Karlsruher Institut für Technologie

Exim - da geht doch was!

Exim und seine Möglichkeiten bezüglich Spam, Viren, Greylisting, Ratelimiting

ATIS - Abteilung Technische Infrastruktur, Fakultät für Informatik

Agenda

- Vorstellung
- Exim Grundlagen

- Exim und LDAP
  - SMTP-Auth mit LDAP
  - LDAP-Mailrouting

- Keep the bad guys out
  - Denial-of-Service-Abwehr
  - Ratelimiting
  - Greylisting

- Exim und (kein) Amavis
  - Viren
  - Spamassassin
  - Spam-Relay-Vermeidung

Agenda

- Vorstellung
- Exim Grundlagen
- Exim und LDAP
  - SMTP-Auth mit LDAP
  - LDAP-Mailrouting
- Keep the bad guys out
  - Denial-of-Service-Abwehr
  - Ratelimiting
  - Greylisting
- Exim und (kein) Amavis
  - Viren
  - Spamassassin
  - Spam-Relay-Vermeidung

Die ATIS

- KIT: ***K**arlsruher **I**nstitut für **T**echnologie*
- Fusion aus der Universität Karlsruhe und dem Forschungszentrum Karlsruhe
- Fakultät für Informatik leistet sich, im Gegensatz zu den restlichen Fakultäten, zusätzlich zum Uni-RZ einen eigenen IT-Dienstleister

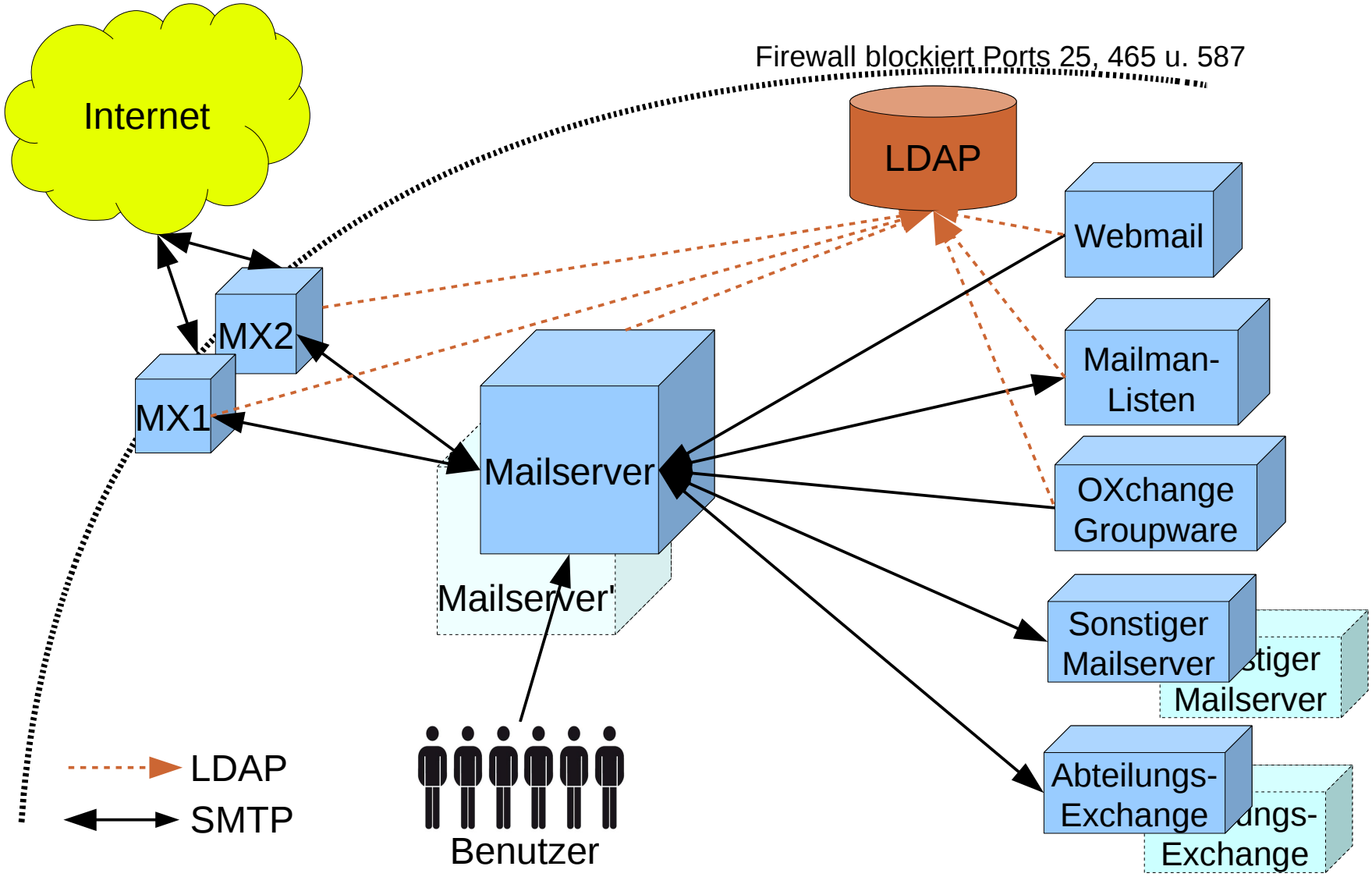
- ATIS: ***A**bteilung **T**echnische **I**nfra**s**truktur*
  - Betrieb der Netz-Infrastruktur bis zur Dose
  - Betrieb der Fakultäts-zentralen IT-Dienste
    - WLAN, DNS, DHCP, LDAP, VPN, Poolräume, E-Mail, SVN, Groupware
  - Keine Weisungsbefugnis gegenüber den Lehrstühlen!

- Vortragender:
  - Leiter des Bereich „IT-Dienste“ und stellv. Abteilungsleiter
  - Seit 11 Jahren Postmaster

E-Mail an der Fakultät für Informatik

- ~400 Mitarbeiter und ~2.500 Studierende
  - ~4.500 Postfächer und ~15.000 Mailadressen
  - $\Longrightarrow$ viele „tote“ Konten, viele Weiterleitungen
  - Nur etwa 1.000 Konten sind aktiv
- ~250.000 Mails (Ham und Spam) pro Woche = alle 2 Sek. eine E-Mail
- Sonderlösungen der Lehrstühle
  - Dezentrale Exchange-Server / Mailserver
  - *„Mein Spam gehört mir“*
  - Freiheit von Forschung und Lehre
- Viren werden abgewiesen
- Spam wird markiert und in der Regel in die Spambox gelegt
- Spam älter als 30 Tage wird gelöscht
- Nutzer erhält jeden Montag ein Inhaltsverzeichnis seiner Spambox
  - Inhaltsverzeichnis ist nach aufsteigenden Spampunkten sortiert

Darstellung der Mail-Infrastruktur



Agenda

- Vorstellung
- Exim Grundlagen
- Exim und LDAP
  - SMTP-Auth mit LDAP
  - LDAP-Mailrouting
- Keep the bad guys out
  - Denial-of-Service-Abwehr
  - Ratelimiting
  - Greylisting
- Exim und (kein) Amavis
  - Viren
  - Spamassassin
  - Spam-Relay-Vermeidung

Exim Grundlagen

- Eine einzige monolithische Konfigurationsdatei **exim.conf**
- Anatomie der exim.conf:
  - globale Variablen und Macros
    am Anfang der exim.conf
  - **`begin acl`**
    Die **A**ccess**C**ontrol**L**ists
  - **`begin authenticators`**
    Regelwerk für das SMTP-Auth
  - **`begin rewrite`**
    Umschreiben von Adressen
  - **`begin routers`**
    Die Router entscheiden über die Mailauslieferung. Was nicht geroutet werden kann, wird abgewiesen. ***Order does matter!***
  - **`begin transports`**
    Wie wird was ausgeliefert. Aufruf erfolgt aus den Routern
  - **`begin retry`**
    Regeln für Zustellversuche bei 400er Fehlern

Exim: Access Control Lists (ACLs)

- Die ACLs entscheiden darüber, was mit einer Mail geschieht
- Aufgerufen werden sie je nach Definition zu den unterschiedlichen Zeitpunkten im aktiven SMTP-Dialog
- In den einzelnen ACLs wird mit den Verben
  - **`accept`**
  - **`deny`**
  - **`defer`**
  - **`warn`**

  und weiteren entschieden, was mit der Mail geschehen soll.
- Jedes Verb kann mit unterschiedlichen Bedingungen näher spezifiziert werden
-
```
acl_smtp_connect = check_connect
[...]
begin acl
check_connect:
     deny message = Bad Host. Go Away!
            hosts = badhost.example.org
```

Exim: ACLs

- Einige ACL-Bezeichner:
  - **`acl_smtp_connect`**
    Wird nach dem TCP-Handshake aufgerufen
  - **`acl_smtp_helo`**
    wird nach den helo/ehlo gerufen
  - **`acl_smtp_mail`**
    wird nach dem „Mail from“ gerufen
  - **`acl_smtp_rcpt`**
    wird nach der Angabe des Empfängers gerufen
  - **`acl_smtp_data`**
    wird am Ende der Data-Phase gerufen, wenn der Sender den Punkt „**.**“ gesendet hat, bevor man das mit „2xx OK“ quittiert. Hier kann dann eine Inhaltsprüfung stattfinden.
  - ...
- Damit kann man sich an fast jede Stelle des SMTP-Dialogs einklinken und Entscheidungen über die (Nicht-)Annahme der Mail treffen

Agenda

- Vorstellung
- Exim Grundlagen
- Exim und LDAP
  - SMTP-Auth mit LDAP
  - LDAP-Mailrouting
- Keep the bad guys out
  - Denial-of-Service-Abwehr
  - Ratelimiting
  - Greylisting
- Exim und (kein) Amavis
  - Viren
  - Spamassassin
  - Spam-Relay-Vermeidung

Exim und LDAP

- Exim und LDAP arbeiten hervorragend zusammen
- Dies wird im folgenden an zwei Beispielen demonstriert
  - SMTP-Authentifizierung gegen LDAP
  - Mailrouting mit LDAP
    - Welche Nutzer kenne ich?
    - Welcher Mail-Alias wird auf welchen Account abgebildet?
    - Auf welchen IMAP-Server liegt in welchem Verzeichnis das jeweilige Postfach?

- Warum LDAP?
  - Sehr schnelle Leseoperationen (und sehr schlechtes Schreibverhalten)
    - Verifikation aller 15.000 Adressen in einer Minute = 250 Verifies/s
  - Replikation
  - Über Schemaerweiterungen den Bedürfnissen anpassbar
  - Authentifizierung (fast) aller Systeme
    - Exim, Courier, Apache, Radius, Samba, UNIX-Logins, …
  - Telefonbuch / Adressverzeichnis für Nutzer als „Abfallprodukt“

LDAP Grundlagen

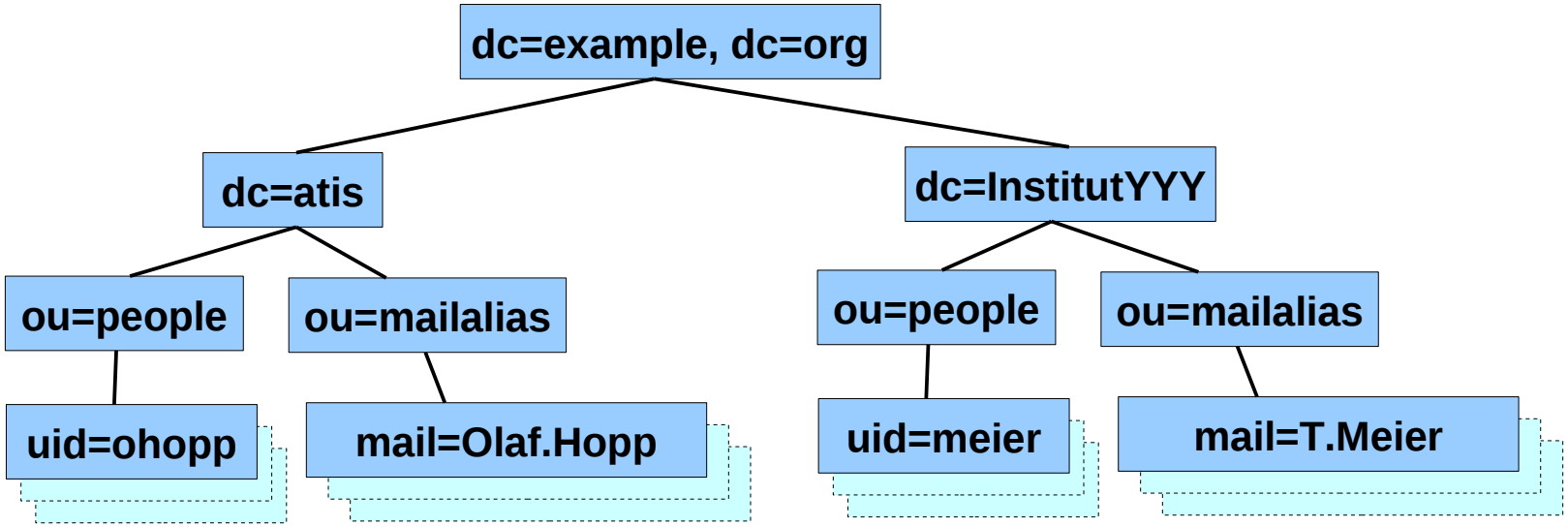
- Verzeichnisdienst mit baumartiger Datenstruktur
- Die Blätter des Datenbaums nennt man Attribute
- Eine Sammlung verschiedener zusammengehöriger Attribute nennt man Objektklasse
- Es existieren eine Menge vordefinierter Objektklassen z.B.

  - **`posixAccount`** (Pendant zu /etc/passwd bzw. /etc/shadow)
    - **`uid = ohopp`**
    - **`userPassword = geheim`**

  - **`qmailUser`**
    - **`mail = Olaf.Hopp`**
    - **`mailForwardingAdress = ohopp@atis.example.org`**
    - **`mailHost = imap.atis.example.org`**
    - **`mailMessageStore = /home/atis/ohopp`**

LDAP der ATIS

- Strukturierter LDAP:
  - Ein LDAP-Zweig pro Abteilung
  - Pro Abteilung zwei Zweige:
    - **`people`**
    - **`mailalias`**
  - Dort die einzelnen Objekte / Nutzer



Agenda

- Vorstellung
- Exim Grundlagen

- Exim und LDAP
  - SMTP-Auth mit LDAP
  - LDAP-Mailrouting

- Keep the bad guys out
  - Denial-of-Service-Abwehr
  - Ratelimiting
  - Greylisting

- Exim und (kein) Amavis
  - Viren
  - Spamassassin
  - Spam-Relay-Vermeidung

SMTP-Auth: Wer braucht es und wozu ?

- Nur die „Guten“ sollen senden dürfen
- Problematisch bei root-/cron-Jobs
- Interessant für Aussendienstmitarbeiter / Heimarbeiter
  - Beispiel: WLAN am KIT (nicht Informatik-Netz)
- Voraussetzung: der jew. Provider lässt Mails auf Port 25 raus
  - Hintertür: Port 587 (SMTP-Submission) geht oft, manchmal auch Port 465 (SMTP over SSL)
- Problem: Passwörter wandern u.U. im Klartext übers Netz
  - Deshalb:

```
tls_advertise_hosts = *
tls_certificate     = smtp.pem
tls_privatekey      = smtp.key
[...]
server_advertise_condition = ${if eq{$tls_cipher}{}{0}{1}}
```

SMTP-Auth: Wie sieht das aus ?

- C: ehlo localhost
- S: 250-ms1.example.org Hello localhost [127.0.0.1]
  ...
  250-STARTTLS
  ...
- C: STARTTLS
- S: 220 TLS go ahead ...
- C: ehlo localhost
- S: 250-ms1.example.org Hello localhost [127.0.0.1]
  ...
  250-AUTH PLAIN LOGIN
  ...
- C: AUTH PLAIN \0ohopp\0geheim\0
- S: 235 AUTHENTICATION SUCCEEDED
- C: Mail from: ohopp@atis.example.org
- ....

SMTP-Auth

- 2 gängige Modi:
  - PLAIN (Username in $auth1, Passwort in $auth2)
  - LOGIN (Username in $auth2, Passwort in $auth3)
- Weitere: CRAM_MD5, ...
- Received Zeile bei Verschlüsselung und Authentifizierung:

```
Received: from localhost ([127.0.0.1])
    by smtp.informatik.example.org with esmtpsa port 25
    for <ohopp@atis.example.org>
    id 1Q9tYZ-00008U-4V; Wed, 13 Apr 2011 08:28:10 +0200
```

SMTP-Auth: Umsetzung

```
begin authenticators
plain:
    driver plaintext
    public_name     = PLAIN
    server_condition=${if and {{eq{$auth2}{ohopp}}
                              {eq{$auth3}{geheim}}}
    server_set_id   = $auth2
login:
    driver plaintext
    public_name     = LOGIN
    server_prompts  = Username:: : Password::
    server_condition=${if and {{eq{$auth1}{ohopp}}
                              {eq{$auth2}{geheim}}}
    server_set_id   = $auth1
```

- Der authentifizierte Nutzer ist über die server_set_id-Zeile hinterher über die Variable $authenticated_id verfügbar und Log steht dann

2011-04-13 08:28:10 1Q9tYZ-00008U-4V <= ohopp@atis.example.org H=localhost [127.0.0.1] P=esmtpsa X=TLSv1:AES256-SHA:256 A=plain:ohopp S=286

SMTP-Auth gegen LDAP

- Wenn man gegen einen LDAP-Server authentifiziert:

```
server_condition = \
     ${if ldapauth { \
           user="uid=ohopp,ou=people,dc=atis,dc=example,dc=org" \
           pass="geheim"
           ldap://ldap.example.org }}
```

- Statt "**ohopp**" nimmt man **${quote_ldap_dn:$auth1}** und anstatt „**geheim**" schreibt man **${quote:auth2}**

- Aber: LDAP-Server mit Struktur
  - **uid=ohopp, ou=people, dc=atis, dc=example, dc=org**
  - **uid=meier, ou=people, dc=abteilungYYY, dc=example, dc=org**
  - **uid=mueller, ou=people, dc=abteilungZZZ, dc=example, dc=org**

- Man „weiß" noch nicht, wo man suchen muss!

SMTP-Auth: strukturierter LDAP

- Man muss erst nach den Benutzer suchen
- Die UIDs sind jedoch eindeutig bei uns
- Wir haben mehrere zueinander redundante LDAP-Server:

```
ldap_default_servers = ldap1.example.org:ldap2.example.org
LDAP_URL             = ldap:///dc=example,dc=org
```

```
server_condition = ${if ldapauth {\
  user= \
  ${lookup ldapdn{LDAP_URL??sub?(uid=${quote_ldap_dn:$auth1})}}\
  pass=${quote:$auth2}          \
                              ldap:/// }}
```

- Der „**lookup ldapdn**“ sucht überall („**sub**“) nach „**uid=$auth1**“ und findet den DN wie z.B.

  uid=ohopp,ou=people,dc=atis,dc=example,dc=org

  und damit wird dann das „bind“ gegen den LDAP-Server gemacht

Agenda

- Vorstellung
- Exim Grundlagen

- Exim und LDAP
  - SMTP-Auth mit LDAP
  - LDAP-Mailrouting

- Keep the bad guys out
  - Denial-of-Service-Abwehr
  - Ratelimiting
  - Greylisting

- Exim und (kein) Amavis
  - Viren
  - Spamassassin
  - Spam-Relay-Vermeidung

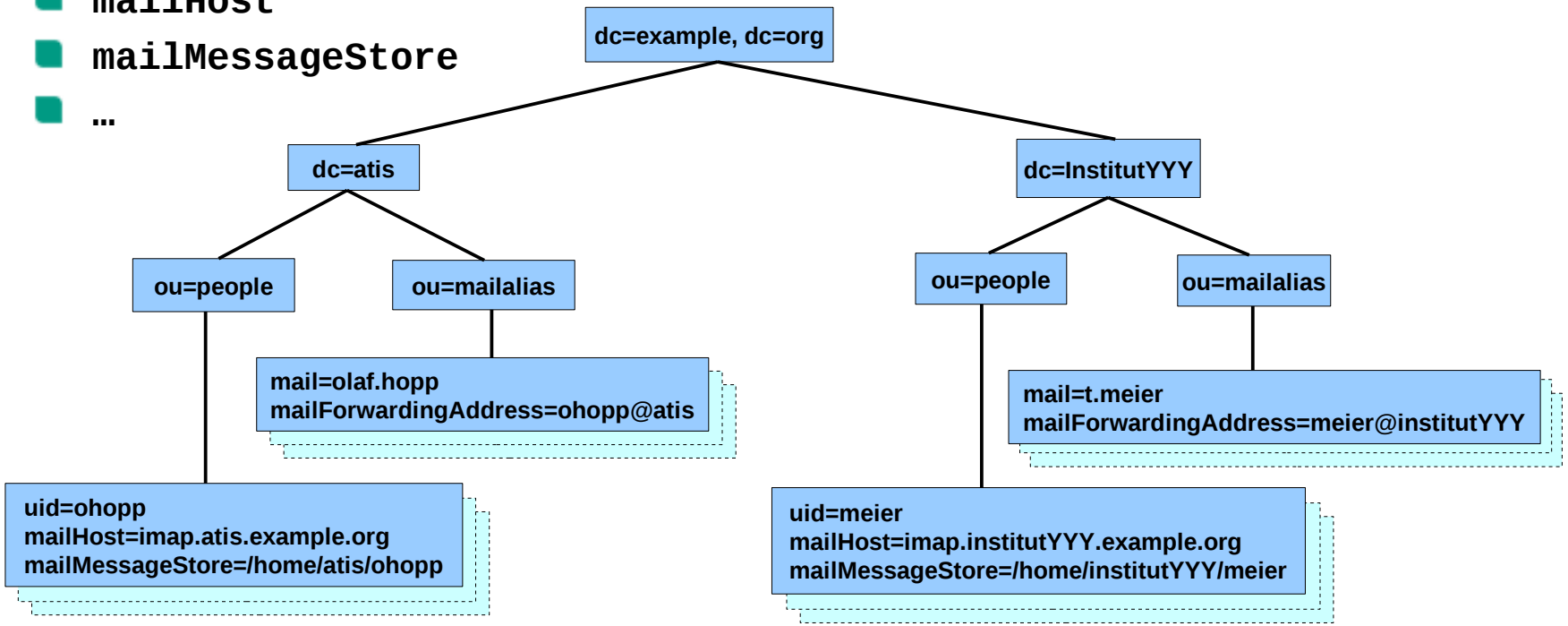
LDAP Mailrouting

- Mit Hilfe des LDAP-Servers sollen folgende Fragen beantwortet werden:
  - Kenne ich die Adresse?
  - Wenn das ein Mailalias ist, wo zeigt er hin?
  - Auf welchem Server liegt das Postfach?
  - Wie lautet der Pfad zum Postfach?

- Im Folgenden:
  - E-Mail an **Olaf.Hopp@atis.example.org**
  - Alias auf **ohopp@atis.example.org**
  - IMAP-Server **imap.atis.example.org**
  - Postfach **/home/atis/ohopp**

LDAP Mailrouting

- Es existiert eine Klasse **`qmailUser`**
- Diese besitzt viele Attribute, die für das Speichern von Mail-Informationen im LDAP geschaffen sind:
  - **`mail`**
  - **`mailForwardingAddress`**
  - **`mailHost`**
  - **`mailMessageStore`**
  - **…**



LDAP Mailrouting

- Mail für **Olaf.Hopp@atis.example.org:**
  - `${local_part} = Olaf.Hopp`
  - `${domain}     = atis.example.org`

- Mailalias im LDAP:

```
dn:mail=olaf.hopp,ou=mailalias,dc=atis,dc=example,dc=org
mail:                    olaf.hopp
mailForwardingAddress: ohopp@atis.example.org
```

- Der Router dazu:

```
ldap_aliases:
  driver = redirect
  data   = ${lookup ldap {\
      ldap:///dc=${extract{-3}{\.}{${domain}}},dc=example,dc=org\
           ?mailForwardingAddress?sub?(mail=$(local_part})}}
```

- Ergibt **ohopp@atis.example.org**

LDAP Mailrouting zum IMAP-Server

- Mail für **ohopp@atis.example.org :** gesucht wird der IMAP-Server
- Mailkonto im LDAP:

```
dn: uid=ohopp,ou=people,dc=atis,dc=example,dc=org
uid:                ohopp
mail:               ohopp@atis.example.org
mailMessageStore:   /home/atis/ohopp
mailHost:           imap.atis.example.org
```

- Der Router dazu:

```
ldap_mailhost:
  driver = redirect
  data   = ${local_part}@${lookup ldap {\
      ldap:///dc=${extract{-3}{\.}{${domain}}},dc=example,dc=org\
                  ?mailHost?sub?(mail=${local_part}@${domain})}}
```

- Ergibt ohopp@**imap**.atis.example.org
  - Auf den Mail-Exchangern ist **imap** nicht-lokal
    $\Longrightarrow$ Weiterleitung nach „innen“ zum IMAP-Server
  - Auf dem IMAP-Server dann **local_delivery** machen, dazu **mailMessageStore** auswerten

Agenda

- Vorstellung
- Exim Grundlagen

- Exim und LDAP
  - SMTP-Auth mit LDAP
  - LDAP-Mailrouting

- Keep the bad guys out
  - Denial-of-Service-Abwehr
  - Ratelimiting
  - Greylisting

- Exim und (kein) Amavis
  - Viren
  - Spamassassin
  - Spam-Relay-Vermeidung

Denial-of-Service-Abwehr

- Exim bietet diverse Schalter, um einem ***D**enial **o**f **S**ervice* zu begegnen
- Im wesentlichen sind das globale Variablen, die die Anzahl der SMTP-Verbindungen begrenzen
  - Anzahl aller SMTP-Verbindungen
    `smtp_accept_max = 500`
  - Max. Anzahl pro Host
    `smtp_accept_max_per_host = 25`
  - Die „guten“ Hosts
    `smtp_reserve_hosts = 127.0.0.1 : 192.168.0.0/16`
  - reservierte Zahl an SMTP-Verbindungen für die „Guten“
    `smtp_accept_reserve = 100`
  - ab dieser load dürfen nur noch die „Guten“
    `smtp_load_reserve = 20`
  - ab dieser Anzahl an SMTP-Verbindungen / Load wird gequeued
    `smtp_accept_queue = 100`
    `queue_only_load   = 20`

Distributed DOS: dynamisches Blacklisting

- Was passiert bei DDOS ?
  - Viele, viele Verbindungen gleichzeitig von vielen verschiedenen Hosts
  - **`smtp_accept_max_per_host`** blockierte einzelne Hosts
  - Dennoch erschöpfte sich **`smtp_accept_max`**
  - Die externen „Guten“ konnten nicht mehr senden
- Was tun ?
- Wenn **`smtp_max_per_host`** greift, dann findet man im Log

```
2011-04-19 07:39:04 Connection from [194.143.137.215] refused:
too many connections from that IP address
```

- Per cron-Job die IPs aus Log raussuchen und in einer Datei ablegen
  geht auch mehrmals pro Minute („logtail“ liefert gute Hilfe):

```
* * * * * blackips.sh; sleep 30; blackips.sh
```

- ACL acl_smtp_connect schreiben und dort die IPs blocken

```
acl_smtp_connect  = check_connect
check_connect:
deny hosts = \
     ${lookup{$sender_host_address}iplsearch{/opt/exim/blackips}}
```

Agenda

- Vorstellung
- Exim Grundlagen

- Exim und LDAP
  - SMTP-Auth mit LDAP
  - LDAP-Mailrouting

- Keep the bad guys out
  - Denial-of-Service-Abwehr
  - Ratelimiting
  - Greylisting

- Exim und (kein) Amavis
  - Viren
  - Spamassassin
  - Spam-Relay-Vermeidung

Ratelimiting

- Mit dem Ratelimiting vom Exim kann das „Volumen“ der Sender begrenzt werden
- Volumen kann sein:
  - Anzahl an Mails
  - Anzahl an SMTP-Connects
  - Anzahl der SMTP-Kommandos
  - gesendete Bytes
- Zustand muss sich „gemerkt“ werden
- EXIM verwendet dazu eine interne DB
- In einem ACL (wo sonst ?) entscheidet man, was gemacht wird
- Nicht zu verwechseln mit den (veralteten) **`smtp_ratelimit_XXX`** Optionen, die wirken nur pro einzelne SMTP-Verbindung!
- ```
  defer message     = RATELIMIT EXCEEDED – TRY LATER !
        log_message = ratelimit $sender_host_address ($sender_rate)
        ratelimit   = 10 / 1s
  ```
  ```

Ratelimiting

- Wie kommt man zu vernünftigen Startwerten?
- Erstmal nur mitloggen:

```
warn ratelimit = 0 / 1h
   log_message = $sender_host_address  : \
                                 $sender_rate / $sender_rate_period
```

- Dann die Logs analysieren und Schwellwerte wählen
- Man will zwei Werte begrenzen:
  - E-Mails pro Minute
  - E-Mails pro Stunde
  - Getrennt für „normale“ Hosts und Mailserver mit höherem Volumen
-

```
HIGH_PER_MINUTE  =  200
HIGH_PER_HOUR    = 2000
LOW_PER_MINUTE   =   20
LOW_PER_HOUR     =  200
RATELIMITMESSAGE = RATELIMIT EXCEEDED $sender_host_address: \
                   $sender_rate messages / $sender_rate_period
hostlist high_traffic_hosts = hostA : hostB : hostC
```

Ratelimiting

- ACLs:

```
defer   ratelimit = HIGH_PER_HOUR / 1h
      log_message = RATELIMITMESSAGE
          message = RATELIMITMESSAGE
            hosts = +high_traffic_hosts

defer   ratelimit = LOW_PER_HOUR / 1h
      log_message = RATELIMITMESSAGE
          message = RATELIMITMESSAGE
            hosts = !+high_traffic_hosts
```

-
```
defer   ratelimit = HIGH_PER_MINUTE / 1m
        [...]

defer   ratelimit = LOW_PER_MINUTE / 1m
        […]
```

- Damit dürfen bekannte Mailserver 2.000 E-Mails/h senden, aber in Bursts bis zu 200 E-Mails/s
- Alle anderen Hosts lediglich 200 E-Mails/h und kurzzeitig 20 E-Mails/m

Agenda

- Vorstellung
- Exim Grundlagen

- Exim und LDAP
  - SMTP-Auth mit LDAP
  - LDAP-Mailrouting

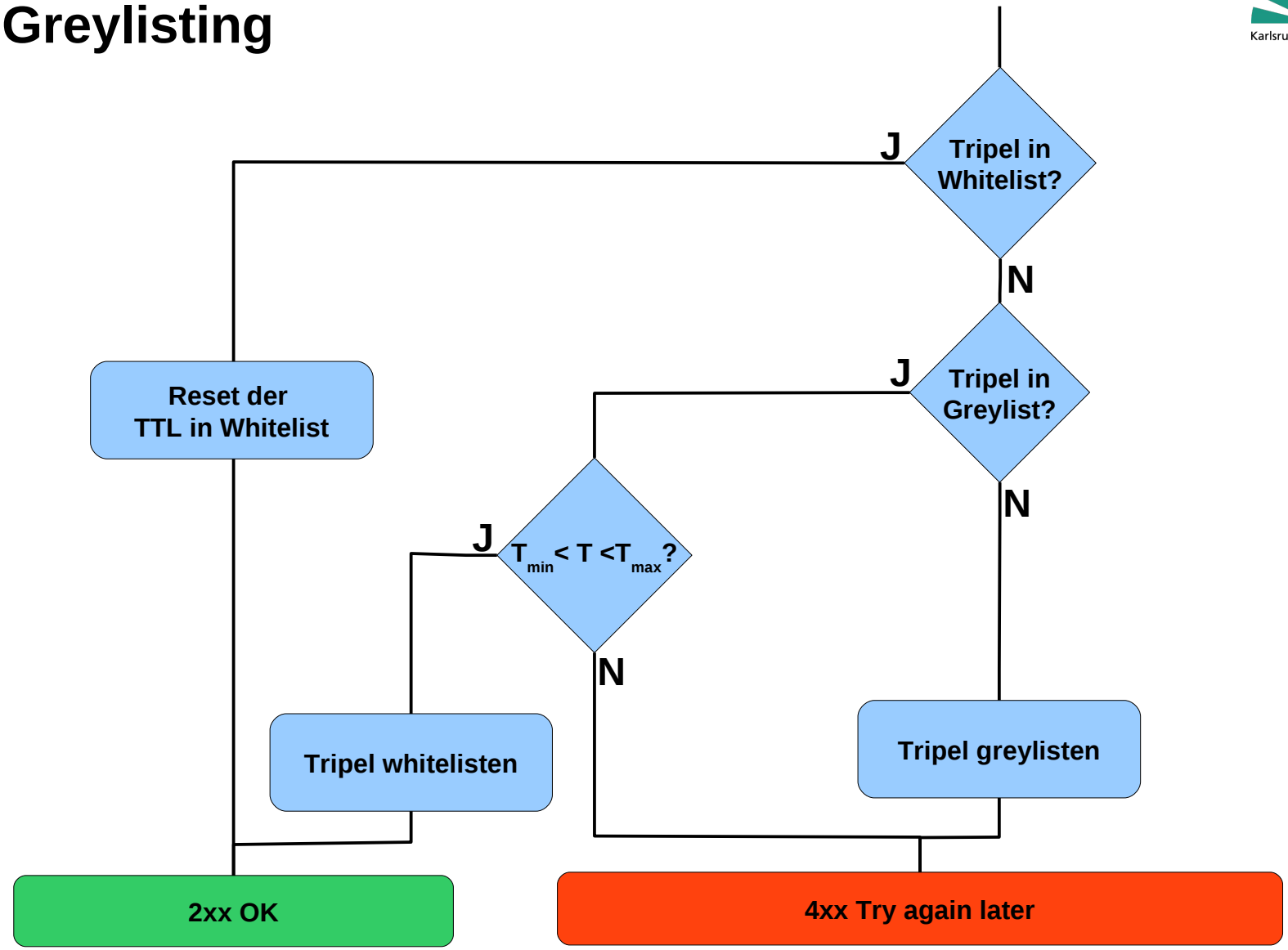
- Keep the bad guys out
  - Denial-of-Service-Abwehr
  - Ratelimiting
  - Greylisting

- Exim und (kein) Amavis
  - Viren
  - Spamassassin
  - Spam-Relay-Vermeidung

Greylisting

- Temporäre Abweisung von Mails von „Unbekannten“
- Spamroboter machen „One-shot-only“
- Nachteil: das „schnelle“ Medium E-Mail wird ausgehebelt
- Methode:
  - Wenn Mail ankommt, schaue nach dem Tripel:
    - Sender-IP
    - Envelope-Sender
    - Envelope-Empfänger
  - 3 Zeiten:
    - Minimale Karenzzeit, wann wieder versucht werden darf (T_{MIN})
    - Maximale Karenzzeit (T_{MAX}), wann spätestens versucht werden muss, sonst verfällt das Tripel
    - Max. Verbleib des Tripel auf Whitelist, wenn Greylisting überwunden wurde
  - Wenn Tripel bekannt, durchlassen und TTL in Whitelist zurücksetzen
  - Wenn unbekannt, temporären 4xx Fehler melden und Tripel merken
  - Wenn $T_{MIN} < T < T_{MAX}$ durchlassen und Tripel auf Whitelist setzen

Greylisting



Greylisting

- KISS*-Lösung: greylistd
- Ohne Datenbank
  - Greylist liegt im RAM
  - Wird ab und zu auf Platte geschrieben und beim Reboot restauriert
- Man kann auch nur mit IPs arbeiten
- Hervorragende Integration in Exim, dafür wurde er entworfen!
-
```
[timeouts]
retryMin     = 300        # 5 Minuten min. Karenzzeit Tmin
retryMax     = 28800      # 8 Sunden max. Karenzzeit  Tmax
expire       = 1814400    # 3 Wochen Verbleib auf Whitelist
[socket]
path         = /var/run/greylistd/socket
mode         = 0660
[data]
update       = 600    # Dump- u. Reinigungsintervall d. Whitelist
statefile    = /var/lib/greylistd/states
tripletfile  = /var/lib/greylistd/triplets
savetriplets = true
```

* **K**eep **I**t **S**mall and **S**imple

Geylisting: exim.conf

- `acl_smtp_rcpt = check_greylist`
  `[...]`
- `check_greylist:`

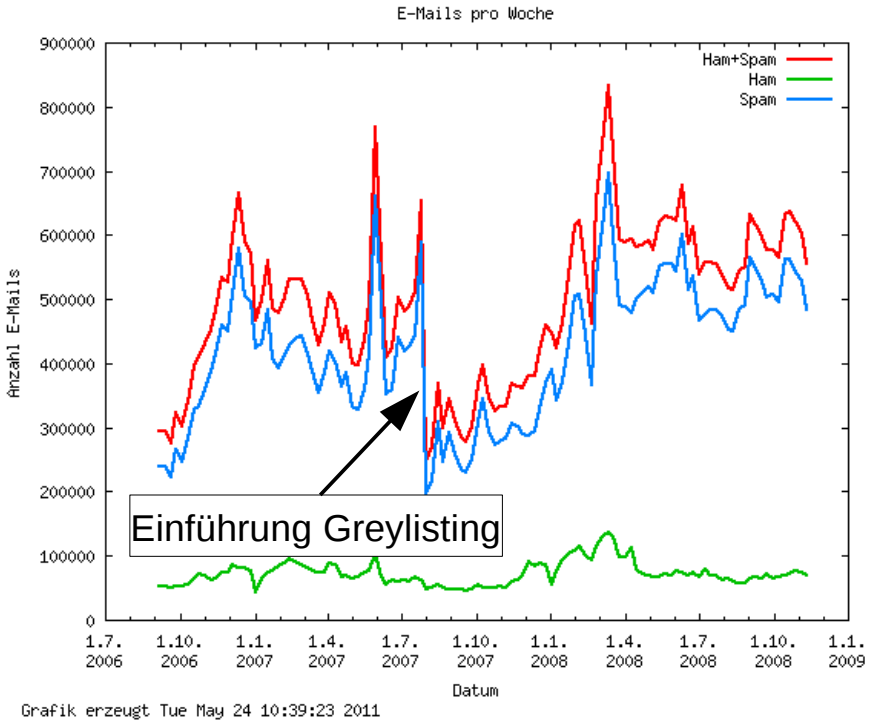
```
  defer
    message        = Greylisted: $sender_host_address is not \
                     yet authorized to deliver mail from
                     <$sender_address> to <$local_part>.
    log_message    = ATIS-greylisted.
    !hosts         = : +relay_hosts : \
                     ${if exists {/opt/exim/local/whitehosts}\
                                 {/opt/exim/local/whitehosts}{}}
    !authenticated = *
    condition      = ${readsocket{/var/run/greylistd/socket}\
                                 {--grey $sender_host_address}\
                                         {5s}{}{false}}
```

- Es wird sich nur die IP gemerkt
- Eine lokale statische Whitelist wird berücksichtigt
- Wer sich authentifiziert hat, darf sofort durch

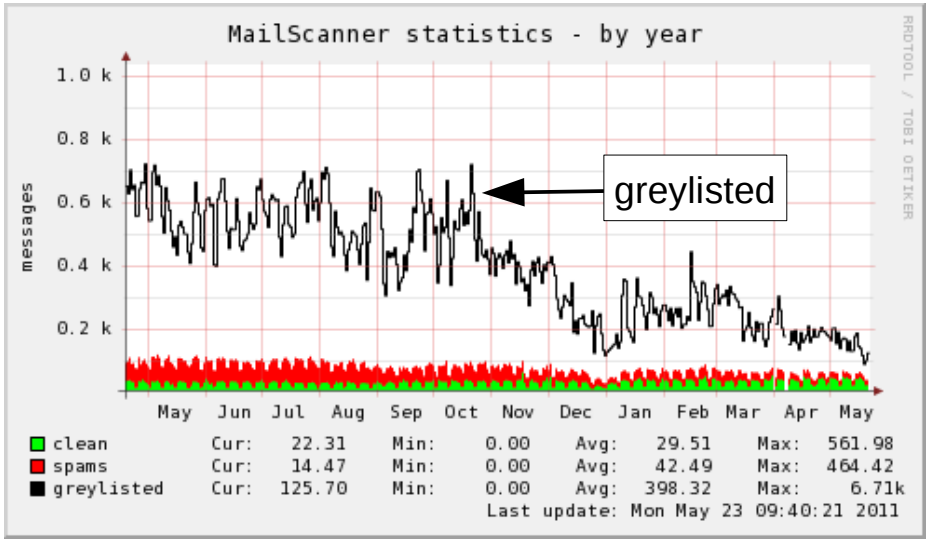
Greylisting

- Was bringt es?

2007



2010/2011



Agenda

- Vorstellung
- Exim Grundlagen

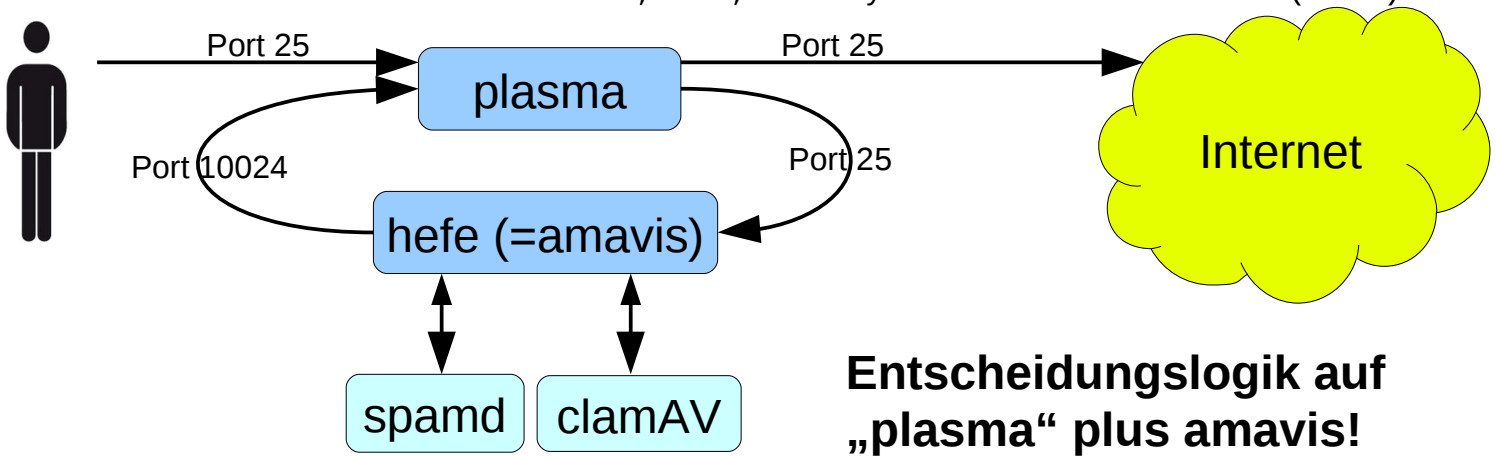
- Exim und LDAP
  - SMTP-Auth mit LDAP
  - LDAP-Mailrouting

- Keep the bad guys out
  - Denial-of-Service-Abwehr
  - Ratelimiting
  - Greylisting

- Exim und (kein) Amavis
  - Viren
  - Spamassassin
  - Spam-Relay-Vermeidung

Amavis loop from hell

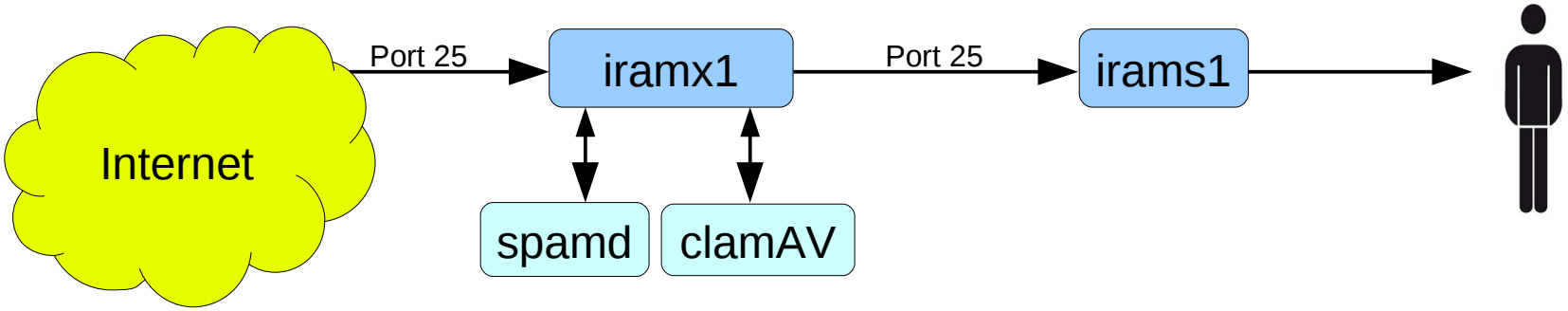
```
Received: from iramx1.ira.uni-karlsruhe.de ([141.3.10.80])
          by irams1.ira.uni-karlsruhe.de with esmtps port 25
          for <ohopp@ira.uka.de>
          id 1QOXQ5-0000RP-G7; Mon, 23 May 2011 17:51:49 +0200
Received: from mail.jpberlin.de ([213.203.238.11])
          by iramx1.ira.uni-karlsruhe.de with esmtps port 25
          id 1QOXPu-0007ef-8B
          for <Olaf.Hopp@atis.uka.de>; Mon, 23 May 2011 17:51:48 +0200
Received: from hefe.heinlein-support.de (hefe.heinlein-support.de [91.198.250.172])
          by plasma.jpberlin.de (Postfix) with ESMTP id 4528C83885;
          Mon, 23 May 2011 17:51:28 +0200 (CEST)
X-Virus-Scanned: amavisd-new at heinlein-support.de
Received: from plasma.jpberlin.de ([91.198.250.140])
          by hefe.heinlein-support.de (hefe.heinlein-support.de [91.198.250.172])
          (amavisd-new, port 10024)
          with ESMTP id ViaNvRLJtZkT; Mon, 23 May 2011 17:51:17 +0200 (CEST)
```



Entscheidungslogik auf „plasma“ plus amavis!

Exim ohne Amavis

```
Received: from iramx1.ira.uni-karlsruhe.de ([141.3.10.80])
          by irams1.ira.uni-karlsruhe.de with esmtps port 25
          for <ohopp@ira.uka.de>
          id 1QOXQ5-0000RP-G7; Mon, 23 May 2011 17:51:49 +0200
Received: from mail.jpberlin.de ([213.203.238.11])
          by iramx1.ira.uni-karlsruhe.de with esmtps port 25
          id 1QOXPu-0007ef-8B
          for <Olaf.Hopp@atis.uka.de>; Mon, 23 May 2011 17:51:48 +0200
Received: from hefe.heinlein-support.de (hefe.heinlein-support.de [91.198.250.172])
          by plasma.jpberlin.de (Postfix) with ESMTP id 4528C83885;
          Mon, 23 May 2011 17:51:28 +0200 (CEST)
X-Virus-Scanned: amavisd-new at heinlein-support.de
Received: from plasma.jpberlin.de ([91.198.250.140])
          by hefe.heinlein-support.de (hefe.heinlein-support.de [91.198.250.172])
          (amavisd-new, port 10024)
          with ESMTP id ViaNvRLJtZkT; Mon, 23 May 2011 17:51:17 +0200 (CEST)
```



2 Mail-HOPs von der „Haustür“ bis ins Postfach inklusive Viren- und Spamfilter!

Agenda

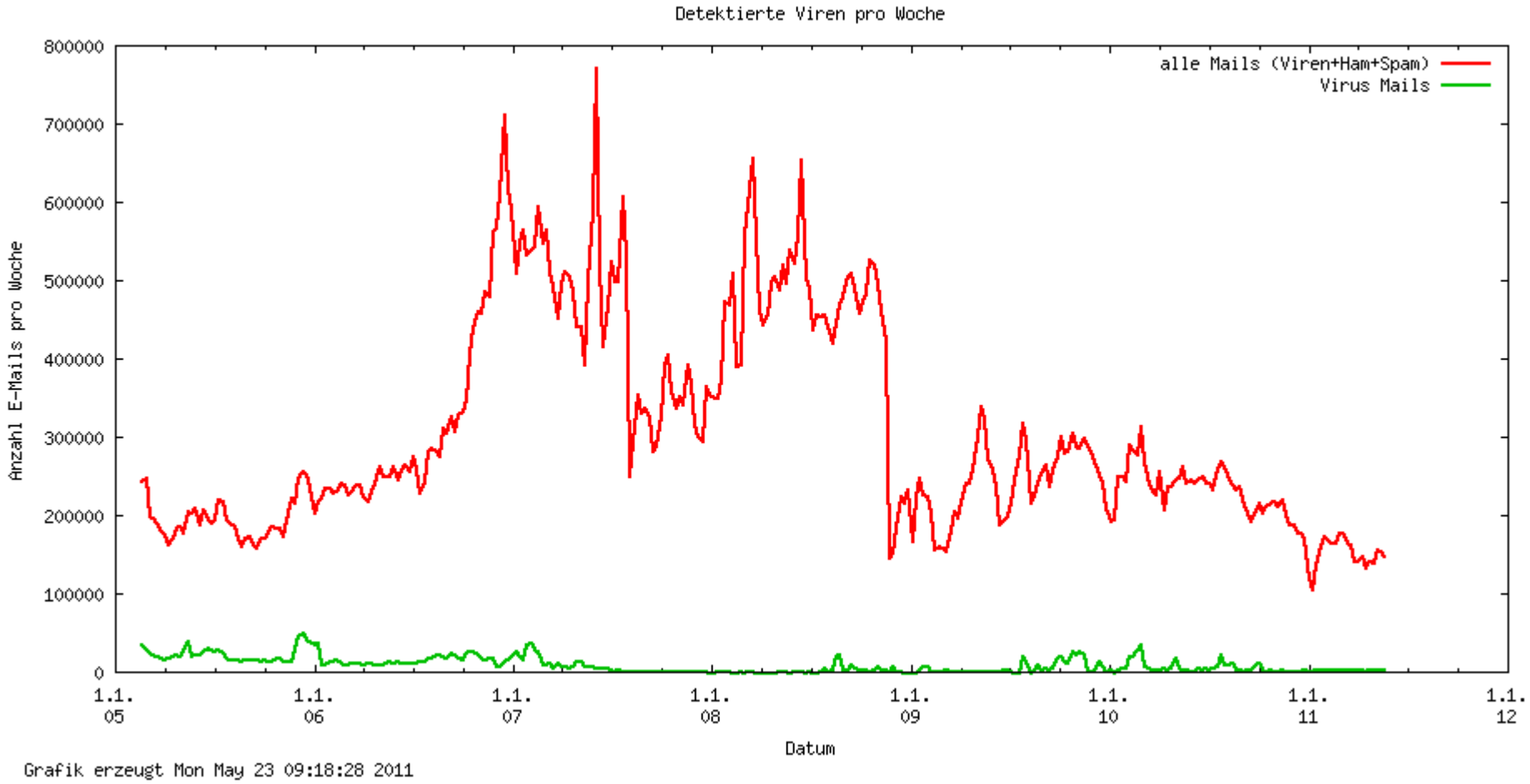
- Vorstellung
- Exim Grundlagen

- Exim und LDAP
  - SMTP-Auth mit LDAP
  - LDAP-Mailrouting

- Keep the bad guys out
  - Denial-of-Service-Abwehr
  - Ratelimiting
  - Greylisting

- Exim und (kein) Amavis
  - Viren
  - Spamassassin
  - Spam-Relay-Vermeidung

Viren


Detektierte Viren pro Woche
alle Mails (Viren+Ham+Spam)
Virus Mails
Anzahl E-Mails pro Woche
800000
700000
600000
500000
400000
300000
200000
100000
0
1.1. 05
1.1. 06
1.1. 07
1.1. 08
1.1. 09
1.1. 10
1.1. 11
1.1. 12
Datum
Grafik erzeugt Mon May 23 09:18:28 2011

Viren: Virenscanner am Beispiel ClamAV

- Viren werden abgewiesen – auch im F&L-Bereich!
- ClamAV:
  - In /etc/clamd.conf:
    `LocalSocket /var/run/clamav/clamd.socket`
  - **`/usr/sbin/clamd &`**

- exim.conf:

```
av_scanner    = clamd:/var/run/clamav/clamd.socket
acl_smtp_data = check_message
[…]
begin acl
check_message:
  deny    message    = VIRUS ($malware_name)
          malware    = *
```

- **Fertig !**
- ... und was ist mit amavis? Wir haben **`malware = * !`**

Viren: EXIM mit zwei Virenscannern

- 2 Virenscanner in Reihe am Beispiel ClamAV und Kaspersky
- ```
  av_scanner    = $acl_m9
  acl_smtp_data = check_message
  ```

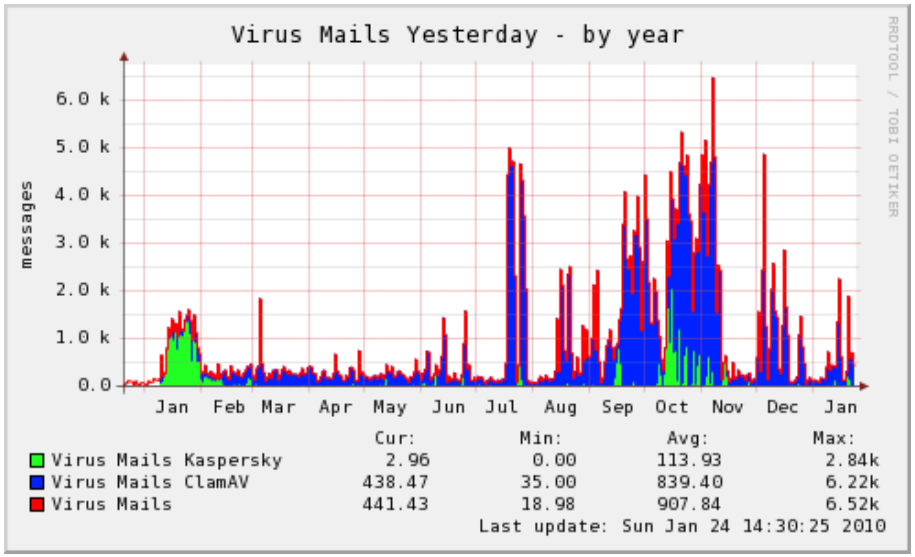
- ```
  beginn acl
  check_message:
    deny    message    = VIRUS ($malware_name) (CAV)
            set acl_m9 = clamd:/var/run/clamav/clamd.socket
            malware    = *

    deny    message    = VIRUS ($malware_name) (KAV)
            set acl_m9 = aveserver:/var/run/kav/aveserver
            malware    = *
  ```

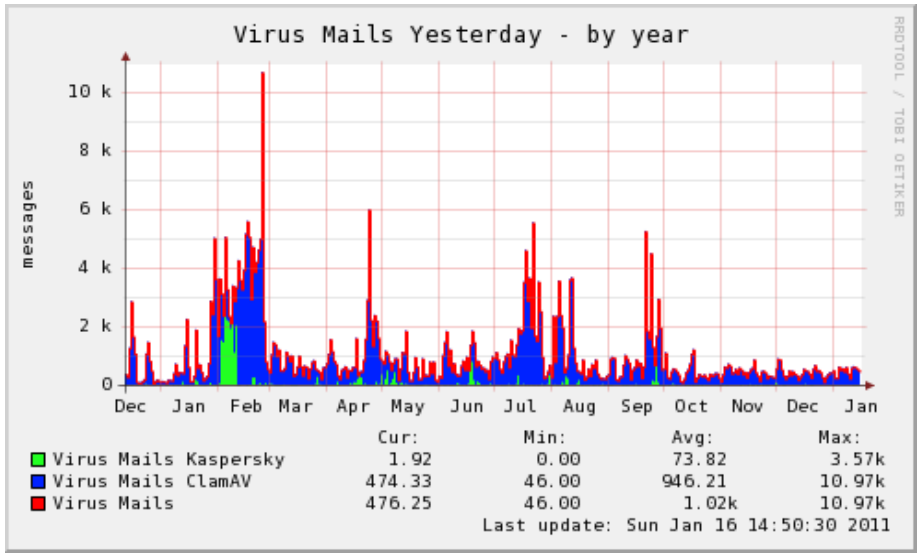
Viren: ClamAV vs. Kaspersky

- Was bringt es?

2009



2010



Agenda

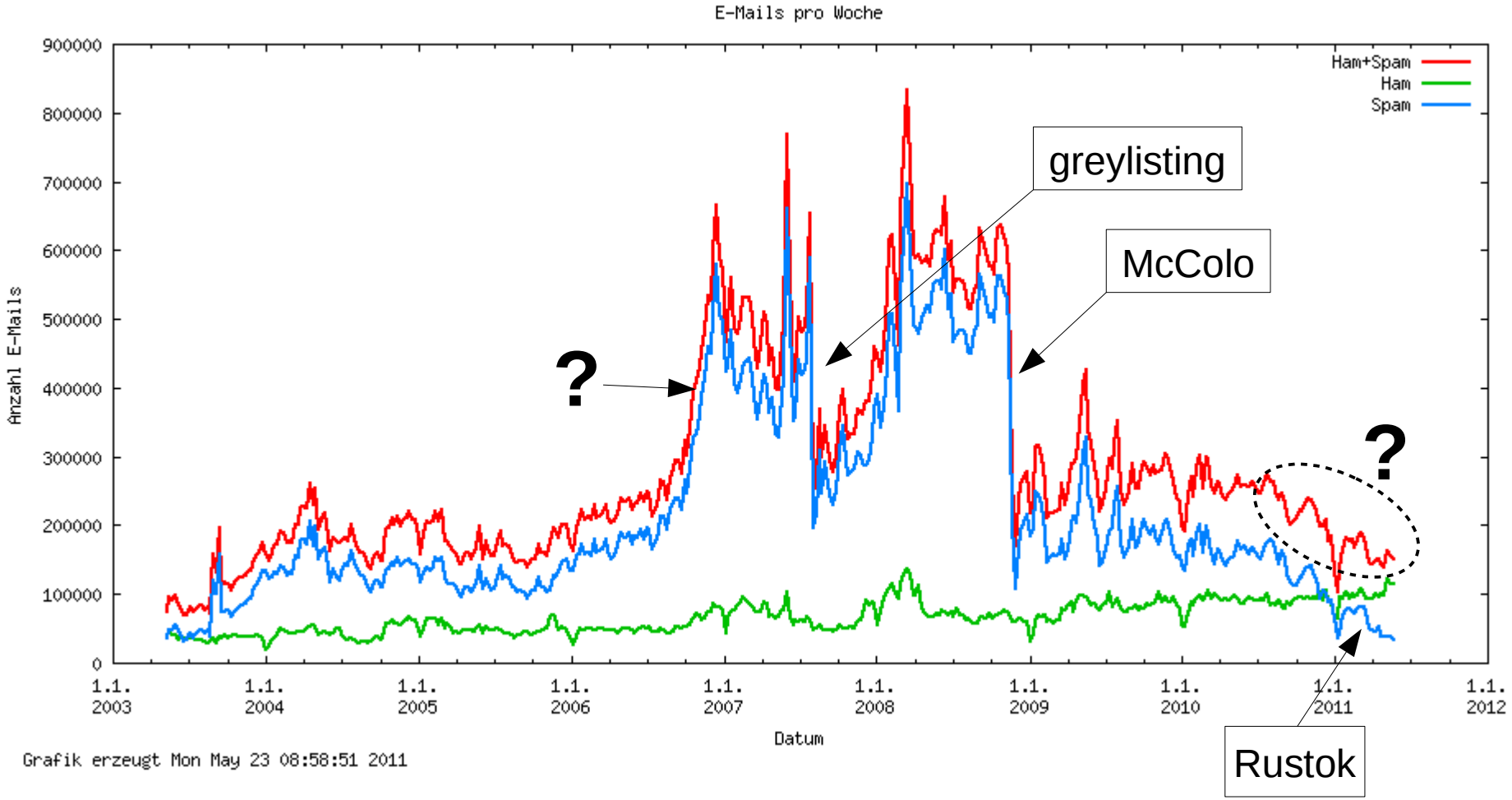
- Vorstellung
- Exim Grundlagen

- Exim und LDAP
  - SMTP-Auth mit LDAP
  - LDAP-Mailrouting

- Keep the bad guys out
  - Denial-of-Service-Abwehr
  - Ratelimiting
  - Greylisting

- Exim und (kein) Amavis
  - Viren
  - Spamassassin
  - Spam-Relay-Vermeidung

Spam: Worum geht es?



Spamassassin

- Integration von Spamassassin und Exim
- /etc/mail/spamassassin/local.cf:

```
report_safe 0           # Mailbody nicht verändern
clear_report_template   # keine Mailheader hinzufügen !
report _SUMMARY_
```

- Eine „spam = nobody“ Zeile im acl_smtp_data belegt automagisch die Variablen **$spam_score, $spam_score_int, $spam_bar** u. **$spam_report**

- exim.conf:

```
spamd_address = /opt/exim/spamassassin/sa.socket
acl_smtp_data = check_message
[...]
begin acl
    check_message:
```

(Fortsetzung nächste Seite)

Spamassassin

(Fortsetzung von vorheriger Seite)

```
check_message:
  #                                          auch Ham wird getaggt
  warn message = X-ATIS-Spam-Status: No\n\
                 X-ATIS-Spam-Score: $spam_score ($spam_bar)\n\
                 X-ATIS-Spam-Report: $spam_report\n\
         spam = nobody
    condition = ${if < {$spam_score_int} {50}{1}{0}}

  #                                                tagging von Spam
  warn message = X-ATIS-Spam-Status: Yes\n\
                 X-ATIS-Spam-Score: $spam_score ($spam_bar)\n\
                 X-ATIS-Spam-Report: $spam_report\n\
                 X-ATIS-Spamflag: YES
         spam = nobody
    condition = ${if >= {$spam_score_int} {50}{1}{0}}
```

- ... und Amavis? Wir haben **`spam = nobody !`**

Spamassassin: Router

- Zwei verschiedene Router für Spam und Ham, die je nach Spam-Flag verschiedene Transports aufrufen
- Berücksichtigt auch die, die ihren Spam in der INBOX haben wollen
- ```
  begin router
  spamuser:
      driver      = accept
      check_local_user
      condition   = ${if match{$h_X-ATIS-Spam-Flag:}{YES}{1}{0}}
      local_parts = !mueller:!meier    # bekommen Spam in die INBOX
      transport   = spam_delivery

  userforward:[...]

  localuser:
      driver      = accept
      check_local_user
      transport   = local_delivery
  ```
- **Achtung**: ggfs. .forward und vacation erst **nach** dem Spam auswerten, sonst droht Kollateral-Spam. ***Order does matter !***
  ```

Spamassassin: Transports

- Die dazugehörigen Transports
- **begin transports**

```
local_delivery:
    driver    = appendfile
    directory = /var/spool/mail/$local_part
    maildir_format

spam_delivery:
    driver    = appendfile
    directory = /var/spool/mail/$local_part/.spambox
    maildir_format
```

Agenda

- Vorstellung
- Exim Grundlagen

- Exim und LDAP
  - SMTP-Auth mit LDAP
  - LDAP-Mailrouting

- Keep the bad guys out
  - Denial-of-Service-Abwehr
  - Ratelimiting
  - Greylisting

- Exim und (kein) Amavis
  - Viren
  - Spamassassin
  - Spam-Relay-Vermeidung

Spam-Relay-Vermeidung

- Problem: Spam wird angenommen und u.U. weitergeleitet
- Hintergrund:
  - viele Alumnis
  - Uni-Adresse steht in Publikationen
  - Mailkonto wird gelöscht und im LDAP durch eine Forwarding-Adresse ersetzt
- Dadurch würde auch Spam weiterleitet werden
- Erzeugt negative Reputation bei Dritten $\Longrightarrow$ Schlecht!
- Diesen Spam sollte man ablehnen
- Aber nur dann, wenn mindestens ein nicht-lokaler Empfänger dabei ist
- Problem: über die Annahme entscheidet man „nach“ dem Envelope noch vor der Data-Phase. Dort wissen wir zwar im Prinzip, ob ein nicht-lokaler Empfänger dabei ist, aber wir wissen noch nicht ob es Spam ist

Spam-Relay-Vermeidung

- Schwierigkeit:
  - Man kann Variablen von den ACLs in die Router „hinunter“ reichen und zwischen den ACLs hin- und herreichen
  - Man kann aber (fast) keine Daten vom Router in den acl_smtp_rcpt schieben und von dort in den acl_smtp_data weiterreichen
- Lösung:
  - Im acl_smtp_rcpt über **`verify = recipient`** den Empfänger überprüfen
  - Dabei werden die Router aufgerufen
  - $address_data im Router setzen, wenn ein nicht-lokaler Empfänger vorhanden ist
  - Im acl_smtp_rcpt $address_data auf eine acl-Variable legen
  - Im acl_smtp_data die Mail „deny“-en wenn
    - die Mail spammig ist
    - und die acl-Variable gesetzt ist

Spam-Relay-Vermeidung

- 2 Router:
  - Einer für lokale Adressen
  - Einer für entfernte Adressen

- **atis:**

```
driver       = manualroute
route_list   = atis.example.org imap.atis.example.org bydns
transport    = remote_smtp_in
address_data = 0
```

- **lookuphost:**

```
driver       = dnslookup
domains      = ! +local_domains
transport    = remote_smtp_out
address_data = 1
```

Spam-Relay-Vermeidung

- ```
  acl_smtp_rcpt:
  accept  domains       = +local_domains
          endpass
          message       = Unknown user.
          verify        = recipient
          set acl_m0 = ${eval: $acl_m0 + $address_data }
  ```

- ```
  acl_smtp_data:
  deny  message   = SPAMRELAY: No spam relay to foreign addresses
        condition = ${if >{$acl_m0}{0} {1}{0}}
        condition = ${if >{$spam_score_int}{75}{1}{0}}
  ```

- Wenn mindestens ein nicht-lokaler Empfänger im Envelope steht und die Mail mehr als 7.5 Spam-Score-Punkte hat wird sie abgewiesen
- ~20 Spam-Mails werden täglich weitergeleitet, ~750 werden geblockt
- Könnte man sich sparen, wenn man Spam generell gleich abweisen würde... 

Agenda

- Vorstellung
- Exim Grundlagen

- Exim und LDAP
  - SMTP-Auth mit LDAP
  - LDAP-Mailrouting

- Keep the bad guys out
  - Denial-of-Service-Abwehr
  - Ratelimiting
  - Greylisting

- Exim und (kein) Amavis
  - Viren
  - Spamassassin
  - Spam-Relay-Vermeidung

Geschafft !