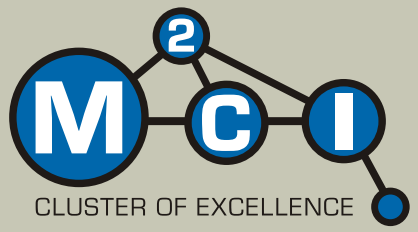

max planck institut informatik

MAX-PLANCK-GESELLSCHAFT

Max Planck Institute for Software Systems

Exim ohne Amavis

Patrick Cernko

pcernko@mpi-sws.org

Max-Planck-Institute
für Softwaresysteme & Informatik
Cluster of Excellence „Multimodal Computing and Interaction”
Saarbrücken & Kaiserslautern

2. Juli 2009

Übersicht

- 1. Teil: Ein paar Grundlagen über Exim
  - Allgemeines, Entstehung, Hintergründe
  - Konzepte von Exim
    - Expansions-Ausdrücke: *Exim's Programmiersprache*
    - ACLs im Detail: „*Wer darf was?*"
    - Kommandozeile & Tools: *Aufruf, Tests, Debugging*
- 2. Teil: DeCon mit Exim
  - Entseuchung des eingehenden Mailverkehrs mit Hilfe von ClamAV, Sophos, SpamAssassin, Postgrey und policyd-weight
  - individiuell angepasst durch Benutzereinstellungen

Allgemeines, Entstehung, Hintergründe

- 1995 entwickelt von Philip Hazel
  University of Cambridge in England, PCRE Library
- Ursprünglich „**EX**perimental **I**nternet **M**ailer”
- Single-Binary Design Modell
  vgl. Sendmail™
- Aktuelle Version: 4.69, OpenSource, GPL
- Distributionen: Debian, Ubuntu, Red Hat, SUSE, Gentoo, FreeBSD, Solaris (CSW)
  Sourcecode portiert auf viele weitere Unix-Derivate

Exim-Konzepte: Expansions-Ausdrücke

- „Programmiersprache” von Exim

Exim-Konzepte: Expansions-Ausdrücke

- „Programmiersprache” von Exim
- Großes Set an Funktionen & Variablen

```
$authenticated_id, $load_average, $smtp_count_at_connection_start, $spool_space,

${substr{6}{2}{$tod_logfile}}
```

  - *PCRE Library*

    ```
    ${sg {$spam_bar} {\N\+\N} {X}}
    ```

  - Lookup *in Files, DNS, DBs, NIS, LDAP, . . .*

    ```
    data = ${lookup mysql {SELECT email FROM login \

                    WHERE user="${quote_mysql:$local_part}"}{$value}fail}
    ```

  - *Programme, Sockets, Embedded Perl*

Exim-Konzepte: Expansions-Ausdrücke

- „Programmiersprache” von Exim
- Großes Set an Funktionen & Variablen

```
$authenticated_id, $load_average, $smtp_count_at_connection_start, $spool_space,

${substr{6}{2}{$tod_logfile}}
```

  - *PCRE Library*

    ```
    ${sg {$spam_bar} {\N\+\N} {X}}
    ```

  - Lookup *in Files, DNS, DBs, NIS, LDAP, . . .*

    ```
    data = ${lookup mysql {SELECT email FROM login \

                    WHERE user="${quote_mysql:$local_part}"}{$value}fail}
    ```

  - *Programme, Sockets, Embedded Perl*

- **Statische** Einstellungen werden **dynamisch**

Exim-Konzepte: Expansions-Ausdrücke (Beispiel)

```
tls_certificate = /etc/ssl/mail.mydomain.de.pem
```

- *Laut Dokumentation von* `tls_certificate` *auf den ersten Blick nur eine Datei möglich*

Exim-Konzepte: Expansions-Ausdrücke (Beispiel)

```
tls_certificate = \
  ${lookup dnsdb                          \
           {ptr=$interface_address} \
           {/etc/ssl/$value.pem}    \
           fail}
```

- *Laut Dokumentation von* `tls_certificate` *scheinbar nur eine Datei möglich*
- *dank Expansions-Ausdruck jedoch dynamisierbar*
- *Statt* `fail` *auch einfach ein weiterer geschachtelter Ausdruck*

Exim-Konzepte: ACLs

- „Access Control Lists” für (SMTP-)Befehle

  Beispiel:

```
acl_smtp_rcpt = acl_check_rcpt
begin acl
  acl_check_rcpt:
    accept
      hosts = +relay_from_hosts

    deny
```

- Vielzahl von Primitiven & Informationsquellen
  - `recipients, senders, domain, authenticated, hosts, verify = sender(/callout), ...`
  - *Sehr flexibel durch*

    `condition = `*`Expansions-Ausdruck`*

Exim-Konzepte: ACLs (Beispiele)

```
acl_check_mail:
  deny # deny servers, which do not start with a proper HELO/EHLO
    message = no HELO given before MAIL command
    condition = ${if def:sender_helo_name {no}{yes}}
  ...
  warn
    message = Forged IP detected in HELO: $sender_helo_name
    log_message = Forged IP detected in HELO: $sender_helo_name
    condition = ${if eq{$sender_helo_name}{$interface_address} {yes}{no}}
  ...

acl_check_rcpt:
  ...
  accept # accept relaying after authentication
    authenticated = *
  ...
  accept
    domains = +relay_to_domains
    endpass
    verify = recipient
  ...
```

Exim-Konzepte: ACLs

- Nicht nur Ablehnung!
  Logging, Header-Ergänzung, „State-Control”, temp. Ablehnung, Blackhole, gesteuerte Delays
- Spam- & Virenfilter
  - *Einbindbar über fertige Primitive oder vorbereitete Sockets*

  $\Rightarrow$ *Ablehnung zur SMTP-Zeit*

  oder: *Nur Tagging möglich (`warn`-Statement)*
- Ideal für eigene Anpassungen und Anforderungen
  - *Logging von Mails mit best. Eigenschaften*
  - *Privilegien-Finetuning*

  z.B. veraltete/unerwünschte MUAs ablehnen.

Exim-Konzepte: Kommandozeile & Tools

- Reichhaltige API
  Single-Binary, Daemon-Startup, Debugging, Management (Mail-Queue)
- Viele Testmöglichkeiten
  - *Expansions-Ausdrücke*
  - *Filter (→ `userforward`)*
  - *Relaying*
  - *Config-Values*
  - *Adressverifikation*
  - *Rewriting*

Exim-Konzepte: Kommandozeile & Tools

- *Sendmail*™ kompatibel
  - `/usr/lib/sendmail -> ../sbin/exim4`
  - *Die meisten Optionen werden „ignoriert”.*
    Nur zur Kompatibilität gegenüber alten Programmen

`-B<type>` This is a Sendmail option for selecting 7 or 8 bit processing. Exim is 8-bit clean; it ignores this option.

- weitere Tools
  - `exim_dumpdb, exim_fixdb, exim_tidydb` zum Modifizieren der internen BerkleyDBs
    z.B. Retry-Timeout nach Backend löschen
  - `exim_checkaccess`
    Wrapper für Relay-Tests

Übersicht (2. Teil)

- 1. Teil: Grundlagen
- 2. Teil: Decon mit Exim
  - Content-Scanner
    - ClamAV
    - Sophos
    - SpamAssassin
  - Postfix Policy Daemons mit Exim nutzen
    - Protocol-Kurzbeschreibung
    - Flexible Nutzungsmöglichkeiten
  - Maßnahmen gemäß Benutzereinstellungen treffen

Exim & Content-Scanner

- ClamAV direkt unterstützt

  `av_scanner = clamd:/var/run/clamav/clamd.ctl`

  - Open-Source (GPL)
  - gute Erkennungsraten
  - schnelle Updates

Exim & Content-Scanner

- ClamAV direkt unterstützt

  `av_scanner = clamd:/var/run/clamav/clamd.ctl`

  - Open-Source (GPL)
  - gute Erkennungsraten
  - schnelle Updates
- Sophos mit Sophie

  `av_scanner = sophie:/var/run/sophie`

  - Sophie *dämonisiert* Sophos
  - alt, aber funktioniert
  - Sophie linkt gegen libsavi.so (32bit!)
  - Vorsicht beim Sophos-Update

Exim & Content-Scanner

- SpamAssassin

  `spamd_address = /var/run/spamd.sock`

  - Einfache Abfrage des laufenden Dämons
  - Redundanz-Setup möglich
- Scan-Ergebnisse in Variablen bzw. ACL-Primitiven abrufbar

  `malware = <pattern>, $malware_name,`
  `spam = <user>[:true], $spam_score,`
  `$spam_bar, $spam_report`
- Keine vordefinierten *Entscheidungen*

$\Rightarrow$ Flexibilität bei der Verwertung

Exim & Postfix Policy Daemons

- Policy Daemon: externe Policy-Erweiterung für Postfix über definierte API, z.B.:
  - postgrey: Greylisting
  - policyd-weight: Gewichtete Beurteilung von Blacklists und Formfehlern
- Kommunikation via Unix-/TCP-Sockets
  $\Rightarrow$ Benutzung von `${readsocket ...}`
- einfaches Request-Answer-Protocol
  - Request besteht aus Werten der SMTP-Connection
  - Answer formuliert eine resultierende Postfix-Aktion

Exim & Postfix Policy Daemons

- Request-Format

```
POSTFIX_POLICYD_PROTOCOL = \
    request=smtpd_access_policy\n\
    protocol_state=RCPT\n\
    protocol_name=${lc:$received_protocol}\n\
    helo_name=$sender_helo_name\n\
    queue_id=$message_exim_id\n\
    client_address=$sender_host_address\n\
    client_name=$sender_host_name\n\
    sender=$sender_address\n\
    recipient=$local_part@$domain\n\
    recipient_count=$rcpt_count\n\
    reverse_client_name=$sender_host_name\n\
    instance=$sender_host_address.$sender_address.$sender_helo_name\n\
    \n
```

- mögliche Antworten:

```
OK, 4NN, 5NN, REJECT, DEFER_IF_REJECT, DEFER_IF_PERMIT, ...
```

Exim & Postfix Policy Daemons

- *Erweiterung* der API für Expansions-Ausdrücke durch Verwendung via `${readsocket ...}`

$\Rightarrow$ sehr flexible Verwendbarkeit der Dämonen

- Nur Logging
- Tagging (z.B. für SpamAssassin)
- Benutzerspezifisch
- Durch zusätzliche Faktoren begrenzbar
- Kombination möglich

Maßnahmen gemäß Benutzereinstellungen treffen

- bei uns: Hohe Anforderungen an Benutzerkontrolle
  - Spam-Ablehnung bei Benutzern umstritten
  - Greylisting vs. automatische Antwortsysteme
  - Extreme Spamwellen
  - Backscatter

Maßnahmen gemäß Benutzereinstellungen treffen

- bei uns: Hohe Anforderungen an Benutzerkontrolle
  - Spam-Ablehnung bei Benutzern umstritten
  - Greylisting vs. automatische Antwortsysteme
  - Extreme Spamwellen
  - Backscatter

$\Rightarrow$ *No-Malware* (statt Quarantäne)

Maßnahmen gemäß Benutzereinstellungen treffen

- bei uns: Hohe Anforderungen an Benutzerkontrolle
  - Spam-Ablehnung bei Benutzern umstritten
  - Greylisting vs. automatische Antwortsysteme
  - Extreme Spamwellen
  - Backscatter

$\Rightarrow$ *No-Malware* (statt Quarantäne)

$\Rightarrow$ *Safe-Greylisting* (nur für Verdächtige)

Maßnahmen gemäß Benutzereinstellungen treffen

- bei uns: Hohe Anforderungen an Benutzerkontrolle
  - Spam-Ablehnung bei Benutzern umstritten
  - Greylisting vs. automatische Antwortsysteme
  - Extreme Spamwellen
  - Backscatter

$\Rightarrow$ *No-Malware* (statt Quarantäne)

$\Rightarrow$ *Safe-Greylisting* (nur für Verdächtige)

$\Rightarrow$ *Spam-Deny* nach Punkten konfigurierbar

Maßnahmen gemäß Benutzereinstellungen treffen

- bei uns: Hohe Anforderungen an Benutzerkontrolle
  - Spam-Ablehnung bei Benutzern umstritten
  - Greylisting vs. automatische Antwortsysteme
  - Extreme Spamwellen
  - Backscatter

⇒ *No-Malware* (statt Quarantäne)

⇒ *Safe-Greylisting* (nur für Verdächtige)

⇒ *Spam-Deny* nach Punkten konfigurierbar

⇒ *Bounce-Protection*

Maßnahmen treffen: No-Malware

- Content-Scan mit Virenscannern
  $\Rightarrow$ alle Viren werden abgelehnt
- Effizient durch Scanner im Daemon-Modus
- Sehr hohe Erkennungsrate durch Kombination unterschiedlicher Scanner
- Keine Reibungsverluste durch Amavis als Middleware
  - Flexible Reaktionsmöglichkeiten bei Ausfall eines Scanners
  - Einheitlichere Konfiguration
  - Kein 64/32bit-Problem mit Sophos bzw. Sophie
  - *„Ein Rad weniger im Getriebe.”*

Maßnahmen treffen: Safe-Greylisting

- Greylisting mit Blacklisting kombiniert
  $\Rightarrow$ Nur die *bösen Jungs* müssen warten
- In `acl_smtp_rcpt`:
  1. Postgrey befragen $\Rightarrow$ greylisting
  2. Benutzereinstellung `FULLGREY` $\Rightarrow$ `defer`
  3. Blacklists prüfen $\Rightarrow$ listed
  4. Header setzen: blacklisted
  5. Benutzereinstellung `SAFEGREY` $\Rightarrow$ `defer`
  6. `accept`

Maßnahmen treffen: Spam-Deny

- Benutzer kann Spam-Deny-Level setzen
- In `acl_smtp_rcpt`:
  1. Benutzereinstellung `SPAMDENY` auslesen
  2. Vergleich mit ACL-Variable: `max($acl_m9, SPAMDENY)`
  3. Maximum in ACL-Variable speichern
- In `acl_smtp_data`:
  1. SpamAssassin befragen: spam-level
  2. Vergleich mit gespeicherter ACL-Variable $\Rightarrow$ `deny`
  3. Header setzen: Spam-Level, Spam-Report, Spam-Tests, Lotus Notes Domino: Blacklist-Tag
  4. `accept`

Maßnahmen treffen: Bounce-Protection

- Schutz vor Backscattern durch *Absendermarkierung*
- In `remote_smtp` Router:
  1. Benutzereinstellung `PROTECT`

  $\Rightarrow$ localpart um Bounce-Protection-Tag erweitern
- In `acl_smtp_rcpt`:
  1. Benutzereinstellung `PROTECT` $\Rightarrow$ weiter
  2. Bounce-Protection-Tag in Adresse prüfen:
     - OK $\Rightarrow$ `GOOD`-Header setzen
     - BAD $\Rightarrow$ `BAD`-Header setzen
- In SpamAssassin:
  - Punkte für `BAD`-Header

Resumee

+ Content-Scanner effizient eingebunden

Resumee

+ Content-Scanner effizient eingebunden
+ Postgrey & policyd-weight in Exim

Resumee

+ Content-Scanner effizient eingebunden
+ Postgrey & policyd-weight in Exim
+ sehr benutzerspezifisches Mail-Handling bei optimiertem Schutz

Resumee

+ Content-Scanner effizient eingebunden
+ Postgrey & policyd-weight in Exim
+ sehr benutzerspezifisches Mail-Handling bei optimiertem Schutz
– Aliase/Verteiler vs. *Benutzer* einstellungen

Resumee

+ Content-Scanner effizient eingebunden
+ Postgrey & policyd-weight in Exim
+ sehr benutzerspezifisches Mail-Handling bei optimiertem Schutz
– Aliase/Verteiler vs. *Benutzer* einstellungen
■ Exim unterstützt *Bounce address tag validation*

Resumee

+ Content-Scanner effizient eingebunden
+ Postgrey & policyd-weight in Exim
+ sehr benutzerspezifisches Mail-Handling bei optimiertem Schutz
– Aliase/Verteiler vs. *Benutzer* einstellungen
■ Exim unterstützt *Bounce address tag validation*

Vielen Dank

Literatur

- *Specification of the Exim Mail Transfer Agent*, Version 4.68
  Philip Hazel, 2007, http://exim.org/exim-pdf-current/doc/spec.pdf
- *The Exim SMTP Mail Server – Official Guide for Release 4*
  Philip Hazel, 2003, UIT Cambridge
- *Sophie: daemonising the Sophos virus engine*, Version 3.05
  Vanja Hrustic, 2002, http://www.clanfield.info/sophie/
- *Using Postgrey with Exim*, Guy Antony Halse, 2007, Postgrey distribution, http://postgrey.schweikert.ch/
- *Tweak your MTA: Spamschutz mit Tricks*, Tobias Eggendorfer, 2007, 3. Mailserver-Konferenz
- *Bounce Address Tag Validation (BATV)*, Network Working Group, 2008, http://mipassoc.org/batv/draft-levine-smtp-batv-01.html