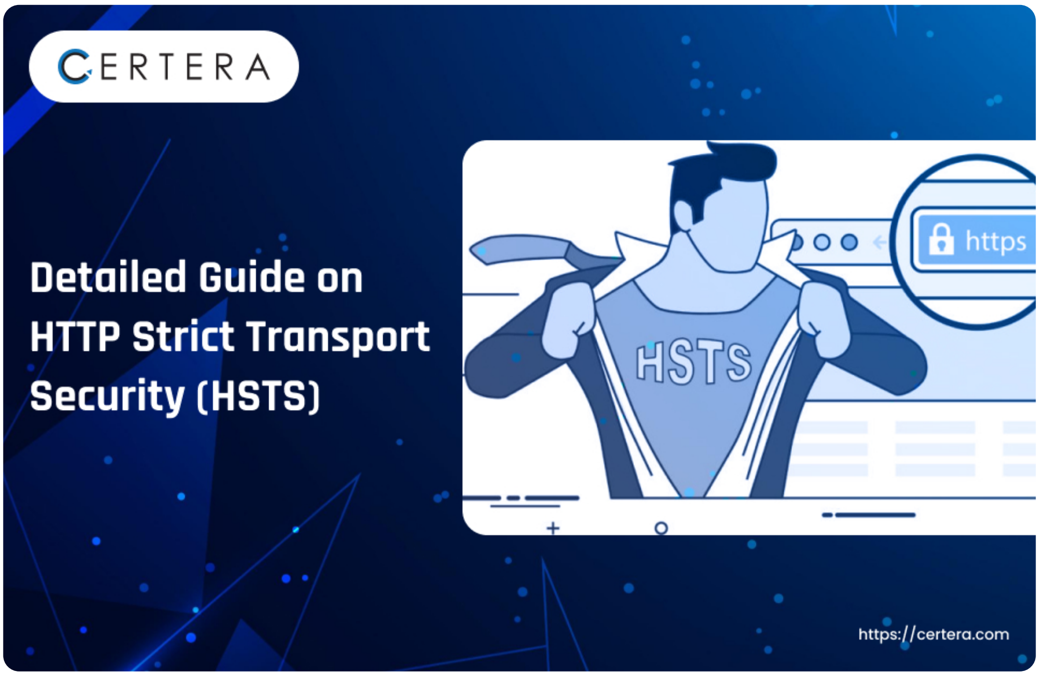
EncryptedFence
BY CERTERA

Home < https://certera.com/blog/ > » HSTS Explained – Detailed Guide on HTTP Strict Transport Security

HSTS Explained – Detailed Guide on HTTP Strict Transport Security

★★★★★ (**15** votes, average: **5.00** out of 5)



What is HSTS (HTTP Strict Transport Security)?

HTTP Strict Transport Security is commonly referred to by the acronym HSTS. Websites utilize this technique to indicate that they should only be viewed over secure connections (HTTPS).

A browser must reject all HTTP connections and stop users from accepting unsafe SSL certificates if a website states an HSTS policy. Most popular browsers presently support HSTS.

It is an approved and simple standard that makes sure users' browsers always connect to websites via HTTPS to maintain user security.

Purpose of HSTS

The primary purpose of HSTS is to eliminate the necessity for the widespread but unsafe practice of rerouting users from http:// to https:// URLs.

A web application can provide this opt-in security feature by using a specific response header. A compatible browser will send all communications over HTTPS instead of HTTP after receiving this header, preventing any communications from being routed over HTTP to the designated domain. It also prevents browsers from displaying HTTPS click-through prompts.

What is the HSTS's Background?

The **2008 publication ForcedHTTPS**: Protecting High-Security Web Sites against Network Attacks by Collin Jackson & Adam Barth formed the basis for the HSTS specification (RFC 6797).

PayPal, Jackon, & Barth published a revised version of the protocol described in the original study **on September 18, 2009**.

As a result, **on December 18, 2009**, the last "community version" of the formerly known "**STS**" specification was released, including changes made in response to comments from the community.

Subsequently, on **October 2, 2012**, the Internet Engineering Steering Group (IESG), which is made up of area directors and the chair of the Internet Engineering Task Force (IETF), accepted the **HSTS standard (RFC 6797**), which was finally published.

Why is the Usage of HSTS Significant?

How people visit webpages and how browsers handle manual URL inputs are both challenging. The browser will automatically change the URL protocol to **HTTP rather than HTTPS, ‹ https://certera.com/blog/http-vs-https-the-technical-difference/›** for instance, when a user fills in a web address like "sample.com" in the address bar.

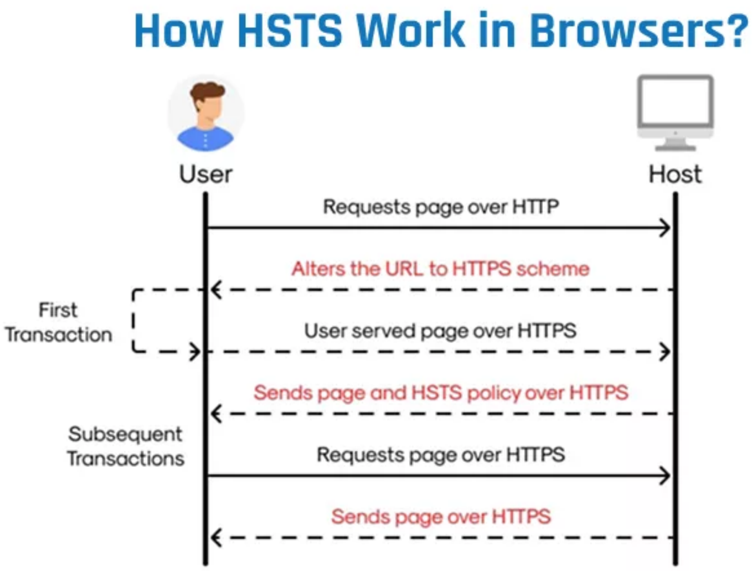
Users may also encounter hostile networking conditions where HTTPS connections might be rewritten to HTTP. Users could accidentally utilize an HTTP URL by clicking on outdated URLs.

Furthermore, some websites may still use HTTP ports to listen while rerouting visitors to HTTP URLs. This redirection is a hazardous technique that puts consumers at risk of **Man-in-the-Middle (MIM) ‹ https://certera.com/blog/man-in-the-middle-mitm-attacks-how-to-detect-and-prevent-it›** attacks and other intrusions like session-cookie hijacking.

How Does HSTS Work?

Web servers can specify through HSTS that all interactions between web browsers and other user agents must occur over HTTPS connections rather than unsecured HTTP connections.

By sending a response header over an HTTPS connection, a server can implement an HSTS Policy (HSTS headers delivered via HTTP response headers are ignored). The "Strict-Transport-Security" HSTS header designates a timeframe for the user agent to utilize HTTPS requests only to reach the service.


How HSTS Work in Browsers?
User
Host
Requests page over HTTP
Alters the URL to HTTPS scheme
First Transaction
User served page over HTTPS
Sends page and HSTS policy over HTTPS
Subsequent Transactions
Requests page over HTTPS
Sends page over HTTPS

This implies that when a website is reached using HTTPS for the first time, it returns the Strict-Transport-Security header. The browser stores this information and uses HTTPS the next time it tries to load the website via HTTP.

Instead of automatically switching to HTTPS, the next attempt to load the website over HTTP will proceed as usual when the Strict-Transport-Security headers' specified expiration period has passed.

However, the expiration time for that site will be updated anytime the Strict-Transport-Security header is given to the user agent. This allows sites to refresh this information and avoid the timeout expiring.

If web servers need to **disable HSTS ‹ https://certera.com/blog/how-to-disable-hsts-in-chrome-firefox›** , they can set the max-age to 0 (over an HTTPS connection), instantly expiring the HSTS header and providing access through HTTP queries.

For instance, a server could send a header with the following content: max-age=31536000, requesting that all requests made in the upcoming year exclusively utilize HTTPS via Strict-Transport-Security.

User agents that comply with an HSTS Policy issued by a web application act as follows:

> Any links that are not secure are immediately changed to secure ones (so, for instance, before gaining access to the server, http://sample1.com/ will change to https://sample1.com).
>
> The user agent will interrupt the connection and prevent the user from accessing the website if a secure connection is not established (for example, because the server lacks a valid certificate).

The primary point is that an HSTS Policy prevents click-through vulnerability by denying the end user access to the compromised connection.

Which Kind of Attacks is HSTS Resistant to?

The HSTS Policy protects users of web applications against various active and passive network attacks. **Let's now explore the most frequent risks that HSTS can reduce.**

Protocol Downgrade Attacks

Protocol downgrade is a kind of MIM attack when a hacker pushes customers to utilize less secure HTTP over HTTPS, which is a weaker protocol. Even if the user inputs an HTTP URL, HTST-enabled websites avoid protocol degradation by asking the browser to use HTTPS for subsequent requests.

Man-in-the-middle (MIM) and Certificate-based MIM attacks

HTTP domains don't use encryption while communicating over the network. As a result, there is a greater chance that man-in-the-middle attacks, which accomplish just that, will intercept the HTTP transmission and reroute the HTTP request to hostile websites.

Ensuring browser requests are encrypted, HSTS keeps outside parties from eavesdropping or intercepting communications.

Moreover, MIM attacks can also happen when an attacker provides users who are browsing insecure websites with an unauthorized certificate. The goal is to get the user to accept and trust the certificate, which might undermine security.

HSTS successfully prevents users from ignoring the alert regarding the invalid certificate.

Cookie Hijacking

Cookie hijacking is the act of an attacker attempting to mimic a user by gaining unauthorized access to the user's session cookies. By requiring HTTPS and ensuring that session cookies are transferred over encrypted, secure connections, HSTS assists in preventing such attacks.

Recommended: Session Hijacking: A Detailed Guide for Safeguarding Your Online Interactions < https://certera.com/blog/session-hijacking-a-detailed-guide-for-safeguarding-your-online-interactions>

The Advantages of HSTS

There are several benefits to HTTP Strict Transport Security (HSTS), including:

Enhanced Security:

By mandating HTTPS for all traffic, HSTS enhances security. Using HSTS guarantees that no unencrypted HTTP traffic is delivered because HSTS headers are only valid over HTTPS connections.

Mixed Content Defense:

When a domain has mixed content, HSTS automatically upgrades fetches to HTTPS

Improved Online Security:

HSTS makes enhanced speed, user experience, and trust possible.

Implementation Made Simpler:

HSTS makes implementation easier.

Cyberattack Defense:

HSTS assists in preventing several cyberattacks, such as session hijacking, SSL stripping, and other downgrade attacks.

Moreover, HSTS offers the following benefits:

- Switches all **HTTP to HTTPS < https://certera.com/blog/http-vs-https-the-technical-difference>** connections.
- Eliminates insecure connections.

Add a single line of code that appears next to the HSTS header and contains the term “preload” to be added to the preload list. Next, proceed to Google’s registration website and include yourself in the list. With each release of a new browser version, the HSTS preload list is updated.

What are the Best Practices for HSTS Deployment?

If you would like to preload HSTS and your website is dedicated to HTTPS, you need to:

Provide a legitimate certificate.

On the same host, redirect HTTP to HTTPS if you listen on **port 80 < https://certera.com/blog/port-80-http-vs-port-443-https-everything-to-know-about>** .

Use HTTPS to serve all subdomains.

If there is a DNS record for the www subdomain, you can provide HTTPS for that subdomain.

For HTTPS requests, the base domain can serve the HSTS header:

The maximum age requirement is 31536000 seconds or at least one year.

The directive for includeSubDomains must be provided.

It is required to specify the preload directive.

The HSTS header must remain on the extra redirection that you are providing from your HTTPS site (rather than the web page it redirects to)

Caution: Start with a modest max age until you can handle your site’s HTTPS configuration. If pages are not loading correctly via HTTPS later, users will be prevented from visiting your website until the issues are resolved.

Which Web Browsers Support HSTS?

Google Chrome and Chromium since version 4.0.211.0.

Since version 4 of Firefox, Mozilla has incorporated a list of websites that accept HSTS in Firefox 17.

Since version 12 of Opera.

On Windows 10, Microsoft Edge, and Internet Explorer 11.

WebView and the BlackBerry 10 Browser since BlackBerry OS 10.3.3

Safari since version 10.9 of OS X Mavericks.

With KB 3058515 installed, Internet Explorer 11 runs on Windows 8.1 and 7.

What are the Limitations of HSTS?

The following are some limitations placed on HTTP Strict Transport Security (HSTS):

Unavailability: The website is inaccessible to those who cannot connect over HTTPS.

Initial connection: A website is vulnerable to attacks since its first connection isn't secured by HSTS.

Malicious attacks: **SSL/TLS protocol ‹ https://certera.com/blog/what-is-ssl-tls-https/ ›** attacks are beyond the scope of HSTS protection. Additionally, it is defenseless against TLS and DNS-based attacks.

Privacy: HSTS could give rise to challenges with privacy

Lack of accessibility: A website becomes unavailable until the problem is fixed if its **SSL certificate ‹ https://certera.com/ ›** is incorrect or expires.

Browser preload list: Users who visit a website for the first time are susceptible to attacks if it isn't included in their browser's preload list.

FAQ's

What is HSTS Preloading?

An HTTPS-enforced website can only be accessed by preloading HSTS (HTTP Strict Transport Security). Firefox, Safari, and Chrome all utilize this list, which Google prepared.

Websites preloaded with HSTS preloading will always be accessed by web browsers over HTTPS. Among the preloaded websites in all popular web browsers is whitehouse.gov.

Over a secure HTTPS channel, the HSTS protocol compels a web connection. Your browser won't load a website if the SSL certificate isn't active.

Does HSTS Impact Website Performance?

Website performance is affected by HSTS (HTTP Strict Transport Security) to a substantial extent. On the other hand, others claim that if all mixed material is removed and everything is provided over HTTPS, HSTS can offer a marginal performance boost.

Only HTTPS connections are compatible with HSTS headers. This ensures that no HTTP communication is transmitted without encryption. By doing away with server redirection from HTTP to HTTPS, HSTS can also speed up page loads.

What information does an online HSTS tool provide about a website's HSTS configuration?

By notifying browsers that a website should only be viewed over HTTPS, an online HSTS tool might provide information on the HSTS configuration of a website.

What steps should I take if I encounter HSTS-related issues on my website?

Try the following if you experience HSTS-related challenges with your website:

Verify the SSL certificate you have.

Try using a different web browser.

Clear the HSTS cache < https://certera.com/blog/how-to-