CSP: The Good, the Bad and the Ugly

Ilya Nesterov | Shape Security
Agenda
What does CSP stand for?

Content Security Policy (CSP) - a defense-in-depth mechanism that web applications can use to mitigate a broad class of content injection vulnerabilities, such as XSS.
CSP Level 1

- connect-src
- frame-src
- img-src
- font-src
- media-src
- default-src
- object-src
- script-src
- style-src
- sandbox *
- report-uri

* optional
CSP Level 1

- Policy delivery via HTTP header only
- Multiple CSP headers allowed
- Sandbox directive is optional
- script-src governs workers
CSP Level 2

- form-action
- frame-ancestors* **
  - Navigation directives
- base-uri
- plugin-types
- sandbox* **
  - Document directives
- report-uri*
  - Reporting directives
- changes/new
- no change

- frame-src
- child-src
  - connect-src
  - img-src
  - font-src
  - media-src
  - object-src
  - script-src
  - style-src
  - default-src

* ignored in <meta>    ** ignored when monitored
New in CSP Level 2

✦ Policy delivery via <meta>
✦ New directives: child-src, form-action, frame-ancestors, base-uri, plugin-types
✦ Source-expression supports hash and nonce
✦ host-source can use path for matching
✦ SecurityPolicyViolationEvent
✦ Extended violation report
✦ child-src governs workers
CSP Level 3 Editor's draft

Navigation directives:
- form-action
- frame-ancestors
- navigation-to

Document directives:
- base-uri
- plugin-types
- sandbox
- disown-opener

Reporting directives:
- report-to
- report-uri

Fetch directives:
- child-src
- connect-src
- img-src
- font-src
- manifest-src
- media-src
- object-src
- script-src
- style-src

* ignored in <meta>
** ignored when monitored

#FluentConf
New in CSP Level 3

✦ New directives: manifest-src, worker-src, report-to, block-mixed-content, upgrade-insecure-requests, require-sri-for, navigation-to, disown-opener
✦ frame-src undeprecated
✦ child-src, report-url deprecated
✦ New in source-expression: ‘strict-dynamic'
✦ Changes in url and source-expression matching algorithms
✦ Additional changes to violation reports
# Browser compatibility CSP level 1

Mitigate cross-site scripting attacks by whitelisting allowed sources of script, style, and other resources.

## Content Security Policy 1.0

<table>
<thead>
<tr>
<th></th>
<th>IE</th>
<th>Edge</th>
<th>Firefox</th>
<th>Chrome</th>
<th>Safari</th>
<th>Opera</th>
<th>iOS Safari</th>
<th>Opera Mini</th>
<th>Android Browser</th>
<th>Chrome for Android</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>11</td>
<td>15</td>
<td>54</td>
<td>59</td>
<td>10.1</td>
<td>45</td>
<td>10.3</td>
<td>all</td>
<td>56</td>
<td>59</td>
</tr>
</tbody>
</table>

### Notes

The standard HTTP header is `Content-Security-Policy` which is used unless otherwise noted.

† Supported through the `x-Content-Security-Policy` header.
Browser compatibility CSP level 2

Mitigate cross-site scripting attacks by whitelisting allowed sources of script, style, and other resources. CSP 2 adds hash-source, nonce-source, and five new directives.

<table>
<thead>
<tr>
<th>Browser</th>
<th>IE</th>
<th>Edge</th>
<th>Firefox</th>
<th>Chrome</th>
<th>Safari</th>
<th>Opera</th>
<th>iOS Safari</th>
<th>Opera Mini</th>
<th>Android Browser</th>
<th>Chrome for Android</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>14</td>
<td>52</td>
<td>54</td>
<td>49</td>
<td>57</td>
<td>58</td>
<td>9.2</td>
<td>10.2</td>
<td>4.4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>11</td>
<td>15</td>
<td>59</td>
<td>10.1</td>
<td>45</td>
<td>10.3</td>
<td>all</td>
<td>56</td>
<td>59</td>
<td></td>
</tr>
<tr>
<td></td>
<td>16</td>
<td>54</td>
<td>60</td>
<td>11</td>
<td>46</td>
<td>11</td>
<td>47</td>
<td>62</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Notes: Firefox 45+ is missing the plugin-types directive.

#FluentConf
# CSP directives compatibility matrix

<table>
<thead>
<tr>
<th>Directive</th>
<th>Chrome</th>
<th>Edge</th>
<th>Firefox</th>
<th>IE</th>
<th>Opera</th>
<th>Safari</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Fetch directives</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>child-src</td>
<td>40</td>
<td>No support</td>
<td>45</td>
<td>No support</td>
<td>27</td>
<td>10</td>
</tr>
<tr>
<td>connect-src</td>
<td>25</td>
<td>14</td>
<td>23</td>
<td>No support</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>default-src</td>
<td>25</td>
<td>14</td>
<td>23</td>
<td>No support</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>font-src</td>
<td>25</td>
<td>14</td>
<td>23</td>
<td>No support</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>frame-src</td>
<td>25</td>
<td>14</td>
<td>23</td>
<td>No support</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>img-src</td>
<td>25</td>
<td>14</td>
<td>23</td>
<td>No support</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>manifest-src</td>
<td>yes</td>
<td>No support</td>
<td>41</td>
<td>No support</td>
<td>Yes</td>
<td>No support</td>
</tr>
<tr>
<td>media-src</td>
<td>25</td>
<td>14</td>
<td>23</td>
<td>No support</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>object-src</td>
<td>25</td>
<td>14</td>
<td>23</td>
<td>No support</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>script-src</td>
<td>25</td>
<td>14</td>
<td>23</td>
<td>No support</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>style-src</td>
<td>25</td>
<td>14</td>
<td>23</td>
<td>No support</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>worker-src</td>
<td>58</td>
<td>No support</td>
<td>No support</td>
<td>No support</td>
<td>No support</td>
<td>No support</td>
</tr>
<tr>
<td><strong>Document directives</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>base-uri</td>
<td>40</td>
<td>No support</td>
<td>35</td>
<td>No support</td>
<td>27</td>
<td>10</td>
</tr>
<tr>
<td>plugin-types</td>
<td>40</td>
<td>No support</td>
<td>No support</td>
<td>No support</td>
<td>27</td>
<td>10</td>
</tr>
<tr>
<td>sandbox</td>
<td>25</td>
<td>14</td>
<td>50</td>
<td>10</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td><strong>Navigation directives</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>form-action</td>
<td>40</td>
<td>No support</td>
<td>36</td>
<td>No support</td>
<td>27</td>
<td>10</td>
</tr>
<tr>
<td>frame-ancestors</td>
<td>40</td>
<td>No support</td>
<td>33</td>
<td>No support</td>
<td>26</td>
<td>10</td>
</tr>
<tr>
<td><strong>Reporting directives</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>report-uri</td>
<td>25</td>
<td>14</td>
<td>23</td>
<td>No support</td>
<td>15</td>
<td>7</td>
</tr>
<tr>
<td>report-to</td>
<td>No support</td>
<td>No support</td>
<td>No support</td>
<td>No support</td>
<td>No support</td>
<td>No support</td>
</tr>
<tr>
<td><strong>Other directives</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>block-all-mixed-content</td>
<td>yes</td>
<td>?</td>
<td>48</td>
<td>No support</td>
<td>yes</td>
<td>No support</td>
</tr>
<tr>
<td>require-sri-for</td>
<td>48</td>
<td>No support</td>
<td>No support</td>
<td>No support</td>
<td>41</td>
<td>No support</td>
</tr>
<tr>
<td>upgrade-insecure-requests</td>
<td>44</td>
<td>No support</td>
<td>48</td>
<td>No support</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>
I want CSP, what should I do?

Where not to use CSP:

✧ Static website with public information
✧ Large application with many XSS

Understand what triggers a CSP violation
The bad
CSP violations

✦ object-src and default-src is not defined

```html
<object data="https://evil-hacker.com/evil.swf"
    <param name="allowscriptaccess" value="always">
</object>
```

✦ usage of unsafe-inline

```html
<img src="x" onerror="evil()">
<script src="data:text/javascript,evil()"></script>
```

✦ path restriction and redirect

Content-Security-Policy: script-src example.org partially-trusted.org/foo/bar.js

```html
<script src="/example.org?redirect=partially-trusted.org/evil/script.js">
```
CSP violations due to SOP

CSP only on some pages

```
var secret = "42";
```

```
CSP

No CSP

injected.js

http://attacker.com
```

```
No CSP

injected.js

CSP

var secret = "42";
```
Strict CSP

- Define default-src or script-src
- Prevent fetching and executing plugin resources embedded: object-src 'none'
- Use nonce/hash to whitelist inline scripts
- Do not use 'unsafe-eval' unless you use eval()
- Tighten your source expression
CSP adoption steps

✦ Refactor, refactor, refactor
  • nonce for inlined scripts
  • inline event handlers and javascript: document.write -> document.createElement
  • strict-dynamic
✦ Delivery mechanism (header vs <meta>)
✦ Start with report-only
✦ Test, test, test
✦ Analyze violation reports
✦ Make you policy backward compatible
the ugly
CSP Backward compatibility

CSP:
object-src 'none'; script-src 'nonce-{random-base64-value}' 'unsafe-inline' 'strict-dynamic';

CSP level 3 browser view:
object-src 'none'; script-src 'nonce-{random-base64-value}' 'strict-dynamic';

CSP level 2 browser view:
object-src 'none'; script-src 'nonce-{random-base64-value}';

CSP level 1 browser view:
object-src 'none'; script-src 'unsafe-inline';
Deployment into production

- Prepare CSP reports collector
- Start with report only
- A/B testing
- Continuously analyse CSP reports
CSP reports are not easy

✦ How to:
  - identify different versions of your CSP
  - report only vs enforced
  - filter noise
  - find if someone is trying to break in
✦ There is no one simple solution
Alexa top 1 000 000 data

- upgrade-insecure-requests only: 616 (Sep. 2016), 1390 (Jun. 2017)
CSP policies closer look
XSS protection vs Strict XSS protection policies

XSS protection:
- 72.36% (Sep. 16)
- 69.03% (Jan. 17)
- 66.33% (Jun. 17)

Strict CSP:
- 3.87% (Sep. 16)
- 5.48% (Jan. 17)
- 4.45% (Jun. 17)
Issues found in Alexa top million CSP

- CSP Infos: 29.36% in Jan. 17, 29.71% in Jun. 17
- CSP Errors: 5.92% in Jan. 17, 7.47% in Jun. 17
- CSP Warnings: 42.89%
Common errors found in Alexa top million CSP

- Unrecognised or outdated directive-name: 46.26% (Jan. 17), 29.82% (Jun. 17)
- Gibberish instead of source-expression: 18.37% (Jan. 17), 19.30% (Jun. 17)
- Gibberish instead of directive-name: 17.01% (Jan. 17), 10.53% (Jun. 17)
- Non-ASCII and non-printable character in directive-value: 10.88% (Jan. 17), 17.19% (Jun. 17)
- Comma in source-expression: 10.20% (Jan. 17), 12.63% (Jun. 17)
Resources

- https://csp-evaluator.withgoogle.com/
- https://csp.withgoogle.com
- https://github.com/shapesecurity/salvation
- https://report-uri.io
- https://www.w3.org/TR/CSP3/
- https://www.w3.org/2011/webappsec/
- public-webappsec@w3.org
Questions?

mailto: ilya.nesterov@gmail.com
mailto: ilya@shapesecurity.com
twitter: @ilya_online