Prevent Vulnerabilities?

(HTTPS://GEEKFLARE.COM/AUTHOR/CHANDAN/)

BY CHANDAN KUMAR | DECEMBER 25, 2018 | SECURITY (HTTPS://GEEKFLARE.COM/CATEGORY/SECURITY/)

Do you know most the security vulnerabilities can be fixed by implementing necessary headers in the 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/).

Geekflare

Notes:

before the implementation.

- 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
  - Apache HTTP Server
  - Nginx
  - MaxCDN
  - Microsoft IIS
- HTTP Strict Transport Security
  - Apache HTTP Server
  - Nginx
  - Cloud Flare
  - Microsoft IIS
- X-Frame-Options
  - Apache
  - Nginx
  - F5 LTM
  - WordPress
  - Microsoft IIS
- X-Content-Type-Options
  - Apache
  - Nginx
  - WordPress
  - Microsoft IIS
- HTTP Public Key Pinning
- Content Security Policy
  - Apache
  - Nginx
  - Microsoft IIS

Apache
Nginx

Nginx

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

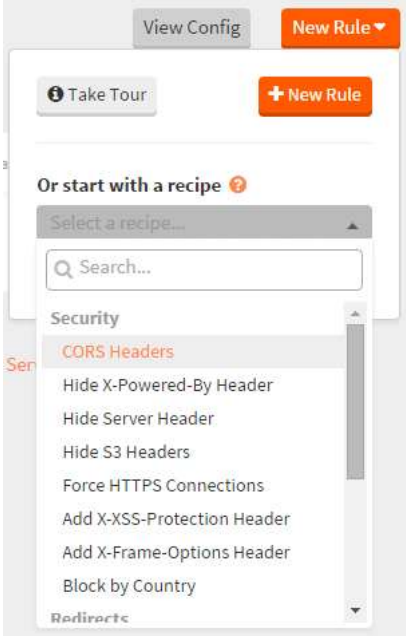
Nginx

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


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

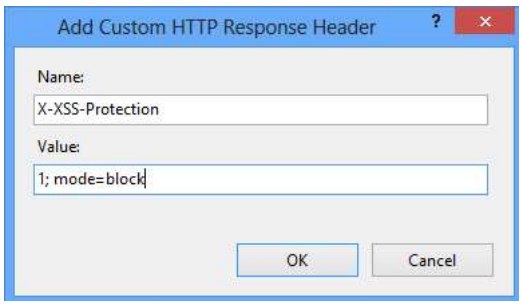
If you are using MaxCDN (https://tracking.maxcdn.com/c/245992/3982/378) then adding header is easy and on-the-fly.

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



Geekflare

- Restart IIS to see the results

HSTS (HTTP Strict Transport Security) header to ensure all communication from a browser is sent over HTTPS (HTTP

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

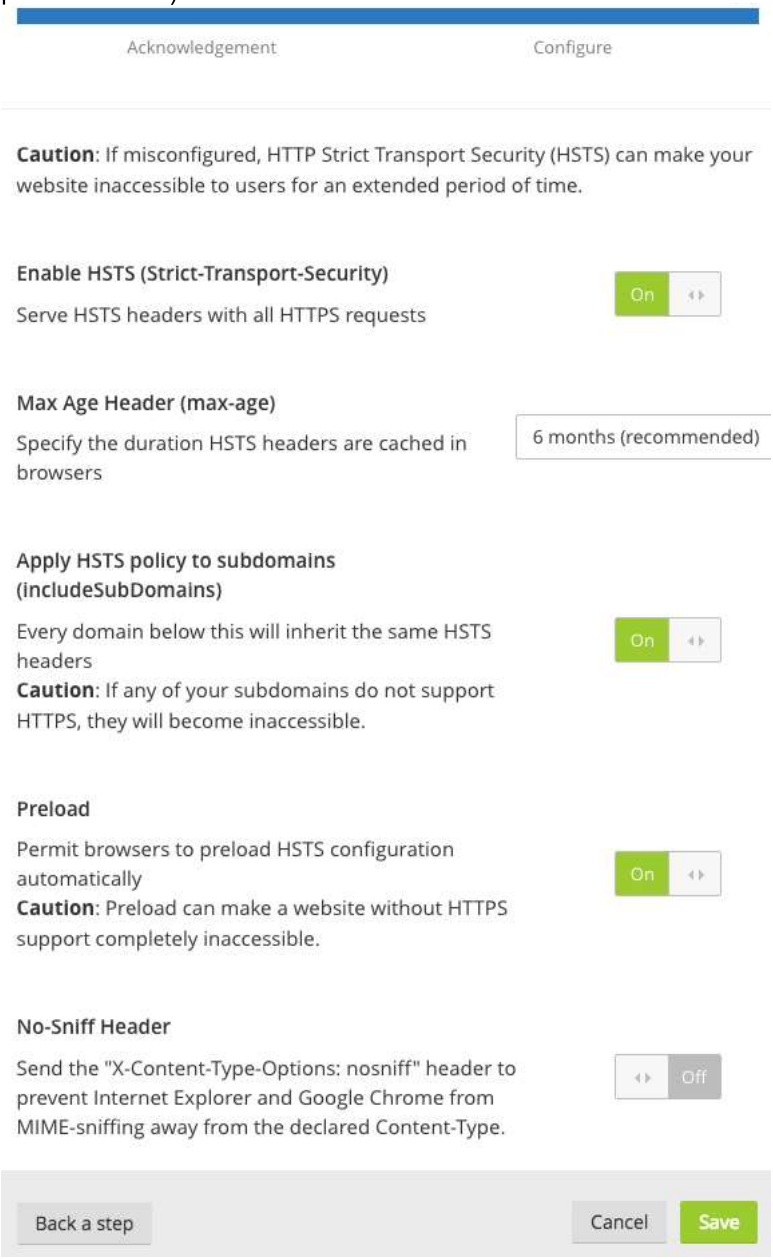
To configure HSTS in Nginx, add the next entry in nginx.conf under server (SSL) directive

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

As usual, you will need to restart Nginx to verify

Cloud Flare

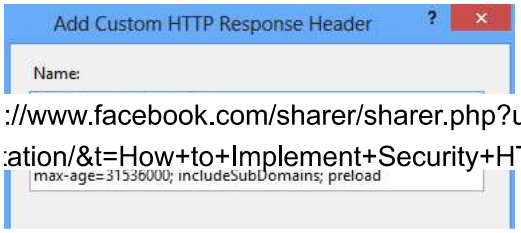
- Log in to Cloud Flare (https://www.cloudflare.com) and select the site



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 the respective site.



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 at 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


```
  when HTTP_RESPONSE {
```



```
  }
```

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

WordPress

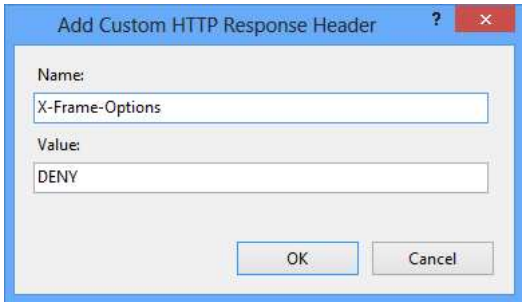
You can get this header implemented through WordPress too. Add the following in a wp-config.php file

```
  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.



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.

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

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

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.

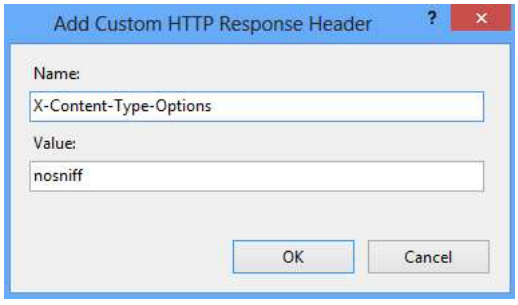
WordPress

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



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_.28HP

You can pin the root certificate public key or immediate certificate. At the time of writing, HPKP currently works in Firefox

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="WoiWRyIOVNa9ihaBciRSC7XHjliYS9VwUGOIud4PB
```

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 the 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 achieve 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,

Parameter Value	Meaning
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 the web server to get effective.

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

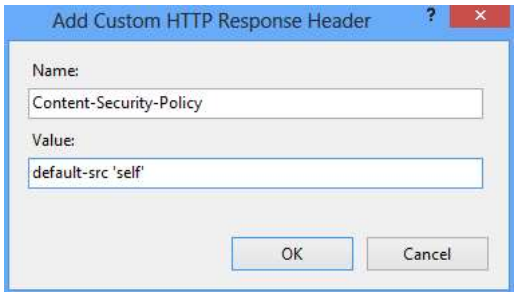
Nginx

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



X-Permitted-Cross-Domain-Policies

Using Adobe products like PDF, Flash, etc.? You can implement this header to instruct browser how to handle the requests over a cross domain. By implementing this header, you restrict loading your site's assets from other domain to avoid

There are a few options available.

none	no policy is allowed
master-only	allow only the master policy
all	everything is allowed
by-content-only	Allow only a certain type of content. Example – XML
by-ftp-only	applicable only for an FTP server

Apache

If you don't want to allow any policy.

```
Header set X-Permitted-Cross-Domain-Policies "none"
```

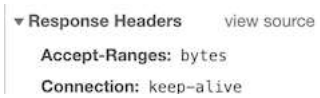
You should see the header like the following.

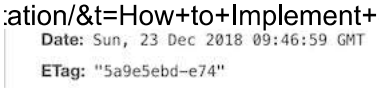
Nginx

And, let's say you need to implement master-only then add the following in `nginx.conf` under `server` block.

```
add_header X-Permitted-Cross-Domain-Policies master-only;
```

and the result.





Referrer Policy

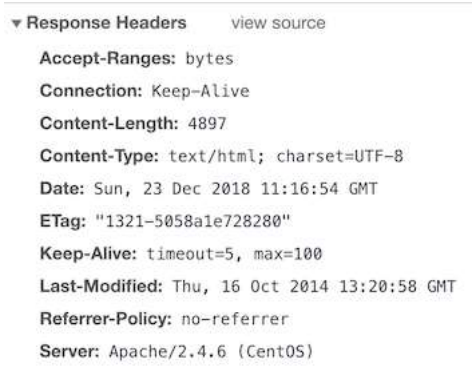
Looking to control the referrer policy of your site? There are certain privacy and security benefits. However, not all the options are supported by all the browsers so review your requirement before the implementation.

Referrer-Policy supports the following syntax.

Value	Description
no-referrer	Referrer information will not be s ent with the request.
no-referrer-when-downgrade	the default setting where referrer is sent to the same protocol like H TTP to HTTP, HTTPS to HTTPS.
unsafe-url	full URL will be sent with the req uest.
same-origin	referrer will be sent only for same origin site.
strict-origin	send only when a protocol is HTT PS
strict-origin-when-cross-origin	the full URL will be sent over a str ict protocol like HTTPS
origin	send the origin URL in all the req uests
origin-when-cross-origin	send FULL URL on the same origi n. However, send only origin URL

Apache


```
Header set Referrer-Policy "no-referrer"
```

Nginx

Let's say you need to implement same-origin so you got to add the following.

```
add_header Referrer-Policy same-origin;
```

Once configured, you should have the results as below.

Expect-CT

A new header still in experimental status is to instruct the browser to validate the connection with web servers for certificate transparency (CT). This project by Google aims to fix some of the flaws in the SSL/TLS certificate (https://geekflare.com/ssl-test-certificate/) system.

The following three variables are available for Expect-CT header

Value	Description
enforce	An optional directive to enforce the policy.

Apache

Let's assume you want to enforce this policy, report, and cache for 12 hours then you got to add the following.

```
Header set Expect-CT 'enforce, max-age=43200, report-uri="https://somedomain.com/report"'
```

And, here is the result.

Nginx

What if you want to report and cache for 1 hour?

```
add_header Expect-CT 'max-age=60, report-uri="https://mydomain.com/report"';
```

Output would be.

Securing a website is challenging, and I hope by implementing above headers you add a layer of security. If you are running a business site, then you may also consider using cloud-WAF like SUCURI

your online business.

More Articles for you!

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

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

(https://geekflare.com/securing-smartwatch/)

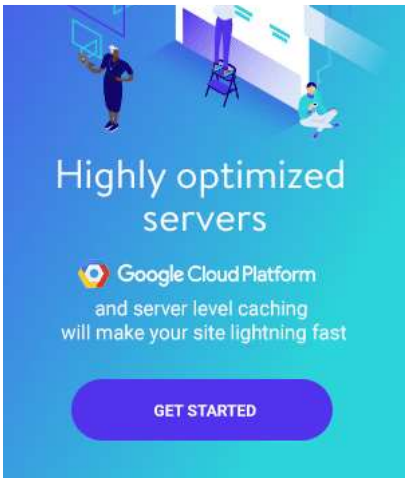
Four Essential Tips to Secure Smartwatch
(https://geekflare.com/securing-smartwatch/)

About Chandan Kumar

Since founded Geekflare, I've helped millions of professional through my articles and tools.

Let's connect on Twitter (https://twitter.com/ConnectCK) or LinkedIn (https://www.linkedin.com/in/chandank/).

Geekflare



Comments

dhillibabu *says*
JULY 18, 2018 AT 1:01 AM (HTTPS://GEEKFLARE.COM/HTTP-HEADER-IMPLEMENTATION/#COMMENT-55577)

Hi

after implementing this.. i am getting below error

“to use this site, first enable your browsers java script support and then refresh this page” kindly advice.

REPLY

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

CHANDAN KUMAR (HTTPS://CHANDAN.IO/) *says*
DECEMBER 14, 2017 AT 1:41 PM (HTTPS://GEEKFLARE.COM/HTTP-HEADER-IMPLEMENTATION/#COMMENT-46879)

Thanks Rex

Geekflare