Postfix Server

Projekt im Fach
Mehrplatzbetriebssysteme

Von:
Bettina Dutschmann
Stefanie Uhrig

Inhaltsverzeichnis

Kapitel 6

Kapitel 7

Kapitel 8

Kapitel 9

Anhang

Quelle

Aufgabenstellung

- E-Mailserver mit Postfix erstellen
- Das versenden und empfangen von E-Mails soll durch einen Windows-Client und Linux-Client möglich sein.
- Es sollen 2 verschiedene E-Mailprogramme verwendet werden.
- Authentifizierung beim empfangen und versenden von E-Mails
- Dokumentation über das Projekt

Anforderung von Soft- und Hardware

- Suse Linux Software 9.2
- Postfix
- TLS-Paket von Postfix installieren
- 2 verschiede E-Mailprogramme
- 2x E-Mailclient
- 1x Postfixserver
- 1x HUB
- 3x Netzwerkkabel

Probleme die während unserem Projekt auftraten.

Das erste Problem das bei uns auftauchte war, wie kann ich eine Verbindung zu dem Linux Client herstellen? Nach einigen Überlegungen stellten wir fest, das der Client sich in einem anderen Netzwerk befand wie unser Postfixserver. Das Problem lösten wir, indem wir die Netzwerkadresse so änderten das sich der Linux Client im gleichen Netzwerk befand wie unser Postfixserver.
Nachdem wir einen Benutzer auf den Linux Client eingerichtet hatten und einen Benutzer auf den Postfix Server, versuchten wir eine erste E-Mail zu versenden. Die ersten versuche schlugen fehl. Das Problem lag darin, das wir den Benutzer Stefanie auf dem Postfix-Server groß geschrieben hatten und auf dem E-Mailclient klein und da Linux auf die Groß- und Kleinschreibung achtet hatte er den Benutzer nicht zuordnen können.

Nun standen wir vor der Aufgaben, wie ein Benutzer eine E-Mail sicher empfangen kann. Das war eine sehr schwierige Aufgabe. Das Problem konnten wir leider auch nicht vollständig lösen. Wir kamen aber zu dem Entschluss das es nur mit einer SMTP Authentifizierung möglich ist. Dafür mussten wir ein TLS Zertifikat erstellen und einige Konfigurationen in der main.cf vornehmen. Das Zertifikat mussten wir erstellen damit wir die Passwörter verschlüsselt zum Server verschicken können.
Das Problem das wir immer noch haben ist das unser Postfix Server die SMTP Authentifizierung nicht unterstützt. Nach stundenlangen suchen im Internet haben wir herausgefunden das wir ein Paket von Postfix nicht Installiert hatten. Wir klammerten alle Parameter die in der main.cf mit smtp zu tun hatten aus und probierten erneuert, eine E-Mail zu empfangen. Nach dem wir Postfix neu gestartet hatten konnten wir wieder E-Mails empfangen. Nach langen suchen im Internet fanden wir eine Seite die beschrieb das wir erst das TLS-Paket von Postfix installieren müssen, damit Postfix

TLS akzeptiert. Wir installierten alle Pakete, die irgendetwas mit TLS zu tun hatten. Danach überprüften wir laut Postfixbuch alle TLS Parameter in der main.cf.

Wir überprüften durch die Eingabe von tail –f /var/log/warr ob Postfix fehlerfrei läuft. Wir stellten leider fest das Postfix mit einigen Parameter Schwierigkeiten hatte. Nach Überprüfung von alle Parameter, fanden wir heraus das es genau die Parameter in der main.cf sind, die wir wegen Fehlersuche eingerückt hatten.

Das nächste Problem war unser Zertifikat. Da wir immer noch keine verschlüsselten E-Mails verschicken konnten, haben wir, in die Datei nachgesehen, indem die Fehlermeldungen der E-Mails stehen. Diese Informationen haben uns etwas weitergeholfen. Wir bekamen als Information das Postfix nicht auf die Datei indem die Schlüssel für das Zertifikat liegen, zugreifen kann. Der Grund dafür war, das die Dateien in einem anderen Verzeichnis lagen, als wir sie in der main.cf angegeben hatten.

Wir konnten zwar jetzt Email per TLS verschicken, aber die Authentifizierung funktioniert immer noch nicht. Das war ein sehr komplexes Thema. Eine genau Anleitung dafür gab es leider auch nicht. Der Fehler lag darin, das Postfix auf die Konfigurationsdatei in der Benutzer und Passwörter stehen nicht zugreifen konnte. Und zwar aus folgenden Grund: Postfix läuft in einer geschützten Umgebung („chroot Umgebung“). Sobald Postfix gestartet wir, läuft es abgeschottet vom Rest des Systems. Es ist zwar ein guter Schutz gegen Hacker, aber schlecht, wenn Postfix auf ein anderes Programm zugreifen wie in unserem Fall auf den saslauthd. Es ist uns irgendwie gelungen durch ständiges probieren zu ermöglichen das Postfix trotzdem den Zugriff auf den saslauthd hat. Wir mussten die Konfigurationsdatei sasl2 in das Verzeichnis kopieren, wo auch die restlichen Konfigurationsdatei von Postfix liegen (etc/postfix) sobald ein smtpd.conf vorhanden ist, liest Posftix dessen Einstellungen ein und weiß, wie es mit dem saslauthd kommunizieren kann

1 Postfix

1.1 Die Geschichte

Postfix wurde 1998 von Wietse Venema entwickelt und zunächst unter dem Namen Vmailer veröffentlicht. Im Rahmen einer markenrechtlichen Prüfung fand man heraus, dass dieser Name einer anderen Marke sehr ähnlich klingt und vergab den Namen IBM Secure Mailer + Postfix.

Dr. Wietse Venema ist ein niederländischer Programmierer und Physiker, der vor allem durch sein Mailprogramm Postfix bekannt ist.

1.2 Was ist Postfix ?

Postfix ist ein so genannter MTA, ein Mail Transfer Agent, er kümmert sich um den Versand und Empfang von Email. Er nimmt die Emails von den Nutzern im Netzwerk an, verteilt sie im eigenen Netzwerk oder übergibt sie an einen anderen MTA. Des weiteren kann er auch Emails außerhalb des eigenen Netzes annehmen und weiterleiten. Er ist quasi der Postbote; ein Tux mit einem Rucksack, der durch das Netz flitzt.

Postfix ist ein sehr sicheres und ziemlich einfach zu administrierendes Programm. Postfix ist modular aufgebaut und besteht aus vielen kleinen Programmen.

2. Was ist MTA – Der Posteingang des Mailservers?

Mailserver wie Postfix, qmail, exim, sendmail usw. sind allein für den Transport zuständig. Was die Mails enthalten, wie sie geschrieben oder gelesen werden, muss sie nicht interessieren. Der Post ist es ja auch egal, wie ein Brief zustande gekommen ist oder was darin steht, Hauptsache er gelangt ans Ziel.

2.1 Was ist MDA – Mail Delivery Agent?

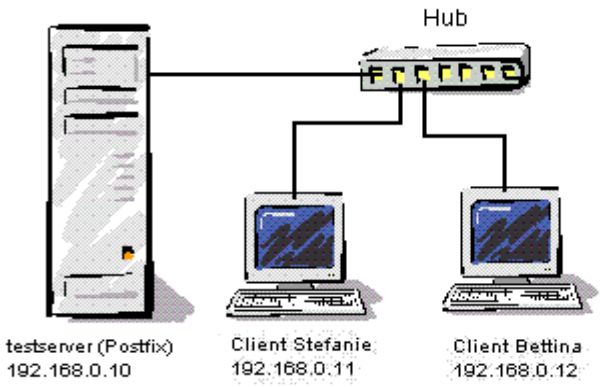
Der MDA ist ein kleines, in der Regel wenig auffälliges Programm, das vom MTA aufgerufen wird, um die E-Mails auf der Festplatte des Servers in die Mailordner zu speichern. Hier bietet sich die Möglichkeit, durch einen eigenen MDA E-Mails besonders abzuspeichern, z.B. in einer Datenbankumgebung oder nach einem vorherigen Anti-Viren-Check. In aller Regel wird man sich um einen MDA aber nicht kümmern müssen. Bei Postfix ist das Programm lokal dabei, das alles ohne weitere Konfiguration erledigt. Postfix kümmert sich um dessen Aufruf.

2.2 Was ist MUA – Mail User Agent?

MUA ist ein Mailprogramm des Benutzers, also Mailreader wie The Bat, Kmail, Netscape, Outlook, Thunderbird usw. Sie regeln die Übergabe geschriebener Mails per SMTP an einen MTA oder fragen Postfächer auf einem Server per POP3/IMAP ab. Sie sind es auch, die den Inhalt der Mail zu verwalten haben.

Die generelle Aufgabe eines Hubs besteht darin, als aktive Komponente sternförmige LAN-Verteilung und LAN-Konzentrationen aufzubauen. Ein Hub ist funktions-technisch in der OSI-Schicht-1 angesiedelt, D.h. der Hub kann also nur physikalische Signale empfangen und weiterleiten. Jedes empfangene Paket wird an jeden Port des Hubs gesendet, was einem Broadcast der einzelnen Pakete gleich kommt. Da der Hub bei der Weiterleitung eines jeden Paketes selbst Broadcasts erzeugt, hat er demzufolge auch keinerlei Mechanismen, um die Ausbreitung von Broadcasts zu kontrollieren.

2.3 Der Aufbau



Da 2 PCs im Netzwerk Ihren Dienst erledigen, so war die Anschaffung eines Hubs nicht mehr zu vermeiden. Wir mussten eine Vernetzung über Hubs gegenüber der Direktverbindung vorziehen, weil bei der Direktverbindung ständig die Verbindung zwischen Client und Server verloren ging.

3. Installationen

3.1 Postfix Installation

Postfix musste nicht installiert werden, da er als fester Bestandteil der Grundinstallation von Suse Linux 9.2 automatisch mitinstalliert wird.

3.2 Was ist Qpopper (POP3)?

Pop3 (Post Office Protokoll Version 3) ist ein Übertragungsprotokoll, über welches ein Client E-Mails von einem E-Mail Server abholen kann. Pop3 ist ein ASCII-Protokoll, wobei die Steuerung der Datenübertragung durch Kommandos geschieht, die standardmäßig an den Port 110 geschickt werden. Eine ständige Verbindung zum Mailserver ist bei Pop3 nicht notwendig. Die Verbindung wird bei Bedarf vom Client zum Server erzeugt und danach wieder beendet.

Pop3 ist in der Funktionalität sehr beschränkt und erlaubt nur das Abholen und Löschen von E-Mails am E-Mail Server. Als Gegenstück zu Pop3 zum Versenden von E-Mails ist üblicherweise in Clients und Servern das SMTP- bzw.- ESMTP Protokoll implementiert. Pop3 bietet optional die Authentifizierung über Benutzernamen und Passwort. Wird diese Möglichkeit genutzt, so wird das Passwort im Klartext übertragen.

3.2.1 Installation Qpopper

Qpopper ist standardmäßig nicht in der Grundinstallation von Suse Linux 9.2 enthalten, jedoch im umfangreichen Installationspaket integriert. Um Qpopper nachzuinstallieren sind wir wie folgt vorgegangen:

Als erstes haben wir das Konfigurationsprogramm YAST2 gestartet.

Danach haben wir den Menüpunkt „Software“ mit dem Untermenüpunkt „Software installieren oder löschen“ gewählt.

Im folgenden Fenster suchen wir mit dem dafür vorgesehenen Feld gezielt nach dem zu installierendem Paket Qpopper. Wird uns dieses nach erfolgreicher Suche angezeigt, so muss das Programm nur noch durch ein Häkchen in der Checkbox und durch das bestätigen von dem Button „Akzeptieren“ zur Installation freigegeben werden. Das Installationsprogramm weißt uns zusätzlich noch darauf hin, dass weitere Programme in Verbindung mit Qpopper installiert werden müssen. Dies geschieht ganz automatisch. Durch eine weitere Bestätigung des Buttons „Akzeptieren“ wird die Installation fortgesetzt.

3.2.2 Konfiguration Qpopper

Im Verzeichnis „/etc/xinetd.d“ werden die letzten Einstellungen zur Aktivierung vorgenommen. Dazu haben wir lediglich den Eintrag „disabled =Yes“ auf „disabled = no“ ändern.

4. Grundeinstellungen und Grundinformation von Postfix

4.1 Benutzer anlegen und Passwortvergabe

Als erstes haben wir jeden von uns mit folgendem Befehl als Benutzer angelegt.
Useradd –m name

Danach haben wir jedem von uns ein Passwort vergeben
Passwd name

4.2 Konfigurationsdateien von Postfix

Die Konfiguration von Postfix ist zum größten Teil im Verzeichnis etc/postfix zu finden. Dort stehen die Konfigurationsdateien für den Postmasterprozess (master.cf) und für Postfix im allgemeinen (main.cf).

4.3 Konfigurationsdatei master.cf

Das Postfix Paket selbst bringt ausführlich kommentierte Konfigurationsdateien mit. Der größte Teil der master.cf besteht aus Erläuterungen. Die Konfiguration erfolgt zeilenweise in einer Tabellenähnlichen Datenstruktur. Jede Zeile definiert hier eine Komponente des Postfixsystems, die vom Masterprozess gestartet werden kann. Die einzelnen Spalten steuern wie und wann der zentrale Daemon die jeweiligen Prozesse startet oder stoppt.

Eine Konfigurationszeile der Konfigurationsdatei beginnt immer in der ersten Spalte. Nach einem Zeilenumbruch wird sie nach einem Whitespace fortgesetzt. Es werden also keine \ benötigt um Zeilenumbrüche zu maskieren.

Unsere master.cf sieht wie folgt aus:

```
Linux:~# joe /etc/postfix/master.cf
# service    type        private     Unpriv      chroot      wakeup      maxproc     command + args
#                        (yes)       (yes)       (yes)       (never)     (50)
# ==========================================================================================
smtp         inet        n                       y                                   smtpd
smtps        inet        n                       n                                   smtpd -o
```

Bedeutung der einzelnen Spalten

Jede Konfigurationszeile ist in maximal 8 Spalten unterteilt. Die Spalten werden durch Whitespace voneinander getrennt. In jeder Zeile haben die einzelnen Spalten dieselbe Bedeutung für die jeweilige Teilkomponente. Einige der Spalten sind mit einem Defaultwert vorbelegt. Wenn in dieser Spalte ein "-" eingetragen wird kommt der Defaultwert zum tragen.

- service and type

Benennt den „Dienst“ und wie er mit dem Rest des Systems kommuniziert. Dabei steht „inet“ für ein TCP-Socket, „unix“ für ein Unix-Domain-Socket (zur Kommunikation auf demselben Rechner) und „fifo“ für einen Fifo. Nur inet-Dienste sind potentiell von anderen Rechnern aus zugänglich. Dienstnamen dürfen mehrmals auftreten solange der Transporttyp verschieden ist.

- privat

Regelt, ob der Zugriff auf den Dienst nur für das Postfix-System möglich sein soll. Dies wird über Linux-Datei Zugriffsrechte geregelt; „inet-Dienste“ können nicht privat sein, da TCP keine Zugriffsrechte kennt. Wenn nichts anderes angegeben wurde, ist ein Dienst privat.

- unprivilegd

Regelt, ob der Dienst als Postfix-Benutzer laufen kann (im Gegensatz zu root). Normalerweise können alle Komponenten des Postfix-Systems ohne root-Privilegien laufen; die einzigen Ausnahmen (neben dem postmaster) sind die Zustell-mechanismen local, virtual und pipe, die in der Lage sein müssen, die Identität beliebiger Benutzer anzunehmen.

- chroot

Regelt, ob der Dienst per chroot im /var/spool/postfix-Verzeichnis eingesperrt werden kann. Dies begrenzt potentiell den Schaden, wenn Postfix über eine Sicherheitslücke kompromittiert würde und ein Angreifer beliebigen Code ausführen könnte. Chroot ist für die meisten Postfix-Komponenten möglich, mit denselben Ausnahmen wie bei unprivilegd.

- wakeup
regelt ob der Dienst nach Sekunden aufgeweckt wird. Der Wert 0 steht für „gar nicht aufwecken“. Ein eingehängtes Fragezeichen bedeutet, dass der Dienst nur aufgeweckt werden soll, wenn er wirklich benutzt wird.

- maxprocs
Die maximale Anzahl von Prozessen, die gleichzeitig für diesen Dienst gestartet werden können. Der Standartwert ist durch den Postfix-Parameter default_process_limit gegeben. 0 steht für keine Begrenzung.

- command + args
Das Programm, das ausgeführt werden soll, mit seinen Argumenten. Programme werden im Programmverzeichnis von Postfix (typischerweise /usr/lib/postfix) gesucht.

Zum Schluss muss die Konfiguration durch das Kommando „etc/inet.de/xinetd restart“ neu gestartet werden.

4.4 Konfigurationsdatei main.cf (/etc/postfix/main.cf)

Die Datei main.cf konfiguriert das Laufzeitverhalten von Postfix. Es erledigt u.a. die Steuerung der Adressmaskierung, z.B welche Domainnamen wo verwendet werden. Die Konfiguration erfolgt über Parameter, die jeweils einen Wert oder eine Liste von Werten zugewiesen bekommen. Mit dem Pfad /etc/postfix/main.cf kommen wir in die Konfigurationsdatei hinein.

Wir werden hier nicht alle Parameter der Datei erklären, sondern nur die wichtigsten, die wir für unseren Server gebraucht bzw. abgeändert haben.

Parameter:

Setzt den Pfad zu den Programmen, die zu Postfix gehören.
Command_directory= /usr/sbin

gibt an wo sich die Daemonprogramme befinden. Hierzu gehören auch die Programme, die in main.cf aufgeführt werden. Der Besitzer dieses Verzeichnisses muss Root sein.
Daemon_directory = /usr/lib/postfix

Gibt den SMTP-Server Antwort Code an, nach dem im Falle einer nicht zustellbaren Mail gehandelt werden soll.
Hier gibt es 2 Möglichkeiten:
Code = 550 (reject_mail – Mail direkt an den Absender zurücksenden)
Code = 450 (try again later – Mail zurückstellen und später die Zustellung versuchen.
Unknown_local_recipient_reject_code = 450

An diesem Punkt wird die Netzwerkadresse eingetragen, die Postfix als vertrauenswürdig einstufen soll.

Unsere Berechnung für die Netzwerkadresse:

IP- Adresse	192.168.0.10
Subnetmaske	255.255.255.0
Netz ID	192.168.0.0

Dies bedeutet, das alle Clients die mit der Netzwerkadresse 192.168. anfangen mit unserem Postfix Server konfigurieren können.
Mynetworks = 192.168.0.0/24, 172.0.0.0/8

Pfad zum Spoll-directory
Mail_spool_directory = /var/mail

legt fest welchen Namen der Mailserver haben soll. Gibt man hier keinen Namen an, probiert Postfix den Namen mit Hilfe von gethostname() selber heraus zu finden.
Myhostname = testserver.pfalz.de

Pfad zu den Postfix-Programmen
Programm_directory = /usr/lib/postfix

legt fest wofür der Server zuständig ist; d.h. für welche Domains die Emails angenommen werden.
Mydestination = $myhostname, localhost.$mydomain

legt fest wohin Postfix, die Emails, die er selber nicht ausliefern kann (also alles außerhalb der Domain), senden soll. Wir wollen sie zu dem Mailserver unseres Providers senden, und zwar auf Port 25 (Standart SMTP- Port).
Relayhost = [mail.pfalz.de]

legt fest welche Alias-Datenbank verwendet werden soll und welches Format sie hat.
Alias_maps = /hash:/etc/aliases

Standarteinstellung. IP Nummern, auf denen Postfix einen Port öffnet und Verbindungen annimmt.
Inet_interfaces = all

5. Allgemeines zu SMTP (Simple Mail Transfer Protocol)

SMTP - Simple Mail Transfer Protocol ist für die Übertragung von E-Mails zuständig. SMTP ist unabhängig vom Netzwerkprotokoll. In der Regel wird für die Übertragung das Transportprotokoll TCP verwendet. Die Kommunikation erfolgt über den Port 25.

Ein Benutzer wird zumeist vom Ablauf des SMTP-Protokolls nichts mitbekommen, da dies sein Mailprogramm im Hintergrund für ihn erledigt. Dieses Programm verbindet sich zu einem SMTP-Server (Mail Transfer Agent, MTA), der die Mail zum Empfänger weiterleitet.

SMTP setzt voraus, dass eine Übertragung vom Sender initiiert wird. Aus diesem Grund wird es nicht dazu benutzt, eine Mail von einem Server auf den Arbeitsplatzrechner zu übertragen. Dazu werden Post Office Protokolle wie das POP3-Protokoll, das IMAP-Protokoll oder andere verwendet.

Da SMTP ein sehr einfaches Protokoll ist, kann man unter den gängigen Betriebssystemen mit den Kommandozeilen- Programm telnet selbst eine E-Mail von Hand verschicken. Absender- und Empfängeradressen sind frei wählbar. Es können sich sogar die Adressen im *Mail From*- und *RCPT to*- Kommando von den Adressen im *From:* -und *To:* -Mailheader unterscheiden.

Der Server bietet verschiedene Authentifizierungsmechanismen an, wovon sich der Client einen aussucht. Der Server stellt dem Client daraufhin je nach Mechanismus einen Challenge oder weist ihn zum Fortfahren an.

5.1 SMTP AUTH (Eingehende Verbindung)

Diese Konfiguration dient dazu, die eingehenden Emails, also alle Emails die Postfix annimmt, zu überprüfen. Es ist erstmal egal, ob die Emails lokal verteilt oder an den Relayhost geliefert werden. Der Server soll so konfiguriert werden, dass er ohne Authentifizierung keine Emails mehr annimmt.

5.2 SMTP - Authentifizierung (Postfix lernt SASL)

Die Benutzer holen eMails via POP3 ab. Aus Sicherheitsgründen erlaubt Postfix nur den Benutzer, die in Mynetworks definiert wurden, den Versand und den Empfang von eMails. Würde man jedem erlauben, über den eMail-Server eMails an jede mögliche eMail-Adresse zu versenden, würden wir damit einen sogenanntes Offenes Relais haben. Also müssen wird die Benutzer zur Authentifizierung zwingen indem sie ihren Benutzernamen und Passwort eingeben müssen. Wenn diese Daten korrekt sind, kann der Benutzer seine eMails versenden oder empfangen.

Der Server bietet verschiedene Authentifizierungsmechanismen an, wovon sich der Client einen aussucht. Der Server stellt dem Client daraufhin je nach Mechanismus einen Challenge oder weist ihn zum Fortfahren an.

5.3 SASL (Simple Authentication and Security Layer) in Postfix konfigurieren (/etc/postfix/main.cf)

SASL dient zur Authentifizierung eines Clients, damit dieser Emails versenden kann. In unserem Fall werden die Daten verschlüsselt übertragen. Erwähnenswert ist, dass die Authentifizierung erst nach Aufbau einer verschlüsselten Verbindung stattfindet.

Hier legen wir fest, daß ein Benutzer sich per SASL authentifizieren muss.
smtpd_sasl_auth_enable = yes

Dieser Eintrag MUSS leer bleiben, hierzu kommen wir aber später, da das nicht-setzen diverse Probleme mit sich bringt.
smtpd_sasl_local_domain = pfalz.de

Hier legen wir lediglich fest, daß kein anonymer Emailversand akzeptiert werden soll.
smtpd_sasl_security_options = noanonymous

Workaround für ältere Clients und Outlook
broken_sasl_auth_clients = yes

5.4 Konfiguration von saslauthd

Postfix läuft in einer chroot Umgebung, deshalb bedarf es einer speziellen Konfiguration des saslauthd´s, damit Postfix darauf zugreifen kann. Die entsprechenden Dateien haben wir deswegen unbedingt innerhalb des Postfix chroot anlegen müssen (/var/spool/postfix).
Parallel dazu haben wir in der /etc/postfix/master.cf überprüft, ob Postfix überhaupt in einer chroot Umgebung läuft. Dafür haben wir uns die erste Zeile in der master.cf betrachtet, die wie folgt aussehen muss:

Smpt inet n - **y** - 100 smptd

Bei uns steht an der fünften stelle ein y, das heißt Postfix läuft in einer chroot Umgebung.

Als erstes passten wir die Cyrus SASL Konfiguration an, die sich am Anfang in der Datei /usr/lib/sasl2/smtpd.conf befindet. Die haben wir aber später in ein anderes Verzeichnis kopierten, damit Postfix darauf zugreifen kann. In dieser Datei steht drin, woher SASL die Usernamen und Paßwörter beziehen soll.

Konfigurationsdatei smtpd.conf anpassen

In der ersten Zeile befindet sich der Pfad den Postfix innerhalb de Change Chroot Umgebung sieht.

Die zweite Zeile legt fest, welche Authentifizierungsmethoden zur Verfügung stehen. Die meisten Client können z.B. plaintext, Outlook verwendet. Wir verwenden aber login. Einige Clients beherrschen noch andere Verfahren, wie z. B. CRAM-MD5.

Die letzte Zeile legt die Methode fest, mit der Postfix die Passwörter des AUTH Kommandos überprüfen soll. In unserem Fall über saslauthd. Er ist Bestandteil des sasl2-bin-Packets und kommuniziert mit Postfix über einen Socket.

/etc/postfix/sasl2/smpdt

```
Saslauthd_path: /var/run/sasl2/mux
Mech_list: PLAIN LOGIN
Pwcheck_method: saslauthd
```

Die smtpd.conf muss in /etc/postfix/sasl2 kopiert werden, da Postfix beim starten des Systems als erstes in etc die Konfigurationsdateien (smtpd.conf) sucht und einliest.

Damit Postfix auf saslauthd zugreifen kann, haben wir die Konfigurationsdatei smtpd.conf nach /etc/postfix/sasl2/ kopiert.

Jetzt haben wir unsere User angelegt, dazu diente das Programm saslpasswd: Durch die Eingabe saslpasswd2 -a smtpd -c stefanie wird in der sasldblistusers ein neuer User angelegt.

Des weiteren haben wir folgendes Verzeichnis in der Postfix Chroot Umgebung angelegt, damit Saslauthd und Postfix miteinander kommunizieren können.
mkdir –p /var/spool/postfix/var/run/sasl2

Weil saslauthd außerhalb der change chroot Umgebung läuft, muss der Pfad den saslauthd durchlaufen muss genau angegeben werden.

Das haben wir erreicht, indem wir in der Datei **/etc/init.d/saslauthd** durch den Aufruf um folgenden Parameter -m /var/spool/postfix/var/run/sasl2/ erweitert haben. Saslauthd verwendet somit nicht den Pfad aus der smtpd.conf sondern den oben genannten Pfad.

Damit saslauthd beim nächsten Systemstart von Linux automatisch den saslauthd-dienst aktiviert, haben wir wie folgt mit Hilfe vom Yast Runleveleditor die Startkonfiguration geändert:

Yast 2 starten => Menüpunkt “System” auswählen => Runlevel-Editor anklicken => In der angezeigten Liste saslauthd auswählen und mit dem Button “aktivieren” die Eingabe bestätigen

6. TLS zur Verschlüsselung von SMTPD-Verbindungen nutzen

Das primäre Ziel von TLS besteht darin, die Integrität und eine geschützte Verbindung zwischen zwei miteinander kommunizierenden Anwendungen herzustellen. Hierzu wird eine Verschlüsselungsmethode eingesetzt, die auf Zertifikat und Schlüssel basiert. TLS prüft anhand des Zertifikats die Identität des Servers und stellt mit den ausgehandelten Verschlüsselungsmethoden eine sichere Datenübertragung her.

Damit wir eine TLS Verbindung herstellen können, haben wir ein Server Zertifikat generiert. Danach wird es entweder von einer CA, Certification Authority, signiert, oder man signiert es selbst (wir signierten unser Zertifikat selbst). Das CA Cert, den private key und das signierte Server Zertifikat haben wir dann in /etc/postfix/certs gelegt.
Der einzigste Nachteil eines eigenen Zertifikates ist, dass das eMail-Programm das CA nicht kennt und daher den Benutzer warnt. Der Benutzer muss entweder die Warnung ignorieren, oder aber das Zertifikat auf dem Rechner installieren.

6.1 TLS auf dem Server aktivieren (/etc/postfix/main.cf)

Parameter

Aktiviert STARTTLS, auf unseren Postfixserver (Testserver)
smtpd_use_tls = yes

Gibt den Pfad des SSL-Zertifikates an
smtpd_tls_cert_file = /etc/postfix/mail.cert

Gibt den Pfad des privaten Schlüssels an.
smtpd_tls_key_file = /etc/postfix/mail.key

So werden diese Dateien erstellt:

Hiermit wird ein RSA Schlüssel mit 2048 Bit Länge erstellt
Openssl genrsa –out mail.key 2048

Hiermit wird ein CSR (Certificate Signing Request) mit dem öff. Schlüssel des vorher angelegten Schlüsselpaares erstellt.
Openssl req –new –key mail.key –out mail.csr

Mit der Eingabe dieses Befehls wird ein Zertifikat mit der Gültigkeit von 4312 Tagen erstellt.
Openssl x509 –req –days 4312 –in mail.csr –out mail.cert –signkey mail.key

Sobald wir den oben genannten Befehl eingegeben heben, stellt openssl ein paar Fragen zu den Details für das Zertifikat. Es ist eigentlich egal was man bei den Fragen eingibt, außer bei der Frage „Common Name“. Da müssen wir den Rechnername des Mailservers eintragen.

```
Country Name (2 letter code) [AU]:DE

State or Province Name (full name) [Some-State]:Neustadt

Locality Name (eg, city) []:Neustadt

Organization Name (eg, company) [Internet Widgits Pty Ltd]:XXXXXXX

Organizational Unit Name (eg, section) []:Master of Disaster

Common Name (eg, YOUR name) []:testserver

Email Address []:stefanie@testerver.pfalz.de
```

TEST
Damit Postfix die vorgenommen Konfiguration speichert, müssen wir Postfix neu starten, indem wir den Befehl postfix reload eingeben haben. Um zu überprüfen ob der Port 25 noch funktioniert, haben wir den Befehlt telnet localhost 25 eingeben.

7. Installation von Thunderbird

Das Emailprogramm Thunderbird ist eine kostenfreie Software, die man sich aus dem Internet downloaden kann. Das funktioniert folgendermaßen: Wir haben den Namen des E-Mail Programms in die Suchmaschine Google eingegeben und es werden verschiedene Möglichkeiten zum Downloaden angezeigt. Jetzt haben wir die Möglichkeit den gewünschten Download mit dem entsprechenden Betriebssystem zu wählen. Bevor dieser Download vollzogen wird, müssen wir noch den Pfad zur Speicherung angeben.

7.1 Benutzerkonto anlegen (Thunderbird)

Wenn Thunderbird nach der Installation zum ersten Mal gestartet wird, erscheint automatisch der Konten-Assistent und fordert uns auf, ein neues E-Mail Konto anzulegen. Falls dieser Assistent sich nicht automatisch öffnet, können wir den Konto Assistenten wie folgt aufgerufen:

Wir wählen in der oberen Symbolleiste **Extras** => **Konten** => **Konto hinzufügen** aus.

Der Konto Assistent hat uns dabei geholfen das Benutzerkonto richtig anzulegen. Mit dem Button „Weiter“ wird unsere Eingabe bestätigt und es wird ein neues Fenster im Konto Assistent geöffnet.

Fenster „Neues Konto einrichten
Im ersten Fenster das geöffnet wird, haben wir ausgewählt was für ein Konto erstellt werden soll.

Auswahlmöglichkeiten:

- E- Mail Konto
- RSS News & Blogs
- Newsgruppen-Konto

Wir wählten natürlich das E-Mail Konto aus.

Fenster „Identität“
Im Feld „ Ihr Name:“ haben wir den Namen eingegeben der dem Empfänger einer E-Mail angezeigt werden soll.
Eingabe: **stefanie.**

Im Feld „E-Mail-Adresse“ mussten wir unsere E-Mail Adresse eingeben die wir auch auf unseren Postfixserver hinterlegt haben.
Diese Adresse verwenden jene, die uns eine E-Mail schicken wollen.
Eingabe: **stefanie@testserver.de**

Fenster “Server-Informationen“
In diesem Fenster haben wir den Typ unseres Posteingangs ausgewählt.

Auswahlmöglichkeiten:

- POP
- IMAP

Wir haben uns für den POP Server entschieden, weil wir diesen auch auf unserem Postfixserver installiert haben (POP3).

Im Feld „Posteingang-Server:“ haben wir den Namen unseres Posteingangserver angeben müssen.
Eingabe: **testserver**

Thunderbird bieten an, einen globalen Posteingangserver zu erstellen. Das heißt, dass die E-Mails aller Konten in einem Posteingang landen. Möchte man aber, das die Konten getrennt verwalten werden z.B. wenn private und geschäftliche im gleichen Profil genutzt werden sollen, so muss das Häkchen im Feld „Globaler Posteingang (im Lokalen Konto) entfernt werden. In unserem Fall haben wird das Häkchen entfernt.

Im Feld „Postausgang-Server (SMTP)“ haben wir den Namen unseres Postausgangs-Server angeben müssen.
Eingabe: **testserver**

Fenster „Posteingang-Server Benutzername“

Im Feld „Posteingang-Server Benutzername“ haben wir als Bezeichnung für das Konto die E-Mail Adresse genommen.
Eingabe: **stefanie@testserver.de**

Im Feld „Postausgang-Server Benutzername“ haben wir als Bezeichnung für das Konto die E-Mail Adresse genommen.
Eingabe: **stefanie@testserver.de**

Fenster „Konten-Bezeichnung“

In diesem Fenster haben wir die Bezeichnung für das Konto vergeben, mit der es im Programm erscheint.
Eingabe: **stefanie**

Im letzten Fenster wird eine Zusammenfassung von allen Daten angezeigt die wir ausgewählt oder eingetragen haben. Mit dem Button „Fertigstellen“ wurde das E-Mailkonto erstellt.

8. Benutzerkonto anlegen (Linux => K-Mail)

KMail's Einrichtungsdialog ermöglicht eine einfache Einrichtung von KMail. Der Dialog ist in sechs Karteikarten aufgeteilt, von denen jede durch eines der Symbole auf der linken Seite aufgerufen werden kann.
Die sechs Karteikarten sind wie folgt aufgeteilt:

Auswahlmöglichkeiten:

- Identität
- Netzwerk
- Erscheinungsbild
- Komposen
- Sicherheit und
- Diverses

Um ein E-Mailkonto untere K-Mail richtig anzulegen, haben folgende Konfiguration vornehmen müssen.

Wir wählten auf der oberen Symbolleiste **Einstellungen => K-Mail einrichten**. Es öffnet sich neues Fenster, indem wir bereits erstellte Benutzerkonten ändern oder neue Benutzerkonten anlegen können.

Karteikarte „Identität

Fenster: Identität verwalten

Um eine neue Identität anzulegen, wählten wir „neu“ aus. Daraufhin öffnete sich ein weiteres Fenster, indem wir noch zusätzliche Einstellungen vornehmen können.

Den Namen der neuen Identität haben wir in das Feld „Neue Identität“ eingetragen. Dieser Name erscheint in der Identitätsliste.
Eingabe: **bettina**

Nach dem Bestätigen mit „OK“ öffnet sich ein weiteres Fenster, in dem zusätzliche Einstellungen vorgenommen werden können. Im Register „Allgemein“ kann man Namen, Organisation und die E-Mail-Adresse eingeben.

Im Feld „Ihr Name“ haben wir den Namen eingegeben, der dem Empfänger einer E-Mail angezeigt werden soll.
Ihr Name: Eingabe: **bettina**

Im Feld „E-Mail-Adresse“ mussten wir unsere E-Mail Adresse eingeben die wir auch auf unseren Postfixserver hinterlegt haben.
Diese Adresse verwenden jene, die uns eine E-Mail schicken wollen.
E-Mail-Adresse: bettina@testserver.pfalz.de

Die anderen Register Kryptografie, Erweitert, Signatur ist für unser Projekt nebensächlich.

Karteikarte „Netzwerk“

Fenster Einstellungen zum Senden und Empfangen von Nachrichten

Die zweite Einstellung die für uns noch notwendig ist, ist das Netzwerk.

In dieser Karteikarte gibt es zwei Register, jeweils für das Versenden und das Empfangen von Emails.

Versand

Beim Versand „Hinzufügen“ wird ein Fenster geöffnet, indem wir die Versandart auswählen können. Folgende Auswahlmöglichkeiten werden angezeigt:

- SMTP
- Sendmail

Wir wählten SMTP aus, weil Sendmail® sich auf eine lokale Installation bezieht. Dieses Programm hat den Ruf, schwierig zu konfigurieren zu sein. Man sollte es nur verwenden, wenn man bereits eine funktionierende Konfiguration für Sendmail® hat. Außerdem konnten wir kein Sendmail verwenden, weil wir eine Sasldatenbank verwenden.

Als nächstes wird ein Fenster mit dem Register **Allgemein** und **Sicherheit** angezeigt. Im Reiter „Allgemein“ müssen wir die Daten des SMTP-Servers eintragen

In diesem Feld muss der Name angegeben werden.
Name: Eingabe: **bettina**

Hier ist es wichtig, das wir die richtige IP-Adresse angeben. In dieses Feld muss nämlich die IP-Adresse vom E-Mailserver „testserver“ eingetragen werden.
Server: Eingabe: **192.168.0.10**

In diesem Feld haben wir den Standartwert angeben.
Port: Eingabe :**25**

Nach diesen Eingaben haben wir noch „Server verlangt Autorisierung“ ausgewählt, die aktiviert einerseits die Felder Benutzer und Passwort:
Benutzer: Eingabe: **bettina**
Passwort Eingabe: **xxxxxx**

Wir haben diese Option gewählt, damit der Benutzer bei versenden von email´s aufgefordert wird seinen Namen und das dazugehörige Passwort einzugeben.

Das ganze Bestätigen wir noch mit „OK“.

Empfang

Beim Empfang „Hinzufügen“ wird ebenfalls ein Fenster geöffnet, indem wir einen Postfach-Typen auswählen. Folgende Auswahlmöglichkeiten werden angezeigt:

- Lokales Postfach
- POP3
- IMAP
- Disconnected IMAP
- Maildir_Postfach

Wir wählten den Postfach-Typ POP3 aus.

Auch hier wird wieder ein neues Fenster mit einem Register „Allgemein“ angezeigt. Hier geben wir folgende Daten ein:

Fenster „Postfachtyp POP-Postfach

Beim einrichten eines POP3-Postfachs, sollte mindestens der Benutzername, die POP3-Server-Adresse und ggf. noch das Passwort eingetragen werden, wenn man nicht will, dass dies bei jeder Überprüfung von neuen Mails erneut eingegeben werden muss.

Für unser Postfach haben wir folgende Daten festgelegt:

Im Feld “Name” Eingabe **bettina**

Im Feld „Benutzer“ Eingabe **bettina**

Im Feld „Passwort“ Eingabe **bettina**

Im Feld Server haben wir die IP-Adresse von unserem Server eintragen müssen, weil er sonst keine Verbindung zum Server bekommt
Im Feld „Server“: Eingabe **192.168.0.10**

Im Feld „Port“ Eingabe:**110**

Im unteren Feld des Fensters wählen wir noch zusätzlich „POP_Passwort in Konfigurationsdatei speichern“ aus. Damit erreichen wir, das der Benutzer bei empfangen von email´s aufgefordert, wird seinen Namen und das dazugehörige Passwort anzugeben.

Die ganzen Einstellungen bestätigten wir mit OK.

Bevor sich das komplette Fenster schließt, bestätigten wir die kompletten Einstellungen mit „Anwenden“. Dies muss unbedingt beachtet werden, ansonsten werden die Einstellungen nicht gespeichert.
Jetzt sind wir mit allen notwindigen Einstellungen fertig und haben ein Kmail-Konto unter Linux erstellt.

9. Verbindung mit Linux und Mircosoftclient herstellen

Voreinstellung Microsoft-Client

Um eine Verbindung mit dem Postfix-Server (testserver) herzustellen, haben wir folgende Arbeitschritte durchführen müssen.

Schritt 1
Ordner *„Windows“* öffnen => Ordner *„system32“* öffnen => Ordner *„drivers“* öffnen
Ordner *„etc“* öffnen => im Ordner „etc“ muss dann die Datei *„host“* anklicken werden. In diese Datei muss die IP-Adresse und der Name des Servers eingetragen werden. Man darf keine Leertaste zwischen der IP-Adresse und dem Namen setzen sondern unbedingt einen Tab.
Datei speichern

2. Schritt.
Netzwerksymbole anklicken (Startsymbolleiste)

Man befindet sich dann in der „Status von LAN Verbindung“
⇨ Button „Eigenschaften“ auswählen

Im Feld: Diese Verbindung verwendet folgende Elemente: müssen wir
⇨ *Internetprotokoll TCP/IP* doppelt anklicken

Fenster: Eigenschaften von Internetprotokoll
⇨ Button *„Erweitern“* bestätigen

Fenster: Erweiterte TCP/IP Einstellungen
Sparte:WINS auswählen
⇨ *LMHOST Abfrage* aktivieren
⇨ Button *„LMHOSTS importieren“* bestätigen
⇨ Erstellte Textdatei auswählen *„LMHOSTS“*
⇨ Inhalt der Datei überprüfen / bei Richtigkeit Datei speichern
⇨ Button „*OK“* bestätigen

Um zu testen ob die zwei Rechner miteinander kommunizieren können, haben wir auf der DOS Ebene den Befehl ping 192.168.0.10 eingeben.

Anhang

Konfigurationsdatei

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 postconf(5) manual page for a
# complete list.
#
# The general format of each line is: parameter = value. 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 VIRTUAL_README).
#
# 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
# STANDARD_CONFIGURATION_README).
#
# 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, localhost
#mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain
#mydestination = $myhostname, localhost.$mydomain, localhost, $mydomain,
#       mail.$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, $inet_interfaces or $proxy_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 local(8)).
#
# 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
```

```
# ${proxy,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 = 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 postconf(5).
#
# 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 = 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 = 192.168.0.0/24, 127.0.0.0/8,
#mynetworks = $config_directory/mynetworks
#mynetworks = hash:/etc/postfix/network_table
```

```
# The relay_domains parameter restricts what destinations this system will
# relay mail to.  See the smtpd_recipient_restrictions description in
# postconf(5) 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 or $proxy_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 description in postconf(5).
#

# 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 = [mailserver.isp.tld]
#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
#
# The ADDRESS_REWRITING_README document gives information about
# address masquerading or other forms of address rewriting including
# username->Firstname.Lastname mapping.

# ADDRESS REDIRECTION (VIRTUAL DOMAIN)
#
# The VIRTUAL_README document gives information about the many forms
# of domain hosting that Postfix supports.

# "USER HAS MOVED" BOUNCE MESSAGES
#
# See the discussion in the ADDRESS_REWRITING_README document.

# TRANSPORT MAP
#
# See the discussion in the ADDRESS_REWRITING_README document.

# 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@$mydestination,
# unknown@[$inet_interfaces] or unknown@[$proxy_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. The file
# SMTPD_ACCESS_README provides an overview.

# 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 "man header_checks".
#
#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".
# See the ETRN_README document for a detailed description.
#
# The fast_flush_domains parameter controls what destinations are
# eligible for this service. By default, they are all domains that
# this server is willing to relay mail to.
#
#fast_flush_domains = $relay_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

# html_directory: The location of the Postfix HTML documentation.
#
html_directory = /usr/share/doc/packages/postfix/html
```

```
# 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.
# This parameter is obsolete as of Postfix 2.1.
#
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
biff = no
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 = testserver.pfalz.de
program_directory = /usr/lib/postfix

masquerade_domains =
mydestination = $myhostname, localhost.$mydomain
defer_transports =
disable_dns_lookups = no
relayhost = [mail.pfalz.de]
mailbox_command =
mailbox_transport =
#SMTP mit SASL-Authentification verwenden
smtp_sasl_auth_enable = yes
smtpd_use_tls = yes
alias_maps = hash:/etc/aliases
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
smtpd_sasl_local_domain =
broken_sasl_auth_clients = yes
#Die Paßwörter stehen in der Datei /etc/postfix/smtp_
smtp_sasl_password_maps = hash:/etc/postfix/sasl_passwd
smtpd_tls_cert_file = /etc/postfix/mail.cert
smtpd_tls_key_file = /etc/postfix/mail.key
```

Konfigurationsdatei

master.cf

```
ï"¿
#
# 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.
#
# General main.cf options can be overridden for specific services.
# To override one or more main.cf options, specify them as arguments
# below, preceding each option by "-o".  There must be no whitespace
# in the option itself (separate multiple values for an option by
# commas).
#
# 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
  -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 -o
smtpd_etrn_restrictions=reject
#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       oqmgr
#tlsmgr   fifo  -       -       n       300     1       tlsmgr
rewrite   unix  -       -       y       -       -       trivial-rewrite
bounce    unix  -       -       y       -       0       bounce
defer     unix  -       -       y       -       0       bounce
trace     unix  -       -       n       -       0       bounce
verify    unix  -       -       n       -       1       verify
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
anvil     unix -      -      n      -      1      anvil
#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
procmail  unix -      n      n      -      -      pipe
flags=R user=nobody argv=/usr/bin/procmail -t -m /etc/procmailrc ${sender}
${recipient}
```

Quellenverzeichnis

Sasl - Konfiguration
Quelle: http://helpdesk.std-service.de/staticpages/index.php/2004111610415756
Quelle: http://www.nextportal.de/viewtopic.php?p=4487&highlight=
Quelle: http://postfix.state-of-mind.de/patrick.koetter/easterhegg2004/
Quelle :http://workaround.org/articles/ispmail-sarge/index.shtml.de#id2515201

Netzwerk
Quelle: http://www.different-thinking.de/heimnetz.php

master.cf
Quelle: http://www.werthmoeller.de/doc/microhowtos/postfix/postfixConfig/main.cf

saslauth Kohnfiguration
Quelle: http://holl.co.at/howto-email.de

SMTP Auth mit Postfix
Quelle: http://www.tuxhausen.de/postfix_smtp_auth-3.html
Quelle: http://postfix.state-of-mind.de/patrick.koetter/easterhegg2004/

Das Postfixbuch von Peer Heinlein
„Sichere Mailserver mit Linux“
ISBN 393751404