Protecting Web-based Single Sign-on Protocols against Relying Party Impersonation Attacks through a Dedicated Bi-directional Authenticated Channel

Yinzhi Cao
Yan Shoshitaishvili
Kevin Borgolte
Christopher Kruegel
Giovanni Vigna
Yan Chen

University of California, Santa Barbara
Northwestern University

September 17th, 2014

RAID 2014 — Authentication & Privacy
Roadmap

- Single Sign-on
- Threat Model
- Problems with Existing Designs
- Our Design
- Evaluation
Single Sign-on (SSO) (1)

- Idea: log in to a website with your Facebook, Google, etc. account
Single Sign-on (SSO) (1)

• Idea: log in to a website with your Facebook, Google, etc. account
Single Sign-on (SSO) (1)

- Idea: log in to a website with your Facebook, Google, etc. account
Single Sign-on (SSO) (1)

- Idea: log in to a website with your Facebook, Google, etc. account
Single Sign-on (SSO) (1)

- Idea: log in to a website with your Facebook, Google, etc. account
OAuth 2.0 Flow

Problems

- SSO vulnerabilities mean
  - User impersonation
  - Data/privacy leaks
Problems

- SSO vulnerabilities mean
  - User impersonation
  - Data/privacy leaks

- Vulnerabilities are prolific
  - Wang et al. identified five vulnerabilities in which an attacker can impersonate a user [Oakland ’12].
  - Sun et al. show that 6.5% of relying parties are vulnerable to impersonation attacks [CCS ’12].
Threat Model - Concepts

- Identity provider (IdP)
  - A centralized identification service
  - Trusted and benign

- Relying party (RP)
  - A third party using the IdP to authenticate users
  - Potentially malicious

- User
  - Wants to use the RP’s service
  - Trusted and benign
Threat Model - Attacks (1)

- In-scope
  - Benign RP initiates request, malicious RP receives response
Threat Model - Attacks (1)

- In-scope
  - Benign RP initiates request, malicious RP receives response

GET https://www.idp.com/login?
app_id=****&redirection_url=https://www.idp.com/granter?
next_url=https://www.rp.com/login

Host: www.idp.com

Referer: https://www.rp.com/login

Cookie: ****
Threat Model - Attacks (1)

- In-scope
  - Benign RP initiates request, malicious RP receives response

GET https://www.idp.com/login?
app_id=****&redirection_url=https://www.idp.com/granter?
next_url=https://www.rp.com/login

Host: www.idp.com

Referer: https://www.rp.com/login

Cookie: ****
Threat Model - Attacks (1)

- In-scope
  - Benign RP initiates request, malicious RP receives response
  - Malicious RP initiates the attack

GET https://www.idp.com/login?
app_id=****&redirection_url=https://www.idp.com/granter?
next_url=https://www.rp.com/login

Host: www.idp.com

Referer: https://www.rp.com/login

Cookie: ****
• In-scope
  • Benign RP initiates request, malicious RP receives response
  • Malicious RP initiates the attack

⇒ Information leakage or user impersonation!
Threat Model - Attacks (2)

- Out-of-scope
  - Social engineering
  - Compromised or vulnerable RP
  - Malicious user (browser)
  - Implementation issues
  - Privacy leaks
• Existing identities
  
  • IdP, usually web origin (<scheme, host, port>)
  
  • RP, unique identifier, depending on protocol, app_id or AppName
  
  • User, unique identifier like username or email address
Revisit - Identities

- Existing identities
  - IdP, usually web origin (<scheme, host, port>)
  - RP, unique identifier, depending on protocol, app_id or AppName
  - User, unique identifier like username or email address

Main issue: RP identifier can be forged.
• Communication between RP and IdP
Revisit - Communication

- Communication between RP and IdP
  - HTTP(s) redirection to 3rd party server (1-way channel)
Revisit - Communication

• Communication between RP and IdP
  • HTTP(s) redirection to 3rd party server (1-way channel)
  • In-browser communication channel (no authentication)
• Clean-slate design, replaces existing protocols
  • Identity
    • Web origin for RP and IdP: <scheme, host, port>
  • Communication channel
    • Dedicated
    • Bi-directional
    • Authenticated
    • Secure
Establishing the channel: handshake

1. \( PK_{RP} \)
2. Identity Check
3. \( PK_{RP}(SK, N_{IdP}) \)
4. \( SK(N_{RP}) \)
5. \( N_{IdP}, N_{RP}, SK(CB, msg) \)

PK\(_{RP}\): Public Key of RP
SK: Session Key
\( N_{IdP}\): Channel Number of IdP
\( N_{RP}\): Channel Number of RP
CB: Control Byte
Identity Provider Deployment (2)

- Establishing the channel: handshake
- Sending messages
• Establishing the channel: handshake
• Sending messages
• Receiving messages
Identity Provider Deployment (2)

- Establishing the channel: handshake
- Sending messages
- Receiving messages
- Terminating the connection: releasing resources
Relying Party / Proxy Deployment

- Allows smooth transition to more secure protocol
  - Does not require you to replace existing protocol
- Proxy communicates with legacy IdP
- RPs communicate with proxy
• Allows smooth transition to more secure protocol
• Does not require you to replace existing protocol
• Proxy communicates with legacy IdP
• RPs communicate with proxy
Implementation

• Prototype implementation
  • Clean-slate / IdP deployment
    • Two protocols: OpenID-like and OAuth-like
    • 252 LOC JavaScript, 264 LOC HTML, 243 LOC PHP
    • External libraries: JavaScript Cryptography Toolkit + Stanford JavaScript Crypto Library
  • Proxy / RP deployment
    • Based on a Facebook application
Evaluation - Formal Verification

- Formally verified design with ProVerif
  - Channel verification
    - Attacker: passive (sniffing), active (sending messages)
    - Result: an attacker cannot obtain the plain text message
  - Protocol verification
    - Attacker: network (passive) and web attackers (active)
    - Result: an attacker cannot obtain any useful information
  - Proxy verification
    - Attacker: passive (sniffing), active (sending messages)
    - Result: an attacker can obtain and modify the messages sent over the insecure communication channel between proxy and legacy IdP
Evaluation - Security Analysis

• Our protocol prevents all impersonation attacks identified by Wang et al. [Oakland ’12]:
  • Facebook and New York Times
  • Facebook and Zoho
  • Facebook Legacy Canvas Auth
  • JanRain wrapping GoogleID
  • JanRain wrapping Facebook
Channel operation

<table>
<thead>
<tr>
<th>Operation</th>
<th>Delay [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Establishing the channel</td>
<td>164±12</td>
</tr>
<tr>
<td>Sending a message</td>
<td>32±2</td>
</tr>
<tr>
<td>Destroying a channel</td>
<td>70±3</td>
</tr>
</tbody>
</table>
## Evaluation - Performance

**Channel operation**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Delay [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Establishing the channel</td>
<td>164±12</td>
</tr>
<tr>
<td>Sending a message</td>
<td>32±2</td>
</tr>
<tr>
<td>Destroying a channel</td>
<td>70±3</td>
</tr>
</tbody>
</table>

**Establishing the channel**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Delay [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Message #1: PK_RP</td>
<td>92±9</td>
</tr>
<tr>
<td>Message #2: PK_RP(SK, N_IdP)</td>
<td>29±2</td>
</tr>
<tr>
<td>Message #3: SK(N_RP)</td>
<td>43±3</td>
</tr>
</tbody>
</table>
Detailed breakdown of the protocol

<table>
<thead>
<tr>
<th>Operation</th>
<th>Delay [ms]</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Creating the channel between RP and IdP</td>
<td>164±11</td>
</tr>
<tr>
<td>(2) Creating the IdP inline frame</td>
<td>57±3</td>
</tr>
<tr>
<td>(3) Sending the first message from RP to IdP</td>
<td>32±2</td>
</tr>
<tr>
<td>(4) Creating the IdP inline frame for authentication</td>
<td>57±3</td>
</tr>
<tr>
<td>(5) Creating the second channel inside the IdP</td>
<td>165±11</td>
</tr>
<tr>
<td>(6) Authenticating the user</td>
<td>56±4</td>
</tr>
<tr>
<td>(7) Requesting the user’s permissions</td>
<td>57±3</td>
</tr>
<tr>
<td>(8) Sending the token inside the IdP’s inline frame</td>
<td>32±2</td>
</tr>
<tr>
<td>(9) Sending the token to the RP</td>
<td>33±2</td>
</tr>
</tbody>
</table>

Total 653±21

(2), (4), (6), and (7) are dominated by network latency, which is 50ms here.
Conclusion

- Pointed out root cause why RPI attacks exist: non-dedicated, insecure, one-way channel between RP and IdP
Conclusion

• Pointed out root cause why RPI attacks exist: non-dedicated, insecure, one-way channel between RP and IdP

• Proposed a dedicated bi-directional secure channel to remedy existing short-comings
• Pointed out root cause why RPI attacks exist: non-dedicated, insecure, one-way channel between RP and IdP

• Proposed a dedicated bi-directional secure channel to remedy existing short-comings

• Designed SSO protocol on top of channel design
Conclusion

• Pointed out root cause why RPI attacks exist: non-dedicated, insecure, one-way channel between RP and IdP
• Proposed a dedicated bi-directional secure channel to remedy existing short-comings
• Designed SSO protocol on top of channel design
• Presented a proxy design for easy adoptability
• Pointed out root cause why RPI attacks exist: non-dedicated, insecure, one-way channel between RP and IdP
• Proposed a dedicated bi-directional secure channel to remedy existing short-comings
• Designed SSO protocol on top of channel design
• Presented a proxy design for easy adoptability
• Formally verified security of the SSO protocol
Conclusion

• Pointed out root cause why RPI attacks exist: non-dedicated, insecure, one-way channel between RP and IdP

• Proposed a dedicated bi-directional secure channel to remedy existing short-comings

• Designed SSO protocol on top of channel design

• Presented a proxy design for easy adoptability

• Formally verified security of the SSO protocol

• Evaluated protocol performance / overhead
Thank you for your attention!

Questions?
## Related Work

<table>
<thead>
<tr>
<th>Deployment</th>
<th>Protection Crowd</th>
<th>Preventing Impersonation Attacks</th>
<th>Proactive Deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>InteGuard: IdP, Gateway</td>
<td>IdP Users, physical machines</td>
<td>✔️</td>
<td>❌</td>
</tr>
<tr>
<td>AuthScan: IdP</td>
<td>IdP Users</td>
<td>✔️</td>
<td>❌</td>
</tr>
<tr>
<td>Explicating SDKs: IdP</td>
<td>IdP Users</td>
<td>✔️</td>
<td>❌</td>
</tr>
<tr>
<td>Defensive JavaScript: IdP, RP</td>
<td>IdP Users, RP Users</td>
<td>❌</td>
<td>✔️</td>
</tr>
<tr>
<td>WebSSO (our work): IdP, RP</td>
<td>IdP Users, RP Users</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>