19. Authentication and Access Control Part 1

Blase Ur and David Cash
February 24th, 2021
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 ≠ Access Control
  - *Authorization* is deciding whether an entity should have access to a given resource
  - *Access control* lists / policies
- **Principal**: legitimate owner of an identity
- **Claimant**: entity trying to be authenticated
Authentication Use Cases

• Explicit authentication
  – Single-factor authentication
  – Multi-factor authentication (e.g., with Duo)

• Implicit authentication
  – Continuous authentication

• Risk-based authentication: vary auth requirements based on estimated risk
How We Authenticate (1/3)

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

• Something you have
  – Private key (of a public-private key pair)
  – Hardware device (often with a key/seed)
  – Phone (running particular software)
  – Token (e.g., hex string stored in a cookie)
How We Authenticate (2/3)

• Something you are
  – Biometrics (e.g., iris or fingerprint)

• Somewhere you are
  – Location-limited channels
  – IP address
How We Authenticate (3/3)

• 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?

• Easy to use
• Easy to deploy
• Nothing to carry
• No “silver-bullet” alternative
## Why Are Passwords So Prevalent?

<table>
<thead>
<tr>
<th>Memorywise-Effortless</th>
<th>Scalable-for-Users</th>
<th>Nothing-to-Carry</th>
<th>Physically-Effortless</th>
<th>Easy-to-Learn</th>
<th>Efficient-to-Use</th>
<th>Infrequent-Errors</th>
<th>Easy-Recovery-from-Loss</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Usability

<table>
<thead>
<tr>
<th>Accessible</th>
<th>Negligible-Cost-per-User</th>
<th>Server-Compatable</th>
<th>Browser-Compatable</th>
<th>Mature</th>
<th>Non-Proprietary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Deployability

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

## Security

Why Are Passwords So Prevalent?


<table>
<thead>
<tr>
<th>Category</th>
<th>Scheme</th>
<th>Described in section</th>
<th>Usability</th>
<th>Deployability</th>
<th>Security</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Incumbent)</td>
<td>Web passwords</td>
<td>III [13]</td>
<td>★★★★☆☆☆☆</td>
<td>★★★★☆☆☆☆</td>
<td>★★★★☆☆☆☆</td>
</tr>
<tr>
<td>Password managers</td>
<td>Firefox</td>
<td>IV-A [22]</td>
<td>★★★★★★☆☆</td>
<td>★★★★★☆★</td>
<td>★★★★☆★☆☆</td>
</tr>
<tr>
<td></td>
<td>LastPass</td>
<td>[43]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proxy</td>
<td>URRSA</td>
<td>IV-B [5]</td>
<td>★☆☆★☆</td>
<td>★★★★★★☆★</td>
<td>★★★★★☆★☆</td>
</tr>
<tr>
<td></td>
<td>Imposter</td>
<td>[23]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Federated</td>
<td>OpenID</td>
<td>IV-C [27]</td>
<td>★★★★★★☆★</td>
<td>★★★★★★☆★</td>
<td>★★★★★☆★☆</td>
</tr>
<tr>
<td></td>
<td>Microsoft Passport</td>
<td>[43]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Facebook Connect</td>
<td>[44]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>BrowserID</td>
<td>[45]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OTP over email</td>
<td>[46]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graphical</td>
<td>PCCP</td>
<td>IV-D [7]</td>
<td>★☆☆★☆</td>
<td>★★★★★★☆★</td>
<td>★★★★★☆★☆</td>
</tr>
<tr>
<td></td>
<td>PassGo</td>
<td>[47]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cognitive</td>
<td>GrIDSure (original)</td>
<td>[30]</td>
<td>★☆☆★☆</td>
<td>★★★★★★☆★</td>
<td>★★★★★☆★☆</td>
</tr>
<tr>
<td></td>
<td>Weinshall</td>
<td>[48]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hopper Blum</td>
<td>[49]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Word Association</td>
<td>[50]</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Paper tokens</td>
<td>OTPW</td>
<td>IV-F [33]</td>
<td>★★★★★☆☆☆☆</td>
<td>★★★★★☆☆☆☆</td>
<td>★★★★★☆☆☆☆</td>
</tr>
<tr>
<td></td>
<td>S/KEY</td>
<td>[32]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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

- LinkedIn
- Ashley Madison
- Sony
- Gawker
- 000webhost.com
- Yahoo!
- Stratfor
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 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

• Both hash and salt passwords
Data-Driven Statistical Attacks

• (2009) 32 million passwords: rockyou

• (2016) 117 million passwords: LinkedIn

• (2017) 3 billion passwords: Yahoo!

• Total: >10 billion passwords stolen from >500 services
Have I Been Pwned (HIBP)

Largest breaches:
- 772,904,991 Collection #1 accounts
- 763,117,241 Verifications.io accounts
- 711,477,622 Onliner Spambot accounts
- 622,161,052 Data Enrichment Exposure From PDL Customer accounts
- 593,427,119 Exploit.In accounts
- 457,962,538 Anti Public Combo List accounts
- 393,430,309 River City Media Spam List accounts
- 359,420,698 MySpace accounts
- 268,765,495 Wattpad accounts
- 234,842,089 NetEase accounts

Recently added breaches:
- 645,786 Filmai.in accounts
- 10,585 NurseryCam accounts
- 358,822 People's Energy accounts
- 1,436,435 NetGalley accounts
- 1,110,156 CityBee accounts
- 2,481,121 Ge.tt accounts
- 1,047,200 StoryBird accounts
- 1,906,808 Pixlr accounts
- 1,422,717 MeetMindful accounts
- 2,811,929 Bonobos accounts
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
Understanding Users’ Password Behaviors
Some Ways to Understand Users

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