

# Multiple Postfix-Instanzen

Ralf Hildebrandt

T-Systems

3. Mailserver-Konferenz  
Berlin, 2.-3. Juli 2007

# Einsatzgebiete

- `fallback_relay` auf einem Mailrelay oder List-Exploder

# Einsatzgebiete

- `fallback_relay` auf einem Mailrelay oder List-Exploder
- Mail soll auf einem Server mit mehreren IP Adressen zum Raussenden dieselbe IP Adresse wie zum Empfangen benutzen

# Einsatzgebiete

- `fallback_relay` auf einem Mailrelay oder List-Exploder
- Mail soll auf einem Server mit mehreren IP Adressen zum Raussenden dieselbe IP Adresse wie zum Empfangen benutzen
- Unterschiedliche `maximal_queue_lifetime` für eingehende (Backup MX!) und ausgehende Mail

# Einsatzgebiete

- `fallback_relay` auf einem Mailrelay oder List-Exploder
- Mail soll auf einem Server mit mehreren IP Adressen zum Raussenden dieselbe IP Adresse wie zum Empfangen benutzen
- Unterschiedliche `maximal_queue_lifetime` für eingehende (Backup MX!) und ausgehende Mail
- TLS Server mit unterschiedlichen Zertifikaten (Common Name) auf derselben Maschine  
Siehe: Subject Alternatives Problem bei Zertifikaten

# Einsatzgebiete

- `fallback_relay` auf einem Mailrelay oder List-Exploder
- Mail soll auf einem Server mit mehreren IP Adressen zum Raussenden dieselbe IP Adresse wie zum Empfangen benutzen
- Unterschiedliche `maximal_queue_lifetime` für eingehende (Backup MX!) und ausgehende Mail
- TLS Server mit unterschiedlichen Zertifikaten (Common Name) auf derselben Maschine  
Siehe: Subject Alternatives Problem bei Zertifikaten
- individuelle `header_checks` und `body_checks` für ein- und ausgehende Mails

# Einsatzgebiete

- `fallback_relay` auf einem Mailrelay oder List-Exploder
- Mail soll auf einem Server mit mehreren IP Adressen zum Raussenden dieselbe IP Adresse wie zum Empfangen benutzen
- Unterschiedliche `maximal_queue_lifetime` für eingehende (Backup MX!) und ausgehende Mail
- TLS Server mit unterschiedlichen Zertifikaten (Common Name) auf derselben Maschine  
Siehe: Subject Alternatives Problem bei Zertifikaten
- individuelle `header_checks` und `body_checks` für ein- und ausgehende Mails
- Generell: Trennung des ein- und ausgehenden Mailverkehrs

# Vorteile und Nachteile

## Vorteile:

- eigene Queues für jede Instanz (z.B. “vor `content_filter`” und “hinter `content_filter`”)



# Vorteile und Nachteile

## Vorteile:

- eigene Queues für jede Instanz (z.B. “vor `content_filter`” und “hinter `content_filter`”)
- eigene Einstellungen für jede Instanz (`main.cf`, `master.cf`)

## Nachteile:

- etwas unübersichtlich (Wo bin ich?)

# Vorteile und Nachteile

## Vorteile:

- eigene Queues für jede Instanz (z.B. “vor `content_filter`” und “hinter `content_filter`”)
- eigene Einstellungen für jede Instanz (`main.cf`, `master.cf`)

## Nachteile:

- etwas unübersichtlich (Wo bin ich?)
- dadurch höhere Fehleranfälligkeit



Konfuzius sagt:  
Postfix zum Laufen kriegen – erst dann eine zweite Instanz  
hinzufügen!

Das Hinzufügen einer zweiten Instanz besteht aus den folgenden Schritten:

- Eine neue IP Adresse hinzufügen

Das Hinzufügen einer zweiten Instanz besteht aus den folgenden Schritten:

- Eine neue IP Adresse hinzufügen
- `/etc/postfix` nach `/etc/postfix-out` duplizieren

Das Hinzufügen einer zweiten Instanz besteht aus den folgenden Schritten:

- Eine neue IP Adresse hinzufügen
- `/etc/postfix` **nach** `/etc/postfix-out` **duplizieren**
- `/var/spool/postfix-out` **anlegen**

Das Hinzufügen einer zweiten Instanz besteht aus den folgenden Schritten:

- Eine neue IP Adresse hinzufügen
- `/etc/postfix` nach `/etc/postfix-out` duplizieren
- `/var/spool/postfix-out` anlegen
- Zum Laufen bringen

Das Hinzufügen einer zweiten Instanz besteht aus den folgenden Schritten:

- Eine neue IP Adresse hinzufügen
- `/etc/postfix` nach `/etc/postfix-out` duplizieren
- `/var/spool/postfix-out` anlegen
- Zum Laufen bringen
- Konfiguration entzerren



Das Hinzufügen einer zweiten Instanz besteht aus den folgenden Schritten:

- Eine neue IP Adresse hinzufügen
- `/etc/postfix` nach `/etc/postfix-out` duplizieren
- `/var/spool/postfix-out` anlegen
- Zum Laufen bringen
- Konfiguration entzerren
- Skripte erstellen

# Eine neue IP Adresse hinzufügen

Wir legen eine neue IP Adresse an - unsere neue Postfix Instanz soll Mail von innen nach außen transportieren:

```
# ifconfig eth0:1 10.0.0.1 up
```

# /etc/postfix nach /etc/postfix-out duplizieren

```
# cp -a /etc/postfix /etc/postfix-out
```

Nun muß man /etc/postfix-out/main.cf so ändern, daß eine eigene Queue benutzt wird:

```
queue_directory = /var/spool/postfix-out
```

# Die `-c` Option

Mit der `-c /config/directory` Option werden Postfixs Komponenten angewiesen, ander Konfigurationsverzeichnisse zu nutzen.

Zum Beispiel:

```
# postfix -c /config/directory flush
```

# /var/spool/postfix-out anlegen

Bevor wir das vergessen - Verzeichnis anlegen und bevölkern:

```
# mkdir /var/spool/postfix-out  
# postfix -c /etc/postfix-out check
```

# Mißtrauisch?

```
# ll /var/spool/postfix-out/  
total 56  
drwx----- 2 postfix root      4096 2007-06-05 14:59 active  
drwx----- 2 postfix root      4096 2007-06-05 14:59 bounce  
drwx----- 2 postfix root      4096 2007-06-05 14:59 corrupt  
drwx----- 2 postfix root      4096 2007-06-05 14:59 defer  
drwx----- 2 postfix root      4096 2007-06-05 14:59 deferred  
drwx----- 2 postfix root      4096 2007-06-05 14:59 flush  
drwx----- 2 postfix root      4096 2007-06-05 14:59 hold  
drwx----- 2 postfix root      4096 2007-06-05 14:59 incoming  
drwx-wx--- 2 postfix postdrop 4096 2007-06-05 14:59 maildrop  
drwxr-xr-x 2 root      root      4096 2007-06-05 14:59 pid  
drwx----- 2 postfix root      4096 2007-06-05 14:59 private  
drwx--x--- 2 postfix postdrop 4096 2007-06-05 14:59 public  
drwx----- 2 postfix root      4096 2007-06-05 14:59 saved  
drwx----- 2 postfix root      4096 2007-06-05 14:59 trace
```

# chroot vergessen?

Wenn man in der `master.cf` für Daemonen `chroot` konfiguriert hat, so muss natürlich das `chroot` unterhalb von `/etc/postfix-out` ebenfalls angelegt werden...

# Postfix in Kenntnis setzen

```
# postconf -e "alternate_config_directories = /etc/postfix-out"
```

So weiß Postfix welche Verzeichnisse mit dem Kommandozeilenparameter `-c /config/directory` oder über die Umgebungsvariable `MAIL_CONFIG` angegeben werden können.



# Netzwerkinterfaces gerecht aufteilen

In der `/etc/postfix-out/master.cf`

```
10.0.0.1:smtp      inet  n       -       n       -
```

Das ist nötig, damit die `smtp`-Daemonen der `postfix-out`-Instanz auch an offizielle IPs außerhalb des `10/8` Netzes senden können!

In der ursprünglichen Instanz:

```
# postconf \  
-e "inet_interfaces = 88.198.105.204"
```

# Start!

```
# postfix -c /etc/postfix-out start
# postfix start
# tail -f /var/log/mail.log
example postfix/postfix-script: starting the Postfix mail system
example postfix/master[16288]: daemon started -- version 2.3.8, \
configuration /etc/postfix-out
example postfix/postfix-script: starting the Postfix mail system
example postfix/master[16357]: daemon started -- version 2.3.8, \
configuration /etc/postfix
```

**Aber: Die Logeinträge kann man ja gar nicht auseinanderhalten!**

# Besseres Logging

```
# postconf -c /etc/postfix-out \  
-e "syslog_name = postfix-out"  
  
# postfix -c /etc/postfix-out stop  
postfix-out/postfix-script: stopping the Postfix mail system  
  
# postfix -c /etc/postfix-out start  
postfix-out/postfix-script: starting the Postfix mail system  
  
# tail -f /var/log/mail.log  
example postfix-out/postfix-script: starting the Postfix mail system  
example postfix-out/master[5440]: daemon started -- \  
version 2.3.8, configuration /etc/postfix-out
```

# mailq

```
# mailq -c /etc/postfix-out  
mailq: fatal: display queue mode requires no recipient
```

**Och nee! Das nervt!**

```
# MAIL_CONFIG=/etc/postfix-out mailq  
Mail queue is empty
```

**Dafür kann man sich ja mal ein Skript mailq-out bauen:**

```
#!/bin/sh  
MAIL_CONFIG=/etc/postfix-out /usr/bin/mailq
```

# mailq II

```
# mailq -C /etc/postfix-out  
Mail queue is empty
```

## Und wenn man schon mal dabei ist. . .

... dann sollte man Wrapper für die wichtigsten Befehle schreiben:

- `sendmail`

denn das sollte die postfix-out Instanz nutzen

## Und wenn man schon mal dabei ist. . .

... dann sollte man Wrapper für die wichtigsten Befehle schreiben:

- `sendmail`  
denn das sollte die postfix-out Instanz nutzen
- `postconf-out`  
editiert die Konfig der postfix-out Instanz

## Und wenn man schon mal dabei ist. . .

... dann sollte man Wrapper für die wichtigsten Befehle schreiben:

- `sendmail`  
denn das sollte die postfix-out Instanz nutzen
- `postconf-out`  
editiert die Konfig der postfix-out Instanz
- `postfix-out`  
steuert die postfix-out Instanz



# Konfiguration entzerren

Auf unserer “postfix-out” Instanz:

- können alle möglichen `smtpd*_checks` wegfallen

# Konfiguration entzerren

Auf unserer “postfix-out” Instanz:

- können alle möglichen `smtpd*_checks` wegfallen
- kann evtl. der `content_filter` wegfallen

... da sie ja nur aus den eigenen Netzen erreichbar ist!

# Startup-Skripte anpassen

```
if start-stop-daemon --start --exec \  
  ${DAEMON} -- quiet-quick-start; then  
  log_end_msg 0  
else  
  log_end_msg 1  
fi
```

**ergänzen um:**

```
if start-stop-daemon --start --exec \  
  ${DAEMON} -c /etc/postfix-out -- quiet-quick-start; then  
  log_end_msg 0  
else  
  log_end_msg 1  
fi
```

# Startupskripte anpassen II

Wenn man mehr als zwei Instanzen hat, ist vielleicht eine Lösung mit einer `for`-Schleife sinnvoll (Pseudocode):

```
for config_dir in /etc/postfix* do
    ${DAEMON} -c ${config_dir} start
done
```

# Für `smtp_fallback_relay`

Die erste Instanz nutzt kürzere Timeouts und unsere zweite Instanz als `smtp_fallback_relay`.  
Somit unternimmt die erste Instanz die initialen, hoffentlich “schnellen” Zustellungen und die zweite kriegt den Bodensatz.

# Für smtp\_fallback\_relay - Implementation I

smtp\_fallback\_relay **einstellen:**

- smtp\_fallback\_relay = [10.0.0.1]:25

# Für `smtp_fallback_relay` - Implementation II

## Erlaubte Syntax:

- `domain`

# Für smtp\_fallback\_relay - Implementation II

## Erlaubte Syntax:

- domain
- host



# Für smtp\_fallback\_relay - Implementation II

## Erlaubte Syntax:

- domain
- host
- host:port

# Für `smtp_fallback_relay` - Implementation II

## Erlaubte Syntax:

- `domain`
- `host`
- `host:port`
- `[host]:port`

# Für smtp\_fallback\_relay - Implementation II

## Erlaubte Syntax:

- domain
- host
- host:port
- [host]:port
- [address]

# Für smtp\_fallback\_relay - Implementation II

## Erlaubte Syntax:

- domain
- host
- host:port
- [host]:port
- [address]
- [address]:port

Die Form [host] schaltet MX Lookups ab.

# Für `smtp_fallback_relay` - Implementation III

Timeouts verringern (Beispiele für mögliche Modifikationen):

- `smtp_connect_timeout = 5s`

# Für `smtp_fallback_relay` - Implementation III

Timeouts verringern (Beispiele für mögliche Modifikationen):

- `smtp_connect_timeout = 5s`
- `smtp_data_done_timeout = 100s`

# Für smtp\_fallback\_relay - Implementation III

Timeouts verringern (Beispiele für mögliche Modifikationen):

- `smtp_connect_timeout = 5s`
- `smtp_data_done_timeout = 100s`
- ...

# Zweite Instanz

Betrieb mit Standardeinstellungen, aber:

- `smtpd` horcht nur auf `10.0.0.1:25`



# Zweite Instanz Implementation

## In der master.cf

```
# =====  
# service type private unpriv chroot wakeup maxproc command  
#           (yes)   (yes)   (yes)   (never) (50)  
# =====
```

```
smtp      inet  n       -       n       -       -       smtpd
```

### ändern in

```
10.0.0.1  inet  n       -       n       -       -       smtpd
```

# Aber...

```
postfix/smtp[5132]: warning: host unknown[10.0.0.1]:25 greeted me
with my own hostname hostname.example.org
postfix/smtp[5132]: warning: host unknown[10.0.0.1]:25 replied to
HELO/EHLO with my own hostname hostname.example.org
```

**Ergo in der zweiten Instanz den Hostnamen ändern:**

```
myhostname = example-out.example.org
```

**Aber man sollte im praktischen Einsatz aber auch darauf achten, dass dieser nun nicht im HELO/EHLO auftaucht:**

```
smtp_helo_name = example.example.org
```

# Jetzt aber...

```
postfix-out/smtpd[5157]: NOQUEUE: reject: RCPT from  
unknown[10.0.0.1]: 554 5.7.1 <recipient@example.example.org>:  
Relay access denied; ...
```

**Ergo in der zweiten Instanz die IP der ersten in `mynetworks` ergänzen:**

```
mynetworks = 127.0.0.0/8, 10.0.0.1
```

# Für content\_filter

Die erste Instanz gibt die Mail an den `content_filter` ab.  
Der `content_filter` gibt danach die Mail an die zweite Instanz ab.

# amavisd-new anpassen

```
$forward_method = 'smtp:[10.0.0.1]:25';
```

Wir müssen die `$forward_method` ändern, damit unsere zweite Instanz auf `10.0.0.1` genutzt wird.

# Das war's auch schon I

```
postfix/pickup[5654]: 19519B48F1: uid=0 from=<hildeb@charite.de>
postfix/cleanup[5659]: 19519B48F1: \
  message-id=<20070626085908.19519B48F1@fatush.arschkrebs.de>
postfix/qmgr[5655]: 19519B48F1: from=<hildeb@charite.de>, \
  size=294, nrcpt=1 (queue active)
postfix/lmtp[5662]: 19519B48F1: to=<hildeb@charite.de>,
  relay=127.0.0.1[127.0.0.1]:10024, delay=2.3,
  delays=0.17/0.03/0.04/2.1, dsn=2.6.0, status=sent (250 2.6.0 Ok,
  id=05632-01, from MTA([10.0.0.1]:25): 250 2.0.0 Ok: queued as
  DB1D2B80EB)
postfix/qmgr[5655]: 19519B48F1: removed
```

# Das war's auch schon II

```
postfix-out/smtpd[5664]: DB1D2B80EB: client=unknown[10.0.0.1]
postfix-out/cleanup[5668]: DB1D2B80EB: \
  message-id=<20070626085908.19519B48F1@fatush.arschkrebs.de>
postfix-out/qmgr[5647]: DB1D2B80EB: from=<hildeb@charite.de>, \
  size=689, nrcpt=1 (queue active)
amavis[5632]: (05632-01) Passed CLEAN, <hildeb@charite.de> ->
  <hildeb@charite.de>, Message-ID:
  <20070626085908.19519B48F1@fatush.arschkrebs.de>, mail_id:
  VEz7pjPs2ca7, Hits: -, queued_as: DB1D2B80EB, 2114 ms
```

# Fragen?

Die Folien gibt es unter

<http://www.arschkrebs.de/slides/mi-slides.pdf>