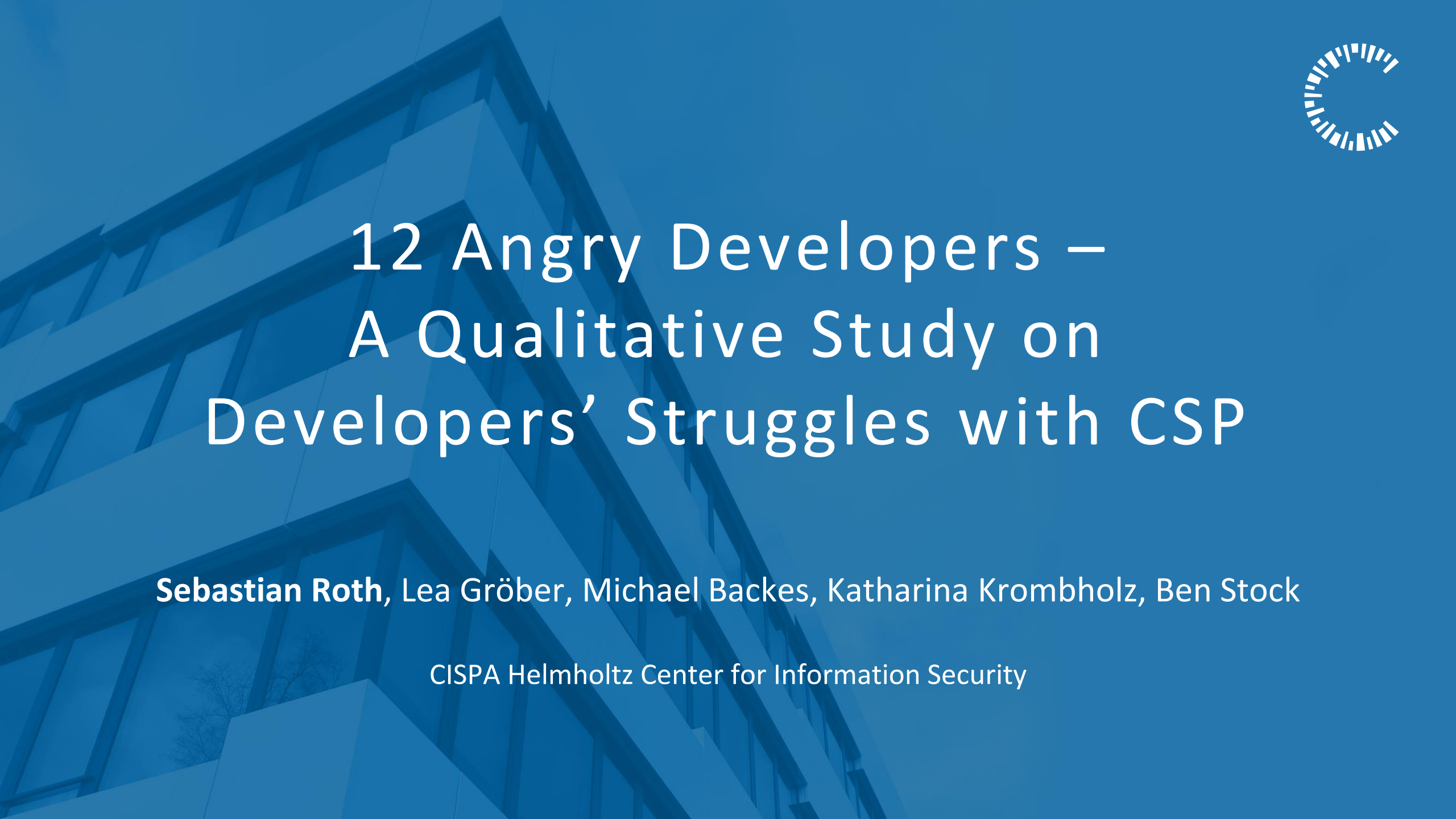
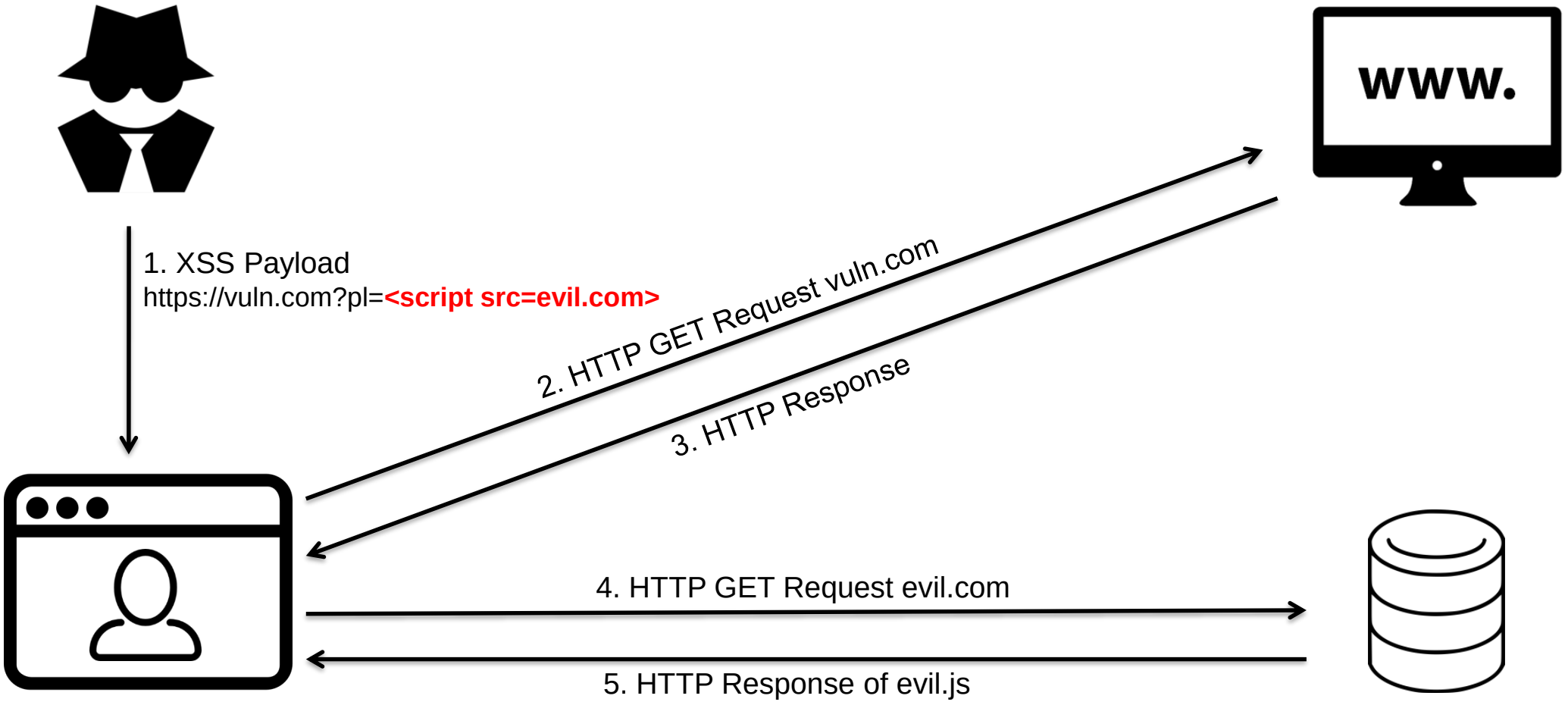


12 Angry Developers – A Qualitative Study on Developers' Struggles with CSP

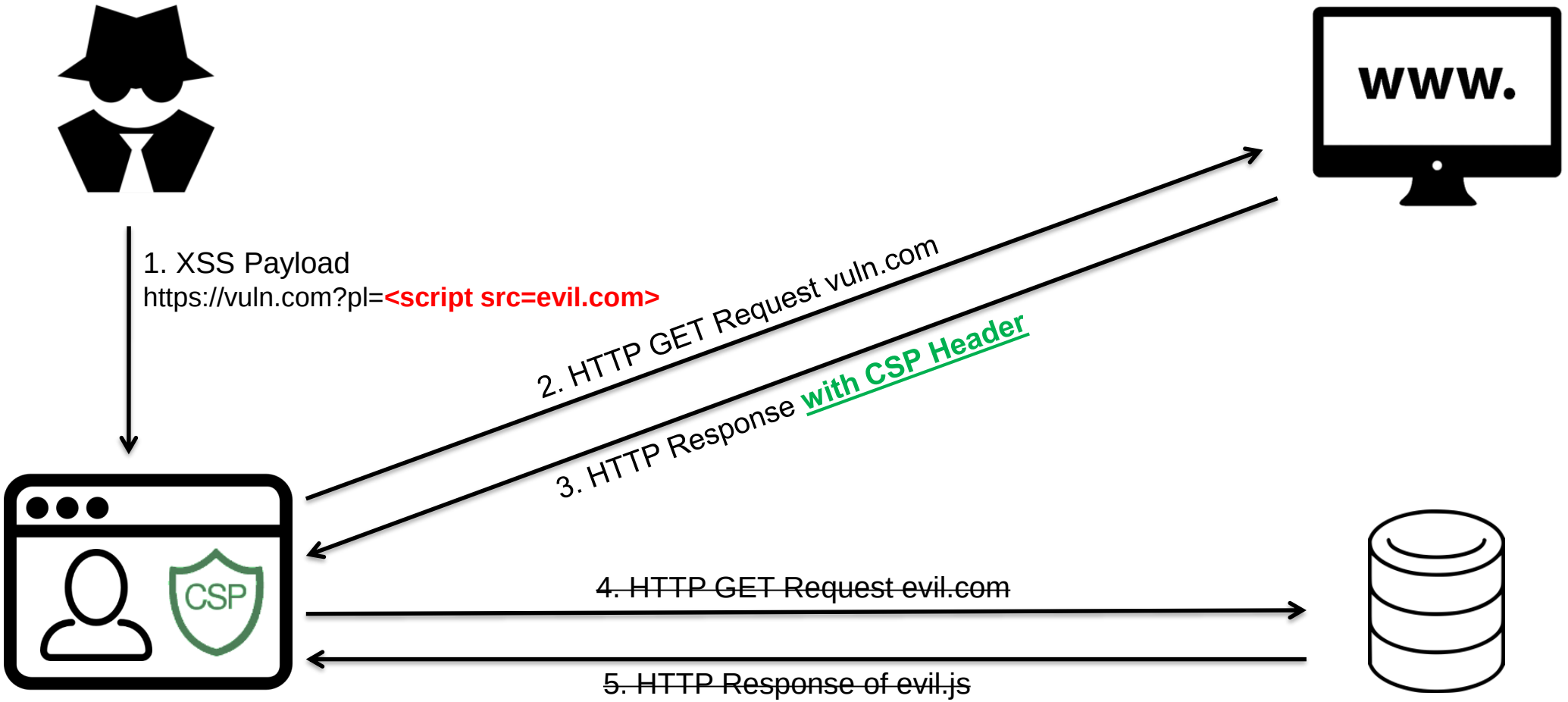
Sebastian Roth, Lea Gröber, Michael Backes, Katharina Krombholz, Ben Stock

CISPA Helmholtz Center for Information Security

Cross-Site Scripting (XSS)



Content Security Policy (CSP)



Content Security Policy (CSP)

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

CSP (2012)

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

Requires the Content Security Policy:

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

CSP (2014)

```
<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>
```

Requires the Content Security Policy:

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

CSP (2016)

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

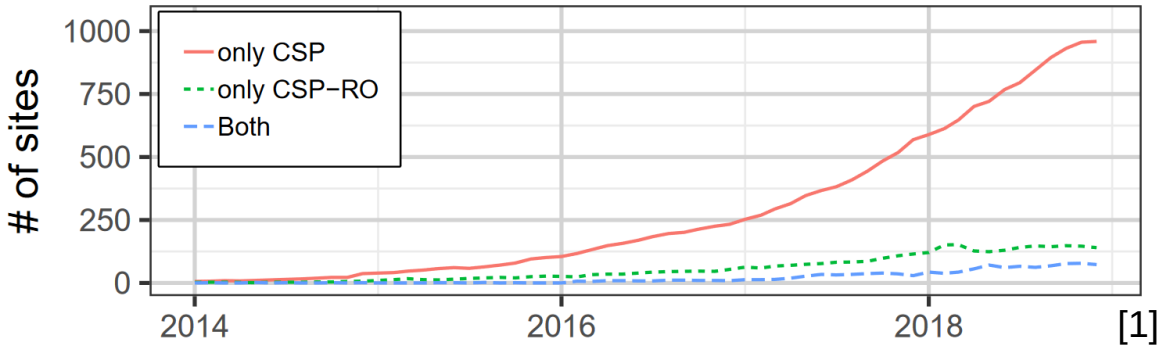
Requires the Content Security Policy:

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

Previous Work

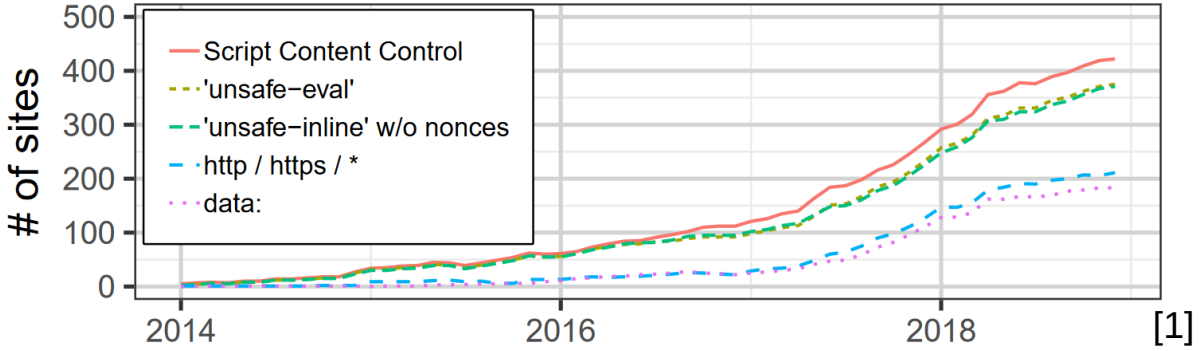
[1] **Complex Security Policy? – A Longitudinal Analysis of Deployed Content Security Policies**
Sebastian Roth, Timothy Barron, Stefano Calzavara, Nick Nikiforakis, and Ben Stock
Network and Distributed System Security Symposium (NDSS '20)

Not many sites are using CSP…



[1]

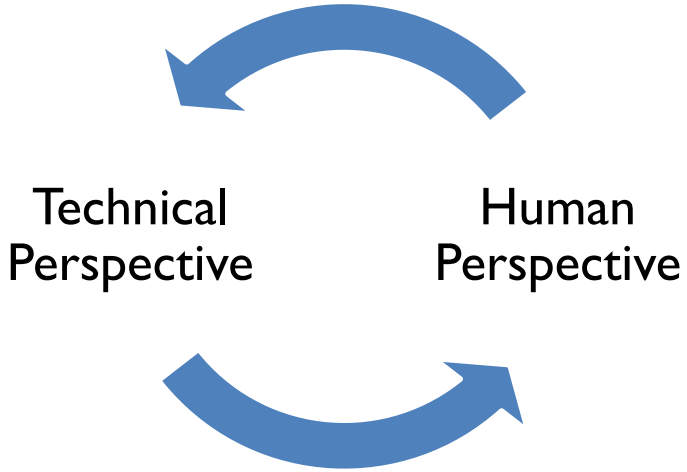
… and those that do, misconfigure it.



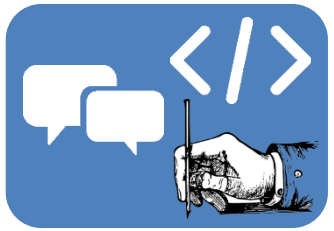
[1]

Research Questions

1. What are the root causes of insecure practices when deploying a CSP?
2. What strategies do developers adopt when creating a CSP?
3. How well do developers understand the associated threat models of CSP?
4. What are the perceptions and motivations of developers in terms of deploying a CSP?



Methodology



Semi-Structured Interview
Incl. Drawing & Coding Task

Interview Transscription

Open Coding Process

Find Motivations, Strategies,
and Roadblocks of CSP

Drawing Task

- Setup:
  - Participants were asked to draw and explain their favorite XSS attack.
  - … and later asked where CSP would block the execution of malicious JS
- Server-side XSS was exclusivly drawn
  - Only one mentioned client-side XSS
- Two participants actively mentioned XSS as server-side problem and also reported that CSP is enforced by the server.

Motivations

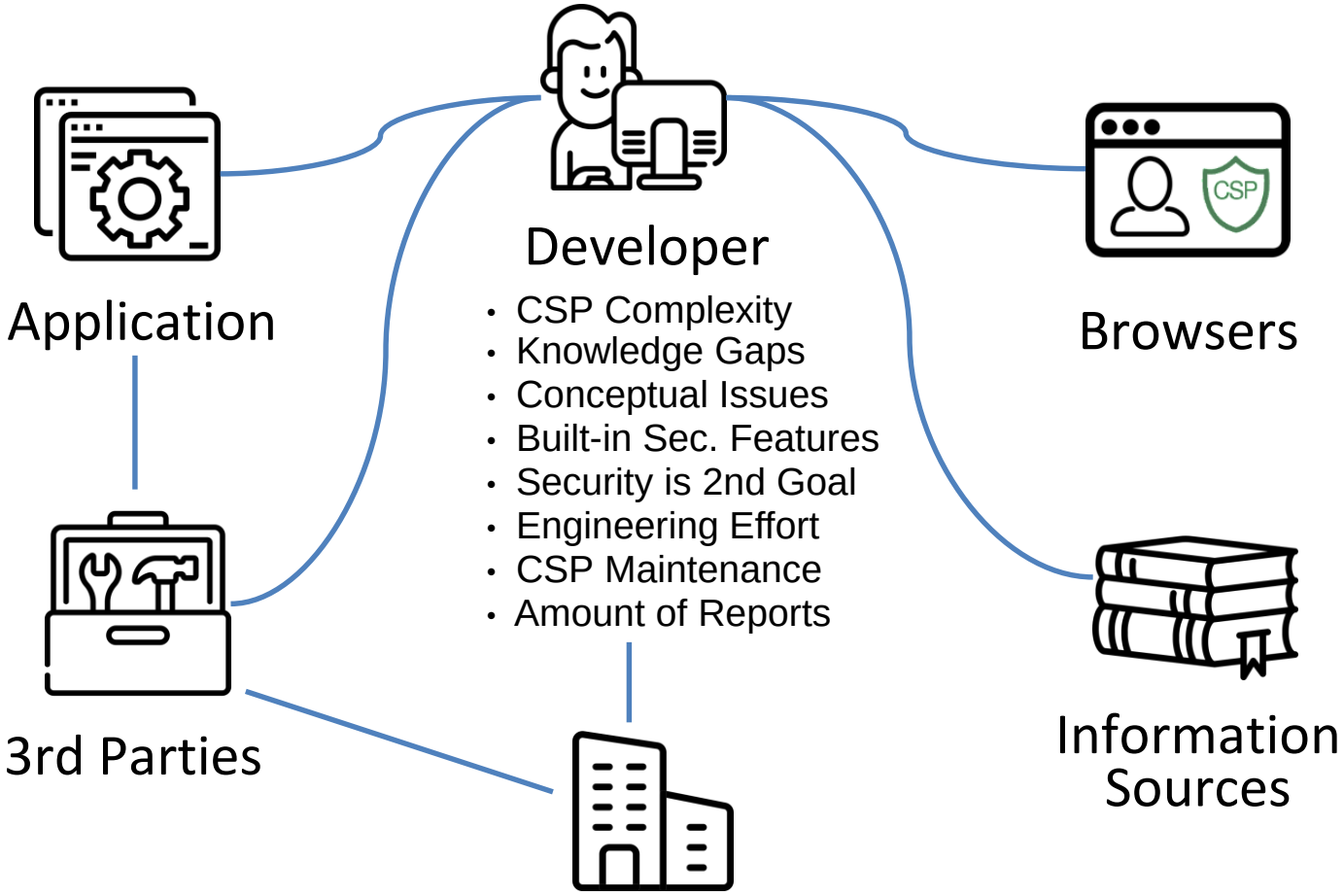
Attack Mitigation

- XSS Mitigation
- Resource Control
- Framing Control
- TLS Enforcement
- Data Connection Control

External Motivation

- Pentest / Consulting
- Additional Security Layer
- Reputation
- Role Model
- Security Training
- Build Pipeline Warning
- Financial Implications

Roadblocks



- Inline JavaScript
- Inline Events
- Legacy Code

- CSP Complexity
- Knowledge Gaps
- Conceptual Issues
- Built-in Sec. Features
- Security is 2nd Goal
- Engineering Effort
- CSP Maintenance
- Amount of Reports

- Browser Inconsistency
- Browser Console Messages
- False Positive Reports
- Insufficient Reports
- Browser Extensions

- Framework Support
- 3rd-party Services
- 3rd-party Libraries

- Lack of bigger picture
- Misleading claims

- Different Development Teams
- Financial Consequences

Strategies

- Restrictive Report-Only Policy
- Restrictive Enforcement Policy
- Lax CSP
- Start with generated CSP
- Use tools for Initial Deployment

- Report-Only to debug live App
- In-the-field Testing
- Iterative Deployment
- CSP Integral Part of Development
- One general CSP
- Separate CSP for Subpages
- Functionality > Security
- Use Meta Tag CSP
- Used Tools during Deployment

Inline Code	• …
Inline Events	• …
3rd- Party Code	• …

Results: Problem Solving Strategies

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY

Inline Code

- Externalize inline code
- Allow inline code using the content hash
- Tools to help with inline code
- Use unsafe-inline as fallback
- Nonce inline scripts

Inline Events

- Externalize events
- Changing functionality
- Use the script-src-attr directive
- Allow events using their content hash
- Use unsafe-inline

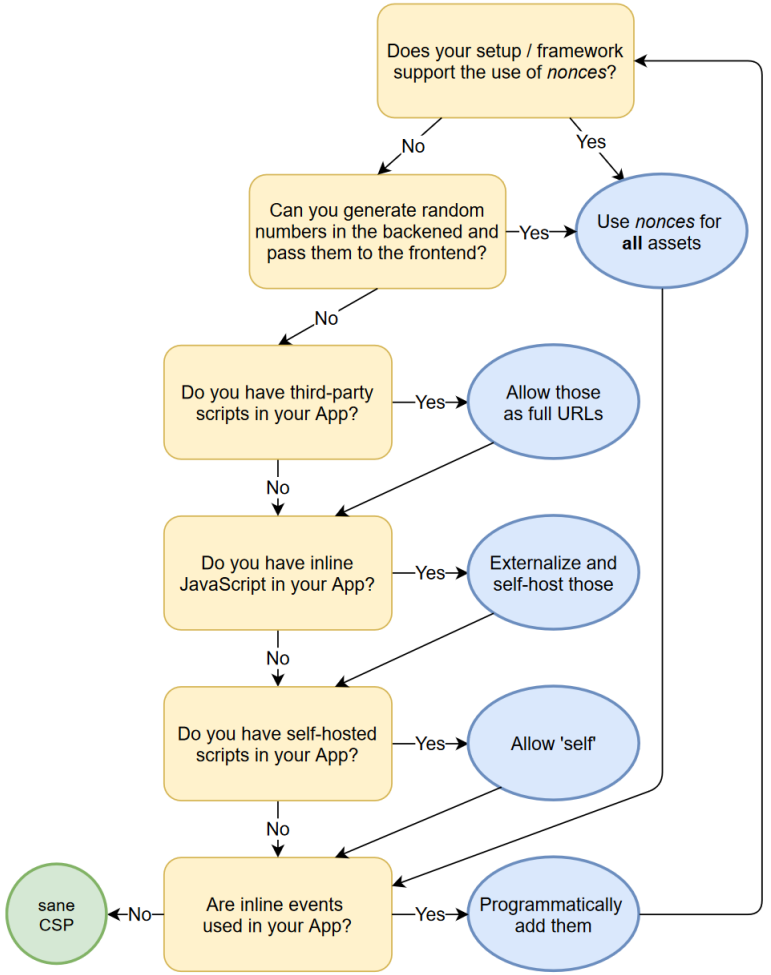
3rd- Party Code

- Self-host 3rd-Party code
- Remove dependencies
- Nonce 3rd-Party code

How can we improve the situation?

- The browser vendors:
  - Mitigate 3rd-Party impact:
    - Restricting API access (SecurerContexts[1])
    - Seperate first and 3rd-Party Code
  - Improve Documentation, Console messges, and reporting APIs

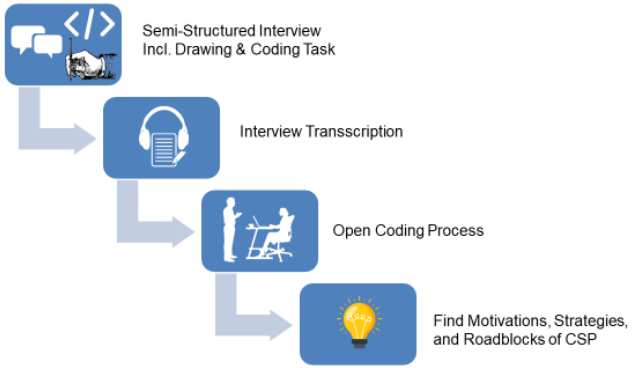
- We, as a community:
  - Better Information Sources
  - Better Tools



[1] Mike West – SecurerContexts
https://github.com/mikewest/securer-contexts

Conclusion

Methodology



Motivations

Attack Mitigation

- XSS Mitigation
- Resource Control
- Framing Control
- TLS Enforcement
- Data Connection Control

External Motivation

- Pentest / Consulting
- Additional Security Layer
- Reputation
- Role Model
- Security Training
- Build Pipeline Warning
- Financial Implications

Roadblocks

- Inline JavaScript
- Inline Events
- Legacy Code

Application

- Framework Support
- 3rd-party Services
- 3rd-party Libraries

3rd Parties

Developer

- CSP Complexity
- Knowledge Gaps
- Conceptual Issues
- Built-in Sec. Features
- Security is 2nd Goal
- Engineering Effort
- CSP Maintenance
- Amount of Reports

Company

- Different Development Teams
- Financial Consequences

Browsers

- Browser Inconsistency
- Browser Console Messages
- False Positive Reports
- Insufficient Reports
- Browser Extensions

Information Sources

- Lack of bigger picture
- Misleading claims

Strategies

Initial Deployment

- Restrictive Report-Only Policy
- Restrictive Enforcement Policy
- Lax CSP
- Start with generated CSP
- Use tools for Initial Deployment

Problem Solving Strategies

Inline Code	• …
Inline Events	• …
3rd- Party Code	• …

Deployment Principles

- Report-Only to debug live App
- In-the-field Testing
- Iterative Deployment
- CSP Integral Part of Development
- One general CSP
- Separate CSP for Subpages
- Functionality > Security
- Use Meta Tag CSP
- Used Tools during Deployment