Signing Me onto Your Accounts through Facebook and Google: a Traffic-Guided Security Study of Commercially Deployed Single-Sign-On Web Services

Presentation: Dimitriadis Evangelos
Single Sign-On (SSO)

- Enables a user to log in once and gain access to multiple websites.
- Companies such as Facebook, Google, Yahoo, Twitter and PayPal all offer SSO services.
Motivation of this research

- Design formal methods to find protocol flaws
- Extent these to real systems:
  - The protocol serves merely as a guideline
  - Integrating web APIs, SDKs and sample code offered by the SSO providers
  - Vulnerabilities that do not show up on the protocol level
Methodology

● Three parties:
  ○ the user represented by a browser
  ○ the ID provider (a.k.a, IdP, e.g., Facebook)
  ○ and the relying party (a.k.a, RP)

● An automated black-box test on the HTTP messages (browser relayed messages, BRMs).

Figure 2: an SSO triangle and our visibility as an outsider
Browser Relayed Messages (BRMs)

1. An HTTP 3xx redirection response
2. A response including a form for automatic submission
3. A response with a script or a Flash object to make a request

Example:

code smuggling

```
src=a.com   dst=Facebook.com/a/foo.php
cookies:  sessionID=6739485
x=123 & user=john
fbs=a1b2c3 & foo=43da2c2a                                            
```

Set-

Arguments:

Cookies:
Attacking Scenarios

A. Bob is a client in an SSO and attempts to convince the RP.
B. When Alice visits Bob's website, Bob acts as an RP to the IdP.
C. Bob leaves malicious web content in Alice's browser during her visiting in his website.

Figure 5: three basic types of exploitations by Bob
The BRM Analyzer

- UU (user-unique)
- MU (client-machine-unique)
- SU (session-unique)
- BG (browser-generated)
- pChain (propagation chain)
- NC (newly-created)
- SIG (signature)
- SEC (secret)
Scenario (A): Bob acts as a browser

- All elements are readable
- An element not covered by a signature is writable
- For an element protected by a signature, if it is newly created (NC), then it is not writable; otherwise, inherit the writability label from its ancestor using pChain.
Scenario (B): Bob acts as an RP to the IdP in order to get Alice's credential for the target RP

- Replace any occurrence of “RP” in the trace with “Bob”
- For any BRM sent to Bob (or the dst element is writable), all Argument or Cookie elements in the BRM are readable
- For any BRM made by Bob, the dst element, or any Argument or Set-cookie element in the BRM is writable, if the element is not protected by the IdP’s signature
- For an element protected by a signature, if it is newly created (NC), then it is not writable; otherwise, inherit the writability label from its ancestor using pChain.
Scenario (C): Bob deposits a page in Alice’s browser

- No element is readable
- Cookies and set-cookies are not writable
- Because the BRM can be generated by Bob, the dst element or any Argument element in a BRM is writable, if the element is not protected by a signature;
- For an element protected by a signature, if it is newly created (NC), then it is not writable; otherwise, inherit the writability label from its ancestor using pChain.
Google ID

Does the RP check whether the email element in BRM3 is protected by the IdP’s signature, even though the protection has been explicitly required by BRM1?
Facebook

- BRM3 carries a secret token result from BRM2.
- All elements in BRM1, including app_id, are writable.
Facebook

- Facebook still protects the secret token result due to same-origin policy.
- Is it possible to let Flash B deliver the secret to the web page from Bob.com?
- Is Flash A allowed to communicate with a Flash object from Bob.com?

Figure 11: The complete view of a benign BRM3
Freelancer.com

- **Goal:**
  - Link Alice’s Freelancer account to Bob’s Facebook account.
  - Two things must happen: (1) the page signs Alice’s browser onto Bob’s Facebook account, and then (2) it makes the browser do the linking.

```
| BRM1: src=RP dst=http://!IdP/permissions.req |
| Arguments: app_id[BLOB] & cb[SEC] [BG] & |
| next [URL] {
|   http://!IdP/connect/xd_proxy.php? |
|   origin[BLOB] & transport[WORD] |
| } & ... & ... & ... (other 14 elements) |
```

```
| BRM2: src=!IdP dst=http://!IdP/xd_proxy.php |
| Arguments: origin[BLOB] & transport[WORD] | |
| result[SEC] & ... & ... (other 4 elements) |
```

```
| BRM3: src=!IdP dst=http://RP/facebook/lnk.php |
| Arguments: auto_link[INT] & goto_url[URL] |
| Cookies: fbs[SEC] |
```

Figure 14: Traffic for scenario (C)
Questions