Moderne Kurierdienste

von Martin Müller

Kaum vorzustellen, dass das Kommunikationsmittel eMail vor fünf Jahren nur für wenige ausgesuchte Anwender eine Bedeutung hatte, wo es doch heute beinahe jedem, der einen Computer besitzt, zur Verfügung steht. Die WCM-Linux-Box stellt durch ihre zentrale Position im Netzwerk eine ideale Grundlage für den Einsatz als Mailserver dar. Mit Postfix, Courier und Co. werten wir die Box durch deren Funktionsumfang zu einem mächtigen Open Source-Mailserver auf.

Hallo und willkommen zu einem weiteren Teil des WCM-Linux-Box-Workshops.

Eine eMail zu schreiben und zu versenden kann in diesen Zeiten fast jeder. Einfach und schnell zischen die Mails zwischen Sender und Empfänger hin und her. Tatsächlich aber, sind der Versand, der Transport und die Zustellung von eMails ein recht komplexer Vorgang.

Vorweg möchte ich anmerken, dass das Betreiben eines Mailservers eine verantwortungsvolle Arbeit ist, da Sie einige Pflichten erfüllen und das Postgeheimnis wahren müssen. Zu den Pflichten gehört zum Beispiel, dafür zu sorgen, Ihren Mailserver abzusichern und eine hohe Verfügbarkeit zu gewährleisten. Der Server sollte also 24 Stunden am Tag, 7 Tage die Woche, erreichbar sein und über eine statische IP-Adresse verfügen. Dienste wie DynDNS sind für solch eine Anwendung nur die halbe Miete und sollten für einen problemlosen Betrieb vermieden werden. Schließlich wollen Sie doch nicht, dass Teile Ihrer wertvollen Kommunikation verloren gehen, oder verspätet eintreffen.

Ganz nach Protokoll

Man sollte meinen, dass der Versand der Mail und deren Abholung aus dem Postfach mit einem Protokoll auskommen. Tatsächlich sind dies zwei höchst unterschiedliche Vorgänge, die eigenen Regeln folgen müssen.

Für die Einlieferung von eMails ist SMTP zuständig. Dieses Simple-Mail-Transfer-Protokoll sorgt dafür, dass eMails von jedem beliebigen Anwender angenommen werden. Die Einlieferung erfolgt fast anonym, unser Mailserver muss die einliefernde Stelle also nicht „kennen“ damit er Mails entgegen nehmen kann. Genau das ist aber auch eine der großen Schwachstellen im System – jeder kennt das Problem der unerwünschten Werbe-Massenemails die die Mailboxen verstopfen: Spam.

Die Abholung von Nachrichten hingegen erfordert jedoch eine Authentifizierung. Schließlich sollen die Nachrichten nicht für jedermann zugänglich und lesbar sein. Damit das Briefgeheimnis gewahrt beleibt findet die Abholung über das Post-Office-Protocol (POP3) oder das Internet-Mail-Access-Protocol (IMAP) statt.

Beschäftigen wir uns zunächst einmal mit dem Versand und der Einlieferung der Nachrichten. Das SMTP ist ein weltweiter Standard, sozusagen eine Sprache die alle Mailserver unserer Zivilisation sprechen. Bei jeder Zustellung läuft ein genau definierter Kommunikationsprozess ab, der sich einfach abbilden lässt. Zuerst begrüßen sich die beiden Server durch das HELO/EHLO-Kommando, danach geben sie sich gegenseitig ihre Namen bekannt, der Absender- und der Empfänger werden übertragen und schließlich erfolgt die Übertragung der eigentlichen Maildaten. Bei Erfolg wird die Verbindung wieder abgebaut.

Postfix als Postfuchs

Wir verwenden die freie Software Postfix für den Prozess der Zustellung. Postfix wird als Benutzer root durch apt-get install postfix postfix-mysql installiert. Bei der Konfigurationsabfrage wählen wir Internet Site, anschließend müssen wir den vollen Domainnamen angeben (FQDN) von dem die eMails kommen sollen. Möglicherweise werden Sie gefragt was mit den Mails für root passieren soll. Entweder lassen Sie sich diese Nachrichten auf ein reguläres Mailkonto zustellen, oder Sie belassen die Einstellungen wie Sie bisher waren (NONE) und da wir auf der Box das ReiserFS-Filesystem fahren, kön-

```
# /etc/postfix/ids.mysql
user=mailadmin
password=IhrPasswort
hosts=127.0.0.1 # Nicht localhost verwenden!
dbname=mailbasis
table=mailuser
select_field=uid
where_field=konto
```

nen wir beruhigt auf synchronisierte Updates verzichten (NO).

Die Konfigurationsdateien finden wir unter /etc/postfix/. Wir interessieren uns zuerst für die Datei main.cf in der sämtliche Einstellungen betreffend die Außenwirkung und einige Optionen die nur intern wirken, zu finden sind. Wir starten den Editor pico /etc/postfix/main.cf.

Da wäre zunächst der Eintrag myhostname. Durch ihn definieren wir welchen Absender eMails von unserem Mailserver erhalten. Es ist sehr wichtig hier richtige Angaben zu machen, da immer mehr Gegenstellen bei der Einlieferung prüfen, ob dieser Absender auch tatsächlich im DNS eingetragen ist. Dadurch sinkt die Wahrscheinlichkeit, dass Spam eingeliefert wird. Paranoid konfigurierte Mailserver lehnen die Einlieferung von Nachrichten ohne DNS-Eintrag ab, Ihre Mail kann nicht zugestellt werden!

Es folgt mydestination bei dem angegeben wird, für welche Domain sich Postfix zuständig fühlen soll. Nur für die Domain die hier eingetragen ist, nimmt Postfix eMails an. Bitte vergessen Sie nicht, auch localhost mit aufzunehmen.

Um von Spammern nicht als frei zugängliche Versandstation für Massenemails missbraucht zu werden, konfigurieren wir den Eintrag für mynetworks auf 127.0.0.0/8, 192.168.123.0/24.

Somit dürfen nur Rechner aus dem internen Netz und der Mailserver selber Nachrichten versenden. Danach prüfen wir, ob smtpd_banner nur $myhostname ESMTP bekannt gibt. Unser Mailserver gibt bei der Begrüßung anderer Server somit nur unseren Hostnamen und die gewünschte Kommunikationsform bekannt. Weitere Informationen über unser System behält er nun für sich und liefert sich möglichen Angreifern nicht gleich von selbst ans Messer.

Haben Sie mehrere Festplatten oder Partitionen in der Linux-Box, wollen Sie vielleicht die Standardwarteschlange der eMails an einen anderen Ort als /var/spool/postfix/ verlegen. Durch die Verlegung könnten Sie den Speicherplatz erweitern oder die Warteschlange über ein ausfallssicheres Festplattenarray lenken. Jedenfalls ist der Eintrag queue_directory dafür verantwortlich wo die Nachrichten auf ihre Auslieferung warten. Fehlt dieser Eintrag wird /var/spool/postfix benutzt. Wenn Sie den Speicherplatz verlegen, denken Sie bitte bei der nachfolgenden Konfiguration in der MySQL-Datenbank daran und setzten Sie die entsprechenden Pfade.

Der erste Konfigurationsdurchlauf ist fertig, wir starten Postfix durch das Kommando /etc/init.d/postfix restart neu und installieren mit apt-get install telnet das altbewährte Telnet-Werkzeug um später eine Verbindung zu Postfix herzustellen.

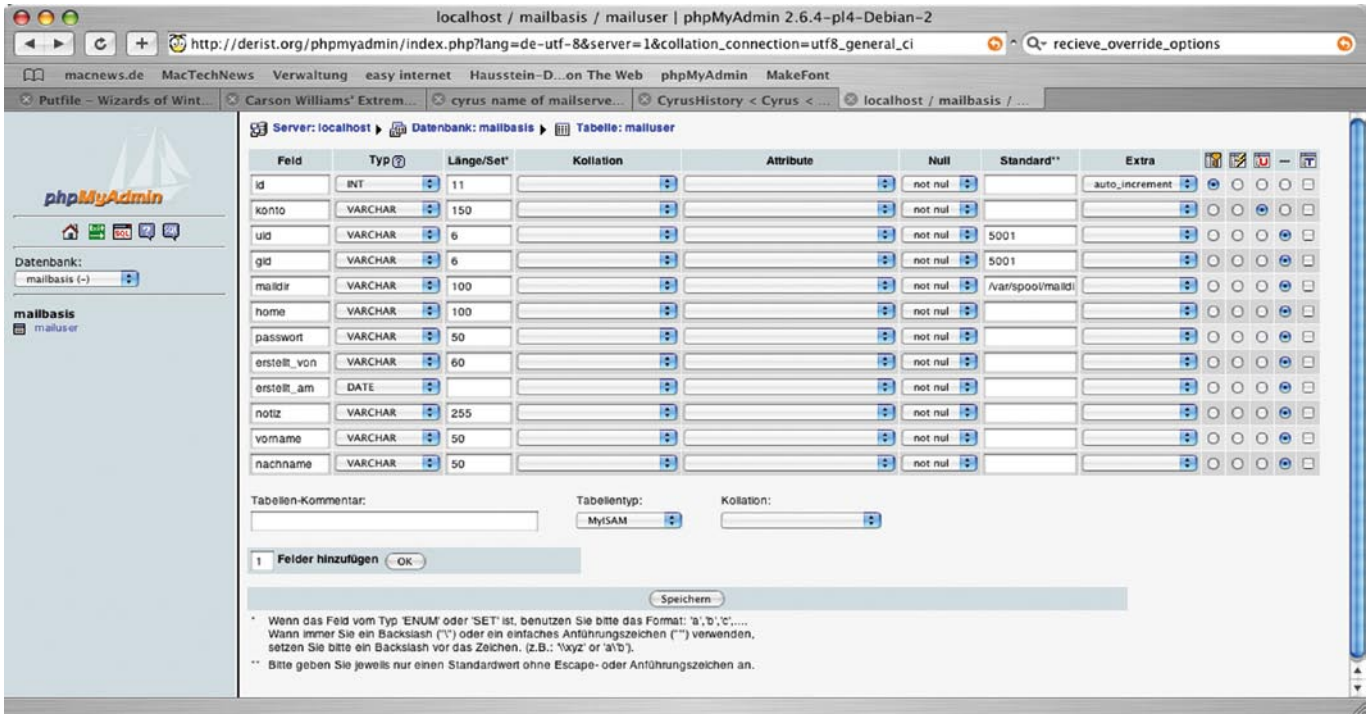
Ist da jemand?

Zuerst sehen wir nach, ob Postfix läuft. ps awx|grep postfix sollte uns mindestens eine Ausgabezeile liefern, in der folgender Pfad zu finden ist /usr/lib/postfix/master. Postfix läuft - schauen wir weiters, ob Postfix auch auf Port 25 seine Ohren offen hat: lsof –i :25 sollte eine Zeile liefern die mit master beginnt und mit (LISTEN) abschließt.

Wird auf Port 25 nicht gelauscht, könnte möglicherweise die Firewall nicht richtig konfiguriert sein. Wie das geht haben wir in der ersten Folge des Workshops beschrieben.

Weiters klopfen wir die Maschine nach offenen Ports mit nmap localhost ab (apt-get install nmap). Die Ausgabeliste muss die Ports 25 und 110 enthalten. Mögliche Fehlerquelle ist auch hier wieder die Firewall.

Jetzt treten wir über telnet localhost 25 mit Postfix in Verbindung. Postfix sollte sich melden und nach der Ausgabe von 220 mail.IhreDomain.at ESMTP auf Ihre Eingabe warten. Nun bilden wir die Kommunikation der Mailserver nach:

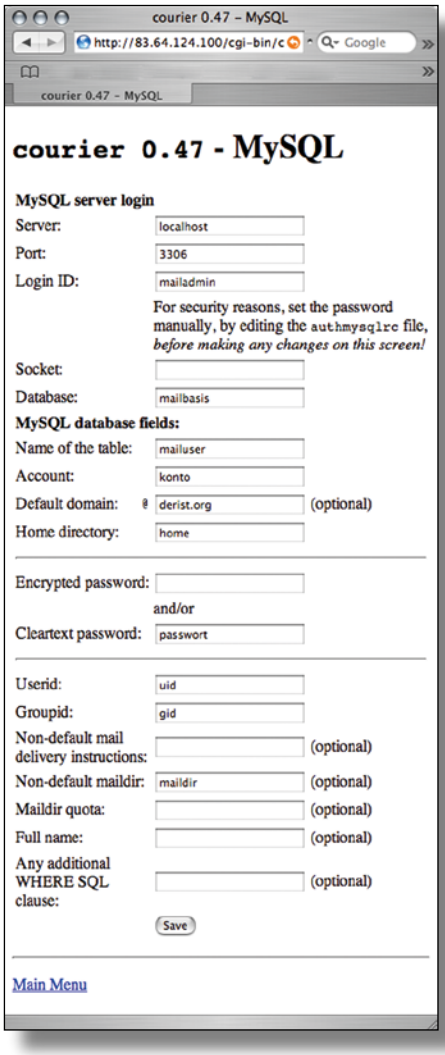


Sie begrüßen den Mailserver durch helo EineBestehendeDomäne und schließen mit Enter ab, Postfix antwortet mit 250 IhrRechnername. Danach kündigen Sie eine eMail durch Eingabe von Mail from: IhreBestehendeEmailadresse@EineDomäne.org

an. Postfix antwortet wieder mit dem Code 250, der OK bedeutet. Wir senden eine eMail mit rcpt to: root@IhreDomäne.org und teilen nun durch data mit, dass die Daten der Email kommen. Schreiben Sie einige Zeichen, drücken Sie Enter und schreiben zum Abschluss einen Punkt (.) in eine Zeile. Postfix interpretiert den Punkt als fertig und stellt die Mail root zu. Wir beenden die Verbindung durch QUIT. Um den Prozess richtig durchzuspielen, verwenden wir den kommandozeilen-

basierenden eMail-Client mutt. Sollten Sie, anders als von uns empfohlen, kein anderes Benutzerkonto als root zur täglichen Arbeit eingerichtet haben, senden Sie sich eben selbst (root) eine eMail.

Mutt rufen sie durch Eingabe von mutt ohne Parameter auf, die Befehle zur Steuerung von mutt sind am oberen Bildschirmrand aufgelistet. Wir schreiben eine neue Mail durch Drücken der Taste m, geben als Empfänger IhrBenutzer@localhost ein, nach dem frei wählbaren Subject (Betreff) dürfen wir den Editor vi benutzen, um Text einzugeben. vi ist sehr mächtig, aber nicht gerade intuitiv zu bedienen. Zuerst müssen Sie die Eingabe



durch Drücken der Taste i aktivieren, tippen Sie ein paar Buchstaben und verlassen Sie vi durch Betätigen der ESCape-Taste und Eingabe der Zeichenfolge :wq abgeschlossen durch die ENTER-Taste.

Sie sehen die Mail noch mal zur Kontrolle und können sie nun mit y versenden. Wechseln sie an eine neue Konsole, melden Sie sich als der Benutzer an, an dem Sie die Mail versendet haben und rufen Sie die Nachricht durch Eingabe von mail ab. Im Programm mail können Sie die Nachricht mit der Taste n für „next“ abrufen, verlassen Sie mail durch q.

Gratulation! Sie haben die erste eMail über Ihren neuen Mailserver versendet und abgeholt.

Ist die Mail nicht zugestellt worden, prüfen Sie bitte als root mit dem Kommando mailq ob die Mail noch in der Warteschlange wartet und sehen Sie die Log-Files unter /var/lov/mail.log und /var/log/mail.err nach Hinweisen zur Fehlerursache durch.

MySQL zieht mit

Wie wir soeben gesehen haben, können wir Benutzern die ein lokales Systemkonto haben eine Nachricht zustellen. Stellen Sie sich vor, sie wollen zehn Personen eMail-Adressen vergeben, wollen diesen zehn Personen jedoch keinen weiteren Zugriff auf Ihren Rechner gewähren. Na gut, Sie können diesen Personen beim Anlegen der Konten keine Shell (Terminalfenster) zuweisen und ihnen somit den direkten Zugriff versperren, dennoch sind diese User im System vorhanden und ein potenzielles Sicherheitsrisiko. Warum die Benutzerverwaltung nicht an eine MySQL-Datenbank übergeben? Rasch den nächstgelegenen Internetbrowser aufrufen und mit phpmyadmin eine neue Datenbank mailbasis angelegt (siehe letzte WCM-Ausgabe). In diese Datenbank legen wir eine Tabelle namens mailuser, die Felder und deren Werte entnehmen Sie bitte dem Screenshot MySQL-Mailuser.

Weiters brauchen wir einen privilegierten Datenbank-User, der auf diese Datenbank zugreifen darf. Legen Sie in phpmyadmin im Punkt Rechte einen User mailadmin an. Im Host-Feld tragen Sie localhost ein, das Kennwort vergeben Sie frei nach Ihren Vorstellungen. Vergeben Sie diesem User ausschließlich Rechte zur Manipulation der Daten (SELECT, INSERT, UPDATE, DELETE).

Danach wechseln Sie in ein MySQL-Terminal (mysql –u root –p) und geben im MySQL-Terminal folgenden Syntax ein: USE mysql; INSERT INTO db (Host, Db, user, Select_priv) VALUES(‚localhost‘,‘mailbasis‘,‘postfix‘,‘Y‘); verlassen sie MySQL durch QUIT und kehren Sie zu phpmyadmin zurück. Legen Sie in der Datenbank mailbasis eine weitere Tabelle weiterleitungen an. Für Feldtypen und Definitionen ziehen Sie bitte wieder den Screenshot zu rate. Legen Sie nun mit Hilfe von phpmyadmin in der Datenbank mailbasis in der Tabelle mailuser durch die phpmyadmin-Funktion „Einfügen“ einen ersten eMail-Benutzer an. Das erste Feld id lassen Sie bitte leer, es erstellt seinen Wert automatisch, das Feld konto soll mit der vollen eMail-Adresse inklusive dem Teil nach dem Klammeraffen @ gefüttert werden, uid und gid haben immer den Standardwert 5001, maildir enthält das Verzeichnis relativ zu dem in der main.cf modifizierten Eintrag und muss immer zwingend mit einem / abschließen! Es reicht hier also der Eintrag IhrBenutzer/, da der vordere Teil sich durch den Eintrag virtual_mailbox_base in der Datei main.cf ergibt.

```
# /etc/postfix/mailbox.mysql
user=mailadmin
password=IhrPasswort
hosts=127.0.0.1 # Nicht localhost verwenden
dbname=mailbasis
table=mailuser
select_field=maildir
where_field=konto
```

Wir simulieren das Homeverzeichnis, weil es der Simulation eines „richtigen“ Systemusers am nächsten kommt. Um Courier zufrieden zu stellen, nehmen wir gleich das maildir des jeweiligen Benutzers (/var/spool/maildir/IhreDomain/IhrBenutzer).

Das Feld Passwort ist selbsterklärend. Die weiteren Felder dienen lediglich der bessern Verwaltung der Konten und sind technisch nicht erforderlich.

Damit MySQL später keine Probleme mit der Verbindung hat, fügen wir in der Datei /etc/hosts folgende Zeile ein: 127.0.0.1 localhost localhost.

Wer bringt´s?

Courier ist ein POP3- und IMAP-Server der von Sam Varshavchik entwickelt wurde.

Wir haben uns für Courier als MTA (MailTransferAgent) und für das IMAP-Protokoll entschieden, da es den Anforderungen mobiler und dennoch zentraler Mailverwaltung entspricht.

Viele kennen das Problem: Man richtet sich am Arbeitsplatz auch das private eMail-Konto ein, hat am Laptop einige Mailkonten und natürlich am Rechner zu Hause auch noch. Und dann beginnt´s: Von welchem Rechner habe ich ein bestimmtes Mail abgerufen? Oder Kunde X meinte, man habe ihm von der allgemeinen Firmenadresse ein Mail geschrieben, doch kein Mitarbeiter kann das besagte Stück in seinen „gesendeten Objekten“ finden. Diese Szenarien passieren, wenn POP3 Verwendung findet. Die Mails

werden abgerufen und lokal gespeichert. Der Überblick geht verloren, da ja jeder Rechner seine eigene Mail-Datenbank betreibt.

IMAP lässt die Mails am Server und zeigt sozusagen nur das Inhaltsverzeichnis der Mailbox an. Bei Bedarf wird eine Verbindung vom Mailclient zum Server hergestellt und die entsprechende Mail abgerufen. Wenn am Rechner A und am Rechner B das Konto testmuster@wcm.at über IMAP eingerichtet ist, haben beide Clients auf dieselben Daten zugriff. Doppelte oder verschollene Mails gehören der Vergangenheit an, da die Daten zentral am Server liegen. Um die Privatsphäre zu schützen können auch Zugriffsbeschränkungen auf einzelne Order oder ganze Ordnerstrukturen ver-

```
# /etc/postfix/forward.mysql
user=mailadmin
password=IhrPasswort
hosts=127.0.0.1 # Nicht localhost verwenden!
dbname=mailbasis
table=weiterleitungen
select_field=forward_to
where_field=forward_from
```

geben werden. Nun hat diese Technik aber auch Nachteile. Um Mails lesen zu können, braucht man eine Verbindung zum Server. Im internen Netz sollte das kein Problem sein, da der Server sowieso immer eingeschaltet ist. Anders sieht die Sache für Laptop-User aus. Um mal schnell den Inhalt der Mail von voriger Woche zu durchforsten, braucht´s eine Verbindung zum Server. Sonst sieht der User nur den Betreff. Dieses Manko kann durch moderne MailUserAgents, kurz MUA, ausgeglichen werden. Fast alle Mailclients bieten die Möglichkeit Kopien der Mails lokal zu speichern. Wie viele Mails es sein dürfen und wie viel Speicherplatz sie belegen dürfen kann eingestellt werden. Wir installieren Courier als root durch Eingabe von apt-get install.

Die Abfrage nach den Konfigurationsverze ichnissen bejahen wir, danach tragen wir die gültige Domain ein von der die Mails kommen, danach bestätigen wir die Mail-From-Abfrage durch OK und aktivieren das CGI-Programm (JA). Jetzt vergeben wir noch ein Passwort für den Administrator von Courier. Zurück an der Shell legen wir einen symbolischen Link durch ln –s /usr/share/courier/webadmin/ /IhrWebverzeichnis/webadmin von der Installationsquelle in Ihr Apache2-DocumentRoot. Wir gehen sicher dass wir auch mit PHP4 CGIs ausführen können (apt-get install php4-cgi) und erstellen mit dem Befehl touch die Datei unsecureok im Verzeichnis /etc/courier/webadmin/. Jetzt dürfen wir trotz fehlender SSL-Unterstützung Courier per Webinterface konfigurieren.

Die Mail-Datenbank wandert

Loggen Sie sich mit dem Browser durch http://IhreServerIP/cgi-bin/courierwebadmin im Courier-Sytem ein. Im ersten Punkt Mailservername tragen wir bei Server Name wie bei Postfix den richtigen Domainnamen ein. Bestätigen Sie mit SAVE und gehen Sie zurück zum Hauptmenü. Im nächsten Punkt Password authentication modules wählen Sie im Ausklappmenü Add authentictation module das authmysql-Modul und fügen es durch SAVE der Liste hinzu. Außerdem müssen Sie alle anderen Einträge aus der Liste entfernen. Sichern sie mit SAVE und wechseln Sie innerhalb von Courierwebadmin zum Punkt MySQL.

Die Einstellungen für MySQL entnehmen Sie bitte dem Screenshot.

Die Installation beenden Sie durch klicken auf Install new configuration.

Leider funktioniert diese Art der Konfiguration nicht immer vollständig korrekt, weshalb Sie die Datei /etc/courier/authmysqlrc zur Sicherheit überprüfen sollten, insbesondere das Passwort für den Datenbankzugriff.

Nun geht's wieder ans Terminal um Postfix auf MySQL umzustellen. Wir müssen Postfix anweisen, dass nicht mehr der MailTransferAgent (MTA) für die Auslieferung zuständig ist, sondern der MailDeliveryAgent (MDA).


```
# /etc/postfix/forward.mysql
user=mailadmin
password=IhrPasswort
hosts=127.0.0.1 # Nicht localhost verwenden!
dbname=mailbasis
table=weiterleitungen
select_field=forward_to
where_field=forward_from
```


Wir beginnen damit, dass wir die Dateien ids.mysql und mailbox.mysql im Verzeichnis /etc/postfix/ anlegen. Die Dateien enthalten die Zugangsdaten zur MySQL-Datenbank, den genauen Inhalt entnehmen Sie bitte der Box MySQL-Konfiguration. Damit Postfix nicht versucht die Mails lokal zuzustellen, sondern den Weg über die MySQL-Datenbank geht, müssen wir unsere Domain aus $mydestination nehmen und dafür in virtual_mailbox_domains eintragen.

Zur Vollendung des Transportweges müssen wir für die Domain noch die Transportmethode auf virtual umstellen. Das erledigen wir indem wir die Datei /etc/postfix/transport erstellen und die Zeile ihre.domain virtual: eintragen und anschließend den Befehl postmap transport ausführen.

Da wir einen echten User mit der MySQL-Datenbank nachbilden, legen wir einen richtigen User mit der System-UID 5001 und eine Gruppe mit der GID 5001 an. Wir reservieren sozusagen die IDs damit sie später nicht unbeabsichtigt vergeben werden können. useradd -u 5001 -d /va/spool/postfix -s /bin/false mailsystem und groupadd -g 5001 mailsystem ist die richtige Syntax für diese Aufgabe.

Nun legen wir noch das Ablageverzeichnis für die Mails an mkdir –p /var/spool/maildir/IhrMailBenutzername und weisen ihm den richtigen Eigentümer zu (chown 5001:5001 /var/spool/maildir/IhrMailBenutzername). In einem weiteren Teil des Workshops werden wir dies automatisieren.

Damit wirklich alle Änderungen vollzogen sind, starten Sie den Server am besten komplett neu und richten anschließend ein Konto in Ihrem IMAP-fähigen eMail-Client ein (zB. Thunderbird). Verwenden Sie bei der Konfiguration noch keine Domainnamen, sondern die IP-Adressen des lokalen Netzes. Schließlich haben wir noch keinen richtigen MX-Eintrag im DNS.

Der Login wir vorerst noch scheitern da noch keine Courier-Verzeichnisse angelegt sind. Schicken Sie deshalb eine Initialisierungs-Email an die soeben erstellte Email-Adresse.

In unserem Beispiel wäre dies eine eMail an max.muster@192.168.123.1 Beobachten Sie das Logfile /var/log/mail.info um etwaige Fehler aufzuspüren.

Gern gemachte Fehler betreffen die Schrägstriche bei den Verzeichnisangaben in der MySQL-Tabelle und die Groß-Klein-Schreibung bei den Passwörtern.

DNS

Jetzt wird's ernst, wir stellen den Mailserver scharf. Bis jetzt haben wir den Mailserver zwar richtig konfiguriert, den Mailverkehr

für unsere Domain hat allerdings ein anderer Mailserver abgewickelt, nämlich jener der im Domain-Name-System beim MX-Record (MailEXchange) eingetragen wurde. Für gewöhnlich ist das der Mailserver ihres Providers. Viele Provider und Registrierungsstellen bieten ein Webinterface für Ihre Kunden an, mit denen Sie Einträge im DNS manipulieren können.

Geben Sie nun über dieses Interface Ihre IP-Adresse für den MX-Record ein, an der Postfix lauscht.

Gibt es kein Webinterface müssen Sie ihre Domainregistrierungsstelle nach der genauen Vorgangsweise befragen.

Außerdem müssen Sie im lokalen DNS an der WCM-Linux-Box ebenfalls den richtigen MX-Record eingeben. Sonst funktioniert der Versand innerhalb des lokalen Netzes nicht richtig. Wie das geht haben wir in einer der letzten Ausgaben beschrieben.

Bis DNS-Änderungen auch bei anderen DNS-Servern vermerkt sind, vergehen oft 24 Stunden.

Erst ab dem Zeitpunkt ab der der Mailserver richtig im DNS bekannt ist, können Sie in Konfigurationsdialogen und als eMailadresse IhreDomain verwenden.

Ist Ihr Mailserver richtig am Netz und im DNS richtig eingetragen, sollten Sie prüfen ob Ihr Server als nicht OpenRelay dient. OpenRelays werden von Spammern benutzt, um Ihren Werbemüll unters Volk zu bringen. Dafür gibt es verschiedene Dienste, die Ihren Server gratis testen. Einer der Bekanntesten ist http://www.abuse.net/relay.html der auf siebzehn verschiedene Arten versucht, ihren Server als Relay zu verwenden.

Bei den Eingabefeldern auf der abuse.net-Seite brauchen sie lediglich in das Feld Adress to test Ihre IP-Adresse eingeben, die anderen Felder können Sie leer lassen.

Dieser Test ist sehr wichtig, denn ist Ihr Mailserver erst in einer der schwarzen Listen der Spambekämpfer gespeichert, nehmen unzählige Mailserver keinerlei Nachrichten mehr von ihnen an!

Dann müssen Sie erst Ihr OpenRelay schließen und bei den Blacklist-Betreibern eine Streichung beantragen. Seien Sie also vorsichtig bei der Konfiguration des Servers!

Schlüssel zum Erfolg

Bis jetzt können wir nur aus dem lokalen Netz eMails versenden. Das ist eigentlich kein Problem, denn wenn jemand von einem anderen Standort aus eMails versenden will, kann er dies noch immer über den Mailserver seines Providers.

Auch auf eine Verschlüsselung bei der Authentifizierung konnten wir bisher verzichten, da wir von einem friedlichen internen Netz ausgehen.

Eleganter Weise bringen wir Postfix jedoch SASL (Simple Authentication and Security Layer) bei, das aufgrund der Verschlüsselung auch den Versand von außen zulässt.

Wir legen ein Verzeichnis mit mkdir /etc/

postfix/sasl an und erstellen darin die Datei smtpd.conf. Den Inhalt entnehmen Sie der Box MySQL-Konfiguration. In der Datei stehen wieder sämtliche Zugangsdaten für die Kommunikation zwischen MySQL und SASL.

Zusätzlich installieren wir postfix-tls libsasl2 libsasl2-modules libsasl2-modules-sql openssl. Wir erstellen uns ein SSL-Zertifikat, dass für den Rechnernamen mail.IhreDomain.org gültig ist.

Wechseln Sie ins Verzeichnis /etc/postfix und generieren Sie ein SSL-Zertifikat durch openssl req -new -outform PEM -out smtpd.cert -newkey rsa:2048 -nodes -keyout smtpd.key -keyform PEM -days 365 -x509. Das Zertifikat ist ein Jahr gültig. Die Fragen zum Zertifikat beantworten Sie wahrheitsgemäß, wenn Common Name abgefragt wird, müssen Sie den Rechnernamen eintragen! Das ist der gleiche Name den sie in der main.cf bei myhostname angegeben haben.

Wenn eine eMail-Client nun auf Ihren Server zugreift, wird er Sie fragen ob er Ihr Zertifikat annehmen soll, da es möglicherweise unsicher ist.

Sicheres Zertifikat

Natürlich ist unser Zertifikat genau so sicher wie ein bei einer offiziellen Zertifizierungsstelle beantragtes, nur dass es eben nicht in deren Verzeichnis aufgenommen ist. Sie können das Zertifikat also bedenkenlos akzeptieren. Wir tätigen noch einen kleinen Eingriff in der main.cf damit über SASL authentifiziert wird (siehe Box main.cf). Wenn Sie die Änderungen durchgeführt haben, lesen Sie die main.cf neu ein (postfix reload) und prüfen Sie den Syntax (postfix check).

Sollten Sie Probleme mit SASL haben, fügen Sie der smtpd.conf die Zeile log_level: 7 hinzu und prüfen Sie die Ausgaben in /var/log/mail.info.

Um auf Nummer sicher zu gehen, starten Sie eine telnet-Sitzung und begrüßen Sie den Mailserver durch ehlo EineDomain.org. In der Ausgabe werden Sie mit einer Zeile 250 STARTTLS belohnt. Auch hier sollten Sie wieder abschließend den OpenRelay-Test von abuse.org ausführen.

Stellen Sie gegebenenfalls in den Einstellungen Ihres eMail-Clients die Verbindung zum Mailserver so um, dass TLS (Transport Layer Security) verwendet wird.

In der aktuellen Konfiguration haben Sie auf Ihre eMails über das IMAP und das POP3-Protokll Zugriff.

In den nächsten Ausgaben wollen wir die WCM-Linux-Box um eine Anti-Spam-Lösung und um einen Virenscanner erweitern. Außerdem steht für eine der nächsten Ausgaben ein Webinterface für die Userverwaltung und ein WebEmail-Interface am Programm. ■

```
#Zusätzlicher Abschnitt für SASL in /etc/main.cf
smtpd_sasl_auth_enable = yes
broken_sasl_auth_clients = yes
smtpd_recipient_restrictions = permit_mynetworks, permit_sasl_authenticated, reject_unauth_destination
smtpd_use_tls = yes
smtpd_tls_cert_file=/etc/postfix/smtpd.cert
smtpd_tls_key_file=/etc/postfix/smtpd.key
```