SAuth: Protecting User Accounts from Password Database Leaks

Georgios Kontaxis‡, Elias Athanasopoulos‡
Georgios Portokalidis*, Angelos Keromytis‡

‡Columbia University
*Stevens Institute of Technology
Authentication recognizes passwords not users ...
... and unfortunately passwords get leaked
With stolen password, Attacker impersonates Alice

1a) I'm Alice on Evernote

1b) Welcome to Alice's account
Password leaks happen all the time

- May go unnoticed until it’s too late

<table>
<thead>
<tr>
<th>Year</th>
<th>Company</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>2009</td>
<td>RockYou Gaming</td>
<td>32.0 million</td>
</tr>
<tr>
<td>2010</td>
<td>Gawker Media</td>
<td>1.5 million</td>
</tr>
<tr>
<td></td>
<td><em>Domino attack prompted resets in other sites</em></td>
<td></td>
</tr>
<tr>
<td>2011</td>
<td>Sony</td>
<td>1.0 million</td>
</tr>
<tr>
<td>2012</td>
<td>LinkedIn</td>
<td>6.5 million</td>
</tr>
<tr>
<td>2013</td>
<td>Twitter</td>
<td>250.000</td>
</tr>
<tr>
<td></td>
<td><em>Before being detected and shut down</em></td>
<td></td>
</tr>
<tr>
<td>2013</td>
<td>Adobe</td>
<td>150.0 million</td>
</tr>
</tbody>
</table>
Passwords get cracked all the time

• Weak passwords
  – short, dictionary words, names, patterns, etc.

• Fast hardware
  – Commodity parallel architectures (GPUs)
  – Cloud-powered cracking platforms
    • 6 days after the 6.5 million LinkedIn password leak, 90% of them were cracked
Enhanced Authentication Today

- Two-Factor Authentication
  - How many tokens/app can a user handle?

- Single sign-on services
  - Single point of failure
  - Relying party gets to find out user identity*
  - Privacy issues from coarse-grained data sharing
How about Authentication Synergy?

- Forgot your password?
How about Authentication Synergy?

- User’s Authentication State
SAuth: Synergy-based Enhanced Authentication

• *We propose: cooperating sites pool authentication resources*
SAuth: Synergy-based Enhanced Authentication

- We propose: cooperating sites pool authentication resources
SAuth: Synergy-based Enhanced Authentication

- We propose: cooperating sites pool authentication resources
SAuth: Synergy-based Enhanced Authentication

- We propose: cooperating sites pool authentication resources
SAuth: Synergy-based Enhanced Authentication

- We propose: cooperating sites pool authentication resources
SAuth: Synergy-based Enhanced Authentication

- **We propose:** cooperating sites pool authentication resources
SAuth: Synergy-based Enhanced Authentication

- **We propose:** cooperating sites pool authentication resources
SAuth: Synergy-based Enhanced Authentication

- Password leak on Evernote will protect account access
SAuth: Synergy-based Enhanced Authentication

- Attacker has compromised Alice’s password on Evernote
SAuth: Synergy-based Enhanced Authentication

- Attacker impersonates Alice on Evernote

User Agent \[\rightarrow\] Alice on Evernote, (1a) Twitter will vouch \[\rightarrow\] Evernote

(1b) passwords match, Twitter should vouch

Attacker
SAuth: Synergy-based Enhanced Authentication

- Attacker is unable to produce Alice’s Twitter password
SAuth: Synergy-based Enhanced Authentication

- Authentication process fails, Evernote denies access
Password Reuse Woes

Stolen passwords re-used to attack Best Buy accounts

**Summary:** Customer re-use of the same user name and password across multiple sites is being blamed for attacks on customer accounts at BestBuy.com.

- User has 7 passwords, re-uses 5 of them
- Password shared across 6 sites [Florenicio WWW ’07]
Decoy Passwords

- Uncertainty about the actual password
- Store N-1 decoy passwords along
- Attack reduced to online guessing
- All decoys are valid passwords, server does not know the difference

| Username | P[0] | P[1] | P[…] | P[N] |

- How many decoys?
  - 16,384 for NIST L2 security when password is reused
Realistic Decoy Passwords

- User password must blend-in with the decoys
  - Crackers are already factoring in human behavior
  - Complex vs Popular Passwords

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>string-digit</td>
<td>37%</td>
</tr>
<tr>
<td>digit-string</td>
<td>05%</td>
</tr>
<tr>
<td>!</td>
<td>10%</td>
</tr>
<tr>
<td>$</td>
<td>03%</td>
</tr>
</tbody>
</table>

- Ideal: have the user type N passwords, remember 1
- Practical: generation within the password ecosystem
  - Any blind automated method will generate outliers
  - Probabilistic production seeded by user’s password, biased towards structures of similar popularity and semantics
Summary

• Authentication Synergy results in leak-resistant password authentication
  – Complements existing security
  – Respect for user privacy, verifiable site cooperation
  – Minimal changes server-side, no changes client-side

• Decoys mitigate password reuse habits
  – Generated off the user password, consider its context and general human password habits
Intentionally left blank
Unintentionally left blank
Honeywords, Kamouflage and SAuth Decoy Passwords

• **Honeywords**
  – Does not yet consider human password habits
  – Honeywords are not valid passwords
  – Use of any honeyword will raise an alarm
  – Auxiliary honeychecking server

• **Kamouflage password manager**
  – Considers human password habits
  – Master password decoys are all valid
  – Online guessing attack should raise alarm

• **SAuth Decoy Passwords**
  – Considers human password habits
  – Decoy passwords are all valid
  – Online guessing attack should raise alarm