Spass mit SpamAssassin

Themen

- SpamAssassin? Einordnung.
- Was ist Spam?
- Rückblick SpamAssassin, ML in Antispam-Tools
- Naiver Bayes-Klassifikator
- Integration SpamAssassin (Amavis)
- Konfiguration (local.cf)

Themen

- Netzwerkchecks: DCC, pyzor, rayzor, RBLs
- Reporting an externe Dienste
- Regexe: Wie schreibe ich selber SpamAssassin-Regeln
- Einbinden externer Regexe (z.B. Heinlein)
- Tests auf Spam
- Logging Performance-Check
- Ausblick Version 4

Geschichte

Apache SpamAssassin

Aktuelles Logo 2014 von James Thompson entworfen.
Based on a American Indian „dream catcher“, magisches Artefakt

Geschichte

- Veröffentlichung April 20, 2001 auf SourceForge
- Seit Sommer 2004 Teil der Apache Software Foundation

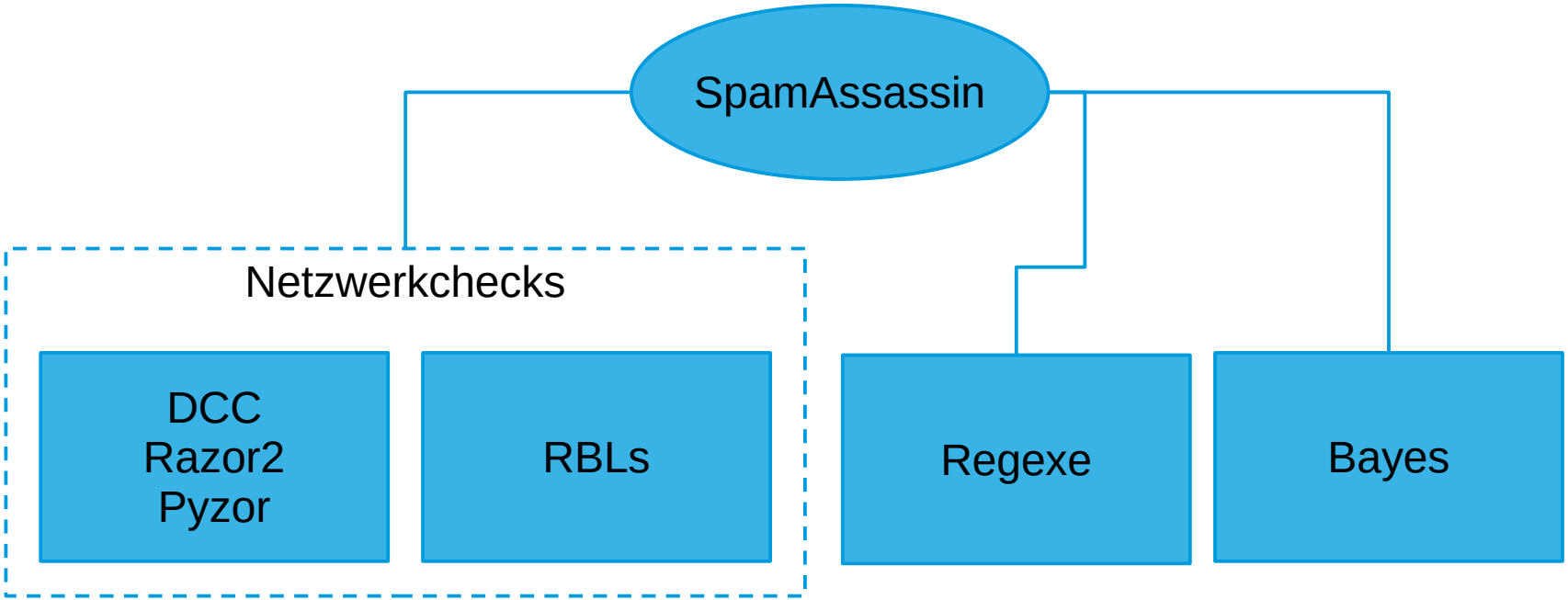
Was ist und kann SpamAssasin

- Ist ein „Framework“ für die Abwehr von Spam mit verschiedenen Techniken:
  - naive Bayes-Filter, Regexe, externe Dienste wie Razor, DCC und RBLs
- (Geschrieben in Perl)
- Zufridenstellende Ergebnisse nach Tuning-Phase
- Es gibt performantere Lösungen mit einem etwas grösseren Funktionsumfang → z.B. Rspamd (Funktionen von SpamAssassin können auch hier genutzt werden).

Regelbasiert

- Ein System verschiedener Regeln, welche einzelnen globalen Symbolen (dem Resultat der Regeln) Punkte zuweisen
- z.B. die 95%ige Wahrscheinlichkeit dass eine Email nach dem Bayes-Klassifikator Spam ist wird in folgendem Symbol beschrieben: BAYES_95
- Diesem wird per default ein Wert von 3 zugeordnet
- Die Werte der Symbole werden addiert
- Ab einem eingestellten Wert (~5) werden die Mails als Spam markiert (verworfen, rejected) (mehrstufige Reaktion möglich)

AntiSpam-Techniken SpamAssassin



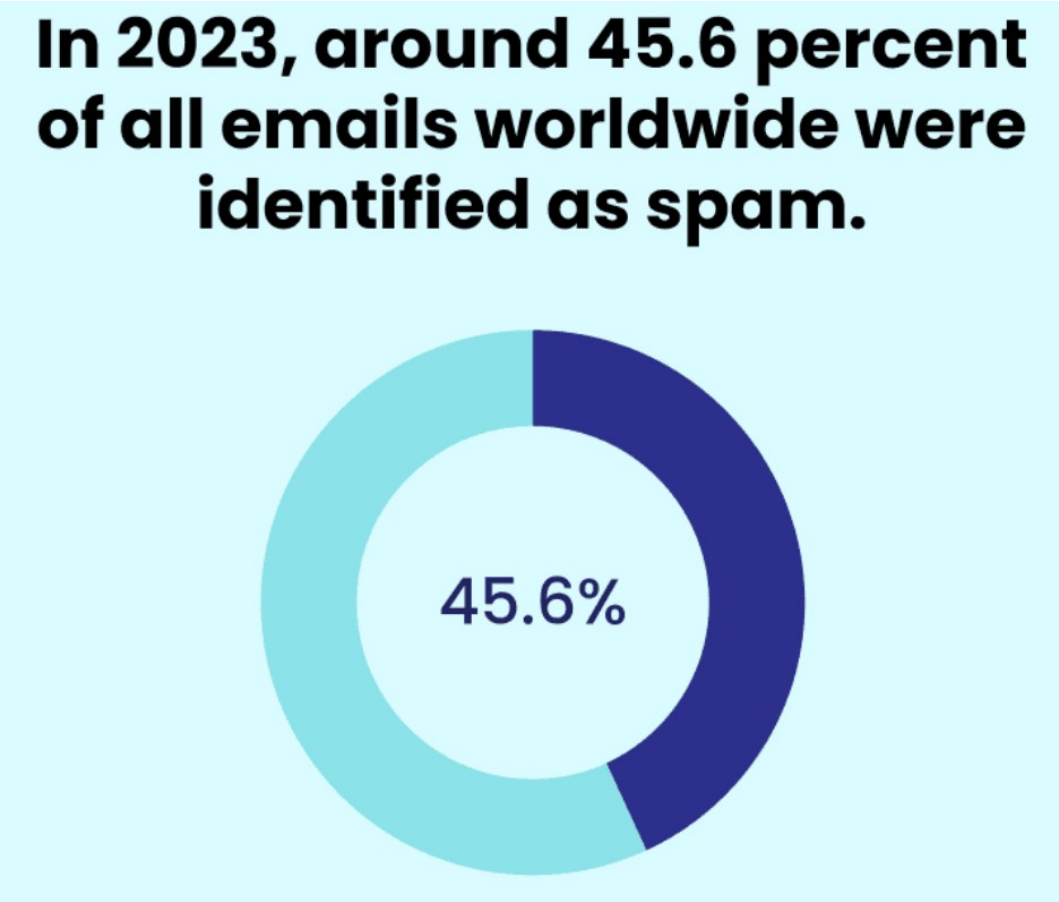
Was ist Spam(Mail)?

Arten von Spam:

- UBE: Unsolicited Bulk E-Mail (Massen-E-Mail)
- UCE: Unsolicited Commercial E-Mail (gezielte unerwünschte Werbung) → kann durchaus einen legitimen Hintergrund haben z.B. Zustimmung auf einer Shopping-Website
- Phishing / Scam
- False Positives / False Negatives

Kann die Bewertung was Spam ist auch eine subjektive Komponente haben?

Aktuelle Statistik weltweiter Spam



Quelle: mailmodo

Frühe Erfolge in der Spamfilterung

Regelbasierte Tools

Maschinelles Lernen in der Spambekämpfung

Statistische Methoden

Exkurs ML für Antispam

- Maschinelles Lernen für Antispam → Bayes
- Wer ist verantwortlich für den Durchbruch von ML bei Antispam?

Thomas Bayes (1701 - 1761)

Satz von Bayes

Bayes-Theorem

Satz von Bayes

$$P(A|B) = (P(B|A) * P(A)) / P(B)$$

Naiver Bayes Text Klassifikator

- Text-Klassifikator:
- Beispiel: Texte über verschiedene Themen sollen einzelnen Themenbereichen wie Sport, Politik, Wirtschaft, Kultur über maschinelles Lernen zugeordnet werden
- Das Dokument (z.B. Mail) wird „zerlegt“ in Tokens, das sind hier Wörter, und die Wahrscheinlichkeit der Wörter berechnet dass es zu einer gewissen Kategorie gehört

Naiver Bayes Text Klassifikator

- Gegeben sind n Kategorien C0 bis Cn-1.
- Um festzustellen welcher Kategorie ein Dokument zugewiesen werden kann muss die Wahrscheinlichkeit berechnet werden dass das Dokument D in Kategorie Ci ist: P(Ci|D), für jede Kategorie Ci.
- Nach der Bayes-Regel kann P(Ci|D) wie folgt berechnet werden: P(Ci|D) = (P(D|Ci) * P(Ci)) / P(D)
- Weil P(D) eine Konstante ist, kann sie ignoriert werden, somit:
- **P(Ci|D) = P(D|Ci) * P(Ci)**
- P(D|Ci) ist die Wahrscheinlichkeit dass für eine gegebene Kategorie Ci die Wörter in D (der E-Mail) zu dieser Kategorie gehören.
- P(Ci) ist die Wahrscheinlichkeit dass ein Dokument einer Kategorie zugehörig ist ohne deren Inhalt zu beachten

P(Ci|D)?

P(Ci|D) = P(Spam|Wort einer Email)

Wie kann P(Spam|Wort einer Email) bestimmt werden?

Paper von Paul Graham

PAUL GRAHAM

A Plan for Spam

Like to build things? Try Hacker News.

August 2002

(This article describes the spam-filtering techniques used in the spamproof web-based mail reader we built to exercise Arc. An improved algorithm is described in Better Bayesian Filtering.)

I think it's possible to stop spam, and that content-based filters are the way to do it. The Achilles heel of the spammers is their message. They can circumvent any other barrier you set up. They have so far, at least. But they have to deliver their message, whatever it is. If we can write software that recognizes their messages, there is no way they can get around that.

Praktische Umsetzung naiver Bayes-Klassifikator

- Nach „a Plan for Spam“ von Paul Graham
- Jeweils ein Korpus von Spam und Ham ~4000 Mails
- Alphanumerische Zeichen wie einige Sonderzeichen (-$´) sind Teil der Wörter
- Stop-Wörter wie und, oder… werden ignoriert
- 2 Hash-Tabellen für die Anzahl der Wörter in dem jeweiligen Korpus
- Eine dritte Hash-Tabelle mit dem Mapping für jedes Wort mit der Wahrscheinlichkeit dass es Spam ist

Wann ist ein Wort Spam?

```
(let ((g (* 2 (or (gethash word good) 0)))
      (b (or (gethash word bad) 0)))
  (unless (< (+ g b) 5)
    (max .01
         (min .99
              (float (/ (min 1 (/ b nbad))
                        (+ (min 1 (/ g ngood))
                           (min 1 (/ b nbad)))))))))
```

Erklärung der Funktion

- word ist das Wort für welches die Wahrscheinlichkeit Spam/Ham ermittelt werden soll
- g und b ist die Häufigkeit des Vorkommens in einer der Hashtabellen
- Verdopplung des Wertes für Ham, um die Wahrscheinlichkeit von False-Positives zu mindern
- Es werden nur Wörter betrachtet die mehr als 5 Mal in beiden Kategorien kombiniert vorkommen
- Wörter die nur in einem Korpus vorkommen 0.01 und 0.99
- nbad und ngood sind die Anzahl an Spam/Ham-Emails
- (Spam / Anzahl Spam Mails) / ((Ham / Anzahl Ham Mails) + (Spam / Anzahl Spam Mails))
- Cap jeweils bei 1

Ermitteln der kombinierten Wahrscheinlichkeit

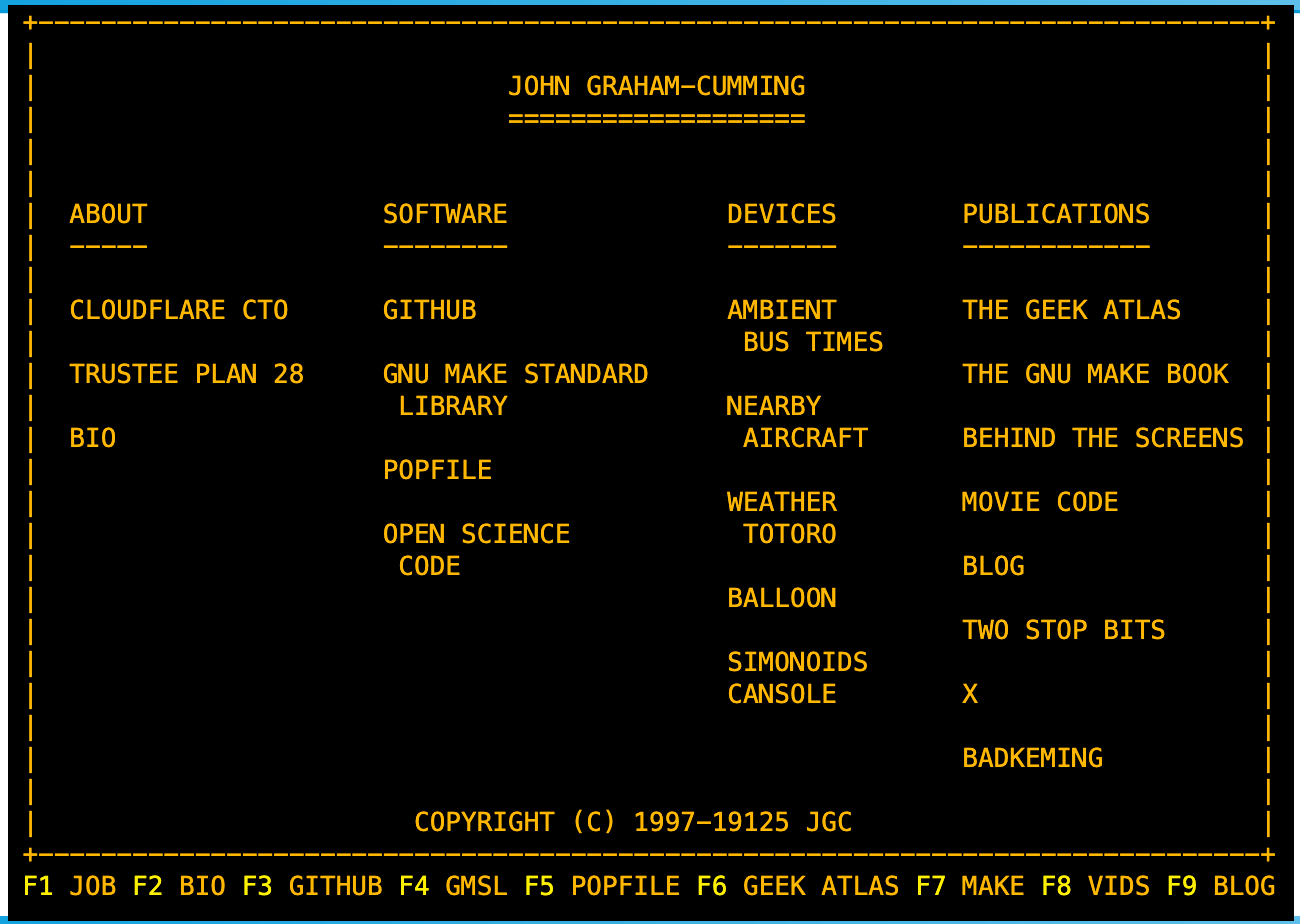
$$\frac{ab}{ab + (1 - a)(1 - b)}$$

a und b jeweils die einzelne Wahrscheinlichkeit eines Wortes

Praktischer Einsatz von Bayes in Antispam-Tools

Wann und wodurch wurde der Einsatz des naiven Bayes-Text-Klassifikators populär?

John Graham-Cunning



Frühe Anti-Spam-Tools / Bayes

- John Graham-Cumming Autor von POPFile. Einsatz des Naiven Bayes-Klassifikators 2002, seit 2020 CTO von Cloudflare
- (Die Applikation ist geschrieben in Perl)
- Initial release 22 September 2002
- Durchbruch für die Nutzung des naiven Bayes-Klassifikators: Vortrag Paul Graham auf der **MIT Spam Conference**, „A plan for Spam“, August 2002
- Better Bayesian Filtering, Paul Graham, Januar 2003

Eröffnung Vortrag von Paul Graham MIT-Spamkonferenz

Nachteile des Bayes-Klassifikators

Database poisoning

Schnelle „Vergiftung“ der statistischen Grundlage

Integration SpamAssassin

Welche Möglichkeiten gibt es SpamAssassin zu integrieren?

Integration SpamAssasin

- Aufruf über Perl-Modul Mail::SpamAssassin
- Perl script spamassasin
- spamd
- SpamAssassin integriert über Perl-Modul in **Amavis**
- Die Integration über Amavis ist die wahrscheinlich am meisten verwendete Form in Kombination mit MTAs
- SpamAssassin lokal vor einem Mail-Client (mutt) als Proxy

Konfiguration SpamAssassin

Auch bei der Verwendung der Integration von SpamAssassin in Amavis bleibt folgende Datei die zentrale Konfigurationsdatei:

- /etc/spamassassin/local.cf

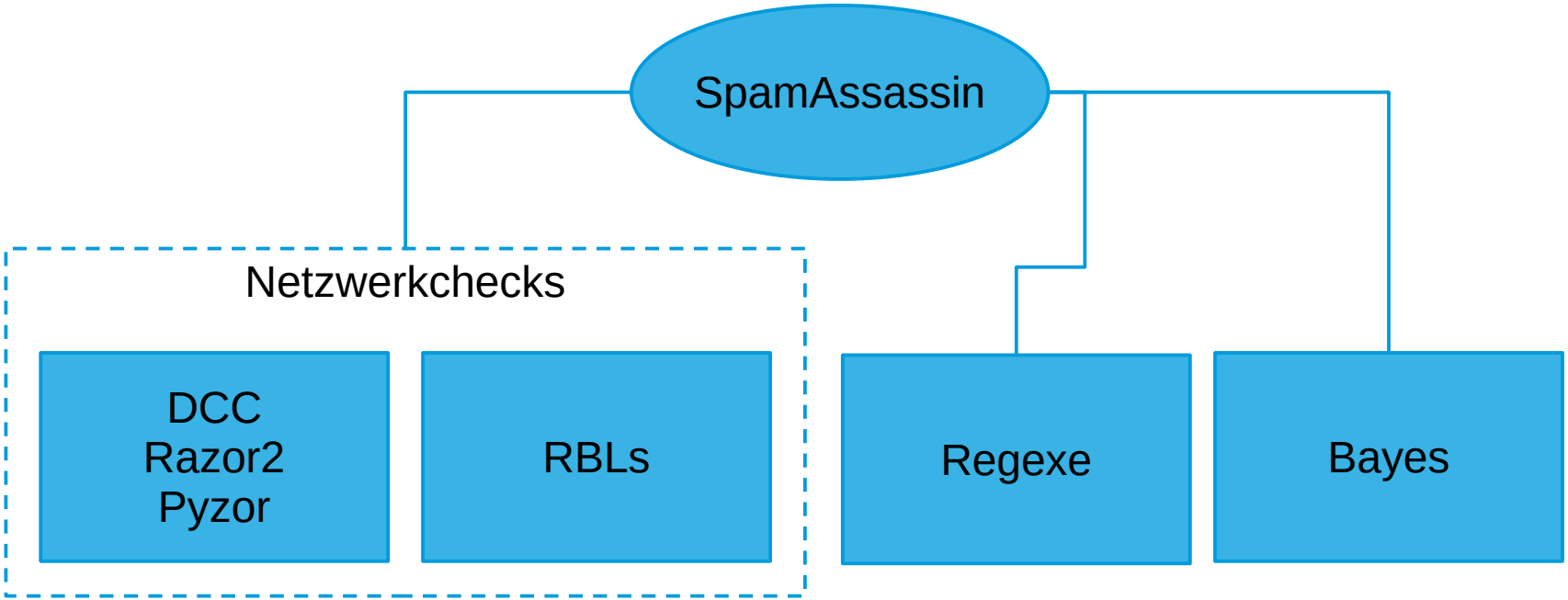
local.cf

```
#   Set the threshold at which a message is considered spam (default: 5.0)
# required_score 5.0
#   Use Bayesian classifier (default: 1)
use_bayes 1
#   Bayesian classifier auto-learning (default: 1)
bayes_auto_learn 1
bayes_store_module    Mail::SpamAssassin::BayesStore::MySQL
bayes_sql_dsn         DBI:mysql:spamassassin:127.0.0.1
bayes_sql_username    spamassassin
bayes_sql_password    xxx
```

local.cf

```
score BAYES_99 4.5
score URIBL_BLACK 2.5
score URIBL_DBL_SPAM 2.5
score FORGED_SPF_HELO 2
whitelist_from contact@fsfe.org
# Pyzor
use_pyzor       1
pyzor_path      /usr/bin/pyzor
...
```

AntiSpam-Techniken SpamAssassin



Netzwerkchecks

Prüfsummenbasierte kollaborative Netzwerkchecks

DCC

- Distributed Checksum Clearinghouse
- Basiert auf fuzzy checksums. Hat den Vorteil das es auch bei leicht geändertem Inhalt einer Mail matched
- DCC basiert angeblich auf einer Idee von unscharfen Inhalts-Filtern, um Spam abzuwehren, entwickelt 1997 von Paul Vixie

Razor2

- Prüfsummen basiertes gemeinschaftliches Anti-Spam-Netzwerk
- Die Reputation der einzelnen Benutzer wird berücksichtigt
- Wird betrieben von der Firma Cloudmark

Pyzor

- Ebenfalls Prüfsummen-basiertes kollaboratives Netzwerk
- Das komplette System released als GPL
- Öffentlicher Server verfügbar: public.pyzor.org:24441

Spam Reporting

spamassassin –report

Report this message as manually-verified spam. This will submit the mail message read from STDIN to various spam-blocker databases. Currently, these are the Distributed Checksum Clearinghouse "http://www.rhyolite.com/anti-spam/dcc/", Pyzor "http://pyzor.sourceforge.net/", Vipul's Razor "http://razor.sourceforge.net/", and SpamCop "http://www.spamcop.net/".

Regex-Regeln

Regeln basierend auf Regexen

Regexe

- Es gibt bereits eine grosse Anzahl von Regeln, basierend auf Regexen
- /usr/share/spamassassin default rules mit SpamAssassin
- /var/lib/spamassassin/<version>/updates_spamassassin_org updates
- Nicht default channel: /var/lib/spamassassin/<version>/<other channel>
- sa-update

Eine SpamAssasin Regel

/var/lib/spamassassin/3.004006/updates_spamassassin_org/20_drugs.cf

jm: keep this case-sensitive, otherwise it FP's

body DRUG_ED_CAPS /\b(?:CIALIS|LEVITRA|VIAGRA)/

describe DRUG_ED_CAPS Mentions an E.D. drug

Aufbau Spamassasin Regel

- Header- oder Body-Rule
- Header kann ein beliebiges Merkmal sein z.B. Subject oder From:
- header __LOCAL_FROM_SHOP From =~ /\.shop/i
- header __LOCAL_ETF_SUBJECT Subject =~ /geld verdienen/i
- Body-Rule bezieht sich auf den Body-Teil der Email
- body __LOCAL_ECOMMERCE_BODY1 /growth analysis/i
- i = ignoriert Gross- und Kleinschreibung
- \b = matched nur innerhalb Wortgrenzen

Eigene Regel bauen

In /etc/spamassassin/local.cf

```
body __LOCAL_VIAGRA1 /\borgasmus\b/i
body __LOCAL_VIAGRA2 /\bmanneskraft\b/i
body __LOCAL_VIAGRA1 /\berrektion\b/i
meta LOCAL_VIAGRA (__LOCAL_VIAGRA1 || __LOCAL_VIAGRA2 || __LOCAL_VIAGRA3)
describe LOCAL_VIAGRA German viagra spam
score LOCAL_VIAGRA 0.4
```

Kombination von privaten mit globalen Symbolen

```
header __LOCAL_FROM_SHOP  From =~ /\.shop/i

meta LOCAL_SHOP_SPAM  (__LOCAL_FROM_SHOP &&
BAYES_99 && (T_REMOTE_IMAGE ||
HTML_FONT_LOW_CONTRAST))

score LOCAL_SHOP_SPAM 2.0
```

Externe Regeln hinzufügen

Über sa-update-cronjob auf Debian:

```
env -i LANG="$LANG" PATH="$PATH" http_proxy="$http_proxy" \
    start-stop-daemon --chuid debian-spamd:debian-spamd --start \
    --exec /usr/bin/sa-update -- \
    --nogpg --channel spamassassin.heinlein-support.de 2>&1
```

Genauso sollte es einen Cronjob für die SA-Default-Rules geben.

Debugging/Benchmarking

Debugging / Benchmarking

- Testen einzelner Emails:
- spamassassin -D timing,async -t <testmail.eml
- spamassassin -D pyzor -t <testmail.eml

Mar 16 16:21:56.216 [69359] dbg: **timing: total 4955 ms** - init: 905 (18.3%), b_tie_ro: 7 (0.1%), parse: 0.61 (0.0%), extract_message_metadata: 23 (0.5%), get_uri_detail_list: 3.2 (0.1%), tests_pri_-2000: 14 (0.3%), compile_gen: 126 (2.6%), compile_eval: 13 (0.3%), tests_pri_-1000: 1.81 (0.0%), tests_pri_-950: 1.22 (0.0%), tests_pri_-900: 1.41 (0.0%), tests_pri_-100: 1.34 (0.0%), tests_pri_-90: 1.01 (0.0%), tests_pri_0: 225 (4.5%), check_spf: 10 (0.2%), poll_dns_idle: 0.02 (0.0%), dkim_load_modules: 13 (0.3%), check_dkim_signature: 1.72 (0.0%), **tests_pri_10: 3353 (67.7%), check_dcc: 3348 (67.6%)**, tests_pri_20: 241 (4.9%), check_razor2: 237 (4.8%), tests_pri_30: 88 (1.8%), check_pyzor: 84 (1.7%), tests_pri_500: 88 (1.8%), tests_pri_2000: 1.52 (0.0%)

Verbose SA in Logs via Amavis

Verbose Logs für SA, Konfiguration für Amavis:

- $log_level = 2;
- $log_templ = $log_verbose_templ;
- Auf Debian in /etc/amavis/conf.d/20-debian_defaults

Beispiel-Log mit Timings

Mar 19 01:10:17 server3 amavis[510628]: (510628-16) Blocked SPAM {BouncedInbound}, [2a01:111:f403:2809::81b]:20480 [2a01:111:f403:2809::81b] ESMTP/ESMTP <kung-noi@hotmail.com> -> <xxx@xxx.com>, (ESMTPS://[2a01:111:f403:2809::81b]:20480), Message-ID: <JH0PR03MB8350DD23A0DDA7725F7C64039DD92@jh0pr03mb8350.apcprd03.prod.outlook.com>, mail_id: fe-waRi3F32Y, b: AsSrDGL6x, **Hits: 6.875**, size: 11910, Subject: "Entdecke, wie du mit Künstlicher Intelligenz Geld verdienen kannst! (raw: =?utf-8?B?RW50ZGVja2UsIHdpZSBkdSBtaXQgS8O8bnN0bGljaGVyIEludGVsbGln?=\t=?utf-8?B?ZW56IEdlbGQgdmVyZGllb)", From: <kung-noi@hotmail.com>, helo=APC01-PSA-obe.outbound.protection.outlook.com, Tests: [BAYES_95=3,DCC_REPUT_70_89=0.1,DKIM_SIGNED=0.1,DKIM_VALID=-0.1,DKIM_VALID_AU=-0.1,FORGED_HOTMAIL_RCVD2=0.874,FORGED_SPF_HELO=2,FREEMAIL_FROM=0.001,HTML_MESSAGE=0.001,MSGID_HDR_MALF=1], autolearn=no autolearn_force=no, autolearnscore=4.189

Log Timings

Mar 19 01:10:17 server3 amavis[510628]: (510628-16) TIMING-SA total 4582 ms - parse: 3.0 (0.1%), extract_message_metadata: 9 (0.2%), get_uri_detail_list: 0.63 (0.0%), tests_pri_-2000: 2.1 (0.0%), tests_pri_-1000: 0.79 (0.0%), tests_pri_-950: 0.47 (0.0%), tests_pri_-900: 0.75 (0.0%), tests_pri_-100: 0.53 (0.0%), tests_pri_-90: 14 (0.3%), check_bayes: 13 (0.3%), b_tie_ro: 1.01 (0.0%), b_tokenize: 3.2 (0.1%), b_tok_get_all: 2.3 (0.1%), b_comp_prob: 1.29 (0.0%), b_tok_touch_all: 3.7 (0.1%), b_finish: 0.54 (0.0%), tests_pri_0: 55 (1.2%), check_spf: 4.1 (0.1%), check_dkim_signature: 2.3 (0.1%), poll_dns_idle: 0.09 (0.0%), tests_pri_10: 3410 (74.4%),check_dcc: 3402 (74.2%), tests_pri_20: 985 (21.5%), check_razor2: 983 (21.5%), tests_pri_30: 88 (1.9%), check_pyzor: 87 (1.9%), tests_pri_500: 2.5 (0.1%), tests_pri_2000: 0.48 (0.0%), get_report: 0.30 (0.0%)

Erklärung der Regeln

DCC_REPUT_00_12 DCC reputation between 0 and 12 % (mostly ham)

DCC_REPUT_13_19 DCC reputation between 13 and 19 %

DCC_REPUT_70_89 DCC reputation between 70 and 89 %

DCC_REPUT_90_94 DCC reputation between 90 and 94 %

DCC_REPUT_95_98 DCC reputation between 95 and 98 % (mostly spam)

DCC_REPUT_99_100 DCC reputation between 99 % or higher (spam)

FORGED_HOTMAIL_RCVD2 hotmail.com 'From' address, but no 'Received:'

MSGID_HDR_MALF Has invalid message ID header

SpamAssassin 4.x

- Ausblick SpamAssassin 4.x
- Mehr asynchrone Checks (schneller)
- DMARC-Überprüfung

Ende

Das wars

Viel Spass mit SpamAssassin!