Fetchmail SSL Fingerprint

von Geiststreicher am 30. April 2014

in HowTo, Linux, Tipps & Tricks

Was lange währt, wird endlich gut und wurde als DAS neue Sicherheitsfeature von GMX und web.de verkauft. Die SSL-Verschlüsselung beim Abruf von Mails per POP3. Ruft man seine Mails auf einem Linux-Rechner per fetchmail ab, muss man unter Umständen ein wenig an seiner Konfiguration drehen. Wie es funktioniert, möchte ich hier kurz erklären.

Wie ich in meiner Artikelserie „Bye, bye, cloud“ schon angedeutet habe, rufe ich meine verschiedenen Postfächer automatisch per fetchmail ab und führe sie auf meinem Server zusammen. Die Einzelheiten dazu findet ihr demnächst im „Bye, bye, cloud“-Artikel, der sich speziell mit meiner Mailkonfiguration befasst.

Hier soll es nur kurz um die Konfiguration von fetchmail im Speziellen gehen und darum, wie man das aktuelle Zertifikat und den Fingerprint für die SSL-Verschlüsselung bekommt und nutzt. Alles, was ich im folgenden exemplarisch für GMX sage, gilt 1:1 auch für web.de und lässt sich leicht auf andere Email-Dienste übertragen.

Als zusätzliches Beispiel gibt es noch die Befehlszeile für die Ermittlung des Fingerprints über einen IMAP-Abruf mit STARTTLS.

Fetchmail

Der Eintrag in meiner fetchmailrc sieht folgendermaßen aus:

```
poll pop.gmx.net with service pop3s interval 6
    user '1234567' there with password 'MyVeryS3cr3tP@assword' is sascha here options fetchall
    ssl
    sslproto tls1
    sslfingerprint "8A:B7:78:CF:0D:73:4E:EE:FF:EB:B8:C0:90:7D:46:56"
    sslcertck
    sslcertpath /etc/ssl/certs
```

Wie man sieht rufe ich meinen GMX-Account per pop3s ab. Der Service ist in /etc/services definiert und bestimmt den Port, mit dem auf GMX zugegangen wird. Im Falle eines SSL-gesicherten Abrufs ist das fast immer Port 995.

Anschließend sieht man die Anmeldung am GMX-Server und welcher Benutzer auf meinem Server das Empfängerkonto ist.

Der interessante Teil bezüglich der Sicherheit spielt sich in den folgenden Zeilen ab. Zuerste wird prinzipiell der Abruf per SSL „eingeschaltet“, aschließend die Protokollversion festgelegt. Dann wird sowohl der Server-Fingerprint überprüft, wie auch das komplette Zertifikat, dass man vorher im /etc/ssl/certs/ Directory abgelegt haben muss. Schlägt eine der Prüfungen fehl, findet keine weitere Datenübertragung statt.

In einer Mehrbenutzerumgebung würde man das Passwort natürlich nicht in die fetchmailrc schreiben, sondern ggf. in einer Userspezifischen .netrc.

Zertifikat abrufen

Das Serverzertifikat wird übermittelt, wenn man eine SSL-Verbindung mit GMX aufbaut. Das kann man auf der Console mit folgendem Befehlt tun:

```
openssl s_client -connect pop.gmx.net:995 -showcerts
```

In der Ausgabe sieht man alle Zertifikate der Trust-Chain. An den Bezeichnungen erkennt man, dass das erste Zertifikat auf die „1&1 Mail und Media GmbH“ ausgestellt ist. Dieses kopiert man einschließlich der Zeilen `-----BEGIN CERTIFICATE-----` und `-----END CERTIFICATE-----` und speichert es in einer Datei im Verzeichnis /etc/ssl/certs/

Anschließend muss man in dem Directory noch ein `c_rehash` ausführen!

Fingerprint ermitteln

Der Fingerprint leitet sich vom gespeicherten Zertifikat ab und kann ebenfalls über einen openssl-Befehl ermittelt werden:

```
openssl x509 -dates -fingerprint -md5 -noout -in /etc/ssl/certs/gmx_de.pem
```

Dabei gehe ich davon aus, dass das im vorherigen Abschnitt ermittelte GMX-Zertifikat in /etc/ssl/certs/gmx_de.pem gespeichert ist. Der Befehlt gibt zusätzlich zum Fingerprint auch die Gültigkeit mit aus.

Hat man das Zertifikat noch nicht lokal vorliegen, kann man den Fingerprint auch mit einer Befehlszeile ausgeben lassen, die den Connect und die Formatierung der Ausgabe komplett übernimmt.

In diesem Beispiel erfolgt der Abruf über IMAP von meinem Provider Uberspace mit STARTTLS:

```
openssl s_client -connect vulpecula.uberspace.de:143 -starttls imap </dev/null  2>/dev/null   | sed -n
/BEGIN/,/END/p  | openssl x509 -dates -fingerprint -md5  -noout
```

Revision #2
Created 2 October 2023 08:28:19 by Gerald Amrhein
Updated 2 October 2023 08:29:45 by Gerald Amrhein