You Are How You Click

Clickstream Analysis for Sybil Detection

Gang Wang, Tristan Konolige, Christo Wilson†, Xiao Wang‡
Haitao Zheng and Ben Y. Zhao

UC Santa Barbara
†Northeastern University
‡Renren Inc.
Sybils in Online Social Networks

- **Sybil**: fake identities controlled by attackers
  - Friendship is a pre-cursor to other malicious activities
  - Does not include benign fakes (secondary accounts)

- Large Sybil populations*
  - [Facebook](#): 14.3 Million Sybils (August, 2012)
  - [Twitter](#): 20 Million Sybils (April, 2013)

*Numbers from CNN 2012, NYT 2013
Sybil Attack: a Serious Threat

• Social spam
  – Advertisement, malware, phishing

• Steal user information

spies used Facebook to steal Nato chiefs’ details
Taliban uses sexy Facebook profiles to lure troops into giving away military secrets

• Sybil-based political lobbying efforts

Fake Twitter Accounts? Obama’s Political Group Pushes Gun Control
Russian Twitter political protests 'swamped by spam'
Sybil Defense: Cat-and-Mouse Game

Social Networks

- Attackers

Crowdsourcing CAPTCHA solving
  - [USENIX’10]

Realistic profile generation
  - Complete bio info, profile pic
  - [WWW’12]
Graph-based Sybil Detectors

• A key assumption
  – Sybils have difficulty “friending” normal users
  – Sybils form tight-knit communities

• Measuring Sybils in Renren social network [IMC’11]
  – Ground-truth 560K Sybils collected over 3 years
  – Most Sybils befriend real users, integrate into real-user communities
  – Most Sybils don’t befriend other Sybils

Sybils don’t need to form communities!
Sybil Detection Without Graphs

• Sybil detection with static profiles analysis [NDSS’13]
  – Leverage human intuition to detect fake profiles (crowdsourcing)
  – Successful user-study shows it scales well with high accuracy

• Profile-based detection has limitations
  – Some profiles are easy to mimic (e.g. CEO profile)
  – Information can be found online

• A new direction: look at what users do!
  – How users browse/click social network pages
  – Build user behavior models using clickstreams
Clickstreams and User Behaviors

- Clickstream: a list of server-side user-generated events
  - E.g. profile load, link follow, photo browse, friend invite

<table>
<thead>
<tr>
<th>UserID</th>
<th>Event Generated</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>345678</td>
<td>Send Friend Request_23908</td>
<td>1303022295242</td>
</tr>
<tr>
<td>214567</td>
<td>Visit Profile_12344</td>
<td>1300784205886</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

- Intuition: Sybil users act differently from normal users
  - **Goal-oriented**: concentrate on specific actions
  - **Time-limited**: fast event generation (small inter-arrival time)

Analyze ground-truth clickstreams for Sybil detection
Outline

• Motivation

• Clickstream Similarity Graph
  – Ground-truth Dataset
  – Modeling User Clickstreams
  – Generating Behavioral Clusters

• Real-time Sybil Detection
Ground-truth Dataset

• Renren Social Network
  – A large online social network in China (280M+ users)
  – Chinese Facebook

• Ground-truth
  – Ground-truth provided by Renren’s security team
  – 16K users, clickstreams over two months in 2011, 6.8M clicks

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Users</th>
<th>Sessions</th>
<th>Clicks</th>
<th>Date (2011)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sybil</td>
<td>9,994</td>
<td>113,595</td>
<td>1,008,031</td>
<td>Feb.28-Apr.30</td>
</tr>
<tr>
<td>Normal</td>
<td>5,998</td>
<td>467,179</td>
<td>5,856,941</td>
<td>Mar.31-Apr.30</td>
</tr>
</tbody>
</table>

*Our study is IRB approved.*
Normal users use many social network features
Sybils focus on a few actions (e.g. friend invite, browse profiles)

Sybils and normal users have very different click patterns!
Identifying Sybils From Normal Users

• Goal: quantify the differences in user behaviors
  – Measure the similarity between user clickstreams

• Approach: map user’s clickstreams to a similarity graph
  – Clickstreams are nodes
  – Edge-weights indicate the similarity of two clickstreams

• Clusters in the similarity graph capture user behaviors
  – Each cluster represents certain type of click/behavior pattern
  – Hypothesis: Sybils and normal users fall into different clusters
Model Training

① Clickstream Log

② Similarity Graph

③ Behavior Clusters

④ Labeled Clusters

Detection

Unknown User Clickstream

- Good Clusters
- Sybil Cluster
Capturing User Clickstreams

1. **Click Sequence Model**: order of click events
   - e.g. ABCDA ...

2. **Time-based Model**: sequence of inter-arrival time
   - e.g. \{t_1, t_2, t_3, ...\}

3. **Complete Model**: sequence of click events with time
   - e.g. A(t_1)B(t_2)C(t_3)D(t_4)A ...
Clickstream Similarity Functions

• Similarity of sequences
  – Common subsequence
    \[ S_1 = \text{AAB} \]
    \[ S_2 = \text{AAC} \]
    \[ \text{ngram}_1 = \{ A, B, AA, AB, AAB \} \]
    \[ \text{ngram}_2 = \{ A, C, AA, AC, AAC \} \]
    \[ D_{1,2} = \frac{\text{ngram}_1 \cap \text{ngram}_2}{\text{ngram}_1 \cup \text{ngram}_2} \]

  – Common subsequence with counts
    \[ S_1 = \text{AAB} \]
    \[ S_2 = \text{AAC} \]
    \[ \text{ngram}_1 = \{ A(2), B(1), AA(1), AB(1), AAB(1) \} \]
    \[ \text{ngram}_2 = \{ A(2), C(1), AA(1), AC(1), AAC(1) \} \]

• Adding “time” to the sequence
  – Bucketize inter-arrival time, encode time into the sequence
  – Apply the same sequence similarity function

\[ V_1 = (2,1,0,1,0,1,1,0) \]
\[ V_2 = (2,0,1,1,1,0,0,1) \]
Clickstream Clustering

- Similarity graph (fully-connected)
  - **Nodes**: user’s clickstreams
  - **Edges**: weighted by the similarity score of two users’ clickstreams

- Clustering similar clickstreams together
  - Minimum edge weight cut
  - Graph partitioning using METIS

- Perform clustering on ground-truth data
  - Complete model produces very accurate behavior clusters
  - 3% false negatives and 1% false positives

Sybils in normal clusters
Normal users in Sybil clusters
Outline

• Motivation

• Clickstream Similarity Graph

• Real-time Sybil Detection
  – Sybil Detection Using Similarity Graph
  – Unsupervised Approach
Detection in a Nutshell

- **Sybil detection methodology**
  - Assign the unclassified clickstream to the “nearest” cluster
  - If the nearest cluster is a Sybil cluster, then the user is a Sybil

- **Assigning clickstreams to clusters**
  - $K$ nearest neighbor (KNN)
  - Nearest cluster (NC)
  - Nearest cluster with center (NCC)
Detection Evaluation

• Split 12K clickstreams into training and testing datasets
  – Train initial clusters with 3K Sybil + 3K normal users
  – Classify remaining 6K testing clickstreams

<table>
<thead>
<tr>
<th>Detection Algorithm</th>
<th>False Positive</th>
<th>False Negative</th>
<th>Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>K-nearest neighbor</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nearest Cluster</td>
<td></td>
<td></td>
<td>&lt; 0.7%</td>
</tr>
<tr>
<td>Nearest Cluster (center)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

NCC (fastest) is as good as the others
(Semi) unsupervised Approach

- What if we don’t have a big ground-truth dataset?
  - Need a method to label clusters
- Use a (small) set of known-good users to color clusters
  - Adding known users to existing clusters
  - Clusters that contain good users are “good” clusters

- 400 random good users are enough
- For unknown dataset, add good users
- Still achieve high detection accuracy

Known Good Users

Details here
Real-world Experiments

• Deploy system prototypes onto social networks
  – Shipped our prototype code to Renren and LinkedIn
  – All user data remained on-site

• Scanned 40K ground-truth user’s clickstreams
• Flagged 200 previous unknown Sybils

• Scanned 1M user’s clickstreams
• Flagged 22K suspicious users
• Identified a new attack

“Image” Spammers

- Embed spam content in images
- Easy to evade text/URL based detectors
Evasion and Challenges

• In order to evade our system, Sybils may …
  – Slow down their click speed
  – Generate “normal” actions as cover traffic

• Practical challenges
  – How to update behavior clusters over time (incrementally)?
  – How to integrate with other existing detection techniques?
    (e.g. profile, content based detectors)

Force Sybils to mimic normal users

= Win
Thank You!

Questions?