November 2022

DOMAIN BASED HSTS (HTTP STRICT TRANSPORT SECURITY) MANAGEMENT

HP INC

Domain based HSTS (HTTP Strict Transport Security) Management

Overview

HTTP-based APIs, including websites hosting web content, want to force HTTPS for security; however, the current mechanism is not absolute in its intentions and requires third-party configuration to help enforce the policy. On top of that, the configuration is very coarse grained and does not meet many use cases where fine grained control is required, such as having different policies for sub-domains of a second-level domain.

By allowing web domains to host their own configuration, we can give fine grained control to the owners of domains and help to secure the web from malicious actors.

Problems Solved

The migration from HTTP to HTTPS across the industry has been slow to progress and the lack of forced HTTPS leads to Man in The Middle (MITM) attacks, but API and Website providers have an option to set a HTTP header called Strict-Transport-Security to tell browsers that all connections must use HTTPS. The shortcoming is that the first connection could be intercepted via MITM and result in a browser not receiving HSTS.

Prior Solutions

As noted, the Strict-Transport-Security header can be used to tell browsers that all connections should use HTTPS. Additional details on suggested configurations are found in [8] and [9]. Even with these configurations, however, “there’s a small window where a user visiting the site can still be subject to a MITM attack.” [9] There are already several known exploits available to bypass HSTS. [10][11]

To combat this a preload service has been built that is maintained by Google. [1][2] Preloading a domain through this service will result in the domain being “hardcoded into Chrome as being

HTTPS only." [3] However setting this up for subdomains is prohibited as it is expected to be done at the second level domain. [4] If an entire domain is not able to support HSTS due to not being able to redirect from port 80 to 443 directly or if there are for instance multiple redirects [5] then the domain cannot be added. Additionally, and specific to this disclosure a subdomain would not then be able to use preload and could not therefore increase the security of a service running on a subdomain. Further there needs to be a header set at the top-level domain which often cannot be done by the owner of a subdomain.

Description

Describe the invention including any novelty and detectability.

Browsers could be made to check a service or endpoint running at the top-level domain by default and companies could self manage the list of subdomains included in the preload list. For example, when a call is made to `sub.example.com` or `sub2.sub1.example.com` the first connection the browser would make would be to `https://example.com/.well-known/hsts.txt` and what would be returned would be a list via Transport Layer Security (TLS) of all subdomains that are required to be connected via HTTPS only. The requirement to prevent MITM attacks would be that this connection must be made using HTTPS and have a valid CA issued TLS certificate that can be verified. In this way, domain owners would not need to presubmit their domains to a preloading service such as the one hosted by Google, but rather could self manage. Further subdomains could be added at will and not suffer increased lag times should the need arise to remove them from the list since the current model can take 6-12 weeks to have a domain removed.[6]

Client libraries that are used outside of a browser environment could be modified to support the same functionality in a way that did not require the collaboration of browser vendors but would function in a similar manner where the http client library would make calls to the Second Level Domain (SLD) and check for a service to indicate which subdomains would require HTTPS for all connections prior to beginning to negotiate traffic.

To implement the ideas in this disclosure, the process could build upon current standards already in place and perhaps create a new one. For example, there is already wide adoption of the `robots.txt` standard to indicate to web crawlers where to look in your website for searchable pages. The ideas in this disclosure could use the same data formats in `robots.txt` to help speed adoption. Another example is the "well known" [7] endpoints where the ideas in this disclosure could be standardized to be available as a well-known endpoint. With these standards, the ideas in this disclosure could be standardized in the IETF or some other appropriate governing body.

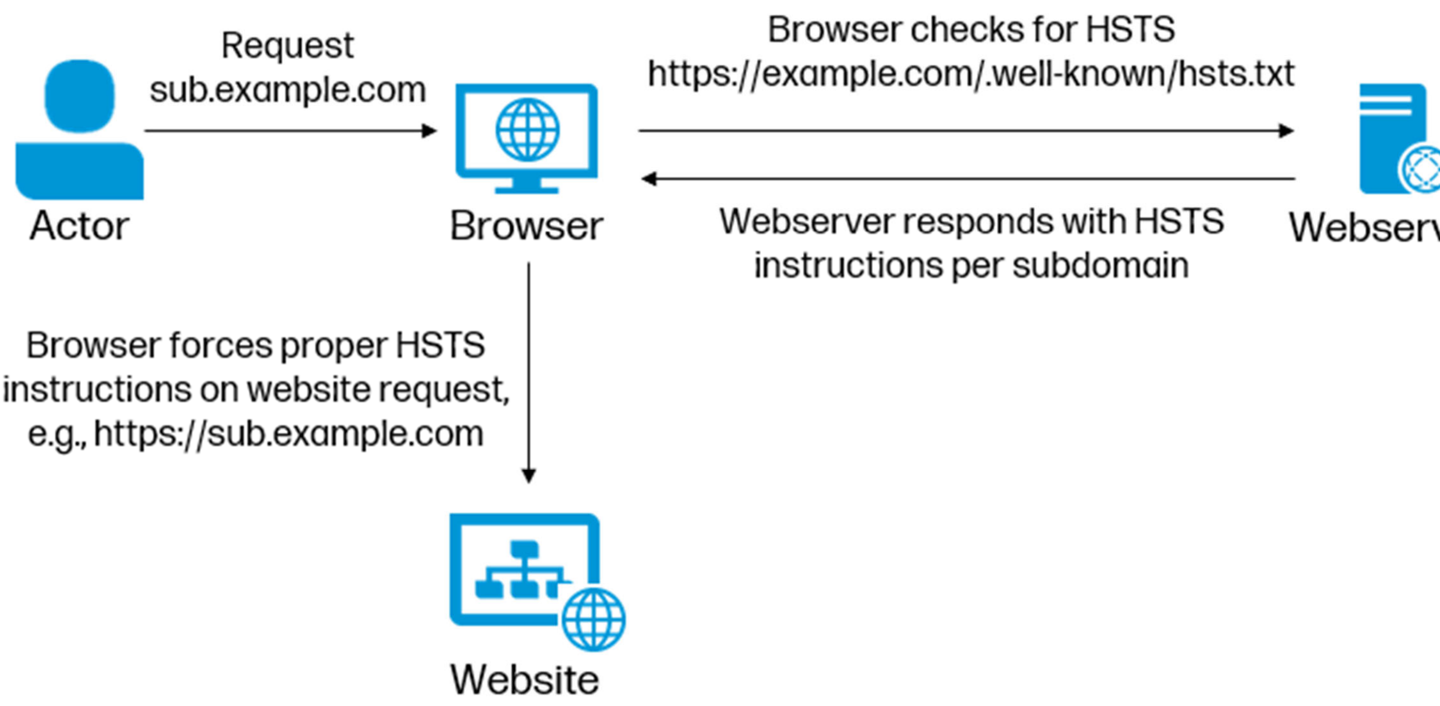
Request
sub.example.com
Actor
Browser
Browser checks for HSTS
https://example.com/.well-known/hsts.txt
Webserver responds with HSTS
instructions per subdomain
Webserv
Browser forces proper HSTS
instructions on website request,
e.g., https://sub.example.com
Website

Figure 1: Example flow for HSTS instructions enforcement

Advantages

Detail the advantages of the disclosure
See above.

This innovation could be applied to devices, e.g., MFP printers, in the firmware to have tighter control over secure connections. This would help our customers using these devices, especially commercial, who want to ensure their data to/from the printer is always encrypted, i.e., between cloud storage. Customers could choose to deploy this service internally to their network as supported by hp devices and maintain control individually without outside collaboration on their network.

References

[1] https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security#preloading_strict_transport_security

[2] https://hstspreload.org/

[3] Ibid.

[4] https://hstspreload.org/?domain=id.hp.com

[5] https://hstspreload.org/?domain=hp.com

[6] https://hstspreload.org/removal/

[7] https://en.wikipedia.org/wiki/Well-known_URI

[8] https://scotthelme.co.uk/hsts-cheat-sheet/

[9] https://www.maxivanov.io/http-strict-transport-security/

[10] https://sathisharthars.wordpress.com/2015/02/27/bypassing-hsts-http-strict-transport-security-with-mitmf/

[11] https://cloudacademy.com/lab/using-mitmf-bypass-hsts/

Disclosed by Chris Myers, Shane I Saunders, Leonardo Eloy and Paul Michael Anderson, HP Inc.