HTTP Public Key Pinning

In this lesson, we'll study HTTP Public Key Pinning.

We'll cover the following

- Why HTTP Public Key Pinning?
- HPKP is dangerous

Why HTTP Public Key Pinning?

HTTP Public Key Pinning (abbr. HPKP) is a mechanism that allows us to advertise which SSL certificates to expect when a browser connects to our servers. It is a *trust on first use* header, just like HSTS, meaning that, once the client connects to our server, it will store the certificate's info for subsequent interactions.

If at any point in time the client detects that another certificate is being used by the server, it will politely refuse to connect, rendering *man in the middle* (MITM) attacks very hard to pull off.

This is what an HPKP policy looks like:

```
Public-Key-Pins:
  pin-sha256="9yw7rfw9f4hu9eho4fhh4uifh4ifhiu=";
  pin-sha256="cwi87y89f4fh4fihi9fhi4hvhuh3du3=";
  max-age=3600; includeSubDomains;
  report-uri="https://pkpviolations.example.org/collect"
```

HPKP is dangerous

The header advertises what certificates the server will use (in this case it's two of them) using a hash of the certificates. It includes additional information like the time-to-live of this directive (`max-age=3600`), and a few other details. Sadly, there's no point in digging deeper to understand what we can do with public key pinning, as this feature is being deprecated by Chrome, a signal that its adoption is destined to plummet very soon.

Chrome's decision is not irrational, it is simply a consequence of the risks associated with public key pinning. If you lose your certificate, or simply make a mistake while testing, your website is gone for the duration of the `max-age` directive, which is typically weeks or months. As a result of these potentially catastrophic consequences, adoption of HPKP has been extremely low, and there have been incidents where big-time websites have been unavailable because of a misconfiguration.

All things considered, Chrome decided users were better off without the protection offered by HPKP, and security researchers aren't entirely against this decision.

> ### ℹ HPKP gone wrong
>
> Smashing Magazine, a leading website in the field of web design, documented its disastrous experience with HPKP in a blog post in late 2016.
>
> Long story short, the website was unavailable, due to a misconfiguration in their `Public-Key-Pins` header. When their SSL certificate expired, they had no way to issue a new certificate that would not violate their previously set HPKP policy. As a result, most of their users could not access the website for four days.
>
> Moral of the story? HPKP is dangerous and even the best make mistakes.

While HPKP has been deprecated, a new header stepped in to prevent fraudulent SSL certificates from being served to clients, `Expect-CT`. Let's study it in the next lesson.