# A BREACH IN THE SAME-ORIGIN POLICY INDUCED BY MIRRORING EXTERNAL CONTENT

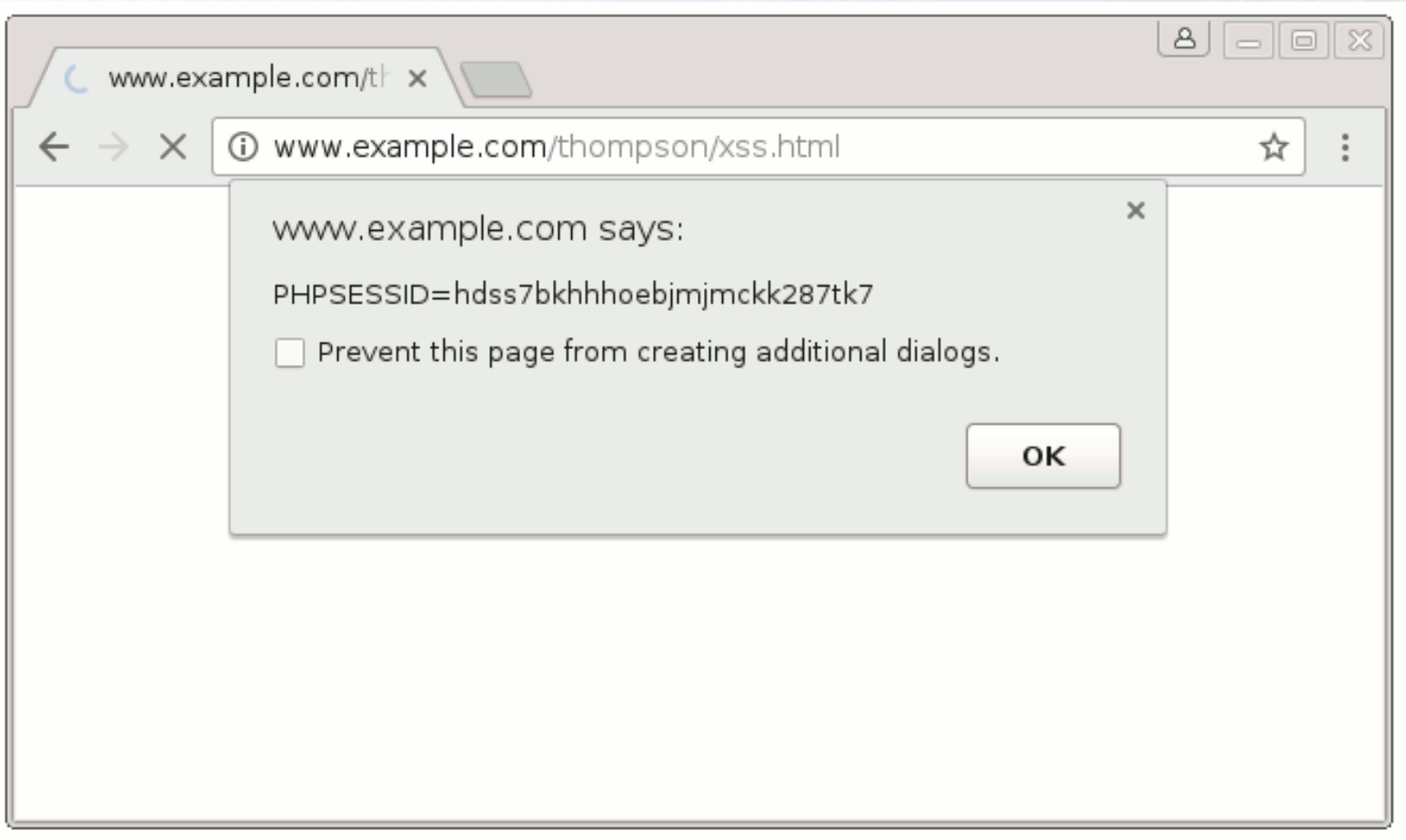
Jacob Thompson

October 16, 2016

# About ISE

- We are:
  - Computer Scientists
  - Academics
  - Ethical Hackers
- Our customers are:
  - Fortune 500 enterprises
  - Entertainment, software security, healthcare
- Our perspective is:
  - White box

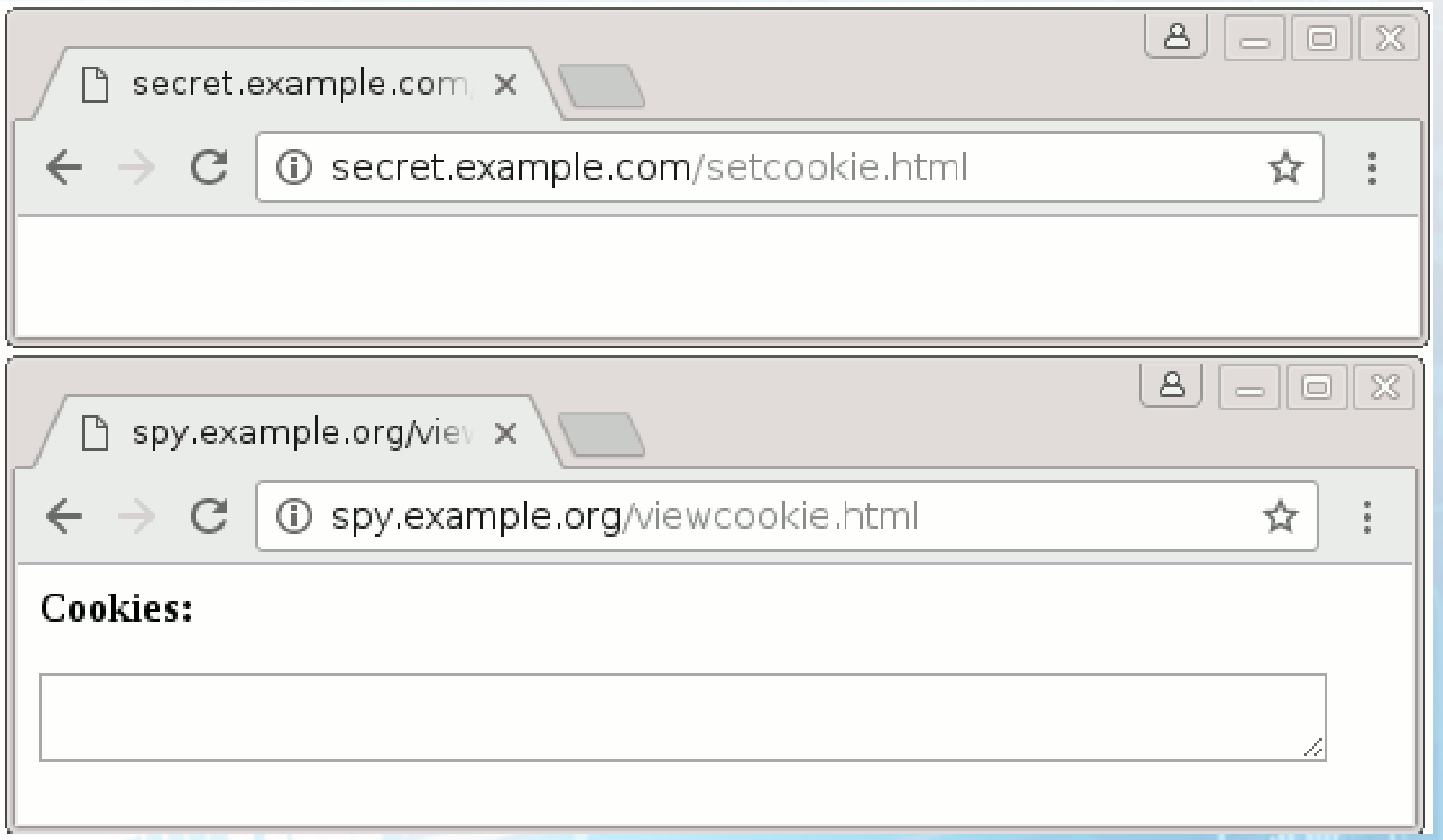
# Cross-Site Scripting



# Same-Origin Policy

- *spy.example.org* cannot read cookies for *secret.example.com*
- *spy.example.org* JavaScript code cannot interact with DOM for *secret.example.com*
- Cross-server interaction strictly controlled
- And so on…

# Example


secret.example.com
secret.example.com/setcookie.html
spy.example.org/viev
spy.example.org/viewcookie.html
Cookies:

# Seen This Before?

- http://www.example.net/fetch?url=http:%2f%2fsecret.example.com%2f
- http://www.example.net/fetch?url=http:%2f%2fspy.example.org%2f

# Mirror Site

- Serve a copy of HTML and JavaScript content from an origin other than where it originated

# “Pure” Mirrors

- Search engine page caches
- Translation services
- Internet Archive

# Proxy Sites

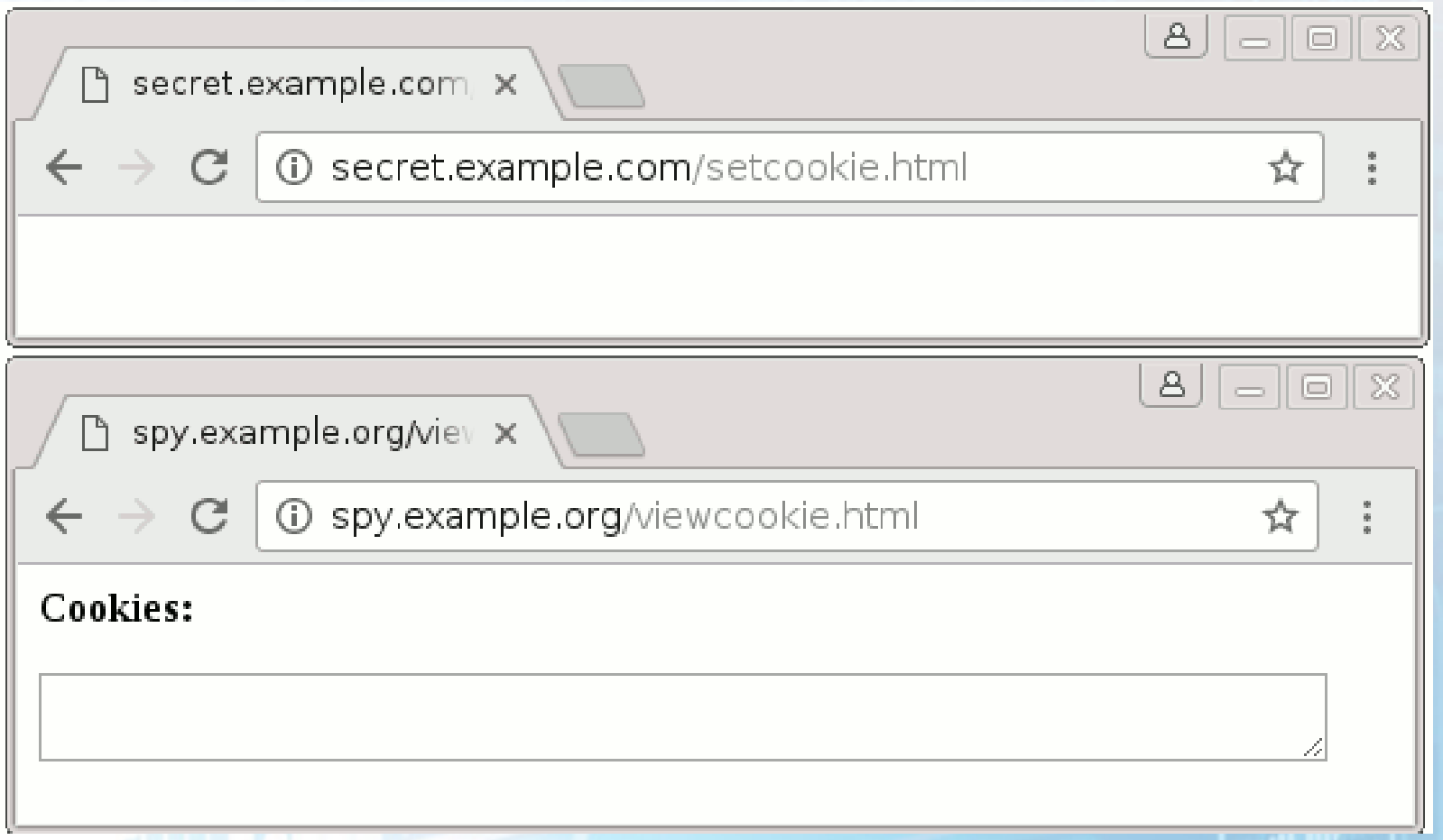
- Bad approximation of Tor?
- Hide.me
- Whoer.net
- CGIProxy software
- PHProxy software

# The Problem

- Mirroring security-relevant content causes a breakdown in the same-origin policy
- Privacy compromise
  - History leak
- Security compromise
  - Session leak
  - Full same-origin bypass
- Let's see examples of different behavior

# Normal vs. Mirrored Behavior

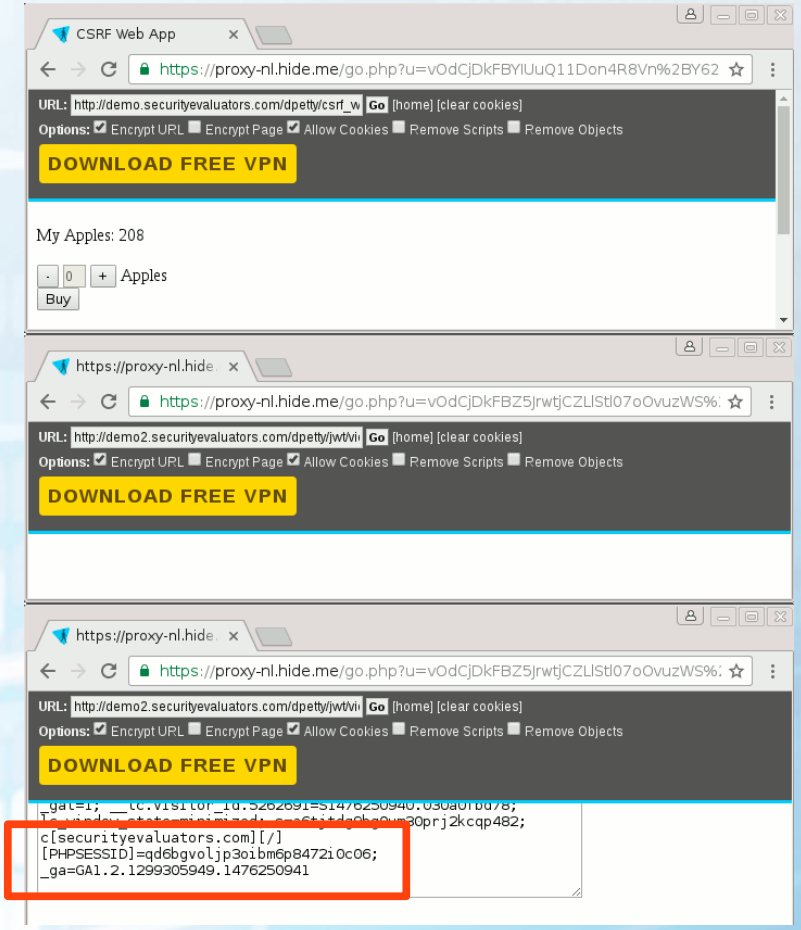
# History Leak – Normal (safe)



# History Leak – Translator (unsafe)

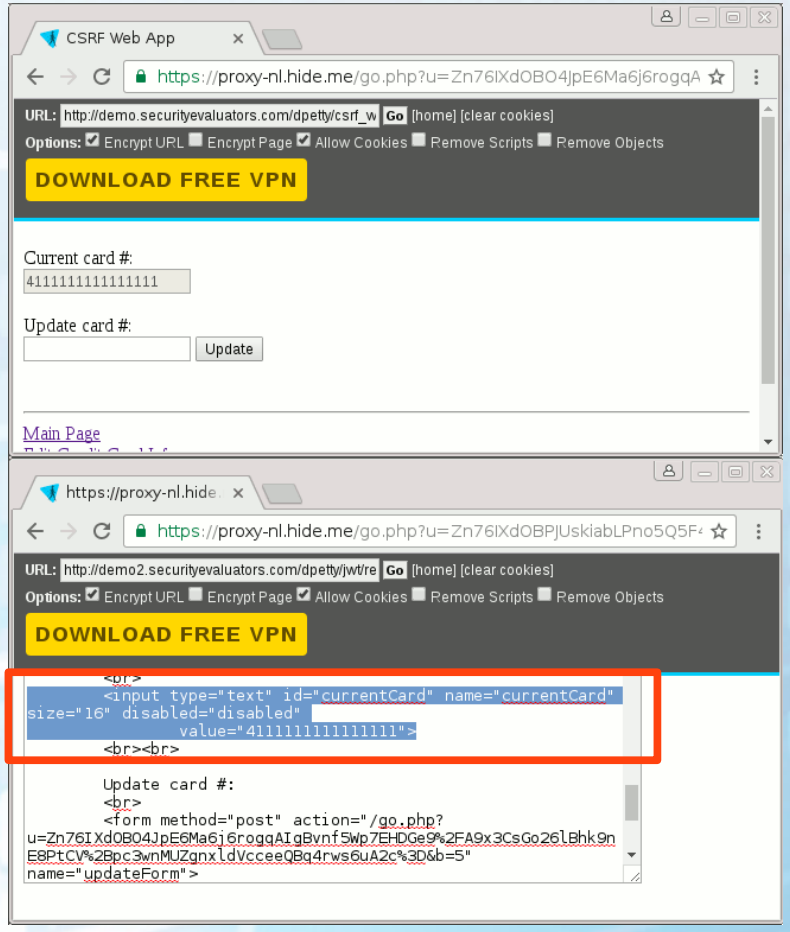
# Session Leak – Normal (safe)

# Session Leak – Proxy (unsafe)



# Same Origin – Normal (safe)

# Same Origin – Proxy (unsafe)



# Mitigations by Mirror Sites

# Cookie Blocking

- HTTP headers
- JavaScript *document.cookie*
- Breaking legitimate sites
- Verdict: no good

# Script/Object Stripping

- Breaks legitimate sites
- Verdict:  no good

# Clearing/Resetting Cookies

- Interesting?
- Verdict:  maybe

# One-to-One Unique Hostnames

- e.g., www.example.com => fBdnswUStgA08LmGKhIlg.proxy.example.net, www.example.org => STDMnt7vcf7ii72NzUA.proxy.example.net
- Verdict: probably

Mirroring still seems risky regardless of these strategies...

# Mitigations by Websites?

# HttpOnly Flag

- Protects against cookie leakage via JavaScript
- Can't protect against same origin policy bypass attack
- Verdict: no

# Authorization Header

- E.g., as in RESTful APIs
- Can't be used in all situations
- Verdict:  maybe

# Hardening Headers

- Strict Transport Security
- Public Key Pinning
- Any workable proxy would have to strip these…
- Verdict:  no

# Blocking Proxy Sites

- Bad for usability?
- Where would blacklist be found?

# No great solution to protect applications from broken mirror sites

# End users

- Use real proxies for anonymization (e.g., Tor)?
- Caution with static mirrors

# Future Work

- Only about one week thinking about this issue
- Worse exploits?
- More examples?
- Better mitigation techniques?
- Blog to follow

# Questions?