $forward_method;

# prefer to collect mail for forwarding as BSMTP files?
#$forward_method = "bsmtp:$MYHOME/out-%i-%n.bsmtp";
#$notify_method = $forward_method;


# Net::Server pre-forking settings
# You may want $max_servers to match the width of your MTA pipe
# feeding amavisd, e.g. with Postfix the 'Max procs' field in the
# master.cf file, like the '2' in the:  smtp-amavis unix - - n - 2 smtp
#
$max_servers  =  2;   # number of pre-forked children          (default 2)
$max_requests = 10;   # retire a child after that many accepts (default 10)

$child_timeout=5*60;  # abort child if it does not complete each task in n sec
                      # (default: 8*60 seconds)

# Check also the settings of @av_scanners at the end if you want to use
# virus scanners. If not, you may want to delete the whole long assignment
# to the variable @av_scanners, which will also remove the virus checking
# code (e.g. if you only want to do spam scanning).

# Here is a QUICK WAY to completely DISABLE some sections of code
# that WE DO NOT WANT (it won't even be compiled-in).
# For more refined controls leave the following two lines commented out,
# and see further down what these two lookup lists really mean.
#
# @bypass_virus_checks_acl = qw( . );  # uncomment to DISABLE anti-virus code
# @bypass_spam_checks_acl  = qw( . );  # uncomment to DISABLE anti-spam code
#
# Any setting can be changed with a new assignment, so make sure
# you do not unintentionally override these settings further down!


# Lookup list of local domains (see README.lookups for syntax details)
#
# NOTE:
#   For backwards compatibility the variable names @local_domains (old) and
#   @local_domains_acl (new) are synonyms. For consistency with other lookups
#   the name @local_domains_acl is now preferred. It also makes it more
#   obviously distinct from the new %local_domains hash lookup table.
#
# local_domains* lookup tables are used in deciding whether a recipient
# is local or not, or in other words, if the message is outgoing or not.
# This affects inserting spam-related headers for local recipients,
# limiting recipient virus notifications (if enabled) to local recipients,
# in deciding if address extension may be appended, and in SQL lookups
```

```
# for non-fqdn addresses. Set it up correctly if you need features
# that rely on this setting (or just leave empty otherwise).
#
# With Postfix (2.0) a quick reminder on what local domains normally are:
# a union of domains spacified in: $mydestination, $virtual_alias_domains,
# $virtual_mailbox_domains, and $relay_domains.
#
@local_domains_acl = ( ".$mydomain" );  # $mydomain and its subdomains
# @local_domains_acl = qw();  # default is empty, no recipient treated as
local
# @local_domains_acl = qw( .example.com );
# @local_domains_acl = qw( .example.com !host.sub.example.net .sub.example.net
);
# @local_domains_acl = ( ".$mydomain", '.example.com', 'sub.example.net' );

# or alternatively(A), using a Perl hash lookup table, which may be assigned
# directly, or read from a file, one domain per line; comments and empty lines
# are ignored, a dot before a domain name implies its subdomains:
#
#read_hash(\%local_domains, '/var/amavis/local_domains');

#or alternatively(B), using a list of regular expressions:
# $local_domains_re = new_RE( qr'[@.]example\.com$'i );
#
# see README.lookups for syntax and semantics


#
# Section II - MTA specific (defaults should be ok)
#

# if $relayhost_is_client is true, IP address in $notify_method and
# $forward_method is dynamically overridden with SMTP client peer address
# if available, which makes possible for several hosts to share one daemon
#$relayhost_is_client = 1;        # (defaults to false)

#$insert_received_line = 1;       # behave like MTA: insert 'Received:' header
                            # (does not apply to sendmail/milter)
                            # (default is true)

# AMAVIS-CLIENT PROTOCOL INPUT SETTINGS (e.g. with sendmail milter)
#   (used with amavis helper clients like amavis-milter.c and amavis.c,
#   NOT needed for Postfix and Exim)
$unix_socketname = "$MYHOME/amavisd.sock"; # amavis helper protocol socket
#$unix_socketname = undef;        # disable listening on a unix socket
                                  # (default is undef, i.e. disabled)
                                  # (usual setting is $MYHOME/amavisd.sock)

# Do we receive quoted or raw addresses from the helper program?
# (does not apply to SMTP;  defaults to true)
#$gets_addr_in_quoted_form = 1;   # "Bob \"Funny\" Dude"@example.com
#$gets_addr_in_quoted_form = 0;   # Bob "Funny" Dude@example.com



# SMTP SERVER (INPUT) PROTOCOL SETTINGS (e.g. with Postfix, Exim v4, ...)
#   (used when MTA is configured to pass mail to amavisd via SMTP or LMTP)
$inet_socket_port = 10024;        # accept SMTP on this local TCP port
```

```
                                  # (default is undef, i.e. disabled)
# multiple ports may be provided: $inet_socket_port = [10024, 10026, 10028];

# SMTP SERVER (INPUT) access control
# - do not allow free access to the amavisd SMTP port !!!
#
# when MTA is at the same host, use the following (one or the other or both):
#$inet_socket_bind = '127.0.0.1'; # limit socket bind to loopback interface
                                  # (default is '127.0.0.1')
@inet_acl = qw( 127.0.0.1 172.16.0.0/16);      # allow SMTP access only from
localhost IP
                                  # (default is qw( 127.0.0.1 ) )

# when MTA (one or more) is on a different host, use the following:
#@inet_acl = qw(127/8 10.1.0.1 10.1.0.2);  # adjust the list as appropriate
#$inet_socket_bind = undef;       # bind to all IP interfaces

#
# Example1:
# @inet_acl = qw( 127/8 10/8 172.16/12 192.168/16 );
# permit only SMTP access from loopback and rfc1918 private address space
#
# Example2:
# @inet_acl = qw( !192.168.1.12 172.16.3.3 !172.16.3/255.255.255.0
#             127.0.0.1 10/8 172.16/12 192.168/16 );
# matches loopback and rfc1918 private address space except host 192.168.1.12
# and net 172.16.3/24 (but host 172.16.3.3 within 172.16.3/24 still matches)
#
# Example3:
# @inet_acl = qw( 127/8
#             !172.16.3.0   !172.16.3.127 172.16.3.0/25
#             !172.16.3.128 !172.16.3.255 172.16.3.128/25 );
# matches loopback and both halves of the 172.16.3/24 C-class,
# split into two subnets, except all four broadcast addresses
# for these subnets
#
# See README.lookups for details on specifying access control lists.


#
# Section III - Logging
#

# true (e.g. 1) => syslog;  false (e.g. 0) => logging to file
$DO_SYSLOG = 1;                   # (defaults to false)
#$SYSLOG_LEVEL = 'user.info';     # (defaults to 'mail.info')

# Log file (if not using syslog)
$LOGFILE = "$MYHOME/amavis.log";  # (defaults to empty, no log)

#NOTE: levels are not strictly observed and are somewhat arbitrary
# 0: startup/exit/failure messages, viruses detected
# 1: args passed from client, some more interesting messages
# 2: virus scanner output, timing
# 3: server, client
# 4: decompose parts
# 5: more debug details
$log_level = 2;           # (defaults to 0)
```

```
# Customizable template for the most interesting log file entry (e.g. with
# $log_level=0) (take care to properly quote Perl special characters like '\')
# For a list of available macros see README.customize .

# only log infected messages (useful with log level 0):
# $log_templ = '[? %#V |[? %#F ||banned filename ([%F|,])]|infected ([%V|,])]#
# [? %#V |[? %#F ||, from=<%o>, to=[<%R>|,][? %i ||, quarantine %i]]#
# |, from=<%o>, to=[<%R>|,][? %i ||, quarantine %i]]';

# log both infected and noninfected messages (default):
$log_templ = '[? %#V |[? %#F |[?%#D|Not-Delivered|Passed]|BANNED name/type
(%F)]|INFECTED (%V)], #
<%o> -> [<%R>|,][? %i ||, quarantine %i], Message-ID: %m, Hits: %c';


#
# Section IV - Notifications/DSN, BOUNCE/REJECT/DROP/PASS destiny, quarantine
#

# Select notifications text encoding when Unicode-aware Perl is converting
# text from internal character representation to external encoding (charset
# in MIME terminology)
#
#   to be used in RFC 2047-encoded header field bodies, e.g. in Subject:
#$hdr_encoding = 'iso-8859-1';  # (default: 'iso-8859-1')
#
#   to be used in notification body text: its encoding and Content-
type.charset
#$bdy_encoding = 'iso-8859-1';  # (default: 'iso-8859-1')

# Default template texts for notifications may be overruled by directly
# assigning new text to template variables, or by reading template text
# from files. A second argument may be specified in a call to read_text(),
# specifying character encoding layer to be used when reading from the
# external file, e.g. 'utf8', 'iso-8859-1', or often just $bdy_encoding.
# Text will be converted to internal character representation by Perl 5.8.0
# or later; second argument is ignored otherwise. See PerlIO::encoding,
# Encode::PerlIO and perluniintro man pages.
#
# $notify_sender_templ      = read_text('/var/amavis/notify_sender.txt');
# $notify_virus_sender_templ=
read_text('/var/amavis/notify_virus_sender.txt');
# $notify_virus_admin_templ = read_text('/var/amavis/notify_virus_admin.txt');
# $notify_virus_recips_templ=
read_text('/var/amavis/notify_virus_recips.txt');
# $notify_spam_sender_templ = read_text('/var/amavis/notify_spam_sender.txt');
# $notify_spam_admin_templ  = read_text('/var/amavis/notify_spam_admin.txt');

# If notification template files are collectively available in some directory,
# use read_l10n_templates which calls read_text for each known template.
#
#   read_l10n_templates('/etc/amavis/en_US');


# Here is an overall picture (sequence of events) of how pieces fit together
# (only virus controls are shown, spam controls work the same way):
#
```

```
#   bypass_virus_checks set for all recipients? ==> PASS
#   no viruses?   ==> PASS
#   log virus     if $log_templ is nonempty
#   quarantine    if $virus_quarantine_to is nonempty
#   notify admin  if $virus_admin (lookup) nonempty
#   notify recips if $warnvirusrecip and (recipient is local or $warn_offsite)
#   add address extensions for local recipients (when enabled)
#   send (non-)delivery notifications
#      to sender if DSN needed (BOUNCE) or ($warnvirussender and D_PASS)
#   virus_lovers or final_destiny==D_PASS  ==> PASS
#   DISCARD (2xx) or REJECT (5xx) (depending on final_*_destiny)
#
# Equivalent flow diagram applies for spam checks.
# If a virus is detected, spam checking is skipped entirely.

# The following symbolic constants can be used in *destiny settings:
#
# D_PASS     mail will pass to recipients, regardless of bad contents;
#
# D_DISCARD  mail will not be delivered to its recipients, sender will NOT be
#            notified. Effectively we lose mail (but will be quarantined
#            unless disabled). Not a decent thing to do for a mailer.
#
# D_BOUNCE   mail will not be delivered to its recipients, a non-delivery
#            notification (bounce) will be sent to the sender by amavisd-new;
#            Exception: bounce (DSN) will not be sent if a virus name matches
#            $viruses_that_fake_sender_re, or to messages from mailing lists
#            (Precedence: bulk|list|junk);
#
# D_REJECT   mail will not be delivered to its recipients, sender should
#            preferably get a reject, e.g. SMTP permanent reject response
#            (e.g. with milter), or non-delivery notification from MTA
#            (e.g. Postfix). If this is not possible (e.g. different
recipients
#            have different tolerances to bad mail contents and not using
LMTP)
#            amavisd-new sends a bounce by itself (same as D_BOUNCE).
#
# Notes:
#   D_REJECT and D_BOUNCE are similar, the difference is in who is responsible
#            for informing the sender about non-delivery, and how informative
#            the notification can be (amavisd-new knows more than MTA);
#   With D_REJECT, MTA may reject original SMTP, or send DSN (delivery status
#            notification, colloquially called 'bounce') - depending on MTA;
#            Best suited for sendmail milter, especially for spam.
#   With D_BOUNCE, amavisd-new (not MTA) sends DSN (can better explain the
#            reason for mail non-delivery, but unable to reject the original
#            SMTP session). Best suited to reporting viruses, and for Postfix
#            and other dual-MTA setups, which can't reject original client
SMTP
#            session, as the mail has already been enqueued.

$final_virus_destiny      = D_REJECT;  # (defaults to D_BOUNCE)
$final_banned_destiny     = D_BOUNCE;  # (defaults to D_BOUNCE)
$final_spam_destiny = D_PASS;
$final_bad_header_destiny = D_PASS;  # (defaults to D_PASS), D_BOUNCE
suggested
```

```
# Alternatives to consider for spam:
# - use D_PASS if clients will do filtering based on inserted mail headers;
# - use D_DISCARD, if kill_level is set safely high;
# - use D_BOUNCE instead of D_REJECT if not using milter;
#
# There are no sensible alternatives to D_BOUNCE for viruses, but consider:
# - use D_PASS (or virus_lovers) and $warnvirussender=1 to deliver viruses;
# - use D_REJECT instead of D_BOUNCE if using milter and under heavy
#   virus storm;
#
# Don't bother to set both D_DISCARD and $warn*sender=1, it will get mapped
# to D_BOUNCE.
#
# The separation of *_destiny values into D_BOUNCE, D_REJECT, D_DISCARD
# and D_PASS made settings $warnvirussender and $warnspamsender only still
# useful with D_PASS.

# The following $warn*sender settings are ONLY used when mail is
# actually passed to recipients ($final_*_destiny=D_PASS, or *_lovers*).
# Bounces or rejects produce non-delivery status notification anyway.

# Notify virus sender?
#$warnvirussender = 1;  # (defaults to false (undef))

# Notify spam sender?
#$warnspamsender = 1;   # (defaults to false (undef))

# Notify sender of banned files?
#$warnbannedsender = 1; # (defaults to false (undef))

# Notify sender of syntactically invalid header containing non-ASCII
characters?
#$warnbadhsender = 1;   # (defaults to false (undef))

# Notify virus (or banned files) RECIPIENT?
#  (not very useful, but some policies demand it)
$warnvirusrecip = 1;    # (defaults to false (undef))
#$warnbannedrecip = 1;  # (defaults to false (undef))

# Notify also non-local virus/banned recipients if $warn*recip is true?
#  (including those not matching local_domains*)
#$warn_offsite = 1;     # (defaults to false (undef), i.e. only notify locals)


# Treat envelope sender address as unreliable and don't send sender
# notification / bounces if name(s) of detected virus(es) match the list.
# Note that virus names are supplied by external virus scanner(s) and are
# not standardized, so virus names may need to be adjusted.
# See README.lookups for syntax.
#
$viruses_that_fake_sender_re = new_RE(
  qr'nimda|hybris|klez|bugbear|yaha|braid|sobig|fizzer|palyh|peido|holar'i,
  qr'tanatos|lentin|bridex|mimail|trojan\.dropper'i,
);


# where to send ADMIN VIRUS NOTIFICATIONS (should be a fully qualified
address)
```

```
# - the administrator address may be a simple fixed e-mail address (a scalar),
#   or may depend on the SENDER address (e.g. its domain), in which case
#   a ref to a hash table can be specified (specify lower-cased keys,
#   dot is a catchall, see README.lookups).
#
#   Empty or undef lookup disables virus admin notifications.

$virus_admin = "virusalert\@$mydomain";
# $virus_admin = undef;   # do not send virus admin notifications (default)
# $virus_admin = {'not.example.com' => '', '.' => 'virusalert@example.com'};
# $virus_admin = 'virus-admin@example.com';

# equivalent to $virus_admin, but for spam admin notifications:
# $spam_admin = "spamalert\@$mydomain";
# $spam_admin = undef;    # do not send spam admin notifications (default)
# $spam_admin = {'not.example.com' => '', '.' => 'spamalert@example.com'};

#advanced example, using a hash lookup table:
#$virus_admin = {
# 'baduser@sub1.example.com' => 'HisBoss@sub1.example.com',
# '.sub1.example.com'  => 'virusalert@sub1.example.com',
# '.sub2.example.com'  => '',                  # don't send admin
notifications
# 'a.sub3.example.com' => 'abuse@sub3.example.com',
# '.sub3.example.com'  => 'virusalert@sub3.example.com',
# '.example.com'       => 'noc@example.com',   # catchall for our virus
senders
# '.'                  => 'virusalert@hq.example.com',  # catchall for the
rest
#};


# whom notification reports are sent from (ENVELOPE SENDER);
# may be a null reverse path, or a fully qualified address:
#   (admin and recip sender addresses default to $mailfrom
#   for compatibility, which in turn defaults to undef (empty) )
#   If using strings in double quotes, don't forget to quote @, i.e. \@
#
$mailfrom_notify_admin     = "virusalert\@$mydomain";
$mailfrom_notify_recip     = "virusalert\@$mydomain";
$mailfrom_notify_spamadmin = "spam.police\@$mydomain";

# 'From' HEADER FIELD for sender and admin notifications.
# This should be a replyable address, see rfc1894. Not to be confused
# with $mailfrom_notify_sender, which is the envelope address and
# should be empty (null reverse path) according to rfc2821.
#
# $hdrfrom_notify_sender = "amavisd-new <postmaster\@$mydomain>";
# $hdrfrom_notify_sender = 'amavisd-new <postmaster@example.com>';
#   (defaults to: "amavisd-new <postmaster\@$myhostname>")
# $hdrfrom_notify_admin = $mailfrom_notify_admin;
#   (defaults to: $mailfrom_notify_admin)
# $hdrfrom_notify_spamadmin = $mailfrom_notify_spamadmin;
#   (defaults to: $mailfrom_notify_spamadmin)

# whom quarantined messages appear to be sent from (envelope sender)
$mailfrom_to_quarantine = undef; # original sender if undef, or set explicitly
                          # (default is undef)
```

```
# Location to put infected mail into: (applies to 'local:' quarantine method)
#   empty for not quarantining, may be a file (mailbox),
#   or a directory (no trailing slash)
#   (the default value is undef, meaning no quarantine)
#
$QUARANTINEDIR = '/var/spool/amavis/virusmails';

#$virus_quarantine_method = "local:virus-%i-%n";    # default
#$spam_quarantine_method  = "local:spam-%b-%i-%n";  # default
#
#use the new 'bsmtp:' method as an alternative to the default 'local:'
#$virus_quarantine_method = "bsmtp:$QUARANTINEDIR/virus-%i-%n.bsmtp";
#$spam_quarantine_method  = "bsmtp:$QUARANTINEDIR/spam-%b-%i-%n.bsmtp";

# When using the 'local:' quarantine method (default), the following applies:
#
# A finer control of quarantining is available through variable
# $virus_quarantine_to/$spam_quarantine_to. It may be a simple scalar string,
# or a ref to a hash lookup table, or a regexp lookup table object,
# which makes possible to set up per-recipient quarantine addresses.
#
# The value of scalar $virus_quarantine_to/$spam_quarantine_to (or a
# per-recipient lookup result from the hash table %$virus_quarantine_to)
# is/are interpreted as follows:
#
# VARIANT 1:
#   empty or undef disables quarantine;
#
# VARIANT 2:
#   a string NOT containg an '@';
# amavisd will behave as a local delivery agent (LDA) and will quarantine
# viruses to local files according to hash %local_delivery_aliases (pseudo
# aliases map) - see subroutine mail_to_local_mailbox() for details.
# Some of the predefined aliases are 'virus-quarantine' and 'spam-quarantine'.
# Setting $virus_quarantine_to ($spam_quarantine_to) to this string will:
#
# * if $QUARANTINEDIR is a directory, each quarantined virus will go
#   to a separate file in the $QUARANTINEDIR directory (traditional
#   amavis style, similar to maildir mailbox format);
#
# * otherwise $QUARANTINEDIR is treated as a file name of a Unix-style
#   mailbox. All quarantined messages will be appended to this file.
#   Amavisd child process must obtain an exclusive lock on the file during
#   delivery, so this may be less efficient than using individual files
#   or forwarding to MTA, and it may not work across NFS or other non-local
#   file systems (but may be handy for pickup of quarantined files via IMAP
#   for example);
#
# VARIANT 3:
#   any email address (must contain '@').
# The e-mail messages to be quarantined will be handed to MTA
# for delivery to the specified address. If a recipient address local to MTA
# is desired, you may leave the domain part empty, e.g. 'infected@', but the
# '@' character must nevertheless be included to distinguish it from variant
2.
#
```

```
# This method enables more refined delivery control made available by MTA
# (e.g. its aliases file, other local delivery agents, dealing with
# privileges and file locking when delivering to user's mailbox, nonlocal
# delivery and forwarding, fan-out lists). Make sure the mail-to-be-
quarantined
# will not be handed back to amavisd for checking, as this will cause a loop
# (hopefully broken at some stage)! If this can be assured, notifications
# will benefit too from not being unecessarily virus-scanned.
#
# By default this is safe to do with Postfix and Exim v4 and dual-sendmail
# setup, but probably not safe with sendmail milter interface without
# precaution.

# (the default value is undef, meaning no quarantine)

$virus_quarantine_to  = 'virus-quarantine';    # traditional local quarantine
#$virus_quarantine_to = 'infected@';           # forward to MTA for delivery
#$virus_quarantine_to = "virus-quarantine\@$mydomain";   # similar
#$virus_quarantine_to = 'virus-quarantine@example.com';  # similar
#$virus_quarantine_to = undef;                 # no quarantine
#
#$virus_quarantine_to = new_RE(                # per-recip multiple
quarantines
#  [qr'^user@example\.com$'i => 'infected@'],
#  [qr'^(.*)@example\.com$'i => 'virus-${1}@example.com'],
#  [qr'^(.*)(@[^@])?$'i      => 'virus-${1}${2}'],
#  [qr/.*/                   => 'virus-quarantine'] );

# similar for spam
# (the default value is undef, meaning no quarantine)
#
$spam_quarantine_to = undef;
#$spam_quarantine_to = "spam-quarantine\@$mydomain";
#$spam_quarantine_to = new_RE(                 # per-recip multiple
quarantines
#  [qr'^(.*)@example\.com$'i => 'spam-${1}@example.com'],
#  [qr/.*/                   => 'spam-quarantine'] );

# In addition to per-recip quarantine, a by-sender lookup is possible. It is
# similar to $spam_quarantine_to, but the lookup key is the sender address:
#$spam_quarantine_bysender_to = undef;   # dflt: no by-sender spam quarantine


# Add X-Virus-Scanned header field to mail?
$X_HEADER_TAG = 'X-Virus-Scanned';  # (default: undef)
# Leave empty to add no header field     # (default: undef)
$X_HEADER_LINE = "by amavisd-new at $mydomain";

$remove_existing_x_scanned_headers = 0; # leave existing X-Virus-Scanned alone
#$remove_existing_x_scanned_headers= 1; # remove existing headers
                              # (defaults to false)
#$remove_existing_spam_headers = 0;     # leave existing X-Spam* headers alone
$remove_existing_spam_headers  = 1;     # remove existing spam headers if
                              # spam scanning is enabled (default)

# set $bypass_decode_parts to true if you only do spam scanning, or if you
# have a good virus scanner that can deal with compression and recursively
# unpacking archives by itself, and save amavisd the trouble.
```

```
# Disabling decoding also causes banned_files checking to only see
# MIME names and MIME content types, not the content classification types
# as provided by the file(1) utility.
# It is a double-edged sword, make sure you know what you are doing!
#
#$bypass_decode_parts = 1;          # (defaults to false)

# don't trust this file type or corresponding unpacker for this file type,
# keep both the original and the unpacked file
# (lookup key is what file(1) utility returned):
#
$keep_decoded_original_re = new_RE(
  qr'^(ASCII|text|uuencoded|xxencoded|binhex)'i,
);


# Checking for banned MIME types and names. If any mail part matches,
# the whole mail is rejected, much like the way viruses are handled.
# A list in object $banned_filename_re can be defined to provide a list
# of Perl regular expressions to be matched against each part's:
#
#  * Content-Type value (both declared and effective mime-type),
#    including the possible security risk content types
#    message/partial and message/external-body, as specified by rfc2046;
#
#  * declared (recommended) file names as specified by MIME subfields
#    Content-Disposition.filename and Content-Type.name, both in their
#    raw (encoded) form and in rfc2047-decoded form if applicable;
#
#  * file content type as guessed by 'file(1)' utility, both the raw result
#    from file(1), as well as short type name, classified into names such as
#    .asc, .txt, .html, .doc, .jpg, .pdf, .zip, .exe, ..., which is always
#    beginning with a dot - see subroutine determine_file_types().
#    This step is done only if $bypass_decode_parts is not true.
#
#  * leave $banned_filename_re undefined to disable these checks
#    (giving an empty list to new_RE() will also always return false)

$banned_filename_re = new_RE(
   qr'\.[a-zA-Z][a-zA-Z0-9]{0,3}\.(vbs|pif|scr|bat|com|exe|dll)$'i, # double
extension
#  qr'.\.(exe|vbs|pif|scr|bat|com)$'i,               # banned extension -
basic
#  qr'.\.(ade|adp|bas|bat|chm|cmd|com|cpl|crt|exe|hlp|hta|inf|ins|isp|js|
#         jse|lnk|mdb|mde|msc|msi|msp|mst|pcd|pif|reg|scr|sct|shs|shb|vb|
#         vbe|vbs|wsc|wsf|wsh)$'ix,                  # banned extension - long
#  qr'^\.(exe|zip|lha|tnef)$'i,                      # banned file(1) types
#  qr'^application/x-msdownload$'i,                  # banned MIME types
#  qr'^message/partial$'i, qr'^message/external-body$'i, # rfc2046
);
# See http://support.microsoft.com/default.aspx?scid=kb;EN-US;q262631
# and http://www.cknow.com/vtutor/vtextensions.htm

# A little trick: a pattern qr'\.exe$' matches both a short type name '.exe',
# as well as any file name which happens to end with .exe. If only matching
# a file name is desired, but not the short name, a pattern qr'.\.exe$'i
# or similar may be used, which requires that at least one character preceeds
# the '.exe', and so it will never match short file types, which always start
```

```
# with a dot.


#
# Section V - Per-recipient and per-sender handling, whitelisting, etc.
#

# %virus_lovers, @virus_lovers_acl and $virus_lovers_re lookup tables:
#   (these should be considered policy options, they do not disable checks,
#   see bypas*checks for that!)
#
# Exclude certain RECIPIENTS from virus filtering by adding their lower-cased
# envelope e-mail address (or domain only) to the hash %virus_lovers, or to
# the access list @virus_lovers_acl - see README.lookups and examples.
# Make sure the appropriate form (e.g. external/internal) of address
# is used in case of virtual domains, or when mapping external to internal
# addresses, etc. - this is MTA-specific.
#
# Notifications would still be generated however (see the overall
# picture above), and infected mail (if passed) gets additional header:
#   X-AMaViS-Alert: INFECTED, message contains virus: ...
# (header not inserted with milter interface!)
#
# NOTE (milter interface only): in case of multiple recipients,
# it is only possible to drop or accept the message in its entirety - for all
# recipients. If all of them are virus lovers, we'll accept mail, but if
# at least one recipient is not a virus lover, we'll discard the message.


# %bypass_virus_checks, @bypass_virus_checks_acl and $bypass_virus_checks_re
# lookup tables:
#   (this is mainly a time-saving option, unlike virus_lovers* !)
#
# Similar in concept to %virus_lovers, a hash %bypass_virus_checks,
# access list @bypass_virus_checks_acl and regexp list $bypass_virus_checks_re
# are used to skip entirely the decoding, unpacking and virus checking,
# but only if ALL recipients match the lookup.
#
# %bypass_virus_checks/@bypass_virus_checks_acl/$bypass_virus_checks_re
# do NOT GUARANTEE the message will NOT be checked for viruses - this may
# still happen when there is more than one recipient for a message, and
# not all of them match these lookup tables. To guarantee virus delivery,
# a recipient must also match %virus_lovers/@virus_lovers_acl lookups
# (but see milter limitations above),

# NOTE: it would not be clever to base virus checks on SENDER address,
# since there are no guarantees that it is genuine. Many viruses
# and spam messages fake sender address. To achieve selective filtering
# based on the source of the mail (e.g. IP address, MTA port number, ...),
# use mechanisms provided by MTA if available.


# Similar to lookup tables controlling virus checking, there exist
# spam scanning, banned names/types, and headers_checks control counterparts:
#   %spam_lovers, @spam_lovers_acl, $spam_lovers_re
#   %banned_files_lovers, @banned_files_lovers_acl, $banned_files_lovers_re
#   %bad_header_lovers, @bad_header_lovers_acl, $bad_header_lovers_re
# and:
```

```
#   %bypass_spam_checks/@bypass_spam_checks_acl/$bypass_spam_checks_re
#   %bypass_banned_checks/@bypass_banned_checks_acl/$bypass_banned_checks_re
#   %bypass_header_checks/@bypass_header_checks_acl/$bypass_header_checks_re
# See README.lookups for details about the syntax.

# The following example disables spam checking altogether,
# since it matches any recipient e-mail address (any address
# is a subdomain of the top-level root DNS domain):
#   @bypass_spam_checks_acl = qw( . );

#   @bypass_header_checks_acl = qw( user@example.com );
#   @bad_header_lovers_acl    = qw( user@example.com );


# See README.lookups for further detail, and examples below.

# $virus_lovers{lc("postmaster\@$mydomain")} = 1;
# $virus_lovers{lc('postmaster@example.com')} = 1;
# $virus_lovers{lc('abuse@example.com')} = 1;
# $virus_lovers{lc('some.user@')} = 1;  # this recipient, regardless of domain
# $virus_lovers{lc('boss@example.com')} = 0; # never, even if domain matches
# $virus_lovers{lc('example.com')} = 1; # this domain, but not its subdomains
# $virus_lovers{lc('.example.com')}= 1; # this domain, including its
subdomains
#or:
# @virus_lovers_acl = qw( me@lab.xxx.com !lab.xxx.com .xxx.com yyy.org );
#
# $bypass_virus_checks{lc('some.user2@butnot.example.com')} = 1;
# @bypass_virus_checks_acl = qw( some.ddd !butnot.example.com .example.com );

# @virus_lovers_acl = qw( postmaster@example.com );
# $virus_lovers_re = new_RE( qr'(helpdesk|postmaster)@example\.com$'i );

# $spam_lovers{lc("postmaster\@$mydomain")} = 1;
# $spam_lovers{lc('postmaster@example.com')} = 1;
# $spam_lovers{lc('abuse@example.com')} = 1;
# @spam_lovers_acl = qw( !.example.com );
# $spam_lovers_re = new_RE( qr'^user@example\.com$'i );


# don't run spam check for these RECIPIENT domains:
#   @bypass_spam_checks_acl = qw( d1.com .d2.com a.d3.com );
# or the other way around (bypass check for all BUT these):
#   @bypass_spam_checks_acl = qw( !d1.com !.d2.com !a.d3.com . );
# a practical application: don't check outgoing mail for spam:
#   @bypass_spam_checks_acl = ( "!.$mydomain", "." );
# (a downside of which is that such mail will not count as ham in SA bayes db)


# Where to find SQL server(s) and database to support SQL lookups?
# A list of triples: (dsn,user,passw).   (dsn = data source name)
# Specify more than one for multiple (backup) SQL servers.
# See 'man DBI', 'man DBD::mysql', 'DBD::Pg', ... for details.
#
# @lookup_sql_dsn =
#   ( ['DBI:mysql:mail:host1', 'some-username1', 'some-password1'],
#     ['DBI:mysql:mail:host2', 'some-username2', 'some-password2'] );
# ('mail' in the example is the database name, choose what you like)
```

```
# With PostgreSQL the dsn (first element of the triple) may look like:
#      'DBI:Pg:host=host1;dbname=mail'

# The SQL select clause to fetch per-recipient policy settings.
# The %k will be replaced by a comma-separated list of query addresses
# (e.g. full address, domain only, catchall).  Use ORDER, if there
# is a chance that multiple records will match - the first match wins.
# If field names are not unique (e.g. 'id'), the later field overwrites the
# earlier in a hash returned by lookup, which is why we use '*,users.id'.
# No need to uncomment the following assignment if the default is ok.
#   $sql_select_policy = 'SELECT *,users.id FROM users,policy'.
#     ' WHERE (users.policy_id=policy.id) AND (users.email IN (%k))'.
#     ' ORDER BY users.priority DESC';
#
# The SQL select clause to check sender in per-recipient whitelist/blacklist
# The first SELECT argument '?' will be users.id from recipient SQL lookup,
# the %k will be sender addresses (e.g. full address, domain only, catchall).
# The default value is:
#   $sql_select_white_black_list = 'SELECT wb FROM wblist,mailaddr'.
#     ' WHERE (rid=?) AND (sid=mailaddr.id) AND (mailaddr.email IN (%k))'.
#     ' ORDER BY mailaddr.priority DESC';
#
# To disable SQL white/black list, set to undef (otherwise comment-out
# the following statement, leaving it at the default value):
$sql_select_white_black_list = undef;  # undef disables SQL white/blacklisting


# If you decide to pass viruses (or spam) to certain recipients using the
# above lookup tables or using $final_virus_destiny=1, you can set
# the variable $addr_extension_virus ($addr_extension_spam) to some
# string, and the recipient address will have this string appended
# as an address extension to the local-part of the address. This extension
# can be used by final local delivery agent to place such mail in different
# folders. Leave these two variables undefined or empty strings to prevent
# appending address extensions. Setting has no effect on recipient which will
# not be receiving viruses/spam. Recipients who do not match lookup tables
# local_domains* are not affected.
#
# LDAs usually default to stripping away address extension if no special
# handling is specified, so having this option enabled normally does no harm,
# provided the $recipients_delimiter matches the setting on the final
# MTA's LDA.

# $addr_extension_virus  = 'virus'; # (default is undef, same as empty)
# $addr_extension_spam   = 'spam';  # (default is undef, same as empty)
# $addr_extension_banned = 'banned';     # (default is undef, same as empty)


# Delimiter between local part of the recipient address and address extension
# (which can optionally be added, see variables $addr_extension_virus and
# $addr_extension_spam). E.g. recipient address <user@example.com> gets
changed
# to <user+virus@example.com>.
#
# Delimiter should match equivalent (final) MTA delimiter setting.
# (e.g. for Postfix add 'recipient_delimiter = +' to main.cf)
# Setting it to an empty string or to undef disables this feature
# regardless of $addr_extension_virus and $addr_extension_spam settings.
```

```
$recipient_delimiter = '+';         # (default is '+')

# true: replace extension;  false: append extension
# $replace_existing_extension = 1;  # (default is false)

# Affects matching of localpart of e-mail addresses (left of '@')
# in lookups: true = case sensitive, false = case insensitive
$localpart_is_case_sensitive = 0;   # (default is false)


# ENVELOPE SENDER WHITELISTING / BLACKLISTING  - GLOBAL (RECIPIENT-
INDEPENDENT)

# WHITELISTING: use ENVELOPE SENDER lookups to ENSURE DELIVERY from
whitelisted
# senders even if the message is recognized as spam. Effectively, for the
# specified senders, message RECIPIENTS temporarily become 'spam_lovers', with
# further processing being the same as otherwise specified for spam lovers.
# It does not turn off inserting spam-related headers, if they are enabled.
#
# BLACKLISTING: messages from specified SENDERS are DECLARED SPAM.
# Effectively, for messages from blacklisted senders, spam level
# is artificially pushed high, and the normal spam processing applies,
# resulting in 'X-Spam-Flag: YES', high 'X-Spam-Level' bar and other usual
# reactions to spam, including possible rejection. If the message nevertheless
# still passes (e.g. for spam loving recipients), it is tagged as BLACKLISTED
# in the 'X-Spam-Status' header field, but the reported spam value and
# set of tests in this report header field (if available from SpamAssassin,
# which may have not been called) is not adjusted.
#
# A sender may be both white- and blacklisted at the same time,
# settings are independent. For example, being both white- and blacklisted,
# message is delivered to recipients, but is tagged as spam.
#
# If ALL recipients of the message either white- or blacklist the sender,
# spam scanning (calling the SpamAssassin) is bypassed, saving on time.
#
# The following variables (lookup tables) are available, with the semantics
# and syntax as specified in README.lookups:
#
# %whitelist_sender, @whitelist_sender_acl, $whitelist_sender_re
# %blacklist_sender, @blacklist_sender_acl, $blacklist_sender_re

# SOME EXAMPLES:
#
#ACL:
# @whitelist_sender_acl = qw( .example.com );
#
# @whitelist_sender_acl = ( ".$mydomain" );  # $mydomain and its subdomains
# NOTE: This is not a reliable way of turning off spam checks for
#       locally-originating mail, as sender address can easily be faked.
#       To reliably avoid spam-scanning outgoing mail,
#       use @bypass_spam_checks_acl .

#RE:
# $whitelist_sender_re = new_RE(
#   qr'^postmaster@.*\bexample\.com$'i,
```

```
#   qr'^owner-[^@]*@'i,  qr'-request@'i,
#   qr'\.example\.com$'i );
#
$blacklist_sender_re = new_RE(
    qr'^(bulkmail|offers|cheapbenefits|earnmoney|foryou|greatcasino)@'i,
    qr'^(investments|lose_weight_today|market.alert|money2you|MyGreenCard)@'i,
    qr'^(new\.tld\.registry|opt-out|opt-in|optin|saveonlsmoking2002k)@'i,
    qr'^(specialoffer|specialoffers|stockalert|stopsnoring|wantsome)@'i,
    qr'^(workathome|yesitsfree|your_friend|greatoffers)@'i,
    qr'^(inkjetplanet|marketopt|MakeMoney)\d*@'i,
);

#HASH lookup variant:
# NOTE: Perl operator qw splits its argument string by whitespace
# and produces a list. This means that addresses can not contain
# whitespace, and there is no provision for comments within the string.
# You can use the normal Perl list syntax if you have special requirements,
# e.g. map {...} ('one user@bla', '.second.com'), or use read_hash to read
# addresses from a file.
#

# a hash lookup table can be read from a file,
# one address per line, comments and empty lines are permitted:
#
# read_hash(\%whitelist_sender, '/var/amavis/whitelist_sender');

# ... or set directly:

# $whitelist_sender{''} = 1;  # don't spam-check MTA bounces

map { $whitelist_sender{lc($_)}=1 } (qw(
  cert-advisory-owner@cert.org
  owner-alert@iss.net
  slashdot@slashdot.org
  bugtraq@securityfocus.com
  NTBUGTRAQ@LISTSERV.NTBUGTRAQ.COM
  security-alerts@linuxsecurity.com
  amavis-user-admin@lists.sourceforge.net
  notification-return@lists.sophos.com
  mailman-announce-admin@python.org
  owner-postfix-users@postfix.org
  owner-postfix-announce@postfix.org
  owner-sendmail-announce@Lists.Sendmail.ORG
  owner-technews@postel.ACM.ORG
  lvs-users-admin@LinuxVirtualServer.org
  ietf-123-owner@loki.ietf.org
  cvs-commits-list-admin@gnome.org
  rt-users-admin@lists.fsck.com
  clp-request@comp.nus.edu.sg
  surveys-errors@lists.nua.ie
  emailNews@genomeweb.com
  owner-textbreakingnews@CNNIMAIL12.CNN.COM
  spamassassin-talk-admin@lists.sourceforge.net
  yahoo-dev-null@yahoo-inc.com
  returns.groups.yahoo.com
));
```

```
# ENVELOPE SENDER WHITELISTING / BLACKLISTING - PER-RECIPIENT

# The same semantics as for global white/blacklisting applies, but this
# time each recipient (or its domain, or subdomain, ...) can be given
# an individual lookup table for matching senders. The per-recipient lookups
# override the global lookups, which serve as a fallback default.

# Specify a two-level lookup table: the key for the outer table is recipient,
# and the result should be an inner lookup table (hash or ACL or RE),
# where the key used will be the sender.
#
#$per_recip_blacklist_sender_lookup_tables = {
#
'user1@my.example.com'=>new_RE(qr'^(inkjetplanet|marketopt|MakeMoney)\d*@'i),
# 'user2@my.example.com'=>[qw( spammer@d1.example,org .d2.example,org )],
#};
#$per_recip_whitelist_sender_lookup_tables = {
# 'user@my.example.com' => [qw( friend@example.org .other.example.org )],
# '.my1.example.com'    => [qw( !foe.other.example,org .other.example,org )],
# '.my2.example.com'    => read_hash('/var/amavis/my2-wl.dat'),
# 'abuse@' => { 'postmaster@'=>1,
#               'cert-advisory-owner@cert.org'=>1, 'owner-alert@iss.net'=>1 },
#};


#
# Section VI - Resource limits
#

# Sanity limit to the number of allowed recipients per SMTP transaction
# $smtpd_recipient_limit = 1000;  # (default is 1000)

# Resource limitations to protect against mail bombs (e.g. 42.zip)


# Maximum recursion level for extraction/decoding (0 or undef disables limit)
$MAXLEVELS = 14;        # (default is undef, no limit)

# Maximum number of extracted files (0 or undef disables the limit)
$MAXFILES = 1500;       # (default is undef, no limit)

# For the cumulative total of all decoded mail parts we set max storage size
# to defend against mail bombs. Even though parts may be deleted (replaced
# by decoded text) during decoding, the size they occupied is _not_ returned
# to the quota pool.
#
# Parameters to storage quota formula for unpacking/decoding/decompressing
#   Formula:
#     quota = max($MIN_EXPANSION_QUOTA,
#                 $mail_size*$MIN_EXPANSION_FACTOR,
#                 min($MAX_EXPANSION_QUOTA, $mail_size*$MAX_EXPANSION_FACTOR))
#   In plain words (later condition overrules previous ones):
#     allow MAX_EXPANSION_FACTOR times initial mail size,
#     but not more than MAX_EXPANSION_QUOTA,
#     but not less than MIN_EXPANSION_FACTOR times initial mail size,
#     but never less than MIN_EXPANSION_QUOTA
#
$MIN_EXPANSION_QUOTA =      100*1024;  # bytes  (default undef, not enforced)
```

```
$MAX_EXPANSION_QUOTA = 300*1024*1024;  # bytes  (default undef, not enforced)
$MIN_EXPANSION_FACTOR =   5;  # times original mail size  (must be specified)
$MAX_EXPANSION_FACTOR = 500;  # times original mail size  (must be specified)


#
# Section VII - External programs, virus scanners
#

# Specify a path string, which is a colon-separated string of directories
# (no trailing slashes!) to be assigned to the environment variable PATH
# and to serve for locating external programs below.

# NOTE: if $daemon_chroot_dir is nonempty, the directories will be
#       relative to the chroot directory specified;

$path = '/usr/local/sbin:/usr/local/bin:/usr/sbin:/sbin:/usr/bin:/bin';

# Specify one string or a search list of strings (first match wins).
# The string (or: each string in a list) may be an absolute path,
# or just a program name, to be located via $path;
# Empty string or undef (=default) disables the use of that external program.
# Optionally command arguments may be specified - only the first substring
# up to the whitespace is used for file searching.

$file   = 'file';   # file(1) utility; use 3.41 or later to avoid
vulnerability

$gzip   = 'gzip';
$bzip2  = 'bzip2';
$lzop   = 'lzop';
$uncompress = ['uncompress', 'gzip -d', 'zcat'];
$unfreeze   = ['unfreeze', 'freeze -d', 'melt', 'fcat'];
$arc        = ['nomarch', 'arc'];
$unarj      = ['arj', 'unarj'];  # both can extract, same options
$unrar      = ['rar', 'unrar'];  # both can extract, same options
$zoo    = 'zoo';
$lha    = 'lha';
$cpio   = 'cpio';   # comment out if cpio does not support GNU options


# SpamAssassin settings

# $sa_local_tests_only is passed to Mail::SpamAssassin::new as a value
# of the option local_tests_only. See Mail::SpamAssassin man page.
# If set to 1, no tests that require internet access will be performed.
#
$sa_local_tests_only = 1;   # (default: false)
#$sa_auto_whitelist = 1;    # turn on AWL (default: false)

$sa_mail_body_size_limit = 64*1024;  # don't waste time on SA if mail is
larger
                      # (less than 1% of spam is > 64k)
                      # default: undef, no limitations

# default values, can be overridden by more specific lookups, e.g. SQL
$sa_tag_level_deflt  = 3.0; # add spam info headers if at, or above that level
$sa_tag2_level_deflt = 5.0;
```

```
$sa_kill_level_deflt = $sa_tag2_level_deflt; # triggers spam evasive actions
                       # at or above that level: bounce/reject/drop,
                       # quarantine, and adding mail address extension
#
# The $sa_tag_level_deflt, $sa_tag2_level_deflt and $sa_kill_level_deflt
# may also be hashrefs to hash lookup tables, to make static per-recipient
# settings possible without having to resort to SQL or LDAP lookups.

# a quick reference:
#   tag_level  controls adding the X-Spam-Status and X-Spam-Level headers,
#   tag2_level controls adding 'X-Spam-Flag: YES', and editing Subject,
#   kill_level controls 'evasive actions' (reject, quarantine, extensions);
# it only makes sense to maintain the relationship:
# tag_level <= tag2_level <= kill_level

# string to prepend to Subject header field when message exceeds tag2 level
#$sa_spam_subject_tag = '***SPAM*** ';    # (defaults to undef, disables)
                        # (only seen when spam is not to be rejected
                        # and recipient is in local_domains*)

#$sa_spam_modifies_subj = 1; # may be a ref to a lookup table, default is true

# Example: modify Subject for all local recipients except user@example.com
#$sa_spam_modifies_subj = [qw( !user@example.com . )];


# @av_scanners is a list of n-tuples, where fields semantics is:
#  1. av scanner plain name, to be used in log and reports;
#  2. scanner program name; this string will be submitted to subroutine
#     find_external_programs(), which will try to find the full program
#     path name; if program is not found, this scanner is disabled.
#     Besides a simple string (full program path name or just the basename
#     to be looked for in PATH), this may be an array ref of alternative
#     program names or full paths - the first match in the list will be used;
#     As a special case for more complex scanners, this field may be
#     a subroutine reference, and the whole n-tuple is passed to it as args.
#  3. command arguments to be given to the scanner program;
#     a substring {} will be replaced by the directory name to be scanned,
#     i.e. "$tempdir/parts"
#  4. an array ref of av scanner exit status values, or a regexp (to be
#     matched against scanner output), indicating NO VIRUSES found;
#  5. an array ref of av scanner exit status values, or a regexp (to be
#     matched against scanner output), indicating VIRUSES WERE FOUND;
#     Note: the virus match prevails over a 'not found' match, so it is safe
#     even if 4. matches for viruses too;
#  6. a regexp (to be matched against scanner output), returning a list
#     of virus names found.
#  7. and 8.: (optional) subroutines to be executed before and after scanner
#     (e.g. to set environment or current directory);
#     see examples for these at KasperskyLab AVP and Sophos sweep.

# NOTES:
#
# - NOT DEFINING @av_scanners (e.g. setting it to empty list, or deleting the
#   whole assignment) TURNS OFF LOADING AND COMPILING OF THE ANTIVIRUS CODE
#   (which can be handy if all you want to do is spam scanning);
#
# - the order matters: although _all_ available entries from the list are
```

```
#   always tried regardless of their verdict, scanners are run in the order
#   specified: the report from the first one detecting a virus will be used
#   (providing virus names and scanner output); REARRANGE THE ORDER TO WILL;
#
# - it doesn't hurt to keep an unused command line scanner entry in the list
#   if the program can not be found; the path search is only performed once
#   during the program startup;
#
#   CORROLARY: to disable a scanner that _does_ exist on your system,
#   comment out its entry or use undef or '' as its program name/path
#   (second parameter). An example where this is almost a must: disable
#   Sophos 'sweep' if you have its daemonized version Sophie or SAVI-Perl
#   (same for Trophie/vscan, and clamd/clamscan), or if another unrelated
#   program happens to have a name matching one of the entries ('sweep'
#   again comes to mind);
#
# - it DOES HURT to keep unwanted entries which use INTERNAL SUBROUTINES
#   for interfacing (where the second parameter starts with \&).
#   Keeping such entry and not having a corresponding virus scanner daemon
#   causes an unnecessary connection attempt (which eventually times out,
#   but it wastes precious time). For this reason the daemonized entries
#   are commented in the distribution - just remove the '#' where needed.


@av_scanners = (

# ### http://www.vanja.com/tools/sophie/
# ['Sophie',
#   \&ask_daemon, ["{}/\n", '/var/run/sophie'],
#   qr/(?x)^ 0+ ( : | [\000\r\n]* $)/,  qr/(?x)^ 1 ( : | [\000\r\n]* $)/,
#   qr/(?x)^ [-+]? \d+ : (.*?) [\000\r\n]* $/ ],

# ### http://www.csupomona.edu/~henson/www/projects/SAVI-Perl/
# ['Sophos SAVI', \&sophos_savi ],

# ### http://clamav.elektrapro.com/
# ['Clam Antivirus-clamd',
#   \&ask_daemon, ["CONTSCAN {}\n", '/var/amavis/clamd'],
#   qr/\bOK$/, qr/\bFOUND$/,
#   qr/^.*?: (?!Infected Archive)(.*) FOUND$/ ],
# # NOTE: run clamd under the same user as amavisd,
# # match the socket name in clamav.conf to the socket name in this entry

# ### http://www.openantivirus.org/
# ['OpenAntiVirus ScannerDaemon (OAV)',
#   \&ask_daemon, ["SCAN {}\n", '127.0.0.1:8127'],
#   qr/^OK/, qr/^FOUND: /, qr/^FOUND: (.+)/ ],

# ### http://www.vanja.com/tools/trophie/
# ['Trophie',
#   \&ask_daemon, ["{}/\n", '/var/run/trophie'],
#   qr/(?x)^ 0+ ( : | [\000\r\n]* $)/,  qr/(?x)^ 1 ( : | [\000\r\n]* $)/,
#   qr/(?x)^ [-+]? \d+ : (.*?) [\000\r\n]* $/ ],

# ### http://www.f-prot.com/
 ['FRISK F-Prot Daemon',
   \&ask_daemon,
   ["GET {}/*?-dumb%20-archive HTTP/1.0\r\n\r\n",
```

```
     ['127.0.0.1:10200','127.0.0.1:10201','127.0.0.1:10202',
      '127.0.0.1:10203','127.0.0.1:10204'] ],
    qr/(?i)<summary[^>]*>clean<\/summary>/,
    qr/(?i)<summary[^>]*>infected<\/summary>/,
    qr/(?i)<name>(.+)<\/name>/ ],

  ['KasperskyLab AntiViral Toolkit Pro (AVP)', ['avp','kavscanner'],
    '-* -P -B -Y -O- {}', [0,3,8], [2,4],    # any use for -A -K   ?
    qr/infected: (.+)/,
    sub {chdir('/opt/AVP') or die "Can't chdir to AVP: $!"},
    sub {chdir($TEMPBASE) or die "Can't chdir back to $TEMPBASE $!"},
  ],

# NOTE: not sure which entry suits which kavscanner version
# ['KasperskyLab kavscanner 4.5', ['/opt/kav/bin/kavscanner','kavscanner'],
#   '-i1 -xp {}', [0], [5,20,21,25],
#   qr/(?:CURED|INFECTED|CUREFAILED|WARNING|SUSPICION) (.*)/ ,
#   sub {chdir('/opt/kav/bin') or die "Can't chdir to kav: $!"},
#   sub {chdir($TEMPBASE) or die "Can't chdir back to $TEMPBASE $!"},
# ],

  ['KasperskyLab AVPDaemonClient',
    [ '/opt/AVP/kavdaemon',       'kavdaemon',
      '/opt/AVP/AvpDaemonClient', 'AvpDaemonClient',
      '/opt/AVP/AvpTeamDream',    'AvpTeamDream',
      '/opt/AVP/avpdc', 'avpdc' ],
    '{}', [0,8], [3,4,5,6], qr/infected: ([^\r\n]+)/ ],
    # change the startup-script in /etc/init.d/kavd to:
    #   DPARMS="-I0 -Y -* /var/amavis"
    # adjusting /var/amavis above to match your $TEMPBASE.
    # NOTE: cd /opt/AVP/DaemonClients; configure; cd Sample; make
    #       cp AvpDaemonClient /opt/AVP/

  ### http://www.hbedv.com/ or http://www.centralcommand.com/
  ['H+BEDV AntiVir or CentralCommand Vexira Antivirus',
    ['antivir','vexira'],
    '--allfiles -noboot -nombr -rs -s -z {}', [0], qr/ALERT:|VIRUS:/,
    qr/(?x)^\s* (?: ALERT: \s* (?: \[ | [^']* ' ) |
         (?i) VIRUS:\ .*?\ virus\ '?) ( [^\]\s']+ )/ ],
    # NOTE: remove the -z if you only have a demo version

  ### http://www.commandsoftware.com/
  ['Command AntiVirus for Linux', 'csav',
    '-all -archive -packed {}', [50], [51,52,53],
    qr/Infection: (.+)/ ],

  ### http://www.symantec.com/
  ['Symantec CarrierScan via Symantec CommandLineScanner',
    ['cscmdline','savsecls'],
    '-a scan -i 1 -v -s 127.0.0.1:7777 {}',
    qr/Files Infected: 0/, qr/^Infected: /,
    qr/Info:\s+(.+)/ ],

  ### http://drweb.imshop.de/
  ['DrWeb Antivirus for Linux/FreeBSD/Solaris', 'drweb',
    '-al -ar -fm -go -ha -ml -ot -sd -up {}',
    [0], [1], sub {('no-name')} ],
```

```
  ### http://www.f-secure.com/products/anti-virus/
  ['F-Secure Antivirus', 'fsav',
    '--dumb --archive {}', [0], [3,8],
    qr/(?:infection|Infected): (.+)/ ],

  ['CAI InoculateIT', 'inocucmd',
    '-sec -nex {}', [0], [100],
    qr/was infected by virus (.+)/ ],

  ['MkS_Vir for Linux (beta)', ['mks32','mks'],
    '-s {}/*', [0], [1,2],
    qr/--[ \t]*(.+)/ ],

  ['MkS_Vir daemon',
    'mksscan', '-s -q {}', [0], [1..7],
    qr/^... (\S+)/ ],

  ### http://www.nod32.com/
  ['ESET Software NOD32', 'nod32',
    '-all -subdir+ {}', [0], [1,2],
    qr/^.+? - (.+?)\s*(?:backdoor|joke|trojan|virus|worm)/ ],

  ### http://www.nod32.com/
  ['ESET Software NOD32 - Client/Server Version', 'nod32cli',
    '-a -r -d recurse --heur standard {}', [0], [10,11],
    qr/^\S+\s+infected:\s+(.+)/ ],

  ### http://www.norman.com/products_nvc.shtml
  ['Norman Virus Control v5 / Linux', 'nvccmd',
    '-c -l:0 -s -u {}', [0], [1],
    qr/(?i).* virus in .* -> \'(.+)\'/ ],

  ### http://www.pandasoftware.com/
  ['Panda Antivirus for Linux', ['pavcl'],
    '-aut -aex -heu -cmp -nbr -nor -nso -eng {}',
    qr/Number of files infected[ \.]*: 0(?!\d)/,
    qr/Number of files infected[ \.]*: 0*[1-9]/,
    qr/Found virus :\s*(\S+)/ ],

# Check your RAV license terms before fiddling with the following two lines!
# ['GeCAD RAV AntiVirus 8', 'ravav',
#   '--all --archive --mail {}', [1], [2,3,4,5], qr/Infected: (.+)/ ],
# # NOTE: the command line switches changed with scan engine 8.5 !
# # (btw, assigning stdin to /dev/null causes RAV to fail)

  ### http://www.nai.com/
  ['NAI McAfee AntiVirus (uvscan)', 'uvscan',
    '--secure -rv --summary --noboot {}', [0], [13],
    qr/(?x) Found (?:
        \ the\ (.+)\ (?:virus|trojan)  |
        \ (?:virus|trojan)\ or\ variant\ ([^ ]+)  |
        :\ (.+)\ NOT\ a\ virus)/,
  # sub {$ENV{LD_PRELOAD}='/lib/libc.so.6'},
  ],
  # NOTE with RH9: force the dynamic linker to look at /lib/libc.so.6 before
  # anything else by setting environment variable LD_PRELOAD=/lib/libc.so.6

  ### http://www.virusbuster.hu/en/
```

```
  ['VirusBuster', ['vbuster', 'vbengcl'],
    # VirusBuster Ltd. does not support the daemon version for the workstation
    # engine (vbuster-eng-1.12-linux-i386-libc6.tgz) any longer. The names of
    # binaries, some parameters AND return codes (from 3 to 1) changed.
    "{} -ss -i '*' -log=$MYHOME/vbuster.log", [0], [1],
    qr/: '(.*)' - Virus/ ],

# ### http://www.virusbuster.hu/en/
# ['VirusBuster (Client + Daemon)', 'vbengd',
#   # HINT: for an infected file it returns always 3,
#   # although the man-page tells a different story
#   '-f -log scandir {}', [0], [3],
#   qr/Virus found = (.*);/ ],

  ### http://www.cyber.com/
  ['CyberSoft VFind', 'vfind',
    '--vexit {}', [0], [23], qr/##==>>>> VIRUS ID: CVDL (.+)/,
  # sub {$ENV{VSTK_HOME}='/usr/lib/vstk'},
  ],

  ### http://www.ikarus-software.com/
  ['Ikarus AntiVirus for Linux', 'ikarus',
    '{}', [0], [40], qr/Signature (.+) found/ ],

  ### http://www.bitdefender.com/
  ['BitDefender', 'bdc',
    '--all --arc {}', qr/^Infected files *:0(?!\d)/,
    qr/^(?:Infected files|Identified viruses|Suspect files) *:0*[1-9]/,
    qr/(?:suspected|infected): (.*)\033/ ],

);

# If no virus scanners from the @av_scanners list produce 'clean' nor
# 'infected' status (e.g. they all fail to run or the list is empty),
# then _all_ scanners from the @av_scanners_backup list are tried.
# When there are both daemonized and command-line scanners available,
# it is customary to place slower command-line scanners in the
# @av_scanners_backup list. The default choice is somewhat arbitrary,
# move entries from one list to another as desired.

@av_scanners_backup = (

  ### http://clamav.elektrapro.com/
  ['Clam Antivirus - clamscan', 'clamscan',
    '--stdout --disable-summary -r {}', [0], [1],
    qr/^.*?: (?!Infected Archive)(.*) FOUND$/ ],

  ### http://www.f-prot.com/
  ['FRISK F-Prot Antivirus', ['f-prot','f-prot.sh'],
    '-dumb -archive -packed {}', [0,8], [3,6],
    qr/Infection: (.+)/ ],

  ### http://www.trendmicro.com/
  ['Trend Micro FileScanner', ['/etc/iscan/vscan','vscan'],
    '-a {}', [0], qr/Found virus/, qr/Found virus (.+) in/ ],

# Commented out because the name 'sweep' clashes with the Debian package of
```

```
# the same name. Make sure the correct sweep is found in the path when
enabling
#
# ### http://www.sophos.com/
# ['Sophos Anti Virus (sweep)', 'sweep',
#   '-nb -f -all -rec -ss -sc -archive {}',
#   [0,2], qr/Virus .*? found/,
#   qr/^>>> Virus(?:(?: fragment)? '?(.+?)'? found)/,
# # sub {$ENV{SAV_IDE}='/usr/local/sav'},
# ],

);


#
# Section VIII - Debugging
#

# The most useful debugging tool is to run amavisd-new non-detached
# from a terminal window:  # amavisd debug

# Some more refined approaches:

# If sender matches ACL, turn debugging fully up, just for this one message
#@debug_sender_acl = ( "test-sender\@$mydomain" );
#@debug_sender_acl = qw( debug@example.com );

# May be useful along with @debug_sender_acl:
# Prevent all decoded originals being deleted (replaced by decoded part)
#$keep_decoded_original_re = new_RE( qr/.*/ );

# Turn on SpamAssassin debugging (output to STDERR, use with 'amavisd debug')
#$sa_debug = 1;            # defaults to false


#-------------
1;  # insure a defined return
```

Konfigurationsdatei „qpopper“

(abgelegt unter “/etc/xinetd.d/qpopper”)

```
#
# qpopper - pop3 mail daemon
#
service pop3
{
      disable          = no
      socket_type      = stream
      protocol         = tcp
      wait             = no
      user             = root
      server           = /usr/sbin/popper
      server_args      = -s
      flags            = IPv4
}
```