SSL für lau

Kostenlose Zertifikate einrichten

Daniel Bachfeld - *14.01.2010*

Der eigene Webserver mit einem SSL-Zertifikat, das von einer zugelassenen Certificate Authority ausgestellt wurde und somit im Browser auch keine Warnmeldung verursacht, das böte schon seine Vorteile. Doch allein die Preise für ein Zertifikat von Verisign&Co lassen solche Gedankenspiele meist schnell wieder enden. Der israelische Anbieter StartSSL bietet jedoch kostenlose SSL-Serverzertifikate an, die immerhin ein Jahr gültig sind.

Da die Wurzelzertifikate von StartSSL bereits in gängigen Browsern enthalten sind, verursacht der Aufruf eines Zertifikats von diesem Anbieter keine Fehlermeldung. Dieser Artikel beschreibt beispielhaft alle Schritte, die von der Anmeldung bei StartSSL bis zur Integration des Zertifikats in einen Apache-Webserver unter Linux nötig sind.

Um ein Zertifikat für eine Domain zu erhalten, muss man lediglich Mails an einen adminstrativen Account empfangen können (postmaster, hostmaster oder webmaster). Eine weitergehende Prüfung, ob man der Eigentümer der Domain ist, findet nicht statt. (Das handhaben im Übrigen auch andere CAs wie RapidSSL ähnlich). Zudem ist ein ausgestelltes Zertifikat nur dann nützlich, wenn man es in seinem Server auch installieren kann. Inhabern einer "Visitenkarten"-Homepage bei einem Webhoster nützt solch ein Zertifikat also nichts, da sie keinen Zugriff auf die Serverkonfiguration haben

Die Vorgehensweise für Apache unter Windows oder beim Internet Information Server unter Windows ist ähnlich. Beim IIS führt man alle erforderlichen Schritte in der Microsoft Management Console (MMC) aus.

Auf die Plätze!

Grundsätzlich ist der Ablauf bei der Ausstellung eines Zertifikats immer der Gleiche. Zunächst generiert man sich ein Schlüsselpaar (einen öffentlichen und einen privaten Schlüssel). Unter Linux und BSD-Derivaten erledigt man das beispielsweise mit

```
openssl genrsa -out example.com.key 2048
```

Dieser Befehl erzeugt die Schlüssel mit jeweils 2048 Bit Länge und speichert sie zusammen in der Datei *example.com.key*. Der Zusatz `-des3` sichert den Schlüssel mit einem Passwort, allerdings muss man dieses später beim Start des Webservers angeben. Der Befehl

```
openssl req -new -key example.com.key -out example.com.csr
```

liest den öffentlichen Schlüssel aus der Key-Datei und erstellt damit einen sogenannten Certificate Signing Request (CSR). Der CSR enthält üblicherweise Angaben zum Inhaber des Schlüssels wie Organisation, Land, Ort, E-Mail-Adresse und die Adresse des Servers (Common Name, CN) für die das Zertifikat ausgestellt werden soll, also www.*example.com*. Daneben enthält der CSR den öffentlichen Schlüssel. Den CSR schickt man zur Certificate Authority, die ein Zertifikat erstellt. Das Zertifikat wiederum enthält den öffentlichen Schlüssel, Angaben zum Inhaber und zum Aussteller sowie die digitale Signatur des Ausstellers über alle enthaltenen Informationen und den Schlüssel. Damit ist der Schlüssel an eine Identität gebunden. Das fertige Zertifikat speichert man auf seinem Server, der es auf Anfrage eines Clients ausliefert. So weit die Theorie.

Fertig, los!

In der Praxis stellt sich die Vorgehensweise dann etwas anders dar, als in der Theorie beschrieben. StartSSL ignoriert beispielsweise sämtliche Angaben zum Antragsteller im CSR und verarbeitet nur den eingebetteten öffentlichen Schlüssel. Zudem muss der eigene Webserver nicht nur das von StartSSL ausgestellte Zertifikat ausliefern, sondern

zusätzlich noch ein sogenanntes Intermediate-Zertifikat von StartSSL, um Fehlermeldungen im Browser von Besuchern zu verhindern – doch dazu später mehr.

Startpunkt für den Online-Zertifikatsantrag ist die "Express Lane" des **Certificate Control Panel [1]** bei StartSSL. Das folgende Formular muss man zunächst mit seinen Daten wahrheitsgemäßg ausfüllen (sonst verstößt man gegen die Richtlinien und das Zertfikat kann gesperrt werden) und eine E-Mail-Adresse angeben. An diese Adresse schickt StartSSL sodann einen "Verfication Code", den man im nächsten Schritt auf der Webseite eingeben muss. Anschließend erzeugt der Server ein SSL-Clientzertifikat für die spätere Authentifizierung auf den Webseiten des Anbieters. Dazu generiert er zunächst ein Schlüsselpaar und bietet dann die Installalation des Zertifikats im Browser an.

Nun geht es an die Erzeugung des eigentlichen SSL-Zertifikats, wozu die Angabe der Domain notwendig ist, in unserem Beispiel *example.com*. Die Angabe muss ohne führendes www oder anderer Prefixe erfolgen – also nur example.com. Danach schlägt StartSSL eine Email-Adresse vor, an die die Verfikationsmail mit einem Legitimationscode gesendet werden soll. Im nächsten Schritt fordert die Express Lane die Eingabe des Code.

Anschließend bietet StartSSL zwar freundlicherweise die Generierung eines Schlüsselpaares für das Zertifikat an. Da man seinen privaten Schlüssel zur Sicherung des eigenen Servers aber nie aus der Hand geben oder ihn von jemand anderem erzeugen lassen sollte, wählt man die Option "Skip" und lädt den eigenen, zuvor erzeugten Certificate Signing Request auf den Server. Der Befehl

```
cat example.com.csr
```

gibt den Request in der Shell aus, von wo er sich in die Zwischenablage kopieren und im Formularfeld im Browser wieder einfügen lässt.

Wie bereits oben angedeutet, ignoriert StartSSL alle Angaben im CSR und trägt stattdessen eigene Informationen zur Organisation und Unit in das Zertfikat ein. Dafür fragt der Dialog artig, für welche Subdomain das Zertifikat gelten soll und verwendet diese als Common Name. Typischerweise ergänzt man dazu im Formular einfach das www. Zusätzlich trägt das Zertifikat noch den Alt Name example.com ein. Damit funktioniert das Zertifikat sowohl beim Aufruf https://www.example.com als auch mit https://example.com.

Im Anschluss generiert StartSSL das Zertifikat und bietet es als Base64-codierten Text an. Um das Zertifikat auf dem eigenen PC zu speichern, markiert man den gesamten Text inklusive der Markierungen

```
-----BEGIN CERTIFICATE-----
...
-----END CERTIFICATE-----
```

kopiert ihn und fügt ihn in die Datei *example.com.crt* ein.

Endspurt

Sowohl die CRT-Datei als auch die KEY-Datei legt man nun in den passenden Verzeichnissen des für SSL-konfigurierten Apache-Webservers ab, beispielsweise als *example.com.crt* unter */etc/ssl/certs/* und *example.com.key* unter */etc/ssl/private/*. Die Pfade in der Apache-Konfiguration muss man eventuell ebenfalls anpassen:

```
SSLCertificateFile /etc/ssl/certs/example.com.crt
SSLCertificateKeyFile /etc/ssl/private/example.com.key
```

Da einige Zertifikate von StartSSL nicht in allen Browsern enthalten sind, muss der Web-Server das Intermediate-Zertifikat (IM) von StartSSL ausliefern, mit dem das SSL-Zertifikat unterschrieben wurde. Erst damit können dann alle Browser das neue Zertifikat bis auf eine vertrauenswürdige Instanz zurückverfolgen. Das benötigte IM liegt in der Datei `sub.class1.server.ca.pem` auf dem **StartSSL-Server [2]**. Man kann es dort herunterladen und in */etc/ssl/certs/* ablegen. Mit der Angabe

```
SSLCertificateChainFile /etc/ssl/certs/sub.class1.server.ca.pem
```

in der Apache-Konfiguration macht man dem Webserver das Intermediate-Zertifikat bekannt. Um alle Änderungen wirksam zu machen, ist nun nur noch ein Neustart des Apache erforderlich. Der Aufruf von https://www.example.com bestätigt dann hoffentlich, dass man stolzer Inhaber eines regulären SSL-Zertifikats ist.

URL dieses Artikels:
http://www.heise.de/security/artikel/SSL-fuer-lau-880221.html

Links in diesem Artikel:
[1] http://www.startssl.com/?app=12
[2] http://www.startssl.com/certs

453639 **Copyright © 2015 Heise**