ClickMiner: Towards Reconstructing User-Browser Interactions from Network Traces

Chris Neasbitt
The University of Georgia
cjneasbi@uga.edu
Outline

- Problem
- Goals
- Approach
- Contributions
- System Design
- Challenges
- Evaluation
- Case Study
The modern web is becoming increasingly complex.
- Dynamic Pages
- Scripting Languages
  - e.g. JavaScript
- Browser Plug-ins
- Asynchronous requests

Increasing *Semantic Gap* between network traffic and user actions.
Problem
Problem

“Given the network traffic trace of a browsing session can we determine what interactions with the browser a user made?”

Benefactors

- Forensic Analysis
- Web Usage Miners
User-browser interaction i.e. *click*

- A user interaction that causes the browser to initiate and HTTP request for a new web page.
  - Mouse click on an image with an `onclick` event
  - Touch gesture on a form submit button
  - Pressing Enter while focused a link to follow it
  - Typing a URL into the address bar
  - Clicking on a bookmarked link
Outline

- Problem
- Goals
- Approach
- Contributions
- System Design
- Challenges
- Evaluation
- Case Study
Goals

- Accurately infer \textit{clicks} from full packet network traces.
- Reconstruct the sequence of web pages explicitly requested by the user.
- Infer what page element(s) in a web page was clicked by the user.
Outline

- Problem
- Goals
- Approach
- Contributions
- System Design
- Challenges
- Evaluation
- Case Study
Previous Approach

ReSurf
- Referrer-based click inference (RCI)
- Build *Referrer graph* from traffic
- Prune referrer graph based on heuristics

Referrer Graph
- Node: HTTP request
- Edge: Defines request referrer → request referred relationship
ClickMiner Approach

“Let the browser do the heavy lifting.”
ClickMiner Approach

Network traffic replay within an instrumented browser.

- Through its execution the browser will *consume* traffic.
- Analyze what remains against open pages.

*Click graph* analysis of replay results.
- Utilize referrer information to fill in gaps
Outline

- Problem
- Goals
- Approach
- Contributions
- System Design
- Challenges
- Evaluation
- Case Study
Contributions

- ClickMiner, a novel system dedicated to automatically reconstructing user-browser interactions from full packet captures.
- Evaluate both ClickMiner and RCI in a user study.
- Case study involving a real social engineering-based malware download attack.
Outline

- Problem
- Goals
- Approach
- Contributions
- System Design
- Challenges
- Evaluation
- Case Study
System Design

Click Graph

- Nodes: annotated HTTP Requests \((p, e, q)\)
  - \(p\) = source page for the click
  - \(e\) = element clicked during interaction
  - \(q\) = HTTP request generated

- Edge: \((p_w, e_w, q_w) \rightarrow (p_y, e_y, q_y)\)
  - \(p_y\) reached if as a consequence of \(q_w\)
System Design

ACI (Augmented Click Inference)
- ClickMiner might fail to detect click.
- Leverage the referrer graph
- Fill in the gaps in click paths with partial click nodes
Outline

- Problem
- Goals
- Approach
- Contributions
- System Design
- Challenges
- Evaluation
- Case Study
Challenges

- Missing content
- Request URLs with dynamic content
- JavaScript mediated requests
- HTTPS
Challenges

Missing Content

- Requests with missing response payloads cannot be replayed.
  - Browser Cache
  - Corrupted or Loss Packets
- *Best effort* replay skips these gaps to continue processing what traffic remains.
Challenges
Challenges

Request URLs with dynamic content

- URL parameters containing:
  - Randomly generated values
  - Time-dependent values
  - System-dependent values
- Dynamically generated paths

Replay proxy utilizes an *approximate* matching algorithm for HTTP requests
Challenges

Approximate matching algorithm compares HTTP requests based on:

- Domain name or IP address
- URL path
- URL parameter names
- URL parameter values
- Timestamps

If a match is found it's response is served otherwise respond with 404.
Challenges

JavaScript Mediated Clicks

- DOM elements with JavaScript event handlers

**Network-oriented** approach

- Discover JavaScript mediated elements
- Programmatically activate each one
- If by activation the expected HTTP request is generated then we've found the element
  - Otherwise respond with 204
Challenges

- HTTPS
  - Migration toward ubiquitous use on the web
  - Many enterprise networks already deploy SSL-MITM proxies
    - mitmproxy
    - HoneyProxy
    - Paros
    - Burp
Outline

- Problem
- Goals
- Approach
- Contributions
- System Design
- Challenges
- Evaluation
- Case Study
Evaluation

User Study
- Users performed generic web browsing activities
- Both traffic trace and user interactions were recorded
- 21 Participants, 24 Traces
- 2 Groups
  • Group 1: browser caching disabled
  • Group 2: browser caching enabled with “warmed up” cache
Evaluation

ClickMiner Results Summary

– Avg. between 82% and 90% of clicks reconstructed
– Avg. Between 0.74% and 1.16% false positives
– Greatly outperforms RCI
## Evaluation

<table>
<thead>
<tr>
<th>Trace Number</th>
<th>HTTP Requests</th>
<th>Recorded Clicks</th>
<th>Mined Clicks avg (stddev)</th>
<th>Matching Clicks avg (stddev)</th>
<th>TPR</th>
<th>FPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3925</td>
<td>21</td>
<td>50.80 (0.40)</td>
<td>20.00 (0.00)</td>
<td>95.24%</td>
<td>0.79%</td>
</tr>
<tr>
<td>2</td>
<td>1114</td>
<td>25</td>
<td>39.00 (0.00)</td>
<td>25.00 (0.00)</td>
<td>100.00%</td>
<td>1.29%</td>
</tr>
<tr>
<td>3</td>
<td>2884</td>
<td>16</td>
<td>41.00 (0.00)</td>
<td>13.00 (0.00)</td>
<td>81.25%</td>
<td>0.98%</td>
</tr>
<tr>
<td>4</td>
<td>1030</td>
<td>10</td>
<td>16.00 (0.00)</td>
<td>10.00 (0.00)</td>
<td>100.00%</td>
<td>0.59%</td>
</tr>
<tr>
<td>5</td>
<td>3405</td>
<td>23</td>
<td>46.20 (0.75)</td>
<td>22.80 (0.40)</td>
<td>99.13%</td>
<td>0.69%</td>
</tr>
<tr>
<td>6</td>
<td>3800</td>
<td>21</td>
<td>51.60 (0.80)</td>
<td>19.00 (0.00)</td>
<td>90.48%</td>
<td>0.86%</td>
</tr>
<tr>
<td>7</td>
<td>4891</td>
<td>11</td>
<td>30.20 (0.40)</td>
<td>11.00 (0.00)</td>
<td>100.00%</td>
<td>0.39%</td>
</tr>
<tr>
<td>11</td>
<td>9247</td>
<td>37</td>
<td>75.00 (2.61)</td>
<td>32.20 (0.75)</td>
<td>87.03%</td>
<td>0.46%</td>
</tr>
<tr>
<td>14</td>
<td>6508</td>
<td>32</td>
<td>50.00 (1.10)</td>
<td>28.00 (0.00)</td>
<td>87.50%</td>
<td>0.34%</td>
</tr>
<tr>
<td>16</td>
<td>1167</td>
<td>32</td>
<td>28.60 (0.49)</td>
<td>22.00 (0.00)</td>
<td>68.75%</td>
<td>0.58%</td>
</tr>
<tr>
<td>18</td>
<td>4073</td>
<td>20</td>
<td>76.60 (1.50)</td>
<td>17.20 (0.40)</td>
<td>86.00%</td>
<td>1.47%</td>
</tr>
<tr>
<td>22</td>
<td>5005</td>
<td>23</td>
<td>51.40 (0.80)</td>
<td>21.00 (0.00)</td>
<td>91.30%</td>
<td>0.61%</td>
</tr>
<tr>
<td>23</td>
<td>722</td>
<td>14</td>
<td>15.00 (0.00)</td>
<td>11.00 (0.00)</td>
<td>78.57%</td>
<td>0.56%</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>3674.69</strong></td>
<td><strong>21.92</strong></td>
<td><strong>43.95</strong></td>
<td><strong>19.40</strong></td>
<td><strong>89.63%</strong></td>
<td><strong>0.74%</strong></td>
</tr>
<tr>
<td><strong>Stddev</strong></td>
<td><strong>2350.46</strong></td>
<td><strong>7.88</strong></td>
<td><strong>18.21</strong></td>
<td><strong>6.60</strong></td>
<td><strong>9.58</strong></td>
<td><strong>0.34</strong></td>
</tr>
</tbody>
</table>
## Evaluation

<table>
<thead>
<tr>
<th>Trace Number</th>
<th>HTTP Requests</th>
<th>Recorded Clicks</th>
<th>Mined Clicks avg (stddev)</th>
<th>Matching Clicks avg (stddev)</th>
<th>TPR</th>
<th>FPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>8</td>
<td>4786</td>
<td>28</td>
<td>64.40 (0.80)</td>
<td>21.00 (0.00)</td>
<td>75.00%</td>
<td>0.91%</td>
</tr>
<tr>
<td>9</td>
<td>2212</td>
<td>19</td>
<td>42.80 (1.60)</td>
<td>14.00 (0.00)</td>
<td>73.68%</td>
<td>1.35%</td>
</tr>
<tr>
<td>10</td>
<td>1639</td>
<td>15</td>
<td>23.20 (0.40)</td>
<td>15.00 (0.00)</td>
<td>100.00%</td>
<td>0.50%</td>
</tr>
<tr>
<td>12</td>
<td>1219</td>
<td>10</td>
<td>15.60 (0.49)</td>
<td>7.00 (0.00)</td>
<td>70.00%</td>
<td>0.71%</td>
</tr>
<tr>
<td>13</td>
<td>1250</td>
<td>15</td>
<td>17.00 (0.00)</td>
<td>13.00 (0.00)</td>
<td>86.67%</td>
<td>0.32%</td>
</tr>
<tr>
<td>15</td>
<td>500</td>
<td>34</td>
<td>34.20 (0.40)</td>
<td>28.00 (0.00)</td>
<td>82.35%</td>
<td>1.33%</td>
</tr>
<tr>
<td>17</td>
<td>4682</td>
<td>25</td>
<td>63.00 (0.00)</td>
<td>19.00 (0.00)</td>
<td>76.00%</td>
<td>0.94%</td>
</tr>
<tr>
<td>19</td>
<td>2239</td>
<td>21</td>
<td>38.00 (1.26)</td>
<td>19.20 (0.40)</td>
<td>91.43%</td>
<td>0.85%</td>
</tr>
<tr>
<td>20</td>
<td>3980</td>
<td>21</td>
<td>117.00 (1.26)</td>
<td>19.00 (0.00)</td>
<td>90.48%</td>
<td>2.48%</td>
</tr>
<tr>
<td>21</td>
<td>2312</td>
<td>18</td>
<td>60.60 (0.49)</td>
<td>16.00 (0.00)</td>
<td>88.89%</td>
<td>1.93%</td>
</tr>
<tr>
<td>24</td>
<td>943</td>
<td>22</td>
<td>28.40 (0.49)</td>
<td>14.40 (0.49)</td>
<td>65.45%</td>
<td>1.52%</td>
</tr>
<tr>
<td>Average</td>
<td>2342.00</td>
<td>20.73</td>
<td>45.84</td>
<td>16.87</td>
<td>81.81%</td>
<td>1.16%</td>
</tr>
<tr>
<td>Stddev</td>
<td>1428.86</td>
<td>6.33</td>
<td>28.11</td>
<td>5.10</td>
<td>10.61</td>
<td>0.64%</td>
</tr>
</tbody>
</table>
Evaluation

![Evaluation Graph]

- Average true positive rate vs. Average false positive rate.
- Blue line represents RCI.
- Red dashed line represents ClickMiner.
- Key points:
  - 0.14s.
  - 3.2s.
  - 9.1s.
  - 12.8s.
  - 36.2s.
  - 51.2s.

The graph illustrates the performance comparison between RCI and ClickMiner.
Outline

- Problem
- Goals
- Approach
- Contributions
- System Design
- Challenges
- Evaluation
- Case Study
Malware download incident

- Visited bing.com
- Searched with terms “far cry 3 hackz tools crack”
- Clicked on allhackz[dot]net from search results
- Clicked on “Download” button, opened two pages
  - gameadvert[dot]com
  - wellmediaonline[dot]com
- From wellmediaonline[dot]com download started via script from effortlessdownload[dot]com
Case Study
Case Study

Network
Security
Intelligence