Mail-Server: Funktionsweise von SpamAssassin

Einleitung:

Der Aufbau jedes Mail-Systems ist bereits kompliziert. Sobald noch externe Module dazu kommen wird es manchmal unübersichtlich. Daher hier mal eine Übersicht über die Funktionen und Funktionsweisen von SpamAssassin

SpamAssassin

SpamAssassin ist überwiegend in der Script-Sprache Perl programmiert. Daher stößt man auch häufiger in dem Zusammenhang auf das Perl-Archiv CPAN.

Ein weiterer Nachteil, ist, daß das Kernstück bei jedem Aufruf neu kompiliert werden muß. Daher ist die Verwendung von `spamd` bei massenweisem Scannen von Mails.

Es basiert grundsätzlich auf zwei verschiedene Systeme:

a) Regelbasierte Filterung

b) Ein Bayes-Filter

Zusätzlich gibt es noch eine Plugin-Schnittstelle um weitere Filter zu integrieren.

Konfiguration und Umgang

Die Konfigurationsdateien finden sich meist im eigenen Verzeichnis `/etc/spamassassin/`. Bei älteren Systemen evtl. auch in `/etc/mail/spamassassin/`.

Zusätzlich finden sich evtl. auch noch weiter Konfigurationen und eigene Bayes-Datenbanken im Home-Verzeichnis des jeweiligen Users: `$HOME/.spamassassin/`.

Das *Regelwerk* findet sich häufig im Verzeichnis `/usr/share/spamassassin/`. Die interessantesten sind hier die die Dateien `30_text_de.cf` und `50_scores.cf`.

Wer gerne eigene Regeln einfügt sollte dies am Besten in eine eigenen Datei schreiben und diese in die Hauptkonfiguration inkludieren.

Es gibt 2 Methoden eine Email von SpamAssassin untersuchen zu lassen:

- Mit dem Befehl `spamassassin`.

  Damit wird SpamAssassin direkt aufgerufen. Sinnvoll, wenn man unregelmäßig kleinere Mails verarbeiten will oder einfach nur um die Konfiguration zu testen.

- Mit dem Befehl `spamc`.

  Dies ist der Client zum Daemon `spamd`. Da der Daemon ständig im vorkompiliert im Speicher liegt, ist dies für größere Anzahl von Scan-Vorgängen zu bevorzugen.

Punkte-System

SpamAssassin arbeitet mit einem Punkte-System. D.h. jeder Filter setzt ein Flag (z.B. `HTML_MESSAGE`. An einem beliebigen anderen Punkt im Regelwerk stehen dann die Punkte zu diesem Flag.

Am Ende werden alle Punkte zu den gesetzten Flags zusammen gerechnet und die Summe ergibt dann den *Spam-Score*.

Ab eine Email dann als Spam klassifiziert wird hängt vom Wert für `Require` ab. Denn nur wenn der Score über dem Require-Score liegt wird das Spam-Flag gesetzt.

Bayes-Filter

Erstmal ein paar Definitionen:

`Token`	Etwa vergleichsweise ein "Wort".
`Bayes-DB`	Die Token-Datenbank mit Wichtung.
`sa-learn`	Programm zum verwaltet und füllen der Bayes-DB.
`Bayes-Filter`	Klassifiziert einen Text anhand der auftretenden Tokens/Wörter in Prozent.

Die Eingliederung des Bayes-Filter kann man sich wie ein Plugin vorstellen. Letztendlich werden nur Flags gesetzt: `BAYES_00`, `BAYES_05`, `BAYES_20`, `BAYES_40`, `BAYES_50`, `BAYES_60`, `BAYES_80`, `BAYES_95`, `BAYES_99`

Mail-Server: Funktionsweise von SpamAssassin

Welche Punktewerte dahinter stecken steht wiederum im Regelwerk von SpamAssassin.

Eindeutige ID: #1248
huschi
2007-08-09 19:38