Means of Authentication (1/2)

- **Something you know**
  - Password or PIN

- **Something you have**
  - Smart card
  - Private key (of a public-private key pair)
  - Phone (running particular software)

- **Something you are**
  - Biometrics (e.g., iris or fingerprint)
Means of Authentication (2/2)

- Somewhere you are
  - Location-limited channels
- Someone you know (social authentication)
  - Someone vouches for you
  - You can identify people you should know
- Some system vouches for you
  - Single sign-on
  - PKI Certificate Authorities
Password Advantages

- Familiar to people
- You can have many different ones
- Nothing to carry
- Easy to revoke / replace
- Easy to deploy
- Low cost
- Doesn’t require a trusted third party
- Not linked to an individual*
Disadvantages of Passwords

- Predictability
- Interference between multiple passwords
  - Limits of human memory
  - Password reuse or “trivial” modification
- Requiring a large portfolio of passwords
- Easy to deploy incorrectly / naively
  - System administrators (store in plaintext?)
  - Users
What about Biometrics?

- Fingerprint
- Retina scans
- Face recognition
- Finger/hand geometry
- Voice or speech recognition
- (Many others)
Practical Challenges for Biometrics

- You cannot change them or create a new one (fingerprint?)
- Potentially sensitive data
- High equipment costs
- Sensitive to changes in the environment
- Biometrics can change over time
- Easy to forge?
Threats to Password Security

- Online attack against live system
  - Rate limit?
Threats to Password Security

- Online attack against live system
  - Rate limit?
- Attack against password-protected files
- Offline attack against stolen database
How to Store Passwords at the Server End?
Passwords, Hashes, Salt

- **Password database**
  - Not a good idea to store plaintext directly

- **Login: don’t directly match the plaintext of password**
  - Hash (Password + salt) $\rightarrow$ Password hash
  - Plaintext password is stored in other places
  - Password hash and salt is used to authenticate users

![Diagram showing password flow]

- User inputs: Username, Salt
- Server computes: Username, Salt $\rightarrow$ Password hash
- User computes: Plaintext password + Salt $\rightarrow$ Password hash
- Server and User compare Password hash

11 VIRGINIA TECH.
Passwords, Hashes, Salt (Con’t)

- Why hashing
  - If database is compromised, not directly reveal the plaintext password
  - Usually, hashed and plaintext passwords are stored in different servers

- Why Salting
  - Add a random string added to each password
  - Make it much difficult brute force guessing the plaintext password
  - Prevents the use of rainbow tables
Anatomy of an Offline Attack

- Attacker compromises database
- Attacker makes and hashes guesses
- Finds match → try on other sites
  - Password reuse is a key problem
How strong is a particular password?
By looking at them?

- iloveyou
- n(c$JZX!2dfa^dafdIAX^N
- j@mesb0nd007

Leet transformation
How to *Measure* password strength?

- Number of characters, types of characters
- Shannon entropy
- John the Ripper (password cracking software)

Which one is better?
Old metric: Entropy

- Calculated based on input symbol size (many)
  - Doesn’t account for human patterns
- NIST back-of-envelope estimate (NIST 2006)
  - Vague, not empirical
- Estimated Shannon entropy (Shay 2010)
  - Requires big sample sizes, underestimates
- Average, doesn’t tell you about your weak links
Better Way: Guessability

- How many guesses to reach password?
  - Subject to guessing algorithm, training data
  - Calculate quickly via lookup algorithm

- Result: guess number or beyond cutoff
  - Model real attacker
  - Per-password estimates

### Example:

<table>
<thead>
<tr>
<th>Password</th>
<th>Guess number</th>
</tr>
</thead>
<tbody>
<tr>
<td>12345678</td>
<td>4</td>
</tr>
<tr>
<td>Password178</td>
<td>$1.4 \times 10^6$</td>
</tr>
<tr>
<td>jn%fKXs1!8@Df</td>
<td>Beyond cutoff</td>
</tr>
</tbody>
</table>
Perception vs. Reality
Evaluating Password Pairs

- iloveyou88
- ieatkale88
  - 4,000,000,000 x more secure!
- brooklyn16
- brooklynqy
  - 300,000 x more secure!
Ways People Were Wrong

- Overstated security benefits of:
  - Digits
  - Character substitutions (e.g., a @)
  - Keyboard patterns (e.g., 1qaz2wsx3edc)
- Did not recognize common words/phrases
Many Ways People Were Right

- Capitalize letters other than the first
- Put digits and symbols in middle, not end
- Use symbols rather than digits
- Avoid:
  - Common first names
  - Words related to account
  - Years and sequences
Many Different Ways to Guess Passwords

- Guessing attacks are data-driven
  - Previously stolen passwords
  - Natural-language corpora

- Array of tools
  - Cracking software
  - Academic algorithms
Markov Models

- Predicts future characters from previous
- Approach requires weighted data:
  - Passwords
  - Dictionaries
Markov Models: Basic Idea
Markov Models: Basic Idea
We also wrote a paper recently...

[CODASPY 2018]

- Password datasets
  - 28.8 million users, 61.5 million passwords
  - 107 online services over 8 years
  - LinkedIn, Myspace, Gmail, etc.

- How do people reuse or modify their passwords across different services?
Password Reuse & Modification across Services

Existing efforts provide an limited understanding

- User study: self-reported result may differ from real-world behavior
  - [SOUPS’16] [SOUPS’15] [SOUPS’14] [CODASPY’13] [CHI’11]

- Existing empirical studies are limited in scale
  - [CCS’17]: 154 users
  - [SOUPS’16]: 134 users
  - [NDSS’14]: 6077 users

A lack of empirical understandings at a large-scale

28,000,000+ Users
**Bridging the Gap: Large-scale Empirical Analysis**

- Collect massive password datasets with email addresses
  - Link the same users’ passwords across services
- Data collection method
  - Searched through online forums, data archives, darknet markets (2017)
  - Obtained 107 public password datasets leaked during 2008-2016

---

**Raw Dataset**
- 107 Services
- 428 Million Users
- 497 Million PWs

**Reverse Hashed PWs**

**Plaintext Passwords**
- 460 Million
- Plaintext PWs (93%)

**Users w/ >=2 PWs**

**Final Dataset**
- 107 Services
- 28 Million Users
- 61 Million PWs

---

IRB Approved!
## Diverse Categories of Online Services

<table>
<thead>
<tr>
<th>Category</th>
<th>#Plain PWs</th>
<th># Datasets</th>
<th>Top 3 Largest Datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Social</td>
<td>286,000,000</td>
<td>7</td>
<td>Myspace, VK.com, LinkedIn</td>
</tr>
<tr>
<td>Adult</td>
<td>75,200,000</td>
<td>9</td>
<td>Zoosk, Mate1, YouPorn</td>
</tr>
<tr>
<td>Game</td>
<td>40,800,000</td>
<td>13</td>
<td>Neopets, 7k7k, Lbsg</td>
</tr>
<tr>
<td>Entertainment</td>
<td>30,700,000</td>
<td>4</td>
<td>Lastfm, Swingbrasileiro, LATimes</td>
</tr>
<tr>
<td>Internet</td>
<td>16,400,000</td>
<td>18</td>
<td>000webhost, Comcast, Yahoo</td>
</tr>
<tr>
<td>Email</td>
<td>9,600,000</td>
<td>3</td>
<td>Gmail, Mail.ru, Yandex</td>
</tr>
<tr>
<td>Forum</td>
<td>1,100,000</td>
<td>25</td>
<td>CrackingForum, Abusewith.us, Gawker</td>
</tr>
<tr>
<td>Shopping</td>
<td>340,000</td>
<td>12</td>
<td>RedBox, 1394store, Myaribags</td>
</tr>
<tr>
<td>Others</td>
<td>210,000</td>
<td>7</td>
<td>Data1, Data2, Data3</td>
</tr>
<tr>
<td>Business</td>
<td>10,000</td>
<td>9</td>
<td>Movatiathletic, Hrsupporten, 99Fame</td>
</tr>
</tbody>
</table>
Research Questions

- How often do users reuse or modify passwords across services?
- How long does it take for users to update their reused passwords after data breaches?
- How guessable are the modified passwords?
37 million password pairs from the same users: given a pair of passwords, determine “reused”, “modified”, or “unknown”
Highlights of Findings

- 53% of the 28.8 million users reused/modified passwords
  - 38% users once reused the same password
  - 21% users once modified an existing password for a new service

- Sensitive services received most reused/modified passwords
  - Ratio = (# reused+modified pws) / (# pws of a service category)
  - Shopping services have the highest ratio (85%)
  - Email services are at the second place (62%)
Delays of Modifying Passwords After Data Breach

Longest time span between any pair of reused/modified passwords

CDF of Users (%)

Reused
Modified

70% users still reuse passwords leaked 1 year ago
40% users still reuse passwords leaked 3 years ago
Password Guessing

**Observation:** password modification patterns have a low variance

- Given a user’s leaked PW → guess modified PW of un-breached services
- Possible for online guessing

**Training-based guessing schemes**

- Learn the different rules of transform one password
- Given a password, learn the optimized order
- Bayesian inference model
Password Guessing Results

0.1% training data, guess the rest of 99.9% modified passwords

30% of modified PWs guessed in 10 attempts (4.2M out of 14.4M modified PWs)

12.2M reused → 1 guess

Low variance confirmed
Modified passwords highly predicable