13. Authentication and Access Control

Blase Ur and David Cash
February 10th, 2020
CMSC 23200 / 33250
Who Am I?

• David Cash
  – Distinguished cryptographer
  – Fan of rare plants
  – All-around good guy
Or Am I?
How (and why) do we authenticate users?
Authentication Abstractly

- 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
- Access control lists / policies
- Terminology:
  - **Principal**: the legitimate owner of an identity
  - **Claimant**: entity trying to be authenticated
How We Authenticate (1/2)
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
- 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

• 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 password
  – 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$04$iHdEgkI681VdDMc3f7edau9phRwORvhYjqWAIb7hb4B5uFJO1g4zi$

• Attacker makes and hashes guesses

• Finds match $\rightarrow$ try on other sites
  – Password **reuse** is a core problem
Password Reuse-Based Attacks

Maximilian Golla, Miranda Wei, Juliette Hainline, Lydia Filipe, Markus Dürmuth, Elissa Redmiles, Blase Ur. “What was that site doing with my Facebook Password?” Designing Password-Reuse Notifications. In *Proc. CCS*, 2018.
People Reuse Passwords
Email

jim@mail.com

Argon2i Hash of Password

$argon2i$v=19$m=4096,...

Memory-Hard Hash Function

Rate-Limiting Guessing

Password Strength Meter

- 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>
<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!

$>\text{hashcat} -m\ 100\ -a\ 0\ \$TARGET\ \$DICT$

123456
Password
ROcky!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>
</tbody>
</table>
# Dead On Arrival

AcmeCo

<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>
Dead On Arrival

<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>
<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>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>
</tbody>
</table>
Dead On Arrival

1 guess is enough!

<table>
<thead>
<tr>
<th>Email</th>
<th>Cracked</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:jim@mail.com">jim@mail.com</a></td>
<td>R0cky!17</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Email</th>
<th>Cracked SHA-1 Hashes</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="mailto:jae@ao.com">jae@ao.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>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>
</tbody>
</table>
Anatomy of a password disaster: Adobe’s giant password dump. How LinkedIn’s password sloppiness hurts us all

';--have i been pwned?
Check if you have an account that has been compromised in a data breach

314 pwned website
5,555,329,164 pwned accounts
80,540 pastes
87,820,647 paste accounts

Facebook says 1 billion 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
June 29, 2016 // 11:35 AM EST

written by LORENZO FRANCESCHI-BICHELLE
staff writer
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.

BY KATIE COLLINS | NOVEMBER 9, 2016 12:56 PM PST
Password-Reuse Notifications
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
jamesbond007!

Guess # 366,163,847,194
Guess # past cutoff
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?
<table>
<thead>
<tr>
<th>Password Set 1</th>
<th>Password Set 2</th>
<th>Password Set 3</th>
<th>Password Set 4</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>password</code></td>
<td><code>pa$$word1234</code></td>
<td><code>passwordpassword</code></td>
<td><code>pa$$word1234</code></td>
</tr>
<tr>
<td><code>iloveyou</code></td>
<td><code>12345678asDF</code></td>
<td><code>1234567812345678</code></td>
<td><code>12345678asDF</code></td>
</tr>
<tr>
<td><code>teamo123</code></td>
<td><code>!q1q!q1q!q1q</code></td>
<td><code>!1@2#3$4%5^6&amp;7*8</code></td>
<td><code>!q1q!q1q!q1q</code></td>
</tr>
<tr>
<td><code>…</code></td>
<td><code>…</code></td>
<td><code>…</code></td>
<td><code>…</code></td>
</tr>
</tbody>
</table>

**4 password sets**

**5 password-cracking approaches**
Five Cracking Approaches

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

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

[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

\texttt{usenix}
\texttt{security}

- \texttt{[add 1 at end]}
- \texttt{[change e to 3]}

\texttt{wordlist}

\texttt{guesses}

\texttt{usenix}
\texttt{security}
\texttt{usenix1}
\texttt{security1}
\texttt{us3nix}
\texttt{s3curity}
John the Ripper

usenix
security

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

wordlist

rules

usenix
security
usenix1
security1
us3nix
s3curity

guesses
Hashcat

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

- wordlist
  - hashcat
    - advanced
    - password
    - recovery

- rules

- guesses
Hashcat

usenix
security

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

wordlist

rules

guesses
Hashcat

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

usenix

security

[ ]

[add 1 at end]

[change e to 3]

wordlist

usenix

usenix1

us3nix

security

security1

s3curity

guesses

rules
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

usenixsecurity
Markov Models

usenixsecurity
Markov Models
Markov Models

usenixsecurity
Markov Models

usenixsecurity
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

password

usenix

5ecurity

iloveyou

nirvana

123
PCFG

password
password123
usenix3
5ecurity
iloveyou
nirvana123
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
- ...

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

- Pa$$w0rd
- iLov3you!
- 1QaZ2W@x
- ...

- pa$$word
- 1234
- 12345678asDF
- !q1q!q1q!q1q
- ...

5 approaches

- John the Ripper
- hashcat
- advanced password recovery
- KoreLogic
- SECURITY
Outline of Results

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

LongComplex

Percent guessed

Guesses
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

Guesses

HC-MWR
HC-Generated2
HC-SpiderLabs
HC-Best64
Configuration Is Crucial

LongComplex

Percent guessed

Guesses

0%
10%
20%
30%
40%

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

HC-MWR-big
HC-MWR
HC-Generated2-big
HC-Generated2
HC-SpiderLabs-big
HC-SpiderLabs
HC-Best64-big
HC-Best64
Outline of Results

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

![Graph showing the comparison of different methods for guessing basic passwords. The x-axis represents the number of guesses, ranging from $10^1$ to $10^{15}$. The y-axis represents the percent guessed, ranging from 0% to 80%. The graph compares Min_auto, PCFG, Hashcat, JTR, and Markov methods.]
Comparison for Complex Passwords
Comparison for Complex Passwords
Comparison for Complex Passwords

![Graph showing percent guessed vs guesses for two different models: PCFG and Markov. The graph demonstrates a significant increase in the percent guessed as the number of guesses rises, highlighting the effectiveness of complex passwords.]
Comparison for Complex Passwords
Comparison for Complex Passwords
Comparison for Complex Passwords
Comparison for Complex Passwords
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
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

Twitter

Please enter a stronger password.

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
Generating Passwords
Generating Passwords

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

passw

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


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:

<table>
<thead>
<tr>
<th>Character</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>10%</td>
</tr>
<tr>
<td>B</td>
<td>1%</td>
</tr>
<tr>
<td>C</td>
<td>4%</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>O</td>
<td>8%</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>Z</td>
<td>0.02%</td>
</tr>
<tr>
<td>0</td>
<td>3%</td>
</tr>
<tr>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>END</td>
<td>6%</td>
</tr>
</tbody>
</table>
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

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CAB</td>
<td>0.006%</td>
</tr>
<tr>
<td>CAC</td>
<td>0.0042%</td>
</tr>
<tr>
<td>ADD1</td>
<td>0.002%</td>
</tr>
<tr>
<td>CODE</td>
<td>0.0013%</td>
</tr>
</tbody>
</table>

...
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
After Requirements Are Met...
...Displays Score Visually

Create Your Password

Username
blase

Password
[_masked]

Confirm Password

Your password could be better.
- Don't use dictionary words or words used on Wikipedia
- Consider inserting digits into the middle
- Consider making your password longer

See Your Password With Our Improvements

How to make strong passwords
...Provides Text Feedback

Create Your Password

Username
blase

Password

Show Password & Detailed Feedback

Confirm Password

Your password could be better.

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

See Your Password With Our Improvements

How to make strong passwords

Continue
Create Your Password

Username
blase

Password
CryptoUnicorn3|

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

Username
blase

Password
CryptoUnicorn3|

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: CryptoUnicorn@

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)
  Attacking software can guess 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

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@

How to make strong passwords

Continue
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
iPhone 5S
Touch ID

Android 4.0 Face
Unlock

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

• Purpose is to reduce the number of times a user must enter their 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 / Smart Cards

• Codes based on a cryptographic key
  – Token manufacturer also knows the key

• What if there is a breach?
WebAuthn (2019)

FIDO2 BRINGS SIMPLER, STRONGER AUTHENTICATION TO WEB BROWSERS

FIDO AUTHENTICATION: THE NEW GOLD STANDARD

Protes 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
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