



A Clean Slate Approach to Web Security

Dan Boneh

Joint work with John Mitchell, Collin Jackson,
Andrew Bortz, Adam Barth

Take home message

- ◆ Current web technology is often designed for features rather than security.
- ◆ Our goal:
 - Study clean-slate approaches that provide features with increased security for end-users
 - ... and propose steps to get there.
- ◆ This talk: two examples
 - XSS and same-origin policy

crypto.stanford.edu/antiphishing








Stanford Anti-Phishing Projects

Stanford University

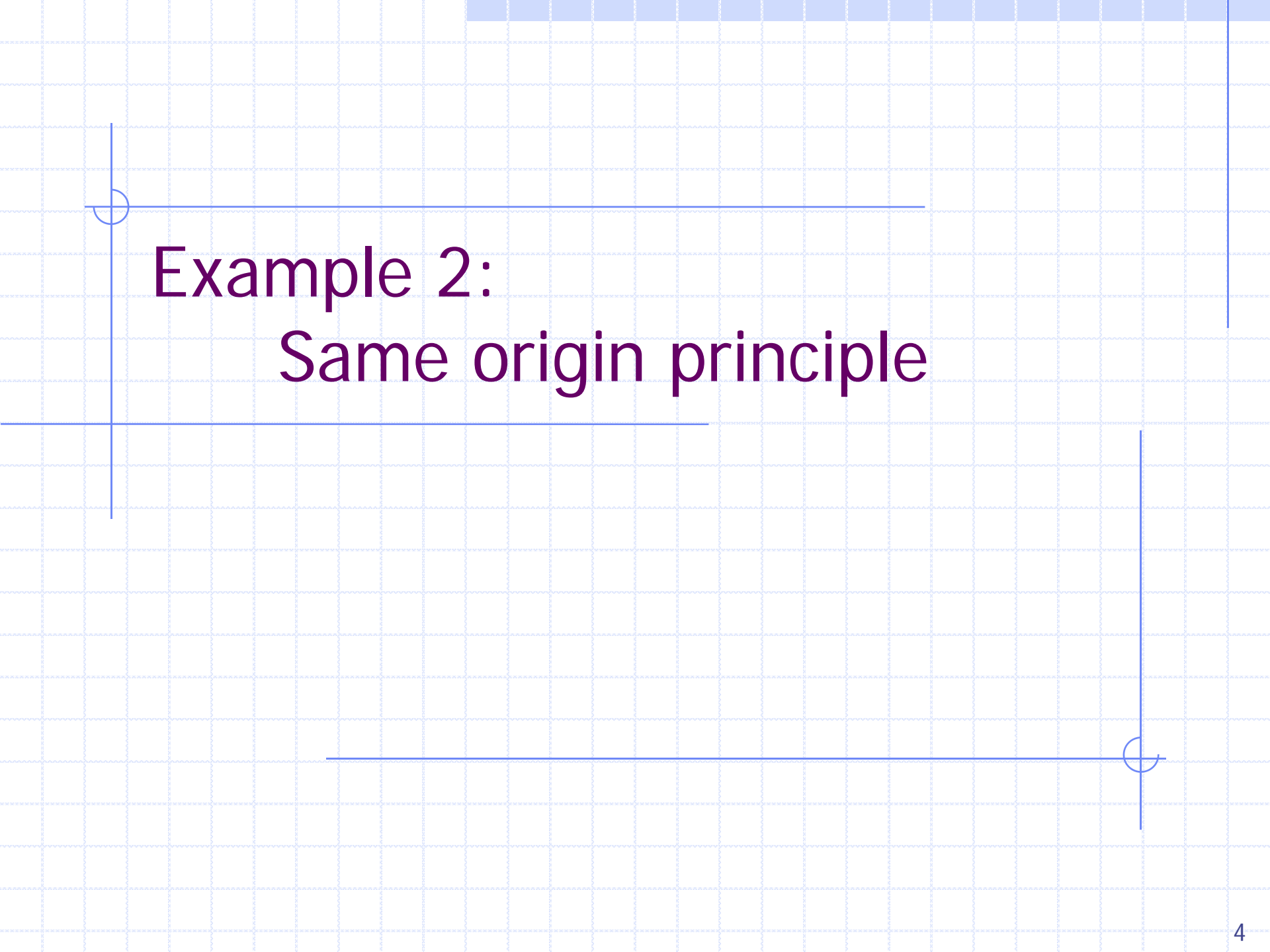
Projects

The following [Stanford Security Lab](#) projects defend against phishing and password theft attacks:

-  [SpoonGuard](#) detects when you visit a phishing page and warns you.
-  [PwdHash](#) generates phishing-resistant passwords.
-  [SafeCache](#) protects your browser cache from context-aware phishing attacks.
-  [SafeHistory](#) protects your visited links from context-aware phishing attacks.
-  [SpyBlock](#) protects user passwords from keyloggers.

Publications

- Neil Chou, Robert Ledesma, Yuka Teraguchi, [Dan Boneh](#), and [John C. Mitchell](#)
[Client-side defense against web-based identity theft](#) (PDF).
11th Annual Network and Distributed System Security Symposium (NDSS '04), San Diego, February, 2004.
- [Blake Ross](#), [Collin Jackson](#), Nicholas Miyake, [Dan Boneh](#), and [John C. Mitchell](#)



Example 2:
Same origin principle

Same origin principle

- ◆ The (abstract) principle:

Web site A should be unable to view any browser state set by Web site B ($A \neq B$)

- ◆ Problems with same origin principle:

- What is a “web site” :

- ◆ Many sites have sub-domains and partners
- ◆ DNS load balancing (e.g. Akamai) means data is loaded from many different locations.

- Poor implementations.

Poor implementation: examples

1. Web site A can determine if user ever visited web site B
 - Put link to “site B” on A’s site and query link color
 - Results in a context aware phishing attack
 - Our solution: SafeHistory an SafeCache

2. Web site A can determine if user is currently logged into web site B
 - ◆ Cross site timing attack [BBN’07]
 - ◆ Javascript error console [G’07]

3. Web site A can issue requests to site B on behalf of user:
 - ◆ Cross site request forgery (e.g. [RSJ’07])

Solutions

◆ Completed projects to date (all deployable today):

- SafeCache, SafeHistory,
- modtimepad,
- BrowserDNS

◆ Our current/future “clean-slate” work:

Clean definition of the same origin principle

- Enable site to define “site” without relying on DNS
- Enforce same origin principle by tagging browser state.

Take home message

- ◆ Current web technology is often designed for features rather than security.
- ◆ Our goal:
 - Study clean-slate approaches that provide features with increased security for end-users.
 - ... and propose steps to get there.



THE END