

# 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>**

# Attack Overview

*Int. Secure Systems Lab*  
*Vienna University of Technology*

- Imagine you are a social network user
- Just like any user, from time to time you interact with the social network, add friends, join groups, etc.
- Then, (maybe a week later) you browse evil.com
  - evil.com has no connection to the social network
- Unknown to you, evil.com starts an attack against you, and finds out your social network identity
  - i.e. the data you entered in your profile, name, photo, etc.
- evil.com can even look up more sensitive data from the social network and, for example, say “Hello Gilbert Wondracek”

# Attack Overview

*Int. Secure Systems Lab*  
*Vienna University of Technology*

- Our aim: Find out the social network identity of website visitors
  - Instead of tracking browsers (cookies, EFF), we track persons
- We leverage information from social networks
  - Attack limited to social network users (hundreds of millions!)
  - Leaked data from social networks and well-known browser attack allow us to compare and find the ID of users
  - All eight social networks that we examined were vulnerable
- Significant abuse potential
  - Ranging from intrusive advertisements to blackmailing
  - Large number of potential victims

# Attack Details

# Building Block A: History Stealing

Int. Secure Systems Lab  
Vienna University of Technology

- Well-known browser attack
  - Requires only HTML and CSS (Javascript helps, though)
  - CSS allows websites to define style templates (e.g. color, URL for background image) for *visited* / *non-visited* links
- This reveals information about the user's browsing history:
  - Current browsers allow **any** website to ask “Is [URL] in the user's browsing history?” by simply embedding links and comparing the style
  - No exhaustive listing of user's browsing history is possible
    - But no limit on number of asked “questions”
    - Can be done covertly

# History Stealing

Int. Secure Systems Lab  
Vienna University of Technology

- Original (ab)use-case of history stealing
  - Spear phishing (targeted attacks): First find out victim's online banking site, then serve “correct” phishing page
- Browser developers paid little attention
  - Mozilla bug tracking list has entries that are 10 years old
  - Security impact deemed too low for sacrificing style feature?
- Browsing history timeout default values
  - 20 days (IE 8), 90 days (Firefox), *Unlimited* (Chrome)

# Building Block B: Social Network Specifics

*Int. Secure Systems Lab*  
*Vienna University of Technology*

- Web applications have similar structure
  - HTTP GET commonly used for state keeping
    - URLs often contain unique IDs, performed operations, or other sensitive data as parameters:  
`http://sn.com/profile?operation=EditMyProfile&user=12345`
    - We found such links for all social networks that we examined

- Examples from real-world sites:

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]`

# Basic Attack Scenario

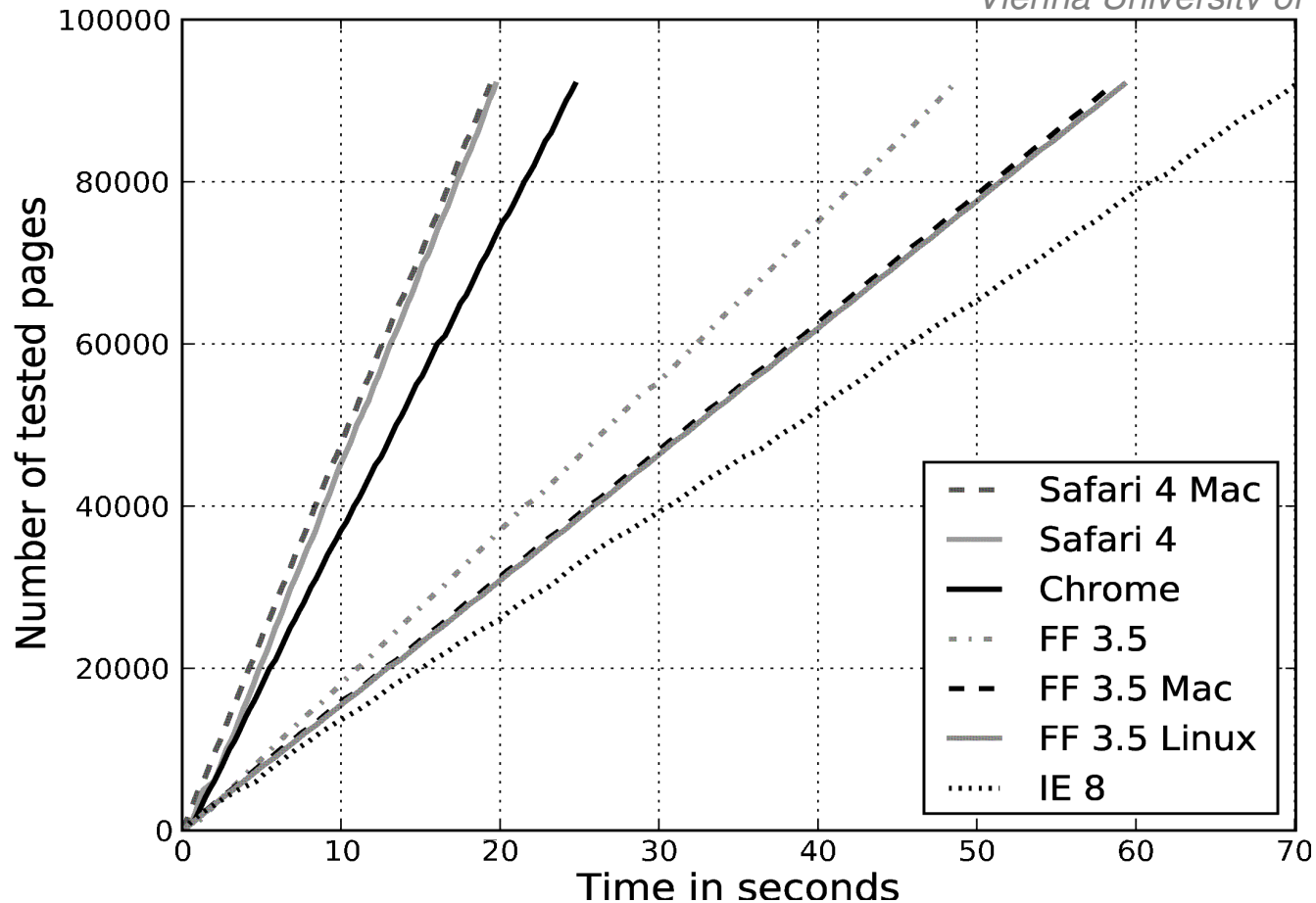


# Basic Attack Scenario

- De-Anonymization attack
  - Combine history stealing and knowledge of SN webapp layout
  - Lure victim to evil.com
  - User ID of the victim can then be found via history stealing
- Attacking website can simply query for (all) user IDs:
  - sn.com/editprofile.html?uid=0
  - sn.com/editprofile.html?uid=1
  - ...
  - sn.com/editprofile.html?uid=[X]
- Look up profile in social network for ID [X]
  - Very unlikely that the URL is in the history if the user is not X

# History Stealing Benchmark

Int. Secure Systems Lab  
Vienna University of Technology



# Not fast enough...

- Social networks have millions of users
  - This also implies millions of URLs that have to be checked via history stealing
- This would take too long for a real-world attack
  - Web surfers might only stay a few seconds on target site
  - Large scale history stealing can get CPU usage to 100%, sluggish UI response is suspicious
- Basic attack would only work for very small social networks
  - Useless?

# Improving the Attack

# Building Block C: Groups

*Int. Secure Systems Lab*  
*Vienna University of Technology*

- Additional hierarchical layer in social networks
  - Subsets of users with similar interests
    - Examples: “Mercedes Drivers”, “IEEE Members”, “Fans of [x]”
  - Groups can be public / closed
    - Public: Anyone can join (immediately)
    - Closed: Admin has to approve new members
- Group features also use specific hyperlinks for interaction
  - Example: [www.sn.com/join\\_group.php?gid=12345](http://www.sn.com/join_group.php?gid=12345)
  - Leaked info → stored in the browsing history again
  - Finding such links in the history is an indicator for membership

# Group Member Enumeration

- How can an attacker get information on group members?
- Social networks typically offer member and/or *group directories*
  - Public lists, so that users can find interesting members / groups
  - Group members can usually list the other members in the same group
- An attacker can use this to collect data on groups
  - 1) Join a group from the directory
  - 2) List all members
  - 3) Leave group
  - 4) Goto step 1
- Eventually, the attacker will know the members of each group

# Group Member Enumeration

- Many SN restricts full listing of (group) members
  - Search features can be abused
    - For example, use US census information to enumerate users, works reasonably well (see paper)
- Attacker can use information from the SN itself to reconstruct membership relations
  - Example: Groups shown in member profiles → Attacker can reconstruct the group directory by crawling the public member directory
  - Example: SN that use systematic (numerical) IDs can be “brute-force crawled”
- At the end of the day, attacker gets info on groups again

# Improved Attack Scenario

*Int. Secure Systems Lab*  
*Vienna University of Technology*

- 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

# Evaluation Overview

*Int. Secure Systems Lab*  
*Vienna University of Technology*

- 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
    - No countermeasures, group information considered non-critical (unlike profiles)
  - Commercial: 80legs.com, \$0.25/million URLs → cheap!
- Controlled and public experiments with volunteers

# Case Study: Xing

Int. Secure Systems Lab  
Vienna University of Technology

- Xing, popular German social network
  - Business-oriented (people use real names, high value target)
  - Similar to LinkedIn in the US
  - About 8 million members, this moderate size allowed us to rely on lab resources for custom crawling
  - We created a user profile and kept on joining / listing / leaving **all** public groups (6,574)
  - Closed groups: We simply asked if we can join
    - 1,306 join attempts, 108 accepted => 404,331 unique members
    - Worked for most large groups (>10<sup>5</sup> members, too hard to maintain?) → important groups for attacker!

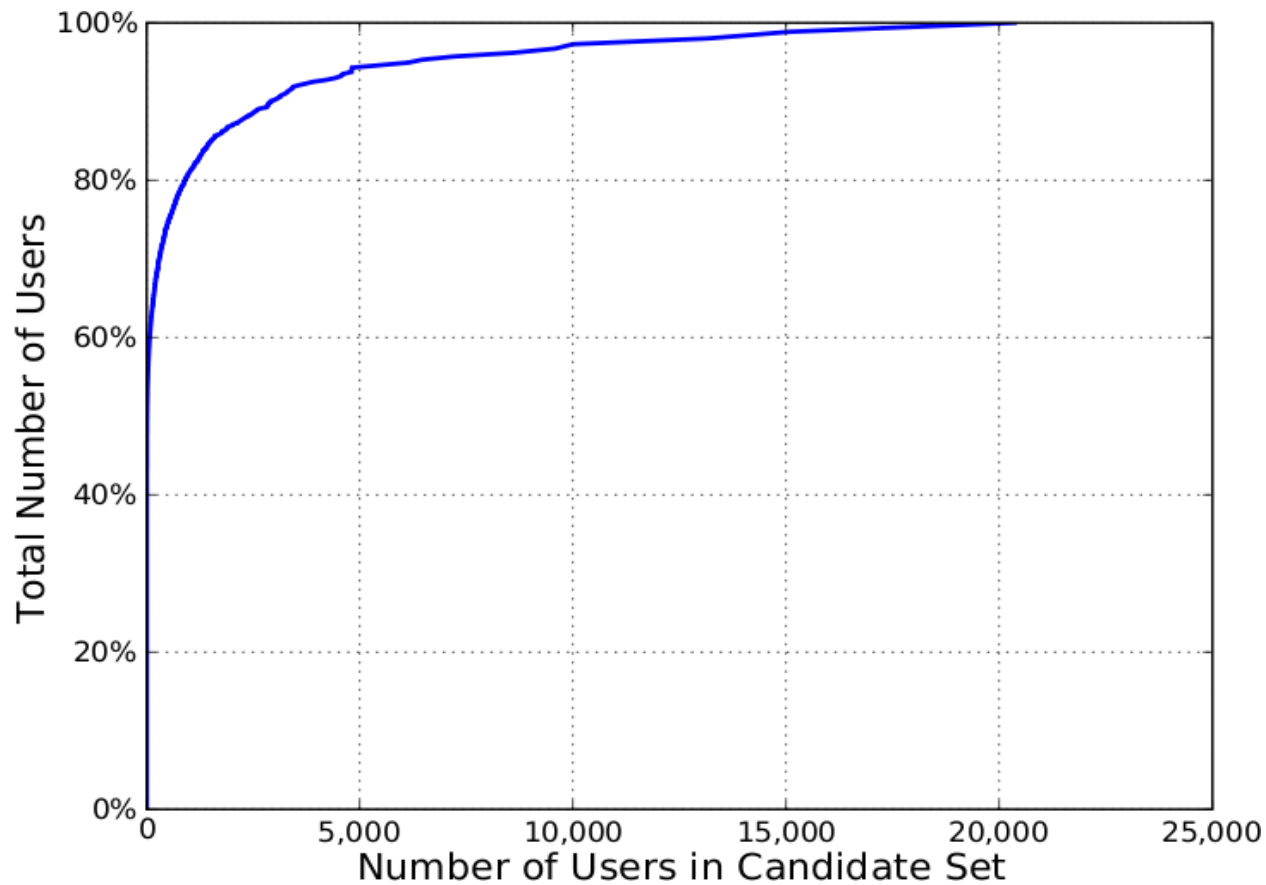
# Xing Analytical Results

Int. Secure Systems Lab  
Vienna University of Technology

- 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!

# Cumulative distribution of candidate set sizes for set intersection

*Int. Secure Systems Lab*  
*Vienna University of Technology*



# Controlled Experiment

*Int. Secure Systems Lab*  
*Vienna University of Technology*

- 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)

# Public Experiment

*Int. Secure Systems Lab*  
*Vienna University of Technology*

- 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



# Mitigation

*Int. Secure Systems Lab*  
*Vienna University of Technology*

- 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/](http://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 (after 10 years of discussion)

# Summary

*Int. Secure Systems Lab*  
*Vienna University of Technology*

- 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

*Int. Secure Systems Lab*  
*Vienna University of Technology*

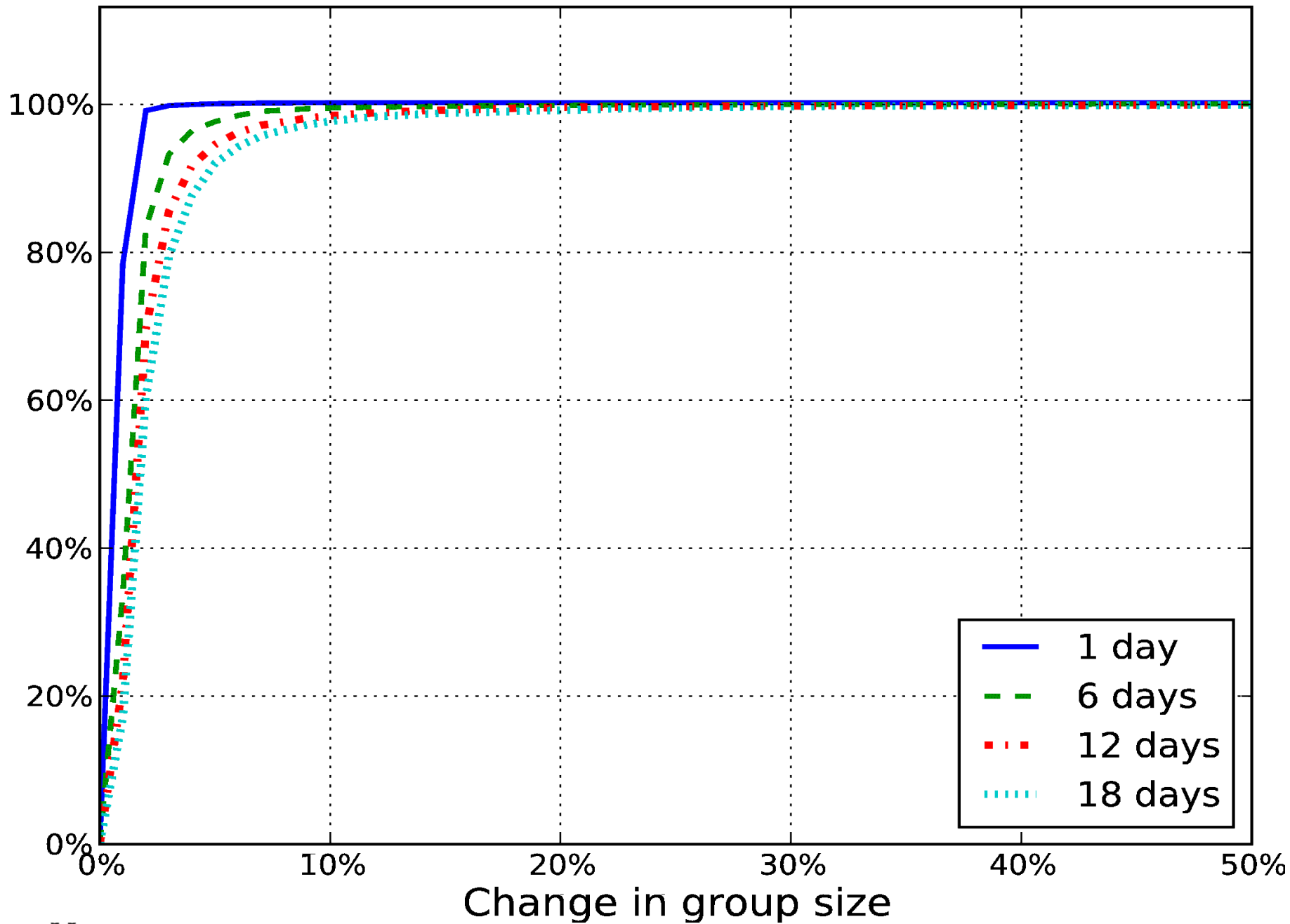
- Existing anonymity techniques (e.g., onion routing, TOR) are evaded
- The necessary effort for preparing and conducting the attack is relatively low
- High de-anonymization rate in experiments
  - Implemented for Xing
  - Facebook, LinkedIn, MySpace & Co. also vulnerable
  - Can be generalized to other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

Thank you!

# Responsible Disclosure

*Int. Secure Systems Lab*  
*Vienna University of Technology*

- We contacted Xing, LinkedIn, and Facebook
- Asked consent of users in experiments
  - Volunteers only, made clear what happens
- Consulted legal department of our university
  - Similar duties like an IRB in US universities



# Feasibility: Facebook / LinkedIn

*Int. Secure Systems Lab*  
*Vienna University of Technology*

- Same data collection principle (join / list / leave)
- Facebook: We stopped our custom crawler after obtaining about **43 million** unique users
  - 3 weeks of non-stop crawling → our machines were never banned / slowed down
- Commercial service
  - Facebook's group directory (public, but huge) was downloaded for \$18.47 → 7.4 million files, 39,156,580 group IDs
  - For other networks (LinkedIn), we used it to brute-force enumerate all active groups (3 million page requests)
- Shows that attack is possible, more details in paper