A Practical Attack to De-Anonymize Social Network Users

Gilbert Wondracek (Vienna University of Technology)
Thorsten Holz (Vienna University of Technology)
Engin Kirda (Institute Eurecom)
Christopher Kruegel (UC Santa Barbara)

http://iseclab.org
Motivation

- Social Network User Activity

- evil.com
  - Steals SN identity
    - i.e. profile, full name, photo, resume, etc.
    - More sensitive data (Google, Yahoo, etc.)
Related Works and Background

- Firesheep – most recent breach
- Koobface - Virus
- GIFAR and Account linking
- Phishing - Jagatic et al
- Pang et al. ; And the reaction Coulls et al.
- Non-sensitive data - Griffith and Jakobsson
- Communication patterns - Diaz et al.
- Non-destructive SN hackings
Attack Overview

- **Aim:** Find out the SN Identity of website users
  - Instead of tracking browsers track people
- **Use information that is public on the SN**
  - Limited to the number of SN users (100,000,000s+)
  - Leaked data from SNs and well-known browser attacks – IDs
- **Abuse Potential**
  - Intrusive advertisements / blackmail
  - Many victims
History Stealing

- Browser attack
  - Requires HTML and CSS (javascript helps)
  - Past visited URLs from CSS
- What does this reveal?
  - “Is <URL> in the user's browsing history?” - embedding links comparing style
    - Note: exhaustive list not possible but...
      - No limit to how many “questions”
      - Covertly done
History Stealing (cont.)

- Spear phishing – banks, etc.
- Browser's attention
- Browser history defaults
  - IE8 – 20 days
  - Firefox – 90 days
  - Chrome - Unlimited
History Stealing

1. List of web pages

2. Script checks locally which pages have been visited

3. Information about visited pages is sent back

Figure 2: Schematic overview of history stealing attack.
Social Network Vulnerabilities

• Web URLs – Ids and Names
  • http://sn.com/profile?operation=EditMyProfile&user=12345

Found in all 8 social networks in study. Examples:

  Facebook: facebook.com/ajax/profile/picture/upload.php?id=[UID]
  Xing: xing.com/net/[GID]/forums
  Amazon: amazon.com/tag/[GID]
  Ebay: community.ebay.de/clubstart.htm?clubid=[GID]
# Look at 8 SNs

<table>
<thead>
<tr>
<th>Name of social network</th>
<th># users</th>
<th>Focus</th>
<th>Alexa traffic rank [1]</th>
<th>Supports groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>Facebook</td>
<td>300,000,000+</td>
<td>general audience, worldwide</td>
<td>2</td>
<td>✓</td>
</tr>
<tr>
<td>MySpace</td>
<td>260,000,000+</td>
<td>music, worldwide</td>
<td>11</td>
<td>✓</td>
</tr>
<tr>
<td>Friendster</td>
<td>90,000,000+</td>
<td>general audience, worldwide</td>
<td>111</td>
<td>✓</td>
</tr>
<tr>
<td>LinkedIn</td>
<td>50,000,000+</td>
<td>business, worldwide</td>
<td>53</td>
<td>✓</td>
</tr>
<tr>
<td>StudiVZ</td>
<td>15,000,000+</td>
<td>students, Germany</td>
<td>179</td>
<td>✓</td>
</tr>
<tr>
<td>Xing</td>
<td>8,000,000+</td>
<td>business, Europe</td>
<td>285</td>
<td>✓</td>
</tr>
<tr>
<td>Bigadda</td>
<td>5,500,000+</td>
<td>teenage audience, India</td>
<td>3,082</td>
<td>✓</td>
</tr>
<tr>
<td>Kiwibox</td>
<td>2,500,000+</td>
<td>teenage audience, worldwide</td>
<td>74,568</td>
<td>✓</td>
</tr>
</tbody>
</table>

Table I: Overview of popular social networking websites. The data is based on information provided by individual social networks, public sources such as Alexa [1], and our analysis.
Basic Attack

- De-Anonymization Attack Protocol:
  - Combine History stealing and SN Info
  - Lure person to evil.com
  - User ID can be found from history stealing

- Attacking website can simply query for UIDs:
  
  sn.com/editprofile/html?uid=0
  sn.com/editprofile/html?uid=1
  ...
  sn.com/editprofile/html?uid=[X]

- Look up UID = [X] in social network
History Stealing Benchmark
Not very good....

- Only works for small SN (useless)
- Sns have lots of users
  - Many URLs to check via history stealing
- Not practical for real world attack
  - Time on a website
  - CPU usage and UI response
Improve the attack: Groups

• Groups = subsets of users with similar interest
• Can be open or closed
  • Open = public: anyone can join
  • Closed = Needs approval by group admins
• Hyperlinked and GID
• Stored in browser history
• Membership from history

- Example: www.sn.com/join_group.php?gid=12345
How to get group member info.?

- Most SN offer group directories
  - Public lists of other groups and members
  - Other interests
- Use this to collect group data:
  - Join group
  - List all members
  - Leave group
  - Repeat
- Eventually attackers knows all members of each group
Group Member Enumeration

- Many SN restricts full listing of (group) members (non-members)
  - Search feature abused
- Attacker can also use SN info itself
  - Groups in user profile - crawl the public member directory
  - Group IDs - “brute force crawled”
- Conclusion – attackers can get group info one way or another
Better Attack!

1) Preparation step: Crawl the targeted social network, get group and membership data

2) Lure victim to attack website

3) Use history stealing to check for links that indicate group membership

4) For these groups, look up the (crawled) members

5) Reduce the candidate set: Calculate intersection set for the found group members
   - If intersection set is empty (data may be inaccurate, history deleted etc), use the union set (slower, but more reliable)

6) Use basic attack on candidate set
   • Ideally, all but one profile will be eliminated – Success!
Evaluation

- Experiments on real-world social networks
  - In-depth analysis of Xing (about 8 million members)
  - Feasibility studies for Facebook and LinkedIn
  - Checked total of 8 social networks, all vulnerable to attack
- We compared custom / commercial service crawling for group data collection
  - Custom crawler was not hard to implement
  - Commercial: 80legs.com, $0.25/million URLs cheap!
- Controlled and public experiments with volunteers
### Whats Vulnerable Comparison

<table>
<thead>
<tr>
<th></th>
<th>Facebook</th>
<th>MySpace</th>
<th>Friendster</th>
<th>LinkedIn</th>
<th>StudiVZ</th>
<th>Xing</th>
<th>Bigadda</th>
<th>Kiwibox</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uses dynamic links</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Group directory</td>
<td>Full</td>
<td>Searchable</td>
<td>Full</td>
<td>Searchable</td>
<td>Searchable</td>
<td>Searchable</td>
<td>Searchable</td>
<td>Full</td>
</tr>
<tr>
<td>Member directory</td>
<td>Full</td>
<td>Searchable</td>
<td>Full</td>
<td>Full</td>
<td>Searchable</td>
<td>Searchable</td>
<td>Searchable</td>
<td>Searchable</td>
</tr>
<tr>
<td>Group member enumeration</td>
<td>≤6,000</td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>≤500</td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>Unlimited</td>
<td>Unlimited</td>
</tr>
<tr>
<td>Public member profiles</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Vulnerable</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

**Table II: Vulnerability Comparison of Social Networks.**
Case study: Xing

• What is Xing?
  • Business-oriented (similar to US LinkedIn)
  • ~8 Million Users
  • Ghost account (join and leave groups to get data)
  • Closed groups
    – 1,306 join attempts, 108 accepts → 404,331 unique profiles
    – Large groups
Results

• Recovered 4.4 million membership relations, 1.8 million unique group members (of 8 million total)
  - Complete coverage: Attacker has to check 6,277 groups
  - Only 6,277 URLs to check instead of 8 million
• About 42% of users have a unique fingerprint
  - I.e. there is only one user with this configuration of group memberships in the SN
• For 90% of all groups members, the intersection size is below 2,912 users
• Shows that the attack is feasible in real-world settings
  - Leveraging groups: Number of potential victims smaller, but still hundreds of millions!
Controlled Experiment

- Website that implements attack against Xing
  - HTML + Javascript + Ajax for history stealing
  - Feedback form for participants
- 26 volunteers from the authors' Xing contacts
- We could not find any URLs that indicate groups in the browsing history of 11 people
- We successfully de-anonymized 15 / 26 users
  - Group member intersection method worked for 11 users (median size 570 members)
  - Fallback to union set for 4 users (median size 30,013 members, still feasible)
• A tech report of our attack found its way to the news
  - Mainly German language news, Spiegel, Slashdot, ...
• 9,969 volunteers who participated and completed the experiment on our website
• We found group traces for 3,717 users (37.3%)
• 1,207 users claim they were correctly de-anonymized
  - 12.1% of overall participants!
• No reliable information on background of volunteers
  - Still, we think that this shows that the threat is serious
  - Success rate is high, large amount of people de-anonymized
Mitigation!!!!!!

- **Server-side**
  - No more HTTP GET parameters with sensitive data
  - Quick fix: Add non-guessable tokens to sensitive URLs
  - We disclosed our attack to Xing, they invited us, now they use links like `www.xing.com/net/pri523ba6x/tuwien/`
    - Problematic, breaks SEO!

- **Client-side**
  - Disable browsing history, (use safe browsing mode)

- **Browser-side**
  - Same origin policy for style infos, prevent access to style infos on links
  - Upcoming Firefox will fix history stealing (......)
Summary

• We presented a novel attack to de-anonymize website visitors who also use social networks
• Social networks are used to collect the ID data
  - Group feature used to identify victims quickly
• Any website can host the de-anonymization code
  - Find traces of groups and user profiles via history stealing
  - Match these traces against data from the social network
• Consequences are severe
  - Hundreds of millions of potential victims
  - Malicious activities limited only by imagination of attacker
Summary cont.

• Existing anonymity techniques (e.g., onion routing, TOR) are evaded
• The necessary effort for preparing and conducting the attack is relatively low
• High de-anonymization rate in experiments
  - Implemented for Xing
  - Facebook, LinkedIn, MySpace & Co. also vulnerable
  - Can be generalized to other websites that generate sparse datasets (Ebay, Amazon are vulnerable too)
Xing

- http://128.111.48.22/experiment/