SwissSign

# ZERTIFIKATS-PINNING MIT SWISSSIGN

## 1. Zertifikats-Pinning (Public Key Pinning Extension for http - RFC 7469)

Mittels Zertifikats-Pinning können Serverbetreiber die für den eigenen Server akzeptablen Zertifikate einschränken. Es können dabei folgende Einschränkungen vorgenommen werden:

- Einzelne Zertifikate
- Einzelne Zertifikats Request
- Zwischenzertifizierungsstellen (Issuing CA)
- Wurzelzertifikate (Root CA)

Mit dieser Einschränkung können Man in the Middle (MITM) Attacken erschwert werden. Sobald ein Client die mit Zertifikats-Pinning geschützte Seite einmal aufgerufen hat, wird dieser Client für die konfigurierte Zeitspanne nur die gepinnten Zertifikate für diese Seite akzeptieren (Mindestens eines in der Zertifikatskette).

Welche Variante oder Kombination zum Einsatz kommt, sollte gut überlegt werden. Es kann keine allgemeingültige Regel empfohlen werden. Diese Anleitung beschreibt wie Sie die SwissSign EV Gold G22 Zwischenzertifizierungstelle sowie einen Zertifikatsrquest (CSR) pinnen.

**Falls es hier zu Änderungen kommt, empfiehlt es sich auf jeden Fall, mindestens 2 Zwischenzertifizierungsstellen anzugeben, damit man bei Wegfall oder Sperrung einer Zertifizierungsstelle immer noch eine Ausweichmöglichkeit hat.**

## 2. CSR erstellen

Dieser CSR dient als Backup, wenn kein Zertifikat der SwissSign EV Gold G22 CA mehr verwendet werden soll. Es wird ein privater Schlüssel und ein dazu passender CSR generiert und anschliessend sicher verwahrt.

Zuerst wird ein Passwort-geschützer privater Schlüssel generiert. Das Passwort ist in der Datei „passphrase.txt“ gespeichert. Als Schlüssellänge werden 4096 bits verwendet. Es kann auch eine andere Länge verwendet werden, sie sollte jedoch in keinem Fall unter 2048 bits liegen. Der konkrete Inhalt des Zertifikatsrequests ist nicht entscheidend, da nur der Public Key gepinnt wird.

```
$>openssl genrsa -aes256 -passout file:passphrase.txt -out backup_csr.privatekey 4096
```

Überprüfung des Schlüssels (Passwort wird in der Kommandozeile verlangt):

```
$>openssl rsa -in backup_csr.privatekey –check
```

Zertifikatsrequest erstellen (Beispiel):

```
$>openssl req -new -key backup_csr.privatekey -passin file:passphrase.txt -utf8 -out backup_csr.csr -subj "/C=CH/ST=Zürich/L=Glattbrugg/O=SwissSign AG/CN=swisssign.com"
```

## 3. Pins berechnen

### 3.1 Pin von Request berechnen

```
openssl req -in backup_csr.csr -pubkey -noout | openssl rsa -pubin -outform der | openssl dgst -sha256 -binary | openssl enc -base64
```

ergibt im Beispiel den Wert: NDirQI6weuLiefh9EFjP0Rg8F7iLvBQE7fdD2e+j5r8=

### 3.2 Pin von Zertifikat berechnen

Um den Pin von einem bereits existierenden Zertifikat zu berechnen muss der erste Aufruf von openssl anstatt von req , von x509 gefolgt sein:

```
$>openssl x509 -in swisssign.com.pem -pubkey -noout | openssl rsa -pubin -outform der | openssl dgst -sha256 -binary | openssl enc -base64
```

### 3.3 Pin von CA berechnen

Pin vom Zwischenzertifizierungstellenzertifikat berechnen (kann direkt übernommen werden). Zertifikat von der Zwischenzertifizerungsstelle herunterladen:

http://swisssign.net/cgi-bin/authority/download/EEFD46CAF7275E91BC5AB6E787CD0AFA550A2642

EV_Gold_G22_2014.der

Dieses Zertifikat ist im DER Format und muss ins PEM Format umgewandelt werden:

```
$>openssl x509 -in EV_Gold_G22_2014.der -inform DER -out EV_Gold_G22_2014.pem -outform PEM
```

Um den PIN zu berechnen wird nun der öffentliche Schlüssel extrahiert, diesen ins DER Format umgewandelt, den SHA256 Hash davon berechnet und in Base64 codiert:

```
$>openssl x509 -in EV_Gold_G22_2014.pem -pubkey -noout | openssl rsa -pubin -outform der | openssl dgst -sha256 -binary | openssl enc -base64
```

Das Resultat sieht folgendermassen aus:

```
mDKR5ptpp7PqVUefxx2Ftq5ymsEuzCEg+EVrLOrQFB8=
```

Liste der PIN von SwissSign CA's:

**EV_Gold_G22_2014.pem**

mDKR5ptpp7PqVUefxx2Ftq5ymsEuzCEg+EVrLOrQFB8=

**Gold_G2_2006.pem**

QPz8KlddzL/ry99s10MzEtpjxO/PO9extQXClCCuAnQ=

**Server_Gold_G22_2014.pem**

skyozdmp140ljrHvjRijq3v2/yQ1nyfFyBiA9uOKuw8=

**Server_Silver_G22_2014.pem**

mJwcSA1WE5bfCsQ5o79wGCvasvwdVsznZlqR1H3YPdI=

**Silver_G2_2006.pem**

kxgib4yDr+R/X0fCT1nOEtuoxzsYG+5rLqH0Cga8GGk=

Diese Pins können nun in der Serverkonfiguration verwendet werden. Bitte beachten Sie, dass durch fehlerhafte oder schlecht durchdachte Konfigurationen die Seite unerreichbar gemacht werden kann.

## 4. Verifikation

Mit dem curl kann einfach verifiziert werden, ob die gemachten Einstellungen funktionieren:

```
$>  curl -kIL https://testsite.domain

....

Public-Key-Pins:                    pin-sha256="1DlRtAGU4OG9/VFiB7K/Zx+MKYqctn8UliRGRYeX0Ko=";
pin-sha256="mJwcSA1WE5bfCsQ5o79wGCvasvwdVsznZlqR1H3YPdI=";
pin-sha256="NDirQI6weuLiefh9EFjP0Rg8F7iLvBQE7fdD2e+j5r8=";
max-age=2592000; includeSubDomains

.......
```