On the Privacy Practices of Just Plain Sites

Amirhossein Aleyasen, Oleksii Starov, Alyssa Au Allan Schiffman, Jeff Shrager

WPES 2015
# Just Plain Sites (JPSs)

<table>
<thead>
<tr>
<th>JPSs</th>
<th>Not JPSs</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Small mom-and-pop shops, food delivery*</td>
<td>• Too Big: Google, Facebook, Amazon, Craigslist, etc.</td>
</tr>
<tr>
<td>• Local teams and clubs, social networks*</td>
<td>• Too Regulated: Healthcare, Finance</td>
</tr>
<tr>
<td>• Personal blogs and community forums</td>
<td>• Too Small: Under 300 visitors/month</td>
</tr>
<tr>
<td></td>
<td>• Governmental</td>
</tr>
</tbody>
</table>

*Note the possibility of Drive-By-Login attacks

Aleyasen, Starov, Au, Schiffman, & Shrager; WPES 2015
JPSs and Privacy

- JPSs have less resources to spend on following privacy and security best practices for their users (even if they do these for themselves)
- JPSs are under less scrutiny by regulators
- JPS operators may not know about obscure privacy considerations, and may not care when they are “useless”
“Useful” vs. “Useless” Privacy Violations

- “Useful” privacy violations serve the business purposes of the JPS.
- “Useless” privacy violations do NOT serve the business purposes of the JPS.

We are primarily concerned with “useless” privacy violations because they are likely to be elastic – i.e., the JPS operator would be willing to fix it if they knew it was there.

* This is orthogonal to intentional/unintentional; We are mainly concerned with intentional and useless violations. (Unintentional ones can be fixed, and intentional+useful are probably inelastic.)
HopScotch (gethopscotch.com) is an iPad app that offers a programming community for kids (esp. girls). Located in NYC, it has <10 (mostly young) employees, and ~500k (mostly very young) users. (Subcommunities are way smaller.)

In a release that one of us (JS) beta tested, they asked for their users’ exact birthday in order to get their age. JS pointed out that this was unnecessary; they could just ask for the user’s age. (It’s as easy to lie about your as about your age!)
Example

On Mon, Jun 29, 2015 at 10:48 PM EDT, Jeff Shrager <jshrager@stanford.edu> wrote:
...
Should NOT ask for complete bd; prob age, or m/y are enough. Keeping people's full BDs, esp. kids' can be a privacy risk.
...

On Jun 30, 2015, at 4:34 PM, -------------- <help@gethopscotch.com> wrote:
...
Good idea re: ages. If we can accomplish the same thing with less sensitive info, better!
...

Duh!

Aleyasen, Starov, Au, Schiffman, & Shrager; WPES 2015
Privacy Principles for JPSs

① Minimize collection
② Minimize sharing
③ Protect what you collect
④ Post and follow a privacy policy
⑤ Give users choice and access
“Front of the Store” Risks

• When you’re “just browsing” at the “front of the store”, you assume that your privacy is not at risk (at least from the proprietor of the store). Privacy risks are unexpected.

• Therefore (on the Web) we focus on landing page contents and page load activities.

Aleyasen, Starov, Au, Schiffman, & Shrager; WPES 2015
Method

- Source: Quantcast sites with ranking <50,000 =~ 300-30,000 visitors/month. Dropped .gov, leaving ~1M sites (943,489)
- Analyzed “on page load” operations by several methods, incl. rough source analysis, detailed form analysis, and on load activity sniffing
- ~100 random cases manually analyzed for categorization, detailing, QA, and FB login call methodology; also created accounts for these cases to examine password processing practices
① Minimize Collection

JPSs should collect only what they need and nothing more

Aleyasen, Starov, Au, Schiffman, & Shrager; WPES 2015
JPSs Ask for Personal Information in Web Forms

Aleyasen, Starov, Au, Schiffman, & Shrager; WPES 2015
Extensive Collection is a Risky Practice

- Around 50% of JPSs collect more sensitive Personally Identifiable Information (PII)
- Some personal information collected is redundant for JPSs, e.g. asking for DOB in 25% of cases is irrational

Aleyasen, Starov, Au, Schiffman, & Shrager; WPES 2015
## Permissions asked for Facebook Login

### Custom FB Login (260 websites)

<table>
<thead>
<tr>
<th>Permission</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>email</td>
<td>89.62%</td>
</tr>
<tr>
<td>user_birthday</td>
<td>27.69%</td>
</tr>
<tr>
<td>publish_stream</td>
<td>23.46%</td>
</tr>
<tr>
<td>user_location</td>
<td>16.54%</td>
</tr>
<tr>
<td>read_stream</td>
<td>10.00%</td>
</tr>
<tr>
<td>user_about_me</td>
<td>9.23%</td>
</tr>
<tr>
<td>offline_access</td>
<td>9.23%</td>
</tr>
<tr>
<td>user_likes</td>
<td>8.85%</td>
</tr>
<tr>
<td>&lt;default&gt;</td>
<td>8.08%</td>
</tr>
<tr>
<td>publish_actions</td>
<td>7.69%</td>
</tr>
<tr>
<td>friends_location</td>
<td>6.15%</td>
</tr>
<tr>
<td>user_photos</td>
<td>5.77%</td>
</tr>
<tr>
<td>user_hometown</td>
<td>5.38%</td>
</tr>
<tr>
<td>user_website</td>
<td>4.23%</td>
</tr>
<tr>
<td>user_work_history</td>
<td>4.23%</td>
</tr>
</tbody>
</table>

### Facebook Login Button (612 websites)

<table>
<thead>
<tr>
<th>Permission</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>email</td>
<td>66.83%</td>
</tr>
<tr>
<td>user_birthday</td>
<td>32.84%</td>
</tr>
<tr>
<td>&lt;default&gt;</td>
<td>31.05%</td>
</tr>
<tr>
<td>publish_stream</td>
<td>21.57%</td>
</tr>
<tr>
<td>offline_access</td>
<td>12.91%</td>
</tr>
<tr>
<td>user_about_me</td>
<td>12.09%</td>
</tr>
<tr>
<td>user_location</td>
<td>10.29%</td>
</tr>
<tr>
<td>user_likes</td>
<td>8.33%</td>
</tr>
<tr>
<td>read_stream</td>
<td>6.86%</td>
</tr>
<tr>
<td>status_update</td>
<td>5.39%</td>
</tr>
<tr>
<td>user_status</td>
<td>4.90%</td>
</tr>
<tr>
<td>publish_actions</td>
<td>4.41%</td>
</tr>
<tr>
<td>user_website</td>
<td>4.25%</td>
</tr>
<tr>
<td>user_hometown</td>
<td>3.10%</td>
</tr>
<tr>
<td>user_photos</td>
<td>2.61%</td>
</tr>
</tbody>
</table>

Aleyasen, Starov, Au, Schiffman, & Shrager; WPES 2015
Revealed Risks with FB Login Permissions

- **30%** of JPSs ask for exact DOB, regardless of Facebook guidelines to ask for age range.

- **12%** of JPSs use deprecated permissions, i.e. do not update their privacy practices, *that is high comparing to non-JPS studies*.

- JPSs may ask for more detailed information like your location and location of your friends, e.g. *websites of politicians*!
Minimize Sharing

Third-party sharing that happens on a web page is risky
When you visit a JPS, **7 other organizations** may know it!

Aleyasen, Starov, Au, Schiffman, & Shrager; WPES 2015
Similarly to top websites, many JPSs include lots of remote JavaScript

More than 50% JPSs use third-party analytics

50% of JPS websites use third-party cookies that may live for years

Prevalent third-parties are “big players”: Google and Facebook
③ Protect the Data You Collect

JPSs should have reasonable security precautions with regards to data storage

Aleyasen, Starov, Au, Schiffman, & Shrager; WPES 2015
5-10% JPSs email passwords explicitly!

Because of Magento and Wordpress default settings

Aleyasen, Starov, Au, Schiffman, & Shrager; WPES 2015
④ Post a Privacy Policy and Follow It

In this work, we actually don’t care what the privacy policy says in the case of JPSs, e.g. it may well have been copied from someplace else.

What we care about is *what the site actually does!*
⑤ Give Users Choice and Access

Technologically hard to measure on the server side, might be a future work

Aleyasen, Starov, Au, Schiffman, & Shrager; WPES 2015
Just Plain Mobile Apps

- What solution Web JPSs use to have mobile clients as well?
- What is the difference in privacy practices and requirements?
- Are there standalone JPMAs?
- How to analyze?
Identifying JPMAs

- Accessing mobile version of a JPS, e.g. “m.” or “.mobi”
  - Only 10% have a separate mobile site

- Searching Google Play by a JPS’s URL
  - Usually JPSs are not app developers

Aleyasen, Starov, Au, Schiffman, & Shrager; WPES 2015
Privacy Practices

**JPSs**
- Collection via HTML Forms can be detected by the special crawlers
- Many remote JS scripts and third-parties are present on a web page
- Browser helps to protect against sharing to third-parties, communication is secured on top of HTTP

**JPMAs**
- Scraping the UI is more complicated, dynamic components in code
- Many third-party libraries used, e.g. analytics works as a first-party inclusion
- Various ways of transmitting the data to back-end server or implement the inputs

Aleyasen, Starov, Au, Schiffman, & Shrager; WPES 2015
Discussion Starters

- Are we making a mountain out of a molehill? Are JPSs too small a problem to be concerned with?

- How does the JPS concept apply to ubiquitous/embedded computing?

- Although more info can be grabbed from mobile (and embedded) devices, app development is harder, and so usually done by professionals

- OTOH, the pros can be the ones grabbing the info!

- Is the "kids coding" movement dangerous in the "cloud era"?

- What happens when young programmers not attuned to privacy concerns start contributing to Open Source projects?