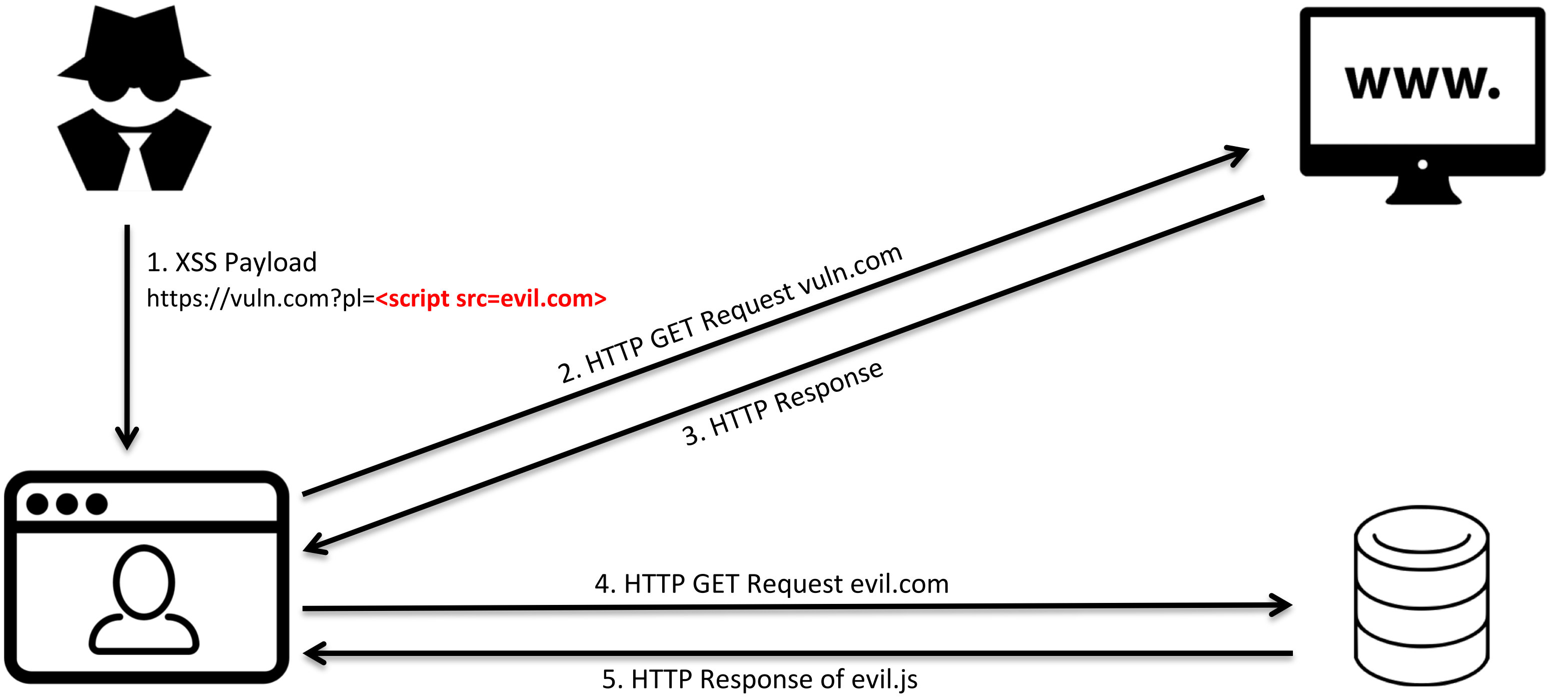
Complex Security Policy?
A Longitudinal Analysis of Deployed Content Security Policies

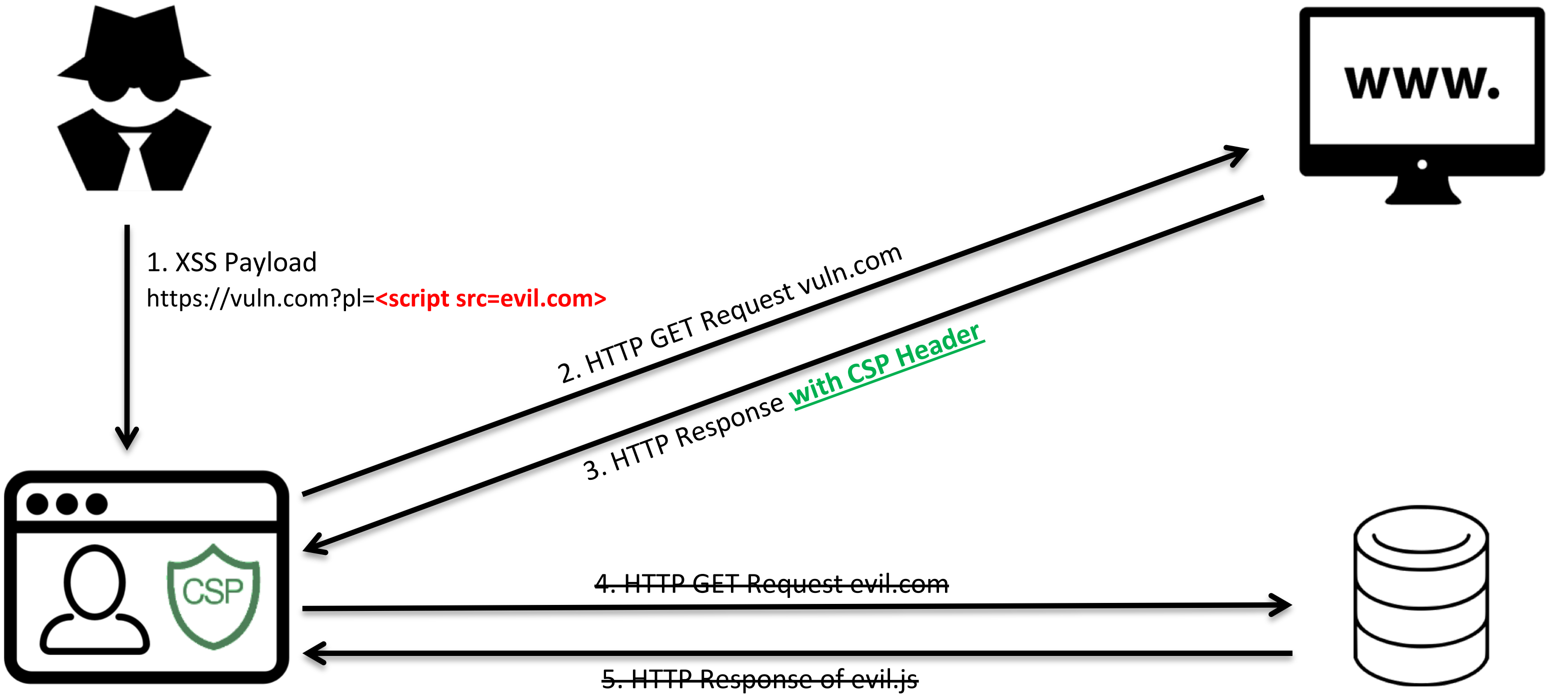
Sebastian Roth, Timothy Barron, Stefano Calzavara, Nick Nikiforakis & Ben Stock

Network and Distributed System Security Symposium (NDSS '20)

Cross-Site Scripting (XSS)


www.
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

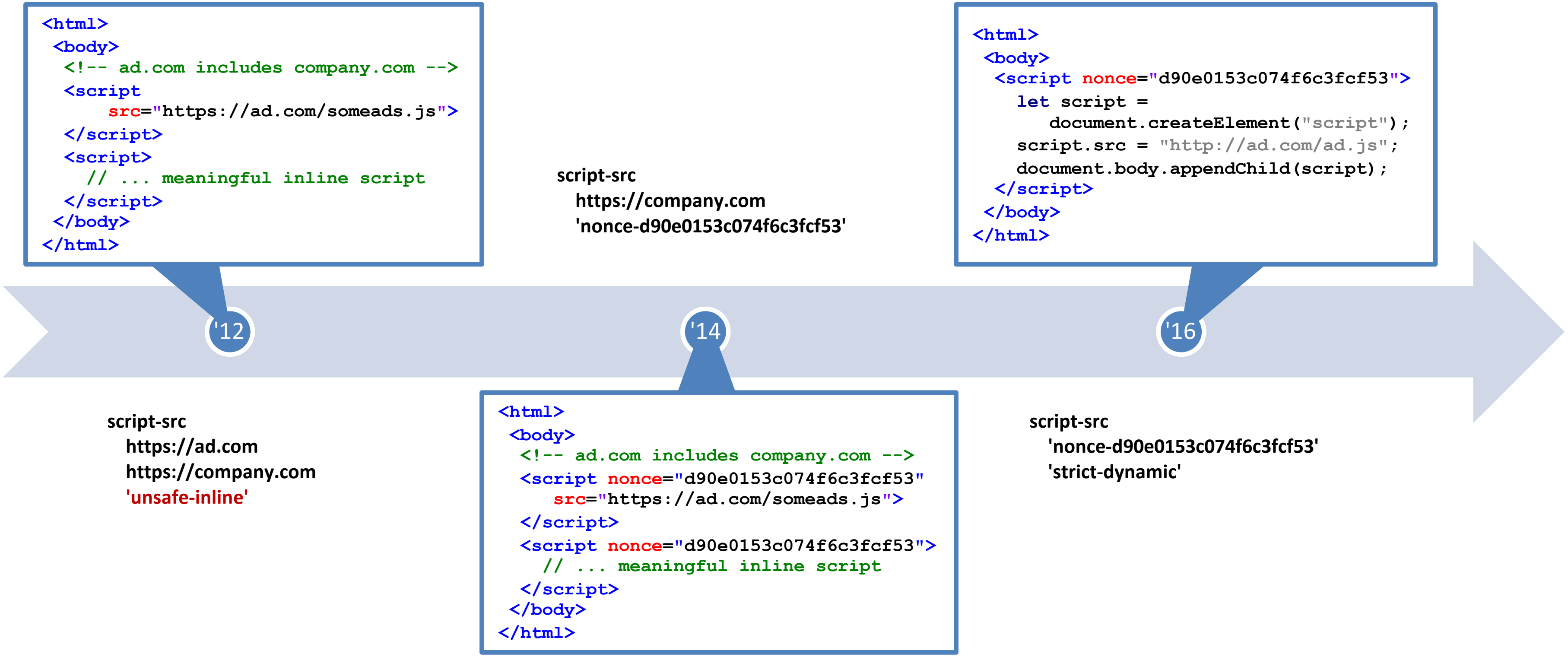
3. HTTP Response **with CSP Header**

~~4. HTTP GET Request evil.com~~

~~5. HTTP Response of evil.js~~

Content Security Policy (CSP)

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY



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

script-src
  https://ad.com
  https://company.com
  'unsafe-inline'

script-src
  https://company.com
  'nonce-d90e0153c074f6c3fcf53'

```
<html>
 <body>
  <!-- ad.com includes company.com -->
  <script nonce="d90e0153c074f6c3fcf53"
        src="https://ad.com/someads.js">
  </script>
  <script nonce="d90e0153c074f6c3fcf53">
    // ... meaningful inline script
  </script>
 </body>
</html>
```

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

script-src
  'nonce-d90e0153c074f6c3fcf53'
  'strict-dynamic'

Research Questions

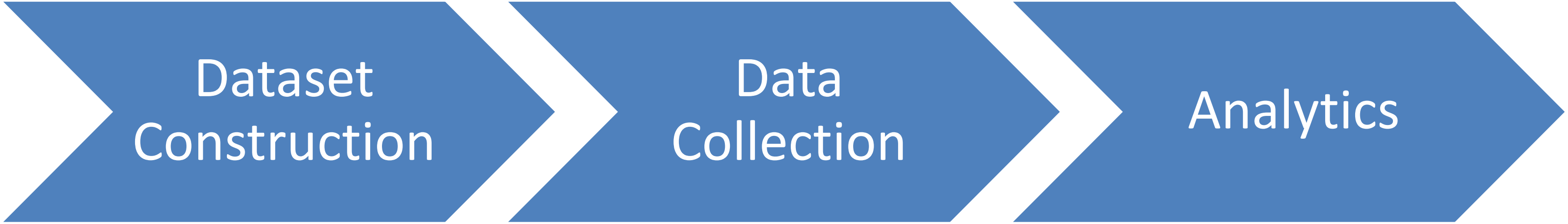
- We know from others studies that:
  - CSP adoption is far behind expectations
  - Many deployed policies are insecure

➢ Why is CSPs adoption so low?

➢ For what purpose is CSP used in the wild?

➢ What are the problems of deploying a CSP?

Methodology

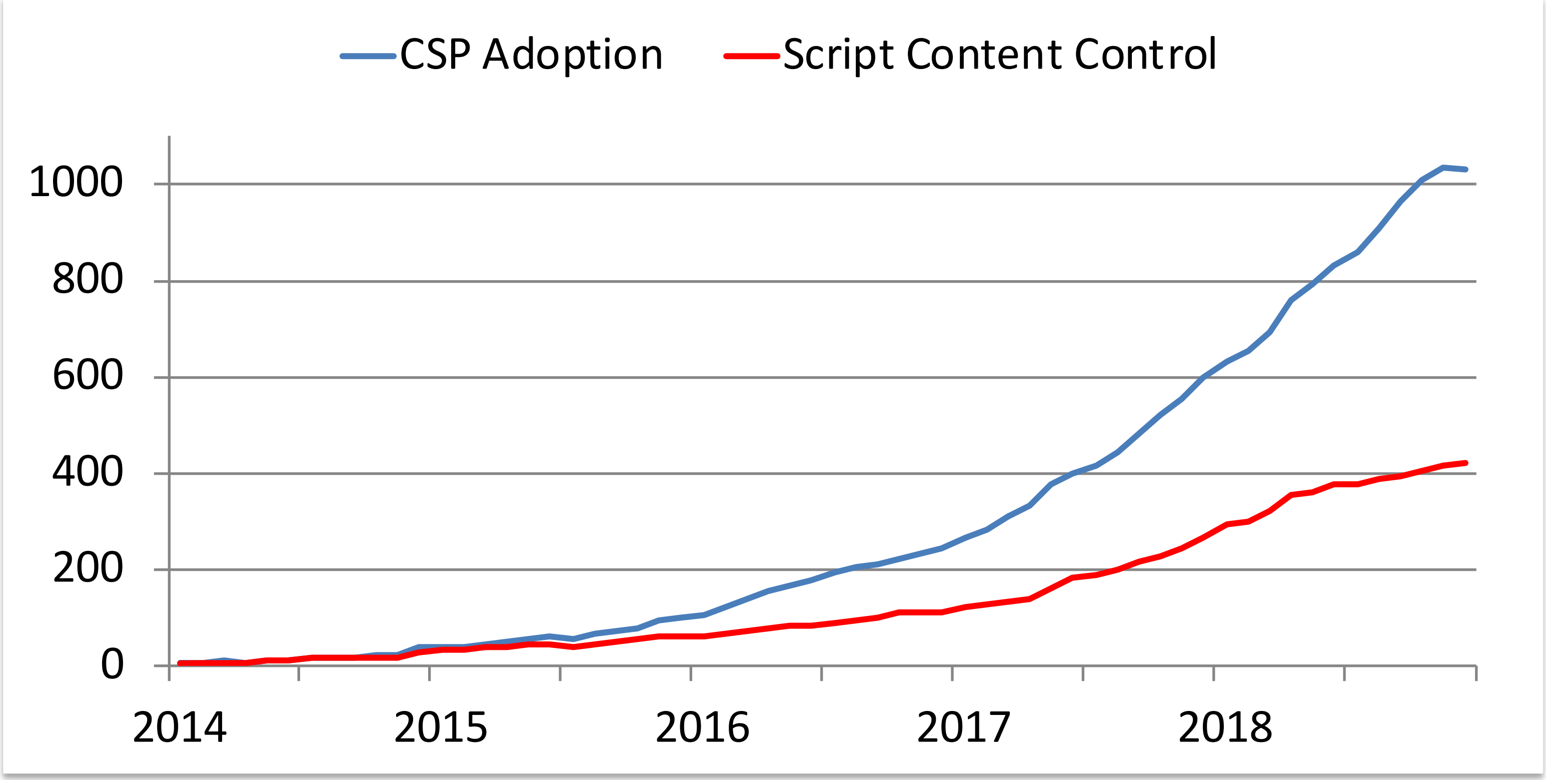


- Create a list of the Top 10k sites over time. Intersection of the Alexa Top sites of each month 2012 – 2018

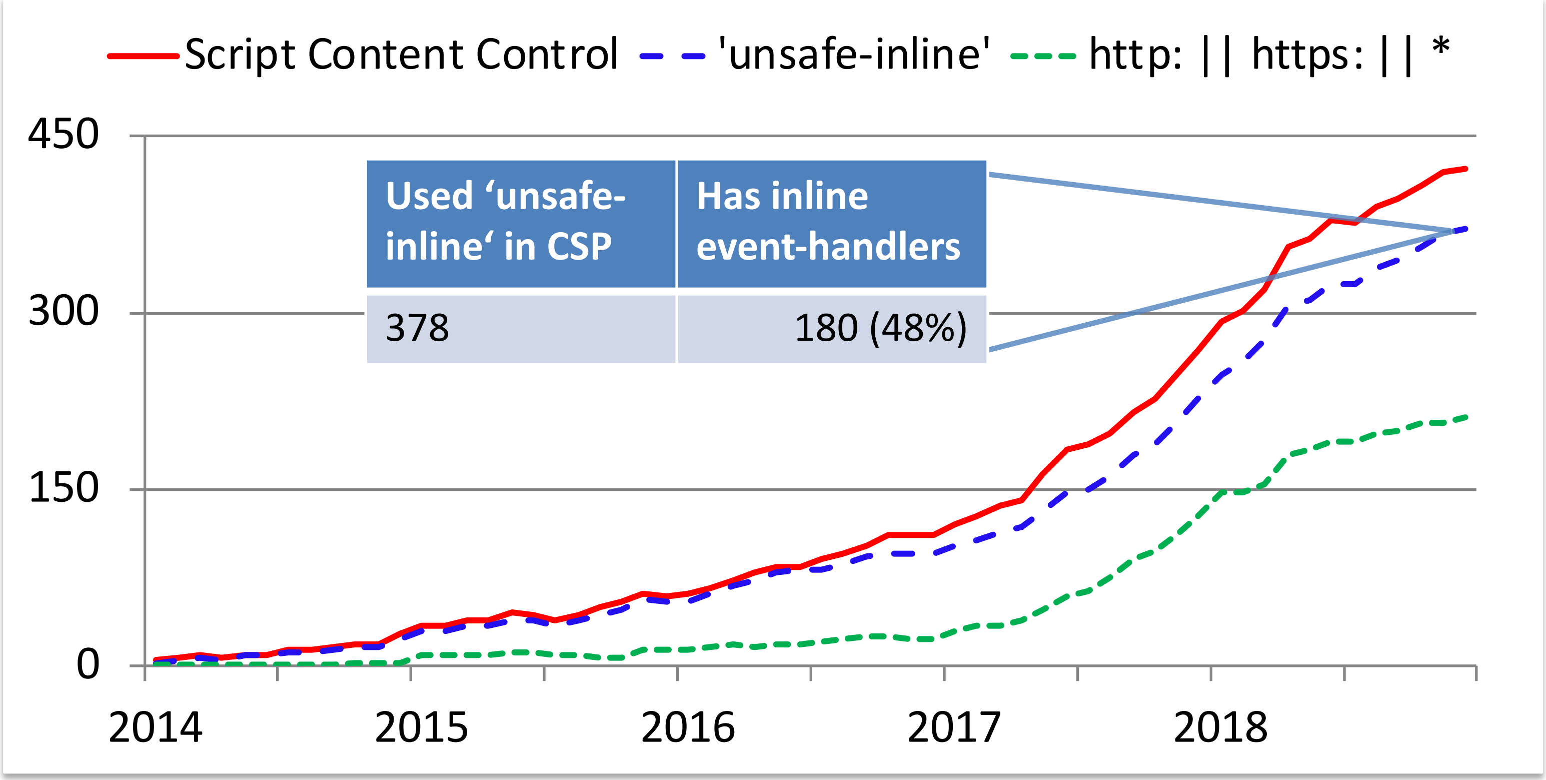
- Use Wayback Machine
- Collected 20,179 CSPs
- Checked Archive Data against Common Crawl

- Classify CSP Use-Cases
- Analyze the Directives and their Use-Cases
- Detailed case-studies & Developer opinions

Use-Case 1: Script Content Control



Use-Case 1: Script Content Control



Used ‘unsafe-inline‘ in CSP	Has inline event-handlers
378	180 (48%)

Script Content Control – Example

Airbnb’s journey to secure their CSP

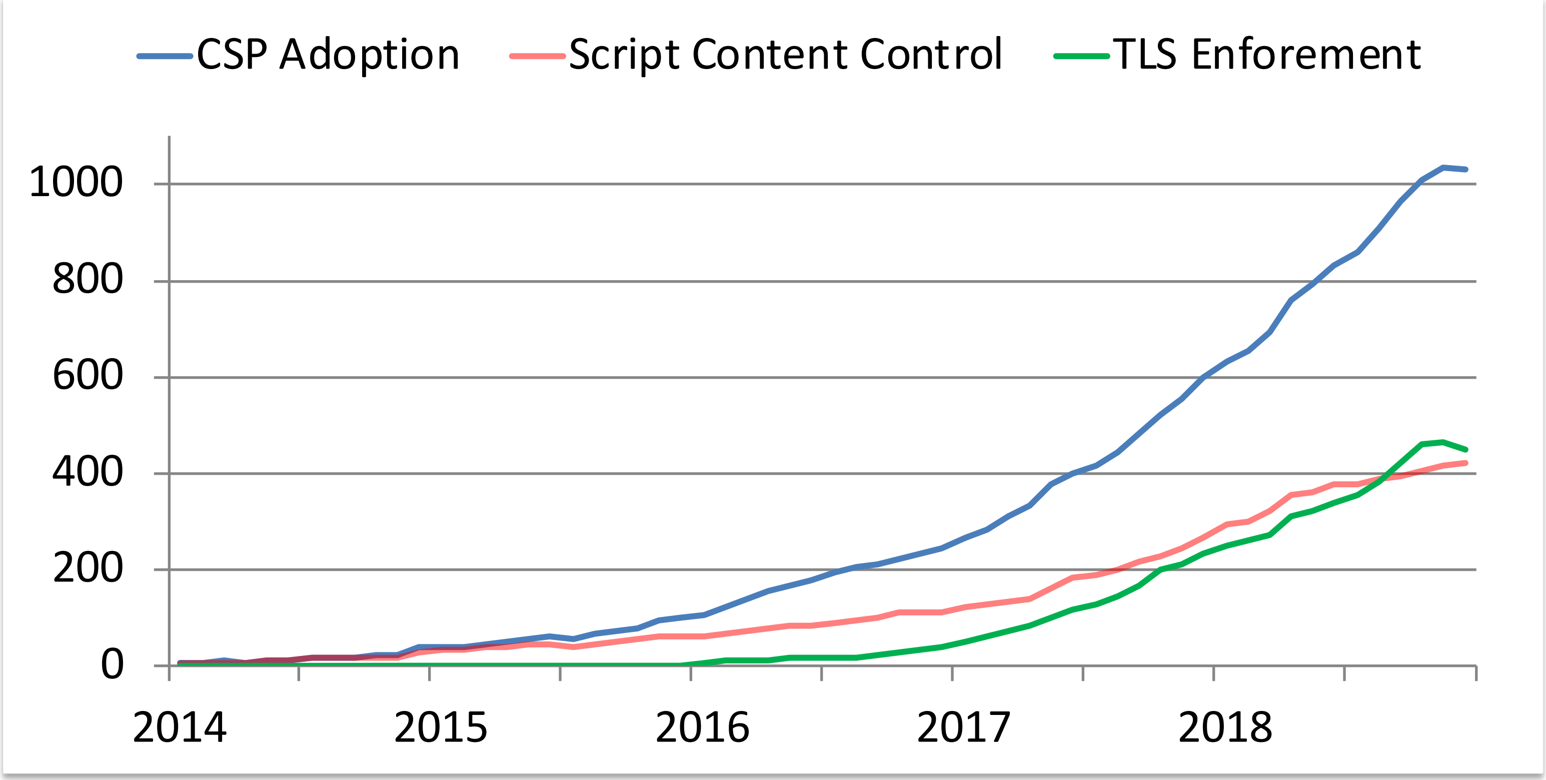
11-2014

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

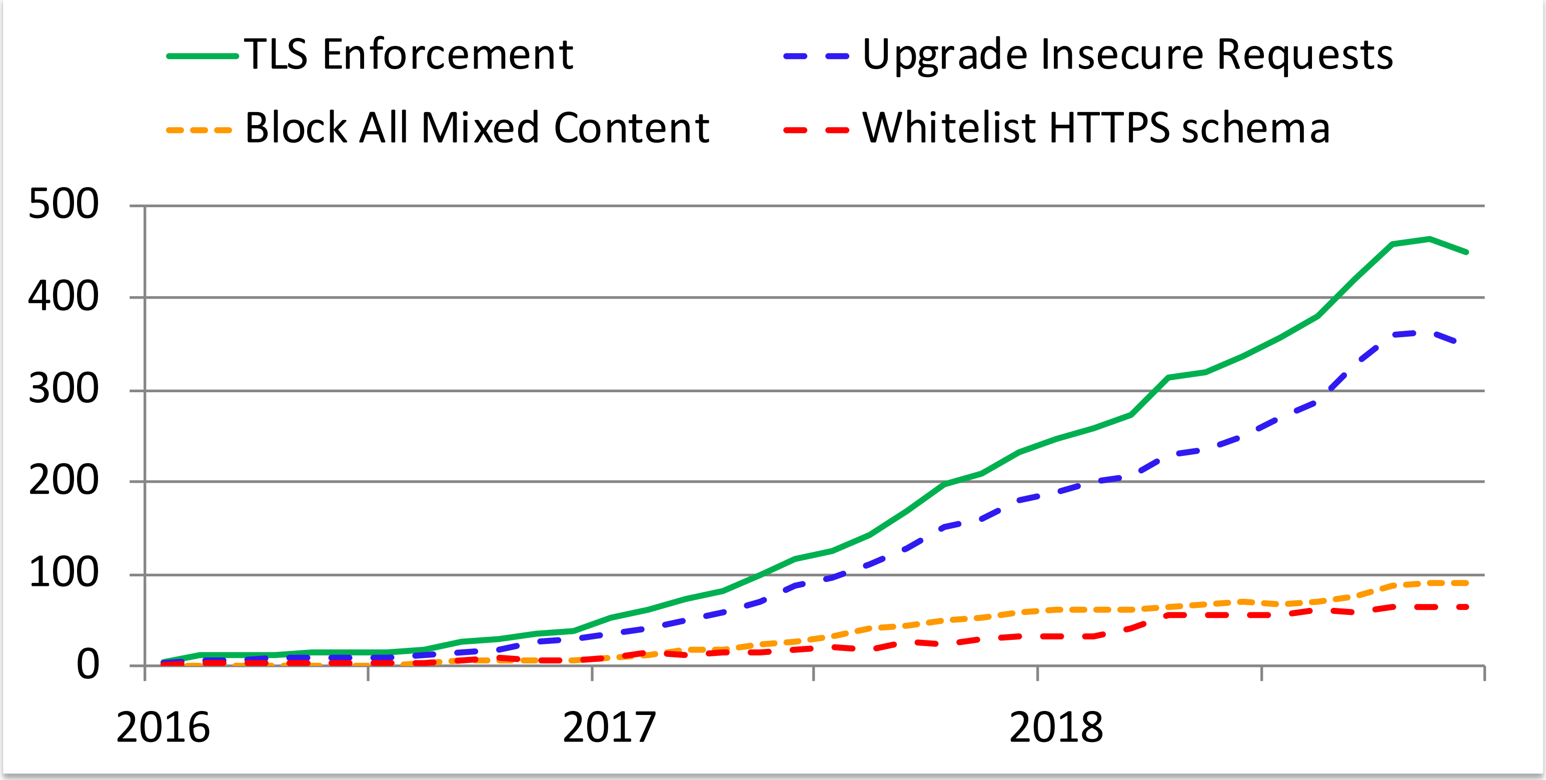
🤨

They needed 3 ½ years to deploy a non-trivially bypassable CSP

Use-Case 2: TLS Enforcement



Use-Case 2: TLS Enforcement

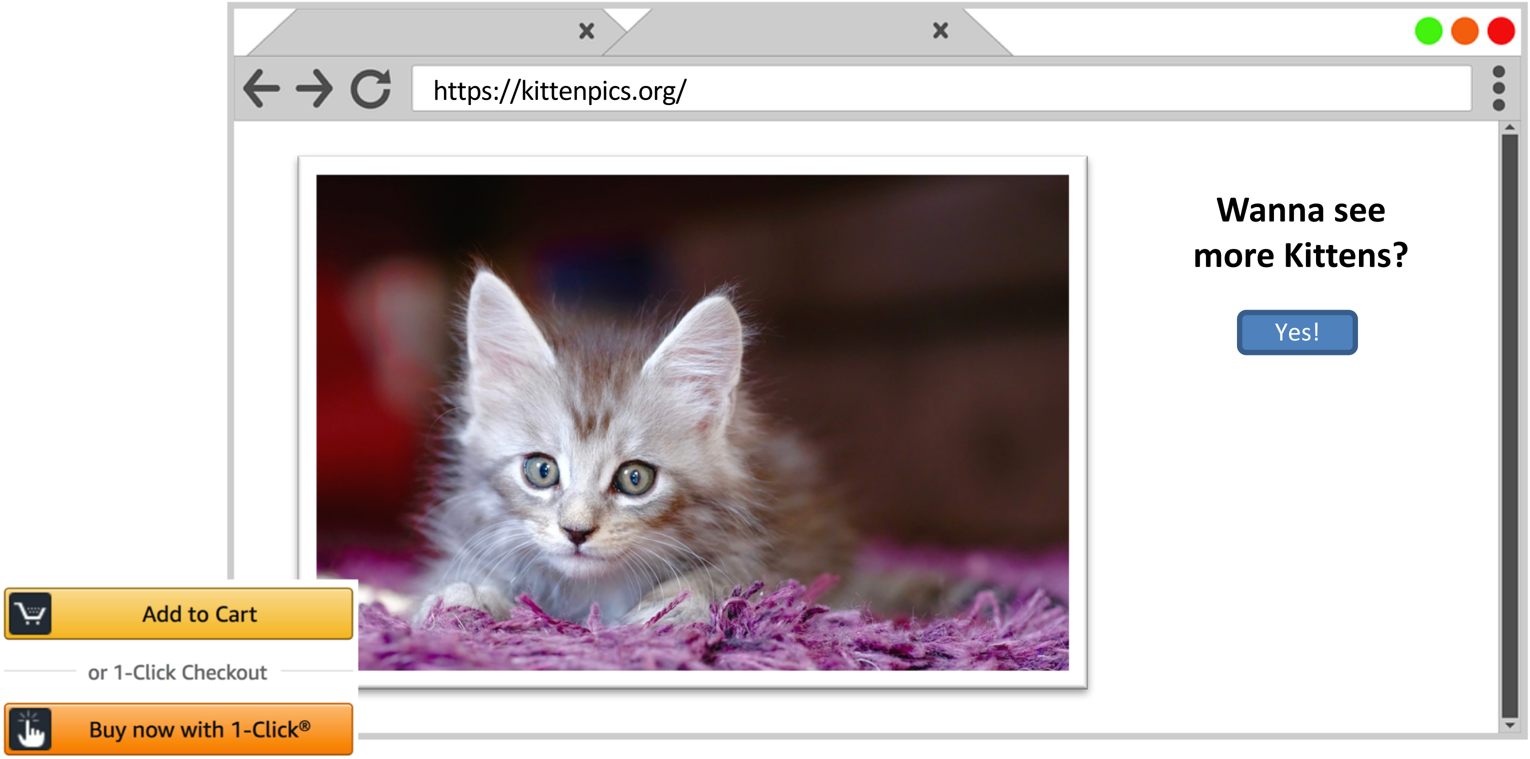


Use-Case 2: TLS Enforcement

- 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



Framing Control – X-Frame-Options

X-Headers are not standardized!

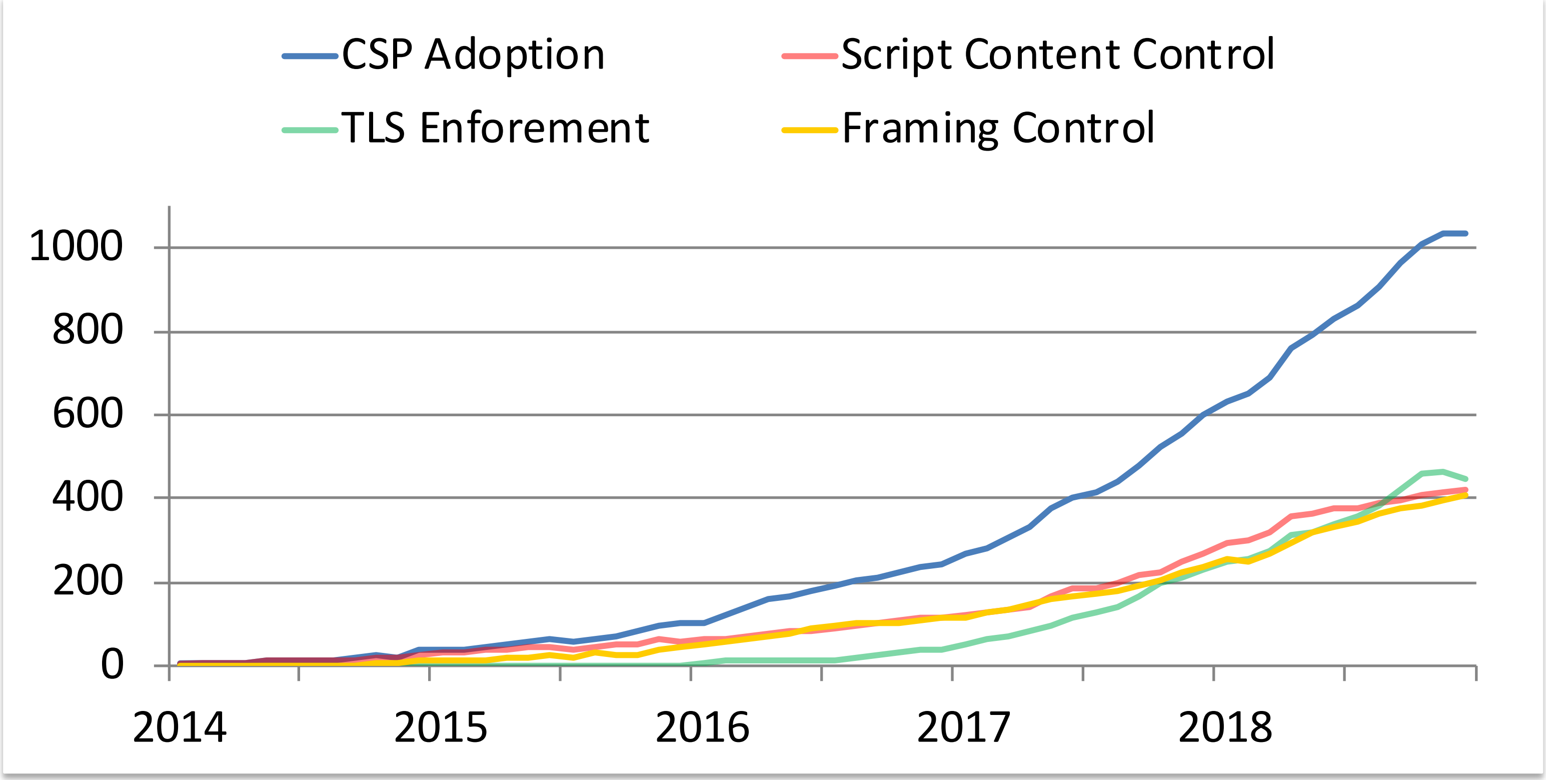
Leads to security problems:

- Partial support
- Double Framing

... as well as functionality problems

- X-Frame-Options can only have a single whitelist entry

Use-Case 3: Framing Control



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:**

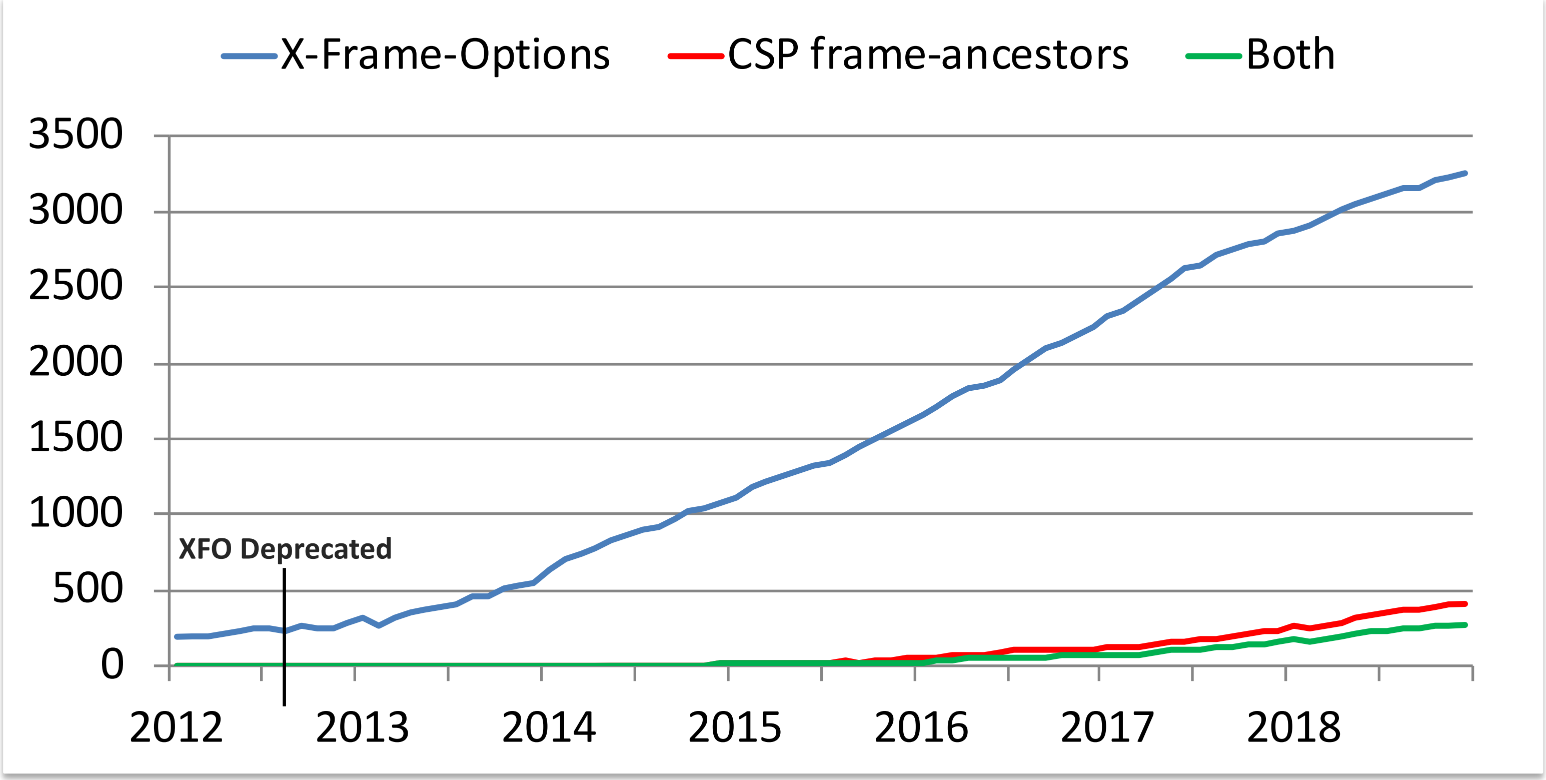
  Applies to all of a frame's ancestors not only the top-most frame.

- **Explicit whitelist:**

  frame-ancestors supports wildcards and multiple source-expressions

  ```
  frame-ancestors www.foo.com ‘self’ *.partner.com
  ```

Use-Case 3: Framing Control



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!

Developer Study

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. “

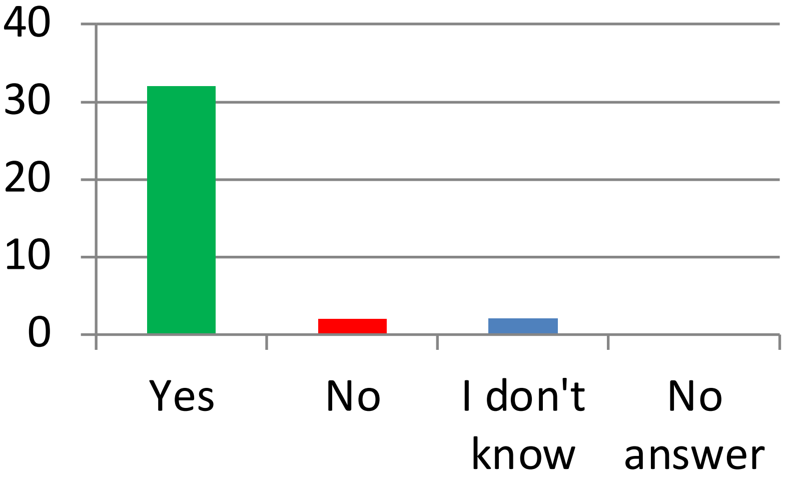
Developer Study

Misconceptions about CSP

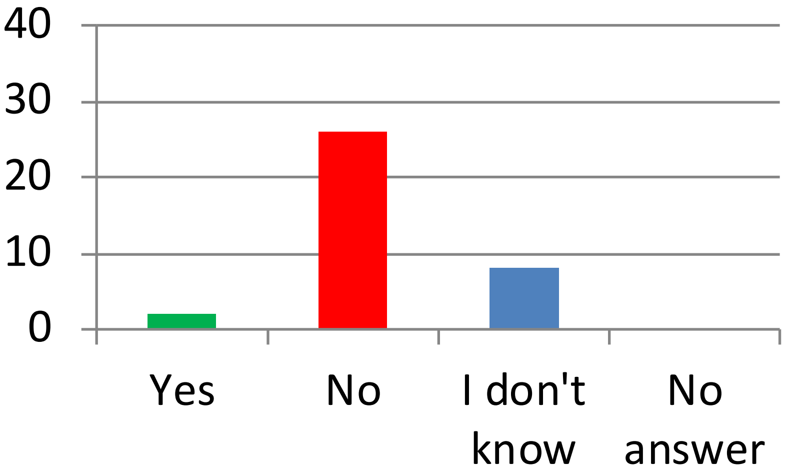
> ” CSP is a complex beast [...]. Some of our partner are iframing our site. We already had issue to implement the X-Frame header, that we did not want to deal with CSP. “

Framing Control – Developer Study

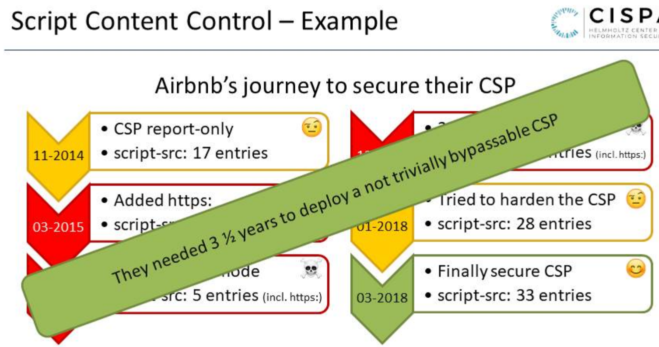
Do you believe CSP is a viable option to improve your site's resilience against XSS attacks?



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



Complex Security Policy?

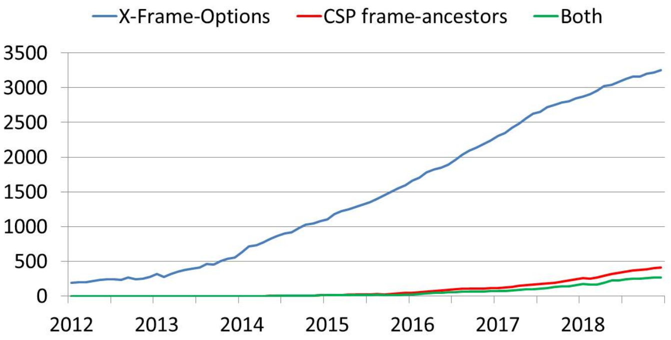


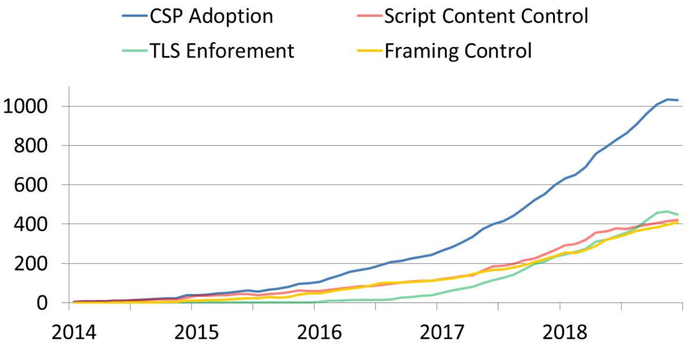
Script Content Control – Example

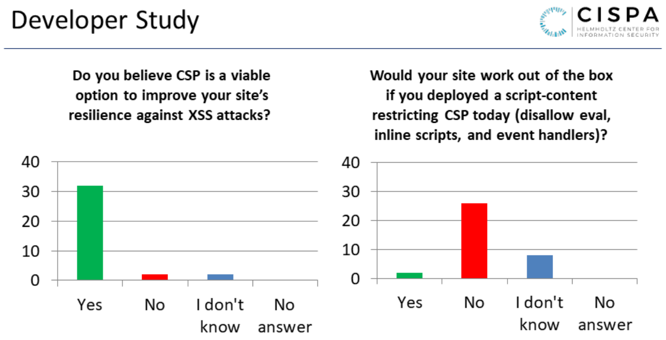
Airbnb's journey to secure their CSP

- 11-2014
  - CSP report-only
  - script-src: 17 entries
- 03-2015
  - Added https:
  - script-s
- ode
  - src: 5 entries (incl. https:)
- 1
  - 3
  - ntries (incl. https:)
- 01-2018
  - Tried to harden the CSP
  - script-src: 28 entries
- 03-2018
  - Finally secure CSP
  - script-src: 33 entries

They needed 3 ½ years to deploy a not trivially bypassable CSP





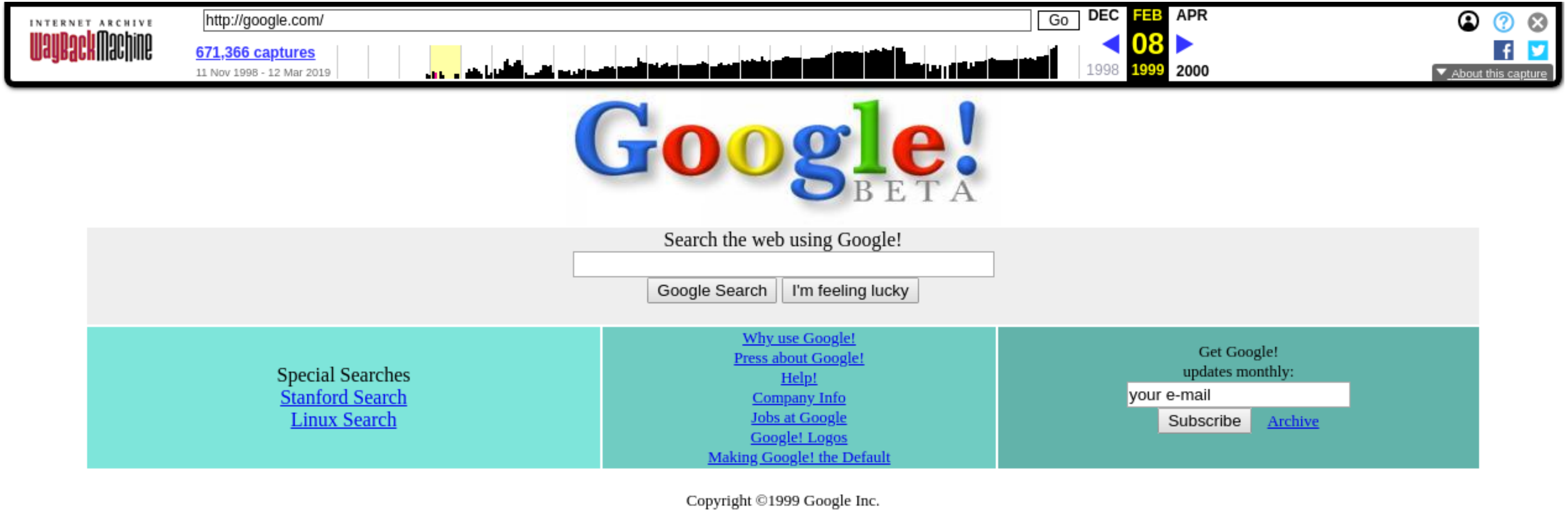


Developer Study

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

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

How to go back in time?



Search the web using Google!

Special Searches
Stanford Search
Linux Search

Why use Google!
Press about Google!
Help!
Company Info
Jobs at Google
Google! Logos
Making Google! the Default

Get Google!
updates monthly:

Copyright ©1999 Google Inc.

Also stores original HTTP headers prefixed with `X-Archive-Orig-`

Appendix – Developer Study

Building a CSP requires massive effort

> ” 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. “

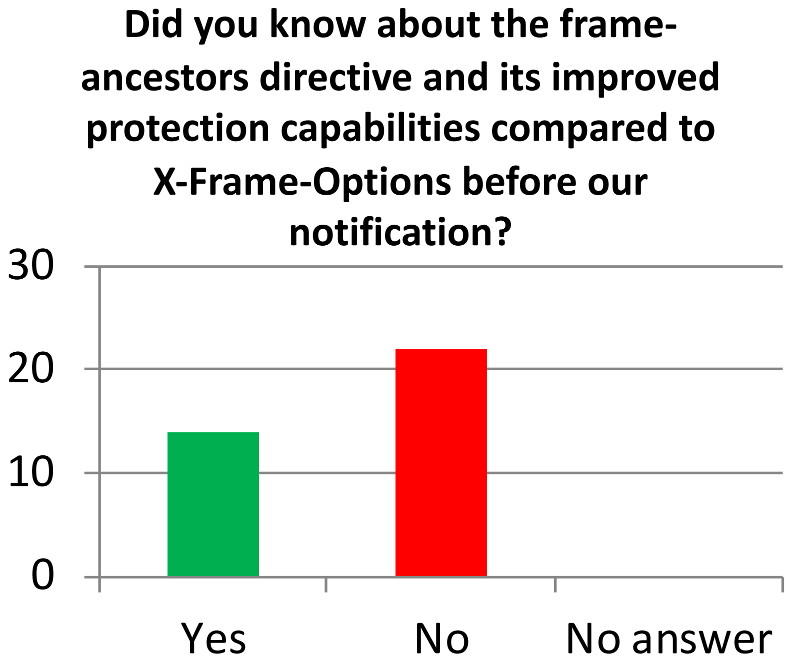
Appendix – Developer Study

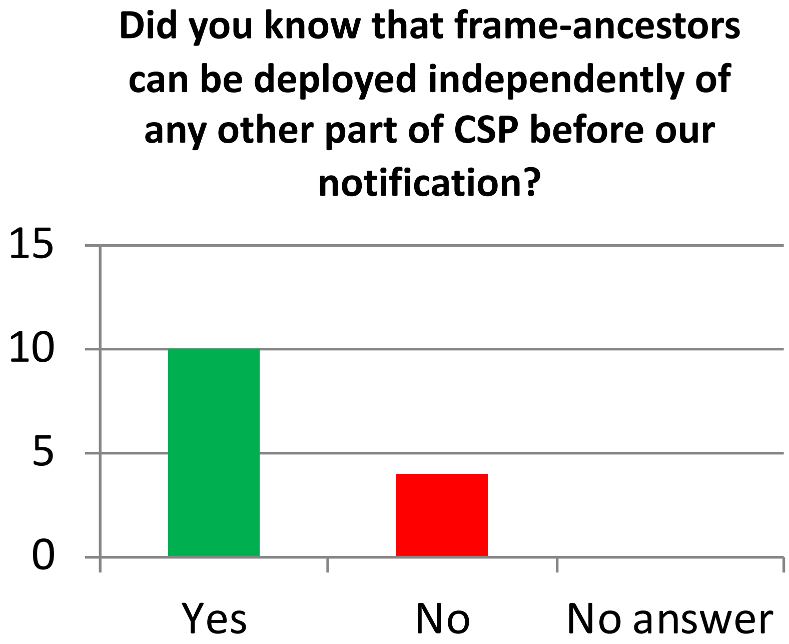
CSP is too complex to deploy

> ” […] 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. “

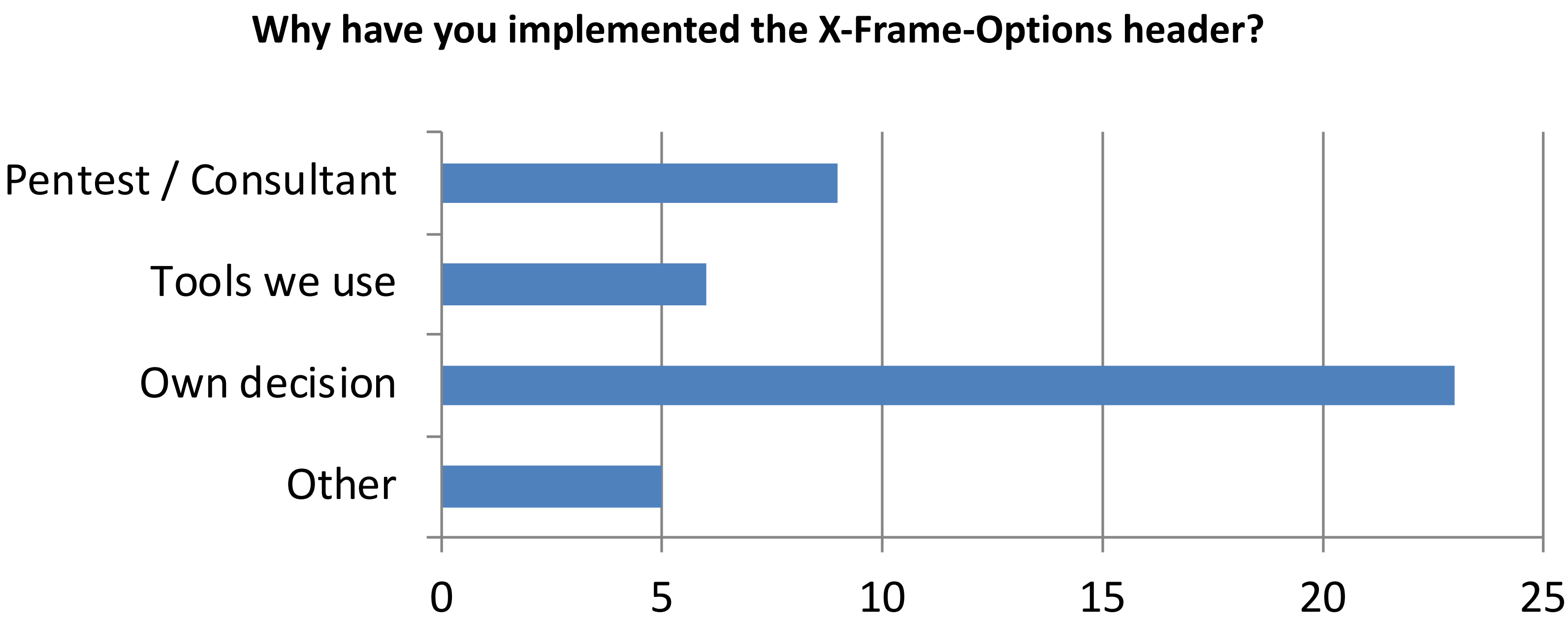
Developer Study

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY





Appendix – Developer Study



Appendix – Good CSP Deployment

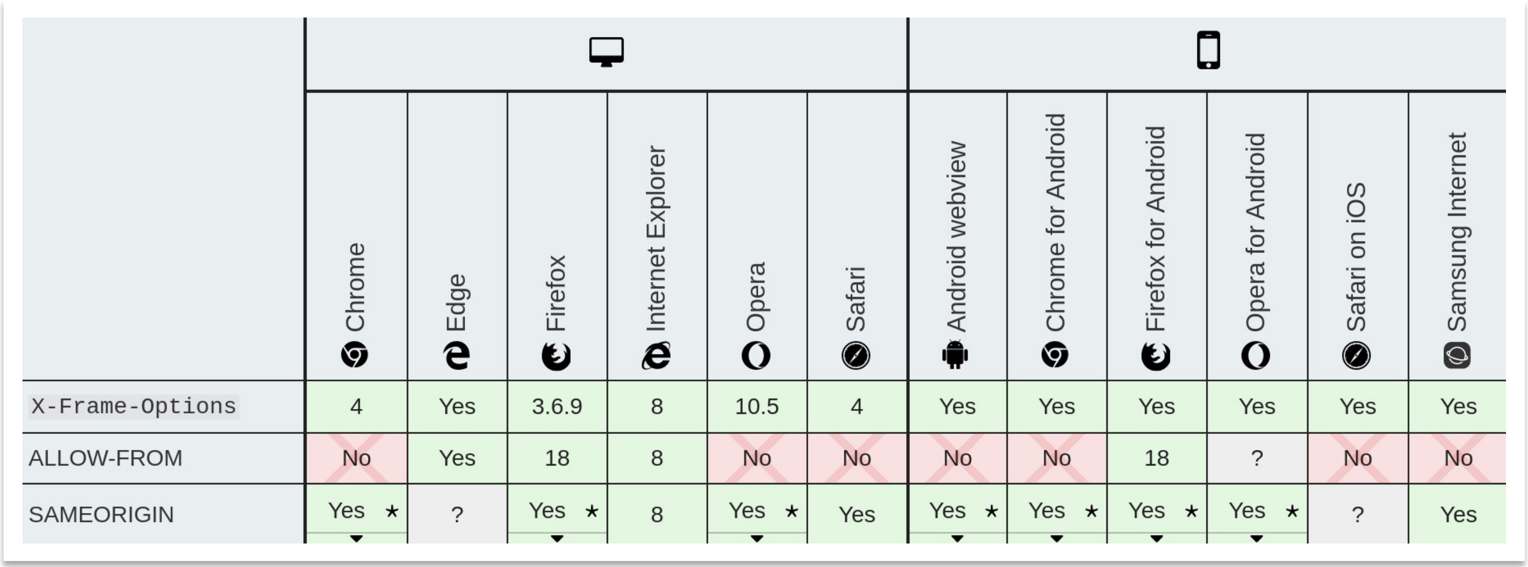
GitHub’s journey to secure their CSP

11-2013

- Started to use CSP in Enforcement Mo
- script-src contains 5 entries (S)

They **never** ever used any dangerous source expression!

Appendix – Partial support

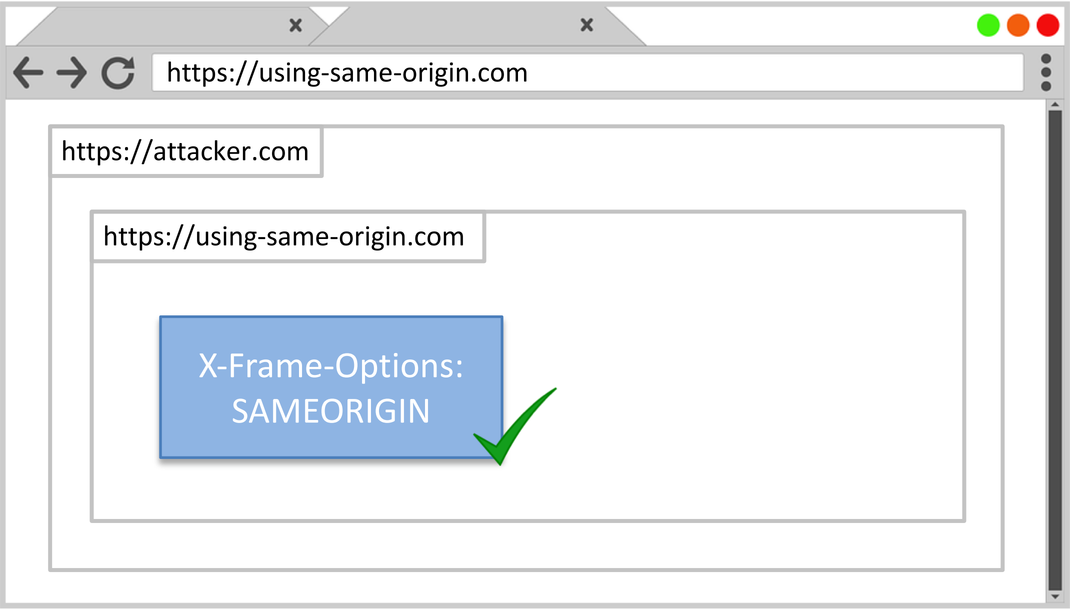


	🖥						📱					
	Chrome	Edge	Firefox	Internet Explorer	Opera	Safari	Android webview	Chrome for Android	Firefox for Android	Opera for Android	Safari on iOS	Samsung Internet
`X-Frame-Options`	4	Yes	3.6.9	8	10.5	4	Yes	Yes	Yes	Yes	Yes	Yes
ALLOW-FROM	No	Yes	18	8	No	No	No	No	18	?	No	No
SAMEORIGIN	Yes ★	?	Yes ★	8	Yes ★	Yes	Yes ★	Yes ★	Yes ★	Yes ★	?	Yes

=> ALLOW-FROM fails insecurely for Chrome & Co. *

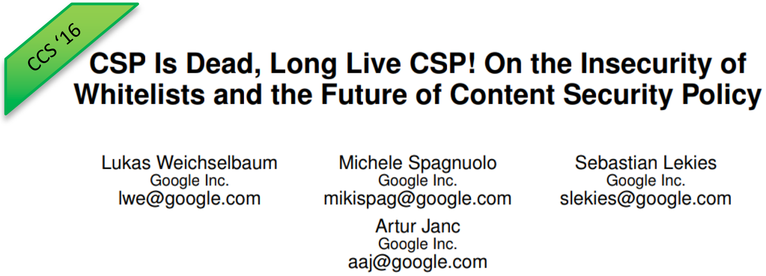
* Meanwhile, since Firefox 70, ALLOW-FROM is no longer supported.

Appendix – Double Framing


https://using-same-origin.com
https://attacker.com
https://using-same-origin.com
X-Frame-Options: SAMEORIGIN
✓

=> In legacy browsers XFO is only checked against top-most frame.

Appendix – Related Work



CCS '16

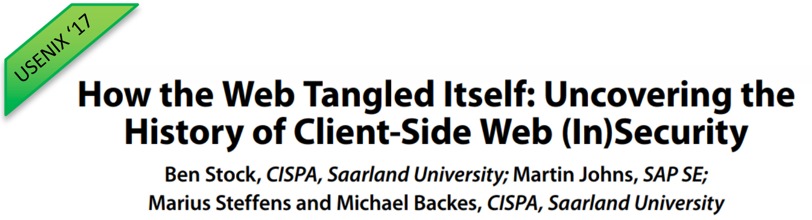
CSP Is Dead, Long Live CSP! On the Insecurity of Whitelists and the Future of Content Security Policy

Lukas Weichselbaum
Google Inc.
lwe@google.com

Michele Spagnuolo
Google Inc.
mikispag@google.com

Sebastian Lekies
Google Inc.
slekies@google.com

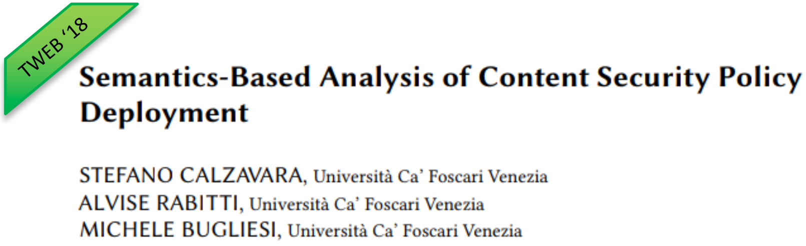
Artur Janc
Google Inc.
aaj@google.com



USENIX '17

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

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



TWEB '18

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