Neuerungen in Postfix 2.8

Ralf Hildebrandt

Charité Universitätsmedizin Berlin

Linuxtag 2011 – Berlin, 13. Mai

- “Botnets now produce 95% of spam”
  `http://sanjose.bizjournals.com/sanjose/stories/2010/08/23/daily29.html`
- . . . hoch von 84 % im April.

- “Botnets now produce 95% of spam”
  `http://sanjose.bizjournals.com/sanjose/stories/2010/08/23/daily29.html`
- . . . hoch von 84 % im April.
- durch die Abschaltung von Rustock ist das Spamvolumen Mitte März 2011 um 33.6% gefallen

- “Botnets now produce 95% of spam”
  `http://sanjose.bizjournals.com/sanjose/stories/2010/08/23/daily29.html`
- . . . hoch von 84 % im April.
- durch die Abschaltung von Rustock ist das Spamvolumen Mitte März 2011 um 33.6% gefallen
- Kein Grund zur Entwarnung!

- “Botnets now produce 95% of spam” `http://sanjose.bizjournals.com/sanjose/stories/2010/08/23/daily29.html`
- . . . hoch von 84 % im April.
- durch die Abschaltung von Rustock ist das Spamvolumen Mitte März 2011 um 33.6% gefallen
- Kein Grund zur Entwarnung!

Effiziente Behandlung von Clients aus Botnetzen wäre sinnvoll!

Spezifische Gegenmaßnahmen

- DNSBLs

Spezifische Gegenmaßnahmen

- DNSBLs
  - Gibt es, ist aber ineffizient!

Spezifische Gegenmaßnahmen

- DNSBLs
  - Gibt es, ist aber ineffizient!
  - Ein `smtpd` pro Verbindung

Spezifische Gegenmaßnahmen

- DNSBLs
  - Gibt es, ist aber ineffizient!
  - Ein `smtpd` pro Verbindung
  - sequentielle Abfrage von DNSBLs

Spezifische Gegenmaßnahmen

- DNSBLs
  - Gibt es, ist aber ineffizient!
  - Ein `smtpd` pro Verbindung
  - sequentielle Abfrage von DNSBLs
- Greet delays

Spezifische Gegenmaßnahmen

- DNSBLs
  - Gibt es, ist aber ineffizient!
  - Ein `smtpd` pro Verbindung
  - sequentielle Abfrage von DNSBLs
- Greet delays
  - Gibt es, aber auch ineffizient!

Spezifische Gegenmaßnahmen

- DNSBLs
  - Gibt es, ist aber ineffizient!
  - Ein `smtpd` pro Verbindung
  - sequentielle Abfrage von DNSBLs
- Greet delays
  - Gibt es, aber auch ineffizient!
  - Nicht selektiv!

Spezifische Gegenmaßnahmen

- DNSBLs
  - Gibt es, ist aber ineffizient!
  - Ein `smtpd` pro Verbindung
  - sequentielle Abfrage von DNSBLs
- Greet delays
  - Gibt es, aber auch ineffizient!
  - Nicht selektiv!
- Greylisting

Spezifische Gegenmaßnahmen

- DNSBLs
  - Gibt es, ist aber ineffizient!
  - Ein `smtpd` pro Verbindung
  - sequentielle Abfrage von DNSBLs
- Greet delays
  - Gibt es, aber auch ineffizient!
  - Nicht selektiv!
- Greylisting
  - Gibt es

Spezifische Gegenmaßnahmen

- DNSBLs
  - Gibt es, ist aber ineffizient!
  - Ein `smtpd` pro Verbindung
  - sequentielle Abfrage von DNSBLs
- Greet delays
  - Gibt es, aber auch ineffizient!
  - Nicht selektiv!
- Greylisting
  - Gibt es :-)

Spezifische Gegenmaßnahmen

- DNSBLs
  - Gibt es, ist aber ineffizient!
  - Ein `smtpd` pro Verbindung
  - sequentielle Abfrage von DNSBLs
- Greet delays
  - Gibt es, aber auch ineffizient!
  - Nicht selektiv!
- Greylisting
  - Gibt es :-)

Spezifische Gegenmaßnahmen

- DNSBLs
  - Gibt es, ist aber ineffizient!
  - Ein `smtpd` pro Verbindung
  - sequentielle Abfrage von DNSBLs
- Greet delays
  - Gibt es, aber auch ineffizient!
  - Nicht selektiv!
- Greylisting
  - Gibt es :-)
- Erkennung von “Earlytalkern”

Spezifische Gegenmaßnahmen

- DNSBLs
  - Gibt es, ist aber ineffizient!
  - Ein `smtpd` pro Verbindung
  - sequentielle Abfrage von DNSBLs
- Greet delays
  - Gibt es, aber auch ineffizient!
  - Nicht selektiv!
- Greylisting
  - Gibt es :-)
- Erkennung von “Earlytalkern”
  - Gibt es noch gar nicht!

Spezifische Gegenmaßnahmen

- DNSBLs
  - Gibt es, ist aber ineffizient!
  - Ein `smtpd` pro Verbindung
  - sequentielle Abfrage von DNSBLs
- Greet delays
  - Gibt es, aber auch ineffizient!
  - Nicht selektiv!
- Greylisting
  - Gibt es :-)
- Erkennung von “Earlytalkern”
  - Gibt es noch gar nicht!
- Erkennung von anderem “abnormen” Verhalten

Spezifische Gegenmaßnahmen

- DNSBLs
  - Gibt es, ist aber ineffizient!
  - Ein `smtpd` pro Verbindung
  - sequentielle Abfrage von DNSBLs
- Greet delays
  - Gibt es, aber auch ineffizient!
  - Nicht selektiv!
- Greylisting
  - Gibt es :-)
- Erkennung von “Earlytalkern”
  - Gibt es noch gar nicht!
- Erkennung von anderem “abnormen” Verhalten
  - Gibt es noch gar nicht!

Spezifische Gegenmaßnahmen

- DNSBLs
  - Gibt es, ist aber ineffizient!
  - Ein `smtpd` pro Verbindung
  - sequentielle Abfrage von DNSBLs
- Greet delays
  - Gibt es, aber auch ineffizient!
  - Nicht selektiv!
- Greylisting
  - Gibt es :-)
- Erkennung von “Earlytalkern”
  - Gibt es noch gar nicht!
- Erkennung von anderem “abnormen” Verhalten
  - Gibt es noch gar nicht!

Aktuelle Entwicklung in Postfix

postscreen

ist der Codename für einen neuen Daemon der **vor** Postfixs `smtpd` sitzt und Verbindungen ausfiltert (“screening”).

Angriff der Zombiehorden

Angriff der Zombiehorden II

- Der Kernel queued Verbindungen
- Postfix arbeitet “nur” 100 Verbindungen (genauer: `default_process_limit`) simultan ab

→ Serverüberlastung

Angriff der Zombiehorden II

- Der Kernel queued Verbindungen
- Postfix arbeitet “nur” 100 Verbindungen (genauer: `default_process_limit`) simultan ab

→ Serverüberlastung

Symptome einer Serverüberlastung

- Clients erfahren eine extreme Verzögerung bevor Postfix antwortet
  - Clients geben auf bevor Postfix antwortet
- Postfix loggt viele “`lost connection`” Einträge
- Postfix ab Version 2.3 loggt “`all server ports busy`”-Warnungen

Symptome einer Serverüberlastung II

Im Log:

> *service “smtpd” has reached its process limit “100”: new smtpd clients may experience noticeable delays to avoid this condition, increase the process count in master.cf or reduce the service time per client*

Lösungsansatz

Ziele:

- Zombies von Postfixs `smtpd` fernhalten
- Erhöhung der Skalierbarkeit...

Lösungsansatz

Ziele:

- Zombies von Postfixs `smtpd` fernhalten
- Erhöhung der Skalierbarkeit. . .
  indem zeitintensive Operationen wie DNSBL Abfragen und SMTP Protokollchecks aus dem `smtpd` ausgelagert werden

Lösungsansatz

Ziele:

- Zombies von Postfixs `smtpd` fernhalten
- Erhöhung der Skalierbarkeit. . .
  indem zeitintensive Operationen wie DNSBL Abfragen und SMTP Protokollchecks aus dem `smtpd` ausgelagert werden
- Einführung eines speziellen Prozesses für genau diesen Zweck

Lösungsansatz

Ziele:

- Zombies von Postfixs `smtpd` fernhalten
- Erhöhung der Skalierbarkeit. . .
  indem zeitintensive Operationen wie DNSBL Abfragen und SMTP Protokollchecks aus dem `smtpd` ausgelagert werden
- Einführung eines speziellen Prozesses für genau diesen Zweck

postscreen - master.cf

Aus

```
smtp      inet  n       -       -       -       -       smtpd
```

wird

```
smtp      inet  n       -       -       -       1       postscreen
smtpd     pass  -       -       -       -       -       smtpd
```

Man beachte das Process Limit von 1 – `postscreen` ist wirklich nur `ein` einzelner Daemon.

postscreen - master.cf

Aus

```
smtp      inet  n       -       -       -       -       smtpd
```

wird

```
smtp      inet  n       -       -       -       1       postscreen
smtpd     pass  -       -       -       -       -       smtpd
```

Man beachte das Process Limit von 1 – `postscreen` ist wirklich nur `ein` einzelner Daemon.

Simple checks

Beinhalten keine Tests durch die in `postscreen` eingebaute SMTP-Engine:

- Client in Whitelist
- Client in Blacklist

Simple checks

Beinhalten keine Tests durch die in `postscreen` eingebaute SMTP-Engine:

- Client in Whitelist
- Client in Blacklist

Simple checks

Beinhalten keine Tests durch die in `postscreen` eingebaute SMTP-Engine:

- Client in Whitelist
- Client in Blacklist
- Client in einer oder mehreren DNSBLs

  With postscreen, DNSBL lookups happen in parallel and for multiple clients the same time, and making Postfix less vulnerable to overload problems.

Simple checks

Beinhalten keine Tests durch die in `postscreen` eingebaute SMTP-Engine:

- Client in Whitelist
- Client in Blacklist
- Client in einer oder mehreren DNSBLs

  With postscreen, DNSBL lookups happen in parallel and for multiple clients the same time, and making Postfix less vulnerable to overload problems.

- Erkennung von “Earlytalkers”

Simple checks

Beinhalten keine Tests durch die in `postscreen` eingebaute SMTP-Engine:

- Client in Whitelist
- Client in Blacklist
- Client in einer oder mehreren DNSBLs

  With postscreen, DNSBL lookups happen in parallel and for multiple clients the same time, and making Postfix less vulnerable to overload problems.

- Erkennung von “Earlytalkers”

Wenn ein Client die Tests “besteht” wird die Verbindung direkt an einen `smtpd` abgegeben.

Simple checks

Beinhalten keine Tests durch die in `postscreen` eingebaute SMTP-Engine:

- Client in Whitelist
- Client in Blacklist
- Client in einer oder mehreren DNSBLs

  With postscreen, DNSBL lookups happen in parallel and for multiple clients the same time, and making Postfix less vulnerable to overload problems.

- Erkennung von “Earlytalkers”

Wenn ein Client die Tests “besteht” wird die Verbindung direkt an einen `smtpd` abgegeben.

Deep protocol checks

Nutzen die in `postscreen` eingebaute SMTP-Engine:

- `postscreen_bare_newline_enable`
- `postscreen_non_smtp_command_enable`

Deep protocol checks

Nutzen die in `postscreen` eingebaute SMTP-Engine:

- `postscreen_bare_newline_enable`
- `postscreen_non_smtp_command_enable`
- `postscreen_pipelining_enable`

Deep protocol checks

Nutzen die in `postscreen` eingebaute SMTP-Engine:

- `postscreen_bare_newline_enable`
- `postscreen_non_smtp_command_enable`
- `postscreen_pipelining_enable`

Bedingt durch die Art der Tests läuft ein Client ersteinmal tief in die Protokollanalyse des `postscreen` daemons hinein.

Deep protocol checks

Nutzen die in `postscreen` eingebaute SMTP-Engine:

- `postscreen_bare_newline_enable`
- `postscreen_non_smtp_command_enable`
- `postscreen_pipelining_enable`

Bedingt durch die Art der Tests läuft ein Client ersteinmal tief in die Protokollanalyse des `postscreen` daemons hinein.
Es gibt dann keine Methode, um diesen Status an den `smtpd` abzugeben.

Deep protocol checks

Nutzen die in `postscreen` eingebaute SMTP-Engine:

- `postscreen_bare_newline_enable`
- `postscreen_non_smtp_command_enable`
- `postscreen_pipelining_enable`

Bedingt durch die Art der Tests läuft ein Client ersteinmal tief in die Protokollanalyse des `postscreen` daemons hinein.
Es gibt dann keine Methode, um diesen Status an den `smtpd` abzugeben.
Daher `TEMPFAIL` und führen einer Datenbank von “guten” Clients.

Deep protocol checks

Nutzen die in `postscreen` eingebaute SMTP-Engine:

- `postscreen_bare_newline_enable`
- `postscreen_non_smtp_command_enable`
- `postscreen_pipelining_enable`

Bedingt durch die Art der Tests läuft ein Client ersteinmal tief in die Protokollanalyse des `postscreen` daemons hinein.
Es gibt dann keine Methode, um diesen Status an den `smtpd` abzugeben.
Daher `TEMPFAIL` und führen einer Datenbank von “guten” Clients.

Deep protocol checks

Nutzen die in `postscreen` eingebaute SMTP-Engine:

- `postscreen_bare_newline_enable`
- `postscreen_non_smtp_command_enable`
- `postscreen_pipelining_enable`

Bedingt durch die Art der Tests läuft ein Client ersteinmal tief in die Protokollanalyse des `postscreen` daemons hinein.
Es gibt dann keine Methode, um diesen Status an den `smtpd` abzugeben.
Daher `TEMPFAIL` und führen einer Datenbank von “guten” Clients.

Greylisting für arme Leute

Das zuvor genannte Verhalten ist ein de-facto Greylisting basierend nur auf IP.

Wie erkenne ich meine Kunden?

- Kunden nutzen (normalerweise?) Port 25
  Da ein legitimer User in einem Dialup-Bereich für `postscreen` von einem Zombie ununterscheidbar ist...
- müssen die Kunden Port 587 nutzen (submission)

Wie erkenne ich meine Kunden?

- Kunden nutzen (normalerweise?) Port 25
  Da ein legitimer User in einem Dialup-Bereich für `postscreen` von einem Zombie ununterscheidbar ist...
- müssen die Kunden Port 587 nutzen (submission)

Viele ISPs blocken eh Port 25 outgoing, sodaß vielerorten Port 587 eh schon benutzt wird!

Wie erkenne ich meine Kunden?

- Kunden nutzen (normalerweise?) Port 25
  Da ein legitimer User in einem Dialup-Bereich für `postscreen` von einem Zombie ununterscheidbar ist...
- müssen die Kunden Port 587 nutzen (submission)

Viele ISPs blocken eh Port 25 outgoing, sodaß vielerorten Port 587 eh schon benutzt wird!

```
submission inet n       -       n       -       -       smtpd
  -o smtpd_enforce_tls=yes
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
```

Wie erkenne ich meine Kunden?

- Kunden nutzen (normalerweise?) Port 25
  Da ein legitimer User in einem Dialup-Bereich für `postscreen` von einem Zombie ununterscheidbar ist...
- müssen die Kunden Port 587 nutzen (submission)

Viele ISPs blocken eh Port 25 outgoing, sodaß vielerorten Port 587 eh schon benutzt wird!

```
submission inet n       -       n       -       -       smtpd
  -o smtpd_enforce_tls=yes
  -o smtpd_sasl_auth_enable=yes
  -o smtpd_client_restrictions=permit_sasl_authenticated,reject
```

postscreen - `main.cf`

```
postscreen_dnsbl_sites =
  zen.spamhaus.org*2, bl.spamcop.net
  b.barracudacentral.org, swl.spamhaus.org*-2
postscreen_dnsbl_threshold = 2
postscreen_access_list = permit_mynetworks
  cidr:/etc/postfix/postscreen_access.cidr

postscreen_bare_newline_enable = yes
postscreen_pipelining_enable = yes
postscreen_non_smtp_command_enable = yes

postscreen_greet_action = enforce
postscreen_dnsbl_action = enforce
postscreen_bare_newline_action = drop
```

```
Feb 21 13:45:33 mail postfix/postscreen[8534]: \
PASS OLD [195.140.185.23]:44965
```

Hier wurde ein “guter” Client im Cache gefunden

```
Feb 21 13:45:33 mail postfix/postscreen[8534]: \
PASS OLD [195.140.185.23]:44965
```

Hier wurde ein “guter” Client im Cache gefunden

postscreen - das Log

```
postscreen[8534]: CONNECT from [117.196.140.106]:11464
postscreen[8534]: PREGREET 22 after 0.4 from [117.196.140.106]:11464: HELO home-0cdf3c9fc3\r\
postscreen[8534]: DNSBL rank 1 for [117.196.140.106]:11464
postscreen[8534]: NOQUEUE: reject: RCPT from [117.196.140.106]:11464: 550 5.7.1 Service unava
```

Hier wurde ein “schlechter” Client in genau einer der `postscreen_dnsbl_sites` gefunden.

postscreen - das Log

```
postscreen[8534]: CONNECT from [117.196.140.106]:11464
postscreen[8534]: PREGREET 22 after 0.4 from [117.196.140.106]:11464: HELO home-0cdf3c9fc3\r\n
postscreen[8534]: DNSBL rank 1 for [117.196.140.106]:11464
postscreen[8534]: NOQUEUE: reject: RCPT from [117.196.140.106]:11464: 550 5.7.1 Service unava
```

Hier wurde ein “schlechter” Client in genau einer der `postscreen_dnsbl_sites` gefunden.

Die in `postscreen` eingebaute SMTP-Engine konnte envelope sender, recipient und HELO bestimmen!

postscreen - das Log

```
postscreen[8534]: CONNECT from [117.196.140.106]:11464
postscreen[8534]: PREGREET 22 after 0.4 from [117.196.140.106]:11464: HELO home-0cdf3c9fc3\r\
postscreen[8534]: DNSBL rank 1 for [117.196.140.106]:11464
postscreen[8534]: NOQUEUE: reject: RCPT from [117.196.140.106]:11464: 550 5.7.1 Service unava
```

Hier wurde ein “schlechter” Client in genau einer der `postscreen_dnsbl_sites` gefunden.
Die in `postscreen` eingebaute SMTP-Engine konnte envelope sender, recipient und HELO bestimmen!

bare newline detection

- bare newline detection

  Real spambots don't make this mistake anymore, but poorly-written software still does.

```
Feb 21 10:26:12 mail postfix/postscreen[27228]: BARE NEWLINE from [79.193.32.98]:62058
```

Böse Falle

Abfragen mit `echoping` scheitern bei Postfix-2.8.x mit aktiviertem postscreen:

```
Jan 26 15:38:01 mail postfix/postscreen[9288]:
   CONNECT from [46.182.18.28]:43024
Jan 26 15:38:05 mail postfix/postscreen[9288]:
   PREGREET 6 after 5 from [46.182.18.28]:43024: QUIT\r\n
Jan 26 15:38:05 mail postfix/postscreen[9288]:
   DISCONNECT [46.182.18.28]:43024
```

Böse Falle

Abfragen mit `echoping` scheitern bei Postfix-2.8.x mit aktiviertem postscreen:

```
Jan 26 15:38:01 mail postfix/postscreen[9288]:
   CONNECT from [46.182.18.28]:43024
Jan 26 15:38:05 mail postfix/postscreen[9288]:
   PREGREET 6 after 5 from [46.182.18.28]:43024: QUIT\r\n
Jan 26 15:38:05 mail postfix/postscreen[9288]:
   DISCONNECT [46.182.18.28]:43024
```

Man muss dann einfach whitelisten:

Böse Falle

Abfragen mit `echoping` scheitern bei Postfix-2.8.x mit aktiviertem postscreen:

```
Jan 26 15:38:01 mail postfix/postscreen[9288]:
   CONNECT from [46.182.18.28]:43024
Jan 26 15:38:05 mail postfix/postscreen[9288]:
   PREGREET 6 after 5 from [46.182.18.28]:43024: QUIT\r\n
Jan 26 15:38:05 mail postfix/postscreen[9288]:
   DISCONNECT [46.182.18.28]:43024
```

Man muss dann einfach whitelisten:

```
postscreen_access_list =
   permit_mynetworks
   cidr:/etc/postfix/postscreen_access.cidr
```

Böse Falle

Abfragen mit `echoping` scheitern bei Postfix-2.8.x mit aktiviertem postscreen:

```
Jan 26 15:38:01 mail postfix/postscreen[9288]:
   CONNECT from [46.182.18.28]:43024
Jan 26 15:38:05 mail postfix/postscreen[9288]:
   PREGREET 6 after 5 from [46.182.18.28]:43024: QUIT\r\n
Jan 26 15:38:05 mail postfix/postscreen[9288]:
   DISCONNECT [46.182.18.28]:43024
```

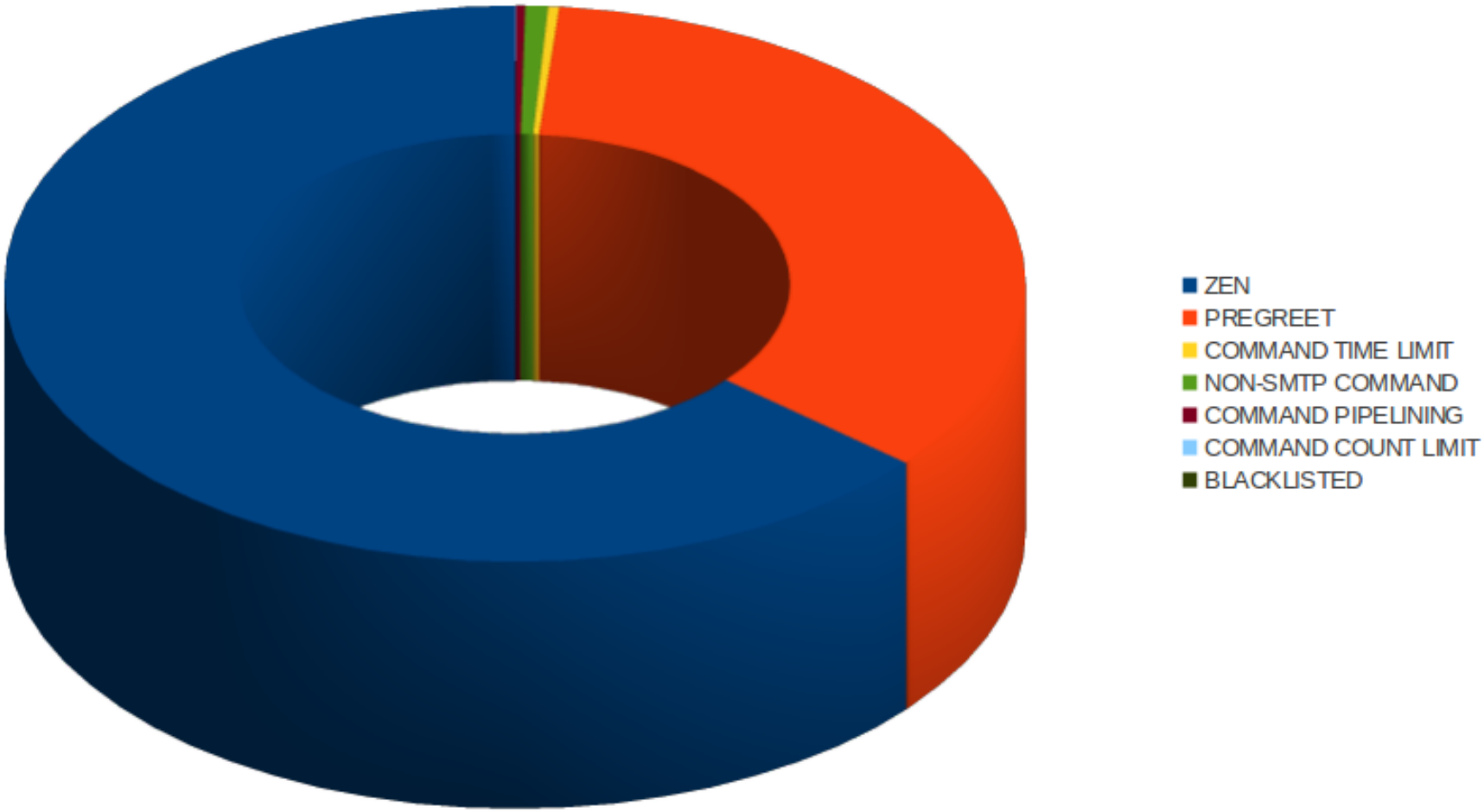
Man muss dann einfach whitelisten:

```
postscreen_access_list =
   permit_mynetworks
   cidr:/etc/postfix/postscreen_access.cidr
```

Weitere Entwicklungen

- Greylisting
  nach Implementierung der SMTP-Engine nur eine kleine Erweiterung des Cache

Analysen vom 4.5.2011 bis 10.5.2011:



Analysen vom 4.5.2011 bis 10.5.2011 `python.org`

- 245.936 Abweisungen durch postscreen (100%)
- 86.641 durch Pregreeting detection (ca. 35,2%)

Analysen vom 4.5.2011 bis 10.5.2011 `python.org`

- 245.936 Abweisungen durch postscreen (100%)
- 86.641 durch Pregreeting detection (ca. 35,2%)
- 159.295 durch DNSBL (`zen.spamhaus.org`) (ca. 64,7%)

Analysen vom 4.5.2011 bis 10.5.2011 `python.org`

- 245.936 Abweisungen durch postscreen (100%)
- 86.641 durch Pregreeting detection (ca. 35,2%)
- 159.295 durch DNSBL (`zen.spamhaus.org`) (ca. 64,7%)

Analysen vom 4.5.2011 bis 10.5.2011 `python.org`

- 245.936 Abweisungen durch postscreen (100%)
- 86.641 durch Pregreeting detection (ca. 35,2%)
- 159.295 durch DNSBL (`zen.spamhaus.org`) (ca. 64,7%)

Analysen vom 4.5.2011 bis 10.5.2011 `charite.de`

- 424.702 Abweisungen durch postscreen (100%)
- 114.813 durch Pregreeting detection (ca. 27,0%)

Analysen vom 4.5.2011 bis 10.5.2011 `charite.de`

- 424.702 Abweisungen durch postscreen (100%)
- 114.813 durch Pregreeting detection (ca. 27,0%)
- 309.889 durch DNSBL (`zen.spamhaus.org`) (ca. 72,9%)

Analysen vom 4.5.2011 bis 10.5.2011 `charite.de`

- 424.702 Abweisungen durch postscreen (100%)
- 114.813 durch Pregreeting detection (ca. 27,0%)
- 309.889 durch DNSBL (`zen.spamhaus.org`) (ca. 72,9%)

Das zeigt das Potential der SMTP-Engine in postscreen.

Analysen vom 4.5.2011 bis 10.5.2011 `charite.de`

- 424.702 Abweisungen durch postscreen (100%)
- 114.813 durch Pregreeting detection (ca. 27,0%)
- 309.889 durch DNSBL (`zen.spamhaus.org`) (ca. 72,9%)

Das zeigt das Potential der SMTP-Engine in postscreen.

Neue DNSBL, DNSWL Syntax

Endlich sind DNS-basierte Whitelists möglich:

- `permit_dnswl_client`
  Client anhand der IP Adresse whitelisten

Neue DNSBL, DNSWL Syntax

Endlich sind DNS-basierte Whitelists möglich:

- `permit_dnswl_client`
  Client anhand der IP Adresse whitelisten
- `permit_rhswl_client`
  Client anhand seines Hostnamen whitelisten

Neue DNSBL, DNSWL Syntax

Endlich sind DNS-basierte Whitelists möglich:

- `permit_dnswl_client`
  Client anhand der IP Adresse whitelisten
- `permit_rhswl_client`
  Client anhand seines Hostnamen whitelisten

Address patterns in `reject_rbl_client`:

Example

```
reject_rbl_client example.com=127.0.0.[2;4;6..8]
```

weist Clients ab, wenn das Lookup Resultat `127.0.0.2`, `127.0.0.40`, `127.0.0.6`, `127.0.0.7` oder `127.0.0.8` ist.
Beispiel:

Example

```
reject_rbl_client list.quorum.to=127.0.0.[2;4;5]
```

Address patterns in `reject_rbl_client`:

Example

```
reject_rbl_client example.com=127.0.0.[2;4;6..8]
```

weist Clients ab, wenn das Lookup Resultat `127.0.0.2`, `127.0.0.40`, `127.0.0.6`, `127.0.0.7` oder `127.0.0.8` ist. Beispiel:

Example

```
reject_rbl_client list.quorum.to=127.0.0.[2;4;5]
```

sqlite Unterstützung

(Lese)-Unterstützung für sqlite:

Example

```
alias_maps = sqlite:/etc/postfix/sqlite-aliases.cf
```

Example

```
# Path to database
dbpath = /some/path/to/sqlite_database

# See sqlite_table(5) for details.
query = SELECT forw_addr FROM mxaliases
   WHERE alias='%s' AND status='paid'
```

Namensauflösung

Der Postfix SMTP Client hängt nicht länger die lokale Domain an einen Hostnamen ohne "." an, wenn der DNS Name aufgelöst werden soll.

Mit `smtp_dns_resolver_options = res_defnames` kriegt man das alte Verhalten zurück.

Namensauflösung

Der Postfix SMTP Client hängt nicht länger die lokale Domain an einen Hostnamen ohne "." an, wenn der DNS Name aufgelöst werden soll.
Mit `smtp_dns_resolver_options = res_defnames` kriegt man das alte Verhalten zurück.

Verbessertes Logging

Bei einem `content_filter` Setup wird nun die QueueID vor dem Filter mitgeloggt:

```
postfix/smtpd[4074]: 6B4A9924782:
  client=localhost[127.0.0.1],
  orig_queue_id=951F692462F
  orig_client=hades.porcupine.org[168.100.189.10]
```

Verbessertes Logging

Bei einem `content_filter` Setup wird nun die QueueID vor dem Filter mitgeloggt:

```
postfix/smtpd[4074]: 6B4A9924782:
  client=localhost[127.0.0.1],
  orig_queue_id=951F692462F
  orig_client=hades.porcupine.org[168.100.189.10]
```

Leider nicht bei `smtp_proxy_filter`!

Verbessertes Logging

Bei einem `content_filter` Setup wird nun die QueueID vor dem Filter mitgeloggt:

```
postfix/smtpd[4074]: 6B4A9924782:
  client=localhost[127.0.0.1],
  orig_queue_id=951F692462F
  orig_client=hades.porcupine.org[168.100.189.10]
```

Leider nicht bei `smtp_proxy_filter`!

Reply Footer

Man kann nun an die REJECT-Meldungen des `smtpd` eine vordefinierte Zeichenkette anhängen:

```
smtpd_reject_footer = Contact postmaster@example.com for technical
 assistance. Please provide the following information in your
 problem report: error message, time ($localtime),
 client ($client_address) and server ($server_name).
 We speak both English and German.
```

Reply Footer

Man kann nun an die REJECT-Meldungen des `smtpd` eine vordefinierte Zeichenkette anhängen:

```
smtpd_reject_footer = Contact postmaster@example.com for technical
 assistance. Please provide the following information in your
 problem report: error message, time ($localtime),
 client ($client_address) and server ($server_name).
 We speak both English and German.
```

Gilt auch für `postscreen` (wegen `postscreen_reject_footer`).

Reply Footer

Man kann nun an die REJECT-Meldungen des `smtpd` eine vordefinierte Zeichenkette anhängen:

```
smtpd_reject_footer = Contact postmaster@example.com for technical
 assistance. Please provide the following information in your
 problem report: error message, time ($localtime),
 client ($client_address) and server ($server_name).
 We speak both English and German.
```

Gilt auch für `postscreen` (wegen `postscreen_reject_footer`).

Kein `undisclosed-recipients` Header mehr

Postfix fügt keinen `To: undisclosed-recipients:;`-Header an, wenn keiner vorhanden ist.

Fragen?

Die Folien gibt es hier:

`http://www.computerbeschimpfung.de/slides/2.8-slides.pdf`