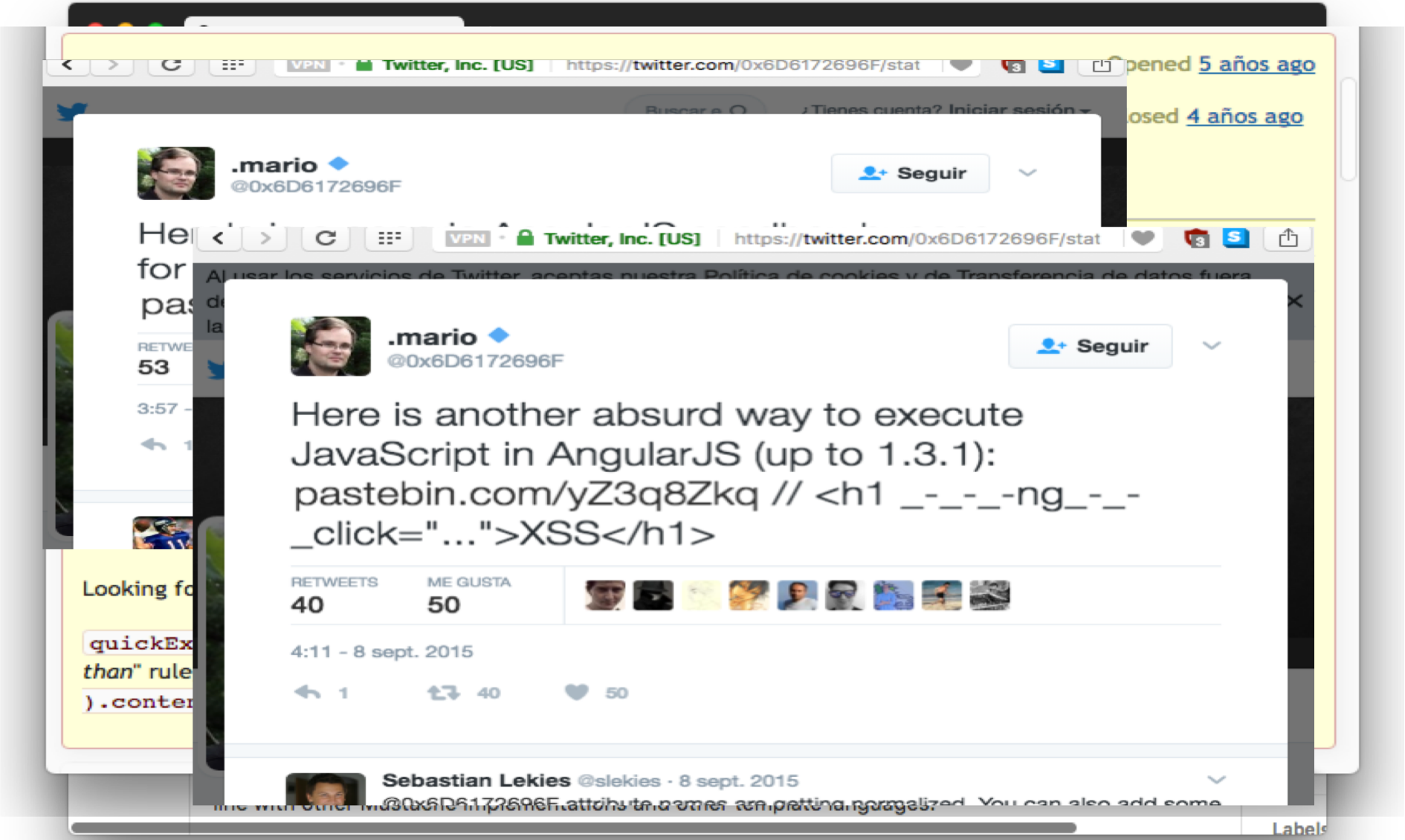
Toby Lauinger, A. Chaabane, S. Arshad,
W. Robertson, C. Wilson, E. Kirda

Thou Shalt Not Depend on Me: Analysing the Use of Outdated JavaScript Libraries on the Web

NDSS 2017

Motivation

- 87% of Alexa websites include third-party JavaScript libraries


Twitter, Inc. [US]
https://twitter.com/0x6D6172696F/stat
.mario
@0x6D6172696F
Seguir
Here is another absurd way to execute JavaScript in AngularJS (up to 1.3.1): pastebin.com/yZ3q8Zkq // <h1 _-_-_-ng_-_-_click="...">XSS</h1>
RETWEETS
40
ME GUSTA
50
4:11 - 8 sept. 2015
1
40
50
Sebastian Lekies @slekies · 8 sept. 2015

Motivation

- 87% of Alexa websites include third-party JavaScript libraries
- How well are these dependencies maintained?
  - How many known vulnerable/outdated inclusions?
  - Who is to blame for inclusions of known vulnerable/outdated versions?
- Exploitability is out of scope

Contributions

- Comprehensive study showing use of vulnerable or outdated JavaScript libraries (client-side)
- Model to represent element creation relationships in complex websites
- Look at origins and common scenarios of (vulnerable) library inclusions

Background

- No mandatory standard for JavaScript libraries
- Semantic versioning
  `<major>.<minor>.<patch>` (e.g., `1.2.3`)
- Vulnerabilities: typically XSS

Methodology Overview

- Collect metadata about libraries
  - Versions, release dates, code samples, vulnerabilities
  - 72 open-source libraries (11 with vulnerability data)
- Detect libraries used in websites
  - Static detection (hash)
  - Dynamic detection (environment fingerprinting)
- Find out how ("why") libraries are included

Library Detection

- Static detection (hash)

```
jquery-1.11.0.js      → 3b8042…
jquery-1.11.0.min.js → 8fc25e…
```

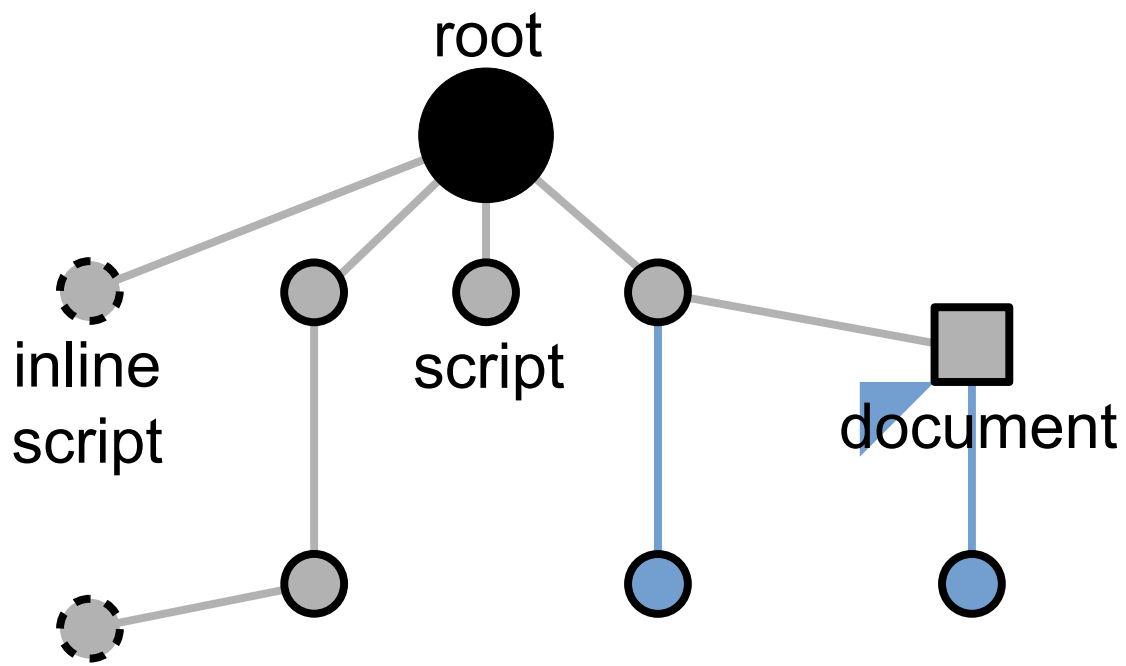
  – Only if source code sample exists
  – Fails for custom builds, minification settings etc.

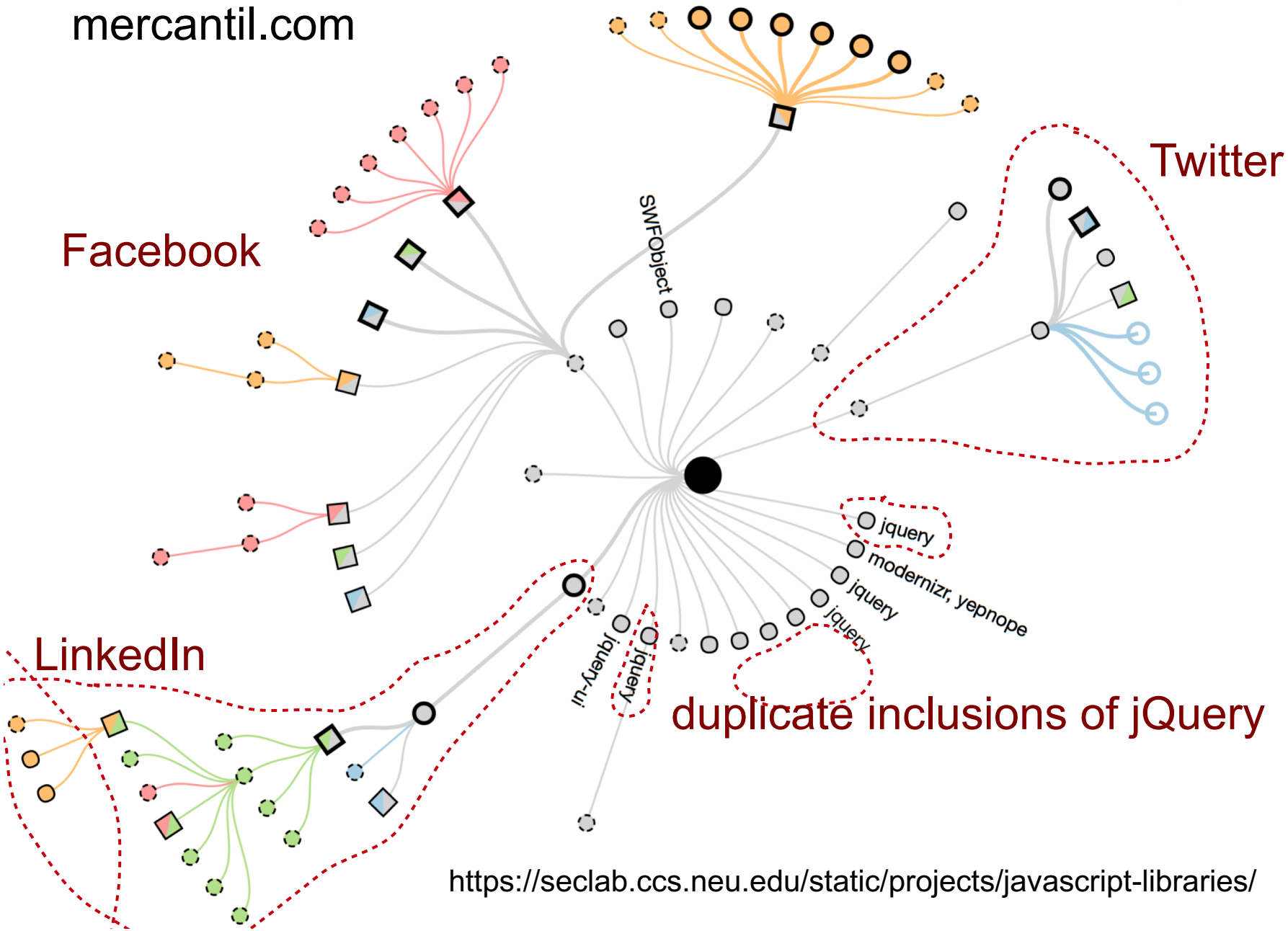
- Dynamic detection (environment signatures)

```
> $.fn.jquery;
"2.1.1"
```

Causality Trees

- Represent element creation relationships in dynamic websites (“A includes B”)
- Causality tree orthogonal to DOM tree





https://seclab.ccs.neu.edu/static/projects/javascript-libraries/

Methodology - Crawl

- Causality tree generation built on Chrome Debugging Protocol and modified browser
- Detect ads/trackers/widgets using modified AdBlock Plus extension
- Sites crawled (May 2016):
  - Alexa Top 75k
  - Random 75k sample of .com zone

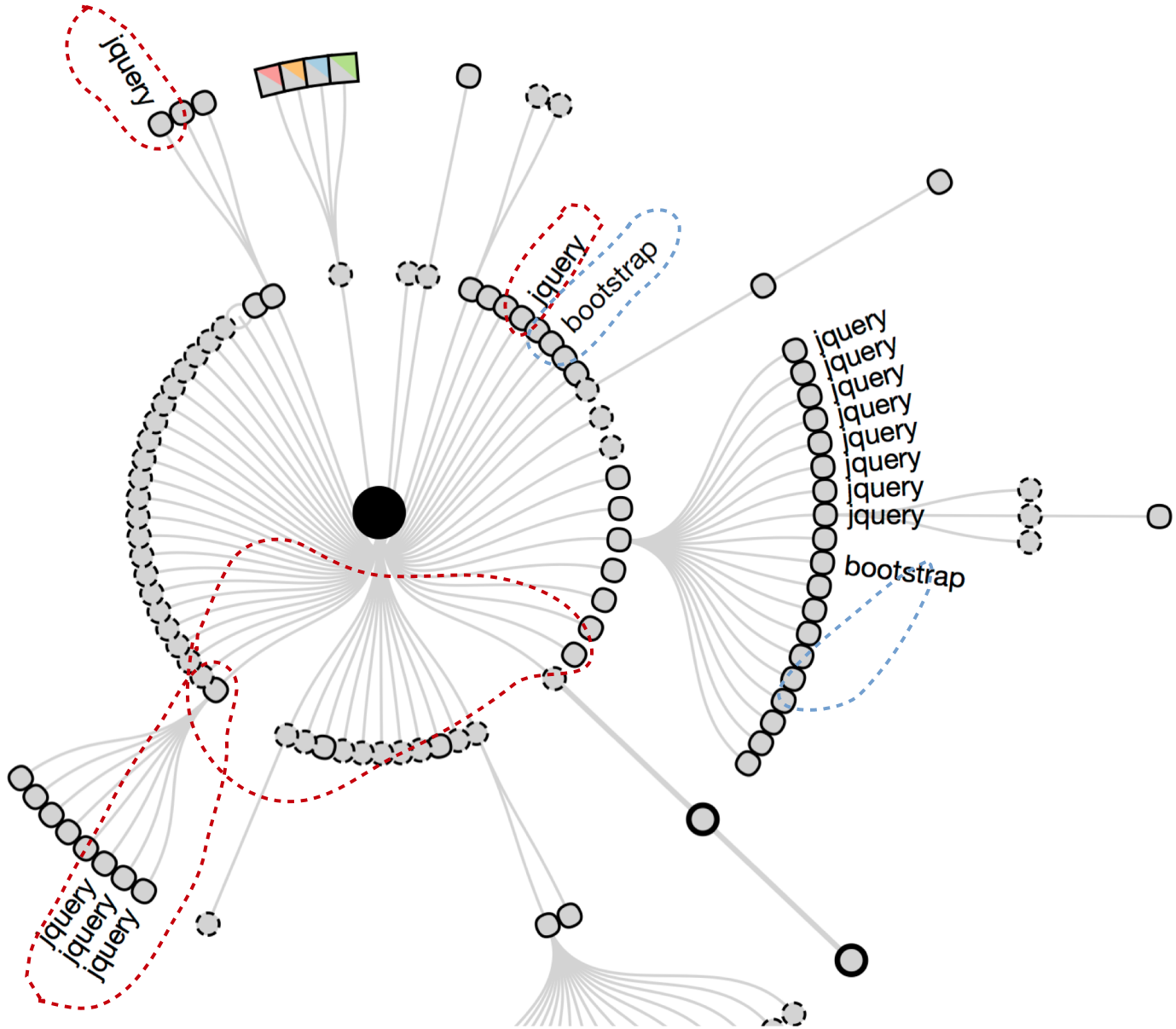
Analysis Results

- JavaScript libraries frequently used (jQuery on 84% of Alexa)
- Libraries sometimes indirectly included (e.g., 7% of Alexa include library via ads etc.)
- 38% of Alexa use at least one known vulnerable version
- 61% of Alexa use library that isn't at the latest patch-level version in the respective branch (i.e., outdated)

Duplicate Inclusions

- Typically, only one copy per library can be used in each document (“window”-global variable)

ms.gov


jquery
jquery
bootstrap
jquery
jquery
jquery
jquery
jquery
jquery
jquery
jquery
bootstrap
jquery
jquery
jquery

Duplicate Inclusions

- Typically, only one copy per library can be used in each document ("window"-global variable)
- Yet… we observed for jQuery in Alexa:
  - 2+ *different* versions in *same* document (11%)
  - 2+ *identical* versions in same document (4%)
- Due to templating, plug-ins, sometimes ads

Concluding Remarks

- JavaScript libraries included in many (unexpected) scenarios
- Many websites use vulnerable or outdated libraries: “maintenance issue”
- Possible reasons:
  - Scant information about vulnerabilities
  - Lack of backwards-compatible patches

security | Search Results | jQ

https://learn.jquery.com/?s=security

Plugins Contribute Events Support jQuery Foundation

jQuery
write less, do more.

Home About Contributing Style Guide security

Search Results for: security

Nothing Found

Apologies, but nothing matched your search criteria.

Northeastern

Secure Systems Lab

http://www.seclab.nu/