A Tale of Two Headers: A Formal Analysis of Inconsistent Click-Jacking Protection on the Web

Stefano Calzavara+, Sebastian Roth*, Alvise Rabitti+, Michael Backes* & Ben Stock*

+ Università Ca’ Foscari
* CISPA Helmholtz Center for Information Security

USENIX Security Symposium (USENIX Security ’20)
Click-Jacking Attacks

Wanna see more Kittens?

Yes!

https://kittenpics.org/
XFO vs. CSP frame-ancestors

X-Frame-Options:
- Deprecated since 2012
- Inconsistently implemented
- Only Partially supported
- Double Framing attacks
- Only one whitelisted entry

CSP frame-ancestors:
- Well defined standard
- Supported if CSP LvL2 is
- Secure against Double Framing
- Whitelist uses CSP semantics
Research Questions

• Can we formally describe the inconsistency between the XFO header and CSP frame-ancestors?

• How inconsistent is framing control implemented in different browsers / deployed on real-world Web sites?

• Can we automatically fix inconsistencies?
## Browser Semantics for Framing Control

<table>
<thead>
<tr>
<th>Browser</th>
<th>CSP</th>
<th>ALLOW-FROM</th>
<th>Multiple Headers</th>
<th>Header Parsing</th>
<th>Double Framing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chrome</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Chrome (Android)</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Edge</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Firefox</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Internet Explorer</td>
<td>✗</td>
<td>✔</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Opera Mini</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>✔</td>
</tr>
<tr>
<td>Safari</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Safari (iOS)</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Samsung Internet</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>UC Browser</td>
<td>✔</td>
<td>✗</td>
<td>✔</td>
<td>✔</td>
<td>✗</td>
</tr>
</tbody>
</table>
Formal Framework

- Based on CoreCSP\textsuperscript{[1]} such that directive values can be ordered by the following relation:

\[ \nu_1 \subseteq \nu_2 \text{ iff the set of origins represented by } \nu_1 \text{ is contained in the set of origins represented by } \nu_2. \]

\textsuperscript{[1]} USENIX Security 2017: CCSP: Controlled Relaxation of Content Security Policies by Runtime Policy Composition
Stefano Calzavara, Alvise Rabitti, and Michele Bugliesi, Università Ca’ Foscari Venezia
Consistent Policy

• Let $w$ be a Web Page and $B$ the set of browsers.

• Consistent Policy:

  The policy of the Web page $w$ is consistent for the set of browsers $B$ iff $\forall b_1, b_2 \in B$, we have $\llbracket w \rrbracket_{b_1} \subseteq \llbracket w \rrbracket_{b_2}$ and $\llbracket w \rrbracket_{b_2} \subseteq \llbracket w \rrbracket_{b_1}$.
Policy Orientation

- $B_l = \text{Part}(B)$ only includes legacy browsers.
- $B_m = \text{Part}(B)$ only modern browsers.
- The policy of $w$ is consistent for both $B_l$ and $B_m$.
- For all $b_1 \in B_l$ and $b_2 \in B_m$
  - Policy is compatibility-oriented iff $[w]_{b_2} \subseteq [w]_{b_1}$.
  - Policy is security-oriented iff $[w]_{b_1} \subseteq [w]_{b_2}$.
Example: Compatibility-Orientation

• Web site example.com deploys:
  XFO ALLOW-FROM advertisements.com
  
  – Edge supports ALLOW-FROM
  – Chrome lacks support for this mode

➢ Not compatibility-oriented, because e.g. Chrome vs. Edge
Example: Security-Orientation

• Web site example.com deploys:
  frame-ancestors *.example.com + XFO SAMEORIGIN

  – Inconsistent because legacy browsers can not be framed by e.g. mail.example.com

  ➢ legacy browsers are more protected against framing based attacks, than modern clients => the policy is security-oriented.
Inconsistency in the Wild

Data Collection

Crawled the Tranco Top 10k Web Sites and collected max. 500 URLs/Site.
Collected all XFO and CSP headers from those URLs with different Browsers.

FrameCheck

Classification of headers based on the formal definitions\(^\text{[1]}\):
- security-oriented
- compatibility-oriented
- inconsistent policies

\[^{[1]}\text{https://github.com/cispa/framing-control-analytics}\]
FrameCheck Results

- ~370k/1M crawled URLs across 5,835 sites use framing control
  - In total, 17,613 policies

- 1,800 polices across 1,779 origins are inconsistent
  - Only XFO: 290/15,415 (1.9%)
  - Only CSP: 705/714 (98.7%)
  - XFO and CSP: 805/1,484 (54%)
Inconsistency in the Wild

Data Collection
Crawled the Tranco Top 10k Web Sites and collected max. 500 URLs/Site. Collected all XFO and CSP headers from those URLs with different Browsers.

FrameCheck
Classification of headers based on the formal definitions[1]:
• security-oriented
• compatibility-oriented
• inconsistent policies

Countermeasures

[1] https://github.com/cispa/framing-control-analytics
Contermeasures

- To sufficiently defend against framing attacks:
  - Use both XFO & CSP to secure modern & legacy browsers.
  - Return only one XFO header for each request.
  - Do not use comma-separated headers.
Retrofitting Security

Conclusion

Browser Semantics for Framing Control

<table>
<thead>
<tr>
<th>Browser</th>
<th>CSP</th>
<th>ALLOW-FROM</th>
<th>Multiple Headers</th>
<th>Header Parsing</th>
<th>Double Framing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chrome</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Chrome (Android)</td>
<td>✔️</td>
<td>✔️</td>
<td>X</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Edge</td>
<td>✔️</td>
<td>✔️</td>
<td>X</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Firefox</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Internet Explorer</td>
<td>✔️</td>
<td>✔️</td>
<td>X</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Opera Mini</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Safari</td>
<td>✔️</td>
<td>X</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Safari (OSR)</td>
<td>X</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Samsung Internet</td>
<td>✔️</td>
<td>X</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>UC Browser</td>
<td>✔️</td>
<td>X</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

FrameCheck Results

- ~370k/1M crawled URLs across 5,835 sites use framing control
  - In total, 17,613 policies
- 1,800 policies across 1,779 origins are inconsistent
  - Only XFO: 290/15,415 (1.9%)
  - Only CSP: 705/714 (98.7%)
  - XFO and CSP: 805/1,484 (54%)

Consistent Policy

- Definition 1 (Consistent Policy):
  The policy of the Web page $w$ is consistent for the set of browsers $B$ iff $\forall b_1, b_2 \in B$, we have $\|w\|_{b_1} = \|w\|_{b_2}$ and $\|w\|_{b_2} \subseteq \|w\|_{b_1}$.

Retrofitting Security