9. Authentication
Part 1

Blase Ur and David Cash
January 31st, 2022
CMSC 23200 / 33250

THE UNIVERSITY OF CHICAGO

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 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 ≠ Authorization ≠ 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)
  - PKI Certificate Authorities


12345
123456789
password
iloveyou
princess
rockyou
1234567
12345678
abc123
nicole
daniel
babygirl
monkey
lovely
jessica
654321
michael
ashley
qwerty
111111
000000
michelle
tigger
sunshine
chocolate
password1
soccer
anthony
friends
butterfly
purple
angel
jordan
liverpool
justin
loveme
football
secret
andrea
carlos
jennifer
basketball
superman
hottie
bubbles
playboy
666666
tinkerbell
jasmine
brandon
elizabeth
flower
eminem
pretty
chelsea
barbie
angels
andrew
matthew
spongebob
whatever
dragon
robert
shadow
summer
melissa
softball
poohbear
hello
charlie
1234567890
computer
cookie
rangers
rainbow
christian
mickey
jesus
samantha
hannah
william
lovers
america
forever
beautiful
thomas
sakura
snoopy
patrick
chicken
heaven
pokemon
miguel

Why Are Passwords So Prevalent?

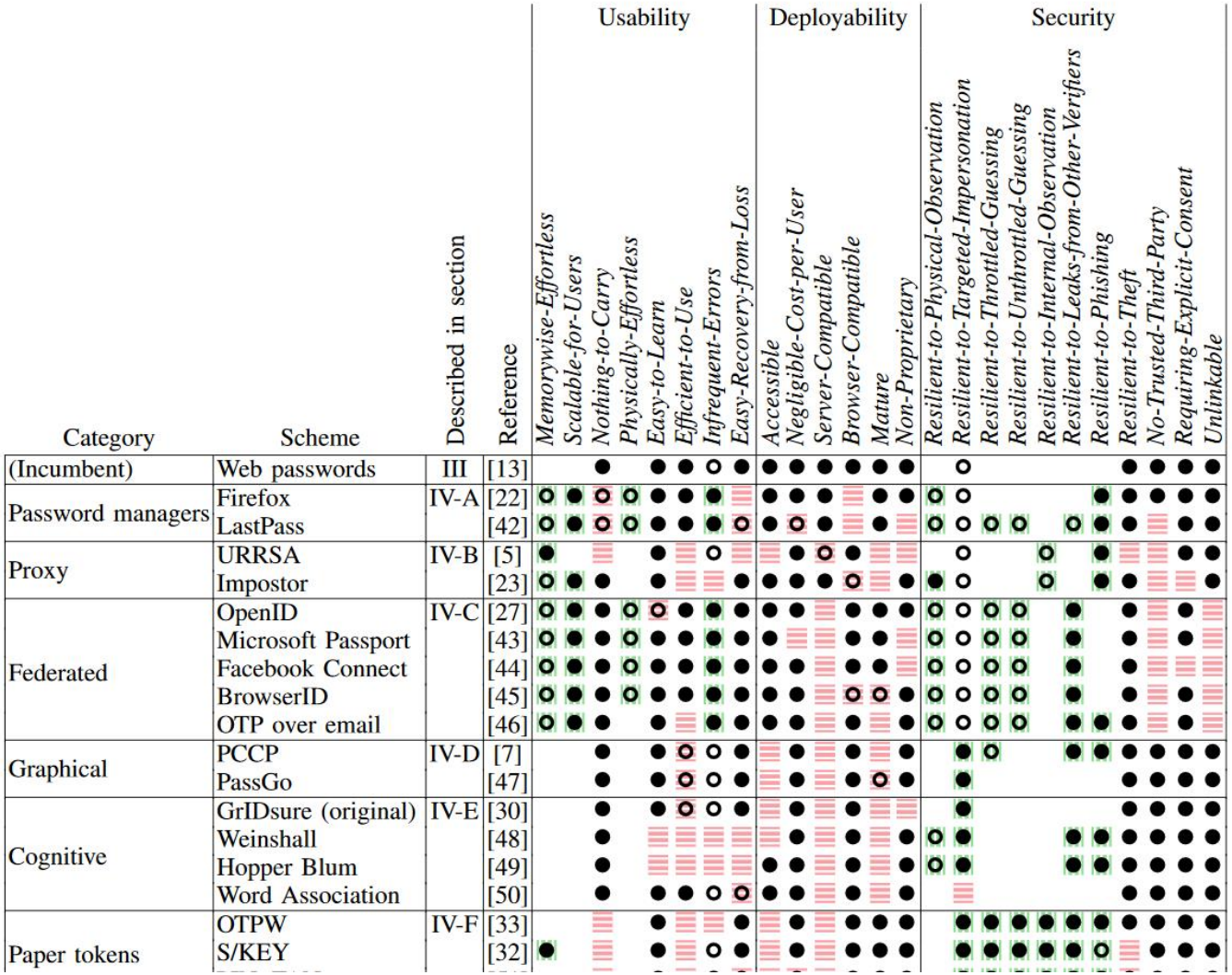
- Easy to use
- Easy to deploy
- Nothing to carry
- No “silver-bullet” alternative

Why Are Passwords So Prevalent?

Memorywise-Effortless	Usability
Scalable-for-Users	
Nothing-to-Carry	
Physically-Effortless	
Easy-to-Learn	
Efficient-to-Use	
Infrequent-Errors	
Easy-Recovery-from-Loss	
Accessible	Deployability
Negligible-Cost-per-User	
Server-Compatible	
Browser-Compatible	
Mature	
Non-Proprietary	
Resilient-to-Physical-Observation	Security
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	
Unlinkable	

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?

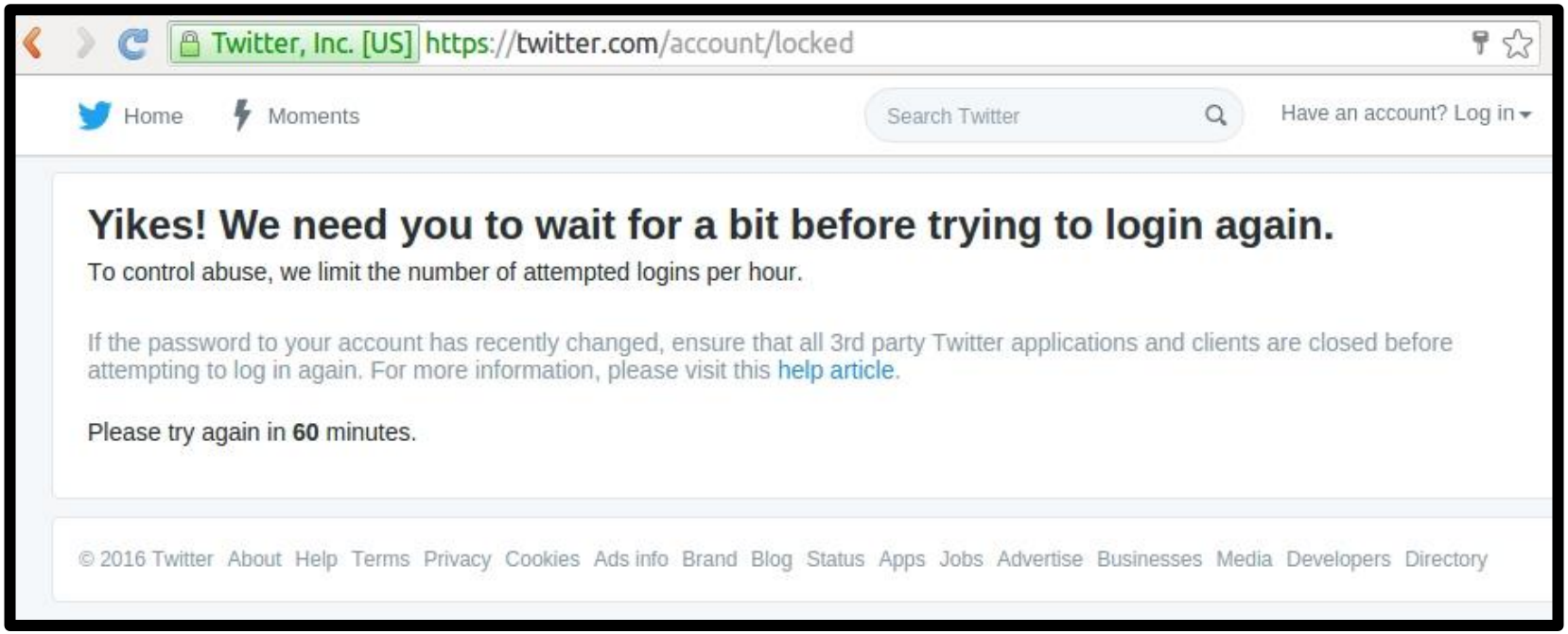


Category	Scheme	Described in section	Reference	Usability: *Memorywise-Effortless*	*Scalable-for-Users*	*Nothing-to-Carry*	*Physically-Effortless*	*Easy-to-Learn*	*Efficient-to-Use*	*Infrequent-Errors*	*Easy-Recovery-from-Loss*	Deployability: *Accessible*	*Negligible-Cost-per-User*	*Server-Compatible*	*Browser-Compatible*	*Mature*	*Non-Proprietary*	Security: *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*	*Unlinkable*
(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

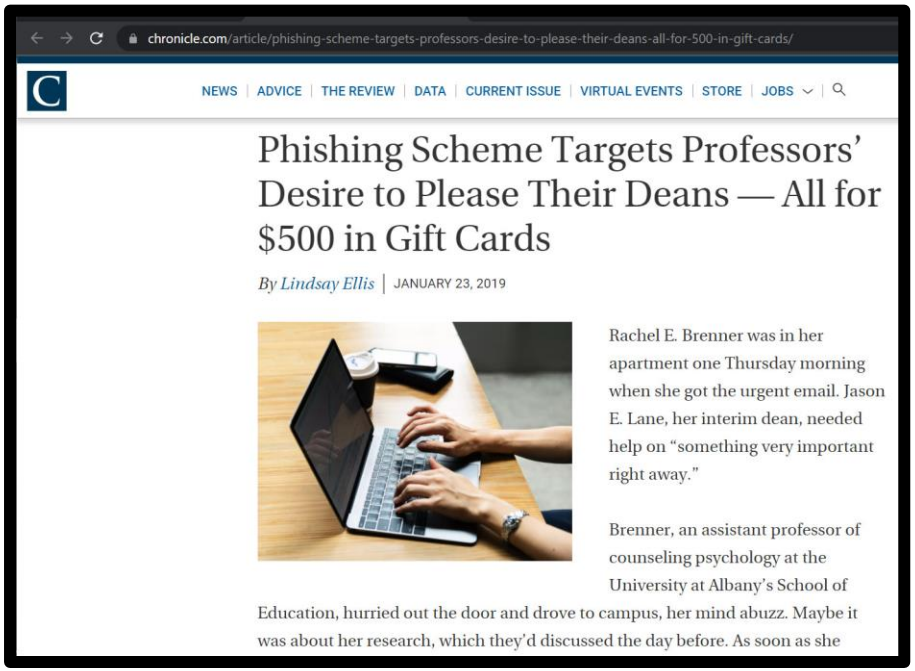
- **Offline attack** (web)
  - Try to guess passwords from the password store / password database

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

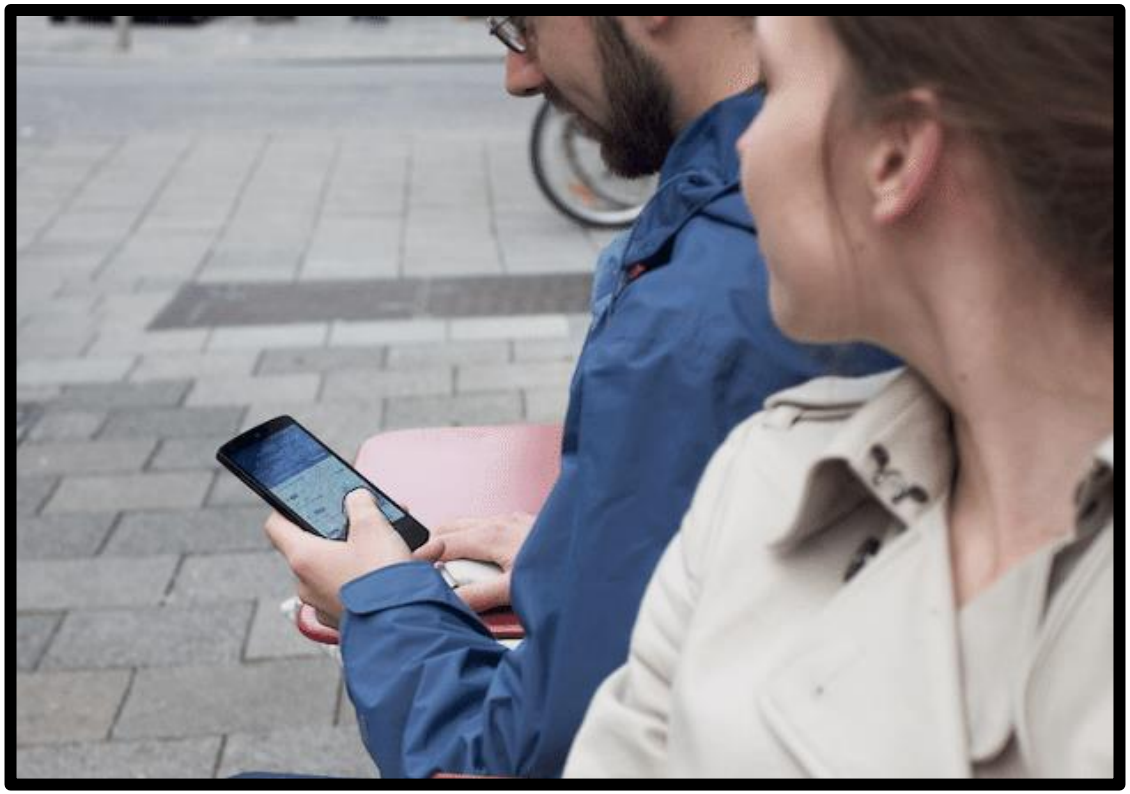


Photo from https://www.researchgate.net/figure/A-shoulder-surfing-situation-in-a-cafe_fig1_312490451

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

https://www.onlinehashcrack.com/tools-benchmark-hashcat-nvidia-rtx-3090.php

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 **password$_0$** over an encrypted tunnel
- Server looks up the salt and hash output associated with that username
- Server computes hash(password$_0$|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$04$iHdEgkI681VdDMc3f7edau9phRwORvhYjqWAIb7hb4B5uFJO1g4zi`

$ = delimiter

2a = bcrypt

04 = 2^4 iterations (cost)

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

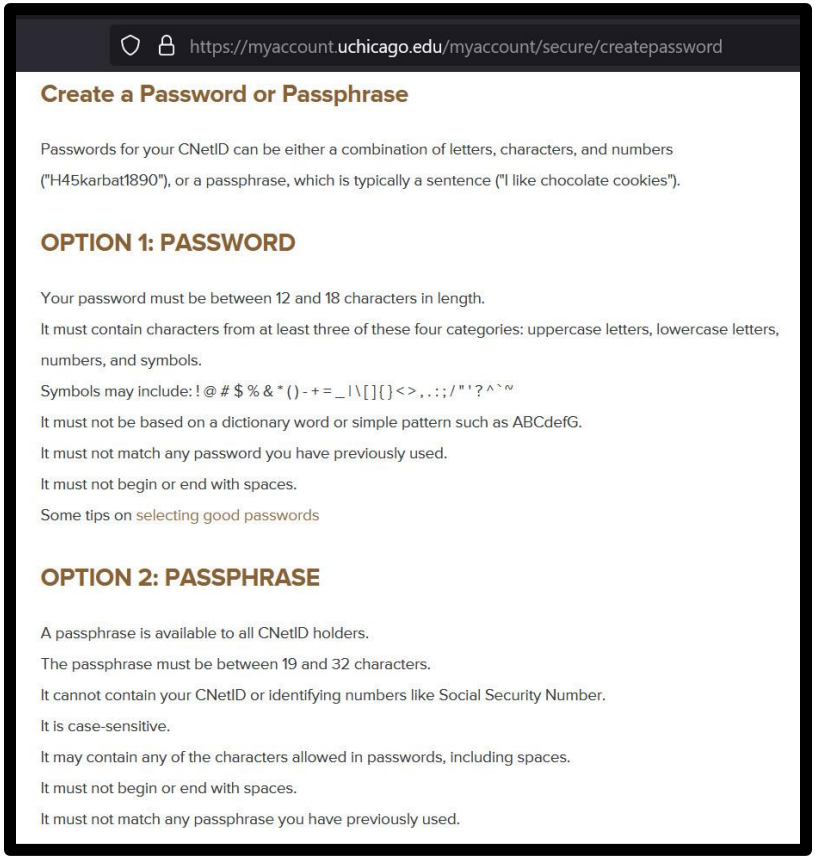
9phRwORvhYjqWAIb7hb4B5uFJO1g4zi = 24 bytes of hash output (radix-64 encoded)

- 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



Password Expiration

- Require password change every X days?

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.)