Wiki / Erweiterte Konfiguration

Dieser Artikel wurde für die folgenden Ubuntu-Versionen getestet:

Dieser Artikel ist größtenteils für alle Ubuntu-Versionen gültig.

Artikel für fortgeschrittene Anwender

Dieser Artikel erfordert mehr Erfahrung im Umgang mit Linux und ist daher nur für fortgeschrittene Benutzer gedacht.

Zum Verständnis dieses Artikels sind folgende Seiten hilfreich:

1. **Installation von Programmen**
2. **Ein Terminal öffnen**
3. **Einen Editor öffnen**
4. **Postfix Grundkonfiguration**

Inhaltsverzeichnis

In diesem Artikel werden einige spezielle Postfix-Konfigurationen beschrieben, die nicht jeder Betreiber eines Mailservers benötigt, und die deswegen aus Gründen der Übersicht aus dem Hauptartikel ausgelagert wurden. Die einzelnen Abschnitte bauen nicht aufeinander auf und können auch unabhängig voneinander umgesetzt werden. Eine funktionierende **Postfix** [https://wiki.ubuntuusers.de/Postfix/]-Installation [4] wird natürlich vorausgesetzt.

Aliase

Mail-Aliase werden in der Datei **/etc/aliases** festgelegt [3]. Pro Zeile steht ein Adresspaar, wobei Mails, die an die erste (lokale) Adresse adressiert sind, an die zweite notierte Adresse weitergeleitet werden. Als Minimum sollte man Umleitungen für *root* und *postmaster* erstellen. Letzteres wird sogar offiziell per **RFC** [http://de.wikipedia.org/wiki/Request_for_Comments] gefordert, wenn man

einen öffentlichen Mailserver betreibt.

```
root:           adminname
postmaster:     adminname
```

Nach jeder Änderung dieser Datei muss man folgenden Befehl ausführen [2], um die Datenbank zu aktualisieren:

```
sudo newaliases
```

Achtung!

Die Benutzung der **aliases**-Datei unterscheidet sich von der Benutzung der anderen Hash-Dateien von Postfix, um Abwärtskompatibilität zu **sendmail**, dem "Urgestein", und zu anderen Mailservern zu erhalten. Man beachte die Doppelpunkte, den ungewöhnlichen Ort der Datei außerhalb von **/etc/postfix** und den besonderen Befehl **newaliases**.

Generic Mapping

Viele Hosts besitzen keine gültige Internetdomain und verwenden stattdessen eine Bezeichnung wie z.B. **localdomain.local**. Will man Mails über das Internet verschicken, so werden diese von vielen Mail-Servern abgelehnt. Mittels Generic Mapping umgeht man das Problem indem man lokale Mail-Adressen durch reale ersetzt bevor sie via SMTP verschickt werden.

Die Datei **/etc/postfix/main.cf** muss zunächst um diese Zeile ergänzt werden.

```
smtp_generic_maps = hash:/etc/postfix/generic
```

Außerdem muss man **/etc/postfix/generic** erstellen und in etwa so mit Einträgen versehen.

```
Sabrin@server1  SabRi@web.de
HeiBr@server1   Heinz.Brand@gmail.com
```

Die linken Einträge sind die (ungültigen) Mail-Adressen auf dem lokalen Server (hier mit dem Hostnamen server1), die rechten Einträge entsprechen den (gültigen) realen eMail-Adressen.

Das Mapping muss jetzt noch aktualisiert werden.

```
sudo postmap /etc/postfix/generic
```

Zum Schluss wird die Konfiguration neu eingelesen. Fertig.

```
sudo /etc/init.d/postfix reload
```

TLS-(SSL-)Verschlüsselung

Wer einen MTA außerhalb des LANs anbieten (oder selber nutzen) möchte, sollte **TLS-**

Verschlüsselung [http://de.wikipedia.org/wiki/Transport_Layer_Security] (ehemals SSL genannt) anbieten. Dafür benötigt man neben Postfix ein zweiteiliges SSL-/TLS-Zertifikat. Eine einfache Möglichkeit, an ein Dummy-Zertifikat zu kommen, bietet das Paket **ssl-cert** [https://wiki.ubuntuusers.de/ssl-cert/], das hier benutzt wird. Wer den Server nicht nur privat nutzt, sollte sich aber auf jeden Fall ein richtiges, von einer respektierten Zertifizierungsstelle signiertes, Zertifikat besorgen.

Um TLS-Verschlüsselung zu aktivieren, muss man nur folgende Zeilen in die **/etc/postfix/main.cf** eintragen [3]:

```
smtpd_tls_cert_file = /etc/ssl/certs/ssl-cert-snakeoil.pem
smtpd_tls_key_file = /etc/ssl/private/ssl-cert-snakeoil.key
smtpd_use_tls = yes
```

Danach natürlich - wie immer nach einer Änderung an der Konfiguration - Postfix neu starten.

Zusätzlich zu den o.a. Konfigurationsdirektiven gibt es auch noch folgende:

```
smtpd_enforce_tls = yes
```

Diese bewirkt, dass Postfix die Verschlüsselung nicht nur optional anbietet, sondern explizit erzwingt. Unterstützt der Client kein TLS, wird die Verbindung abgelehnt.

Hinweis:

In Postfix 2.3 (ab Edgy) wurden die beiden Direktiven *smtpd_use_tls* und *smtpd_enforce_tls* zusammengefasst zu *smtpd_tls_security_level*, mit den möglichen Werten *may* (Verschlüsselung möglich) und *encrypt* (Verschlüsselung erforderlich). Die alten Schlüsselwörter funktionieren aber aus Kompatibilitätsgründen vorerst weiterhin.

Außerdem ist zu beachten, dass ein Erzwingen der Verschlüsselung von vielen Emailservern nicht beachtet wird und Emails von anderen Emailservern (z.B. GMX) nicht an Ihre lokalen Postfächer zugestellt werden können.

Veraltet: ssmtp

Ganz früher benutzte man SSL/TLS nicht über den normalen SMTP-Port, sondern ohne besondere Protokollverhandlungen über den sog. *ssmtp-* (oder *smtps-*)Port 465. Wer das unterstützen möchte, um auch Uralt-Clients Verschlüsselung zu bieten, muss die **/etc/postfix/master.cf** ändern und den zusätzlichen Service eintragen:

```
smtps     inet  n       -       n       -       -       smtpd
  -o smtpd_tls_wrappermode=yes -o smtpd_sasl_auth_enable=yes
```

Diese zwei Zeilen sind in der Ubuntu-master.cf schon enthalten und es müssen nur die

Kommentarzeichen (#) entfernt werden. Das Zertifikat wird wie oben beschrieben eingebunden.

SASL-Authentifizierung als Server

Fehlerhafte Anleitung

Diese Anleitung ist fehlerhaft. Wenn du weißt, wie du sie ausbessern kannst, nimm dir bitte die Zeit und bessere sie aus.

Anmerkung: In den neuen Versionen von Postfix wurde die Konfiguration grundlegend verändert! Das sorgt dafür, dass die Vorgehensweise sich auch grundlegend verändert hat!

Um einen SMTP-Dienst auch außerhalb des sicheren LANs anbieten zu können, sollte man eine Authentifizierung durch die Clients einfordern, damit man kein *offenes Relay* für Spammer o.ä. anbietet. Dafür wird das *Simple Authentication and Security Layer (SASL)*-Framework verwendet, welches z.B. über das Cyrus-Sasl-Projekt realisiert wird.

Achtung!

Bei dieser Installation werden Benutzerpasswörter im Klartext übertragen. Man sollte also zusehen, für die Verbindung **SSL-/TLS-Verschlüsselung** zu verwenden.

Hierfür müssen folgende Pakete installiert werden [1]:

- **sasl2-bin**
- **libsasl2-2**
- **libsasl2-modules**

Cyrus SASL bietet mehrere Arten, Authentifizierungsdienste bereitzustellen. Da man z.B. mit *auxprop* nicht gegen die normalen Shadow-Passwörter des Linuxsystems authentifizieren kann, wird hier die Einrichtung mit *saslauthd* beschrieben.

Zunächst muss die Konfigurationsdatei **/etc/default/saslauthd** angepasst [3] und folgendes eingetragen werden:

```
START=yes
MECHANISMS="shadow"
```

Dann muss eine neue Datei **/etc/postfix/sasl/smtpd.conf** erstellt werden:

```
pwcheck_method: saslauthd
mech_list: PLAIN LOGIN
saslauthd_path: /var/run/saslauthd/mux
```

Und die **/etc/postfix/main.cf** muss natürlich auch noch angepasst werden:

```
smtpd_sasl_auth_enable = yes
smtpd_recipient_restrictions =
        permit_mynetworks permit_sasl_authenticated reject_unauth_destination

smtpd_helo_restrictions = permit_sasl_authenticated, permit_mynetworks,
reject_invalid_hostname, reject_unauth_pipelining, reject_non_fqdn_hostname

# Postfix >= 2.3, ab Edgy
smtpd_sasl_path = smtpd

broken_sasl_auth_clients = yes
```

Die letzte Zeile ist notwendig, weil sonst ein paar ältere Versionen von Microsoft-Software nicht korrekt funktioniert. Sollte es später beim Anmelden Probleme geben, sollte man die Kommentare `# Postfix >= 2.3, ab Edgy` entfernen.

Gruppenberechtigungen für sasl setzen

Standardmäßig läuft der sasl Dienst mit eigenen Rechten, auf die postfix keinen Zugriff hat. Daher ist es nötig, den user "postfix" der Gruppe "sasl" hinzuzufügen:

```
sudo adduser postfix sasl
```

Sicherheitseinstellungen anpassen

Jetzt könnte eigentlich alles funktionieren, wenn es nicht noch ein Problem gäbe, nämlich die Sicherheitseinstellungen von Postfix. Postfix sperrt nämlich den **smtpd** standardmäßig in eine **chroot** [https://wiki.ubuntuusers.de/chroot/]-Umgebung, wo er den Sasl-Socket nicht finden kann. Es gibt mehrere Möglichkeiten, dieses Problem zu umgehen.

Methode 1: chroot abschalten

Das geht am einfachsten und ist wahrscheinlich auch am praktischsten, wenn noch andere Dienste (wie z.B. ein IMAP-Server) den **saslauthd** nutzen wollen. Dazu muss einfach nur in der Datei **/etc/postfix/master.cf** in der *smtpd*-Zeile ein **n** in die *chroot*-Spalte eingetragen werden:

```
# service type  private unpriv  chroot  wakeup  maxproc command + args
#               (yes)   (yes)   (yes)   (never) (100)
# ==========================================================================
smtp      inet  n       -       n       -       -       smtpd
smtps     inet  n       -       n       -       -       smtpd -o
smtpd_tls_wrappermode=yes -o smtpd_sasl_auth_enable=yes
```

Methode 2: Socket in den chroot legen

Da Postfix seinen Prozess nach **/var/spool/postfix** *chrooted*, sucht dieser dann den Saslauthd-Socket in **/var/spool/postfix/var/run/saslauthd/**. Man muss also dem saslauthd befehlen, den Socket dort zu eröffnen, indem man in der oben erwähnten Datei /etc/default/saslauthd den Eintrag ändert:

```
OPTIONS="-c -m /var/spool/postfix/var/run/saslauthd"
```

Aktivieren

Abschließend müssen Saslauthd und Postfix noch neu gestartet werden [2]:

```
sudo /etc/init.d/saslauthd restart
sudo /etc/init.d/postfix restart
```

Testen

Man kann testen, ob alles klappt, indem man mit dem Telnet-Client direkt mit dem Server kommuniziert. Dafür muss man aber erstmal einen String aus den Zugangsdaten zur späteren Verwendung nach **base64** [http://de.wikipedia.org/wiki/Base64] konvertieren. Im folgenden Beispiel lautet der Benutzername "test" und das Passwort "testtest" und der String muss genau so (mit doppeltem Namen und \0 als Trennzeichen) eingegeben werden:

```
$ perl -MMIME::Base64 -e     'print encode_base64("test\0test
\0testtest");'
dGVzdAB0ZXN0AHRlc3R0ZXN0
$ telnet dapper-lamp 25
Trying 192.168.4.55...
Connected to dapper-lamp.otze.
Escape character is '^]'.
220 dapper-lamp.otze ESMTP Postfix (Ubuntu)
ehlo test
250-dapper-lamp.otze
250-PIPELINING
250-SIZE 10240000
250-VRFY
250-ETRN
250-STARTTLS
250-AUTH LOGIN PLAIN
250 8BITMIME
auth plain dGVzdAB0ZXN0AHRlc3R0ZXN0
235 Authentication successful
quit
```

```
221 Bye
Connection closed by foreign host.
```

Sender-abhängige Authentifizierung

Mit fortschreitender Verbreitung von spam-bekämpfenden Techniken wie **Sender Policy Framework** [https://de.wikipedia.org/wiki/Sender%20Policy%20Framework], mit denen verhindert werden soll, dass Emails von anderen Servern versendet werden, als für ihre Domain zuständig sind (Absenderfälschung), kann es leider passieren, dass das normale Versenden über einen **Smarthost** [https://wiki.ubuntuusers.de/Postfix/#Authentifizierung-am-Smarthost] nicht mehr zuverlässig funktioniert. In diesem Fall muss Postfix die Mails je nach Absender über unterschiedliche SMTP-Server mit individuellen Authentifizierungsdaten verschicken.

Der Nachteil gegenüber der Verwendung eines einzelnen Smarthosts ist, dass die Benutzernamen und Passwörter aller Benutzer auf dem Server hinterlegt werden müssen.

Hinweis:

Diese Konfigurationsmöglichkeit existiert erst ab *Postfix 2.3*. Benutzer von *Ubuntu 6.06 LTS Dapper Drake* müssen sich daher aus dem Backports-Repository aktuellere Pakete besorgen.

Als erstes muss die Datei **/etc/postfix/main.cf** folgendermaßen ergänzt werden:

```
smtp_sender_dependent_authentication = yes
smtp_sasl_auth_enable = yes
smtp_sasl_security_options = noplaintext noanonymous
smtp_connection_cache_on_demand = no
smtp_sasl_password_maps = hash:/etc/postfix/sasl_password
sender_dependent_relayhost_maps = hash:/etc/postfix/sender_dependent
sender_canonical_maps = hash:/etc/postfix/sender_canonical
```

Dann müssen die drei dort referenzierten Postfix-Maps erstellt werden. Ein Beispiel mit drei Benutzern:

- **/etc/postfix/sender_canonical** (LokalerName RealeAbsenderAdresse):

```
erna erna@provider.de
otto otto@otto.de
paul paul@paul.com
```

- **/etc/postfix/sender_dependent** (Absenderadresse SMTP-Server):

```
erna@provider.de mail.provider.de
otto@otto.de smtp.provider.com
paul@paul.com smtp.provider.com
```

- **/etc/postfix/sasl_password** (Absenderadresse Loginname:Passwort):

```
erna@provider.de erna@provider.de:ernaspassword
otto@otto.de otto:ottospassword
paul@paul.com paul:paulspassword
```

Nun müssen diese Maps noch in das Postfix-Datenbankformat konvertiert werden:

```
sudo postmap /etc/postfix/sender_canonical
sudo postmap /etc/postfix/sender_dependent
sudo postmap /etc/postfix/sasl_password
```

Und natürlich muss Postfix seine Konfiguration neu einlesen:

```
sudo /etc/init.d/postfix restart
```

Greylisting

Greylisting [https://de.wikipedia.org/wiki/Greylisting] ist eine Form der SPAM-Bekämpfung. Hierbei wird sich zunutze gemacht, dass normale Mailserver einen gescheiterten Zustellversuch später wiederholen, während Spammer sich im Allgemeinen nicht diese Mühe machen. Der Greylisting-Dienst identifiziert dabei jeden Client über eine Kombination aus IP-Adresse, Absenderadresse und Empfängeradresse. Taucht eine derartige Kombination das erste Mal auf, so wird der Zustellversuch mit einer Fehlermeldung bzgl. eines temporären Problems abgelehnt und diese ID in einer Liste eingetragen. Wird die Kombination erneut zugestellt, wird diese dann vom Mailer akzeptiert.

Nachteil dieser Methode ist, dass die jeweils ersten Emails von neuen Kontakten ein wenig verzögert eintreffen. Vorteil gegenüber herkömmlichen, analyse-gestützten Spamerkennungsverfahren ist, dass die Mail gar nicht erst angenommen und unter Aufwendung von Rechenzeit untersucht werden muss, sondern direkt abgelehnt wird. Gerade auf ausgelasteten Servern stellt das einige günstige Möglichkeit dar, einen Großteil des Mülls gar nicht erst aufs System zu lassen.

Installation

```
$sudo apt-get install postgrey
```

Konfiguration

/etc/postfix/main.cf

```
smtpd_recipient_restrictions = permit_mynetworks,
                               permit_sasl_authenticated,
                               reject_unauth_destination,
                               check_policy_service inet:127.0.0.1:60000
```

Nach einem Postfix-Neustart

```
/etc/init.d/postfix restart
```

oder

```
/etc/init.d/postfix reload
```

ist Postgrey als Policy-Server aktiviert. Unter **/var/log/mail.log** sollten ab sofort Meldungen auftauchen wie z.B.

```
Jul 24 18:17:45 v212910458 postfix/smtpd[24178]: NOQUEUE: reject: RCPT
from unknown[83.234.156.154]: 450 4.7.1
<edwardsm.edwardscv@blubberbla.de>: Recipient address rejected:
Greylisted, see http://isg.ee.ethz.ch/tools/postgrey
/help/blubberbla.de.html; from=<august26joke@askmen.com>
to=<edwardsm.edwardscv@blubberbla.de> proto=ESMTP helo=<home-01590bcaf2>
Jul 24 18:17:46 v212910458 postfix/smtpd[24178]: lost connection after
DATA (0 bytes) from unknown[83.234.156.154]
```

Links

- **Postfix-TLS-Readme** [http://www.postfix.org/TLS_README.html]
- **Postfix-SASL-Howto** [http://www.postfix.org/SASL_README.html]
- **Wikipedia-Artikel "Greylisting"** [http://de.wikipedia.org/wiki/Greylisting]
- **Postfix Webfrontend "VBoxAdm"** [http://developer.gauner.org/vboxadm/]
- **Generic mapping for outgoing SMTP mail** [http://www.postfix.org/ADDRESS_REWRITING_README.html#generic]
- **Fedora Wiki - Generic Mapping** [http://www.fedorawiki.de/index.php/Postfix_minimal]
- **Postfix Monitoring mit Mailgraph und pflogsumm** [http://www.howtoforge.de/anleitung/postfix-uberwachung-mit-mailgraph-und-pflogsumm-auf-debian-etch/]
- **Smtpd “Address family not supported by protocol”** [http://www.nerdsheaven.de/magazin/artikel/tipps-und-tricks/linux-mtasmtpd-sendmail-opensmtp-address-family-not-supported-by-protocol-880/]

Diese Revision [https://wiki.ubuntuusers.de/Postfix/Erweiterte_Konfiguration/a/revision/779153/] wurde am 7. Januar 2015 22:26 von **Swissren** erstellt.
Die folgenden Schlagworte wurden dem Artikel zugewiesen: **Server** [https://wiki.ubuntuusers.de/wiki/tags/Server/], **Netzwerk** [https://wiki.ubuntuusers.de/wiki/tags/Netzwerk/], **Internet** [https://wiki.ubuntuusers.de/wiki/tags/Internet/]