

6. Authentication

Part 1



Blase Ur and David Cash
January 18th, 2023
CMSC 23200 / 33250



THE UNIVERSITY OF
CHICAGO

Who Am I?

- David Cash
 - Distinguished cryptographer
 - Fan of rare plants
 - Campuswire expert

Or Am I?

How (and why) do we
authenticate users?

Authentication in the Abstract

- **Principal:** legitimate owner of an identity
- **Claimant:** entity trying to be authenticated
- Verify that **people** or **things** (e.g., a server) are who they claim to be, or maybe that the claimant has some **attribute**
- Authentication $\neq$ Authorization $\neq$ Access Control
 - *Authorization* is deciding whether an entity should have access to a given resource
 - *Access control* lists / policies

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 (1/2)

- 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 (2/2)

- Something you are
 - Biometrics (e.g., iris or fingerprint)
- Somewhere you are
 - Location-limited channels
 - IP address
- Someone you know (social authentication)
 - Someone vouches for you
- Some system vouches for you
 - Single sign-on (e.g., UChicago shib/Okta)
 - PKI Certificate Authorities

[illegible]

Why Are Passwords So Prevalent?

- Easy to use
- Easy to deploy
- Nothing to carry
- No “silver-bullet” alternative

Why Are Passwords So Prevalent?

<i>Memorywise-Effortless</i>	Usability
<i>Scalable-for-Users</i>	
<i>Nothing-to-Carry</i>	
<i>Physically-Effortless</i>	
<i>Easy-to-Learn</i>	
<i>Efficient-to-Use</i>	
<i>Infrequent-Errors</i>	
<i>Easy-Recovery-from-Loss</i>	
<i>Accessible</i>	Deployability
<i>Negligible-Cost-per-User</i>	
<i>Server-Compatible</i>	
<i>Browser-Compatible</i>	
<i>Mature</i>	
<i>Non-Proprietary</i>	
<i>Resilient-to-Physical-Observation</i>	Security
<i>Resilient-to-Targeted-Impersonation</i>	
<i>Resilient-to-Throttled-Guessing</i>	
<i>Resilient-to-Unthrottled-Guessing</i>	
<i>Resilient-to-Internal-Observation</i>	
<i>Resilient-to-Leaks-from-Other-Verifiers</i>	
<i>Resilient-to-Phishing</i>	
<i>Resilient-to-Theft</i>	
<i>No-Trusted-Third-Party</i>	
<i>Requiring-Explicit-Consent</i>	
<i>Unlinkable</i>	

Bonneau et al. "The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes," In *Proc. IEEE S&P*, 2012

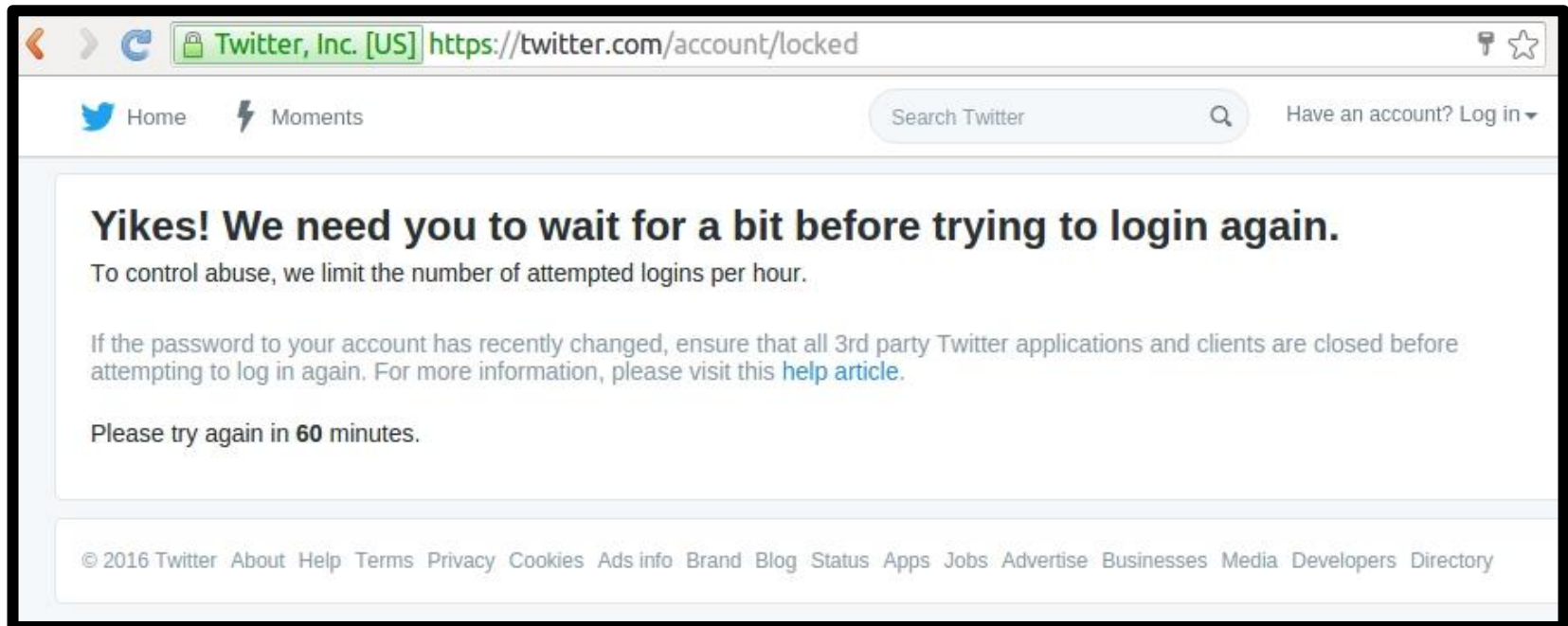
Why Are Passwords So Prevalent?

Category	Scheme	Described in section	Reference	Usability								Deployability						Security									
				Memorywise-Effortless	Scalable-for-Users	Nothing-to-Carry	Physically-Effortless	Easy-to-Learn	Efficient-to-Use	Infrequent-Errors	Easy-Recovery-from-Loss	Accessible	Negligible-Cost-per-User	Server-Compatible	Browser-Compatible	Mature	Non-Proprietary	Resilient-to-Physical-Observation	Resilient-to-Targeted-Impersonation	Resilient-to-Throttled-Guessing	Resilient-to-Unthrottled-Guessing	Resilient-to-Internal-Observation	Resilient-to-Leaks-from-Other-Verifiers	Resilient-to-Phishing	Resilient-to-Theft	No-Trusted-Third-Party	Requiring-Explicit-Consent
(Incumbent)	Web passwords	III	[13]		●		●	●	○	●	●	●	●	●	●	○							●	●	●	●	●
Password managers	Firefox	IV-A	[22]	●	●	○	●	●	●	●	●	■	●	●	●	●	●	○	○				●	●	●	●	●
	LastPass		[42]	●	●	○	●	●	●	●	●	○	●	●	●	●	●	○	○	○	○	○	●	●	●	●	●
Proxy	URRSA	IV-B	[5]	●		■		●	■	○	■	■	●	○	●	■	■	■	■	○			●	●	■	■	●
	Impostor		[23]	●	●	●		●	■	■	■	■	●	○	●	■	■	■	■	■	○		●	●	■	■	●
Federated	OpenID	IV-C	[27]	●	●	●	○	●	●	●	●	●	●	■	●	●	○	○	○	○		●		●	■	■	■
	Microsoft Passport		[43]	●	●	●	○	●	●	●	●	●	■	■	■	●	●	○	○	○	○		●		●	■	■
	Facebook Connect		[44]	●	●	●	○	●	●	●	●	●	■	■	■	●	●	○	○	○	○		●		●	■	■
	BrowserID		[45]	●	●	●	○	●	●	●	●	●	■	○	●	■	■	■	■	■	○		●		●	■	■
	OTP over email		[46]	●	●	●		●	■	■	■	■	■	■	■	■	■	■	○	○	○	○		●	●	■	■
Graphical	PCCP	IV-D	[7]		●		●	○	○	■	■	■	■	■	●	●	●	○	○			●	●	●	●	●	●
	PassGo		[47]		●		●	○	○	■	■	■	■	○	●	●	●	○	○			●	●	●	●	●	●
Cognitive	GrIDsure (original)	IV-E	[30]		●		●	○	○	■	■	■	■	■	■	■	■		●	●				●	●	●	●
	Weinshall		[48]		●		■	■	■	■	■	■	■	■	■	■	■	○	○				●	●	●	●	●
	Hopper Blum		[49]		●		■	■	■	■	■	■	■	■	■	■	■	○	○				●	●	●	●	●
	Word Association		[50]		●		●	○	○	■	■	■	■	■	■	■	■	■	■				●	●	●	●	●
Paper tokens	OTPW	IV-F	[33]		■		●	■	■	■	■	■	■	■	■	■	■	●	●	●	●	●	●	●	●	●	●
	S/KEY		[32]	●	■		●	■	○	●	■	■	■	■	■	■	■	■	■	■	○	■	●	●	●	●	●

Bonneau et al. "The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes," In *Proc. IEEE S&P*, 2012

Attacks Against Passwords

- **Online attack (web)**
 - Try passwords on a live system
 - Usually rate-limited



Attacks Against Passwords

- **Online attack** (web)
 - Try passwords on a live system
 - Usually rate-limited
- Authenticating to a device is often similarly rate-limited (e.g., iPhone PIN) using secure hardware

Attacks Against Passwords

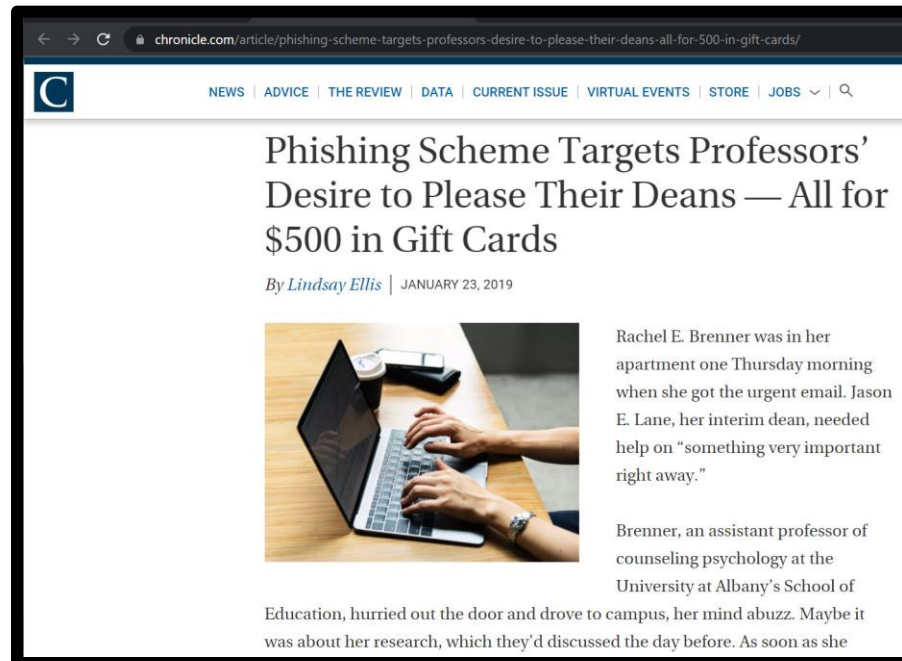
- **Offline attack (web)**
 - Try to guess passwords from the password store / password database

Attacks Against Passwords

- **Offline attack (web)**
 - Try to guess passwords from the password store / password database
- Attacking a file encrypted using a key derived from a password (e.g., with PBKDF2) is similar

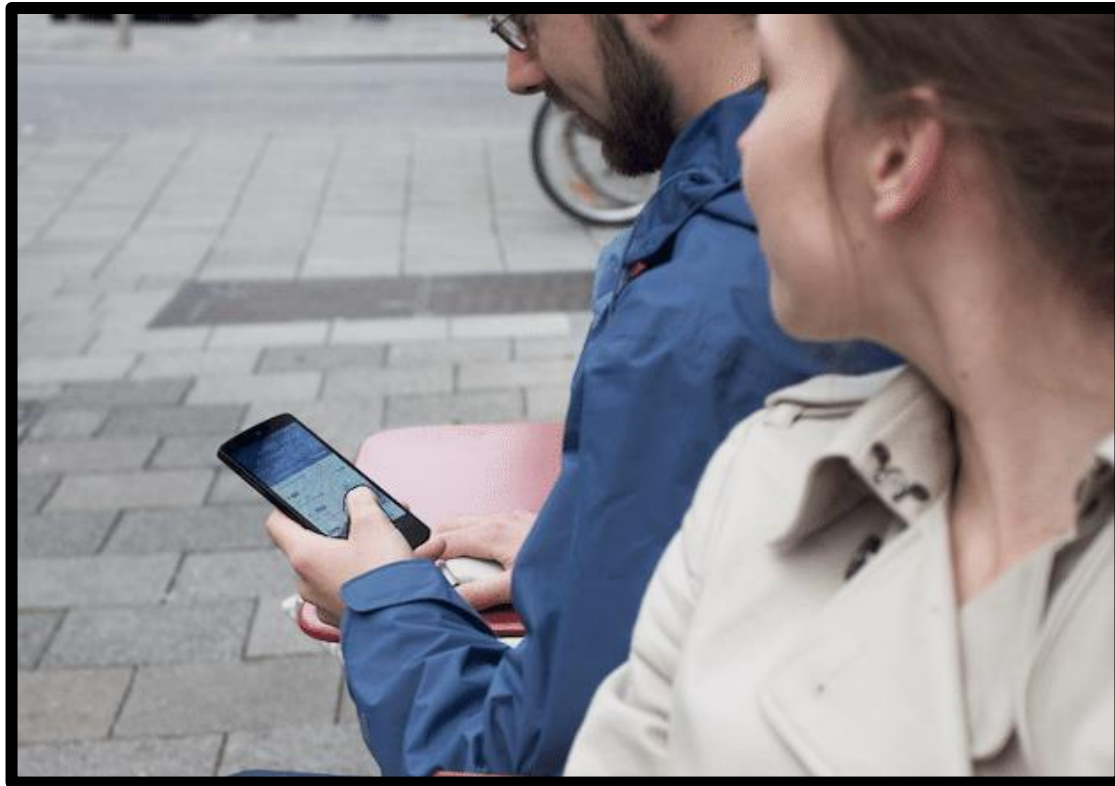
Attacks Against Passwords

- **Phishing** attack: try to trick the user into giving their credentials to you, believing that you are the legitimate system
 - **Spear phishing**: targeted to the recipient



Attacks Against Passwords

- **Shoulder surfing:** looking at someone else entering their credentials



Storing Passwords

- Hash function: one-way function
 - Traditionally designed for efficiency (e.g., MD5, SHA-2), but don't ever use those!
 - Use password-specific hash functions (e.g., bcrypt, scrypt, Argon2)

Hashing on NVIDIA RTX 3090

- Hashcat benchmarks
- MD5: ~ 60 billion / second
- SHA-1: ~ 20 billion / second
- UNIX md5crypt: ~ 20 million / second
- NTLM: ~ 100 billion / second
- SHA-2 (256): ~ 8 billion / second
- bcrypt (32 iterations): ~ 100,000 / second
- scrypt (16384 iterations): ~ 4,000 / second

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
 - Increases the attacker's work proportional to the number of accounts
- **Pepper**: secret salt (relatively uncommon)
- Both **salt** and **hash** passwords

Typical (Web) Account Creation

- User sends **username** and desired **password** over an encrypted tunnel
- Server validates username (e.g., does it exist in the system?) and password (e.g., does it meet composition requirements?)
- Server generates a random salt
 - Think about how long the salt should be!
- Server stores **username**, **salt**, and **hash(password|salt)** in database

Typical (Web) Authentication

- User sends **username** and **password₀** over an encrypted tunnel
- Server looks up the salt and hash output associated with that username
- Server computes $\text{hash}(\text{password}_0|\text{salt})$
- If it matches the hash output in the database, typically send back auth token (long string attacker can't guess associated with that user's session)

Offline Attack (Revisited)

- Attacker compromises database
 - hash("Blase") =

`$2a$04$iHdEgkI681VdDMc3f7edau9phRwORvhYjqWAlb7hb4B5uFJO1g4zi`

\$ = delimiter

2a = bcrypt

04 = 2^4 iterations (cost)

iHdEgkI681VdDMc3f7edau = 16 bytes of salt (radix-64 encoded)

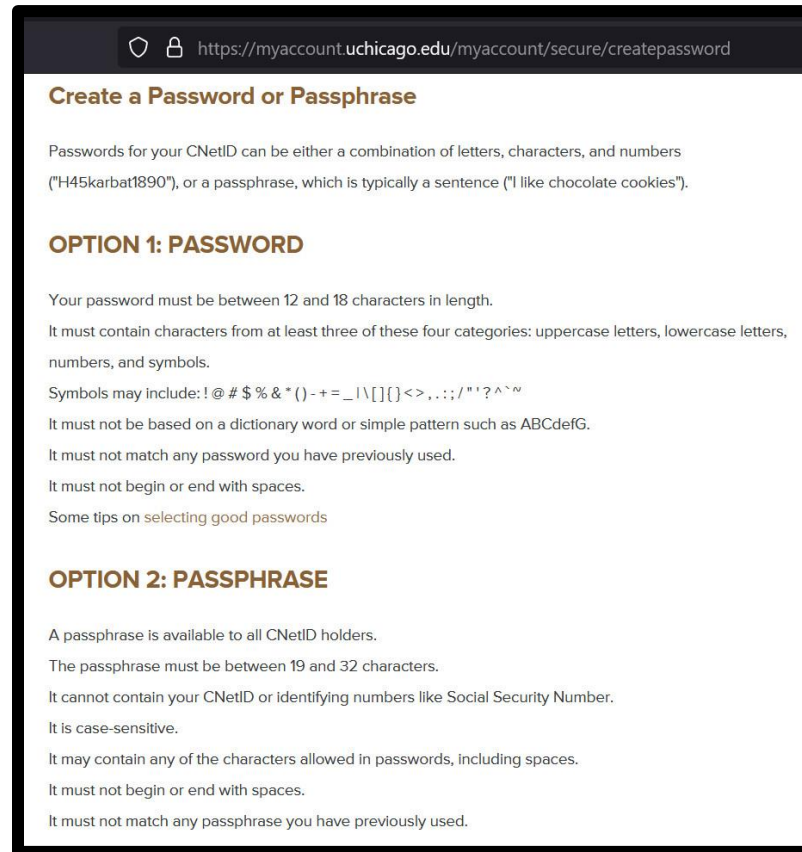
9phRwORvhYjqWAlb7hb4B5uFJO1g4zi = 24 bytes of hash output (radix-64 encoded)

- Attacker makes and hashes guesses
- Finds match → try on other sites
 - Password **reuse** is a core problem

Password Policies (Partial Attempt to Combat Attacks)

Password-Composition Rules

- Initial idea: increase the password space
- In practice: much more nuanced



<https://myaccount.uchicago.edu/myaccount/secure/createpassword>

Create a Password or Passphrase

Passwords for your CNetID can be either a combination of letters, characters, and numbers ("H45karbat1890"), or a passphrase, which is typically a sentence ("I like chocolate cookies").

OPTION 1: PASSWORD

Your password must be between 12 and 18 characters in length.

It must contain characters from at least three of these four categories: uppercase letters, lowercase letters, numbers, and symbols.

Symbols may include: ! @ # \$ % & * () - + = _ \ [] { } < > , . : ; / " ' ? ^ ` ~

It must not be based on a dictionary word or simple pattern such as ABCdefG.

It must not match any password you have previously used.

It must not begin or end with spaces.

[Some tips on selecting good passwords](#)

OPTION 2: PASSPHRASE

A passphrase is available to all CNetID holders.

The passphrase must be between 19 and 32 characters.

It cannot contain your CNetID or identifying numbers like Social Security Number.

It is case-sensitive.

It may contain any of the characters allowed in passwords, including spaces.

It must not begin or end with spaces.

It must not match any passphrase you have previously used.

Password Expiration

- Require password change every X days?



The image is a screenshot of a blog post from the Federal Trade Commission (FTC) website. The header is dark blue with the FTC seal on the left and navigation links: 'ABOUT THE FTC', 'NEWS & EVENTS', 'ENFORCEMENT', 'POLICY', and 'TIPS & A'. The main title is 'Time to rethink mandatory password changes' in a large, dark blue font. Below the title, it says 'By: Lorrie Cranor, Chief Technologist | Mar 2, 2016 10:55AM'. To the right of the title are social media share icons for Facebook, Twitter, and LinkedIn, with the text 'SHARE THIS PAGE' above them. Below the title and author information are 'TAGS: Authentication | Human-computer interaction | Passwords | Research'. The main body of the post contains two paragraphs of text. The first paragraph discusses data security as an evolving process and the FTC's advice to companies. The second paragraph starts with 'When people hear that I conduct research on making passwords more usable and secure', followed by a discussion of common user complaints and the author's response. The third paragraph begins with 'I go on to explain that there is a lot of evidence to suggest that users who are required to change their passwords frequently select weaker passwords to begin with, and then change them in predictable ways that attackers can guess easily. Unless there is reason to believe a password has been compromised or shared, requiring regular password changes may actually do more harm than good in some cases. (And even if a password has been compromised, changing the password may be ineffective, especially if other steps aren't taken to correct security problems.)'

 ABOUT THE FTC | NEWS & EVENTS | ENFORCEMENT | POLICY | TIPS & A

Time to rethink mandatory password changes

By: Lorrie Cranor, Chief Technologist | Mar 2, 2016 10:55AM

SHARE THIS PAGE   

TAGS: [Authentication](#) | [Human-computer interaction](#) | [Passwords](#) | [Research](#)

Data security is a process that evolves over time as new threats emerge and new countermeasures are developed. The FTC's longstanding advice to companies has been to conduct risk assessments, taking into account factors such as the sensitivity of information they collect and the availability of low-cost measures to mitigate risks. The FTC has also advised companies to keep abreast of security research and advice affecting their sector, as that advice may change. What was reasonable in 2006 may not be reasonable in 2016. This blog post provides a case study of why keeping up with security advice is important. It explores some age-old security advice that research suggests may not be providing as much protection as people previously thought.

When people hear that I conduct [research on making passwords more usable and secure](#), everyone has a story to tell and questions to ask. People complain about having so many passwords to remember and having to change them all so frequently. Often, they tell me their passwords (please, don't!) and ask me how strong they are. But my favorite question about passwords is: "How often should people change their passwords?" My answer usually surprises the audience: "Not as often as you might think."

I go on to explain that there is a lot of evidence to suggest that users who are required to change their passwords frequently select weaker passwords to begin with, and then change them in predictable ways that attackers can guess easily. Unless there is reason to believe a password has been compromised or shared, requiring regular password changes may actually do more harm than good in some cases. (And even if a password has been compromised, changing the password may be ineffective, especially if other steps aren't taken to correct security problems.)