HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

SPF, DKIM und Greylisting

Heinlein Professional Linux Support GmbH

Holger Uhlig
<h.uhlig@heinlein-support.de>

Peer Heinlein
<p.heinlein@heinlein-support.de>

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Agenda:

- Sender Policy Framework
  - was / wie / warum
  - Sender Rewriting Scheme
- Domain Key Identified Mail
  - was / wie / warum
- Greylisting
  - Mythen, Legenden und Visionen
- Abschluss

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

SPF:
Sender Policy Framework

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

SPF: Sender Policy Framework
Die Idee

- Das Problem:
  Spammer können Absender beliebig fälschen.
  - SMTP sieht keine Verifizierung eines Absenders vor
  - MX-Records im DNS regeln nur Empfangs, nie aber Versandserver
- Die Lösung:
  Versandserver einer Domain festlegen
  - So könnte geprüft werden, wer Mails mit einem Absender versenden darf
  - Könnte Absenderfälschungen wirksam eindämmen

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

SPF: Sender Policy Framework
Die technische Umsetzung

- Versandserver im DNS bekannt machen
  - TXT-Feld im DNS ist ungenutzt, kann „mißbraucht“ werden
  - eigene SPF-Records seit RFC 4408 (bind 9.4) vorgesehen
  - Prüfung erfolgt auf die *MAIL-FROM / HELO* Angaben
- Aber:
  - flächendeckende DNS-Records nur langwierig einzuführen
  - SPF-Record langsame Verbreitung, da neuere Software notwendig

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

SPF: Sender Policy Framework
Das Schema

- Version
  - v=spf1
- Qualifikatoren
  - \+ (Pass) die Direktive definiert autorisierte Sender (Defaultwert)
  - \- (Fail) die Direktive definiert nicht autorisierte Sender
  - ~ (SoftFail) die Direktive definiert nicht autorisierte Sender, der Empfänger soll diesen Fehlschlag aber großzügig behandeln (gesenkter Score)
  - ? (Neutral) über *nicht* genannte Server wird *keine* Aussage gemacht

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

SPF: Sender Policy Framework
Das Schema

- Mechanismen
  - all — immer
  - a — ist Sender-IP Element der befragten Domäne
  - mx — ist Sender-IP ein MX-Record der befragten Domäne
  - ip4 — Sender-IP ist die angegebene IP-Adresse oder Element des IP-Subnetztes
  - include — zusätzliche SPF-Anfrage zur im Include enthaltenen Domain
  - exists — Check auf einen A-Record oder ein Macro

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

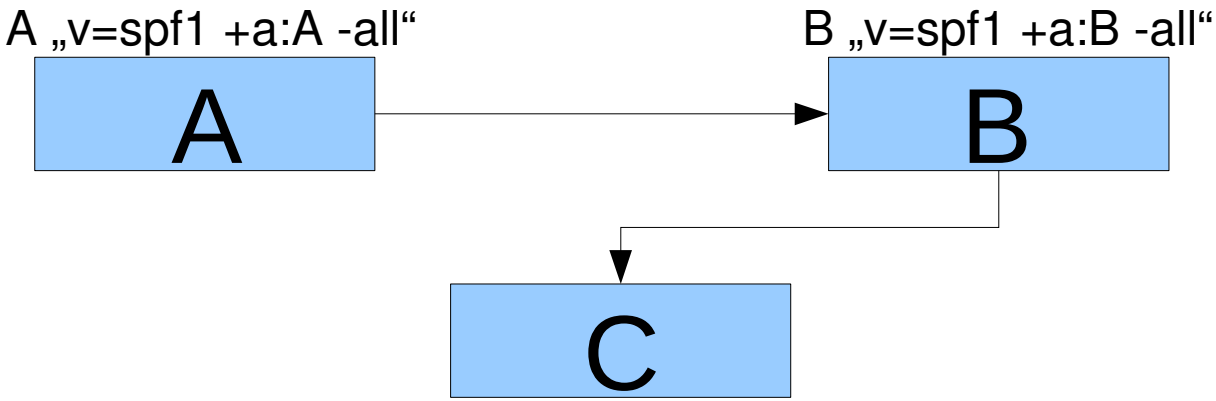
SPF: Sender Policy Framework
Ein Beispiel

- heinlein-support.de. IN TXT
  " v=spf1
  ip4:213.203.238.0/25 ip4:195.10.208.0/24
  +mx +include:jpberlin.de
  ?all "
  - v=spf1 – SPF-Record Version 1
  - ip4:x.x.x.x/xx – Netzbereich Ipv4 (analog: ip6:)
  - mx – erlaubt MX-Inbound-Server der Domain
  - include:domain.tld – SPF-Record einer anderen Domain
  - ?all – Über alle *nicht* genannten Server wird *keine* Aussage getroffen
    - (Alternativ: „-all“ -- alle anderen Server dürfen nicht)

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

SPF: Probleme mit -all

- 1: Weiterleitungen

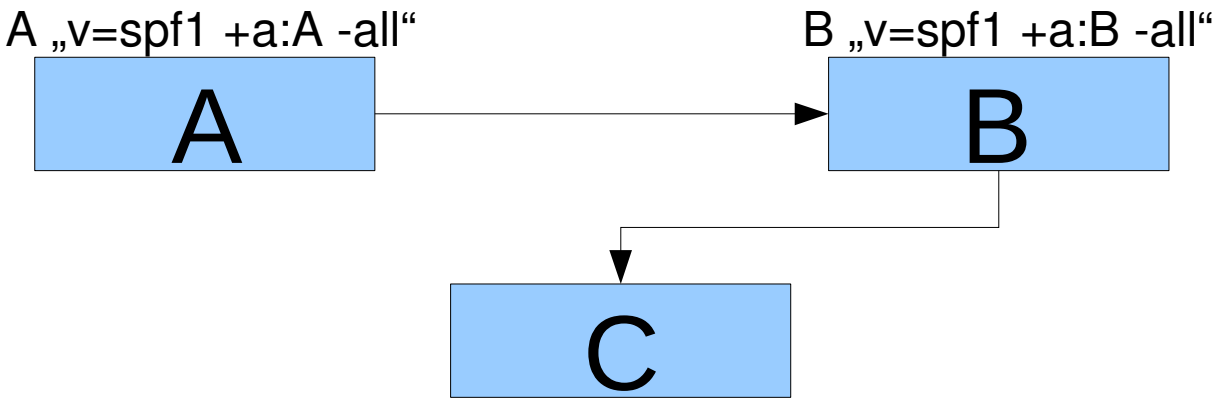


- Empfang der A-Mails von Server B laut SPF nicht erlaubt.
  - Nutzer haben aber millionenfach Weiterleitungen!
  - SPF: C müsste B whitelisten, damit Kommunikation so funktioniert.
  - A erhält Bounce von C obwohl B angeschrieben wurde

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

SPF: Probleme mit -all

- 2: Mailinglisten

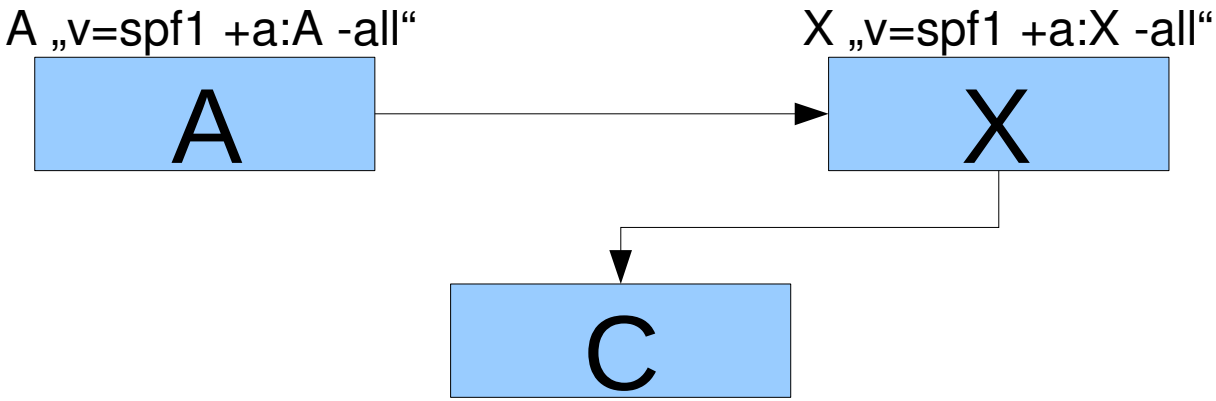


- A schreibt an Mailingliste B
- C Empfängt legal A-Mails über Server B
  - Läßt sich noch dadurch lösen, daß Absender = Mailingliste
  - Wünschenswert ?!?

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

SPF: Probleme mit -all

- 3: Communities, Webforen, Grußkarten, Ticker



- Empfang der A-Mails von Server X laut SPF nicht erlaubt.
  - A trägt sich in Newsletter auf X ein, X setzt Envelope-From direkt auf A
    - Envelope-From müsste aber postmaster@forum lauten
    - Wer macht das schon richtig?

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

SRS: Sender Rewriting Scheme
Die Lösung der Weiterleitungsprobleme?

- SRS: Weiterleitungsmechanismus
  - Aus user@A wird bei B user@B
  - Es gelten die SPF-Records von B
- Problem: Spammer B kann weiterhin Adressen fälschen!
  - Crypto-Hash HHH und Timestamp sollen schützen:
    SRS0 = HHH = TT = A = user@B
- Problem: Bouncebehandlung erfordert Absenderkodierung
  - SRS0 = HHH = TT = A = user#A-user@B

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

SRS: Sender Rewriting Scheme
Die Lösung der Weiterleitungsprobleme?

- Mailserver müßten das erstmal implementieren
- Geht nur, wenn alle mitmachen
  - Implementiert ein Weiterleitungsserver kein SRS, so bricht er die Kette
    - Bounces landen im Nirvana
- Was passiert bei mehrfachen Weiterleitungen?

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

SPF:
Ist das jetzt Spamschutz?

- SPF authentifiziert erstmal nur Absender
- Kritikpunkte
  - Spammer nutzen einfach „Einwegdomains“ mit gültigem SPF
  - „Zwangsrelays“ schaffen Vorteile für zentrale Überwachungsstellen
  - Benutzer müssten mehrere Postausgangsserver verwalten
  - Konkurrierende DNS-Records (TXT und SPF) müssten gepflegt/ausgewertet werden
  - DNS ist über UDP nicht gesichert, daher sind unvollständige Übertragungen wahrscheinlich
  - Phishing nicht berücksichtigt (Header ungeprüft)

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

SPF: Was macht SpamAssassin daraus?

- SPF wird gebrochen:
  - Könnte gefälschte Adresse sein
  - Könnte aber auch Weiterleitung/Liste/Forum sein
  - Kein hartes Ablehnungskriterium – kann aber einfließen!
  - SpamAssassin:
    - score SPF_FAIL 2.600 0.992 1.669 0.699
    - score SPF_SOFTFAIL 2.301 0.654 0.698 0.596

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

SPF: Was macht SpamAssassin daraus?

- SPF wird eingehalten:
  - Soll Mail privilegiert / gewhitelisted werden?
  - Spammer könnte „Einwegdomain“ mit eigenem SPF nutzen!
  - Kein Whitelisting-Kriterium!
  - SpamAssassin:
    - score SPF_PASS -0.001
    - score SPF_NEUTRAL 2.199 1.210 0.756 0.686

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

SPF:
Die Quintessenz

- Die Definition eines eigenen SPF-Records ist einfach und schadet nichts...
  - Schema laut Vortrag, Hilfe auf http://www.openspf.org
  - Aber immer „+all“ / „?all“ angeben!
- ...bringt aber auch nicht wirklich etwas
  - Aber einige nutzen es fälschlicherweise als „starkes“ Kritierium, das kann man sich ja zu nutze machen, wenn es funktioniert.

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

DKIM:
Domain Key Identified Mail

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

DKIM: Die Idee

- Das Problem:
  Absender nicht authentifiziert, Versandserver beliebig
- DKIM-Lösung:
  Daten im Mailheader müssen authentifiziert werden
  - Nutzer sieht und antwortet an From: aus Mailheader!
- Kryptographische Signierung relevanter Headereinträgen und dem Body der E-Mail
  - positiver Nebeneffekt: Fälschungssicherheit!
  - negativer Nebeneffekt: Anhänge machen bei Komplettsignierung Probleme (Mailinglistenfooter)

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

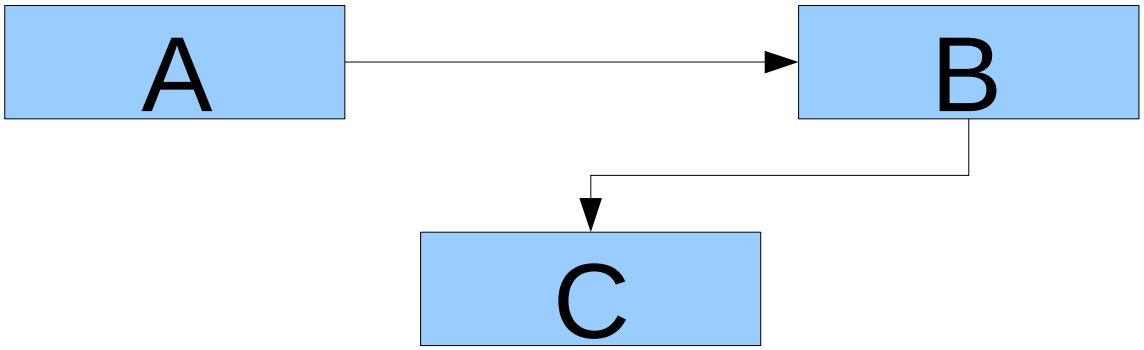
DKIM:
Die technische Umsetzung

- Mailserver einer Domain haben Schlüssel, signieren Mails
- Public-Key der Domain über DNS-TXT abfragbar
- Andere Server können Key fetchen und Mails prüfen

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

DKIM:
Wo ist der Unterschied zu SPF?

- DKIM und SPF im Vergleich:
  Mails können beliebig weitergeleitet werden!



- Server C findet Absender und Signatur von A!
  - DKIM stellt sicher, dass die Mail einst über A versandt wurde, auch wenn sie von B kommt!

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

DKIM: Wie nutzen?

- Bislang DKIM-Proxy:
  - http://dkmiproxy.sourceforge.net
- Besser: Amavis
  - Ab Version 2.6.0 (Juni 2008) native DKIM-Unterstützung auch ohne SpamAssassin.
  - Kann eingehende E-Mails prüfen – aber auch ausgehende signieren!
  - Wohl beste Lösung, wenn Amavis bereits im Einsatz ist.

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

DKIM: Mini-Howto für Amavisd-new

- Schlüssel erzeugen:
  - amavisd genrsa /var/lib/dkim/key.pem (Debian: amavisd-new genrsa ...)
- amavisd.conf anpassen:
  - $enable_dkim_verification = 1; # enable DKIM signatures verification
    $enable_dkim_signing = 1; # load DKIM signing code,
    dkim_key('example.com', 'abc', '/var/lib/dkim/key.pem');
- Key im DNS veröffentlichen
  - amavisd showkeys
    abc._domainkey.example.com. 3600 TXT ("v=DKIM1; p="
    "MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQDQcNuAysGQ4YxBhgPl6u"
    "JutgxhazJDOEw0zeNNbor9nPhDllMwT9WHPBCxQEpE4NvwDFmhaBh0/jdjYEl/kZ"
    "11u5bsMWo/8cf4RYgrEbklc0f9HJ+pyx4eNq9BTgWn8mDFc2Y36cmz5K2tBrpPT0"
    "ElR7qsLo5bIKjBAFkQIDAQAB")
- Testen:
  - amavisd testkeys
    TESTING: abc._domainkey.example.com => pass
- Mails senden und testen

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

DKIM: Die Quintessenz

- Kaum / keine Probleme bei Mailinglisten und Communities
  - Keine Adress-Umschreibungen nötig
  - Keine Anpassungen der MTAs, einfache DKIM-Filter etc. reichen
- Doch gleicher Vor-/Nachteil wie SPF:
  - Positives Whitelisting?

    Spammer können eigene Domains nutzen
  - Negatives Blacklisting?

    Würde sehr viele normale Mails treffen, es werden nie alle mitmachen.
- Erhöhter Rechenaufwand für die Signierung/Validierung

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Greylisting

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Greylisting: Mythen und Falschinformationen

- Das Prinzip:
  E-Mails unbekannter Absender werden zunächst mit temporären Fehler (4xx) abgewiesen
  - Temporäre Fehler sind „normal“: too many connections, dns error, not enough space left on device ...
- Später wird gleiche Mail vom gleichen Client akzeptiert

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Greylisting: Warum der einmalige 4xx

- Client soll zeigen, daß er ein queuender Mailserver ist
  - Botnetze queuen meist nicht => fire & forget im Massenversand
- Greylisting hilft gegen Botnetz-Mails: Spam und Viren

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Mythos 1:
Aber dann werden ja alle Mails verzögert.

- Gute Greylisting-Implementationen lernen automatisch alle die Subnetze aus denen wiederholt Triple bestätigt wurden!
  - Kein unnötiger „Test“ wenn bekannt ist, daß der Client ein Mailserver ist
- Nach kurzer Trainingsphase von wenigen Tagen:
  98% aller erwünschten e-Mails erhalten keinerlei Verzögerung!
  - Mailserver (relevanter Provider und Geschäftspartner) schnell gelernt
  - Nur Mails unbekannter neuer Absender werden verzögert => i.d.R. egal
  - Zeitkritische Empfangspostfächer selektiv vom Greylisting befreien (support@/helpdesk@, bestellung@, hotline@ etc. etc.)

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Mythos 2:
Dann müßten Spammer doch nur queuen

- Ja, richtig, sie könnten natürlich queuen. Aber:
  - Erneute Zustellversuche senken den Durchsatz eines Botnetzes
  - Warum Zeit in erneute Zustellversuche vergeuden, wenn anderswo Mails sofort zugestellt werden können?
  - Spammer haben genügend weitere Mailadressen.
  - Heutige Spamwellen oft ganz massiv unter 2-3 Stunden Gesamtzeit!
- Die Zeit arbeitet für uns:
  - Besitzer des infizierten PCs bemerkt Störung (PC nicht mehr nutzbar)
  - Zwangstrennung am Home-DSL (1h Verzögerung = 1/24 Chance auf neue IP!)
  - IP des PCs landet schnell auf Blacklisten

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Mythos 3: Greylisting ist aufwändig

- Genau das Gegenteil ist richtig.
  - Greylisting ist der „billigste“ einfachste, unkomplizierteste Spam-Schutz, den es derzeit gibt.
  - Es werden zwei Mailadressen, eine IP und ein Timestamp in einer DB gespeichert!
- Faktor 1000 mehr Last würden Mails angenommen und nur durch Content-Filterung geprüft werden!

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Mythos 4: Manche Provider senden nicht erneut

- Temporäre Fehler 4xx sind fest im RFC 2821 (SMTP) definiert
  - 4xx-Codes sind auch ohne Greylisting Alltag
- Gerade große ISPs (web.de, gmail.com, yahoo.com) geben massenweise temporäre Fehler aus
  - Wer damit nicht umgehen kann, hat auch so Probleme.

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Mythos 5: Zentrale/synchrone Greylisting-DBs nötig

- Häufige (aber falsche) Behauptung
  - Bei 4xx auf Mailrelay wandert Client **sofort** zu Mailrelay 2 (...3...4).
  - Alle Mailrelays lernen zeitgleich „unbestätigtes Triple“
  - Erneuter Zustellversuch nach wenigen Minuten bestätigt auf einem der Relays das Triple => Mail geht also ganz normal durch
  - Nächste E-Mail bestätigt auf anderem Server offenes Triple
- Aber: Sehr viele Mailrelays (>4) und sehr seltene E-Mails (alle paar Tage)
  - ggf. langsameres „Lernen“ weil offene Triple expiren => Nicht praktisch relevant.
  - Also: Jedem Mailrelay eigene robuste (Berkley)-DB auf Dateiebene

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Die Welt ohne Greylisting

- Ohne Greylisting:
  - Mailstau bei Spamwellen
    - Teuer Content-Filterung => hoher Serveraufwand
  - SPAM nimmt weiter zu
    - Weitere Skalierung wäre notwendig
- Mit Greylisting:
  - Schnelle Checks schaffen Zeit die Spreu vom Weizen zu trennen
  - Bekannte Kommunikation wird umgehend bedient
- Greylisting verschafft uns Zeit ...
  - Virenwellen werden immer kürzer
  - Problem: Zeit bis zum Signaturupdate des AV
  - Greylisting erhöht die Chance auf ein Patternupdate

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Abschluss

- SPF und DKIM sind der Versuch zur Absender-Verifizierung ...
  - ... jedoch kein direkter Spamschutz
  - Beides schadet nicht:
    - SPF: Sehr einfach. Nur kleine Anpassung im DNS.
    - DKIM: Software nötig / mehr Aufwand.
- Eingehend SPF und DKIM aber nur „soft“ prüfen – wie SpamAssassin.
- Ergo: gute Ergänzung der Kombination verschiedener Techniken:
    - Greylisting
    - RBL
    - SA
    - persönliche. Checks

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

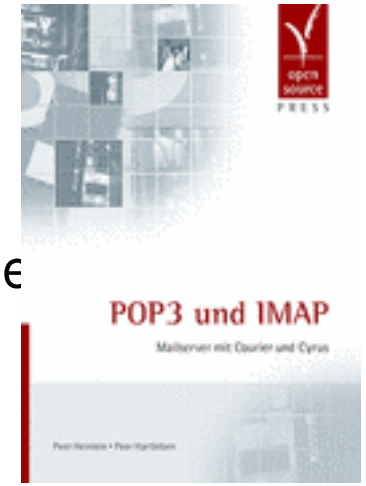
Soweit, sogut.

Fragen?

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Wenn es um echtes Papier geht:

- „POP3 und IMAP“
  Mailserver mit Courier und Cyrus
  - Erläutert auch Detailwissen über IMAP und Mailstorage
  - Courier sehr einfach auf Dovecot übertragbar, Prinzipien und nötiges Wissen sind identisch



- Das Postfix-Buch
  Sichere Mailserver mit Postfix
  - Der Klassiker mit rund 750 Seiten
  - Deckt auch Anbindung der IMAP-Server ab
  - Ab ca. Juni 2008 in stark überarbeiteter 3. Auflage

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Heinlein Support hilft auch bei allen Fragen rund um E-Mails:

- AKADEMIE
  - Von Profis für Profis: Wir vermitteln die oberen 10% Wissen. Geballtes Wissen und umfangreiche Praxiserfahrung aus erster Hand.
- SUPPORT
  - Wir sind das Backup für Ihre Linux-Administration: LPIC-2-Profis lösen im Heinlein CompetenceCall Notfälle, auf Wunsch auch in SLAs mit 24/7-Verfügbarkeiten.
- HOSTING
  - Wenn Hosting kein Massengeschäft sein darf: Individuelles Business-Hosting mit perfekter Maintenance durch unsere Linux-Profis. Sicherheit und Verfügbarkeit werden bei uns groß geschrieben.