

Exim und Perl

12. Deutscher Perl-Workshop

Kurt Jaeger, pi@nepustil.net

<http://nepustil.net/>

Schorndorf, 8. Juni 2010



Übersicht

- ▶ Warum ?
- ▶ Exim
- ▶ Perl in Exim
- ▶ Beispiel

Kurzvorstellung

- ▶ Geschäftsführer regionaler ISP
(Dr.-Ing. Nepustil & Co. GmbH)
- ▶ Techniklastig
- ▶ Exim seit 2003
- ▶ Perl seit 1992

Problem: Mail is dead – is Mail dead ?

- ▶ Spam
- ▶ Re: AW: Re: RE: Re: AW: AW: AW: Hallo
- ▶ wer würde Mail vermissen ?
- ▶ Mail als Medium verliert Akzeptanz
- ▶ Sind wir da, wo Usenet vor 10-15 Jahren war ?

Gelegenheit: Groupware-Integration

- ▶ Kontakte
- ▶ Termine
- ▶ Aufgaben
- ▶ Notizen
- ▶ mobiler Zugriff
- ▶ Multiclient
- ▶ Webmail
- ▶ Workflow: RT, OTRS

Wie funktioniert Exim ?

- ▶ Konfigurationsdatei (ASCII)
- ▶ spool-Verzeichnis mit temporären Dateien
/var/spool/exim/
- ▶ Mail-Verzeichnis
- ▶ Logfiles
/var/log/exim/exim.YYYYMMDD

Konfigurationsdatei

- ▶ relativ gross, nicht trivial (40-50 KB)
- ▶ /usr/local/etc/exim/configure
- ▶ Durch .include auch aufteilbar
- ▶ conditional skips (.ifdef)
- ▶ Hauptbereich und sechs Unterbereiche
 - ▶ main: Definition von Optionen, Variablen und Macros
 - ▶ ACLs
 - ▶ routers
 - ▶ transports
 - ▶ retry, rewrite, authenticators

Variablen und Listen

- ▶ boolean
- ▶ integer
- ▶ strings
- ▶ lists
 - ▶ domain Listen
 - ▶ host Listen: Domains oder IPs
 - ▶ adressen
 - ▶ local parts (local-part@dom.ain)

Routers und Transports

Router

- ▶ Entscheidet, mit welchem Transportmechanismus es weitergeht
- ▶ Reihenfolge der router in configfile wichtig
- ▶ Lookups und Expansions
- ▶ z.B. LDAP, DNS, Blacklists usw

Transport

- ▶ Implementiert Transportmechanismus
- ▶ Viele Optionen

ACLs

http://www.exim.org/exim-html-current/doc/html/spec_html/index.html#toc0310

- ▶ viele Stellen, an denen geprüft wird
- ▶ `acl_smtp_connect`
- ▶ `acl_smtp_predata`
- ▶ `acl_smtp_data`
- ▶ ...
- ▶ insgesamt 18 ACL-Hooks

Was macht Exim ? Was macht Perl ?

Exim:

- ▶ SMTP und Optionen (TLS, AUTH, EHLO usw)
- ▶ Storage
- ▶ Locking
- ▶ Logging
- ▶ Timeout und Retry
- ▶ Datentypen bereitstellen

Perl:

- ▶ Database Lookups
- ▶ Database Update (Spam, Groupware)
- ▶ Verarbeitungslogik
- ▶ Reguläre Ausdrücke

```
spamcheck:
  no_verify
  domains = +virtualdomains
  require_files = /mail/$domain/Conf/${local_part}.spam
  condition = \
    ${if or \
      { \
        {eq {$received_protocol}{spam-scanned}} \
        {and { \
          {exists{/mail/$domain/Conf/${local_part}.whitelis
          {!eq{${lookup{${lc:$sender_address}}\
            nwildlsearch{/mail/$domain/Conf/${local_part}.wl
          }\
        }}}{0}{1} \
      }
  driver = accept
  transport = spamcheck
```



Perl einbinden

http://www.exim.org/exim-html-current/doc/html/spec_html/ch12.html

- ▶ `perl_at_start = yes`
- ▶ `perl_startup = do '/etc/allperlyouneed.pl'`
- ▶ Funktion aufrufen:

```
${perl{foo}{argument1}{argument2} ... }
```

- ▶ Maximal 8 Parameter
- ▶ Zugriff auf Exim-Variablen

```
my $lp = Exim::expand_string('$local_part');
```
- ▶ `Exim::log_write()`
schreibt ins Exim Logfile

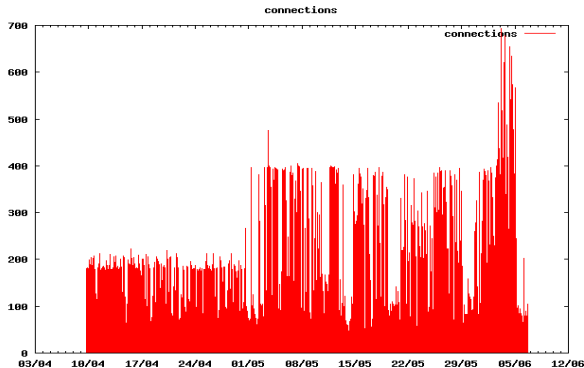


Router: Beispiel mit Perl

```
spamcheck:  
  no_verify  
  condition = ${perl{spamcheck}}  
  driver = accept  
  transport = spamcheck
```

Warum Perl ?

- ▶ postmaster Job ist anstrengend
- ▶ ständig neue Angriffe



- ▶ Wenn's nicht tut: extrem zeitkritisch
- ▶ perl gets the job done

Weitere Anforderungen

- ▶ IPv6
- ▶ Archivierung
- ▶ Verschlüsselung
- ▶ white/grey/blacklisting via IP, Domain, Adresse, Header, Body, ...
- ▶ Individuell pro Server, Domain, User, Tageszeit, Plattenplatz, Zeichensatz, ...
- ▶ Mehr Reputation
- ▶ Sieve (RFC5228)
- ▶ Calendering/Scheduling, analog MS Exchange
- ▶ End-to-End Identitäten ?

Fragen ?

Fragen ?

