Complex Security Policy? A Longitudinal Analysis of Deployed Content Security Policies

Sebastian Roth, Timothy Barron, Stefano Calzavara, Nick Nikiforakis & Ben Stock

Network and Distributed System Security Symposium (NDSS ’20)
Cross-Site Scripting (XSS)

1. XSS Payload
   https://vuln.com?pl=<script src=evil.com>

2. HTTP GET Request vuln.com

3. HTTP Response

4. HTTP GET Request evil.com

5. HTTP Response of evil.js
Content Security Policy (CSP)

1. XSS Payload
   https://vuln.com?pl=\<script src=evil.com\>\

2. HTTP GET Request vuln.com

3. HTTP Response with CSP Header

4. HTTP GET Request evil.com

5. HTTP Response of evil.js
Content Security Policy (CSP)

```html
<html>
<body>
<!-- ad.com includes company.com -->
<script src="https://ad.com/someads.js"></script>
<script>
    // ... meaningful inline script
</script>
</body>
</html>
```

```html
<html>
<body>
<script nonce="d90e0153c074f6c3fcf53"
    src="https://ad.com/someads.js"></script>
<script>
    let script = document.createElement("script");
    script.src = "http://ad.com/ad.js";
    document.body.appendChild(script);
</script>
</body>
</html>
```

```html
<html>
<body>
<script nonce="d90e0153c074f6c3fcf53">
    // ... meaningful inline script
</script>
</body>
</html>
```
Research Questions

- We know from others studies that:
  - CSP adoption is far behind expectations
  - Many deployed policies are insecure

- Why is CSPs adoption so low?
- For what purpose is CSP used in the wild?
- What are the problems of deploying a CSP?
Methodology

Dataset Construction
- Create a list of the Top 10k sites over time.
- Intersection of the Alexa Top sites of each month 2012 – 2018

Data Collection
- Use Wayback Machine
- Collected 20,179 CSPs
- Checked Archive Data against Common Crawl

Analytics
- Classify CSP Use-Cases
- Analyze the Directives and their Use-Cases
- Detailed case-studies & Developer opinions
Use-Case 1: Script Content Control

CSP Adoption vs. Script Content Control

NDSS 2020 – Sebastian Roth – Complex Security Policy?
Use-Case 1: Script Content Control

- Used ‘unsafe-inline’ in CSP
- Has inline event-handlers

<table>
<thead>
<tr>
<th>Year</th>
<th>Used ‘unsafe-inline’</th>
<th>Has inline event-handlers</th>
</tr>
</thead>
<tbody>
<tr>
<td>2014</td>
<td>378</td>
<td>180 (48%)</td>
</tr>
</tbody>
</table>

Graph shows the trend from 2014 to 2018 with the following data points:

NDSS 2020 – Sebastian Roth – Complex Security Policy?
Airbnb’s journey to secure their CSP

- 11-2014
  - CSP report-only
  - script-src: 17 entries

They needed 3 ½ years to deploy a non-trivially bypassable CSP
Use-Case 2: TLS Enforcement

CSP Adoption  Script Content Control  TLS Enforcement

NDSS 2020 – Sebastian Roth – Complex Security Policy?
Use-Case 2: TLS Enforcement

- TLS Enforcement
- Upgrade Insecure Requests
- Block All Mixed Content
- Whitelist HTTPS schema

NDSS 2020 – Sebastian Roth – Complex Security Policy?
Use-Case 2: TLS Enforcement

• We collected all main pages of Upgrade-Insecure-Requests sites from the Archive and extracted the 3rd party URLs

• How hard is HTTPS migration in the wild?
  – Mixed Content on 4,785 sites from the Alexa Top 10k
  – For 89% of them, all HTTP resources are upgradeable
Framing based attacks

Wanna see more Kittens?

Yes!

https://kittenpics.org/
Framing Control – X-Frame-Options

X-Headers are not standardized!

Leads to security problems:
  • Partial support
  • Double Framing

... as well as functionality problems
  • X-Frame-Options can only have a single whitelist entry
Use-Case 3: Framing Control

- CSP Adoption
- Script Content Control
- TLS Enforcement
- Framing Control
Use-Case 3: Framing Control

How does CSP frame-ancestors fix these problems:

- **Partial support / Inconsistent implementation:**
  CSP frame-ancestors is a well-defined standard in CSP since 2014. Thus, all “modern” browsers support it.

- **Double Framing:**
  Applies to all of a frame's ancestors not only the top-most frame.

- **Explicit whitelist:**
  frame-ancestors supports wildcards and multiple source-expressions
  frame-ancestors www.foo.com ‘self’ *.partner.com
Use-Case 3: Framing Control

![Graph showing the use of X-Frame-Options, CSP frame-ancestors, and both over time from 2012 to 2018. The graph indicates a significant increase in the use of framing control measures over the years. XFO is deprecated.](image-url)
Framing Control – Developer Study

- We notified the 2,699 Web sites about their problem using XFO but not CSP frame-ancestors via email.

- Received 117 responses that went beyond automatic answers.

- Many developers have the misconception that different CSP features cannot be used in isolation!
CSP destroys Web applications

“ [...] adding CSP [...] already placed on the roadmap in August of last year. We ran into some trouble with properly enabling the policies, as they ended up effectively killing the website. ”
Misconceptions about CSP

CSP is a complex beast [...]. Some of our partner are iframing our site. We already had issue to implement the X-Frame header, that we did not want to deal with CSP.
Framing Control – Developer Study

Do you believe CSP is a viable option to improve your site’s resilience against XSS attacks?

Yes
No
I don't know
No answer

Would your site work out of the box if you deployed a script-content restricting CSP today (disallow eval, inline scripts, and event handlers)?

Yes
No
I don't know
No answer
Complex Security Policy?

NDSS 2020 – Sebastian Roth – Complex Security Policy?
How to go back in time?

Also stores original HTTP headers prefixed with X-Archive-Orig-
We have a small team. Do we want to update our version of python or do we want to add CSP? Do we want to move to the new LTS version of Ubuntu or CSP? [...] CSP always loose.
CSP is too complex to deploy

"[...] many first and third party integrations [...] having a generic CSP policy that adds value and which is suitable for our entire estate is something that is very difficult to achieve."
Did you know about the frame-ancestors directive and its improved protection capabilities compared to X-Frame-Options before our notification?

Did you know that frame-ancestors can be deployed independently of any other part of CSP before our notification?
Why have you implemented the X-Frame-Options header?

- Pentest / Consultant
- Tools we use
- Own decision
- Other

0  5  10  15  20  25

NDSS 2020 – Sebastian Roth – Complex Security Policy?
Appendix – Good CSP Deployment

GitHub’s journey to secure their CSP

- **11-2013**
  - Started to use CSP in Enforcement Mode
  - script-src contains 5 entries (Self, Analytics, CDNs)

- **05-2014**
  - Removed some Analytics & built own CDN
  - script-src contains 2 entries (GitHub CDN & Google Analytics)

- **10-2014**
  - Removed Google Analytics
  - script-src contains 1 single entry (GitHub CDN)

They *never* ever used any dangerous source expression!
=> ALLOW-FROM fails insecurely for Chrome & Co. *

* Meanwhile, since Firefox 70, ALLOW-FROM is no longer supported.
Appendix – Double Framing

=> In legacy browsers XFO is only checked against top-most frame.
Appendix – Related Work


Lukas Weichselbaum  
Google Inc.  
lwe@google.com

Michele Spagnuolo  
Google Inc.  
mikispag@google.com

Artur Janc  
Google Inc.  
aaj@google.com

Sebastian Lekies  
Google Inc.  
slekies@google.com

**How the Web Tangled Itself: Uncovering the History of Client-Side Web (In)Security**

Ben Stock, CISP, Saarland University; Martin Johns, SAP SE;  
Marius Steffens and Michael Backes, CISP, Saarland University

**Semantics-Based Analysis of Content Security Policy Deployment**

STEFANO CALZAVARA, Università Ca’ Foscari Venezia  
ALVISE RABITTI, Università Ca’ Foscari Venezia  
MICHELE BUGLIESI, Università Ca’ Foscari Venezia