CSE 484 / CSE M 584
Computer Security: Web Security

TA: Franzi Roesner
franzi@cs.washington.edu
Logistics

• Homework #2 (crypto) due 2/22 5pm.
• Lab #2 (web security) due 2/27 5pm.
  – If you haven’t signed up your group yet (email me UW NetIDs, groups name, password), do it now!
  – Not just any SQL injection will work.
SQL Injection

What if this web app does something like this:

```
select * from users where user_name='$user' and user_password=''$password'
```

Attacker can log in by entering the username:

```
a' or '1'='1'; --
```

Why? SQL will execute:

```
select * from users where user_name = 'a' or '1'='1';
```
SQL Injection

What if this web app does something like this:

```sql
select * from users where user_name='$user' and user_password=''$password''
```

Attacker can execute arbitrary SQL commands by entering the user name:

```sql
a'; <Other Commands> --
```

For example: ```sql
a'; DROP TABLE users; --
```
SQL Injection

Hi, this is your son's school. We're having some computer trouble.

Oh, dear - did he break something? In a way-

Did you really name your son Robert'); DROP TABLE Students;-- ?

Oh, yes. Little Bobby tables, we call him.

Well, we've lost this year's student records. I hope you're happy.

And I hope you've learned to sanitize your database inputs.

[http://xkcd.com/327/]
Clickjacking using the Cursor

Figure 1: Cursor spoofing attack page. The target Flash Player webcam settings dialog is at the bottom right of the page, with a "skip this ad" bait link remotely above it. Note there are two cursors displayed on the page: a fake cursor is drawn over the "skip this ad" link while the actual pointer hovers over the webcam access "Allow" button.

[Figure from Huang et al., “Clickjacking: Attacks and Defenses”, USENIX Security, 2012]
Other Web Security Resources

• Web Security Codelab: 
  http://google-gruyere.appspot.com/

• http://uwctf.cs.washington.edu/learntocook.php

• Clickjacking: 
  http://www.grc.com/sn/notes-168.htm

• SQL Injection: http://sqlzoo.net/hack/
Detecting and Defending Against Third-Party Tracking on the Web

Franziska Roesner, Tadayoshi Kohno, David Wetherall
Third-Party Web Tracking

Bigger browsing profiles
= increased value for trackers
= reduced privacy for users

(Hypothetical tracking relationships only.)
Tracking is Complicated

• Much discussion of tracking, but limited understanding of how it actually works.

• Our goals:
  – Understand the tracking ecosystem.
    • How is tracking actually done in the wild?
    • What kinds of browsing profiles do trackers compile?
    • How effective are defenses available to users?
  – Address gaps with new defense (ShareMeNot).
Mechanisms Required By Trackers

• **Ability to store user identity** in the browser
  – Browser cookies
  – HTML5 LocalStorage and Flash cookies (LSOs)
  – Not considering more exotic storage mechanisms or approximate fingerprinting

• **Ability to communicate** visited page and user identity **back to tracker**
  – Identity: Cookies attached to requests
  – Visited page: HTTP referrers
  – Both: scripts that embed information in URLs
Tracking: The Simple Version

- **Within-Site**: First-party cookies are used to track repeat visits to a site.
- **Cross-Site**: Third-party cookies are used by trackers included in other sites to create profiles.

---

Cookie Database

- tracker.com: id=789

**http://site2.com**

```html
<iframe src="tracker.com/ad.html">
  
  ad
  
</iframe>
```

**cookie: id=789**

**9:30am**: user 789 visited site1.com

**9:31am**: user 789 visited site2.com

**logs**

**tracker.com**

**processing engine**
# Our Tracking Taxonomy

<table>
<thead>
<tr>
<th>Name</th>
<th>Scope</th>
<th>User Visits Directly?</th>
<th>Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Within-Site</td>
<td>Yes</td>
<td>Site does its own on-site analytics.</td>
</tr>
<tr>
<td><strong>Evolution: Embedding analytics libraries</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Analytics</td>
<td>Within-Site</td>
<td>No</td>
<td>Site uses third-party analytics engine (e.g., Google Analytics).</td>
</tr>
<tr>
<td>Vanilla</td>
<td>Cross-Site</td>
<td>No</td>
<td>Site embeds third-party tracker that uses third-party storage (e.g., Doubleclick).</td>
</tr>
<tr>
<td><strong>Evolution: Third-party cookie blocking</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forced</td>
<td>Cross-Site</td>
<td>Yes (forced)</td>
<td>Site embeds third-party tracker that forced the user to visit directly (e.g., via popup).</td>
</tr>
<tr>
<td>Referred</td>
<td>Cross-Site</td>
<td>No</td>
<td>Tracker relies on another cross-site tracker to leak unique identifier values.</td>
</tr>
<tr>
<td>Personal</td>
<td>Cross-Site</td>
<td>Yes</td>
<td>Site embeds third-party tracker that the user otherwise visits directly (e.g., Facebook).</td>
</tr>
</tbody>
</table>
Quirks of Third-Party Cookie Blocking

• Option blocks the **setting** of third-party cookies: all browsers

• Option blocks the **sending** of third-party cookies: **only Firefox**

• Result: Once a third-party cookie is somehow set, **it can be used** (in most browsers).
# Our Tracking Taxonomy

<table>
<thead>
<tr>
<th>Name</th>
<th>Scope</th>
<th>User Visits Directly?</th>
<th>Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Within-Site</td>
<td>Yes</td>
<td>Site does its own on-site analytics.</td>
</tr>
<tr>
<td><strong>Analytics</strong></td>
<td>Within-Site</td>
<td>No</td>
<td>Site uses third-party analytics engine (e.g., Google Analytics).</td>
</tr>
<tr>
<td><strong>Vanilla</strong></td>
<td>Cross-Site</td>
<td>No</td>
<td>Site embeds third-party tracker that uses third-party storage (e.g., Doubleclick).</td>
</tr>
<tr>
<td><strong>Forced</strong></td>
<td>Cross-Site</td>
<td>Yes (forced)</td>
<td>Site embeds third-party tracker that forced the user to visit directly (e.g., via popup).</td>
</tr>
<tr>
<td>Referred</td>
<td>Cross-Site</td>
<td>No</td>
<td>Tracker relies on another cross-site tracker to leak unique identifier values.</td>
</tr>
<tr>
<td>Personal</td>
<td>Cross-Site</td>
<td>Yes</td>
<td>Site embeds third-party tracker that the user otherwise visits directly (e.g., Facebook).</td>
</tr>
</tbody>
</table>

**Evolution:**
- Embedding analytics libraries
- Third-party cookie blocking
- Complex ad networks
Referred Tracking

Cookie Database

tracker.com: id=522

http://site1.com

<iframe src=tracker.com/ad.html>

ad

</iframe>

Tracker.com processing engine

http://site2.com

<iframe src=tracker.com/ad.html>

ad

</iframe>

Othertracker.com processing engine

2:34pm: site1.com: user 522

2:35pm: site2.com: user 522

Othertracker.com logs
# Our Tracking Taxonomy

<table>
<thead>
<tr>
<th>Name</th>
<th>Scope</th>
<th>User Visits Directly?</th>
<th>Overview</th>
</tr>
</thead>
<tbody>
<tr>
<td>N/A</td>
<td>Within-Site</td>
<td>Yes</td>
<td>Site does its own on-site analytics.</td>
</tr>
<tr>
<td>Analytics</td>
<td>Within-Site</td>
<td>No</td>
<td>Site uses third-party analytics engine (e.g., Google Analytics).</td>
</tr>
<tr>
<td>Vanilla</td>
<td>Cross-Site</td>
<td>No</td>
<td>Site embeds third-party tracker that uses third-party storage (e.g., Doubleclick).</td>
</tr>
<tr>
<td>Forced</td>
<td>Cross-Site</td>
<td>Yes (forced)</td>
<td>Site embeds third-party tracker that forced the user to visit directly (e.g., via popup).</td>
</tr>
<tr>
<td>Referred</td>
<td>Cross-Site</td>
<td>No</td>
<td>Tracker relies on another cross-site tracker to leak unique identifier values.</td>
</tr>
<tr>
<td>Personal</td>
<td>Cross-Site</td>
<td>Yes</td>
<td>Site embeds third-party tracker that the user otherwise visits directly (e.g., Facebook).</td>
</tr>
</tbody>
</table>

**Evolution: Embedding analytics libraries**

- **Evolution: Third-party cookie blocking**
- **Evolution: Complex ad networks**
- **Evolution: Social networks**
Measurement Study

- Tool: TrackingTracker Firefox add-on that crawls the web and automatically categorizes trackers.
- 3 data sets
  - Alexa Top 500
    - 5 pages per domain: main page and up to 4 links
  - Alexa Non-Top 500
    - Sites ranked #501, #601, #701, etc.
    - 5 pages per domain: main page and up to 4 links
  - AOL search logs
    - 300 unique queries for 35 random users
Tracking Prevalence (Top 500)

- 524 unique trackers on 500 domains

- 457 domains (91%) embed at least one tracker. (97% of those include at least one cross-site tracker.)

- 50% of domains embed between 4 and 5 trackers.

- One domain includes 43 trackers.
Top 20 Trackers on Top 500 Domains

Tracker Prevalence (# Domains)

- Within-Site
- Cross-Site (Personal)
- Cross-Site (Anonymous)

- google-analytics.com: 297
- doubleclick.net: 189
- google.com: 154
- quantserve.com: 149
- twitter.com: 109
- tdmt.com: 105
- imrworldwide.com: 93
- revsci.net: 81
- advertising.com: 60
- addthis.com: 45
- adxns.com: 44
- serving-sys.com: 40
- youtube.com: 34
- addthiscdn.com: 33
- bluekai.com: 32
- mediaplex.com: 30
- invitemedia.com: 29
- mediaspike.com: 27
- thekay.com: 26
Each line represents one user.

AOL Users' Profile Sizes by Top 20 Cross-Site Trackers

- **Doubleclick**: Avg 39% (Max 66%)
- **Facebook**: Avg 23% (Max 45%)
- **Google**: Avg 21% (Max 61%)
LocalStorage and Flash Cookies

• Surprisingly little use of these mechanisms!

• Of 524 trackers on Alexa Top 500:
  – Only 5 set unique identifiers in LocalStorage
  – 35 set unique identifiers in Flash cookies

• Respawning:
  – LS $\rightarrow$ Cookie: 1 case; Cookie $\rightarrow$ LS: 3 cases
  – Flash $\rightarrow$ Cookie: 6 cases; Cookie $\rightarrow$ Flash: 7 cases