

Large Scale Evaluation of Secure Headers in Wild

20214634 Sanggu Han
20213705 Yeongbin Hwang

Contents

- ❖ Motivation
- ❖ Goal
- ❖ Background
- ❖ Methodology
- ❖ Evaluation
- ❖ Limitations and Future Works
- ❖ Conclusion

Motivation

❖ Cross-site scripting (XSS) still exists!

Search Results

There are 20151 CVE Records that match your search.

Name	Description
CVE-2022-46391	AWStats 7.x through 7.8 allows XSS in the hostinfo plugin due to printing a response from Net::XWhois without proper checks.
CVE-2022-46151	Querybook is an open source data querying UI. In affected versions user provided data is not escaped in the error field of the auth callback url in 'querybook/server/app/auth/oauth_auth.py' and 'querybook/server/app/auth/okta_auth.py'. This may allow attackers Content Security Policy (CSP) is not enabled or 'unsafe-inline' is allowed. Users are advised to upgrade to the latest, patched version of querybook (version 3.14.2 or greater). Users unable to upgrade may enable CSP and not allow unsafe-inline or manually esc
CVE-2022-46148	Discourse is an open-source messaging platform. In versions 2.8.10 and prior on the 'stable' branch and versions 2.9.0.beta11 and prior on the 'beta' and 'tests-passed' branches, users composing malicious messages and navigating to drafts page could self-X- which have modified or disabled Discourse's default Content Security Policy. This issue is patched in the latest stable, beta and tests-passed versions of Discourse.
CVE-2022-45990	A cross-site scripting (XSS) vulnerability in the component /signup_script.php of Ecommerce-Website v1.0 allows attackers to execute arbitrary web scripts or HTML via a crafted payload injected into the eMail parameter.
CVE-2022-45916	ILIAS before 7.16 allows XSS.
CVE-2022-45848	Unauth. Stored Cross-Site Scripting (XSS) vulnerability in Contest Gallery plugin <= 13.1.0.9 on WordPress.
CVE-2022-45816	Auth. Stored Cross-Site Scripting (XSS) vulnerability in GD bbPress Attachments plugin <= 4.3.1 on WordPress.
CVE-2022-45769	A cross-site scripting (XSS) vulnerability in ClicShopping_V3 v3.402 allows attackers to execute arbitrary web scripts or HTML via a crafted URL parameter.
CVE-2022-45472	CAE LearningSpace Enterprise (with Intuity license) image 267r patch 639 allows DOM cross-site scripting (XSS) vulnerability to remove and insert content.
CVE-2022-45470	Apache Hima may have a stored cross-site scripting (XSS) vulnerability through parameters and X-headers. Since Apache Hima is a server-side application, these queries are not
CVE-2022-45403	Minio releases 2.1 and earlier do not escape display names of associated files, resulting in a stored cross-site scripting (XSS) vulnerability exploitable by attackers with Item/Configure permission.
CVE-2022-45387	Jenkins BAK Plugin 1.0.3 and earlier do not escape displayed content of build links, resulting in a stored cross-site scripting (XSS) vulnerability.
CVE-2022-45382	Jenkins Naginator Plugin 1.18.1 and earlier does not escape display names of source builds in builds that were triggered via Retry action, resulting in a stored cross-site scripting (XSS) vulnerability exploitable by attackers able to edit build display names.
CVE-2022-45380	Jenkins JUnit Plugin 1159.v0b_396e1e07dd and earlier converts HTTP(S) URLs in test report output to clickable links in an unsafe manner, resulting in a stored cross-site scripting (XSS) vulnerability exploitable by attackers with Item/Configure permission.
CVE-2022-45375	Auth. Stored Cross-Site Scripting (XSS) vulnerability in iFeature Slider plugin <= 1.2 on WordPress.
CVE-2022-45363	Auth. (subscriber+) Stored Cross-Site Scripting (XSS) in Muffingroup Betheme theme <= 26.6.1 on WordPress.
CVE-2022-45280	A cross-site scripting (XSS) vulnerability in the Url parameter in /login.php of EyouCMS v1.6.0 allows attackers to execute arbitrary web scripts or HTML via a crafted payload.
CVE-2022-45225	Book Store Management System v1.0 was discovered to contain a cross-site scripting (XSS) vulnerability in /bsms_ci/index.php/book. This vulnerability allows attackers to execute arbitrary web scripts or HTML via a crafted payload injected into the book_title pa
CVE-2022-45224	Web-Based Student Clearance System v1.0 was discovered to contain a cross-site scripting (XSS) vulnerability in Admin/add-admin.php. This vulnerability allows attackers to execute arbitrary web scripts or HTML via a crafted payload injected into the txtfullnam
CVE-2022-45223	Web-Based Student Clearance System v1.0 was discovered to contain a cross-site scripting (XSS) vulnerability in /Admin/add-student.php. This vulnerability allows attackers to execute arbitrary web scripts or HTML via a crafted payload injected into the txtfullna
CVE-2022-45221	Web-Based Student Clearance System v1.0 was discovered to contain a cross-site scripting (XSS) vulnerability in changepassword.php. This vulnerability allows attackers to execute arbitrary web scripts or HTML via a crafted payload injected into the txtnew_pa
CVE-2022-45218	Human Resource Management System v1.0.0 was discovered to contain a cross-site scripting (XSS) vulnerability. This vulnerability is triggered via a crafted payload injected into an authentication error message.
CVE-2022-45215	A cross-site scripting (XSS) vulnerability in Book Store Management System v1.0.0 allows attackers to execute arbitrary web scripts or HTML via a crafted payload injected into the Name parameter under the Add New System User module.
CVE-2022-45214	A cross-site scripting (XSS) vulnerability in Sanitization Management System v1.0.0 allows attackers to execute arbitrary web scripts or HTML via a crafted payload injected into the username parameter at /php-sms/classes/Login.php.
CVE-2022-45151	The stored-XSS vulnerability was discovered in Moodle which exists due to insufficient sanitization of user-supplied data in several "social" user profile fields. An attacker could inject and execute arbitrary HTML and script code in user's browser in context of vulne

More than 1,500 CVEs in this year!

Motivation

- ❖ What should we do?
- ❖ Input Sanitization

OWASP Java Html Sanitizer

Untrusted Input

```
<p>
    Check out
    <a href="https://www.bennadel.com" target="_blank" onmousedown="alert( 'XSS!' )" >my site</a>.
</p>

<marquee loop="-1" width="100%">
    I am very trustable! You can totes trust me!
</marquee>

<p>
    <strong>Thanks for stopping by!</strong> <em>You Rock!</em> &amp;
    <blink>Woot!</blink>
</p>
```

Removed code attributes and elements that were not explicitly allow-listed in the policy.

Sanitized Input

```
<p>
    Check out
    <a href="https://www.bennadel.com" target="_blank" rel="nofollow noopener noreferrer">my site</a>.
</p>

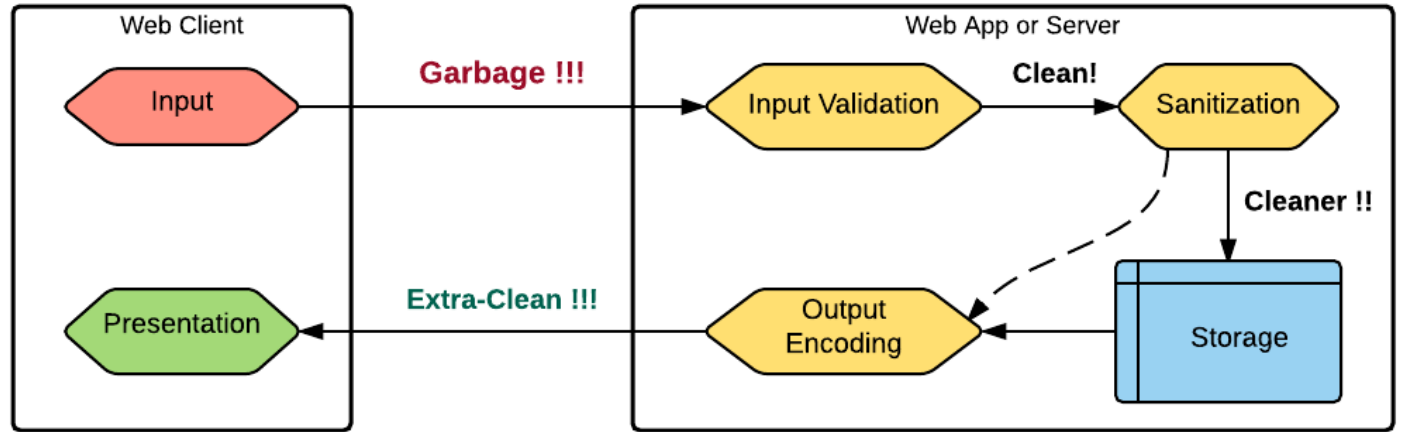
    I am very trustable! You can totes trust me!

<p>
    <strong>Thanks for stopping by!</strong> <em>You Rock!</em> &amp;
    Woot!
</p>
```

Added link-spam and opener attack protection.

Motivation

- ❖ What should we do?
- ❖ Encode data



```
<script><alert>Hello World!</alert> </script>
```



```
&lt;script&gt;&lt;alert&gt;Hello&nbsp;World!&lt;/alert&gt;&lt;/script&gt;
```

Motivation

- ❖ What should we do?
- ❖ Secure headers
 - Content-Security-Policy
 - X-XSS-Protection
 - X-Frame-Options
 - X-Content-Type-Options

Motivation

❖ Website

- There are a lot of websites and header fields
- Different headers are used in each cases

```
"http://www.makeuseof.com": {  
  "Server": "nginx",  
  "Date": "Sun, 04 Dec 2022 22:44:13 GMT",  
  "Content-Type": "text/html; charset=UTF-8",  
  "Transfer-Encoding": "chunked",  
  "X-XSS-Protection": "1; mode=block",  
  "X-Content-Type-Options": "nosniff"  
}
```

```
"http://www.aliexpress.us": {  
  "Content-Type": "text/html; charset=UTF-8",  
  "Vary": "Accept-Encoding",  
  "P3P": "CP=\"CAO PSA OUR\"",  
  "X-Application-Context": "ae-buyer-homepage",  
  "Cache-Control": "max-age=0",  
  "Server-Timing": "cdn-cache; desc=MISS"  
}
```

❖ Browser

- Secure header supported by browser is different.
- Ex) Content Security Policy

Chrome	Edge	Safari	Firefox	Opera	IE
			2-30		
4-35			1 31-34	10-22	
4 36-38	12-14		2 35	4 23-25	
5 39	5 15-18	3.1-9.1	3 36-44	5 26	
40-107	79-106	10-16.0	45-106	27-91	6-10
108	107	16.1	107	92	11
109-111		16.2-TP	108-109		

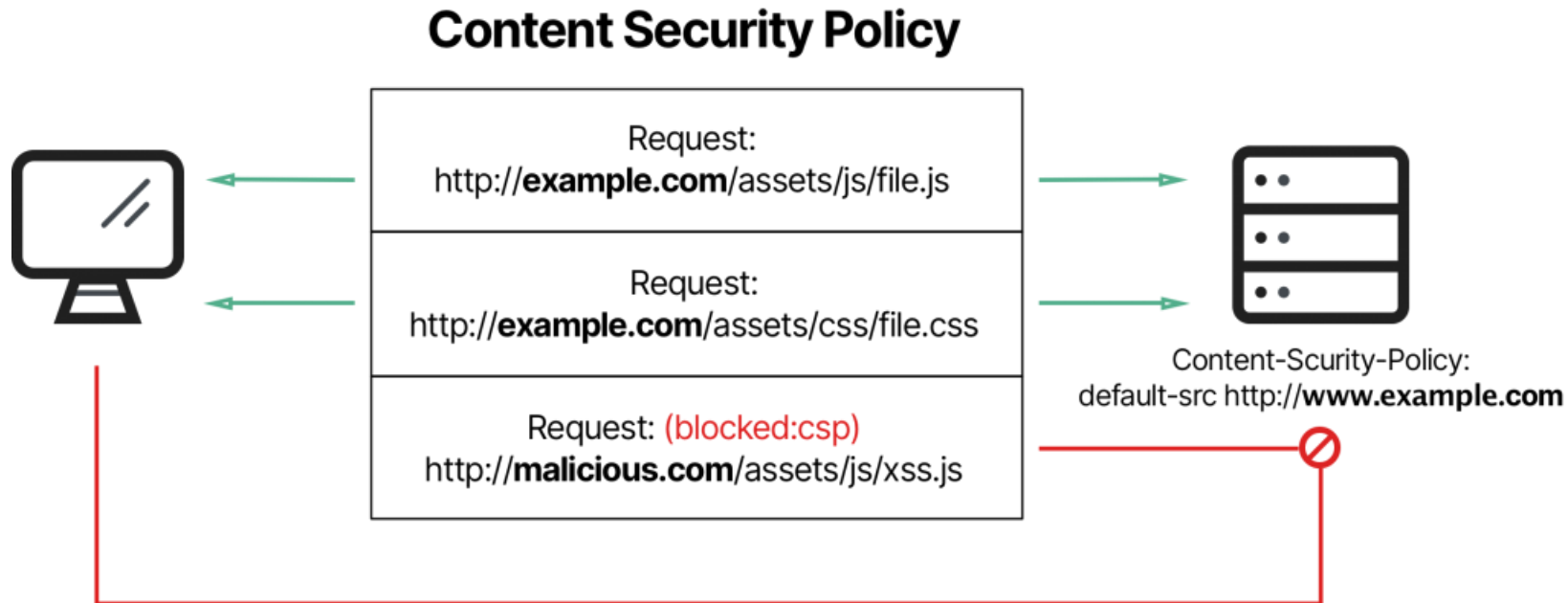
Goal

- ❖ Large-scale evaluation of secure headers
 - Compare the secure headers collected in different days (several month apart)
 - Identify which fields and websites are vulnerable based on secure headers
 - Analyze the websites that can be vulnerable due to a mismatch, which is the difference between a website's request and browser support for a security header for each browser.

Background

❖ Content Security Policy

- Policy that helps to detect and mitigate certain types of attacks, including Cross-Site Scripting



Background

- ❖ X-Content-Type-Options
 - Disables MIME sniffing and forces browser to use the type given in Content-Type.
- ❖ X-Frame-Options
 - Indicates whether a browser should be allowed to render a page in a <frame>, <iframe>, <embed> or <object>.
- ❖ X-XSS-Protection
 - Enables cross-site scripting filtering.

Methodology

- ❖ Define the target secure header
- ❖ Crawl secure headers in response headers
- ❖ Compare the headers between old ones and newly obtained ones
- ❖ Investigate supported headers for each browser

Define the Target Secure Header

- ❖ We considered secure headers based on 2 groups
 - Mozilla
 - OWASP Secure Headers Project
- ❖ We focused on fundamental cause
 - Javascript execution

Security headers
Cross-Origin-Embedder-Policy
Cross-Origin-Opener-Policy
Cross-Origin-Resource-Policy
Content-Security-Policy
Content-Security-Policy-Report-Only
Expect-CT
Feature-Policy
Referrer-Policy
Clear-Site-Data
Origin-Isolation
Strict-Transport-Security
Upgrade-Insecure-Requests
X-Content-Type-Options
X-Frame-Options
X-Permitted-Cross-Domain-Policies
X-XSS-Protection

Define the Target Secure Header

- ❖ Filter out headers related to script execution

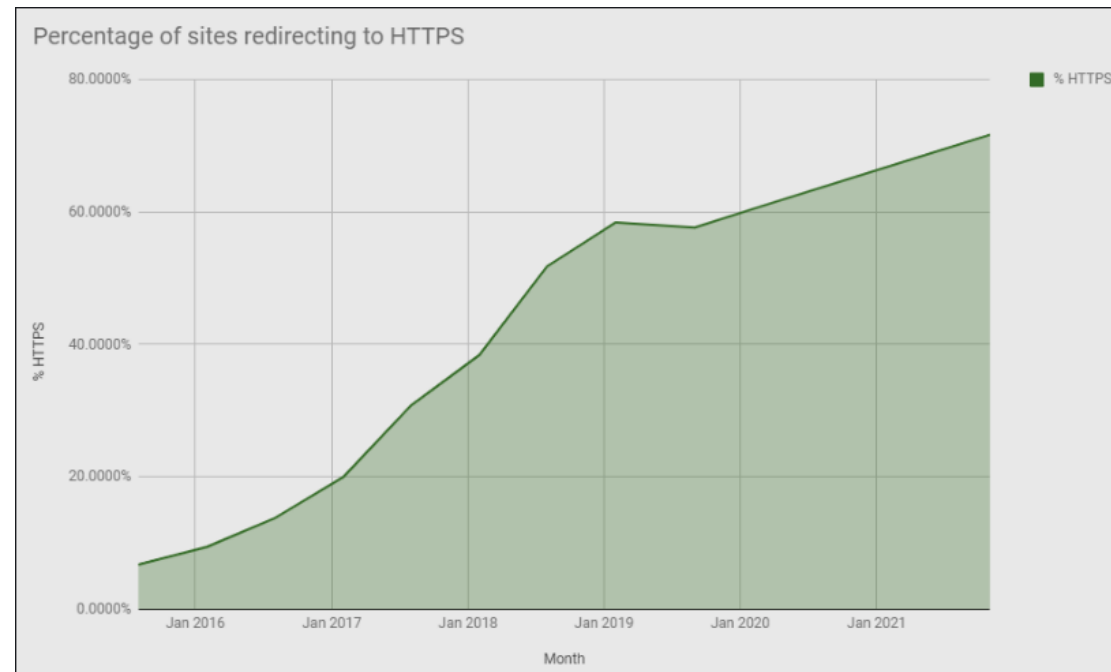
Security headers
Cross-Origin-Embedder-Policy
Cross-Origin-Opener-Policy
Cross-Origin-Resource-Policy
Content-Security-Policy
Content-Security-Policy-Report-Only
Expect-CT
Feature-Policy
Referrer-Policy
Clear-Site-Data
Origin-Isolation
Strict-Transport-Security
Upgrade-Insecure-Requests
X-Content-Type-Options
X-Frame-Options
X-Permitted-Cross-Domain-Policies
X-XSS-Protection



Security headers (related to script)
Cross-Origin-Embedder-Policy
Cross-Origin-Opener-Policy
Cross-Origin-Resource-Policy
Content-Security-Policy
Feature-Policy
X-Content-Type-Options
X-Frame-Options
X-XSS-Protection

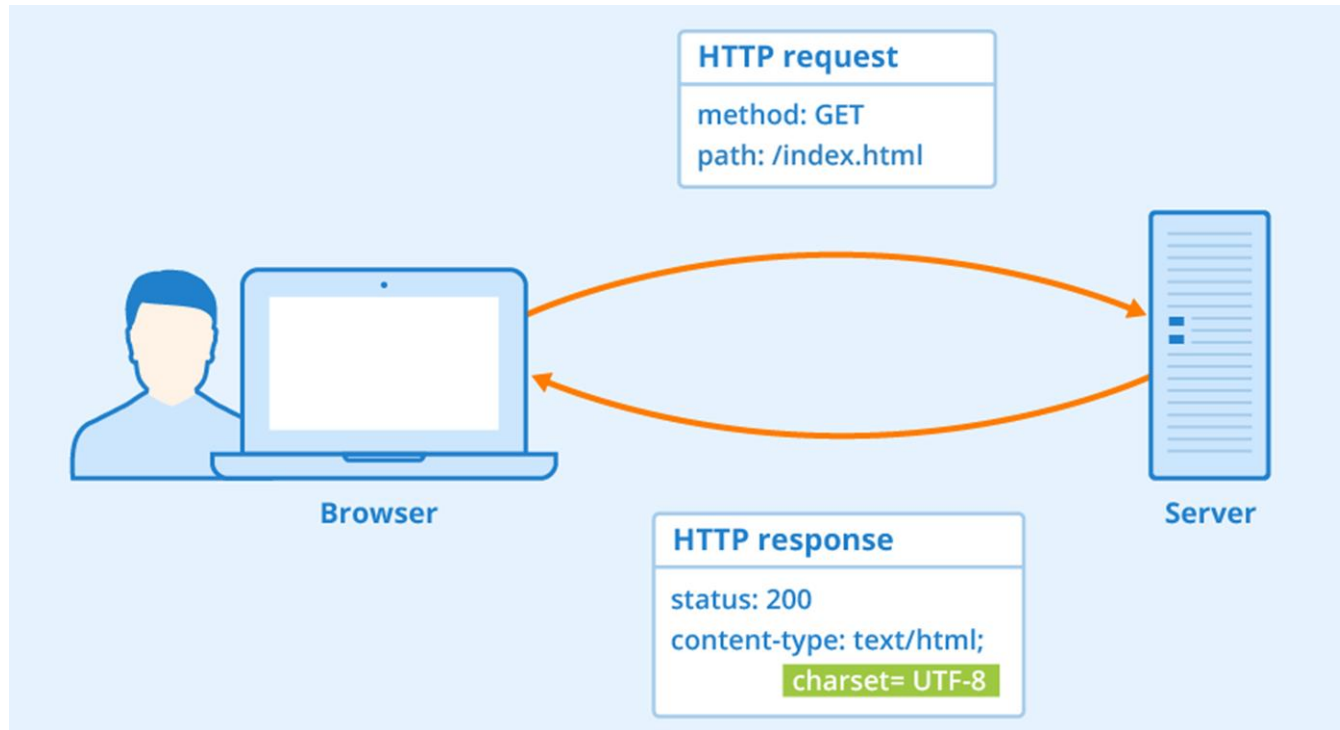
Crawl Security Headers

- ❖ Crawl the past response headers in previous crawl report.
 - <https://crawler.ninja/>
 - They provide raw data and statistics



Crawl Security Headers

- ❖ Implement the crawler that can extract the response headers
- ❖ Collect the response headers from alexa top 1M websites



▼ Response Headers

```
accept-ranges: bytes
age: 47234
alt-svc: h3=":443"; ma=2592000,h3-29=":443"; ma=2592000,h3-Q050=":443";
a=2592000,h3-Q046=":443"; ma=2592000,h3-Q043=":443"; ma=2592000,quic=
443"; ma=2592000; v="46,43"
cache-control: public, max-age=31536000
content-encoding: br
content-length: 949
content-type: text/css
cross-origin-opener-policy-report-only: same-origin; report-to="youtube"
cross-origin-resource-policy: cross-origin
date: Tue, 06 Dec 2022 15:04:36 GMT
expires: Wed, 06 Dec 2023 15:04:36 GMT
```

Compare the Headers

- ❖ Compare the headers between old ones and newly obtained ones
- ❖ Websites change their responses over time

```

"date": "Fri, 25 \\Nov 2022 01:00:09 GMT",
"expires": "-1",
"cache-control": "private, max-age=0",
"content-type": "text/html; charset=UTF-8",
"strict-transport-security": "max-age=31536000",
"cross-origin-opener-policy-report-only": "same-origin-allow-popups; report-to=\"{\\\"group\\\":\\\"gws\\\",\\\"max_age\\\":2592000,\\\"endpoints\\\":[{\\\"url\\\":\\\"https://www.google.com/\\\"}]}\"",
"report-to": "{\\\"group\\\":\\\"gws\\\",\\\"max_age\\\":2592000,\\\"endpoints\\\":[{\\\"url\\\":\\\"https://www.google.com/\\\"}]}\"",
"bfcache-opt-in": "unload",
"p3p": "CP=\\\"This is not a P3P policy! See g.co/p3phelp for more info.\\\"",
"server": "gws",
"x-xss-protection": "0",
"x-frame-options": "SAMEORIGIN",
"set-cookie": "1P_JAR=2022-11-25-01; expires=Sun, 25-\\Nov-2022 01:00:09 GMT; path=/; domain=.google.com; HttpOnly; Secure",
"alt-svc": "h3=\\\":443\\\"; ma=2592000, h3-29=\\\":443\\\"; ma=2592000, h3-Q050=\\\":443\\\"; ma=2592000, quic=\\\":443\\\"; ma=2592000; v=\\\"46,43\\\"",
"accept-ranges": "none",
"vary": "Accept-Encoding"
}

```



```

http://www.google.com": {
  "Date": "Wed, 07 Dec 2022 14:43:32 GMT",
  "Expires": "-1",
  "Cache-Control": "private, max-age=0",
  "Content-Type": "text/html; charset=UTF-8",
  "Strict-Transport-Security": "max-age=31536000",
  "Cross-Origin-Opener-Policy-Report-Only": "same-origin-allow-popups; repo
  "Report-To": "{\\"group\\":\\"gws\\",\\"max_age\\":2592000,\\"endpoints\\":[{\\"ur
  "Accept-CH": "Sec-CH-UA-Platform",
  "BFCache-Opt-In": "unload",
  "Permissions-Policy": "unload=()",
  "Origin-Trial": "AqRrpS1jM/H0s1rGR0CnXerKEP/QFz7qj9ApDSZqAO+0U+KcT/h/
  1xA6akW4ar0kT0V1bw5MD4t807L70FwM5gUAAABfeyJvcmlnaW4iOiJodHRwcovL3d3dy5nb
  SI6MTY3ODIzMzU5OX0=",
  "P3P": "CP=\\\"This is not a P3P policy! See g.co/p3phelp for more info.\\\"
  "Content-Encoding": "gzip",
  "Server": "gws",
  "X-XSS-Protection": "0",
  "X-Frame-Options": "SAMEORIGIN",
  "Set-Cookie": "1P_JAR=2022-12-07-14; expires=Fri, 06-Jan-2023 14:43:32 GM
  "Alt-Svc": "h3=\\\":443\\\"; ma=2592000,h3-29=\\\":443\\\"; ma=2592000,h3-Q050=\\
  ma=2592000,quic=\\\":443\\\"; ma=2592000; v=\\\"46,43\\\"",
  "Transfer-Encoding": "chunked"
}

```


Investigate each Browser

- ❖ Use caniuse api that implemented in npm
- ❖ Can check browser compatibility and supported version
- ❖ We make up supported version lists
 - Based on all security headers

```
caniuse.getSupport('contentsecuritypolicy', true)
{
  and_chr: { y: 107 },
  and_ff: { y: 106 },
  and_qq: { y: 13.1 },
  and_uc: { y: 13.4 },
  android: { n: 4.2, y: 4.4 },
  chrome: { n: 13, y: 14 },
  edge: { y: 12 },
  firefox: { n: 3.6, y: 4 },
  ios_saf: { y: 6, n: 4.2, a: 5 },
  kaios: { y: 2.5 },
  op_mini: {},
  op_mob: { n: 12.1, y: 72 },
  opera: { n: 12.1, y: 15 },
  safari: { n: 5, y: 6, a: 5.1 },
  samsung: { y: 4 }
}
```

Evaluation

- ❖ Overall adoption trend
- ❖ Content-Security-Policy & X-Frame-Options
- ❖ X-XSS-Protection
- ❖ X-Content-Type-Options

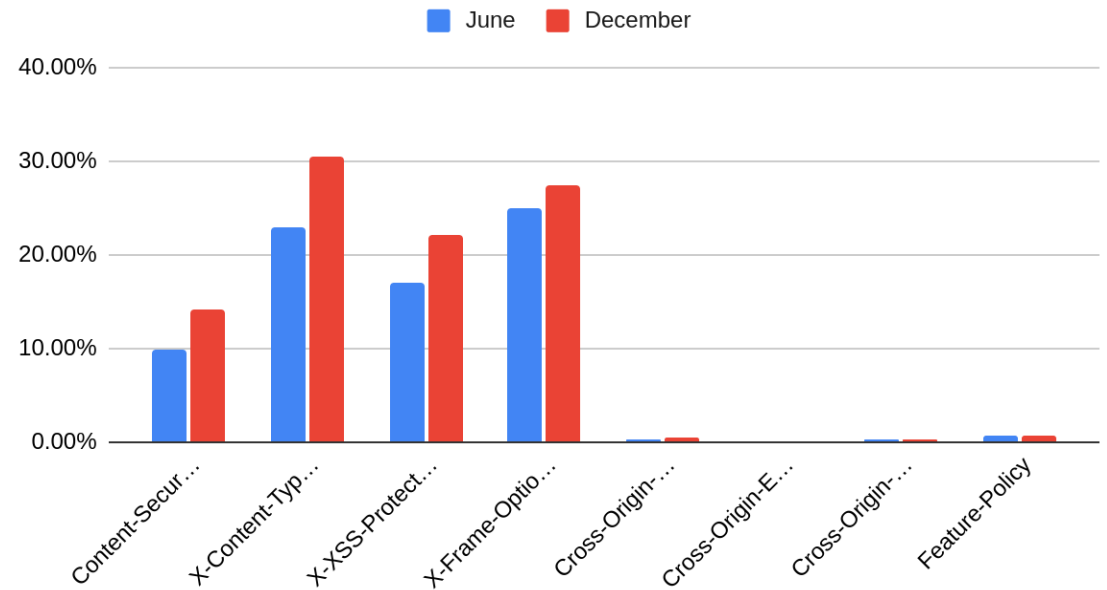
Evaluation

- ❖ Target website
 - Alexa top 1M websites
 - \approx 820k Response header data In June (from crawl.ninja)
 - \approx 650k Response header domains in December (by our team)
- ❖ Target browser
 - Top 15 browsers in mobile and desktop.

Overall Trends

- ❖ # of secure header / # of response
 - On June, December
- ❖ Secure header adoption trend
 - Increases as time goes
 - New secure headers got no spotlight

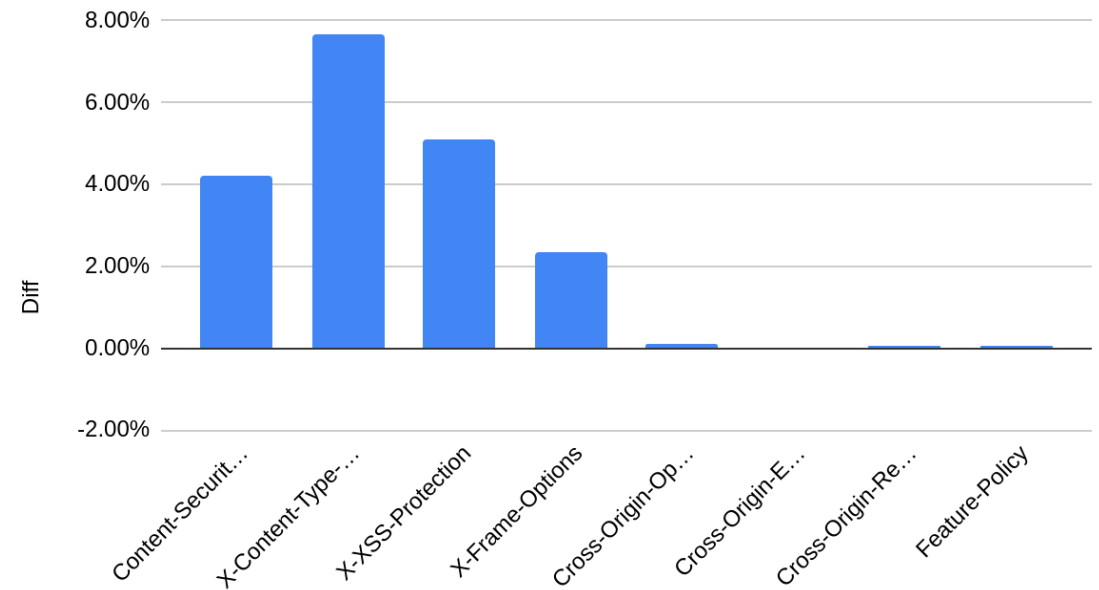
Secure header adoption rate trend



Overall Trends

- ❖ # of newly added secure header
 - On December w.r.t June
- ❖ Newly-Added secure headers barely increased

Secure header adoption change



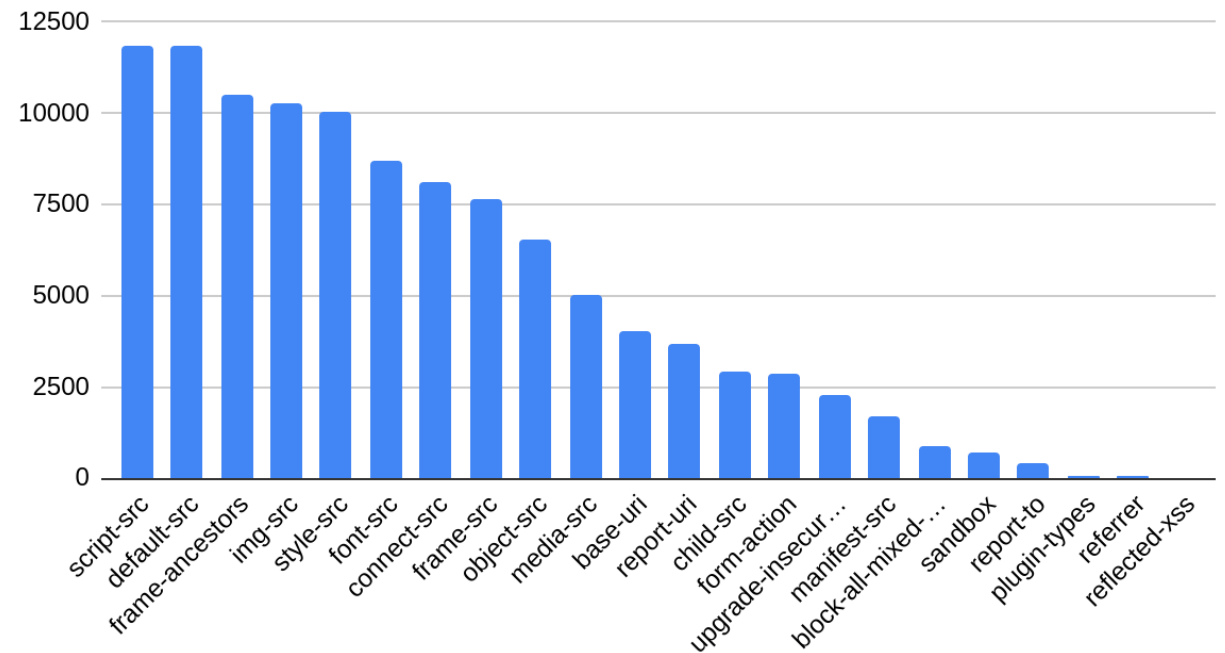
Content-Security-Policy (CSP)

- ❖ 91k of 650k domains (**$\approx 14\%$**) use **CSP** in their header
- ❖ For last 6 months, **4% more domains adopted CSP**
- ❖ Collected **19786 unique rules** from crawled data
 - $\approx 12k$ rules were included script-src to prevent XSS

Content-Security-Policy (CSP)

- ❖ $\approx 20k$ unique rules
- ❖ script-src takes $\approx 60\%$
 - 60% to prevent XSS

Directives in unique rules



Content-Security-Policy (CSP)

- ❖ CSP's *frame-ancestor* directive obsoletes XFO (10k cases)
- ❖ X-Frame-Options (XFO) (180k cases)
- ❖ Either one of *frame-ancestor* or XFO is set, improves protection against clickjacking
 - 186k cases observed
- ❖ So, **≈ 29% domains improved protection against clickjacking**

Content-Security-Policy (CSP)

- ❖ Most common wrong configuration case study
 - Script-src
 - Using unsafe-inline only
 - Using white-list only
 - Syntax
 - Using self, none instead of 'self', 'none'
 - Using semicolon (;) as delimiter for values

X-XSS-Protection (XXSSP)

- ❖ 143k of 650k domains (**≈ 22%**) use **XXSSP** in their header
- ❖ Used for protect website from XSS
 - Detect reflected XSS

X-XSS-Protection (XXSSP)

- ❖ 0 (disabled) : 5873 cases (4%)
- ❖ **1 (enabled) : 137167 cases (95.7%)**
- ❖ Wrong config: 338 cases (0.2%)
 - 0, 1
 - On
 - Sameorigin
 - ...

X-Content-Type-Options (XCTO)

- ❖ 198k of 650k domains (**≈ 30%**) use **XCTO** in their header
- ❖ Only one option exists, *nosniff*
- ❖ Common wrong configuration
 - Values for other secure header (0, 1, script-src, ...)

Browsers

- ❖ Different browser, different support
- ❖ Most of the secure headers are supported in browsers.

	CSP	XCTO	XFO	CORS	XXP	COEP	COMP	CORP
Chrome	O	O	O	O	X	O	O	O
Edge	O	O	O	O	X	O	O	O
Firefox	O	O	O	O	X	O	O	O
Safari	O	O	O	O	O	O	O	O
Safari on iOS	O	O	O	O	O	O	O	O
Android	O	O	O	O	X	O	O	O

Table - Compatibility Matrix on most recent stable version

Browsers

- ❖ XXSSP is deprecated on every browsers except safari

Deprecated on most browsers!

	CSP	XCTO	XFO	CORS	XXP	COEP	COMP	CORP
Chrome	O	O	O	O	X	O	O	O
Edge	O	O	O	O	X	O	O	O
Firefox	O	O	O	O	X	O	O	O
Safari	O	O	O	O	O	O	O	O
Safari on iOS	O	O	O	O	O	O	O	O
Android	O	O	O	O	X	O	O	O

Table - Compatibility Matrix on most recent stable version

XXSSP Revisited

- ❖ Case study
 - XXSSP is enabled, but CSP is not enabled (over 87k cases, **≈ 11.7%**)
 - **Vulnerable to XSS in most browsers (Chrome, Firefox, ...)**
 - **11.7% of domains are potentially vulnerable** to script execution!

Limitations and Future Works

- ❖ Investigate rest of security headers
 - Other factors such as forcing HTTPS, checking certificates, ... can be done with secure headers too.
- ❖ Analysis tool for response header
 - By leveraging best logics for each secure header, let developer know what's the best for their service protection

Conclusion

- ❖ Secure headers are proposed to increase the security of web applications.
 - Prevent script execution, restrict script's permission, ...
- ❖ Deployment rates of critical secure header is insufficient
 - Especially on newly introduced ones
- ❖ There are some wrong configurations in response headers.
 - Wrong configurations still makes website vulnerable

Thank you