

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

Jacob Thompson

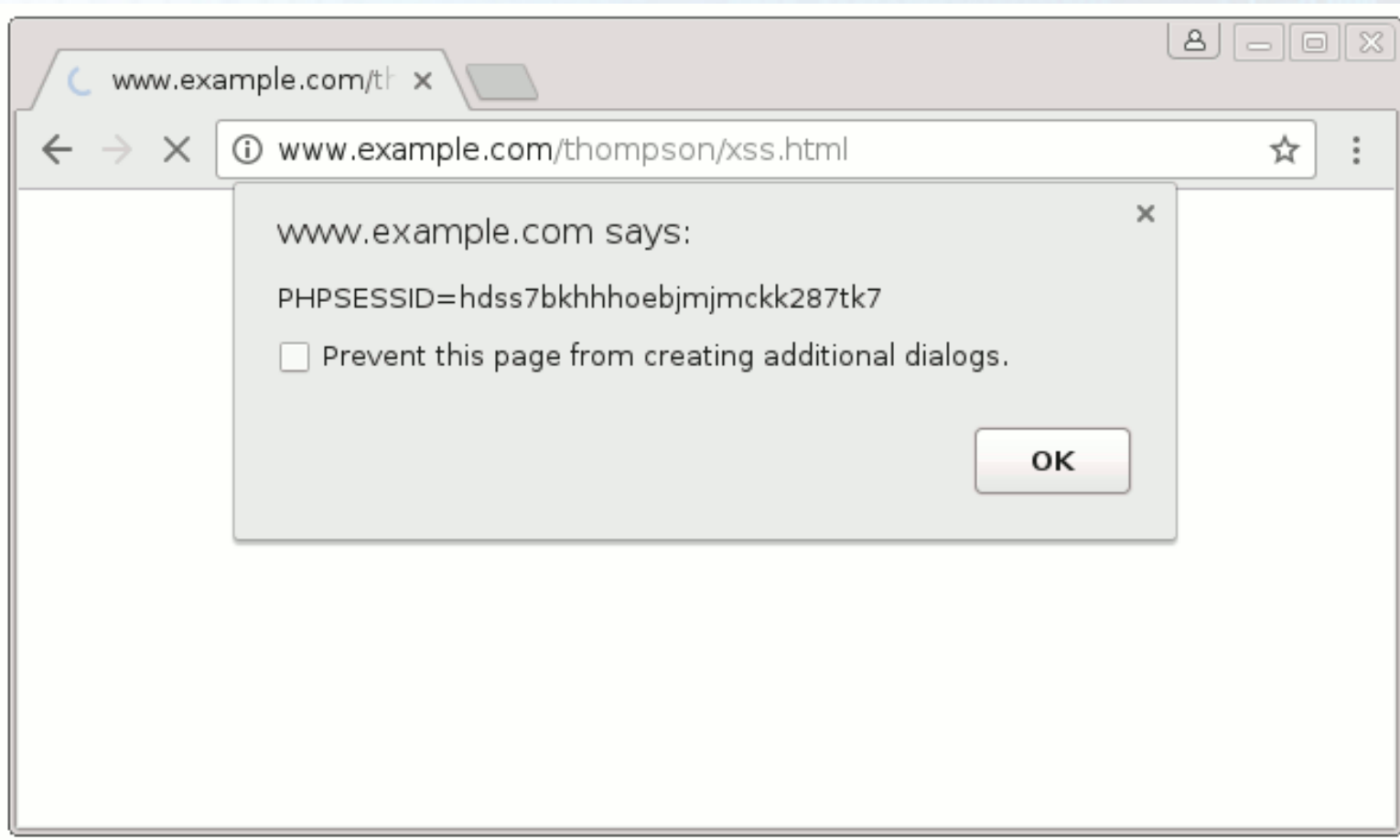
October 16, 2016



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



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
- PHPProxy 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

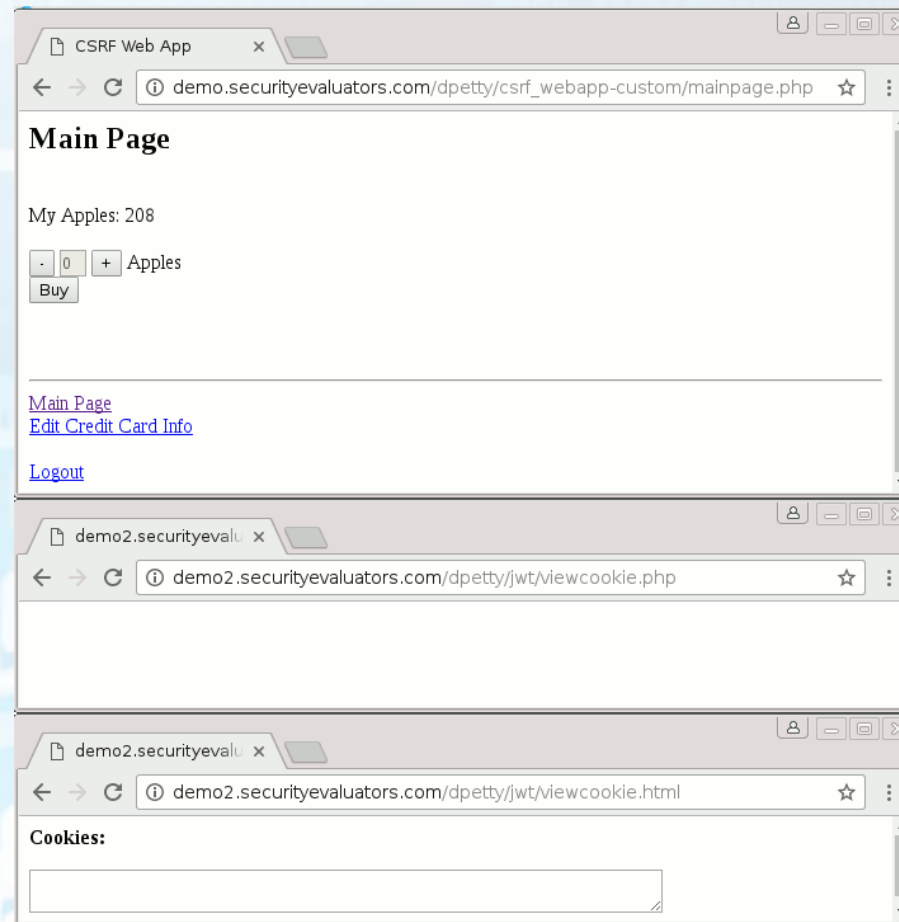
History Leak – Normal (safe)



History Leak – Translator (unsafe)



Session Leak – Normal (safe)

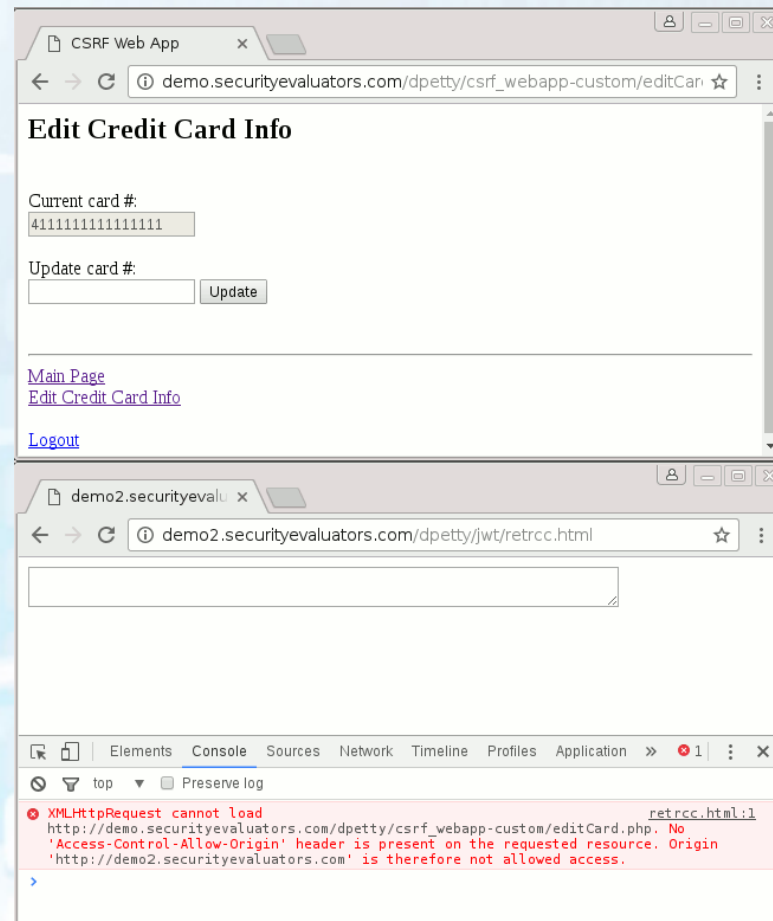


Session Leak – Proxy (unsafe)

The image displays three browser screenshots illustrating a session leak through a proxy. The first screenshot shows a browser window titled "CSRF Web App" with the URL `https://proxy-nl.hide.me/go.php?u=vOdCjDkFBYIUuQ11Don4R8Vn%2BY62`. The page content includes a "DOWNLOAD FREE VPN" button, a "My Apples: 208" counter, and a "Buy" button. The second screenshot shows the same proxy URL but with a different URL in the address bar: `https://proxy-nl.hide.me/go.php?u=vOdCjDkFBZ5jrtwjCZLIStl07oOvuzWS%:`. The third screenshot shows the same proxy URL but with a different URL in the address bar: `https://proxy-nl.hide.me/go.php?u=vOdCjDkFBZ5jrtwjCZLIStl07oOvuzWS%:`. The browser's developer tools are open, showing the following cookies:

```
gat=1; __cc_vist_tor_id_Sz6z691=S1476250940_630a01b078;
_ga=GA1.2.1299305949.1476250941;
c[securityevaluators.com][1]
[PHPSESSID]=qdebgvo1jp3oibm6p8472i0c06;
_ga=GA1.2.1299305949.1476250941
```

Same Origin – Normal (safe)



Same Origin – Proxy (unsafe)

The image displays two browser windows illustrating a proxy-based CSRF attack. The top window shows the original application page with a 'Current card #' field containing the value '4111111111111111'. The bottom window shows the proxy page, where the same field is highlighted in red. The HTML code for this field is shown below the proxy page, indicating that the proxy is correctly reflecting the original page's content.

```
<br>
<input type="text" id="currentCard" name="currentCard"
size="16" disabled="disabled" value="4111111111111111">
<br><br>
Update card #:
<br>
<form method="post" action="/go.php?
u=Zn76IXdOB04JpE6Ma6j6roggAIgBvnf5Wp7EHDGe9%2FA9x3CsGo26L8hk9n
EBPTCV%2Bpc3wnMUZgnxLdVcceeQBq4rws6uA2c%3D&b=5"
name="updateForm">
```

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 => fBdnswUStgA08LmGKhllg.proxy.example.net,`
`www.example.org => STDMMnt7vcf7ii72NzUA.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?