14. Authentication
Part 2
How Do We Help Users Make Better Passwords?
Password-Composition Rules

• Goal: Increase the password space
• In practice, many users comply in predictable ways
Password Expiration

- Goal: Make sure stolen passwords are invalid by the time the attacker cracks them
- Require password change every X days? (No!)
Problem 1: Bad Advice

### Password Requirements

**Must Contain**
- At least 8 characters.
- At least one uppercase alphabetic character (e.g., A-Z).
- At least one lowercase alphabetic character (e.g., a-z).
- At least one number (e.g., 0-9).
- At least one special character (e.g., []~!@#$%^&*()?<>./_-=).

**Cannot Contain**
- Known information (i.e., first name, last name, Andrew userID, date of birth, 9-digit Carnegie Mellon ID number, SSN, job title).
- Four or more occurrences of the same character (e.g., aaaa, 2222, a123a345a678a).*
- A word that is found in a standard *dictionary.*
  (after removing non-alpha characters).

*This requirement does not apply to Andrew account passwords that are more than 19 characters in length (e.g., passphrase).*

### Additional Policies
- Last five passwords cannot be used.
- Cannot be changed more than four times in a day.
Problem 2: Inaccurate Feedback

Password1!
Problem 3: Unhelpful Feedback

Please enter a stronger password.
Proactive Strength Checking

- Initial idea: provide feedback
- In practice: complexities regarding what to model, and how to do so efficiently
User-Centered Security
Some Ways to Understand Users

- Retrospective analysis of password breaches
- Large-scale online studies
- Examine real passwords with permission
- Qualitative studies
Meters’ Security & Usability Impact

Meters Are Ubiquitous
Test Meters’ Impact

• How do meters impact password security?
• How do meters impact usability?
  – Memorability
  – User sentiment
  – Timing
• What meter features matter?
• 2,931-participant online study
A strong password helps prevent unauthorized access to your email account.

Type new password: ****

8-character minimum; case sensitive

Password strength: Bad. Consider adding an uppercase letter or making your password longer.

Retype new password: 

- Make my password expire every 72 days.

Save
Visual Differences

Type new password: [user1X]
8-character minimum; case sensitive

Baseline meter: Fair. Consider adding a digit or making your password longer.

Three-segment: Fair. Consider adding a digit or making your password longer.

Green: Fair. Consider adding a digit or making your password longer.

Tiny: Fair. Consider adding a digit or making your password longer.

Huge: Fair. Consider adding a digit or making your password longer.

No suggestions: Fair.

Text-only: Fair. Consider adding a digit or making your password longer.
Visual Differences

Type new password:

8-character minimum; case sensitive

Baseline meter
Fair. Consider adding a digit or making your password longer.

Three-segment
Fair. Consider adding a digit or making your password longer.

Green
Fair. Consider adding a digit or making your password longer.

Tiny
Fair. Consider adding a digit or making your password longer.

Huge

No suggestions
Fair.

Text-only
Fair. Consider adding a digit or making your password longer.
Scoring Differences

Type new password: usenIX$05
8-character minimum; case sensitive

Baseline meter
Excellent!

Half-score
Poor. Consider adding a different symbol or making your password longer.

One-third-score
Bad. Consider adding a different symbol or making your password longer.

Nudge-16
Poor. Consider making your password longer.

Nudge-Comp8
Excellent!
Key Results

• Stringent meters with visual bars increased resistance to guessing
• Visual differences did not significantly impact resistance to guessing
• No significant impact on memorability
(Revisiting) RNN Password Model Key Results

- Neural networks produce better guesses than previous methods
- Larger model not a major advantage
- Browser implementation in JavaScript
Intelligibility (Explanations)
Building a Data-Driven Meter

We designed & tested a meter with:
1) Principled strength estimates (RNN)
2) Data-driven feedback to users
We designed & tested a meter with:

1) Principled strength estimates (RNN)
2) Data-driven feedback to users
We designed & tested a meter with:

1) Principled strength estimates (RNN)
2) Data-driven feedback to users
Provide Intelligible Explanations

Unic0rns

Don't use simple transformations of words or phrases (unic0rns → Unic0rns)

Capitalize a letter in the middle, rather than the first character

- 21 characteristics
- Weightings determined with regression
After Requirements Are Met…
…Displays Score Visually
…Provides Text Feedback
…Gives Detail (Password Shown)

Create Your Password

Username
blase

Password
CryptoUnicorn3

Show Password & Detailed Feedback

Confirm Password

Your password could be better.

- Don’t use dictionary words (Unicorn) or words used on Wikipedia (Crypto)
- Consider inserting digits into the middle, not just at the end
- Consider making your password longer than 14 characters

A better choice: C3ryptoUnicorn@

How to make strong passwords
…Offers Explanations
Explanations Shown in Modal

Ways to Improve Your Password

CryptoUnicorn3

Show Password & Detailed Feedback

A better choice: C3ryptoUnICorn@

Your password could be better.

- Don’t use dictionary words (Unicorn) or words used on Wikipedia (Crypto). Attackers use software that automatically guesses millions of words commonly found in dictionaries, wordlists, or other people’s passwords.

- Consider inserting digits into the middle, not just at the end. 38% of people also put digits at the end of the password.

- Consider making your password longer than 14 characters. In recent years, attackers have gotten much better at guessing passwords under 16 characters.

How to make strong passwords

OK
Standard Feedback

Create Your Password

Username
blase

Password

Confirm Password

Your password could be better.
- Don’t use dictionary words (Unicorn) or words used on Wikipedia (Crypto)
- Consider making your password longer than 14 characters

A better choice: CRYPTOUnicorn@

How to make strong passwords
Authentication in Practice: Password Add-Ons / Alternatives
Two-Factor Auth

Enter this verification code if prompted during account sign-in:

alice@gmail.com
547173

alice.work@company.com
166123

Choose an authentication method

- **Duo Push** (RECOMMENDED)
- Call Me
- Passcode

Powered by Duo Security
Single Sign-On: Shibboleth

Diagram from https://docs.shib.ncsu.edu/docs/shibworks.html
For a good (long) explanation, see: https://www.switch.ch/aai/demo/
Physical Tokens / Smart Cards

• Codes based on a cryptographic key
  – Token manufacturer also knows the key
• What if there is a breach?
Authentication in Practice: I Forgot My Password
Resetting Accounts

• I forgot my password!
• Send an email?
• Security questions?
• In-person verification?
• Other steps?
• (No backup)
Authentication in Practice: Password Managers
Password Managers

• Trust all passwords to a single master password (still a good idea in most cases)
  – Also trust software
  – Centralized vs. decentralized architectures
Authentication in Practice: Password Reuse 😞
Monitoring the Black Market
Facebook buys black market passwords to keep your account safe

The company's security chief says account safety is about more than just building secure software.
Measuring Vulnerability to Password Reuse

Measuring Vulnerability to Password Reuse

Create Password or Passphrase

You have already used this password before. Please choose a different one.
# UChicago Password History Database

<table>
<thead>
<tr>
<th>Username</th>
<th>Hash of Password</th>
<th>Created</th>
<th>Changed</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>weimf</td>
<td>hash(i&lt;3cats1234)</td>
<td>Sep 17, 2016</td>
<td>Jul 1, 2019</td>
<td></td>
</tr>
<tr>
<td>weimf</td>
<td>hash(i&lt;3cats2019!)</td>
<td>Jul 1, 2019</td>
<td>present</td>
<td></td>
</tr>
<tr>
<td>hszym</td>
<td>hash(p@nc@kes99)</td>
<td>Aug 15, 2018</td>
<td>present</td>
<td></td>
</tr>
<tr>
<td>blase</td>
<td>hash(cyb#rS3curity)</td>
<td>Nov 10, 2017</td>
<td>Aug 23, 2019</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
Study Flow

Part 1: Measurement of password reuse over the past 20 years

Part 2: Survey of users who reused a password on their university account
Study Flow

1) Finding credentials in leaked data
2) Generating guesses for university accounts
3) Checking guesses and protecting accounts
4) Surveying impacted users
Study Flow

1) Finding credentials in leaked data
2) Generating guesses for university accounts
3) Checking guesses and protecting accounts
4) Surveying impacted users
Starting Point: UChicago Usernames

1) Finding credentials in leaked data

227,976 Usernames
1) Finding credentials in leaked data

- 450 individual service breaches
Data Source 2: Breach Compilations

1) Finding credentials in leaked data

- 12 large breach compilations
  - Collection #1, Anti Public, etc.
Matching Strategies

1) Finding credentials in leaked data

- username: blase
- blase@uchicago.edu
- blase@cmu.edu
- blase
- blase99@gmail.com
Study Flow

1) Finding credentials in leaked data

2) Generating guesses for university accounts

3) Checking guesses and protecting accounts

4) Surveying impacted users
## Compliance With Historical Password Policies

2) Generating guesses for university accounts

<table>
<thead>
<tr>
<th>Time Period</th>
<th>Length</th>
<th>Character Classes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Password</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2015 - Present</td>
<td>12 - 19</td>
<td>3+</td>
</tr>
<tr>
<td>2010 - 2015</td>
<td>8 - 16</td>
<td>3+</td>
</tr>
<tr>
<td>Prior to 2010</td>
<td>8 - 16</td>
<td>2+</td>
</tr>
<tr>
<td><strong>Passphrase</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2016 - Present</td>
<td>18 - 32</td>
<td>1+</td>
</tr>
<tr>
<td>2014 - 2016</td>
<td>18 - 50</td>
<td>1+</td>
</tr>
</tbody>
</table>
Guessing Common Passwords

2) Generating guesses for university accounts

LinkedIn1
password1
LinkedIn1
P@ssw0rd1234

UChicago1
password1
P@ssw0rd1234
Guessing Common Passwords

2) Generating guesses for university accounts

- LinkedIn1
- UChicago1
- P@ssw0rd1234
2) Generating guesses for university accounts

Password Tweaking

- Heuristic algorithms
  - Das et al. [1]
  - Wang et al. [2]
- Deep learning
  - pass2path [3]
- Hashcat ruleset
  - Best64.rule [4]

password → password → password123 → p@ssw0rd → ...

---

Study Flow

1) Finding credentials in leaked data

2) Generating guesses for university accounts

3) Checking guesses and protecting accounts

4) Surveying impacted users
Transferring Guesses

3) Checking guesses and protecting accounts

| Username | Password  | ...
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>nisenoff</td>
<td>letmein123</td>
<td>...</td>
</tr>
<tr>
<td>blase</td>
<td>qwerty123</td>
<td>...</td>
</tr>
<tr>
<td>mgolla</td>
<td>Monkey&lt;3</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Credential Guesses

UChicago Password History Database
Notifying Vulnerable Users

3) Checking guesses and protecting accounts
1) Finding credentials in leaked data

2) Generating guesses for university accounts

3) Checking guesses and protecting accounts

4) Surveying impacted users
Survey Users

4) Surveying impacted users

40 participants
Ethical Considerations

• Approved by IRB
• Study design informed by discussions with:
  – IT Leadership
  – Provost’s office
  – Communications team
  – Alumni association
• Minimizing access to password history database
• Password resets to protect UChicago users
Results

12,247 correct guesses based on password reuse
Results

We guessed at least one password for:
Results

We guessed at least one password for:

- 4.5% of all users in the password history database
Results

We guessed at least one password for:

- 4.5% of all users in the password history database
- 6.5% of users for whom we made a guess
Results

We guessed at least one password for:

- 4.5% of all users in the password history database
- 6.5% of users for whom we made a guess
- 32.0% of users with a uchicago.edu email in a data breach
We guessed the current password for 3,618 accounts
Sources of Correct Guesses

71 individual service breaches
Sources of Correct Guesses

71 individual service breaches
…and all 12 breach compilations
Number of Vulnerable Accounts Over Time

![Graph showing the number of vulnerable accounts over time. The graph includes two lines: one for Password Reuse Guesses (blue) and another for Common Password Guesses (purple). There are vertical lines indicating significant changes in the number of accounts, with a notable peak around 2014 and a decline after 2016.]
Some Accounts Remained Vulnerable For Years

Number of Accounts

Date Breach Occurred

Date Breach Became Public
Some Accounts Were Quickly Exploited
Some Accounts Were Quickly Exploited
Some Accounts Were Quickly Exploited

Password resets due to suspicious activity
Passwords Created at UChicago Before Breach

- **5,398 passwords**
  - Valid only before data breach

- **5,915 passwords**
  - Valid before & after data breach

- **934 passwords**
  - Valid only after data breach

Before Data Breach | Date of Data Breach | After Data Breach
## Impact on Specific User Groups

<table>
<thead>
<tr>
<th></th>
<th>LinkedIn</th>
<th>Chegg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Students</td>
<td>11.2%</td>
<td>41.4%</td>
</tr>
<tr>
<td>Faculty</td>
<td>54.3%</td>
<td>2.2%</td>
</tr>
</tbody>
</table>
## Importance of Cracking Hashes

<table>
<thead>
<tr>
<th>Plaintext</th>
<th>Hashed</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>85.3%</strong></td>
<td><strong>14.7%</strong></td>
</tr>
</tbody>
</table>

- Sunshine!
- correctbatteryhorsestaple
- i@mforg3tful!
- ineedapassword

<table>
<thead>
<tr>
<th>Hashed Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>5F4DCC3B5AA765D61D8327DEB882CF99</td>
</tr>
<tr>
<td>482C811DA5D5B4BC6D497FFA98491E38</td>
</tr>
<tr>
<td>62099D23A9D9910879D67449D9E084ED</td>
</tr>
<tr>
<td>1C8F93D67A694EE1DE6363D20228DAC8</td>
</tr>
</tbody>
</table>
Importance of Tweaking Guesses

Verbatim Reuse 54.7%

Tweaked Passwords 45.3%

- Password
  password
- Password!
  password123
- p@ssw0rd
- pa$$word
User Reactions and Experiences (N = 40)

• Users are aware they are reusing passwords
• Users know about some, but not all, relevant breaches
• Some users were unaware they had accounts on sites that had suffered a data breach

“I didn't know that I even had a Chegg account…” (P2)
Key Recommendations for Organizations

- Implement processes to expire unused accounts
- Using credential checking services when passwords are created is not enough
- Promptly check high-risk breaches when they become public
- Check for reuse of hashed and tweaked passwords in less common data breaches
- Use 2FA and consider moving to FIDO2 Passwordless Authentication
Notifying Users About Password Reuse

Password-Reuse Notifications
Authentication in Practice: Checking for Compromised Credentials
Checking for Compromised Credentials

- [Google launches password checkup feature](https://www.zdnet.com/article/google-launches-password-checkup-feature-will-add-it-to-chrome-later-this-year/)
- [iOS monitors your passwords](https://ios.gadgethacks.com/how-to/ios-14-monitors-your-passwords-protect-you-against-data-breaches-heres-works-0341281/)
Checking for Compromised Credentials

Under the hood: How Password Checkup helps keep your accounts safe

01
Whenever Google discovers a username and password exposed by a data breach, we store a strongly hashed and encrypted copy of the data.

02
When you log in to a site you use, Password Checkup will reveal a strongly hashed and encrypted copy of your username and password to Google. This ensures that Google never learns your account details.

03
We use private set intersection with thresholding to search through every unique username and password without revealing your account details, or anyone else, during the process.

04
The final check for whether your username or password was in a data breach is entirely local. If your account details were exposed, you should change your password immediately.

https://security.googleblog.com/2019/02/protect-your-accounts-from-data.html
Authentication in Practice: Moving Towards A Passwordless World?
Passwordless FIDO2

FIDO2 BRINGS SIMPLER, STRONGER AUTHENTICATION TO WEB BROWSERS

COMMITTED SUPPORT FROM LEADING BROWSERS

PLEASE AUTHENTICATE

USE WHAT'S ON YOUR PC...

OR USE YOUR DEVICE!

FIDO AUTHENTICATION: THE NEW GOLD STANDARD

Protects against phishing, man-in-the-middle and attacks using stolen credentials
Log in with a single gesture – HASSLE FREE!
Already supported in market by top online services
Passwordless FIDO2

- Goal: Authenticate on web using public-key crypto
- Created by the FIDO Alliance, now a W3C standard
  - [https://www.w3.org/TR/webauthn-2/](https://www.w3.org/TR/webauthn-2/)
- Originally intended to be implemented in specialized hardware OR in software using a TPM/TEE
Passwordless FIDO2: WebAuthn Protocol
Integrating WebAuthn With Okta

Image from https://developer.okta.com/docs/guides/authenticators-web-authn/main/
Passwordless FIDO2: User Interaction

- Type a PIN into the device, present biometric (fingerprint) to hardware reader, or press a button on the key
• Goal: Make FIDO2 / WebAuthn more usable by syncing the private key across devices
  • See: https://developers.google.com/identity/passkeys
  • Example of Google’s changing approach over the years:

**Our Passwordless journey**

Passkeys bring us much closer to the passwordless future we’ve been mapping out for over a decade.

<table>
<thead>
<tr>
<th>Year</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>2008</td>
<td>Launched Google Password Manager for easier and safer sign-ins.</td>
</tr>
<tr>
<td>2011</td>
<td>Enabled 2-Step Verification (2SV) for Google accounts.</td>
</tr>
<tr>
<td>2012</td>
<td>Introduced phishing-resistant security key for Google employees.</td>
</tr>
<tr>
<td>2013</td>
<td>Joined the FIDO Alliance to drive open standards for a passwordless world.</td>
</tr>
<tr>
<td>2014</td>
<td>Expanded phishing-resistant security keys for everyone.</td>
</tr>
<tr>
<td>2017</td>
<td>Introduced Advanced Protection Program (APP) for high-risk users.</td>
</tr>
<tr>
<td>2019</td>
<td>Extended our FIDO support in Android for passwordless re-auth across websites.</td>
</tr>
<tr>
<td>2023</td>
<td>Enabled passkeys for Google Accounts, Workspace customers and 3rd party partners on Chrome and Android.</td>
</tr>
</tbody>
</table>
What about Biometrics?
Biometrics

- Fingerprint
- Iris scans or retina scans
- Face recognition
- Finger/hand geometry
- Voice or speech recognition
- The way you type
- (Many others)
Practical Challenges for Biometrics

- Immutable (can’t be changed)
- Potentially sensitive data
- High equipment costs
- Sensitive to changes in the environment
- Biometrics can change over time
Storing Biometrics: Templates
Smartphone Biometrics

- Purpose is to reduce the number of times a user must enter their password
- Falls back to the password
- Some facial recognition systems can be tricked by a photo
- Some fingerprint recognition systems can be tricked by a gummy mold

*Images fair use from creativebits.org, and businessinsider.com.