Tangled Web of Password Reuse

Anupam Das (UIUC),
Joseph Bonneau (Princeton University),
Matthew Caesar (UIUC),
Nikita Borisov (UIUC),
XiaoFeng Wang (Indiana University at Bloomington)
Audience Poll

- How many of you use the web?
- How many of you use web sites with accounts/passwords?
- How many of you use different passwords on all the sites you use?
The Problem

However, on average a user has 6-7 passwords! [Florencio et al. WWW’07]

On average a user maintains 25 distinct online accounts! [Florencio et al. WWW’07]

So passwords are reused!

Social Networks
Emails
Online Banking
Online Shopping

Online
Entertainment
Online
Storage
Emails
Online
Banking
Online
Shopping
What’s wrong with reuse?

Some High Profile Leaks:
- Linkedin: 6.5 million
- Yahoo Voice mail: 450,000
- Twitter: 56,000
- RockYou: 32 million
- Adobe: 150 million

Enables cross-site password guessing.
What if passwords are not same?

Say modify passwords based on web site policy.

- Use **leaked password** to guess unknown password
- Potentially **reduce the search space**.
Our work

We study the problem of password reuse across different web sites-

• Characterize extent of the problem
  o Conduct measurement and user study

• Characterize severity of the problem
  o Develop and evaluate cross-site guessing algorithm
Challenges and approaches

- How you obtain data to evaluate on?
  - User study, collect publicly leaked passwords

- How do we analyze data?
  - Derive typical “transformations”

- What’s a good algorithm to guess passwords?
  - Parameterize and derive ordering of transformations that minimizes the number of guesses
Getting data with a user study

To gain insight into users’ behavior and thought processes when creating passwords for different websites, we conducted an anonymous survey. We had a total of 224 participants.

- 51% Reuse a password already used for a different website
- 26% Modify an existing password
- 22% Create an entirely new password
- 1% Prefer not to answer

Other techniques:
- Adding a number at the end or front: 20%
- Capitalization of letters: 17%
- Adding a symbol at the end or front: 10%
- Leet transform: 8%
- Substring movement: 6%
- Using website specific information: 4%
- Prefer not to answer: 3%
- Reversing: 3%
- Use adjacent keys on the keyboard: 2%
- Email address: 1%
Getting more data from leaked passwords

Publicly leaked email and password pairs from 11 different web sites:

<table>
<thead>
<tr>
<th>Site</th>
<th>#</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>csdn.net</td>
<td>6428630</td>
<td>2011</td>
</tr>
<tr>
<td>gawker.com</td>
<td>748559</td>
<td>2010</td>
</tr>
<tr>
<td>voices.yahoo.com</td>
<td>442837</td>
<td>2012</td>
</tr>
<tr>
<td>militarysingles.com</td>
<td>163482</td>
<td>2012</td>
</tr>
<tr>
<td>rootkit.com</td>
<td>81450</td>
<td>2011</td>
</tr>
<tr>
<td>myspace.com</td>
<td>49711</td>
<td>2006</td>
</tr>
<tr>
<td>porn.com</td>
<td>25934</td>
<td>2011</td>
</tr>
<tr>
<td>hotmail.com</td>
<td>8504</td>
<td>2009</td>
</tr>
<tr>
<td>facebook.com</td>
<td>8183</td>
<td>2011</td>
</tr>
<tr>
<td>youporn.com</td>
<td>5388</td>
<td>2012</td>
</tr>
</tbody>
</table>

Total 6077 unique users

<table>
<thead>
<tr>
<th>Passwords Per user</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>97.75%</td>
</tr>
<tr>
<td>3</td>
<td>1.82%</td>
</tr>
<tr>
<td>4</td>
<td>0.26%</td>
</tr>
<tr>
<td>5</td>
<td>0.15%</td>
</tr>
<tr>
<td>6</td>
<td>0.02%</td>
</tr>
</tbody>
</table>
## Snapshot of leaked passwords

<table>
<thead>
<tr>
<th>Email ID</th>
<th>Passwords</th>
<th>Passwords</th>
</tr>
</thead>
<tbody>
<tr>
<td>116</td>
<td>iloveyou</td>
<td>iloveyou</td>
</tr>
<tr>
<td>117</td>
<td>loving</td>
<td>loving1</td>
</tr>
<tr>
<td>118</td>
<td>naughty</td>
<td>NAUGHTY</td>
</tr>
<tr>
<td>119</td>
<td>password</td>
<td>pa55wOrd</td>
</tr>
<tr>
<td>120</td>
<td>logout0616</td>
<td>logout</td>
</tr>
<tr>
<td>121</td>
<td>butcher05</td>
<td>Butcher05</td>
</tr>
<tr>
<td>122</td>
<td>joey1992</td>
<td>joey92</td>
</tr>
<tr>
<td>123</td>
<td>123456</td>
<td>12345678</td>
</tr>
<tr>
<td>124</td>
<td>gzwz0204</td>
<td>0204gzwz</td>
</tr>
<tr>
<td>125</td>
<td>mike04</td>
<td>jade1979</td>
</tr>
<tr>
<td>126</td>
<td>lucky777</td>
<td>lucky7</td>
</tr>
</tbody>
</table>

- Identical
- Substring
- Others (more complex transformations)
Categorizing passwords

We categorize the password pairs to 3 different categories:

- **Identical**: 43%
- **Substring**: 19%
- **Others**: 38%

We focus on trying to guess these passwords.
Characterizing similarity of passwords

To get a better understanding of the similarity of the non-identical passwords, we look at different similarity metrics.

**Distance-like functions**
- Manhattan, Cosine

**Edit distance-like functions**
- Levenshtein, Damerau Levenshtein

**Token-based distance functions**
- Dice, Overlap

**Alignment-like functions**
- Smith-Waterman, Needleman-Wunsch, LCS
Deriving transformation operations

Let's start with passwords that are substring of each other as they require only insertions or deletions.

### Insertion/Deletion Location

<table>
<thead>
<tr>
<th>Location</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>At start</td>
<td>10%</td>
</tr>
<tr>
<td>At end</td>
<td><strong>88%</strong></td>
</tr>
<tr>
<td>At both ends</td>
<td>2%</td>
</tr>
</tbody>
</table>

Similar to what we found in our user study.

Most insertions/deletions are of length= 2
Deriving more complex transformations

- Sequential key:
  qwerasdf → 1234qwer
- Sequential alternative key:
  12345 → !@#$%
- Sequential alphabet:
  abcde → 12345
- Capitalization:
  naughty → NAUGHTY
- Reverse:
  123456 → 654321
- LeetSpeak Transformation:
  password → pa$$w0rd
- Substring Movement:
  gzwz0204 → 0204gzwz

Other less common transformations also exist
- repeating character sequences
- swap of positions
- using email address
- Etc.
Automating password guessing

The question to ask:
Given a leaked password (as seed), can we design an algorithm to automatically guess other passwords?

Answer:
Yes, we can!

Goals:
1. Compute the orderings and parameterizations that require minimum number of guesses
   • We obtain the ordering using 40% of our data
2. Make the design applicable for online attack scenario.
Number of guesses

We were able to guess **75%** of the passwords in the ‘Substring’ category within 100 attempts.

- Able to guess **~30%** passwords within 100 attempts.
- Our approach is therefore more suitable for online attack scenarios.
Similarity of guessed passwords

- Majority of the correctly guessed passwords had high similarity score.
- Majority of the non-cracked passwords had small similarity score.
Conclusion

- Password reuse is common
  - We found 43-51% of users reused their passwords
- Password reuse is harmful
  - Makes cross-site guessing easier.
  - We were able to guess 30% of the non-identical passwords within 100 attempts.

Even a “low-value” website compromise can be serious. A hack of your Zynga (Farmville) account can potentially compromise your Gmail account!

Details about the project is available at:
http://web.engr.illinois.edu/~das17/passwordreuse.html
Related Works

- Guess again (and again and again) [IEEE S&P 2012]
  - Perform comparative strength of different composition policy

  - Uses Probabilistic Context Free Grammar to generate new word mingling rules.

- Adaptive Password Strength Meters from Markov Models [NDSS 2012]
  - Uses Markov models to guess passwords

- Password Strength: An Empirical Analysis [InfoCom 2010]
  - Compare PCFG, Markov model, Dictionary attack

- Security of modern password expiration [CCS 2010]
  - Offline 41% and online 17% (5 guesses) password cracked.

- How does your Password Measure Up? [USENIX 2012]
  - Studies user behavior for different password meter