What are HTTP security Headers?

- When we visit any website in the browser, the browser sends some request headers to the server and the server responds with HTTP response headers. These headers are used by the client and server to share information as a part of the HTTP protocol. Browsers have defined behavior of the web page according to these headers during communication with the server. These headers are mainly a combination of key-value pairs separated by a colon.
- Response headers that the server responds with to instruct the browser what security rules to enforce when it handles your website's content.
- In general, the more security headers you opt-in to sending, the more secure your website is.
- Most security headers come with multiple option you can configure to tweak the behavior to what you want.

Types of HTTP security Headers

1. HTTP Strict Transport Security (HSTS)
2. X-Frame-Options
3. Cross Site Scripting Protection (X-XSS

1. HTTP Strict Transport Security (HSTS)

HTTP Strict Transport Security (HSTS) is a web server directive that informs user agents and web browsers how to handle its connection through a response header sent at the very beginning and back to the browser.

This sets the Strict-Transport-Security policy field parameter. It forces those connections over HTTPS encryption, disregarding any script's call to load any resource in that domain over HTTP. HSTS is but one arrow in a bundled sheaf of security settings for your web server or your web hosting service.

How works HSTS

There are semantically distinct ways to send HSTS headers, as defined in RFC 6797:

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

The HSTS policy is applied only to the domain of HSTS host issuing it and remains in effect for one year.

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

The HSTS policy is applied to the domain of the issuing host as well as its subdomains and remains in effect for one year.

- *Strict-Transport-Security: max-age=0*

Directs the browser to delete the entire HSTS policy.

Why important

- **It will guide the browser**

  The HSTS security header is important because it will guide the browser to use secure connection using with HTTPS protocol when establishing a connection.

- **Prevent some classes of man-in-the-middle (MITM) attacks**

2. X-Frame option

In the Orkut era, a spoofing technique called 'Clickjacking' was pretty popular. It still is. In this technique, an attacker fools a user into clicking something that isn't there. For example, a user might think that he's on the official Orkut website, but something else is running in the background. A user may reveal his/her confidential information in the process.

X-Frame-Options help guard against these kinds of attacks. This is done by disabling the iframes present on the site. In other words, it doesn't let others embed your content

Why Important

- Prevent click-jacking attacks
- It allows content publishers to prevent their own content

  X-Frame-Options allows content publishers to prevent their own content from being used in an invisible frame by attackers.

Disadvantages

- To enable the SAMEORIGIN option across a website, the X-Frame-Options header needs to be returned as part of the HTTP response for each individual page (cannot be applied cross-site).
- X-Frame-Options does not support a whitelist of allowed domains, so it doesn't work with multi-domain sites that need to display framed content between them.
- Only one option can be used on a single page, so, for example, it is not possible for the same page to be displayed as a frame both on the current website and an external site.
- The ALLOW-FROM option is not supported by all browsers.
- X-Frame-Options is a deprecated option in most browsers.

3. Cross Site Scripting Protection (XSS)

As the name suggests, X-XSS header protects against Cross-Site Scripting attacks. XSS Filter is enabled in Chrome, IE, and Safari by default. This filter doesn't let the page load when it detects a cross-site scripting attack.

How works XSS

Cross-site scripting works by manipulating a vulnerable web site so that it returns malicious JavaScript to users. When the malicious code executes inside a victim's browser, the attacker can fully compromise their interaction with the application.

Why important XSS

XSS is a versatile attack vector which opens the door to a large number of social-engineering and client-side attacks. As shown, it could be used to steal sensitive information, such as session tokens, user credentials or commercially valuable data, as well as to perform sensitive operations.

Advantages

- Free Usually have a narrow focus, but do the job well Wide variety to choose from: you can use many simultaneously

Disadvantages

Low to medium quality in most cases No centralized support May integrate poorly (or not at all) with other security tools

How HTTP Security Headers Can Improve Web Application Security

When we talk about web application security, especially on this blog, we usually mean finding exploitable vulnerabilities and fixing them in application code. HTTP security headers provide an extra layer of security by restricting behaviors that the browser and server allow once the web application is running. In many cases, implementing the right headers is a crucial aspect of a best-practice application setup – but how do you know which ones to use?

As with other web technologies, HTTP headers come and go depending on browser vendor support. Especially in the field of security, headers that were widely supported a few years ago can already be deprecated. At the same time, completely new proposals can gain universal support in a matter of months. Keeping up with all these changes is not easy.