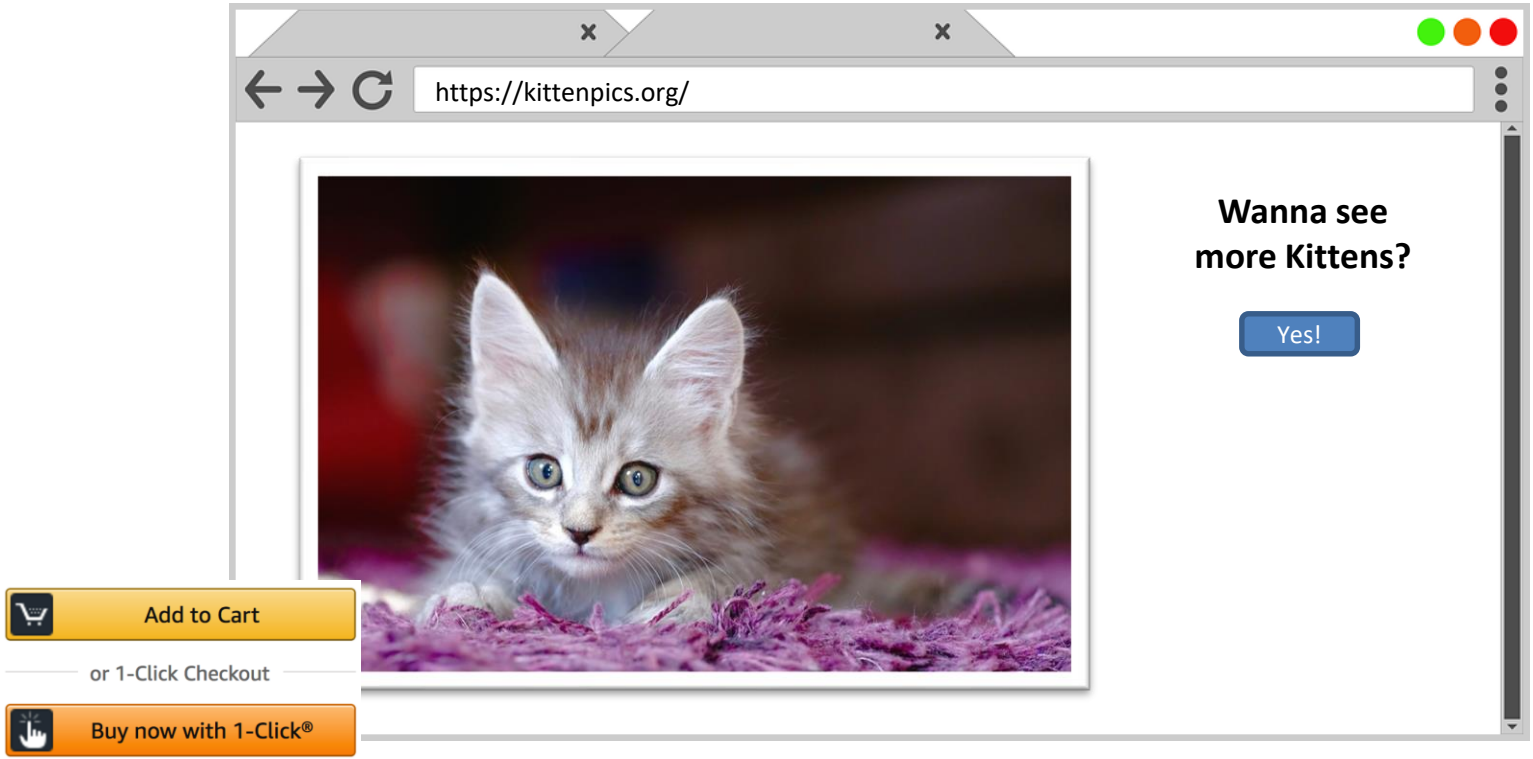
# A Tale of Two Headers: A Formal Analysis of Inconsistent Click-Jacking Protection on the Web

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



# XFO vs. CSP frame-ancestors

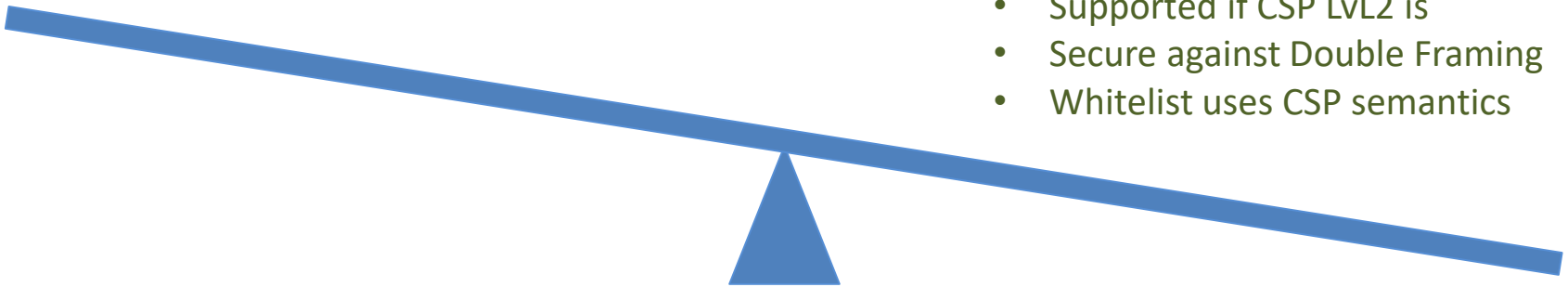
CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

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

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

- 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

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

| Browser | CSP | ALLOW-FROM | Multiple Headers | Header Parsing | Double Framing |
|---|---|---|---|---|---|
| Chrome | ✓ | ✗ | ✓ | ✓ | ✓ |
| Chrome (Android) | ✓ | ✗ | ✓ | ✓ | ✓ |
| Edge | ✓ | ✓ | ✗ | ✗ | ✗ |
| Firefox | ✓ | ✓ | ✓ | ✓ | ✓ |
| Internet Explorer | ✗ | ✓ | ✗ | ✗ | ✗ |
| Opera Mini | ✗ | ✗ | ✗ | ✗ | ✓ |
| Safari | ✓ | ✗ | ✓ | ✓ | ✓ |
| Safari (iOS) | ✓ | ✗ | ✓ | ✓ | ✓ |
| Samsung Internet | ✓ | ✗ | ✓ | ✓ | ✓ |
| UC Browser | ✓ | ✗ | ✓ | ✓ | ✗ |

# Formal Framework

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

  $v_1 \sqsubseteq v_2$ iff the set of origins represented by $v_1$ is contained in the set of origins represented by $v_2$.

[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 $[\![w]\!]_{b_1} \sqsubseteq [\![w]\!]_{b_2}$ and $[\![w]\!]_{b_2} \sqsubseteq [\![w]\!]_{b_1}$.

# Policy Orientation

- $B_l = \mathrm{Part(B)}$ only includes legacy browsers.
- $B_m = \mathrm{Part(B)}$ 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 $[\![w]\!]_{b_2} \sqsubseteq [\![w]\!]_{b_1}$.
  - Policy is security-oriented iff $[\![w]\!]_{b_1} \sqsubseteq [\![w]\!]_{b_2}$.

# Example: Compatibility-Orientation

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

- 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

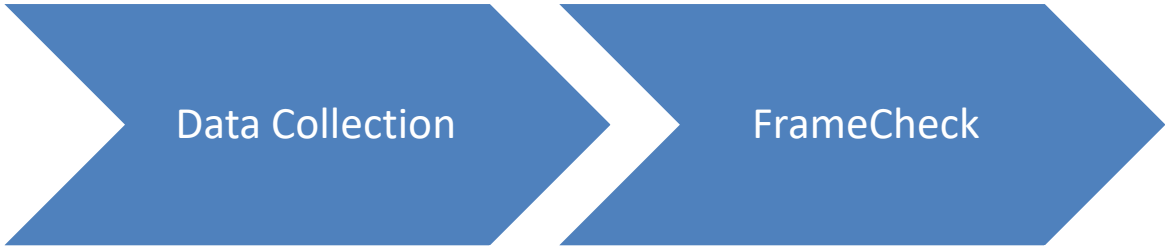
- Web site example.com deploys:

  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




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.

Classification of headers based on the formal definitions[1]:

- security-oriented
- compatibility-oriented
- inconsistent policies

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

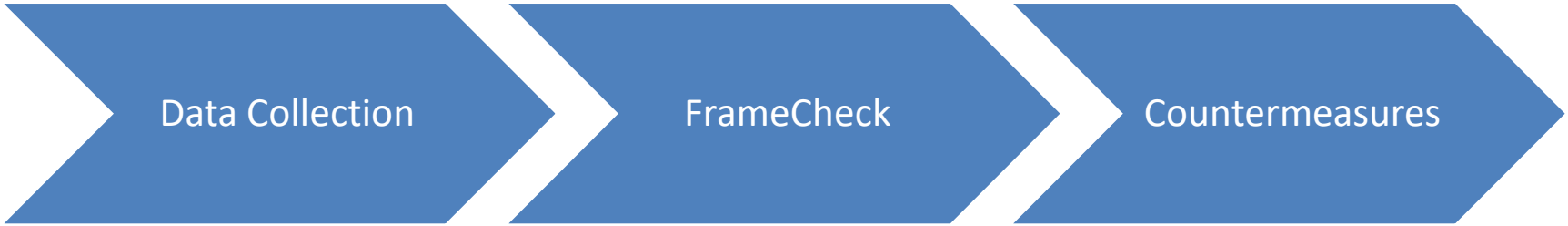
# FrameCheck Results

- ~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

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY




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.

Classification of headers based on the formal definitions[1]:

- security-oriented
- compatibility-oriented
- inconsistent policies

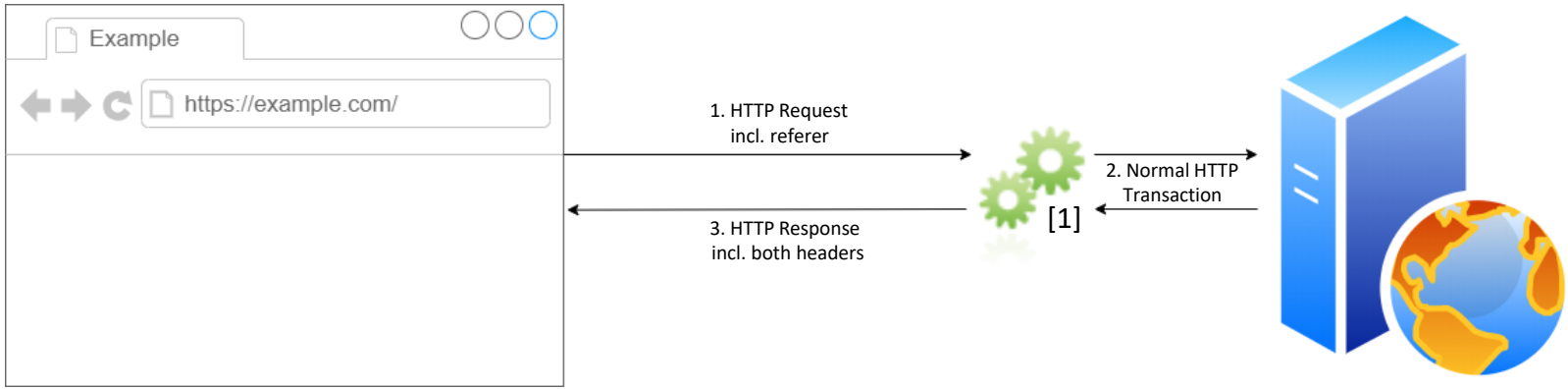
Recommendations and Countermeasures for Operators, Web Developers, and Browser vendors. Retrofitting Security via a server-side proxy.

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

# Contermeasures

- To sufficently 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-seperated headers.

# Retrofitting Security




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

# Conclusion

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

## Browser Semantics for Framing Control

| Browser | CSP | ALLOW-FROM | Multiple Headers | Header Parsing | Double Framing |
|---|---|---|---|---|---|
| Chrome | ✓ | ✗ | ✓ | ✓ | ✓ |
| Chrome (Android) | ✓ | ✗ | ✓ | ✓ | ✓ |
| Edge | ✓ | ✓ | ✗ | ✗ | ✗ |
| Firefox | ✓ | ✓ | ✓ | ✓ | ✓ |
| Internet Explorer | ✗ | ✓ | ✗ | ✗ | ✗ |
| Opera Mini | ✗ | ✗ | ✗ | ✗ | ✓ |
| Safari | ✓ | ✗ | ✓ | ✓ | ✓ |
| Safari (iOS) | ✓ | ✗ | ✓ | ✓ | ✓ |
| Samsung Internet | ✓ | ✗ | ✓ | ✓ | ✓ |
| UC Browser | ✓ | ✗ | ✓ | ✓ | ✗ |

USENIX Security 2020 – Sebastian Roth – A Tale of Two Headers 5

## 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%)

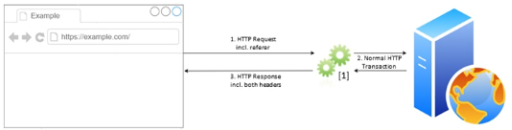
USENIX Security 2020 – Sebastian Roth – A Tale of Two Headers 13

## 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 $[\![w]\!]_{b_1} \sqsubseteq [\![w]\!]_{b_2}$ and $[\![w]\!]_{b_2} \sqsubseteq [\![w]\!]_{b_1}$.

USENIX Security 2020 – Sebastian Roth – A Tale of Two Headers 6

## Retrofitting Security




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

USENIX Security 2020 – Sebastian Roth – A Tale of Two Headers 15