HTTP SECURITY HEADERS

(Protection For Browsers)
BIO

• Emmanuel JK Gbordzor

ISO 27001 LI, CISA, CCNA, CCNA-Security, ITILv3, ...

11 years in IT – About 2 years In Security
Information Security Manager @ PaySwitch
Head, Network & Infrastructure @ PaySwitch
Head of IT @ Financial Institution

Bug bounty student by night – 1st Private Invite on HackerOne
In this presentation, I will introduce you to HyperText Transfer Protocol (HTTP) response security headers. By specifying expected and allowable behaviors, we will see how security headers can prevent a number of attacks against websites. I’ll explain some of the different HTTP response headers that a web server can include in a response, and what impact they can have on the security of the web browser. How web developers can implement these security headers to make user experience more secure.
A Simple Look At Web Browsing
Snippets at the Request and Response Headers

**Request Line**
GET /doc/test.html HTTP/1.1
Host: www.test101.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0
Content-Length: 35

bookId=12345&author=Tan+Ah+Teck

**Request Headers**

**Request Message Body**

**Response Line**
HTTP/1.1 200 OK
Date: Sun, 08 Feb xxxx 01:11:12 GMT
Server: Apache/1.3.29 (Win32)
Last-Modified: Sat, 07 Feb xxxx
ETag: "0-23-4024c3a5"
Accept-Ranges: bytes
Content-Length: 35
Connection: close
Content-Type: text/html

<h1>My Home page</h1>

**Response Headers**

**Response Message Body**
Browser Security Headers help:

➢ to define whether a set of security precautions should be activated or deactivated on the web browser.
➢ to reinforce the security of your web browser to fend off attacks and to mitigate vulnerabilities.
➢ in fighting client side (browser) attacks such as clickjacking, injections, Multipurpose Internet Mail Extensions (MIME) sniffing, Cross-Site Scripting (XSS), etc.
HTTP Strict Transport Security (HSTS)

• HSTS header forces browsers to communicate using secure (HTTPS) connection.
• Protects against “downgrade attacks”
• When configured with the “Preload” option, it can prevent Man-In-The-Middle (MiTM) attack
• “Preload” - https://hstspreload.org/ - from google
HTTP Redirection To HTTPS

Request URL: http://apple.com/
Request Method: GET
Remote Address: 17.178.96.59:80
Status Code: 301 MOVED PERMANENTLY

Version: HTTP/1.1

Response Headers (165 B)
- Connection: close
- Content-type: text/html
- Date: Mon, 02 Mar 2020 09:38:04 GMT
- Location: https://www.apple.com/
- Server: Apache

Request URL: https://www.apple.com/
Request Method: GET
Remote Address: 23.62.140.52:443
Status Code: 200 OK

Version: HTTP/2

Response Headers (530 B)
- cache-control: max-age=186
- content-encoding: gzip
- content-length: 10249
- content-type: text/html; charset=UTF-8
HTTP Redirection To HTTPS - Continued

- date: Mon, 02 Mar 2020 09:38:05 GMT
- expires: Mon, 02 Mar 2020 09:41:11 GMT
- server: Apache
- set-cookie: geo=GH; path=/; domain=.apple.com
- set-cookie: ccl=gbJOi8kj+ktBnVV4fLEw==; path=/; domain=.apple.com
- strict-transport-security: max-age=31536000; includeSubDomains
- vary: Accept-Encoding
- x-content-type-options: nosniff
  X-Firefox-Spdy: h2
- x-frame-options: SAMEORIGIN
- x-xss-protection: 1; mode=block
HTTP Strict Transport Security (HSTS) - Implementation

Syntax:
Strict-Transport-Security: max-age=<expire-time>
includedSubDomains
preload

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

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

Microsoft IIS:
Name: Strict-Transport-Security
Value: max-age=31536000; includeSubDomains; preload
X-Frame-Options

- An iFrame is an element that allows a web app to be nested within a parent web app.
- Can be used maliciously for a clickjacking attack or loading a malicious website inside the frame

Prevention:
- Frame busting
- X-Frame-Option Header
X-Frame-Options - Implementation

Syntax:

X-Frame-Options: deny
sameorigin
allow-from url (deprecated)

Apache:
Header always set X-Frame-Options “deny”

Nginx:
add_header X-Frame-Options “DENY”;

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

Microsoft IIS:
Name: X-Frame-Options
Value: DENY
Expect-CT

- HTTP Public Key Pinning (HPKP) header is being deprecated to Expect-CT
- Expect-CT detects certificates issued by rogue Certificate Authorities (CA) or prevents them from doing so
- This header prevents MiTM attack against compromised Certificate Authority (CA) and rogue issued certificate
Expect-CT - Implementation

Syntax:

    Expect-CT: max-age enforce report-uri

Apache:

    Header set Expect-CT 'enforce, max-age=86400, report-uri="https://foo.example/report"'

Nginx:

    add_header Expect-CT 'max-age=60, report-uri="https://mydomain.com/report"';
Content-Security-Policy (CSP)

This header helps you to whitelist sources of approved content into your browser hence, preventing the browser from loading malicious assets.

This helps prevents XSS, clickjacking, code injection, etc., attacks.

When this header is well implemented, there is no need to implement “X-Frame-Options” and “X-XSS-Protection” headers.
Content-Security-Policy - Directives

Keywords: *, none, self, hosts

Content-Security-Policy:

default-src  Serves as a fallback for the other fetch directives
font-src    Specifies valid sources for fonts loaded
frame-src   Sources for nested contexts such as <frame> and <iframe>
img-src     Sources of images and favicons
media-src   Valid sources for loading <audio>, <video> & <track>
object-src  Sources for the <object>, <embed> and <applet> elements
script-src  Specifies valid sources for JavaScript
style-src   Specifies valid sources for stylesheets
report-uri Reports violations
CSP Sample - https://haveibeenpwned.com

www.google-analytics.com stats.g.doubleclick.net www.gstatic.com;font-src 'self' cdnjs.cloudflare.com fonts.gstatic.com;base-uri 'self';child-src
X-XSS-Protection

These headers detect dangerous HTML input and either prevent the site from loading or remove potentially malicious scripts.
X-XSS-Protection - Implementation

Syntax:

```
X-XSS-Protection: 0
   1
   mode=block
```

Apache:

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

Nginx:

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

Microsoft IIS:

Name: X-XSS-Protection
Value: 1; mode=block
X-Content-Type-Options

• For your seamless experience on the web, MIME sniffing of resource was introduced.
• Adversely, an attacker can introduce a malicious executable script such as an image. When acted on by MIME sniffing could have the script executed.
X-Content-Type-Options - Implementation

Syntax:

X-Content-Type-Options: nosniff

Apache:
Header set X-Content-Type-Options nosniff

Nginx:
add_header X-Content-Type-Options nosniff;

Microsoft IIS:
Name: X-Content-Type-Options
Value: nosniff
Demo Time

- Clickjacking
- iFrame injection
- Harlem shake

https://127.0.0.1/mutillidae/
Takeaways

• Enforce HTTPS using the Strict-Transport-Security header and add your domain to Chrome’s preload list.
• Make your web app more robust against XSS by leveraging the X-XSS-Protection header.
• Block clickjacking using the X-Frame-Options header.
• Leverage Content-Security-Policy to whitelist specific sources and endpoints.
• Prevent MIME-sniffing attacks using the X-Content-Type-Options header.
Resources / Tools

• Check Website HTTP Response Header
  – https://gf.dev/http-headers-test

• Secure Headers Test
  – https://gf.dev/secure-headers-test

• Scott Helme – Security Header Scanner
  – https://securityheaders.com

• HTTP Headers Reference

• HTTP Compatibility Among Browsers
  – https://caniuse.com
References

• https://www.netsparker.com/whitepaper-http-security-headers
• https://www.ntu.edu.sg/home/ehchua/programming/webprogramming/HTTP_Basics.html
• https://owasp.org/www-chapter-ghana/#div-pastevents
• https://www.keycdn.com/blog/http-security-headers
THANK YOU

Questions And Answers

Let’s Connect:
@egbordzor
linkedin.com/in/egbordzor
egbordzor@protonmail.com