HTTP Strict Transport Security Cheat Sheet

Introduction

HTTP Strict Transport Security (also named **HSTS**) is an opt-in security enhancement that is specified by a web application through the use of a special response header. Once a supported browser receives this header that browser will prevent any communications from being sent over HTTP to the specified domain and will instead send all communications over HTTPS. It also prevents HTTPS click through prompts on browsers.

The specification has been released and published end of 2012 as RFC 6797 (HTTP Strict Transport Security (HSTS)) by the IETF.

Threats

HSTS addresses the following threats:

- User bookmarks or manually types `http://example.com` and is subject to a man-in-the-middle attacker
  - HSTS automatically redirects HTTP requests to HTTPS for the target domain
- Web application that is intended to be purely HTTPS inadvertently contains HTTP links or serves content over HTTP
  - HSTS automatically redirects HTTP requests to HTTPS for the target domain
- A man-in-the-middle attacker attempts to intercept traffic from a victim user using an invalid certificate and hopes the user will accept the bad certificate
  - HSTS does not allow a user to override the invalid certificate message

Examples

Simple example, using a long (1 year = 31536000 seconds) max-age. This example is dangerous since it lacks `includeSubDomains`:

`Strict-Transport-Security: max-age=31536000`

This example is useful if all present and future subdomains will be HTTPS. This is a more secure option but will block access to certain pages that can only be served over HTTP:

`Strict-Transport-Security: max-age=31536000; includeSubDomains`

This example is useful if all present and future subdomains will be HTTPS. In this example we set a very short max-age in case of mistakes during initial rollout:

`Strict-Transport-Security: max-age=86400; includeSubDomains`

Recommended:

- If the site owner would like their domain to be included in the HSTS preload list maintained by Chrome (and used by Firefox and Safari), then use the header below.
- Sending the `preload` directive from your site can have **PERMANENT CONSEQUENCES** and prevent users from accessing your site and any of its subdomains if you find you need to switch back to HTTP. Please read the details at preload removal before sending the header with `preload`.

`Strict-Transport-Security: max-age=31536000; includeSubDomains; preload`

The `preload` flag indicates the site owner's consent to have their domain preloaded. The site owner still needs to then go and submit the domain to the list.

Problems

Site owners can use HSTS to identify users without cookies. This can lead to a significant privacy leak. Take a look here for more details.

Cookies can be manipulated from sub-domains, so omitting the `includeSubDomains` option permits a broad range of cookie-related attacks that HSTS would otherwise prevent by requiring a valid certificate for a subdomain. Ensuring the `secure` flag is set on all cookies will also prevent, some, but not all, of the same attacks.

Browser Support

As of September 2019 HSTS is supported by all modern browsers, with the only notable exception being Opera Mini.

References

- Chromium Projects/HSTS
- OWASP TLS Protection Cheat Sheet
- sslstrip
- AppSecTutorial Series - Episode 4
- Nmap NSE script to detect HSTS configuration