Geek Flare

How to Implement Security HTTP Headers to Prevent Vulnerabilities

By **Chandan Kumar** / in **Apache HTTP (https://geekflare.com/category/web-infrastructure/apache/)**,**IHS/IIS (https://geekflare.com/category/web-infrastructure/ihs/)**,**Nginx (https://geekflare.com/category/web-infrastructure/nginx/)**,**Security (https://geekflare.com/category/security/)** / Updated: February 18, 2018

773 SHARES

Do you know most the security vulnerabilities can be fixed by implementing necessary headers in response header?

Security is as important as content and SEO of your website, and thousands of website get hacked (https://geekflare.com/real-time-cyber-attacks/) due to misconfiguration or lack of protection.

If you are a website owner or security engineer and looking to protect your website (http://sucuri.7eer.net/c/245992/212721/3713?u=https%3A%2F%2Fsucuri.net%2Fwebsite-security-platform%2F) from **Clickjacking, code injection, MIME types, XSS**, etc. attacks then this guide will help you.

In this article, I will talk about various HTTP Header to implement in multiple web servers, network edge & CDN provider for better website protection (https://geekflare.com/web-application-firewall/).

Notes:

- You are advised to take a backup of configuration file prior making changes
- Some of the headers may not be supported on all the browsers so check out the compatibility (https://caniuse.com/) before the implementation.
- Mod_headers must be enabled in Apache to implement these headers. Ensure the following line uncommented in httpd.conf file.
- LoadModule headers_module modules/mod_headers.so

If you are using SUCURI Cloud WAF (http://sucuri.7eer.net/c/245992/212721/3713?u=https%3A%2F%2Fsucuri.net%2Fwebsite-firewall%2F), then you don't have to worry about adding these manually on your web server as most of them are automatically enabled.

HTTP Headers List

X-XSS-Protection

X-XSS-Protection header can prevent some level of **XSS** (cross-site-scripting) attacks, and this is compatible with IE 8+, Chrome, Opera, Safari & Android.

Google, Facebook, Github use this header, and most of the penetration testing consultancy will ask you to implement this.

There are four possible ways you can configure this header.

Parameter Value	Meaning
0	XSS filter disabled
1	XSS filter enabled and sanitized the page if attack detected
1;mode=block	XSS filter enabled and prevented rendering the page if attack detected
1;report=http://example.com/report_URI	XSS filter enabled and reported the violation if attack detected

Let’s implement **1;mode=block** in the following web servers.

Apache HTTP Server

Add the following entry in httpd.conf of your apache web server

```
Header set X-XSS-Protection "1; mode=block"
```

Restart the apache to verify

Nginx

Add the following in nginx.conf under http block

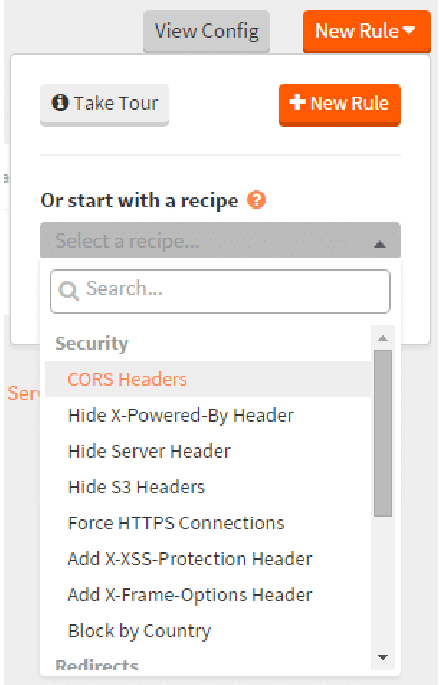
```
add_header X-XSS-Protection "1; mode=block";
```

Nginx restart is needed to get this reflected on your web page response header.

MaxCDN

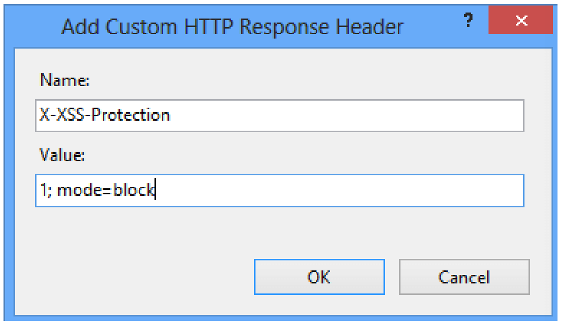
If you are using MaxCDN (https://tracking.maxcdn.com/c/245992/3982/378) then adding header is easy and on-the-fly.

Go to Edge Rules >> click "New Rule" and select "Add X-XSS-Protection Header" from the drop-down.



Microsoft IIS

- Open IIS Manager
- Select the Site you need to enable the header for
- Go to "HTTP Response Headers."
- Click "Add" under actions
- Enter name, value and click Ok



X-XSS-Protection

1; mode=block

- Restart IIS to see the results

HTTP Strict Transport Security

HSTS (HTTP Strict Transport Security) header ensure all communication from a browser is sent over HTTPS (HTTP Secure). This prevents HTTPS click through prompts and redirects HTTP requests to HTTPS.

Before implementing this header, you must ensure all your website page is accessible over HTTPS else they will be blocked.

HSTS header is supported on all the major latest version of a browser like IE, Firefox, Opera, Safari, and Chrome. There are three parameters configuration.

Parameter Value	Meaning
max-age	Duration (in seconds) to tell a browser that requests are available only over HTTPS.
includeSubDomains	Configuration is valid for subdomain as well.
preload	Use if you would like your domain to be included in the HSTS preload list (https://hstspreload.appspot.com/)

So let’s take an example of having HSTS configured for one year including preload for domain and sub-domain (https://geekflare.com/find-subdomains/).

Apache HTTP Server

You can implement HSTS in Apache by adding the following entry in httpd.conf file

```
Header set Strict-Transport-Security "max-age=31536000; includeSubDomains; preload"
```

Restart apache to see the results

Nginx

To configure HSTS in Nginx, add the following entry in nginx.conf under server (ssl) directive

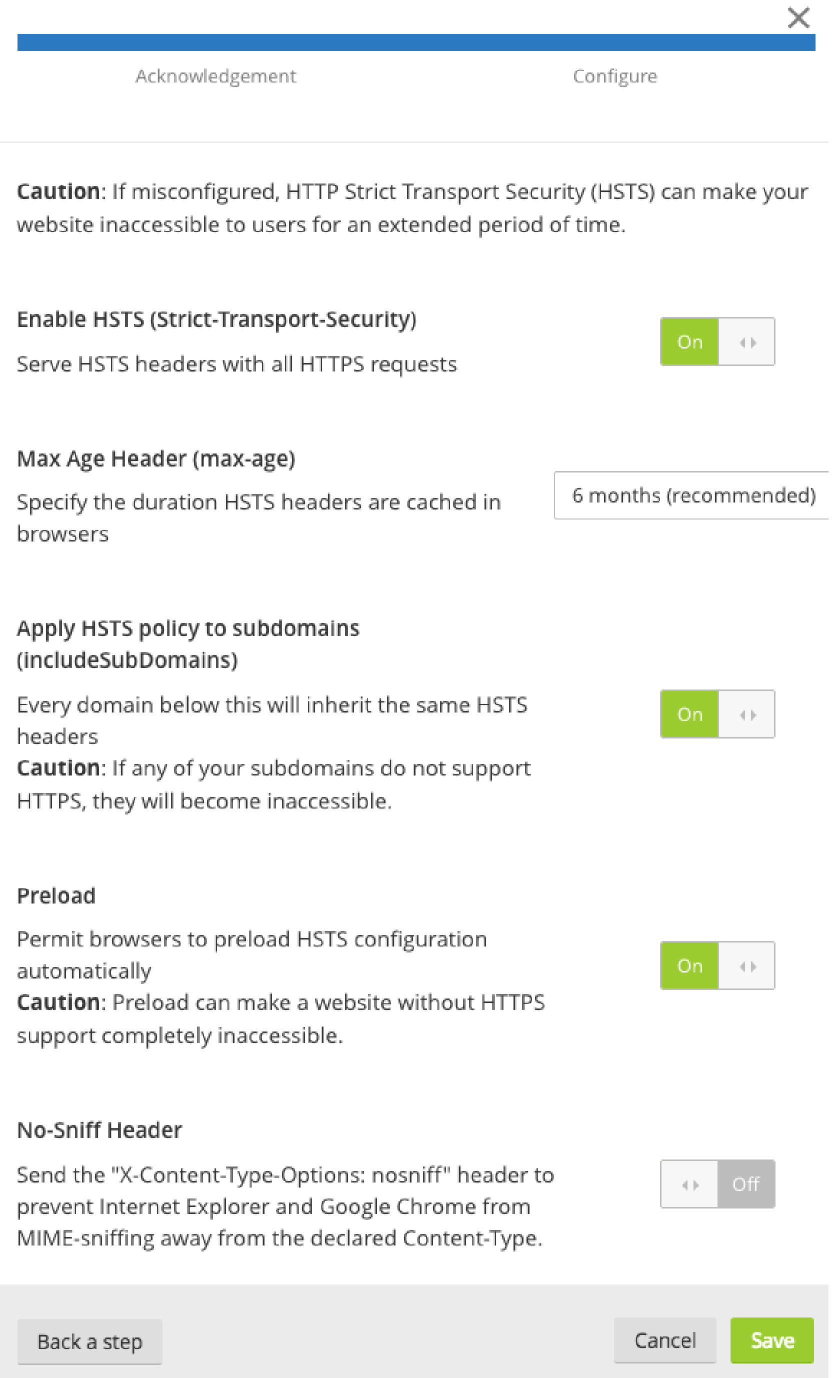
```
add_header Strict-Transport-Security 'max-age=31536000; includeSubDomains; preload';
```

As usual, you will need to restart Nginx to verify

Cloud Flare

If you are using Cloud Flare, then you can enable HSTS in just a few clicks.

- Log in to Cloud Flare (https://www.cloudflare.com) and select the site
- Go to “Crypto” tab and click “Enable HSTS.”



Caution: If misconfigured, HTTP Strict Transport Security (HSTS) can make your website inaccessible to users for an extended period of time.

Enable HSTS (Strict-Transport-Security) On

Serve HSTS headers with all HTTPS requests

Max Age Header (max-age) 6 months (recommended)

Specify the duration HSTS headers are cached in browsers

Apply HSTS policy to subdomains (includeSubDomains) On

Every domain below this will inherit the same HSTS headers
Caution: If any of your subdomains do not support HTTPS, they will become inaccessible.

Preload On

Permit browsers to preload HSTS configuration automatically
Caution: Preload can make a website without HTTPS support completely inaccessible.

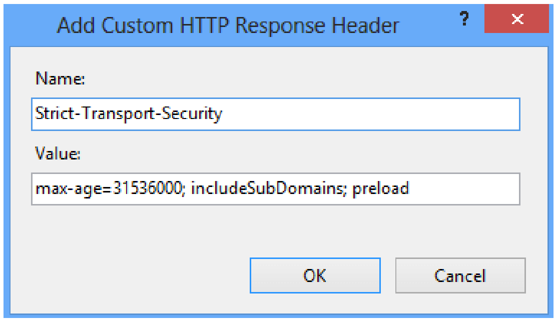
No-Sniff Header Off

Send the "X-Content-Type-Options: nosniff" header to prevent Internet Explorer and Google Chrome from MIME-sniffing away from the declared Content-Type.

Select the settings the one you need and changes will be applied on the fly.

Microsoft IIS

Launch the IIS Manager and add the header by going to “HTTP Response Headers” for respective site.



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

Restart the site

X-Frame-Options

Use X-Frame-Options header to prevent **Clickjacking** vulnerability on your website. By implementing this header, you instruct the browser not to embed your web page in frame/iframe. This has some limitation in browser support, so you got to check before implementing it.

You can configure the following three parameters.

Parameter Value	**Meaning**
SAMEORIGIN	Frame/iframe of content is only allowed from the same site origin.
DENY	Prevent any domain to embed your content using frame/iframe.
ALLOW-FROM	Allow framing the content only on particular URI.

Let's take a look how to implement "**DENY**" so no domain embeds the web page.

Apache

Add the following line in httpd.conf and restart the web server to verify the results.

```
Header always append X-Frame-Options DENY
```

Nginx

Add the following in nginx.conf under server directive/block.

add_header X-Frame-Options “DENY”;

Restart to verify the results

F5 LTM

Create an iRule with the following and associated with the respective virtual server.

```
when HTTP_RESPONSE {

HTTP::header insert "X-FRAME-OPTIONS" "DENY"

}
```

You don’t need to restart anything, changes are reflected in the air.

WordPress

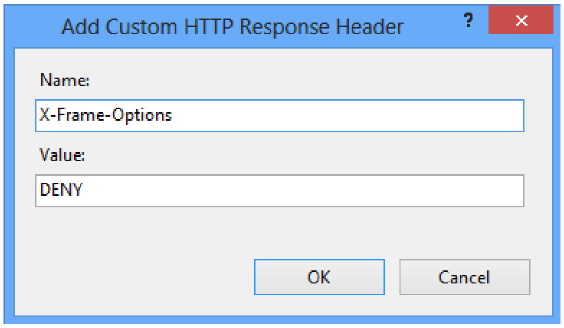
You can get this header implemented through WordPress too. Add the following in wp-config.php file

```
header('X-Frame-Options: DENY);
```

If you are not comfortable editing the file, then you can use a plugin as explained here (https://geekflare.com/wordpress-x-frame-options-httponly-cookie/).

Microsoft IIS

Add the header by going to “HTTP Response Headers” for respective site.



Name: X-Frame-Options

Value: DENY

Restart the site to see the results.

X-Content-Type-Options

Prevent **MIME** types security risk by adding this header to your web page's HTTP response. Having this header instruct browser to consider files types as defined and disallow content sniffing. There is only one parameter you got to add "nosniff".

Let's see how to advertise this header.

Apache

You can do this by adding the below line in httpd.conf file

```
Header set X-Content-Type-Options nosniff
```

Don't forget to restart the Apache web server to get the configuration effective.

Nginx

Add the following line in nginx.conf file under server block.

```
add_header X-Content-Type-Options nosniff;
```

As usual, you got to restart the Nginx to check the results.

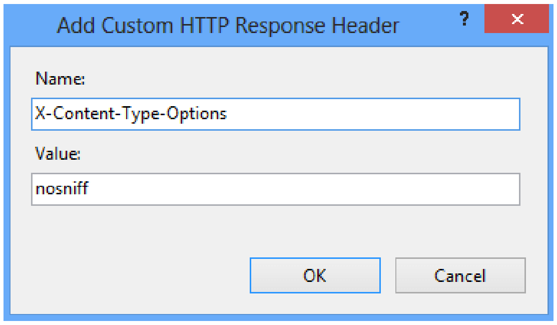
WordPress

If you are using the WordPress, then you may consider using Security Headers (https://wordpress.org/plugins/security-headers/) plugin to implement this header.

Microsoft IIS

Open IIS and go to HTTP Response Headers

Click on Add and enter the Name and Value



Name:
X-Content-Type-Options
Value:
nosniff

Click OK and restart the IIS to verify the results.

HTTP Public Key Pinning

Minimize the man-in-the-middle (**MITM**) attacks risk by pinning certificate. This is possible with HPKP (https://www.owasp.org/index.php/OWASP_Secure_Headers_Project#Public_Key_Pinning_Extension_for_HTTP_.28HPKP.29) (HTTP Public Key Pinning) header.

You can pin the root certificate public key or immediate certificate. At the time of writing, HPKP currently works in Firefox and Chrome and support SHA-256 (https://geekflare.com/test-ssl-sha1-vulnerability-and-fix/) hash algorithm.

There are four possible parameter configurations.

Parameter Value	Meaning
report-uri=”url”	Report to the specified URL if pin validation fails. This is optional.
pin-sha256=”sha256key”	Specify the pins here
max-age=	Browser to remember the time in seconds that site is accessible only using one of the pinned keys.
IncludeSubDomains	This is applicable on a subdomain as well.

Let’s see HPKP header example from facebook.com

```
public-key-pins-report-only:max-age=500; pin-sha256="WoiWRyIOVNa9ihaBciRSC7XHjliYS9VwUGOI
```

If this is something you need to implement on your website, then head to implementation guide written by Scott Helme (https://scotthelme.co.uk/hpkp-http-public-key-pinning/).

Content Security Policy

Prevent XSS, clickjacking, **code injection** attacks by implementing Content Security Policy (CSP) header in your web page HTTP response. CSP (https://content-security-policy.com/) instruct browser to load allowed content to load on the website.

All browsers don't support CSP (https://caniuse.com/#feat=contentsecuritypolicy2), so you got to verify before implementing it. There are three ways you can implement CSP headers.

1. Content-Security-Policy – Level 2/1.0
2. X-Content-Security-Policy – Deprecated
3. X-Webkit-CSP – Deprecated

If you are still using deprecated one, then you may consider upgrading to the latest one.

There are multiple parameters possible to implement CSP, and you can refer OWASP (https://www.owasp.org/index.php/OWASP_Secure_Headers_Project#Content-Security-Policy) for an idea. However, let's go through two most used parameters.

Parameter Value	Meaning
default-src	Load everything from defined source
script-src	Load only scripts from defined source

The following example to load everything from the same origin in various web servers.

Apache

Get the following added in httpd.conf file and restart web server to get effective.

```
Header set Content-Security-Policy "default-src 'self';"
```

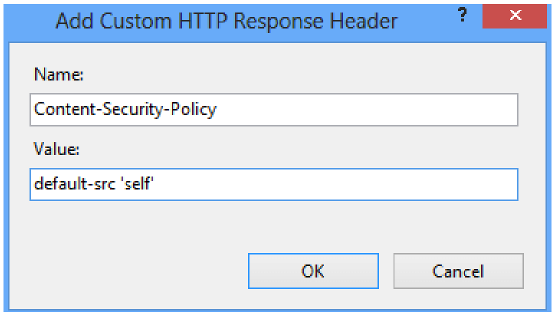
Nginx

Add the following in server block in nginx.conf file

```
add_header Content-Security-Policy "default-src 'self';";
```

Microsoft IIS

Go to HTTP Response Headers for your respective site in IIS Manager and add the following



Name:
Content-Security-Policy
Value:
default-src 'self'

I hope above headers implementation instruction helps to make your web application **secure and safer**. If you are looking to secure IIS web server, then you may also consider WebKnight WAF (https://geekflare.com/webknight-iis-waf/) where you can configure above all and more than that.

773
SHARES

About Chandan

Chandan Kumar is the founder and editor of Geek Flare. Learn more here (https://geekflare.com/about/) and connect with him on Twitter.

(https://twitter.com/ConnectCK)

You'll also like:

(https://geekflare.com/html5-changing-web-security/)

How HTML5 is Changing Web Security? (https://geekflare.com/html5-changing-web-security/)

(https://geekflare.com/tomcat-ssl-guide/)

How to Implement SSL in Apache Tomcat? (https://geekflare.com/tomcat-ssl-guide/)

Comments

Rex *says*
DECEMBER 14, 2017 AT 7:22 AM (HTTPS://GEEKFLARE.COM/HTTP-HEADER-IMPLEMENTATION/#COMMENT-46867)

Awesome article. One typo here. Under “X-XSS-Protection”=>”Apache HTTP Server”
Header set X-XSS-Protection “1; mode=block”
Should be
Header set X-XSS-Protection “1; mode=block”
The difference in double quotation marks will cause syntax error.

Regards

REPLY (HTTPS://GEEKFLARE.COM/HTTP-HEADER-IMPLEMENTATION/?REPLYTOCOM=46867#RESPOND)

> CHANDAN KUMAR (HTTPS://CHANDAN.IO/) *says*
> DECEMBER 14, 2017 AT 1:41 PM (HTTPS://GEEKFLARE.COM/HTTP-HEADER-IMPLEMENTATION/#COMMENT-46879)
>
> Thanks Rex
>
> REPLY (HTTPS://GEEKFLARE.COM/HTTP-HEADER-IMPLEMENTATION/?REPLYTOCOM=46879#RESPOND)