Northeastern University

Systems Security Lab

Why is CSP Failing? Trends and Challenges in CSP Adoption

Symposium on Research in Attacks, Intrusions and Defenses (RAID)
Gothenburg, Sweden, September 2014

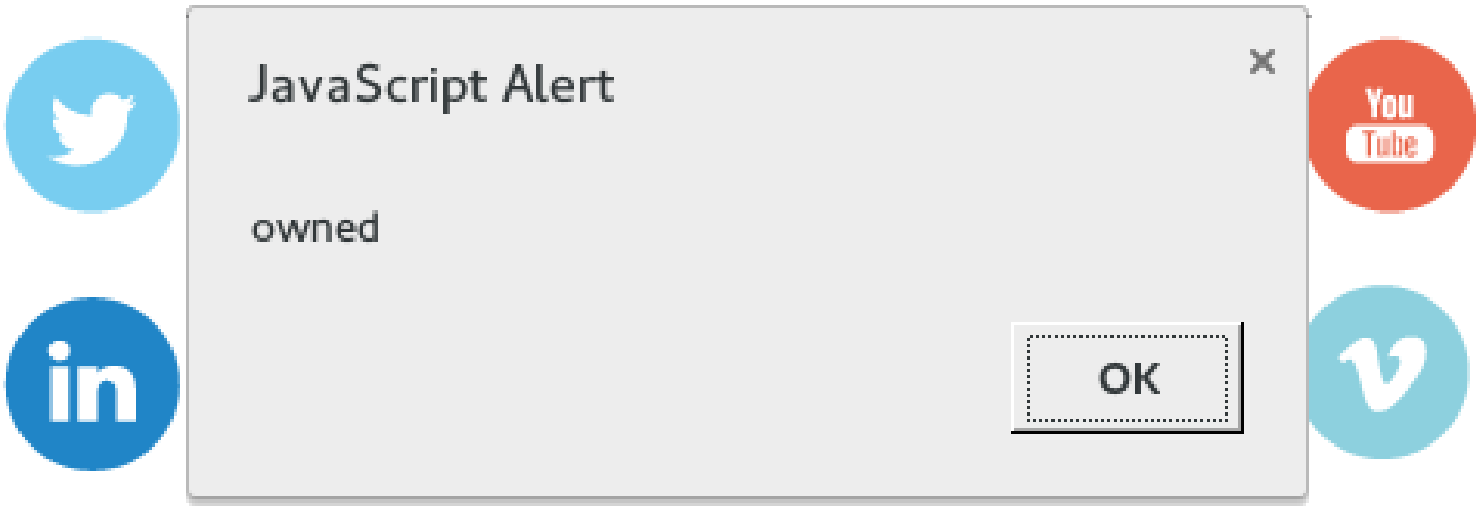
Michael Weissbacher, Tobias Lauinger, William Robertson

{ mw, toby, wkr }[at]ccs.neu.edu

NEU SECLAB

The World Wide Web...

The World Wide Web...



Content Security Policy (CSP) - Overview

- Client-side (browser) policy enforcement
- Document-specific whitelist of resources
- Prevention of site modifications and data leaks
- Two modes of operation
- Recent standard
- Supported by all major browsers
- Delivery as HTTP response header (or meta tag)

CSP Example

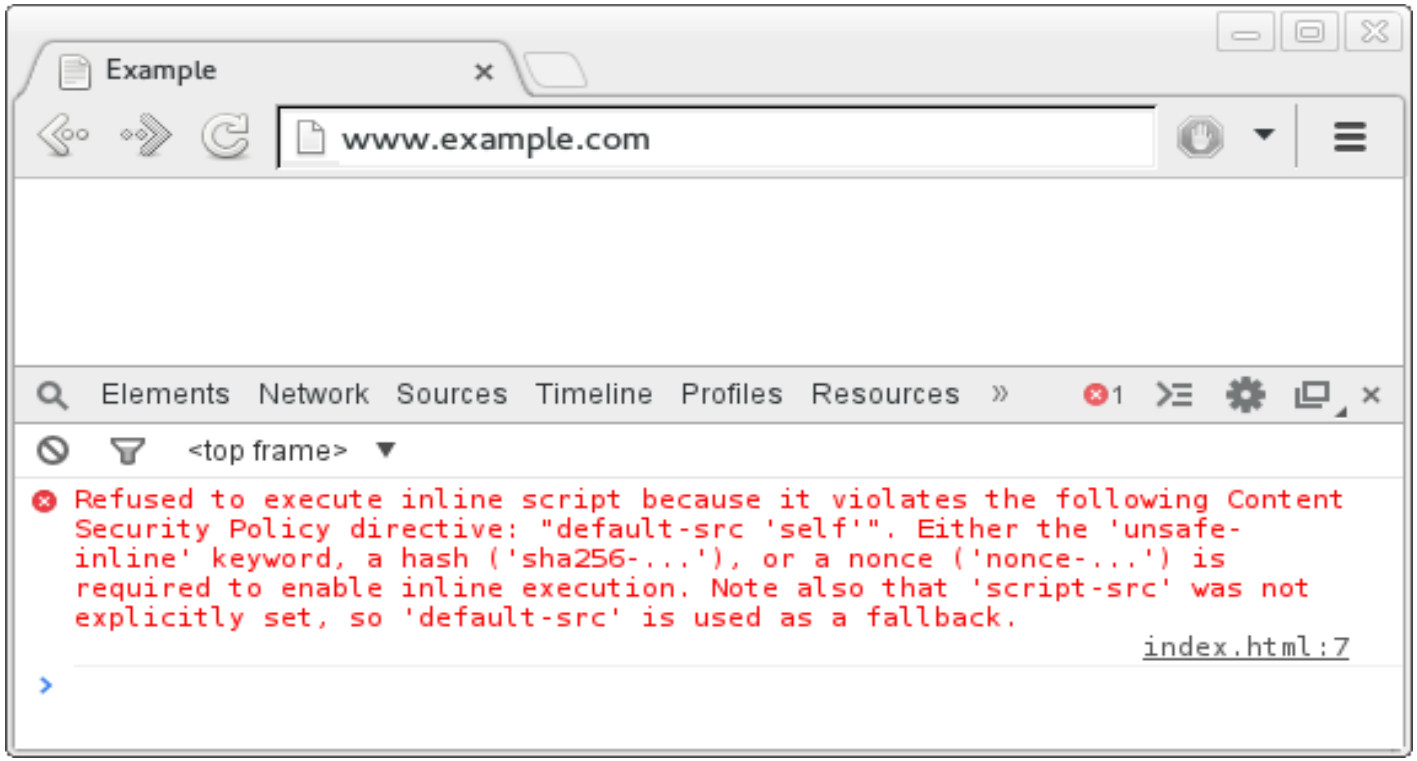
```
$ curl -I http://www.example.com |  grep -i content-
security-policy:

Content-Security-Policy: default-src 'self';
script-src 'static.example.com';
style-src 'static.example.com';
report-uri http://example.com/reports
```

[Line breaks added for clarity]

How to Deploy CSP

- Move inline script to files
- Move resources to separate hosts, e.g., static.example.com
- No eval constructs



“CSP is a great defense – how does it impact the Web?”

Overview

- Study on Content Security Policy
  - Who is using CSP ?
  - Can we improve the state of adoption ?

- Contributions
  - First long-term analysis of CSP adoption in the wild
  - Investigate challenges in adopting CSP
  - Evaluate the feasibility of incremental deployments
  - Suggest avenues for enhancing CSP

“Who is using Content Security Policy,
and how are they using it?”

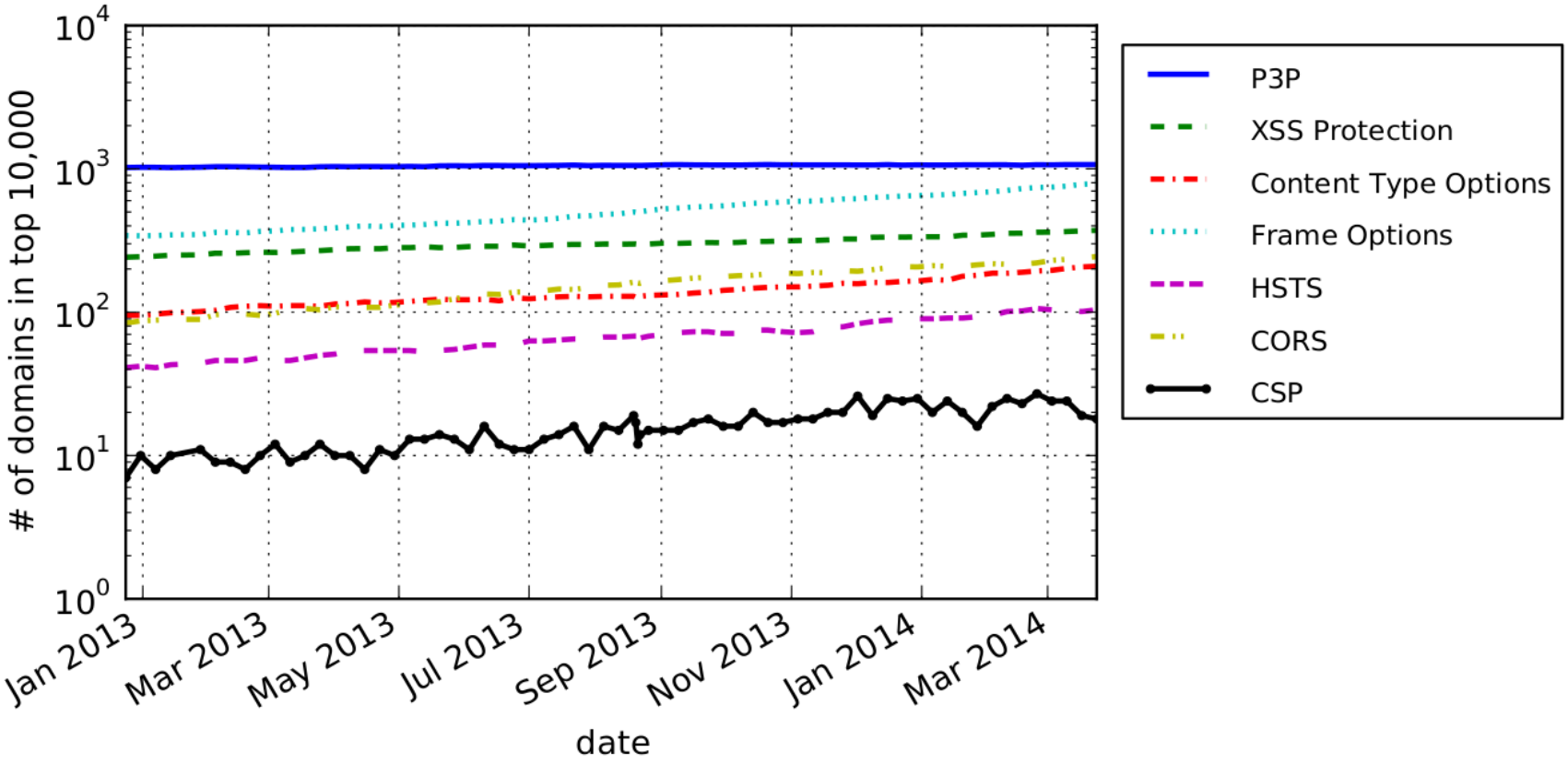
Data Collection Methodology

- Crawl of Alexa Top 1M
- Weekly, starting December '12
- For every domain x.com, connect to:
  - http://x.com
  - https://x.com
  - http://www.x.com
  - https://www.x.com
- Using up to date Firefox User Agent
- We only access the front page

Security Headers We Collected

- Platform for Privacy Preferences (P3P)
- DNS Prefetch Control
- XSS Protection
- Content Type Options
- Frame Options
- HTTP Strict Transport Security (HSTS)
- Cross-Origin Resource Sharing (CORS)

Security Headers Dec '12 – April '14



CSP is the least popular security HTTP response header

Top 1M

- Top 1M snapshot of March '14
- Split into logarithmic brackets, 1-10, 11-100 etc.
- CSP statistics
  - Top 100: 2%
  - 900,000 least popular sites: 0.00086%
- In comparison, CORS:
  - All brackets between 0.7% and 2.6%
- More popular sites are more likely to adopt CSP
- That likelihood is still low

Policy Overview

- Enforcement mode: 815 (23 collecting reports)
- Report-only: 35
- Learned rules: dnb.no, 74 sources
- Observed policies add only marginal benefit
  - Mostly unsafe-inline, eval

“Can we improve adoption by generating rules automatically?”

How To Create Policies

- Manual
- Violation reports by users
- Violation reports by our crawler

Violation Reports

- Policy: img-src 'none'; report-uri /sink.cgi
- http://seclab.nu/test.html
- Includes resource: http://seclab.nu/pic.gif
- JSON Report:


```
{
  "blocked-uri": "http://seclab.nu/pic.gif",
  "violated-directive": "img-src 'none'",
  "document-uri": "http://seclab.nu/test.html"
}
```

Derive CSP Through report-only Mode

- Add headers to four websites
- Disallow all resources
- Collect violation reports
- Derive one whitelist usable for the entire site

Policy For Simple Website

- Expected policy

```
default-src 'none'; img-src 'self' data:; style-src 'unsafe-inline'
```

Policy For Simple Website

- Expected policy

```
default-src 'none'; img-src 'self' data:; style-src 'unsafe-inline'
```

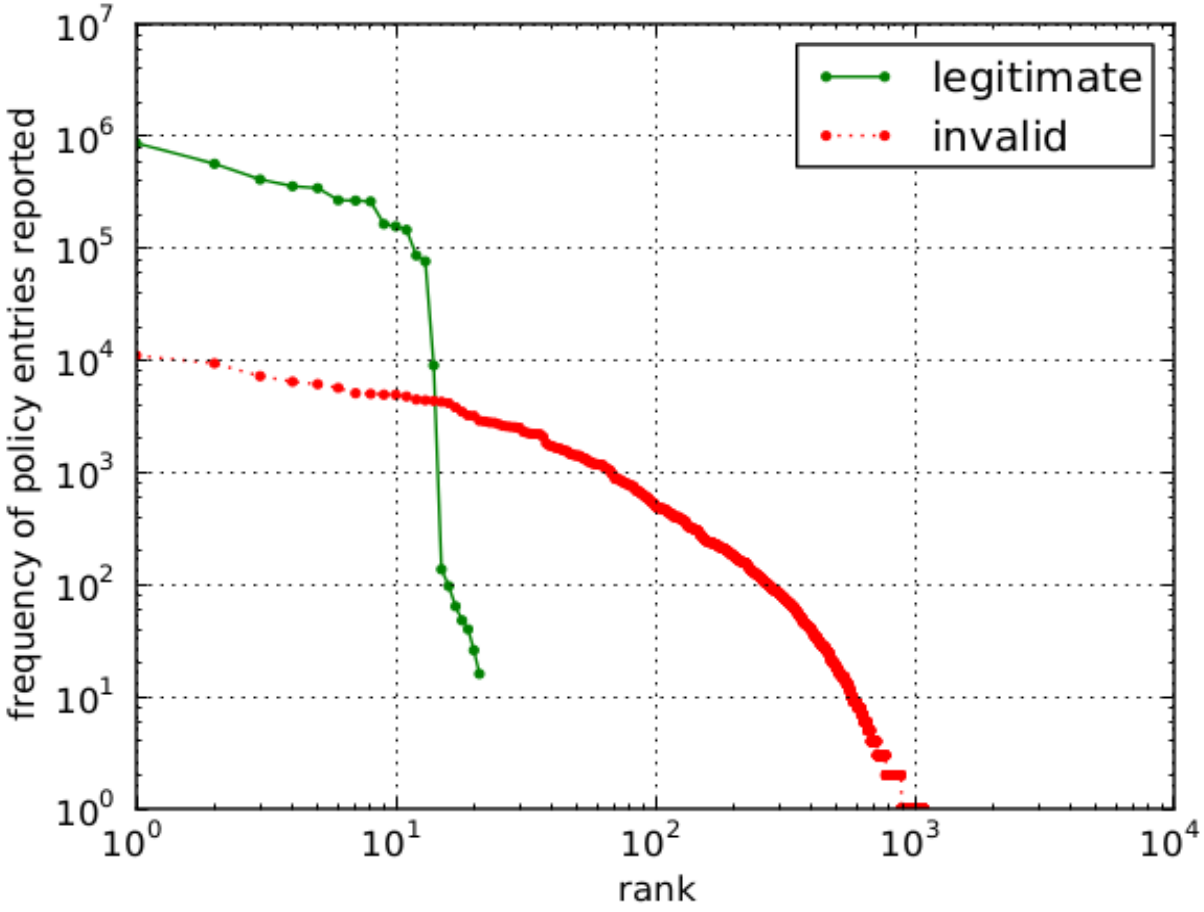
- Derived policy

```
default-src `none';
frame-src https://srv.mzcdn.com;
img-src 'self' data: http://1.2.3.11;
object-src http://www.ajaxcdn.org;
script-src 'unsafe-eval' 'unsafe-inline' http://ajax.googleapis.com
http://f.ssfiles.com http://i.bestoffersjs.info
http://srv.mzcdn.com http://www.superfish.com
https://www.superfish.com;
style-src 'unsafe-inline'
```

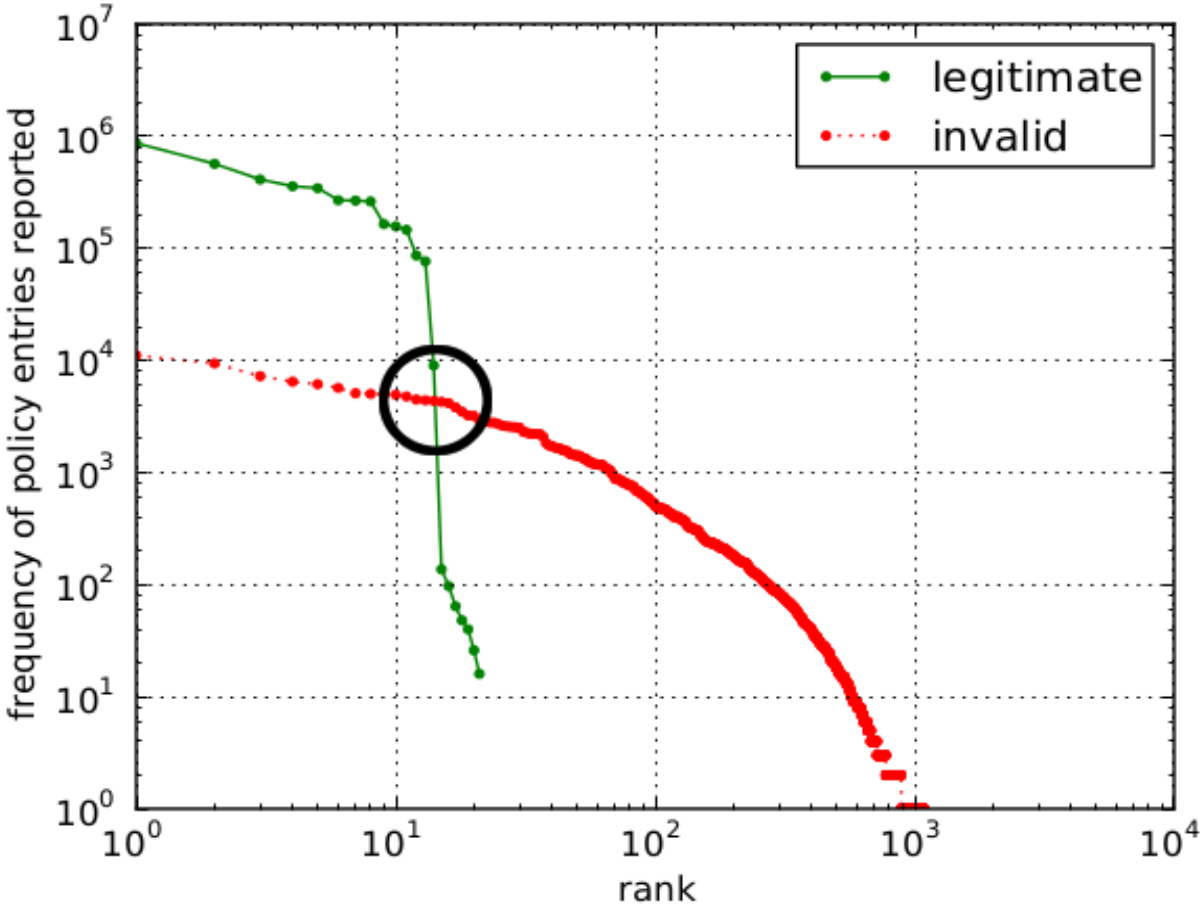
Sources of Noise

- Chrome Extensions
  - AdBlock
  - AdBlock Plus
  - Grooveshark Downloader
  - (Extensions can modify policies)
- Modifications in flight
  - Mobile ISPs
- Can these be filtered automatically, e.g.: by frequency?

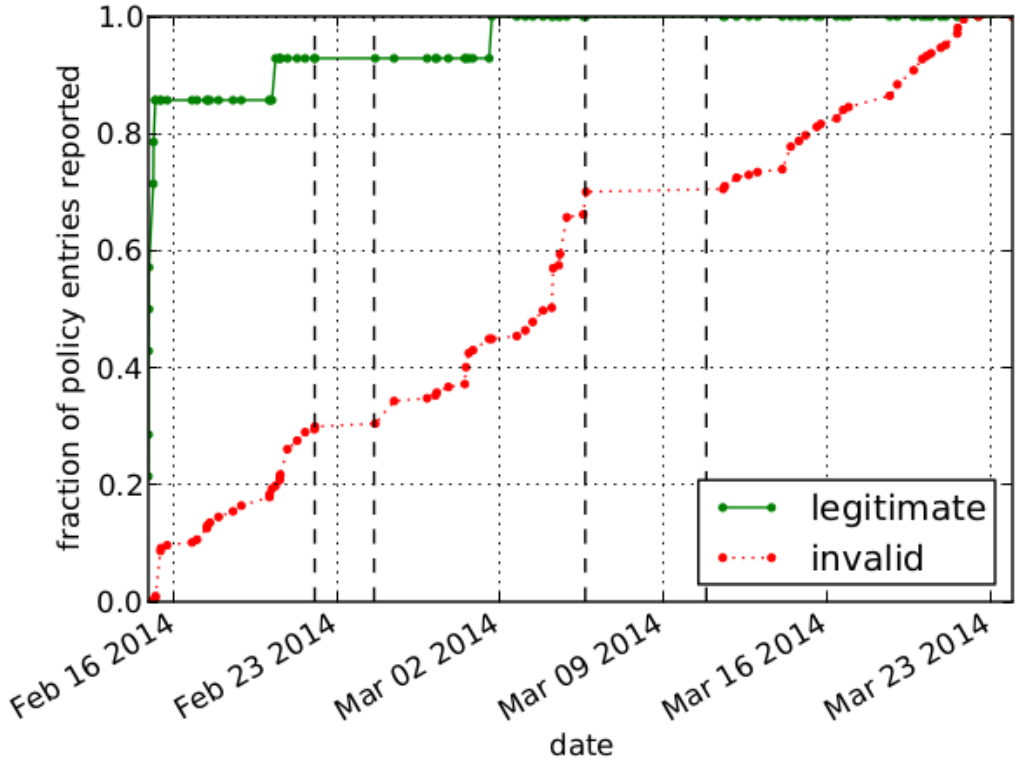
Policy Frequency Comparison



Policy Frequency Comparison



New Policy Entries Discovered Over Time



Derive CSP Through Trusted Crawler

- Approach
  - Use Chrome, driven through browser extension
  - Rewrite HTTP response header in proxy to generate violations
  - Catch violation reports in proxy
  - No semantic gap to other browsers
- Targets
  - Four previous websites
  - Sites that use CSP as ground truth
  - Complex websites

Crawler Overview

- How stable are policies over time?
- How secure are the policies?
- Evaluation
  - Compare to manual browsing
  - Use sites with CSP as ground truth

Crawler Results

- Stability depends on architecture
  - Integration of ads: CNN vs. BBC
  - Separation of content: Twitter
- Security
  - Most sites: eval, inline
  - Benefits: less data leaks

Open Questions

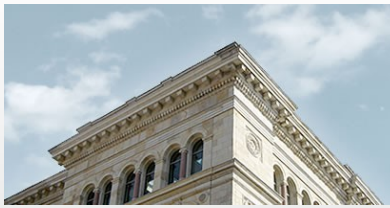
- Can the approaches be combined?
  - High volume, untrusted
  - Low volume, trusted
- Can machine learning help?
- Improve the crawler?

Discussion

- Discussions with engineers
  - Websites want to keep inline script
  - Changes to legacy code are expensive
  - Enforcement over extensions is considered a bug
- Suggested improvements
  - Isolate ads
  - Adoption by frameworks anticipated
  - Browsers should not enforce CSP on extensions
- Vendor Response
  - PHPMyAdmin
  - Facebook
  - GitHub

Conclusions

- CSP adoption significantly lags other headers
- If deployed, benefits are marginalized
- Automatically deriving rules is hard
- CSP holds promise, hopefully v1.1 will address shortcomings



EOF

Thank you!

Questions?

Twitter: @mweissbacher