Authentication and Access Control

Blase Ur, David Cash, Ben Zhao
UChicago CMSC 23200//33250
Who Am I?

• Ben Zhao
  – Distinguished professor
  – Co-director of SAND Lab
  – Fan of pandas
Or Am I?
How (and why) do we authenticate users?
Why We Authenticate

• Verify that **people** or **things** (e.g., a server) are who they claim to be

• Authentication ≠ Authorization
  – *Authorization* is deciding whether an entity should have access to a given resource

• Terminology:
  – **Principal**: the legitimate owner of an identity
  – **Claimant**: entity attempting to be authenticated as the principal
Relationships Among Concepts

- How is **authentication** related to **access control**?
- How is the design of **secure systems** related to **authentication**?
- How is **authentication** related to human factors?
How We Authenticate (1/2)
How We Authenticate (1/2)

• Something you know
  – Password
  – PIN (Personal Identification Number)
How We Authenticate (1/2)

• Something you know
  – Password
  – PIN (Personal Identification Number)

• Something you have
  – Smart card
  – Private key (of a public-private key pair)
  – Phone (running particular software)
How We Authenticate (1/2)

• Something you know
  – Password
  – PIN (Personal Identification Number)

• 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)
How We Authenticate (2/2)

• Somewhere you are
  – Location-limited channels
How We Authenticate (2/2)

• Somewhere you are
  – Location-limited channels

• Someone you know (social authentication)
  – Someone vouches for you
  – You can identify people you should know
How We Authenticate (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 (e.g., UChicago shib)
  – PKI Certificate Authorities
Why Are Passwords So Prevalent?
Why Are Passwords So Prevalent?

- Easy to use
- Easy to deploy
- Nothing to carry
- No “silver-bullet” alternative
Attacks on Passwords Are Common
Attacks Against Passwords
Attacks Against Passwords

• Online attack
  – Try passwords on a live system
  – Usually rate-limited
Attacks Against Passwords

• Online attack
  – Try passwords on a live system
  – Usually rate-limited
Attacks Against Passwords

• Online attack
  – Try passwords on a live system
  – Usually rate-limited

• Offline attack
  – Try to guess passwords from the password store / password database
Some Breached Companies
Attacks Against Passwords

• Online attack
  – Try passwords on a live system
  – Usually rate-limited

• Offline attack
  – Try to guess passwords from the password store / password database

• Phishing attack
Attacks Against Passwords

- Online attack
  - Try passwords on a live system
  - Usually rate-limited
- Offline attack
  - Try to guess passwords from the password store / password database
- Phishing attack
- Shoulder surfing
Attacks Against Passwords

• Online attack
  – Try passwords on a live system
  – Usually rate-limited

• Offline attack
  – Try to guess passwords from the password store / password database

• Phishing attack

• Shoulder surfing

• Attack password-protected file / device
Storing Passwords

• **Hash and salt** passwords

• Hash function: one-way function
  – Traditionally designed for efficiency (e.g., MD5)
  – Password-specific hash functions (e.g., bcrypt, scrypt, PBKDF2)
Storing Passwords

• Salt: random string assigned per-user
  – Combine the password with the salt, then hash it
  – Stored alongside the hashed
  – Prevents the use of rainbow tables
Data-Driven Statistical Attacks

- (2009) 32 million passwords: RockYou

- (2016) 117 million passwords: LinkedIn

- (2017) 3 billion passwords: Yahoo!

- Total: > 5 billions of passwords stolen from > 300 services
Offline Attack

- Attacker compromises database
  - hash(“Blase”) = $2a$04$iHdEgkI681VdDMc3f7edau9phRwORvhYjqWAIb7hb4B5uFJ01g4zi$

- Attacker makes and hashes guesses
- Finds match $\rightarrow$ try on other sites
  - Password **reuse** is a core problem
Password Reuse-Based Attacks

People Reuse Passwords
**Memory-Hard Hash Function**

<table>
<thead>
<tr>
<th>Email</th>
<th>Argon2i Hash of Password</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:jim@mail.com">jim@mail.com</a></td>
<td>$argon2i$v=19$m=4096,...</td>
</tr>
</tbody>
</table>

**Rate-Limiting Guessing**

- I'm not a robot

**Password Strength Meter**

- Username: acmccs18
- Password: acmccs18
- Your password could be better.
  - Consider inserting digits into the middle, not just at the end
  - Make your password longer than 8 characters
  - Consider using 1 or more symbols
- A better choice: a#D18cmccs
  - How to make strong passwords
<table>
<thead>
<tr>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:jim@mail.com">jim@mail.com</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Email</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:jane@aol.com">jane@aol.com</a></td>
</tr>
<tr>
<td><a href="mailto:jessey@gmx.net">jessey@gmx.net</a></td>
</tr>
<tr>
<td><a href="mailto:jenny@gmail.com">jenny@gmail.com</a></td>
</tr>
<tr>
<td><a href="mailto:jim@mail.com">jim@mail.com</a></td>
</tr>
<tr>
<td><a href="mailto:john@hotmail.com">john@hotmail.com</a></td>
</tr>
</tbody>
</table>

...
<table>
<thead>
<tr>
<th>Email</th>
<th>SHA-1 Hash of Password</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:jane@aol.com">jane@aol.com</a></td>
<td>7c4a8d09ca3762af61e595209</td>
</tr>
<tr>
<td><a href="mailto:jessey@gmx.net">jessey@gmx.net</a></td>
<td>5baa61e4c9b93f3f0682250b6</td>
</tr>
<tr>
<td><a href="mailto:jenny@gmail.com">jenny@gmail.com</a></td>
<td>7c222fb2927d828af22f59213</td>
</tr>
<tr>
<td><a href="mailto:jim@mail.com">jim@mail.com</a></td>
<td>ba93664a90285b9ff18a7a081</td>
</tr>
<tr>
<td><a href="mailto:john@hotmail.com">john@hotmail.com</a></td>
<td>b1b3773a05c0ed0176787a4f1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Crack All The Things!

```bash
$> hashcat -m 100 -a0 $TARGET $DICT
123456
Password
R0cky!17
Football!17
CanadaRocks!
```

<table>
<thead>
<tr>
<th>Email</th>
<th>Cracked SHA-1 Hashes</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:jane@aol.com">jane@aol.com</a></td>
<td>123456</td>
</tr>
<tr>
<td><a href="mailto:jessey@gmx.net">jessey@gmx.net</a></td>
<td>5baa61e4c9b93f3f0682250b6</td>
</tr>
<tr>
<td><a href="mailto:jenny@gmail.com">jenny@gmail.com</a></td>
<td>Canada4ever</td>
</tr>
<tr>
<td><a href="mailto:jim@mail.com">jim@mail.com</a></td>
<td>R0cky!17</td>
</tr>
<tr>
<td><a href="mailto:john@hotmail.com">john@hotmail.com</a></td>
<td>HikingGuy89</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Email</td>
<td>Argon2i Hash of Password</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------</td>
</tr>
<tr>
<td><a href="mailto:jim@mail.com">jim@mail.com</a></td>
<td>$argon2i$v=19$m=4096,...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Dead On Arrival
Dead On Arrival

<table>
<thead>
<tr>
<th>Email</th>
<th>Argon2i Hash of Password</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>...</td>
</tr>
<tr>
<td><a href="mailto:jim@mail.com">jim@mail.com</a></td>
<td>$argon2i$v=19$m=4096,</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Email</th>
<th>Cracked SHA-1 Hashes</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:jane@aol.com">jane@aol.com</a></td>
<td>123456</td>
</tr>
<tr>
<td><a href="mailto:jessey@gmx.net">jessey@gmx.net</a></td>
<td>5baa61e4c9b93f3f068225</td>
</tr>
<tr>
<td><a href="mailto:jenny@gmail.com">jenny@gmail.com</a></td>
<td>0b6</td>
</tr>
<tr>
<td><a href="mailto:jim@mail.com">jim@mail.com</a></td>
<td>Canada4ever</td>
</tr>
<tr>
<td><a href="mailto:john@hotmail.com">john@hotmail.com</a></td>
<td>R0cky!17</td>
</tr>
<tr>
<td></td>
<td>HikingGuy89</td>
</tr>
<tr>
<td></td>
<td>...</td>
</tr>
</tbody>
</table>
Dead On Arrival

Email Cracked
...
...
... jim@mail.com \textit{R0cky!17}
...
...

1 guess is enough!

Email Cracked SHA-1 Hashes

\begin{tabular}{|l|l|}
  \hline
  Email & Cracked SHA-1 Hashes \\
  \hline
  jane@aol.com & 123456 \\
  jessey@gmx.net & 5baa61e4c9b93f3f0682250b6 \\
  jenny@gmail.com & \textit{Canada4ever} \\
  jim@mail.com & \textit{R0cky!17} \\
  john@hotmail.com & \textit{HikingGuy89} \\
  \ldots & \ldots \\
  \hline
\end{tabular}
How LinkedIn’s password sloppiness hurts us all

Anatomy of a password disaster: Adobe’s giant.

Facebook says 30 million accounts had personal data stolen in recent breach

Hackers were able to access name, birthdate and other data in nearly half of the 30 million accounts that were affected.

You Can Now Look Up Your Terrible 2006 MySpace Password

written by LORENZO FRANCESCHI-BICCHIERAI

Facebook

Connect with friends and the world. Sign Up
Monitoring the Black Market

The image shows a screenshot of a website with a listing for "LinkedIn 167M". The listing details include:

- **Action:** Order
- **Quantity:** 0.0000 BTC
- **Postage Option:** Not specified
- **Escrow:** Yes, escrow by RealDeal is available.
- **Class:** Digital
- **Ships From:** Worldwide

The website interface includes navigation links to Home, My RealDeal, and Support, along with a logout button.
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.

BY KATIE COLLINS | NOVEMBER 9, 2016 12:56 PM PST
Password-Reuse Notifications
Notification Goals

- timely
- secure actions
- trust
- sufficient background
- legitimate
Our Model Password-Reuse Notification

Please create a new password

Dear Jo,

During routine checks, we learned of a potential security incident in which your AcmeCo account login and password may have been compromised. This incident was likely a data breach of a service unrelated to AcmeCo, but because many people reuse similar passwords on multiple sites, your AcmeCo login information may have been affected. While we have not detected any suspicious activity on your AcmeCo account, you must create a new password as a precaution.

Please go to the AcmeCo website or mobile app and we will guide you through creating a new password.

To further improve your online security, we recommend:
• Enabling AcmeCo’s Two-Factor Authentication.
• Changing all similar passwords on other accounts.
• Using a password manager.

If you would like to learn more, please visit https://acmeco.com/security. If you have any questions or need any further assistance, please visit the Help Center at https://acmeco.com/help.

Thanks,
The AcmeCo Team
Understanding Users’ Password Behaviors
Some Ways to Understand Users

• Retrospective analysis of user-created passwords
• Large-scale online studies
• Examine real passwords
• Qualitative studies
Password Cracking

Password-Strength Metrics

• Statistical approaches
  – Traditionally: Shannon entropy
  – Recently: $\alpha$-guesswork

• Disadvantages for researchers
  – Usually no per-password estimates
  – Huge sample required
  – Not real-world attacks
Parameterized Guessability

• How many guesses a particular cracking algorithm with particular training data would take to guess a password
j@mesb0nd007!

Guess # 366,163,847,194
Guess # past cutoff
Guessability in Practice
Questions About Guessability

1) How does guessability used in research compare to an attack by professionals?
2) Would substituting another cracking approach impact research results?
Approach
Approach

4 password sets

5 password-cracking approaches
Four Password Sets
Four Password Sets

• **Basic** (3,062): 8+ characters

```
password
```
Four Password Sets

- **Basic** (3,062): 8+ characters
  - password

- **Complex** (3,000): 8+ characters, 4 classes
  - Pa$$w0rd
Four Password Sets

- **Basic** (3,062): 8+ characters
  - password

- **Complex** (3,000): 8+ characters, 4 classes
  - Pa$$w0rd

- **LongBasic** (2,054): 16+ characters
  - passwordpassword
Four Password Sets

- **Basic** (3,062): 8+ characters
  
  `password`

- **Complex** (3,000): 8+ characters, 4 classes
  
  `Pa$$w0rd`

- **LongBasic** (2,054): 16+ characters
  
  `passwordpassword`

- **LongComplex** (990): 12+ characters, 3+ classes
  
  `pa$$word1234`
Five Cracking Approaches

- John the Ripper
- Hashcat
- Markov models
- Probabilistic Context-Free Grammar
- Professionals
John the Ripper

- Guesses variants of input wordlist
John the Ripper

• Guesses variants of input wordlist
• Wordlist mode requires:
  – Wordlist (passwords and dictionary entries)
  – Mangling rules
John the Ripper

• Guesses variants of input wordlist
• Wordlist mode requires:
  – Wordlist (passwords and dictionary entries)
  – Mangling rules
• Speed: Fast
John the Ripper

• Guesses variants of input wordlist
• Wordlist mode requires:
  – Wordlist (passwords and dictionary entries)
  – Mangling rules
• Speed: Fast
  – $10^{13}$ guesses
John the Ripper

• Guesses variants of input wordlist
• Wordlist mode requires:
  – Wordlist (passwords and dictionary entries)
  – Mangling rules
• Speed: Fast
  – $10^{13}$ guesses
• “JTR”
John the Ripper

- wordlist
- rules

guesses
John the Ripper

usenix
security

wordlist

rules

guesses
John the Ripper

usenix
security

[ ]
[add 1 at end]
[change e to 3]

wordlist

rules

guesses
John the Ripper

- `usenix`
- `security`

[ ]
- [add 1 at end]
- [change e to 3]

- `wordlist`
- `rules`

- `guesses`
  - `usenix`
  - `security`
  - `usenix1`
  - `security1`
  - `us3nix`
  - `s3curity`
John the Ripper

**Wordlist**

- *usenix*
- *security*

**Rules**

- [add 1 at end]
- [change e to 3]

**Guesses**

- *usenix*
- *security*
- *usenix1*
- *security1*
- *us3nix*
- *s3curity*
John the Ripper

usenix
security

[ ]
[add 1 at end]
[change e to 3]

wordlist

rules

usenix
security
usenix1
security1
us3nix
s3curity
Hashcat

• Guesses variants of input wordlist
Hashcat

- Guesses variants of input wordlist
- Wordlist mode requires:
  - Wordlist (passwords and dictionary entries)
  - Mangling rules
Hashcat

• Guesses variants of input wordlist
• Wordlist mode requires:
  – Wordlist (passwords and dictionary entries)
  – Mangling rules
• Speed: Fast
Hashcat

• Guesses variants of input wordlist
• Wordlist mode requires:
  – Wordlist (passwords and dictionary entries)
  – Mangling rules
• Speed: Fast
  – $10^{13}$ guesses
Hashcat

- wordlist
- rules

• guesses
Hashcat

usenix

security

[ ]
[add 1 at end]
[change e to 3]

wordlist

rules

guesses
Hashcat

- usenix
- security

[ ]
[add 1 at end]
[change e to 3]

wordlist

rules

- usenix
- usenix1
- us3nix

security

- security1
- s3curity

guesses
Hashcat

[ ]
[add 1 at end]
[change e to 3]

usenix
security

wordlist

usenix
usenix1
us3nix
security
security1
s3curity

rules
guesses
Markov Models

- Predicts future characters from previous
Markov Models

• Predicts future characters from previous
• Approach requires weighted data:
  – Passwords
  – Dictionaries
Markov Models

• Predicts future characters from previous
• Approach requires weighted data:
  – Passwords
  – Dictionaries
• Ma et al. IEEE S&P 2014
Markov Models

- Predicts future characters from previous
- Approach requires weighted data:
  - Passwords
  - Dictionaries
- Ma et al. IEEE S&P 2014
- Speed: Slow
Markov Models

- Predicts future characters from previous
- Approach requires weighted data:
  - Passwords
  - Dictionaries
- Ma et al. IEEE S&P 2014
- Speed: Slow
  - $10^{10}$ guesses
Markov Models

diagram
Markov Models

usenixsecurity
Markov Models

usenixsecurity
Markov Models
Markov Models
Probabilistic Context-Free Grammar

• Generate password grammar
  – Structures
  – Terminals
Probabilistic Context-Free Grammar

- Generate password grammar
  - Structures
  - Terminals
  - Based on Weir et al. IEEE S&P 2009
Probabilistic Context-Free Grammar

• Generate password grammar
  – Structures
  – Terminals
• Kelley et al. IEEE S&P 2012
  – Based on Weir et al. IEEE S&P 2009
• Speed: Slow Medium
Probabilistic Context-Free Grammar

• Generate password grammar
  – Structures
  – Terminals

• Kelley et al. IEEE S&P 2012
  – Based on Weir et al. IEEE S&P 2009

• Speed: Slow Medium
  – $10^{14}$ guesses
Probabilistic Context-Free Grammar

- Generate password grammar
  - Structures
  - Terminals
  - Based on Weir et al. IEEE S&P 2009
- Speed: Slow Medium
  - $10^{14}$ guesses
- “PCFG”
PCFG

password
password
password123
usenix3
5ecurity
iloveyou
nirvana123
PCFG

password

password

usenix

security

iloveyou

nirvana
PCFG

password

password

password

usenix

security

iloveyou

nirvana

123

123
Professionals ("Pros")
Professionals ("Pros")

- Contracted KoreLogic
  - Password audits for Fortune 500 companies
  - Run DEF CON "Crack Me If You Can"
Professionals ("Pros")

• Contracted KoreLogic
  – Password audits for Fortune 500 companies
  – Run DEF CON “Crack Me If You Can”
• Proprietary wordlists and configurations
Professionals ("Pros")

• Contracted KoreLogic
  – Password audits for Fortune 500 companies
  – Run DEF CON “Crack Me If You Can”
• Proprietary wordlists and configurations
  – $10^{14}$ guesses
Professionals (“Pros”)

• Contracted KoreLogic
  – Password audits for Fortune 500 companies
  – Run DEF CON “Crack Me If You Can”

• Proprietary wordlists and configurations
  – $10^{14}$ guesses
  – Manually tuned, updated
Approach

4 password sets

- password
- iloveyou
- teamo123
- ...

- passwordpassword
- 1234567812345678
- !1@2#3$4%5^6&7*8
- ...

- Pa$$word
- iLov3you!
- 1QaZ2W@x
- ...

- pa$$word1234
- 12345678asDF
- !q1q!q1q!q1q
- ...

5 approaches

- hashcat
  advanced password recovery

- John the Ripper

- KoreLogic
  Security
Outline of Results

• Importance of Configuration
• Comparison of Approaches
• Impact on Research Analyses
Configuration Is Crucial

LongComplex

Percent guessed

0% 10% 20% 30% 40%

Guesses

$10^1$ $10^3$ $10^5$ $10^7$ $10^9$ $10^{11}$ $10^{13}$ $10^{15}$

HC-Best64
Configuration Is Crucial

LongComplex

Percent guessed

Guesses
Configuration Is Crucial

LongComplex

Percent guessed

Guesses

HC-Generated2

HC-SpiderLabs

HC-Best64
Configuration Is Crucial

LongComplex

Percent guessed vs. Guesses for different configurations:
- HC-MWR
- HC-Generated2
- HC-SpiderLabs
- HC-Best64
Configuration Is Crucial

LongComplex
Outline of Results

• Importance of Configuration
• Comparison of Approaches
• Impact on Research Analyses
Comparison for Basic Passwords
Comparison for Basic Passwords

Graph showing the percent guessed against guesses for basic passwords.
Comparison for Basic Passwords

![Graph showing the percentage of passwords guessed over guesses. It compares PCFG and Markov models. The x-axis represents guesses, and the y-axis represents the percent guessed.](image-url)
Comparison for Basic Passwords

![Graph showing comparison for basic passwords]
Comparison for Basic Passwords
Comparison for Basic Passwords
Comparison for Complex Passwords
Comparison for Complex Passwords
Comparison for Complex Passwords
Comparison for Complex Passwords

![Graph showing the percent guessed against guesses for different types of complex passwords. The graph compares PCFG, JTR, and Markov methods.](image)
Comparison for Complex Passwords

![Graph showing the percent guessed vs guesses for different methods: PCFG, Hashcat, and Markov. The graph illustrates the efficiency of each method in cracking complex passwords.]
Comparison for Complex Passwords
Min_auto Conservative Proxy for Pros
Per-Password Highly Impacted

Password!
Per-Password Highly Impacted

• JTR guess # 801

Password!
Per-Password Highly Impacted

• JTR guess # 801
• Not guessed in $10^{14}$ PCFG guesses

Password!
Per-Password Highly Impacted

- JTR guess # 801
- Not guessed in $10^{14}$ PCFG guesses

Password!
How Do We Help Users Make Better Passwords?
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
Problem 3: Unhelpful Feedback

Please enter a stronger password.
Better Password Scoring

Better Password Scoring

- Real-time feedback
- Runs entirely client-side
- Accurately models password guessability

Recurrent Neural Networks (RNNs)

LSTM Architecture

Image CC by Wes Breazell on the Noun Project
Generating Passwords
Generating Passwords

password → o or maybe 0 or O or ...
Generating Passwords

Next char is:
A: 3%
B: 1%
C: 0.6%
...
O: 55%
...
Z: 0.01%
0: 20%
1: ...

passw
Generating Passwords

Prob: 100%

Next char is:
A: 3%
B: 2%
C: 5%
...
O: 2%
...
Z: 0.2%
0: 1%
1: ...
END: 2%
Generating Passwords

Prob: 100%

Next char is:
A: 3%
B: 2%
C: 5%
...
O: 2%
...
Z: 0.2%
0: 1%
1: ...
END: 2%
Generating Passwords

“C”
Prob: 5%
Generating Passwords

“C”
Prob: 5%

Next char is:
A: 10%
B: 1%
C: 4%
...
O: 8%
...
Z: 0.02%
0: 3%
1: ...
END: 6%
Generating Passwords

“C”
Prob: 5%

Next char is:
A: 10%
B: 1%
C: 4%
...
O: 8%
...
Z: 0.02%
0: 3%
1: ...
END: 6%
Generating Passwords

“CA”
Prob: 0.5%

Next char is:
A: 3%
B: 10%
C: 7%
...
O: 1%
...
Z: 0.03%
0: 2%
1: ...
END: 12%
Generating Passwords

“CAB”
Prob: 0.05%

Next char is:
A: 3%
B: 10%
C: 7%
...
O: 1%
...
Z: 0.03%
0: 2%
1: ...
END: 3%
Generating Passwords

“CAB”
Prob: 0.05%

Next char is:
A: 4%
B: 3%
C: 1%

... 
O: 2%

... 
Z: 0.01%
0: 4%
1: ...
END: 12%
Generating Passwords

“CAB”
Prob: 0.05%

Next char is:
A: 4%
B: 3%
C: 1%
...
O: 2%
...
Z: 0.01%
0: 4%
1: ...
END: 12%
Generating Passwords

“CAB”
Prob: 0.006%
Descending Probability Order

CAB - 0.006%
CAC - 0.0042%
ADD1 - 0.002%
CODE - 0.0013%
...

149
Design Space

• Model size: 3mb (browser) vs. 60mb (GPU)
• Transference learning
  – Novel password-composition policies
• Training data
  – Natural language
• (Many others)
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
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
2) Data-driven feedback to users
Provide Intelligible Explanations

Unic0rns

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

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

• 21 characteristics
• Weightings determined with regression
We designed & tested a meter with:
1) Principled strength estimates
2) Data-driven feedback to users
Main Screen…

Username
blase

Password

Don't reuse a password from another account! (Why?)

Your password must:
- Contain 12+ characters
- Use 3+ of the following: uppercase letters; lowercase letters; digits; symbols

How to make strong passwords

Continue
…Shows Requirements

Create Your Password

Username: blase
Password: ********
Show Password

Don't reuse a password from another account! (Why?)

Your password must:

- [ ] Contain 12+ characters
- [x] Use 3+ of the following: uppercase letters; lowercase letters; digits; symbols

How to make strong passwords

Continue
…Emphasizes Avoiding Reuse

Don't reuse a password from another account! (Why?)

Your password must:
- Use 3+ of the following: uppercase letters, lowercase letters, digits, symbols

How to make strong passwords
…Provides Abstract Advice
After Requirements Are Met…

Create Your Password

Username
blase

Password
••••••••

Show Password & Detailed Feedback

Continue

Your password could be better.

- Don’t use dictionary words or words used on Wikipedia [Why?]
- Consider inserting digits into the middle [Why?]
- Consider making your password longer [Why?]

See Your Password With Our Improvements

How to make strong passwords
…Displays Score Visually
...Provides Text Feedback
…Gives Detail (Password Shown)

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

Continue
Offers Explanations

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) (Why?)
- Consider inserting digits into the middle, not just at the end (Why?)
- Consider making your password longer than 14 characters (Why?)

A better choice: C3ryptoUnicorn@

How to make strong passwords
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
We designed & tested a meter with:
1) Principled strength estimates
2) Data-driven feedback to users
Evaluation

• 2-part online study
  1) Create password; survey; recall password
     (48 hours later, send automated email)
  2) Recall password; survey

• 4,509 Mechanical Turk participants
  – Between-subjects
  – Full-factorial design along three dimensions
Dimension 1: Composition Policy

- 8+ characters (1class8)
  - password

- 12+ characters, 3+ classes (3class12)
  - Password1234
Dimension 2: Stringency

- Low
- Medium
- High
Dimension 2: Stringency

- Low: $10^4$ guesses
- Medium: $10^6$ guesses
- High: $10^8$ guesses
Dimension 2: Stringency

- Low: $10^4$ guesses, $10^8$ guesses
- Medium: $10^6$ guesses, $10^{12}$ guesses
- High: $10^8$ guesses, $10^{16}$ guesses
Dimension 3: Feedback
No Feedback

Create Your Password

Username
blase

Password
***********

Show Password & Detailed Feedback

Confirm Password

Continue
Bar Only
Public (Non-Sensitive) Feedback
Standard Feedback

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: CryptoUnicorn@

How to make strong passwords

(Why?)
Standard Feedback

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
Standard Feedback

Create Your Password

Username
blase

Password

Your password could be better.

- Don’t use dictionary words (Unicorn) or words used on Wikipedia (Crypto)

A better choice: CRYPTOUnicorn@

Confirm Password

A better choice: CRYPTOUnicorn@

Continue

How to make strong passwords
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

How to make strong passwords
Standard, No Bar

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: CryptoUnicorn@

How to make strong passwords
Measure Password Guessability

Percent guessed

0% 20% 40% 60%

Guesses

$10^1$ $10^3$ $10^5$ $10^7$ $10^9$ $10^{11}$ $10^{13}$ $10^{15}$
Measure Password Guessability
Measure Password Guessability

Percent guessed

0%
20%
40%
60%

Guesses

$10^1$  $10^3$  $10^5$  $10^7$  $10^9$  $10^{11}$  $10^{13}$  $10^{15}$
Measure Password Guessability

Password guessability increases as the number of guesses increases. The graph shows the percentage of passwords guessed against the number of guesses. As the number of guesses rises, it becomes significantly harder to guess passwords.
Measure Password Guessability

- Percent guessed vs. Guesses
Feedback → More Secure Passwords

![Graph showing the percent of guesses for different password feedback](image-url)

- 1c8-None
- 1c8-Bar-M
Feedback → More Secure Passwords

The graph shows the percent guessed against the number of guesses for different password schemes:

- **1c8-None**
- **1c8-Bar-M**
- **1c8-Std-M**

The x-axis represents the number of guesses (in powers of 10), while the y-axis represents the percent guessed (0% to 60%). The graph indicates that more secure password schemes (1c8-Bar-M and 1c8-Std-M) require a significantly higher number of guesses to be cracked compared to the non-secure scheme (1c8-None).
Feedback $\rightarrow$ More Secure Passwords

The graph shows the percent guessed against the number of guesses, with various password conditions represented. The conditions include:

- 1c8-None
- 1c8-Bar-M
- 1c8-Std-M
- 3c12-None
- 3c12-Bar-M
- 3c12-Std-M

The x-axis represents the number of guesses, ranging from $10^1$ to $10^{15}$.
Usability Results

• Feedback did not significantly impact password memorability
• More feedback → more difficult, annoying
• All features had value for some participants
Feedback ➔ More Secure Passwords

https://github.com/cupslab/password_meter

- Help us improve the meter
- Demo: https://cups.cs.cmu.edu/meter
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
iPhone 5S
Touch ID

Android 4.0 Face Unlock

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

- Purpose is to reduce the number of times a user must enter his/her password
Smartphone Biometrics

• Purpose is to reduce the number of times a user must enter his/her password
• Falls back to the password
Smartphone Biometrics

- Purpose is to reduce the number of times a user must enter his/her password
- Falls back to the password
- Face recognition can be tricked by a photo
Smartphone Biometrics

- Purpose is to reduce the number of times a user must enter his/her password
- Falls back to the password
- Face recognition can be tricked by a photo
- Fingerprint recognition can be tricked by a gummy mold
Smartphone Biometrics

• Purpose is to reduce the number of times a user must enter his/her password
• Falls back to the password
• Face recognition can be tricked by a photo
• Fingerprint recognition can be tricked by a gummy mold
• Users find fingerprint unlock convenient, but do not particularly like face unlock
Practical Authentication
Single Sign-On

Login with Facebook

OpenID®
Two-Factor Auth
Physical Tokens

- Codes based on a cryptographic key
  - Token manufacturer also knows the key
- What if there is a breach?
Resetting Accounts

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

• Trust all passwords to a single master password
  – Also trust software
Conclusions

• Authentication is really hard!
  – Hard for system administrators
  – Hard for users

• Unfortunately, authentication is necessary
Access Control

• Access control lists
  – Owner, Group, Other
  – chown
  – chmod

• Role-based access control

• Attribute-based access control

• Context-based access control
Access Control

• Role-based access control
  – Authorization based on role (e.g., “Uchicago student”)

• Attribute-based access control
  – Authorization based on attribute(s) (e.g., “Over 7 feet tall”)

• Context-based access control
  – Authorization decision depends on the context (e.g., time of day)
Access Control in the Internet of Things

Factor: Time of Day

“I would not want anyone trying to use the mower at night. The neighbors would most likely get mad.”
“They would be allowed to use it whenever I am home with them.”
Factor: Location of User

Why not play some music?

What is going on???

“Why do you need to use it if you aren’t close?”
Factor: Location of Device

“If it is used in the bedroom then it would matter who has access.”
Factor: Explicit Permission

“Can I...?"

“When they are authorized by the owner.”
Factor: Consequences
Factor: Responsible Usage

“They shouldn't use the lights if they are using them too frequently.”

Icon made by Freepik from www.flaticon.com
Design Implications
For Contextual Access Control
Current: Guest vs. Owner

What level of access do you want to give “John”?

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Guest</td>
<td>✔️</td>
</tr>
<tr>
<td>Owner</td>
<td></td>
</tr>
</tbody>
</table>
Future: Designing for Relationships

Adding a new user:

- spouse
- teenage child
- young child
- visiting family member
- babysitter
- neighbor
Future: Relationships and Capabilities

Default Settings for a Young Child

Voice Assistant

With permission, allowed to play music

Never allowed to order online
Current: Full or Temporary Access
Future: Contextual Factors

![Smart Home Concept]

Your young child can have access when
- I’m around
- I’m not around