

Mail::SPF::Iterator

Was ist SPF

- Sender Permitted Framework
 - RFC 4408 (SPF),
 - RFC 4406 (SenderID, fast das gleiche wie SPF)
- Idee: Eigentümer einer Domain gibt an, welcher Host Mails mit dem Absender der Domain senden darf
- wird u.a verwendet von GMX, sipgate, Google (gmail, blogger.com,...)
- anderes Verfahren: DKIM - Mailhost "signiert" Mail als Zeichen, das es durch einen offiziellen Mailhost versandt wurde

Details

- die IP des erlaubten Senders kann spezifiziert werden über
 - IP-Adressen + Netzmaske
 - Referenzen zu MX
 - Referenzen zu PTR Records
 - RBL-artige Anfragen
 - garniert mit Includes und Redirects zu anderen SPF Records
- als Ergebnis gibt es dann
 - Pass: Sender ist definitiv erlaubt
 - Fail: Sender ist definitiv nicht erlaubt
 - irgendwas dazwischen (SoftFail, Neutral, None, PermError, TempError)
- Fail wird nur selten geliefert, d.h. meist wird der Status nur als Zusatzinfo für den Spamcheck genutzt

Implementationen

- siehe www.openspf.org
- für alle wichtigen Sprachen (perl, python, java, .NET, C)
- gute Testsuite
- bisher in Perl: Mail::SPF

Warum noch eine Implementation?

- haben doch schon Mail::SPF
- aber Mail::SPF macht alle DNS Anfragen blocking
- und SPF kann sehr DNS intensiv sein
- d.h. skaliert nicht wenn viele DNS Anfragen gemacht werden, hier müsste man mit vielen Threads oder Prozessen arbeiten
- nicht brauchbar für Event-driven Programmierung, d.h. z.B. POE, Event...
- damit nicht brauchbar für Nutzung auf der GeNUGate Firewall

Implementationsdetails

- iteratives Verfahren zum SPF Lookup
- man erstellt für jede SPF Anfrage ein Iterator-Objekt...
- ...und bekommt die initialen DNS Anfragen
- füttert mit der Antwort auf die DNS Anfragen das Iterator-Objekt...
- ...und bekommt die nächsten Anfragen oder die finale Antwort
- man kann mehrere Iteratorobjekte gleichzeitig benutzen

Codebeispiel

```
my $spf = Mail::SPF::Iterator->new(
    $ip,          # SMTP Client IP
    $mailfrom,   # MAIL FROM: mailfrom
    $helo,       # HELO|EHLO helo
    $myname      # lokaler Name des Servers
);
my ($result,@ans) = $spf->next;
while ( ! $result ) {
    my $answer = ...dns lookup @ans ...
    ($result,@ans) = $spf->next($answer)
}
```

Erfahrungen

- passt SPF Testsuite von openspf.org
- passte GeNUA internen Code Review :)
- genutzt in GeNUA eigenem Produkt

Danke!