HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

amavisd-new: Schöne Geheimnisse und komische Ideen.

- Technische Tipps zum Einsatz
  - Performancetuning, Datenbanken, ConfigGeheimnisse
- Gedanken zur Spamfilter-Strategie
- Unkonventionelle Meinungen zum Thema false positives...

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Performancetuning: Was ist zu tun?

- Was verursacht welche Belastungen?
  - Archive auspacken: CPU & I/O
  - RegExp-Matching/Spamfilter: CPU
  - Virenkiller starten/prüfen lassen: CPU & I/O
- Paralleles Ausführen lastet Resourcen besser aus
- Irgendwann wird Parallelität kontraproduktiv!
  - 10 Zips parallel auspacken bringt auch nichts mehr
  - 10 parallele Schreibzugriffe auf die Platte sind eher langsamer, als wenn sequentieller geschrieben wird

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Performancetuning: Faustformel

- 1 CPU, normale Harddisk: 7 Prozesse
- 1 CPU, RAM-Disk: 15 Prozesse

- 2 CPU, normale Harddisk: 7-10 Prozesse
- 2 CPU, RAM-Disk: 25 Prozesse

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Performancetuning: Faustformel

- Wichtig: Auslastung mit „top“ unter Volllast (!) beobachten!
  - Welche Belastungen haben die CPUs?
  - Wieviel Wait-I/O gibt es?
  - Welche Auslastung hat der RAM? Vorsicht: Freier RAM wird Cache!

```
top - 13:00:59 up 47 days, 17:16,  1 user,  load average: 0.17, 0.13, 0.09
Tasks: 179 total,   2 running, 177 sleeping,   0 stopped,   0 zombie
Cpu0  :  0.3%us,  0.0%sy,  0.0%ni, 99.0%id,  0.7%wa,  0.0%hi,  0.0%si,  0.0%st
Cpu1  :  0.0%us,  0.0%sy,  0.0%ni, 99.3%id,  0.7%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   2066144k total,  1302020k used,   764124k free,   173280k buffers
Swap:  2104472k total,     2556k used,  2101916k free,   211428k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
11763 vscan     15   0 55128  46m 2912 S    4  2.3   0:00.39 amavisd
```

  - Last läßt sich über smtp-source von Postfix erzeugen
  - Leicht verfälschte Ergebnisse, wenn Spamfilter-Cache dafür sorgt, daß Mails nicht immer voll gescannt werden.

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Eine RAM-Disk für amavisd-new

- Auch wenn sich viel theoretisch bereits im Plattencache abspielt: Die RAM-Disk bringt den Performance-Durchbruch
- Theoretische nötige Größe:
  Mailanzahl x Mailgröße x Auspackfaktor-X = Gesamtvolumen
  - Aber: 15 parallele 30 Mbyte-Mails sind unwahrscheinlich
- Selten zufällig volle Ram-Disk? Kein Problem: Gibt 4xx-Fehler!
  - Mail wird temporär abgewiesen, kommt später wieder.
- Meine Empfehlung bei 30 Mbyte Mailgröße: Ca. 180 Mbyte
  - Natürlich Abhängig vom Gesamt-Ram des Servers. Hier: 2 Gbyte.

```
# /etc/fstab:
none    /var/spool/amavis/tmp    tmpfs    defaults,size=180m,mode=750,uid=65,gid=0,noatime 0 0
```

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Wir bauen uns einen Filter-Cluster

- MTAs können „MX-Routing“, d.h. Failover-/Balancing anhand DNS-MX-Records
  - „Meta-Hostname“ filter.firma.local hat MX-Einträge auf Filter-Maschinen
- Postfix sieht den content_filter wie eine transport_map:
  - A-Record: content_filter=smtp:[filter.firma.local]:10025
  - MX-Record: content_filter=smtp:filter.firma.local:10025
- amavis weiß über $forward_method, wohin die Mail zurückgeht:
  - Genau hierhin: $forward_method = smtp:[hostname.domain.local]:10025 ;
  - An die IP, wo die Mail herkam (!): $forward_method = smtp:*:10025 ;
  - An die IP, wo die Mail herkam, jedoch an den Amavis-Emfangsport + 1: $forward_method = smtp:*:* ;

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Wir bauen uns einen Filter-Cluster: Die Probleme? Die Vorteile? Der Sinn?

- 2 x Postfix-MTA, 2 x amavisd-new-Filter: Diese Lösung liefert bereits fail-over und schließt single-point-of-failure aus
- Teure Loadbalancer sind überflüssig
- Lastverteilung läßt sich hier jedoch schwer steuern

- Problem:
  smtpd_proxy_filter kann kein MX-Routing, sondern verbindet fix zu einem Host!
  content_filter klappt jedoch problemlos.

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Spinnen wir das ganze mal weiter...

- Eigentlich könnte dann auch Postfix bereits auch auf einem der amavis-Maschinen laufen. Postfix verbraucht nicht wirklich viel Resourcen.
  - Im 2 x 2 – Setup sind die Postfix-Maschinen in der CPU nicht ausgelastet!
  - Derzeit brauchen MTAs aber viel RAM für sinnlose Spam-Einlieferungsversuche
- Aber vielleicht ist weniger auch mehr?
  - Eigentlich wären drei Postfix-MTAs mit jeweils lokalem amavisd-new viel schöner und die vierte Maschine wird frei?!
  - Möglich: 2 x Postfix-MTA, verteilen auf 4/6 x amavisd-new-Filter. Wer wirklich viel Feuer braucht...
  - Aber ehrlich: Vielleicht dann doch wieder lieber 2 Amavis-Maschinen mit jeweils feuriger Dual-/Quadcore-CPU und ordentlich RAM? Vier kleine oder zwei starke Server?

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Mehr Flexibilität in Postfix

- In-/Outbound trennen!
  - Unterschiedliche IPs leiten an unterschiedliche amavis-Ports weiter:

```
# Normaler SMTP -- nur authentifizierte Nutzer + normaler inbound
213.203.238.10:smtp     inet  n       -       y       -       20    smtpd
        -o smtpd_proxy_filter=vscanner-ha.jpberlin.de:10024

# mail.jpberlin.de (Unsere User)
213.203.238.11:smtp     inet  n       -       y       -       10    smtpd
        -o smtpd_recipient_restrictions=permit_sasl_authenticated,reject
        -o content_filter=smtp-amavis-n:vscanner-ha.jpberlin.de:10030

localhost:smtp          inet  n       -       y       -       5     smtpd

# smtp-bulk.jpberlin.de -- authentifizierte Nutzer, kein SpamAssasin, kein Headercheck:
# $bulk ist in main.cf definiert!
213.203.238.3:smtp     inet  n       -       y       -       15     smtpd
        -o content_filter=
        -o smtpd_proxy_filter=
        -o receive_override_options=no_header_body_checks
        -o smtpd_recipient_restrictions=$bulk
```

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Mehr Spaß mit Policy-Banks

- amavisd-new hört auf mehrere Ports:

```
$inet_socket_port =  [10024, 10030];
```

- Port 10030 kriegt einen Policy-Namen...

```
$interface_policy{'10030'} = 'AUTH';
```

- ...und seine eigene Config:

```
$policy_bank{'AUTH'} = {
# be slightly more permissive on spam levels for mail from our users:
  spam_kill_level_maps => 7.5,
#  spam_dsn_cutoff_level_maps => 15,
  bypass_banned_checks_maps => 1,  # allow sending any file type or
    name
# Don't use RBL-Checks for authenticated users
  $sa_local_tests_only = 1,        # RBL sollte damit ignoriert werden
};
```

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Userindividuelle Einstellungen: amavisd.conf

- /etc/amavisd.conf ist quasi Perl-Code. „Nichts ist unmöglich...“.
- I.d.R. zweidimensionale Arrays:

```
@virus_lovers_maps = (
 { 'postmaster@example.com'=> 1, # in single quotes the '@' need not be quoted
   'abuse@example.com'=> 1,
   'some.user@'       => 1,  # this recipient, regardless of domain
   'boss@example.com' => 0,  # never, even if domain matches
})
```

- Aber auch dreidimensionale Arrays möglich.
  Erste Lookup-Key: Recipient. Dann dessen Sonder-Einstellung:

```
@score_sender_maps = ({  # a by-recipient hash lookup table
# # per-recipient personal tables  (NOTE: positive: black, negative: white)
'user1@example.com'  => [{'bla-mobile.press@example.com' => 10.0}],
'user3@example.com'  => [{'.ebay.com'                    => -3.0}]
})
```

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Userindividuelle Einstellungen: SQL

- Amavis unterstützt ein eigenes SQL-Schema Out-of-the-Box:

```
@lookup_sql_dsn =
   ( ['DBI:mysql:database=mail;host=127.0.0.1;port=3306', 'user1', 'passwd1'],
#     ['DBI:mysql:database=mail;host=host2', 'username2', 'password2'],
#     ["DBI:SQLite:dbname=$MYHOME/sql/mail_prefs.sqlite", '', '']
);
```

- In der DB sind individuelle White-/Blacklists möglich, aber auch die die Einteilung in Gruppen: Virus-Lovers, User die weich/normal/hart gefiltert werden wollen etc.
- Ein eigenes Webfrontend ist schnell gebaut => User können selbst einstellen, was sie wollen.

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Userindividuelle Einstellungen: LDAP

- Im Prinzip gleiche Möglichkeiten wie SQL-Schema, nur eben anders :-)

```
#dn: cn=schema
#changetype: modify
#add: attributetypes
attributetype ( 1.3.6.1.4.1.15312.2.2.1.4
  NAME 'amavisBypassSpamChecks'
  DESC 'Bypass Spam Check'
  EQUALITY booleanMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.7
  SINGLE-VALUE )

#dn: cn=schema
#changetype: modify
#add: attributetypes
attributetype ( 1.3.6.1.4.1.15312.2.2.1.5
  NAME 'amavisSpamTagLevel'
  DESC 'Spam Tag Level'
  EQUALITY caseIgnoreIA5Match
  SUBSTR caseIgnoreIA5SubstringsMatch
  SYNTAX 1.3.6.1.4.1.1466.115.121.1.26{256}
  SINGLE-VALUE )
```

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Quarantäne in MySQL speichern

- Quarantäne-Mails können auch in SQL abgelegt werden
- So zentrale Speicherung für alle Nodes eines Clusters
- Einrichtung ist einfach, wenn eh schon userindividuelle Einstellungen eingerichtet sind:
  - @storage_sql_dsn = @lookup_sql_dsn; # none, same, or separate database
- Sonst eben wieder normale Zugriffsdefinition:
  - @lookup_sql_dsn = (['DBI:mysql:database=mail;host=127.0.0.1;port=3306', 'user', 'passwd']);

- Für die verschiedenen Bereiche muß die Quarantäne-Methode noch auf 'sql:' gesetzt werden.
  Alles in einem Rutsch zu setzen, ist möglich:
  - $virus_quarantine_method = $spam_quarantine_method = $banned_files_quarantine_method = $bad_header_quarantine_method = 'sql:';
- Wenn sql: angegeben wurde, muß auch @storage_sql_dsn definiert sein.
- README.SQL gibt Tipps, wie hier aufgeräumt wird:
  - DELETE FROM msgs WHERE UNIX_TIMESTAMP()-time_num > 7*24*60*60;
  - DELETE FROM msgs WHERE UNIX_TIMESTAMP()-time_num > 60*60 AND content IS NULL;
  - [Und noch mehr – siehe README.sql!]

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Mails aus der Quarantäne zustellen

- amavisd-release liefert Mails aus der Quarantäne aus:
  - amavisd-release path/to/spam-123456789.gz <secret>
  - <secret> schützt vor unbefugten Zugriff, wird in SQL-Datenbank gespeichert
  - Problem: Spam/Viren im Datei-Quarantäne haben kein <secret>!
  - Dann muß die Absicherung über $auth_required_release='undef'; deaktiviert werden
- amavisd-release nutzt das Amavis Policy Delegation Protocol.
  - Muß in amavisd.conf aktiviert werden:

```
$inet_socket_port = [10024,9998];
$interface_policy{'9998'} = 'AM.PDP';
$policy_bank{'AM.PDP'} = {
    protocol => 'AM.PDP', # Amavis policy delegation protocol
    inet_acl => [qw( 127.0.0.1 )],
};
# $auth_required_release = 'undef' ;
```

  - Eine amavisd-Instanz mehr einrichten, als Postfix in Anspruch nimmt!

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Mails nicht in Quarantäne speichern

- Für die verschiedenen Bereiche muß die Quarantäne-Methode einfach auf 'undef' gesetzt werden.
- Alles in einem Rutsch zu setzen, ist möglich:
  - $virus_quarantine_method = $spam_quarantine_method = $banned_files_quarantine_method = $bad_header_quarantine_method = 'undef';

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Filter-Training: Sinn oder Unsinn?

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Sinn und Unsinn von Training

- Training kann sinnvoll sein, wenn es richtig gemacht wird.
  - Problem: SA ist nur so gut, wie sein Training ist
- Oft wird nur Spam trainiert.
  - Kein Wunder: Dann sehen alle Mails auch eher nach Spam aus
  - false positives sind selbst verschuldet und antrainiert
  - Teufelskreis: Weil ich false positives habe, muß ich weiter trainieren
- Schlechtes halbherziges Training, führt zu schlechteren Ergebnissen, als ganz ohne Training,
- Ein SpamAssassin out-of-the-box filtert bestens.
  - Warum sollte es auch nicht?
  - Training nur sinnvoll, wenn man „besondere“ Mails hat, die andere auf der Welt so nicht haben!

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Wenn's denn sein muß: Wie trainieren?

- Durch die Nutzer
  - Nutzer trainieren, was sie nervt. Auch Mailinglisten, die sie selbst bestellt haben.
  - Spammer schieben Nutzern Müll-Mails zum Filter-Poisening unter. Nutzer trainieren das dann auch schön brav an. „Danke“, sagte dazu der Spammer, denn nun steigen false positives.
  - Nutzer: Denn sie wissen nicht, was sie tun!
- Durch den Admin
  - Arbeitsbelastung: Alle Mails müssen manuell (!) sortiert werden
  - Woher die für alle Nutzer repräsentative ham-Vergleichsbasis nehmen?
- Durch SpamAssassin selbst: bayes_auto_learn
  - ham-Mails < 0 Score-Punkte: Ham-Training!
  - spam-Mails > 12 Score-Punkte: Spam-Training!
  - Vorsicht: Mittelfeld sicherheitshalber nicht nutzen!

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Statt Training: Einzelfallfilter!

- Vereinzelte wenige false negatives nicht durch aufwändiges Training reparieren!
- Stattdessen: Gezielte Filterung durch einzelne Body-/Header-Checks in Postfix & Co, bzw. durch exakt darauf passende neue SpamAssassin-Regel im Eigenbau!
- Derzeit problematisch: Nigeria-Spam, Bilder-Spam, PDF-Spam. Sonst nichts.

HEINLEIN
LINUX · AKADEMIE · SUPPORT · HOSTING

A propos Bilderspam...: FuzzyOCR!

- Das Spam-Assassin-Modul FuzzyOCR nutzt OCR-Programme zur Texterkennung
- Liste mit „bösen Wörtern“ ergibt Score-Wert für das Bild
- FuzzyOCR kommt recht gut mit schlechten Bildern und/oder schlechter Erkennung klar
- FuzzyOCR ist durchaus performant, Technik ist einsetzbar
  - Caching bereits geprüfter Bilder hilft!
- Wortliste restriktiv selbst erstellen, sonst false positives!
  - Übrigens: Bilderspam läßt sich derzeit fast vollständig über Greylisting einfach wegfiltern.
  - Tipp: Einfach mal den Standard-Text von postgrey ändern und neutral „try again later“ etc. eintragen

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Mail-Tagging: Sinn oder Unsinn?

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Warum sollte man Mails taggen wollen?

- ANGST vor false positives.
  - Ein normales SpamAssassin hat mit ca. 1:100.000 keine false positives, solange es nicht kaputttrainiert wurde!
  - Gerade die heute relevanten Mechanismen können nicht taggen, da die Mail nie empfangen wird: Greylisting, policyd-weight.
- Filtert man zu vorsichtig, löschen Nutzer Mails mit schneller Hand weg.
- Nutzer erzeugen mehr false positives, als ausgereifte Filter-Systeme!
- Kleine Umfrage; Welcher Nutzer schaut eigentlich in den Spamverdachts-Ordner?

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Sinn und Unsinn von Tagging

- Was passiert mit Spamverdachts-Ordner-Mails?
  - Ein Blick nur in den Subject reicht nicht aus. Das kann nicht zuverlässiger sein, als SpamAssassin & Co!
  - Ergo: Man müßte alle Mails lesen. Dann gleich alle ungetaggt in die INBOX.
  - De facto wird in diesen Ordner so gut wie nie richtig reingeschaut.
- Problem: Mails versacken im Ordner, kommen nicht an. Absender kriegt kein Bounce, geht von Zustellung aus.
  - false positives fallen erst (zu) spät auf, wenn der Ärger da ist.
- Besser: Mails rejecten, Absender ist sofort informiert.
  - false positives fallen auf, werden sofort geklärt.
  - Insgesamt: Weniger Ärger für alle.

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Tagging/Verdachtsordner führen zu Mailverlusten!

- Problem: Haftung bei „versackter“ Mail?
  - Auftrag wurde nicht geliefert, bzw. Auftrag wurde trotz Stornierung ausgeführt!
  - Wer zahlt Schaden?
  - Absender: Ich habe 250-Empfangsbestätigung! Empfänger hat Post hausintern verschlampt, er haftet.
- Spam-Ordner und Tagging sind nicht im wirklichen Interesse der Nutzer / der Firma!
  - Gerade deshalb erst in der Praxis Mailverlust!
  - Keine Verläßlichkeiten, ob Mails ankamen!
  - Chaos, Ärger, Schaden!

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Für die, die nicht genug kriegen können

- Das Postfix-Buch von mir bei Open Source Press
  http://www.postfixbuch.de
- Übrigens: Komplett überarbeitete 3. Auflage im späten Herbst.
- Hat dann alles, was man braucht und wissen muß:
  - Die ganzen neuen Features und Änderungen
  - Mehr LDAP
  - Richtig schön amavisd-new / SpamAssassin
  - Noch mehr Tipps und Best Practice mit vielen Empfehlungen aus der Praxis
- Die Postfixbuch-Mailingliste:
  http://listi.jpberlin.de/mailman/listinfo/postfixbuch-users

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Für die, die konkrete Hilfe brauchen

- Wir sind die Mailserver-Schmiede:
  - Troubleshooting im Notfall
  - Consulting & Installation, auch Cluster-Installationen mit > 250.000 Nutzern
  - HA Anti-Spam/Viren-Relays
  - Groupware-Lösungen und Migrationsstrategien
- 24/7 CompetenceCall mit LPIC-2-Admins: 030/40 505 - 110
- Postfix-Kurse an unserer Akademie oder inhouse bei Ihnen
- Mail-Outsourcing in unser Rechenzentrum

HEINLEIN
LINUX: AKADEMIE · SUPPORT · HOSTING

Was ich noch zu sagen hätte...

- Bei Fragen: Fragen!

- Und ansonsten: Danke für Eure Aufmerksamkeit und noch eine schöne weitere Konferenz!