PSW GROUP

SecurITy made in Germany

HPKP – Alles was Sie wissen müssen

Was ist HTTP Public Key Pinning?

HTTP Public Key Pinning ist ein Verfahren um Ihrem Browser mitzuteilen, dass ein bestimmtes Zertifikat aus der Zertifikatskette vertrauenswürdig ist.

Wie funktioniert das Puplic Key Pinning?

Public Key Pinning gehört zu den Verfahren, die die Sicherheit von SSL-Zertifikaten immens erhöhen können. Zweck ist es, festzustellen, wann sich der öffentliche Schlüssel eines Zertifikats für einen bestimmten Host geändert hat. Dies kann etwa dann passieren, wenn ein Angreifer eine CA dermaßen beeinträchtigt, dass gültige Zertifikate für beliebige Domains ausgegeben werden können.

Ist eine TLS-Verbindung mit dem Server hergestellt, sucht der Browser jede gespeicherte Pin für den jeweiligen Hostnamen heraus und prüft, ob einer der gespeicherten Pins mit denen des SPKI-Fingerprints (= Ergebnis der SHA256-Anwendung auf die Public Key-Info) in der Zertifikatskette übereinstimmt. Scheitert diese Pin-Verifizierung, so muss die Verbindung sofort abgebrochen werden. Ist serverseitig kein HPKP konfiguriert oder unterstützt der Browser HPKP nicht, kommt die Verbindung dennoch zustande.

Wie wird gepinnt? Step-by-step Anleitung beim Apache Webserver

Private Keys generieren (einen primären und eine backup Key)

```
openssl genrsa -out www.hpkp-faq.de.key1.key 2048
openssl genrsa -out www.hpkp-faq.de.key2.key 2048
```

CSRs erstellen (für beide private Keys)

```
openssl req -new -sha256 -key www.hpkp-faq.de.key1.key -out www.hpkp-faq.de.csr
openssl req -new -sha256 -key www.hpkp-faq.de.key2.key -out
www.hpkp-faq.de.backup-csr.csr
```

Ein recht prominentes Beispiel dafür war die Ausstellung falscher Microsoft-Zertifikate, über das wir in unserem Blog berichtet haben.

SPKI Fingerprints von beiden Public Keys generieren

```
openssl req -pubkey < www.hpkp-faq.de.csr | openssl pkey -pubin -outform der |
openssl dgst -sha256 -binary | base64
hIBFVXDUBPQgeRapi9mRB7127NhGkTc+QS4EHq2LyBA=
openssl req -pubkey < www.hpkp-faq.de.backup-csr.csr | openssl pkey -pubin
-outform der | openssl dgst -sha256 -binary | base64
ZrYB07EvOX0HjbBTjJp3lEMt22nGJGqYDGU21ZnBzb8=
```

virtual Host Konfiguration anpassen

```
Header always set Public-Key-Pins: ,max-age=5184000;
pin-sha256="hIBFVXDUBPQgeRapi9mRB7127NhGkTc+QS4EHq2LyBA=";
pin-sha256="ZrYB07EvOX0HjbBTjJp3lEMt22nGJGqYDGU21ZnBzb8="'
```

Pinnen der beiden Public Keys
Alternativ gibt es die Möglichkeit vorab im Testmodus zu starten. Hierfür geben Sie *„Header always set Public-Key-Pins-Report-Only“* anstatt *„Header always set Public-Key-Pins“* an:

```
Header always set Public-Key-Pins-Report-Only: ,max-age=5184000;
pin-sha256="hIBFVXDUBPQgeRapi9mRB7127NhGkTc+QS4EHq2LyBA=";
pin-sha256="ZrYB07EvOX0HjbBTjJp3lEMt22nGJGqYDGU21ZnBzb8="'
```

Reporting anschalten

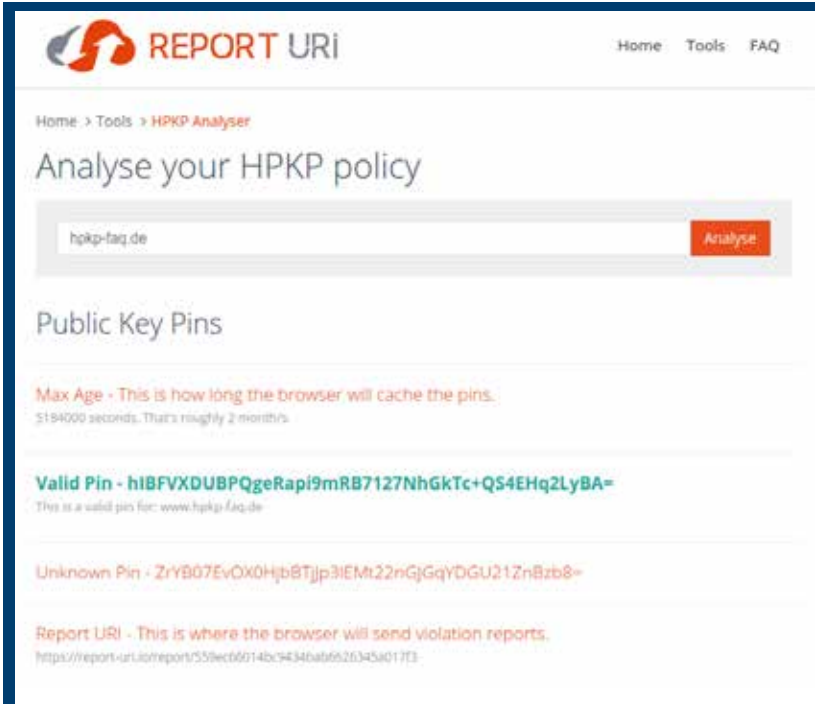
Beim HTTP Public Key Pinning gibt es die Möglichkeit sich vom zugreifenden Browser, Fehlermeldungen anzeigen zu lassen. Allerdings wird dies erst mit dem Chrome Version 46 unterstützt.

Die persönliche URI wird mit dem Flag „report-uri“ ebenfalls in die virtual Host Konfiguration eingebaut:

```
Header always set Public-Key-Pins: ,max-age=5184000;
pin-sha256="hIBFVXDUBPQgeRapi9mRB7127NhGkTc+QS4EHq2LyBA=";
pin-sha256="ZrYB07EvOX0HjbBTjJp3lEMt22nGJGqYDGU21ZnBzb8=";
report-uri="https://report-uri.io/report/559ec66014bc9434bab6626345a017f3"'
```

Über Qualys ssllabs Test können Sie Ihre Konfiguration testen. Nutzen Sie hierfür: https://www.ssllabs.com/ssltest/ oder vorzugsweise https://dev.ssllabs.com/ssltest/.

Registrieren Sie sich kostenlos unter https://report-uri.io/ und kopieren Sie sich Ihre persönliche Reporting URI:



HTTP-Erweiterung HPKP: wie soll der Header aussehen?

Mittels curl – I https:// www.beispiel.de können Sie sich den Header auf der Konsole anzeigen lassen.

Der HTTP Header kann wie folgt aussehen:

```
HTTP/1.1 200 OK
Date: Thu, 12 Nov 2015 10:45:37 GMT
Server: Apache
Strict-Transport-Security: max-age=31536000; includeSubDomains; preload
Public-Key-Pins: max-age=5184000; pin-sha256="hIBFVXDUBPQgeRapi9mRB7127NhGGkT
c+QS4EHq2LyBA=";
pin-sha256="ZrYB07EvOX0HjbBTjJp3lEMt22nGJGqYDGU21ZnBzb8=";
report-uri="https://report-uri.io/report/559ec66014bc9434bab6626345a017f3"
X-Powered-By: PHP/5.5.30
Content-Type: text/html; charset=UTF-8
```

Der Header spezifiziert mindestens pin-sha256 Werte, d. h.: die Pins von zwei öffentlichen Schlüsseln. Dabei ist ein Pin der eines beliebigen öffentlichen Schlüssels, der sich in der aktuellen Zertifikatskette befindet, der andere ist der eines beliebigen öffentlichen Schlüssels, der sich nicht in der aktuellen Zertifiktaskette befinden muss. Bei letzterem könnte es sich einen Backup-Key handeln, der beispielsweise dann zum Einsatz kommt, wenn Ihr Zertifikat abläuft oder zurückgezogen werden muss.

Welche Informationen enthält das Zertifikat?

Senden Sie die Certificate Signing Request (CSR) mit Ihrem Public Key an eine Zertifizierungsstelle, stellt Ihnen diese ein gültiges Zertifikat aus. Dieses enthält den öffentlichen Schlüssel des RSA-Schlüsselpars sowie ein Ablaufdatum. Sowohl der öffentliche Schlüssel als auch das Ablaufdatum werden von der Zertifizierungsstelle signiert, sodas jedwede Veränderungen an diesen beiden Komponenten das Zertifikat sofort ungültig werden lassen. X.509-Zertifikate enthalten auch andere Bereiche, um TLS-Verbindungen vernünftig zu authentifzieren, etwa den Hostnamen Ihres Servers sowie weitere Details.

Wie können RSA-Schlüssel dargestellt werden?

Mit dem Befehl

```
openssl genrsa 2048
```

erzeugen Sie einen 2.048-bit-RSA-Schlüssel und geben ihn auf der Konsole aus. Wenngleich es -----BEGIN RSA PRIVATE KEY----- heißt, wird nicht ausschließlich der private Schlüssel ausgegeben, sondern auch die ASN.1-Struktur, die ebenfalls einen Public Key enthält. So erzeugen Sie also ein RSA-Schlüsselpaar.

Ein weit verbreiteter Irrtum in der Kryptographie besteht darin, dass der RSA-Schlüssel selbst für ein bestimmtes Zertifikat ablaufen kann. RSA-Schlüssel laufen jedoch nie ab – es sind letztlich nur Zahlen. Das Zertifikat jedoch, das den Public Key enthält, kann sehr wohl ablaufen. Deshalb kann auch nur das Zertifikat zurückgezogen werden. Die Schlüssel selbst laufen ab oder werden zurückgezogen, sobald es keine gültigen Zertifikate mehr gibt, die eben diesen öffentlichen Schlüssel verwenden, oder wenn Sie den Schlüssel vernichtet haben oder überhaupt aufgehört haben, diesen zu verwenden.

Was tun, wenn Sie Ihr Zertifikat ersetzen müssen?

Läuft Ihr Zertifikat ab oder wurde Ihr privater Schlüssel kompromittiert, müssen Sie Ihr Zertifikat womöglich zurückziehen, in jedem Fall aber ersetzen. Dies kann dazu führen, dass Ihre Pin ungültig wird: die Beschränkungen, die beim Kauf eines neuen gültigen Zertifikat existieren, sind dieselben, vor denen Angreifer stehen, die versuchen, Ihre Identität anzunehmen und Ihre TLS-Sitzung abzufangen.

Die Pin-Verifikation verlangt, dass sämtliche SPKI-Fingerprints aller Zertifikate einer Kette überprüft werden, und sie gelingt in dem Moment, in dem einer der Public Key zu einem der Pins passt. Nehmen wir an, Zertifizierungsstelle XY hat Ihr Zertifikat signiert und Sie haben ein weiteres Class 1- oder Class 2-Zwischenzertifikat sowie deren Root-Zertifikat in der Kette: der Browser vertraut ausschließlich dem Root-Zertifikat, jedoch werden die Zwischenzertifikate vom Root-Zertifikat signiert. Das Zwischenzertifikat wiederum signiert das auf den Server ausgelieferte Zertifikat. Dies nennt man Vertrauenskette. Haben Sie nun Ihr Zwischenzertifikat gepinnt, ist die einzige Option der Wiederherstellung der Backup-Key. Auf was auch immer dieser verweist, muss im neuen Zertifikat gespeichert sein, wenn Sie jene Benutzer auf Ihrem Server wieder zulassen möchten, die Ihre Pin aus vorherigen Sitzungen gespeichert haben.

Eine einfachere Lösung wäre sicherlich, wenn Sie den SPKI-Fingerprint aus dem Zwischenzertifikat Class 1 verfügbar zu machen. Um eine neue und gültige Zertifikatskette zu schaffen, bitten Sie Ihre Zertifizierungsstelle, ein neues Zertifikat für einen neuen oder Ihren aktuellen Schlüssel auszustellen. Mit einer etwas größeren Angriffsfläche zahlen Sie jedoch einen Preis dafür, denn jemand, der den Private Key des Zwischenzertifikats gestohlen hat, wäre nun in die Lage versetzt, Ihre Seite zu imitieren und Key Pinning-Prüfungen erfolgreich zu durchlaufen.

Eine andere Option besteht darin, das Wurzelzertifikat zu pinnen. Jedes beliebige, durch die CA ausgestellte Zertifikat würde es Ihnen erlauben, eine neue gültige Zertifikatskette zu erstellen. Auch dadurch würde sich die Angriffsfläche dezent erhöhen, da jedes beeinträchtigte Zwischen- oder Wurzelzertifikat es gestatten kann, Ihre Seite zu imitieren und Pinning-Checks zu bestehen.

Welcher Schlüssel soll gepinnt werden?

In Anbetracht all der genannten Szenarien fragen Sie sich vielleicht, welchen Schlüssel Sie am besten fürs Pinnen verwenden, und die Antwort kann nur lauten: das kommt drauf an. Sie können einen oder alle öffentlichen Schlüssel in Ihrer Zertifikatskette pinnen, und das wird funktionieren. Die Spezifikation verlangt, dass Sie mindestens zwei Pins haben, sodass Sie den SPKI-Hash eines anderen Rootzertifikats einfügen müssen, ein anderes Zwischenzertifikat (eine andere Stufe Ihrer aktuellen CA würde auch funktionieren) oder ein anderes untergeordnetes Zertifikat. Die einzige Anforderung ist: der Pin entspricht nicht dem Hashwert von irgendeinem Zertifikat innerhalb der aktuellen Kette. Der Browser kann nicht feststellen, ob Sie ihm ein gültiges, sinnvolles Backup bereitgestellt haben - weshalb er auch gerne Zufallswerte akzeptiert.

Beim Pinnen auf eine kleine Auswahl von CAs zurückzugreifen, bei denen Sie sich wohl und sicher fühlen, hilft Ihnen, das Risiko einzuschränken. Beim Pinnen nur Ihre untergeordneten Zertifikate zu verwenden, birgt ein erhöhtes Risiko, bietet jedoch mehr Sicherheit. Es ist ein bisschen wie Fahren ohne Gurt: es funktioniert meistens, aber falls etwas schief geht, geht es in aller Regel so richtig schief. Und das wollen Sie sicherlich vermeiden.

Wenn Sie ausschließlich Ihre untergeordneten Zertifikate pinnen, besteht außerdem die Gefahr, dass Sie einen Backup-Schlüssel generieren, der an Uraltstandards festhält und vielleicht nicht mehr benutzt werden kann, wenn Sie Ihr aktuelles Zertifikat ersetzen müssen. Drehen wir die Uhr drei Jahre zurück. Ihr Backup-Key ist ein 1.024-Bit-RSA-Schlüsselpaar. Sie pinnen für ein Jahr, dann läuft das Zertifikat ab. Sie gehen zur CA und bitten um ein neues Zertifikat für Schlüssel A, die CA sagt jedoch, dass der Schüssel ist zu kurz und zu schwach ist. Sie fragen nach dem Backup-Schlüssel, der ebenfalls zurückgewiesen wird, weil er zu kurz ist. Ergebnis: Weil Sie beim Pinnen nur von Ihnen kontrollierte Schlüssel verwendet haben, sind sie nun quasi eingemauert bzw. sperren potenzielle Besucher Ihrer Website aus.

PSW GROUP: Mit Sicherheit Ihr Partner

Die PSW GROUP ist Ihr Full-Service-Provider für Internetlösungen mit einem besonderen Schwerpunkt auf Internet Security. Als Dienstleister bieten wir Ihnen sowohl für den Webeinsatz als auch für die E-Mail-Kommunikation maßgeschneiderte Zertifikats-, Signatur-, Verschlüsselungs- und Authentifizierungslösungen an. Unser voll umfassendes Produktportfolio reicht dabei von SSL-Zertifikaten über Code Signing-Zertifikate bis hin zu S/MIME-Zertifikaten.

Neben der großen Produktvielfalt verfügen wir über langjährige Expertise in den Bereichen Internetsicherheit und IT-Recht. Mit unserem tiefgreifenden und zertifizierten Fachwissen als IRCA ISO 27001 ISMS Lead-, Datenschutz- und IT-Security-Auditoren, Certified Information Systems Security Professionals , Security Manager, Cyber-Security Practitioners und Symantec Sales Experts Plus sowie Sophos Certified Architects, stehen wir Ihnen auch bei den komplexesten Fragestellungen beratend zur Seite.

+49 661 480276 10

info@psw-group.de

PSW GROUP
GmbH & Co. KG
Flemingstrasse 20-22
D-36041 Fulda

Phone +49 661 480276 10
Fax +49 661 480276 19
E-Mail info@psw-group.de
Internet www.psw-group.de