digicert®

BEST PRACTICES

07-21-2020

STOP CERTIFICATE PINNING

JEREMY ROWLEY

What is certificate pinning?

Certificate pinning restricts which certificates are considered valid for a particular website, limiting risk. Instead of allowing any trusted certificate to be used, operators "pin" the certificate authority (CA) issuer(s), public keys or even end-entity certificates of their choice. Clients connecting to that

server will treat all other certificates as invalid and refuse to make an HTTPS connection.

Pinning allows websites to control the risk of misissuance, CA compromise, or man-in-the-middle attacks. Pinning takes multiple forms depending on the use case - I can pin my certificate as the only one in my client trust store or write the public key hash into my code so only my key is trusted. When pinning started becoming popular, the hope was that these extra layers of complexity made it harder for bad actors to use certificates in attacks or spoofs.

Google was one of the first to use pinning in 2011, when they pinned the issuing CAs for their main websites in the Chrome browser. When Chrome connected to google.com, it already knew which CAs to accept. If a certificate from any other CA was presented, the connection would be blocked. This meant that if an attacker managed to fool any other trusted CA into giving them a certificate for google.com, it would still be blocked by Chrome.

A few years later, Chrome and Firefox started allowing sites to use HTTP Public Key Pinning (HPKP) headers. The first time your browser connected to a website using HPKP, it recorded the public key from the header, and would only accept that key every time it connected to the site, up until the "max-age" defined in the HPKP policy. If a max-age of 60 days was set, no other keys would be accepted for the next 60 days.

Meanwhile, certificate pinning was also introduced in apps, IoT devices, and other software. Using similar methods an app could pin a certificate, and the app would then refuse any connections to the server if they were not using that certificate, protecting users from any man-in-the-middle attack.

These practices, when implemented correctly, could enhance security, but it did not take long for the web community to find out pinning was not such a great idea.

What can go wrong with Certificate Pinning?

Pinning, especially with HPKP, was extremely risky and error prone. If you configured your pinning settings incorrectly, you could block access to your own website or break connectivity in your application, with limited options for recourse. Here are just a few ways pinning can cause such harm.

Key Compromise

A common practice with HPKP was to pin the end-entity certificate public key to a website for 60 days. Many sites did not specify any backup keys, perhaps because they were unaware it was an option, or they underestimated the risk of using a single key.

This left sites vulnerable to key compromise. Industry standards require that CAs revoke compromised certificates - perhaps stolen from an insecure webserver, or accidentally uploaded to a public GitHub repository - within 24 hours. With your only pinned key now compromised, you have no replacement and clients who recorded your HPKP policy remember that bad key and will not allow connections with your new certificate.

Hackers

HPKP is a great way for hackers to sabotage a website and do long-term damage. If I can take over your server and set a bogus HPKP policy for a fake key and a one-year max-age, browsers will always fail to connect. Long after you re-secure your server, you are still stuck with the effects of that HPKP policy that are not easy to fix.

Certificate Authority Revocations

Sometimes CAs must revoke your certificates. Maybe an audit shows the certificates have previously unknown issues, like misspellings in the subject name or invalid entries in the OU fields. Industry standards say the CA has five days to revoke your certificates, but you pinned them in your client code. How can you push out updates to all your clients in five days to start using your new replacement certificates?

More risk than reward

As a result of these problems and the difficulties of implementing pinning safely and robustly, there were more cases of sites being harmed by pinning than protected. These are just a few of the issues with pinning which led Google and Firefox to remove HPKP support just a couple years after it was introduced.

The biggest problem with pinning is that you lose the ability to respond to certificate issues. If you need to change keys, certificates, issuers, or your CA

vendor, for any reason, you must fix your client, browser, code, IoT device, etc. - sometimes on a short schedule. If you are committed to supporting an application version for years and it contains a pinned certificate, how can you be sure the certificate will remain valid for the entire lifetime of your application? Pinning is especially problematic with publicly trusted TLS certificates because they must adhere to ever-evolving rules, decreasing maximum lifetimes and other surprises.

Luckily, HPKP is a thing of the past, and DigiCert has not been a big proponent of other types of public key pinning. DigiCert recommends you do not use pinning; the complexities and consequences outweigh the benefits.

What is DigiCert doing to discourage inappropriate pinning?

While we haven't recommended or instructed users to implement pinning in recent years, it is still possible to set up pinning on your own. This week, DigiCert is making a change to our CA hierarchy. We will start replacing our public TLS-issuing intermediate CAs (ICAs) with shorter versions, updated every six months. Of course, the validity periods of the intermediates will be long enough to exceed all the one- and two-year certificates issued during the six months the ICAs are used. Shorter ICA lifetimes will disincentivize pinning them since they will be changing more frequently.

We will initially replace the GeoTrust RSA CA 2018 and RapidSSL RSA CA 2018 intermediates for Domain Validated (DV) issuance with the new GeoTrust TLS DV RSA Mixed SHA256 2020 CA-1 and RapidSSL TLS DV RSA Mixed SHA256 2020 CA-1. After six months, these will be replaced by the GeoTrust TLS DV RSA Mixed SHA256 2021 CA-1 and RapidSSL TLS DV RSA Mixed SHA256 2021 CA-1, then six months later, the *2021 CA-2 versions, and so on.

Additional CAs will be replaced over the coming months until all our default public TLS issuers are are rotated every six months. The schedule for these replacements will be posted here:

Beyond helping put pinning behind us all, shortening ICA lifetimes will have other benefits. It will group certificates into smaller buckets so changes to one set of certificates issued under one CA will not always affect others. If an ICA must be deprecated, it will only affect the certificates issued for the six months that CA was actively issuing, and only the specific types of certificates that were allowed under that CA.

Related Stories

Setting Global Standards for Secure Email Certificates

CA/B Forum Update on EV Certificate Improvements

1-Year Certificates are Here: What Now?