Leveraging the Crowds to Disrupt Phishing

Jason Gustafson and Jun Li

October 14, 2013
Phishing As a Persistent Problem

- Many progresses in anti-phishing have been made
- But not always effective
  - Taking down a phishing site takes time
  - Blacklists can be obsolete
  - New tools are only useful if users install them
  - Warnings are only meaningful if users heed them
  - Phishers are getting more smart
- The status quo: Phishers continue to find new victims!
From Preventative to Proactive

- A new approach from a different perspective is to become more aggressive.
- Rather than preventing users from being trapped, focus on the phishers.
- We look at how to disrupt phishing activities.
Our Previous Approach: Humboldt 1.0

- Injects large amount of fake credentials into phishing sites
  - honey tokens
- Any usage of honey tokens will expose phishers (or their customers from the black market)
- Deploys a distributed network of honey token submitters
  - Submissions cannot all come from a small number of IPs
Limitations of Humboldt 1.0

- Depended on an *automated* submission procedure
  - Need to profile the phishing sites and then inject credentials accordingly
- Phisher can make the underlying structure of a phishing site more complex
  - Thus foiling automatic profiling of a phishing site
- Or use CAPTCHA!
Humboldt 2.0
Basic Idea

- Humboldt 2.0 leverages actual people to submit honey tokens
- The phishing page must remain usable by people and must accept their submissions
  - otherwise there is no point in phishing!
- We evaluate the feasibility of this idea in this work
Architecture

- **Central server**: coordinate assignments and submission of honey tokens
- **marketplace**: distribute honey token submission tasks to people
- **exit nodes**: last hop in each submission
- **phishing feeds**: external sources for discovering new phishing sites
Advantages

- Reasonable assurance on the submission
  - Every submission will go through the Humboldt server
- Distributed submission via exit nodes
  - Each with a different IP address
- Exit nodes are cheap, and Humboldt can have a large number of them
Arms Race with the Phisher: Is Humboldt 2.0 Resilient?
Threat Model

- Phishers know about the existence of Humboldt and how it works
- Some human works and exit nodes can be malicious
- Phishers can collect statistics of their visitors
- Phishers can collaborate
Active Tactics

- DDoS the Humboldt server
  - Covered extensively in the literature
- Hire bots to do submission
  - CAPTCHA
- Enlist malicious human workers and/or exit nodes
  - Cannot affect the submission of benign workers and exit nodes
  - Humboldt can tight the recruiting and monitoring of its human workers
Passive Tactics

- Analysis of submitted data
  - E.g., legitimate credentials? IP address local if the target victim is a local bank?

- Indirect verification
  - E.g., email address used as username valid?

- Source heuristic
  - Filter submissions from IP addresses with high submission rates
  - Refer to paper for more theoretical analysis
Effectiveness of Humboldt 2.0
Metrics

- How many honey tokens should Humboldt submit?
  - Thus how many exit nodes to use?
- How many real victims can Humboldt save?
- What is the delay for a human worker to respond to a task?
- What is the reliability of human works?
- What is the effective cost per successful submission?
Number of Honey Tokens

- Totally \( n \) submissions, \( h \) from Humboldt, \( r \) (i.e. \( n-h \)) from real victims
- The phisher uses \( k \) out of \( n \), with \( X \) honey tokens

\[
P(X \geq 1) = 1 - \frac{\binom{n-h}{k}}{\binom{n}{k}}.
\]

- If \( n=100, \ k=10 \), we need 20 honey tokens for \( P \geq 0.9 \).
Number of Real Victims Unprotected

- Assume after \( l \) transactions using honey tokens, we can stop the phisher
  - note knowing a transaction is from phisher does not necessarily stop him from the next transaction
- Denote \( V \) unprotected victims targeted by phisher before that

\[
E(V) = \frac{l \times r}{h + 1}.
\]
**Experimenting Humboldt w/ Amazon Mechanical Turks**

<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total HITs</td>
<td>4643</td>
</tr>
<tr>
<td>Submitted HITs</td>
<td>3829</td>
</tr>
<tr>
<td>Expired HITs</td>
<td>814</td>
</tr>
<tr>
<td>Total Worker Cost</td>
<td>181.42</td>
</tr>
<tr>
<td>Total Amazon Commission</td>
<td>18.14</td>
</tr>
<tr>
<td>Avg. Cost per HIT</td>
<td>0.052</td>
</tr>
<tr>
<td>Unique Workers</td>
<td>213</td>
</tr>
<tr>
<td>Avg. HITs per Worker</td>
<td>17.82</td>
</tr>
</tbody>
</table>

*Jun Li <lijun@uoregon.edu>*
Human Worker Incentives

- Higher price leads to more completed HITs
- But does it lead to a higher quality?
Human Worker Delay

- Delay is from time of HIT creation to the time of token submission.
- X marks incorrect submissions.
- Better payment does not lead to a noticeable difference.
Human Worker Reliability

- Reliability = correctly submitted HITs / total number of accepted HITs.
- Workers with more HITs or better pay are not necessarily more reliable.
Effective Cost

- Net price paid per successful submission: \( C/(1\text{-fail rate}) \)
- We consider the effect of source heuristic
- Details in the paper
Conclusions

- Anti-phishing has mostly been preventative, but the defense could be more proactive
- Via Humboldt 2.0, we demonstrated how we may leverage human workers to inject honey tokens to phishing sites and disrupt phishing
- We studied the resiliency and effectiveness of such an approach
The End
Leveraging the Crowds to Disrupt Phishing

Jason Gustafson and Jun Li