HTTP Public Key Pinning

By Contributors to Wikimedia projects

Published: 2015-03-08 · Archived: 2026-04-05 14:29:13 UTC

From Wikipedia, the free encyclopedia

HTTP Public Key Pinning (**HPKP**) is an obsolete Internet security mechanism delivered via an HTTP header which allows HTTPS websites to resist impersonation by attackers using misissued or otherwise fraudulent digital certificates.[1] A server uses it to deliver to the client (e.g. a web browser) a set of hashes of public keys that must appear in the certificate chain of future connections to the same domain name.

For example, attackers might compromise a certificate authority, and then mis-issue certificates for a web origin. To combat this risk, the HTTPS web server serves a list of "pinned" public key hashes valid for a given time; on subsequent connections, during that validity time, clients expect the server to use one or more of those public keys in its certificate chain. If it does not, an error message is shown, which cannot be (easily) bypassed by the user.

The technique does not pin certificates, but public key hashes. This means that one can use the key pair to get a certificate from any certificate authority, when one has access to the private key. Also the user can pin public keys of root or intermediate certificates (created by certificate authorities), restricting site to certificates issued by the said certificate authority.

Due to HPKP mechanism complexity and possibility of accidental misuse (potentially causing a lockout condition by system administrators), in 2017 browsers deprecated HPKP and in 2018 removed its support in favor of Certificate Transparency.[2][3]

The server communicates the HPKP policy to the user agent via an HTTP response header field named `Public-Key-Pins` (or `Public-Key-Pins-Report-Only` for reporting-only purposes).

The HPKP policy specifies hashes of the subject public key info of one of the certificates in the website's authentic X.509 public key certificate chain (and at least one backup key) in `pin-sha256` directives, and a period of time during which the user agent shall enforce public key pinning in `max-age` directive, optional `includeSubDomains` directive to include all subdomains (of the domain that sent the header) in pinning policy and optional `report-uri` directive with URL where to send pinning violation reports. At least one of the public keys of the certificates in the certificate chain needs to match a pinned public key in order for the chain to be considered valid by the user agent.

At the time of publishing, RFC 7469 only allowed the SHA-256 hash algorithm. (Appendix A. of RFC 7469 mentions some tools and required arguments that can be used to produce hashes for HPKP policies.)

A website operator can choose to either pin the root certificate public key of a particular root certificate authority, allowing only that certificate authority (and all intermediate authorities signed by its key) to issue valid certificates

for the website's domain, and/or to pin the key(s) of one or more intermediate issuing certificates, or to pin the end-entity public key.

At least one backup key must be pinned, in case the current pinned key needs to be replaced. The HPKP is not valid without this backup key (a backup key is defined as a public key not present in the current certificate chain).[4]

HPKP is standardized in RFC 7469.[1] It expands on static certificate pinning, which hardcodes public key hashes of well-known websites or services within web browsers and applications.[5]

Most browsers disable pinning for certificate chains with private root certificates to enable various corporate content inspection scanners[6] and web debugging tools (such as mitmproxy or Fiddler). The RFC 7469 standard recommends disabling pinning violation reports for "user-defined" root certificates, where it is "acceptable" for the browser to disable pin validation.[7]

If the user agent performs pin validation and fails to find a valid SPKI fingerprint in the served certificate chain, it will POST a JSON formatted violation report to the host specified in the report-uri directive containing details of the violation. This URI may be served via HTTP or HTTPS; however, the user agent cannot send HPKP violation reports to an HTTPS URI in the same domain as the domain for which it is reporting the violation. Hosts may either use HTTP for the `report-uri`, use an alternative domain, or use a reporting service.[8]

Some browsers also support the `Public-Key-Pins-Report-Only`, which only triggers this reporting while not showing an error to the user.

Criticism and decline

[edit]

During its peak adoption, HPKP was reported to be used by 3,500 of top 1 million internet sites, a figure that declined to 650 around the end of 2019.[9]

Criticism and concern revolved around malicious or human error scenarios known as HPKP Suicide and RansomPKP.[10][11] In such scenarios, a website owner would have their ability to publish new contents to their domain severely hampered by either losing access to their own keys or having new keys announced by a malicious attacker.

Browser support and deprecation

[edit]

Browser support for HTTP Public Key Pinning

Browser	Version added	Version deprecated	Version removed	Notes
Google Chrome	46[12]	67[13]	72[14][15]	

Opera	33[16]	54[17]	60	
Firefox	35	72[18][19]	78[20]	
Internet Explorer	N/a[21]	N/a	N/a	
Microsoft Edge	N/a[21]	N/a	N/a	
Safari	N/a	N/a	N/a	

- Certificate authority compromise
- Certificate Transparency
- HTTP Strict Transport Security
- List of HTTP header fields
- DNS Certification Authority Authorization
- Public Key Pinning Extension for HTTP (HPKP) on MDN Web Docs

1. ^ Jump up to: ***a*** ***b*** *Evans, Chris; Palmer, Chris; Sleevi, Ryan (April 2015). Public Key Pinning Extension for HTTP. IETF. doi:10.17487/RFC7469. ISSN 2070-1721. RFC 7469.*
2. ^ *Leyden, John (2017-10-30). "RIP HPKP: Google abandons public key pinning". The Register. Retrieved 2018-12-18.*
3. ^ *Tung, Liam (2017-10-30). "Google: Chrome is backing away from public key pinning, and here's why". ZDNet. Retrieved 2018-12-18.*
4. ^ *"About Public Key Pinning". noncombatant.org. Retrieved 2015-05-07.*
5. ^ *"Certificate and Public Key Pinning - OWASP". www.owasp.org. Retrieved 2015-05-07.*
6. ^ *"Security FAQ - The Chromium Projects". www.chromium.org. Retrieved 2015-07-07.*
7. ^ *Evans, C.; Palmer, C.; Sleevi, R. (2015). "RFC 7469 - Public Key Pinning Extension for HTTP". tools.ietf.org. doi:10.17487/RFC7469. Retrieved 2015-07-07.*
8. ^ *"HPKP Violation Reporting". Scott Helme.*
9. ^ *"HPKP is no more". Scott Helme. 2020-01-20. Retrieved 2020-01-30.*
10. ^ *"Abusing Bleeding Edge Web Standards for AppSec Glory". Bryant Zadegan and Ryan Lester. 2016-08-03. Retrieved 2026-01-14.*
11. ^ *"Using security features to do bad things". Scott Helme. 2016-08-15. Retrieved 2020-01-30.*
12. ^ *Stark, Emily (2015-08-31). "Rolling out public key pinning with HPKP reporting". Chrome Developers. Archived from the original on 2023-01-16. Retrieved 2023-03-10.*
13. ^ *Medley, Joe (2018-06-07). "Deprecations and removals in Chrome 67". Google Developers. Archived from the original on 2023-03-10. Retrieved 2023-03-10.*
14. ^ *Palmer; Estark; Rsleevi (2022-09-13). "Remove HTTP-Based Public Key Pinning - Chrome Platform Status". www.chromestatus.com. Archived from the original on 2022-05-25. Retrieved 2019-11-18.*
15. ^ *Medley, Joe (2020-06-27). "Deprecations and removals in Chrome 72 - Chrome Developers". Chrome Developers. Archived from the original on 2022-11-18. Retrieved 2023-03-10.*
16. ^ *Bynens, Mathias (2015-10-27). "Opera 33 released". GitHub. Opera. Archived from the original on 2023-03-10. Retrieved 2023-03-10.*

17. ^ *"What's new in Chromium 67 and Opera 54"*. *GitHub. Opera. 2018-06-28. Archived from the original on 2023-03-10. Retrieved 2023-03-10.*
18. ^ *"HTTP Public Key Pinning is no longer supported"*. *Firefox Site Compatibility. November 14, 2019. Archived from the original on 2020-05-29. Retrieved 2020-02-19.*
19. ^ *Keeler, Dana (2019-11-13). "Mozilla source code change that removed HPKP including discussion and reasons for this change (bug 1412438)". Mozilla Firefox Version Control. Archived from the original on 2023-03-10. Retrieved 2023-03-10.*
20. ^ *"remove HPKP (http public key pinning) entirely (not built-in pins)". bugzilla.mozilla.org. "HPKP is disabled by default (bug 1412438). Due to socket process work, it has already become a maintenance burden (see bug 1485652). This bug will remove HPKP entirely."*
21. ^ Jump up to: ***a*** ***b*** *"The status of Public Key Pinning Extension for HTTP in Microsoft Edge is Under Consideration". Microsoft Edge Development. Archived from the original on 2016-12-20. Retrieved 2018-09-21.*
