

# SpamAssassin

## Was ist Spam?

Als Spam ['spɛm] werden unerwünschte, in der Regel auf elektronischem Weg übertragene Nachrichten bezeichnet, die dem Empfänger unverlangt und unerwünscht zugestellt werden und massenhaft versandt wurden oder werbenden Inhalt haben. Dieser Vorgang wird Spamming oder Spammen genannt, der Täter Spammer. [Wikipedia]

# SpamAssassin – Begriff

- SPAM - spiced pork and ham; Monty Python wiederholt Spam in wenigen Minuten 120 mal
- Assassinen – vom hebr. haschīsch Kräuter, Gräser, Hanf; sagenumwobene militante ismailitische Sekte des Mittelalters im Orient, die mit Gift und Dolch vor allem polit. Morde beging

# Quellen des Spams

- Schlecht konfigurierte Mailserver
- Wegwerfemailadressen á la web.de und gmx
- Offene Webformulare
- Bot-Netze (gekaperte, meist Windows-PCs)
- Spamdienste aus Ländern ohne juristische Verfolgung

# SpamAssassin entlasten

- Emailadresse für öffentliche Weitergabe bei web.de, gmx etc.
- wichtige Emailadressen nur an vertrauenswürdige Empfänger weitergeben
- Catch-All-Postfächer vermeiden

# SpamAssassin – Aufbau

- Download  
<http://spamassassin.apache.org/>
- SpamAssassin ist modular
- Quelloffen in Perl programmiert
- Daemonversion vorhanden
- Client in C
- Konfiguration in SQL speicherbar

# So arbeitet Spamassassin (1)

- Nachrichten-Header werden auf RFC geprüft
- Header und Body werden auf „Spamphrasen“ überprüft (Viagra, MAKE MONEY FAST ...)
- Nachrichten-Hashes koennen in Online-Datenbanken geprüft werden
- IP-Blockinglisten werden geprüft

# So arbeitet Spamassassin (2)

- Führen von Black- und Whitelists für Ips, Domains und Emailadressen
- Trainierbares System. User entscheidet, was Spam bzw. Ham ist
- Protokoll „Sender Policy Framework“ - SPF – zum Abgleich zwischen IP und Domain (<http://spf.pobox.com>)
- Privilegierte Absender mit Hashcash (<http://www.hashcash.org>)

# Wo arbeitet SpamAssassin

- SA kann mit jedem MTA arbeiten
- SA kann auf File- und Directory-Basis arbeiten (Mailbox vs. Maildir)
- Scannen mit procmail
- SpamAssassin ist in Perl und ist OS-unabhängig

# SpamAssassin – die Ausgabe

```
X-Virus-Scanned: by amavisd-new-20030616-p10 (Debian) at sas1
X-Spam-Status: No, hits=4.1 tagged_above=0.5 required=5.0
  tests=BAYES_60, FUZZY_BILLION, SARE_OBFU_MILLIONS
X-Spam-Level:****
```

# SpamAssassin informiert (1)

Unsolicited bulk email from unknown or forged sender.  
Subject: SPAM VON <bartae@m>

According to the 'Received:' trace, the message originated at:  
exim

The message WILL NOT BE delivered to:  
<holderspam@spengler-inter.net>:  
250 2.7.1 Ok, discarded, UBE, id=19932-09

The message has been quarantined as:  
/var/lib/amavis/virusmails/spam-de1070dfb38802dd1f6d40db4fb05fb3-20060602-203805-19932-09.gz

SpamAssassin report:  
Software zur Erkennung von "Spam" auf dem Rechner

sas1.spengler-inter.net

hat die eingegangene E-mail als mögliche "Spam"-Nachricht identifiziert.  
Die ursprüngliche Nachricht wurde an diesen Bericht angehängt, so dass  
Sie sie anschauen können (falls es doch eine legitime E-Mail ist) oder

ähnliche unerwünschte Nachrichten in Zukunft markieren können.  
Bei Fragen zu diesem Vorgang wenden Sie sich bitte an

the administrator of that system

# Spamassassin informiert (2)

Vorschau: Unerwünschte Massen-Mail Von: bartae@m Betreff(Subject):  
Re: 930 AMbBBtEN Der 'Received:' Spur zufolge, stammt diese Nachricht  
von: [222.255.180.171] (helo=m) Die Nachricht WIRD NICHT GESENDET an:  
<baumer@holder-tv.de>: 250 2.7.1 Ok, discarded, UBE, id=09948-02 [...]

Inhaltsanalyse im Detail: (10.5 Punkte, 5.0 benötigt)

Pkte	Regelname	Beschreibung
0.0	UNPARSEABLE_RELAY	Informational: message has unparseable relay lines
0.0	BAYES_50	BODY: Spamwahrscheinlichkeit nach Bayes-Test: 40-60% [score: 0.5001]
1.6	URIBL_SBL	Enthält URL in SBL-Liste ( <a href="http://www.spamhaus.org/sbl/">http://www.spamhaus.org/sbl/</a> ) [URIs: untanikasuon.com]
4.1	URIBL_JP_SURBL	Contains an URL listed in the JP SURBL blocklist [URIs: untanikasuon.com]
2.1	URIBL_WS_SURBL	Enthält URL in WS-Liste ( <a href="http://www.surbl.org">www.surbl.org</a> ) [URIs: untanikasuon.com]
4.5	URIBL_SC_SURBL	Enthält URL in SC-Liste ( <a href="http://www.surbl.org">www.surbl.org</a> ) [URIs: untanikasuon.com]
-1.9	AWL	AWL: From: address is in the auto white-list

# Konfiguration SpamAssassin

- Konfigurationsdatei  
*/etc/spamassassin/local.cf*
- *required\_hits n* – Punktestand für Spam
- *report\_safe 0|1* – Spambericht in Nachricht
- *rewrite\_header subject \*\*\*SPAM\*\*\** –  
Betreff in Spam ändern
- *skip\_rbl\_checks 0|1* - RBL-Prüfung

# SpamAssassin testen

- *spamassassin -test-mode < sample-nospam.txt*
- Testmodus produziert immer Punktestand

# SpamAssassin liefert ...

- Ham (Echte Negative) – SA stimmt mit dem Endanwender überein
- Spam (Echte Positive) – SA und Endanwender bewerten Nachricht als Spam
- Falsche Positive – SA markiert Nachricht als Spam, die der Endanwender empfangen möchte
- Falsche Negative - Spam wird als Ham ausgeliefert

# SpamAssassin-Regeln

- Regel-Name aus bis zu 22 Grossbuchstaben. Beginn mit T\_ Test-Regeln
- Beschreibung der Regel bis zu 50 Buchstaben
- Getestet wird in Header, Body, URI ...
- Was soll gesucht werden (Regexp, Blacklist, SA-Funktion)
- Testflags
- Punktezahl für den Test

# SpamAssassin Regel (1)

*header FROM\_STARTS\_WITH\_NUMS From=~/^|d|d/*

*describe FROM\_STARTS\_WITH\_NUMS From: starts with nums*

*score FROM\_STARTS\_WITH\_NUMS 0.390 1.574 1.0444 0.579*

Punktvergabe:

0.39 ohne Netzwerk- und Bayestests

1.574 mit Netzwerk- aber ohne Bayestests

1.0444 ohne Netzwerk- aber mit Bayestests

0.579 mit Netzwerk- und Bayestests

# SpamAssassin Regel (2)

Prüfung auf [friend@public.com](mailto:friend@public.com) in To, From oder Cc

```
header FRIEND_PUBLIC ALL = ~ /^(?:to|cc|  
from):.*friend\@public\.com/im
```

Prüfung auf X-PMFLAGS

```
header X_PMFLAGS_PRESENT exists:X-PMFLAGS
```

Prüfung auf lange Zeilen HEX-Codes im Body

```
body LARGE_HEX /[0-9a-fA-F]{70,}/
```

Subject nur Grossbuchstaben – SA-Funktion

```
header SUB_ALL_CAPS eval:subject_is_all_caps()
```

Prüfung auf JavaScript Popup

```
body HTML_WIN_OPEN eval:html_test('window_open')
```

**Vorsicht: schlechter Regexp kann System komplett auslasten**

# SpamAssassin - erprobte Regeln

- Verschiedene Communities stellen Regeln bereit
- <http://www.rulesemporium.com/rules.htm>
- <http://www.exit0.us/index.php?pagename=RulesDuJour>

# SpamAssassin – Schwarz, weiss, grau

- Whitelist und Blacklist sind grundverschieden zu den Black- und Whitelists der MTA
- SpamAssassin verändert mit White- und Blacklist den Punktestand von Nachrichten um bis zu 100 Punkten
- MTAs lassen keine Mails aus Blacklists durch, aus Whitelists alle ohne Spamprüfung
- *whitelist\_from [mail@foo.bar](#), \*@foo.bar*
- *whitelist\_from foo.bar, \*.foo.bar*

# SpamAssassin – Schwarz, weiss, grau

- *whitelist\_from\_rcvd* – bessere Kontrolle wegen Beziehung Absender und Relay
- *whitelist\_to* – Spampunkte -6
- *more\_spam\_to* – Spampunkte -20
- *all\_spam\_to* – Spampunkte -100
- Syntax Blackliste analog
- *blacklist\_from* *foo@bar*
- Greylisting sehr wirkungsvoll gegen Spambot-Netzwerke; aber Aufgabe des MTA

# SpamAssassin - Autowhitelist

- AWL erlernt für alle Absender History für Spam und Ham
- Spam-Punktestand für jeden Absender in Datenbank addiert
- *auto\_whitelist\_factor 0..1 0*: historischer Durchschnittswert ignoriert, 1 nur historischer Durchschnittswert relevant; Zwischenwerte gewichten Durchschnittswert entspr.

# SpamAssassin – Bayessche Filterung

- Grundlage Theorem zur bedingten Wahrscheinlichkeit von Thomas Bayes von 1763
- Voraussetzung Sammlung von Mails, die definitiv Spam bzw. Ham sind.

# SpamAssassin - Bayes-Filter (1)

- *use\_bayes 0|1* - Entscheidung, ob Bayes-Filter benutzt werden
- *bayes\_auto\_learn*,  
*bayes\_auto\_learn\_threshold\_nonspam* - Nachrichten mit sehr geringem oder sehr hohem Spampunkten werden dem Klassifizierungssystem automatisch zugeführt
- *bayes\_ignore\_header Header-Name* - bestimmte Header beim Klassifizieren ignorieren

# SpamAssassin – Bayes-Filter (2)

- *bayes\_ignore\_from* Adresse – verhindert Klassifizierung einer Absenderadresse; besondere Form der Whiteliste
- *bayes\_ignore\_to* Adresse – wichtig z.B. für postmaster, der weitergeleiteten Spam erhält
- *bayes\_learn\_during\_report* – während des Sendens von Spam an Clearing-Stellen lernt das Bayessche System

# SpamAssassin – Bayes-Filter (3)

- *bayes\_min\_[ham|spam]\_num* – minimale Anzahl an erlernten Nachrichten, bevor SA neue Nachrichten nach Bayes klassifiziert
- *bayes\_use\_hapaxes* 0|1– (hapax: griech. einmal) Token, die nur einmal angetroffen werden; Token könnten unzuverlässig sein. Standardmässig auf 1 wegen grösserer Genauigkeit

# SpamAssassin – Bayes-Filter (4)

- *bayes\_use\_chi2\_combinig* – Anwendung der math. Funktion für die Token-Wahrscheinlichkeit Chi-Quadrat-Verteilung oder „naiv Bayessche“
- *bayes\_auto\_expire* und *bayes\_learn\_to\_journal* - Performancetuning

# SpamAssassin – Bayes Training (1)

- Alles trainieren – sehr sensibel auf Veränderungen im Spammuster. Aber könnte zu schnell reagieren. Sehr ressourcenintensiv
- Trainieren auf Basis von Fehlern – nur falsche Positive bzw. falsche Negative zum Training. Sehr ressourcenschonend, aber möglicherweise zu langsam.

# SpamAssassin – Bayes Training (2)

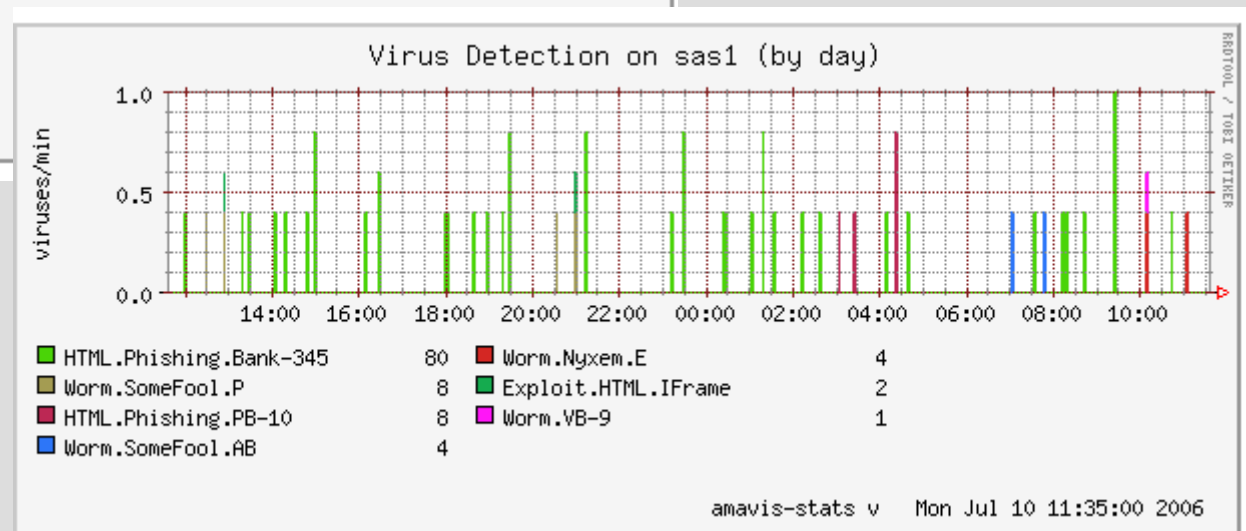
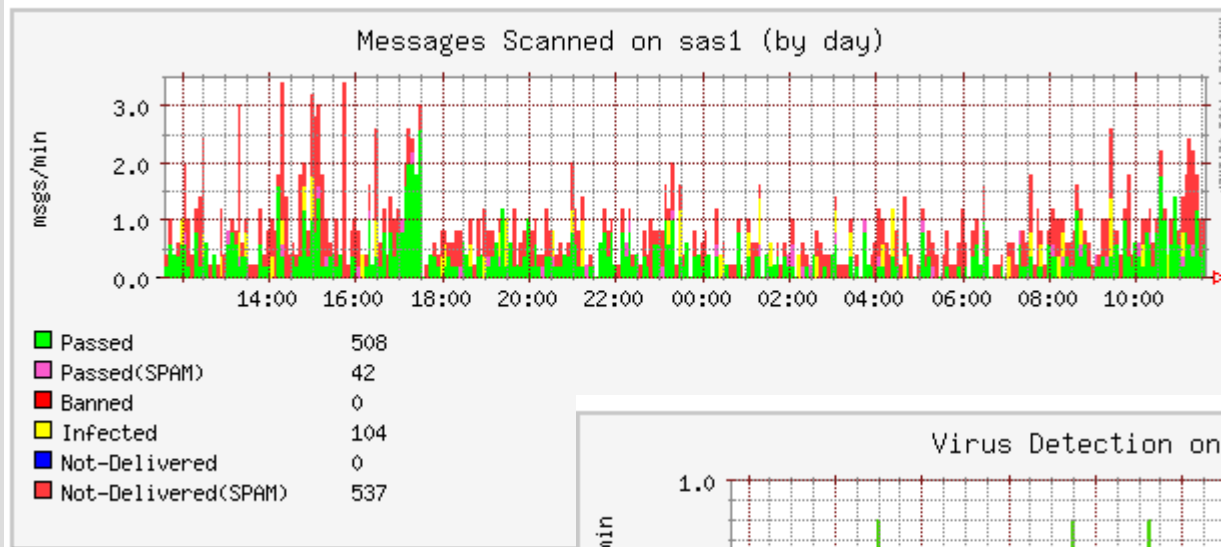
- Empfehlung von Greg Louis nach Experimenten (<http://www.bgl.nu/bogofilter/>): 10.000 Spam- und 10.000 Ham-Nachrichten Alles-Trainieren. Dann umschalten auf Fehler-Training
- *sa-learn* Schnittstelle zur Bayesschen Filterung; Optionen *--spam*, *--ham*
- *sa-learn -dump | sort -n -* Informationen, was Bayes „gelernt hat“

# SpamAssassin – effektiv mit Amavis-New

- Einbindung von SA in MTA sehr bequem über Amavis-New möglich.
- Amavis-New koordiniert neben Spamttests auch Virentests, Black- und Whitelists, externe Checks wie DCC (Distributed Checksum Clearinghouse) und Razor
- Amavis-New übernimmt Logging
- Amavis-New generiert Statistiken
- Amavis-New entscheidet, was mit klassifizierten Nachrichten passiert

# Amavis-New - Statistik

## Daily Graphs



# SpamAssassin – die Zukunft

- Spammer sind sehr erfinderisch und Spam ist sehr viel vielfältiger als Viren
- SA will in Zukunft genetische Algorithmen einführen
- Der Enduser bzw. der Admin sind immer einen Schritt hinten.

# Verwertung und Weitergabe

Diese Skript darf unentgeltlich ohne  
Einschränkung weitergegeben werden.  
Änderungen bitte dem Author melden.

Anton Spengler  
Softwareentwicklung Spengler  
anton@spengler-inter.net