

# Mailserver:

## Best Practice für stressfreie Mailserver

- ▶ False positives, Bounces, verlorene Mails, genervte Nutzer:

Meistens liegt die Ursache im versendenden Mailserver.

- ▶ Ob ich anderswo gefiltert werde, kann ich maßgeblich selbst beeinflussen!
- ▶ Dieser Vortrag beleuchtet die Bereiche:
  - ▶ Netzstrategie
  - ▶ Konfigurationsfehler
  - ▶ Fehlerhafte E-Mails
  - ▶ Spam-Strategie

Netzstrategie:

## Zu späte Bounces („Backscatter“)

- ▶ Mailrelays nehmen Mails an und bouncen sie später:
  - ▶ Weil erst ein nachgeschalteter Spam-/Virenfilter blockt
  - ▶ Weil erst später festgestellt wird, daß der Empfänger unzustellbar ist
- ▶ Spammer fälschen Absender!
  - ▶ Man belästigt unschuldige Dritte mit Bounces
  - ▶ Man landet (zu recht) auf RBL-Sperrlisten
  - ▶ Man hat eine verstopfte Outbound-Queue
- ▶ Ergebnis: Ständig Störungen im eigenen Versand (=DoS!)
- ▶ Besser:  
Spam-/Viren müssen sofort in Echtzeit geprüft werden!  
Das erste Mailrelay muss nicht-existente Empfänger blocken!

Netzstrategie:

# Mailrelays und MX-20 ohne Empfängerliste

- ▶ Auch ein Backup-Mailserver („MX 20“) und extern betriebene Mailrelays müssen Empfänger prüfen
  - ▶ Durch eine statische Empfängerliste (Datei, SQL, LDAP)
  - ▶ Durch dynamische Empfängervalidierung (Kostet bei Postfix exakt einen Parameter „reject\_unverified\_recipient“ in den Restrictions und funktioniert hervorragend!)
- ▶ Auch ein Dienstleister muß Empfänger-Validierungen machen!
  - ▶ Sonst hat er Backscatter-Probleme => RBL-Sperrlisten drohen!
  - ▶ Dynamische Validierung ermöglicht das, selbst wenn seine Kunden die Empfängerliste nicht herausgeben wollen.
  - ▶ Es gibt also keinen Grund dies nicht zu tun.

Konfigurationsfehler:

# Backup-Server ist schlecht konfiguriert

- ▶ Spam muß gefiltert werden, wenn man mit dem Spammer redet
- ▶ Backup-Server sind oft offen wie ein Scheunentor.
  - ▶ Insb. als Backup-Relay nehmen sie oft Mails an, die sie für sich selbst nie empfangen würden.
  - ▶ Ursache sind i.d.R. Denkfehler in den Restrictions: `permit_mx_backup` darf nicht vor Spamschutzmaßnahmen wie RBL oder Greylisting kommen!
  - ▶ Nimmt der Backup-Server den Spam erstmal an, greifen Clientbasierte Filtermechanismen nicht mehr oder nur noch sehr schlecht.
- ▶ Ergebnis: Der Backup-Server „säubert“ den Spam. Er hilft dem Spammer und fällt dem MX-10-Server „in den Rücken“.
- ▶ Besser: MX-20 und MX-10 müssen identisch konfiguriert sein

Netzstrategie:

## Wozu überhaupt einen Backup-MX?

- ▶ Ein externer MX-20 kann i.d.R. nur store+forward machen. Ist das Hauptrelay wieder online, stellt er dorthin zu.
  - ▶ Ein MX-20 hilft nur, wenn man selbst auch Zugriff auf diesen Server hat um Notfallmaßnahmen zu ergreifen.
  - ▶ Ansonsten ist er ebenso eine Blackbox, wie die ebenso queuenden Mailserver anderer Provider, die die Mails ja auch speichern würden.
  - ▶ Es ist NICHT im Interesse des Absenders tagelang im Ungewissen zu sein, ob seine Mail ankam. Eine Erhöhung der Queue-Lifetime ist sinnlos.
  - ▶ Es ist haftungsrechtlich bedenklich, dem Absender eine Zustellung mit „250 OK“ zu quittieren, wenn die Mail tagelang nicht empfangen wurde.
- ▶ Ergebnis: MX-20 sind überflüssig, wenn sie im Fehlerfall nicht am MX-10 vorbei die E-Mails ausliefern können.
- ▶ Besser: Zwei vollwertige MX-10. Lieber gar keinen MX-20.

Netzstrategie:

## Round-Robin im DNS

- ▶ Wird ein DNS-Round-Robin-A-Record auf zwei Incoming-Mailrelays gesetzt, können einliefernde Server nicht erkennen, daß zwei/mehrere Server zur Verfügung stehen
  - ▶ Im Fehlerfall wird nicht auf den anderen Server ausgewichen
  - ▶ Erneute Zustellversuche einer unzustellbaren Mail gehen i.d.R. an die gleiche IP und werden nicht über Round-Robin verteilt!
  - ▶ Ergo: Im Falle eines Ausfalles sind 50% der Mails nicht zustellbar.
  - ▶ Andere Mailserver können Performance / Verbindungsanzahl nicht optimieren, da sie glauben, es stünde nur ein Ziel zur Verfügung.
- ▶ Ergebnis: Gewonnene Sicherheit? Keine. Ganz im Gegenteil.
- ▶ Besser: Sauber zwei MX-10-Records setzen. Dazu sind die da.

Netzstrategie:

## Loadbalancer vor Mailservers

- ▶ Der Einsatz von Loadbalancern vor Mailservers ist überflüssig und Zeit-/Geldverschwendung
  - ▶ Mailservers beherrschen über MX 10 / MX 20 – Einstellungen perfekt Lastverteilung und Failover-Ausfallsicherheit!
  - ▶ Er ist sogar kontraproduktiv: Die Loadbalancer können Fehler auf SMTP-Protokollebene i.d.R. nicht unterscheiden, der defekte Server kriegt immer wieder Verbindungen zugeteilt. Andere Mailservers würden das erkennen.
  - ▶ Und wieder: Andere Server können Performance nicht optimieren.
  - ▶ Notwendige Ausnahme: Die überteuerte Appliance im LAN kann noch nicht mal mit MX-Routing an mehrere Relayhosts umgehen (Tipp: Appliance verschrotten).
- ▶ Ergebnis: Geldverschwendung bei gleichzeitig schlechterer Stabilität und Performance.
- ▶ Besser: MX-Records funktionieren sauber, stabil, zuverlässig. Mailservers sollen selbst entscheiden können.

Netzstrategie:

## Kein SMTP-Proxy vor Postfix

- ▶ Oft SMTP-Proxy der Firewall vor Postfix.
  - ▶ Eine eierlegende „ich-kann-alle-Protokolle“ Wollmilchsau als Appliance soll den auf SMTP optimierten, von IBM Sicherheitsspezialisten Wietse Venema über 10 Jahre hinweg erprobtem Postfix schützen?! Wohl eher andersherum.
  - ▶ Namhafte Hersteller hatten teilweise mehr relevante SMTP-Bugs als Postfix.
  - ▶ SMTP-Proxies haben oft krasse Protokollfehler. Verfälschte Authentifizierungen, zerstörte Timeouts, vorspiegelung nicht-existenter IP-Verbindungen.
  - ▶ Und diese ich-habe-keine-RFCs-gelesen-Appliance soll Postfix schützen?!
  - ▶ Appliances sind oft von Leuten geschrieben, für die das Netz durch http und damit durch Client <-> Server definiert wird.
  - ▶ SMTP ist aber viel komplexer. Hier sind selbst-agierende Server auf jeder Seite.
- ▶ Ergebnis: Gerade die möchtegertransparenten buggy SMTP-Proxies verursachen Ausfälle und Ärger. Sie schützen nicht.
- ▶ Besser: Postfix direkt. Sauber aufgesetzt, sauber abgesichert.



Konfigurationsfehler:

## Kein/falscher Reverse-Lookup

- ▶ Die Plausibilität zwischen HELO und DNS-Reverse-Lookup eines Servers ist ein wichtiges Filterkriterium.
  - ▶ Spammer wollen/müssen diese Angaben oft fälschen.
- ▶ Viele Mailserver haben keine/falsche DNS-Reverse-Lookups.
  - ▶ HELO/Hostname ist mail.firma.de,  
Reverse-PTR ist aber port-195-158-185-250.static.qsc.de
  - ▶ Apache ohne Reverse-Lookup? Völlig egal.  
Mailserver ohne richtigen Reverse-Lookup? Katastrophe.
- ▶ Ergebnis: Ablehnung oder latent schlechtere Bewertung bei anderen ISPs ist vorprogrammiert – aber hausgemacht!
- ▶ Besser: Hostname und Reverse-PTR müssen übereinstimmen.
  - ▶ Richtig, dynamische IPs und Mailserver vertragen sich nicht.

Konfigurationsfehler:

## Nicht auflösbarer HELO / intra-Domains

- ▶ Problem: Mailserver hat einen im DNS nicht-existenten Hostnamen
  - ▶ mailgate.intra oder mailout.firma.de – der aber nicht auflösbar ist
  - ▶ Server agiert mit nicht auflösbarem HELO
  - ▶ Einige Systeme filtern schon dies (ich bin dagegen)
  - ▶ Nicht-auflösbare Namen machen nur Ärger, Ärger, Ärger.
  - ▶ Konvertierungsärger kommt auch noch dazu
  - ▶ Ich kenne kein Setup, wo dies bei scharfem Nachdenken wirklich SINN machen würde
- ▶ Ergebnis: Reverse-Lookup und HELO passen nicht zusammen.
- ▶ Besser: Immer saubere Hostnamen passend zum Reverse-PTR im DNS benutzen.

Fehlerhafte Mails:

# Mails ohne Zeichensatzdefinition erzeugen

- ▶ Mails werden einfach so ohne Zeichensatzdefinition erzeugt.
  - ▶ Betrifft: Web-Formulare, Autoresponder, Buchungsbestätigungen
  - ▶ Richtig:

```
Content-Type: text/plain;  
charset="iso-8859-1"
```

- ▶ Programmierer testen anscheinend nur nach Trial & Error. Problem: Sie lesen mit dem Zeichensatz, mit dem sie auch das Webformular befüllt haben.
- ▶ Ergebnis: Je nach Nutzer-Zeichensatz Umlautefehler
- ▶ Besser: Auf jeden Fall charset-Header setzen. (RFC 2822!)

Fehlerhafte Mails:

# Unkodierte Umlaute im Subject

- ▶ Subject gehört zum Mailheader – nur 7 Bit erlaubt!
  - ▶ Umlaute müssen kodiert werden:  
Falsch: Subject: Schöne Grüße  
Richtig: Subject: =?iso-8859-1?b?U2No9m5l? =?iso-8859-1?b?lEdy/N9l? =
  - ▶ Betrifft: Web-Formulare, Autoresponder, Buchungsbestätigungen
- ▶ Ergebnis: Schlechtere Bewertung in Spam-Filtern, peinliche Außendarstellung bei Kunden.
- ▶ Besser: RFC 2822 lesen, mit verschiedenen Clients, Zeichensätzen, Betriebssystemen testen

Fehlerhafte Mails:

## HTML ohne MIME-Struktur

- ▶ Haben wir mühsam den Mailclients abgewöhnt: Kein HTML ohne MIME-Kapselung und alternativem ASCII-Text
  - ▶ Automatisch generierte Mails haben oft einfach HTML-Tags in die Mail
  - ▶ Schon gesehen: HTML im Subject!
  - ▶ Betrifft: Web-Formulare, Autoresponder, Buchungsbestätigungen
- ▶ Ergebnis: Schlechtere Bewertung in Spam-Filtern, peinliche Außendarstellung bei Kunden.
- ▶ Besser: RFC 2822 lesen, mit verschiedenen Clients, Zeichensätzen, Betriebssystemen testen

Fehlerhafte Mails:

# Webserver erzeugen kaputte Absender

- ▶ Mails werden in PHP an `/usr/bin/sendmail` übergeben
  - ▶ Mails werden unter der Apache-User-ID versandt (`wwwrun`)
  - ▶ Setzen die Programmierer nicht explizit einen korrekten Absender, werden die Mailadressen im Envelope falsch gesetzt: `wwwrun@www.firma.de`
- ▶ Ergebnis: Bounces versacken im Nirvana
- ▶ Besser: Über die `php.ini` oder `httpd.conf` einen Default-Absender vorgeben.

```
▶ php_admin_value sendmail_from user@domain.de
▶ php_admin_value sendmail_path "sendmail -t -i -f user@domain.de"
```

# Webformulare ohne Eingabevalidierung

- ▶ Viele Programmierer übernehmen den Input der Formulare ungeprüft in From:, To: oder Subject: der Mail
  - ▶ Zeilenumbrüche leiten den Body der Mail ein und haben katastrophale Auswirkungen
  - ▶ Spammer befüllt das From:-Feld mit einem Absender, einem Zeilenumbruch, 50.000 To:-Zeilen, einer Leerzeile und seiner Spam-Mail.
  - ▶ Grundsätzlich dürfen Webformulare sowieso nicht die Empfänger frei festlegen lassen.
- ▶ Ergebnis: Webserver werden zur üblen Spam-Schleuder
- ▶ Besser: Daten, die beim Nutzer waren, sind NIE vertrauenswürdig und müssen IMMER validiert werden.

Spam-Strategie:

# Spam-Tagging sorgt für Mailverlust

- ▶ Theorie: Spam-Tagging verhindert false positives
  - ▶ Praxis: Anwender müßten eigentlich (!) alle (!) getaggtten Mails kontrollieren
  - ▶ Tun sie nicht. Also: Empfänger bekommt den false positives nicht mit
  - ▶ Aber: Absender bekommt false positives ebenfalls nicht mit
  - ▶ Problem fällt erst auf, wenn es zu spät ist und der Ärger da ist
  - ▶ Möglichst viel Chaos und Streß für alle Beteiligten
- ▶ Besser: Nicht taggen, sondern live hart rejecten.
  - ▶ Eine Appliance, die das nicht kann, ist keine 10 Cent wert.
- ▶ Ergebnis: False Positives sind nach 10 Sekunden wieder beim Absender, er weiß Bescheid und kann reagieren.
  - ▶ Nur das verhindert, daß false positives zu Mailverlusten führen
  - ▶ Wichtig: Zivilrechtliche Haftungsfragen wenn Mail angenommen, aber „verbummelt“ wird!



Spam-Strategie:

# Hardware statt Brainware

- ▶ Viel Geld wird in Hardware-Redundanz oder 24/7-SLA gesteckt
  - ▶ In aller Regel gibt es aber nur einen Administrator, der das System wirklich gut beherrscht ...  
...und nur selten einen Administrator, der es wirklich perfekt beherrscht.
  - ▶ Redundanz des Administrators? Fehlanzeige.
- ▶ Aber: Wieviele Ausfälle entstehen durch Hardware-Defekt?
- ▶ Aber: Wieviele Ausfälle entstehen durch Fehlkonfiguration?
  - ▶ Die meisten Ausfälle entstehen vor der Tastatur! Absicherung?
- ▶ Ergebnis: Jedes System ist nur so gut, wie es administriert wird.
- ▶ Besser: Brainware statt Hardware!
  - ▶ Hardware-Verzicht und stattdessen Fortbildung!  
Mehr Verfügbarkeit, bessere Ergebnisse.

Spam-Strategie:

## Teuer = Gut?

- ▶ **Niemand kann zaubern. Alle kochen nur mit Wasser.**
  - ▶ Es gibt keinen Zusammenhang zwischen guten Filterergebnissen und teuren (überteuerten) Preisen. Untersuchungen zeigen erschreckende Ergebnisse.
  - ▶ Hersteller haben oft mehr Interesse daran dem Kunden die Spamflut vor Augen zu führen, als sie effektiv und sorgenfrei zu vermeiden („Die verwalten Spam“)
  - ▶ Oft sind Dienstleister wie überteuerte Appliances noch nicht mal rechtskonform und können noch nicht mal schon während des Annahmeprozesses filtern.
  - ▶ Jedes System ist nur so gut, wie es auch administriert wird. Kaufen, einschalten und vergessen bringt nichts.
- ▶ **Ergebnis: Firmen zahlen Irrsinnssummen ohne Realitätsbezug.**
  - ▶ 30.000 € für eine kleine 3.500-Personen-Firma ist zu viel.
- ▶ **Besser: Eine sinnvolle Lösung erreichen**
  - ▶ Die dann ggf. auch gerne durch einen Dienstleister perfekt gepflegt werden kann.
  - ▶ Gespartes Geld bringt investiert in eine Admin-Schulung höhere Absicherung

Quintessenz:

## Ergebnis der ganzen Sache?

- ▶ Mailserver müssen viel penibler, RFC-konformer und fachlich versierter konfiguriert werden, als bspw. ein Webserver
  - ▶ Auch guten Postmastern fehlt häufig noch der Überblick über alle Abhängigkeiten und Besonderheiten
- ▶ Weniger ist mehr: Kein Firlefanz vor dem MTA
  - ▶ Weg mit Sachen, die so nie für SMTP vorgesehen waren
- ▶ Sauberes DNS und saubere Hostnamen sind elementar wichtig
  - ▶ Keine Experimente, keine Spielereien – aber auch keine Schlampereien!
- ▶ Lieber Know-how als Hardware einkaufen
  - ▶ Versierte Administration bringt mehr als Appliance-Gläubigkeit

Bonus:

# Private Mailnutzung am Arbeitsplatz

- ▶ Extrem heikle Gesetzeslage für den Arbeitgeber
  - ▶ Arbeitgeber ist Provider ggü. Arbeitnehmer
  - ▶ Spamschutzschwierigkeiten bei privaten Mails
  - ▶ Keine Archivierung privater Mails
  - ▶ Schwieriger/Kein Zugriff auf die geschäftlichen Mails des Postfachs
  - ▶ Anscheinsvollmacht: Private Mails mit Geschäftssignatur!
  - ▶ Extrem hohe kosten für „billige“ Privatmails (HA, Lizenzen, SAN, Backup)
- ▶ Ergebnis: Private Nutzung der Mailadressen unverantwortlich
- ▶ Besser: Komplette getrennte Infrastruktur für private Mails
  - ▶ Nur Viren, kein Spamschutz. Zugriff über Webmail. Einfaches System ohne Groupware (Postfix/Dovecot). Billiger SATA-Storage. Nur Grund-Backup.
  - ▶ Defacto: Eigene Subdomain [user@privat.firma.de](mailto:user@privat.firma.de) zur sauberen Trennung nötig!

- ▶ Bis hierhin...
- ▶ ...vielen Dank fürs Zuhören,  
ich freue mich über weitere Diskussionen...
- ▶ ...und nun viel Spaß auf der weiteren SLAC!