19. Authentication and Access Control Part 1

Blase Ur and David Cash

February 24th, 2021

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 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


123456789
12345
password
iloveyou
princess
rockyou
1234567
12345678
abc123
nicole
daniel
babygirl
monkey
lovely
jessica
654321
michael
ashley
qwerty
111111
iloveu
000000
michelle
tigger
sunshine
chocolate
password1
soccer
anthony
friends
butterfly
purple
angel
jordan
liverpool
justin
loveme
basketball
football
superman
playboy
secret
jasmine
tinkerbell
andrea
flower
carlos
eminem
jennifer
bubbles
hottie
cookie
dragon
pretty
chelsea
barbie
andrew
angels
shadow
elizabeth
brandon
matthew
softball
whatever
robert
charlie
spongebob
forever
lovers
beautiful
america
rangers
hannah
samantha
melissa
summer
sakura
princess1
1234567890
666666
jesus
mickey
poohbear
patrick
hello

Why Are Passwords So Prevalent?

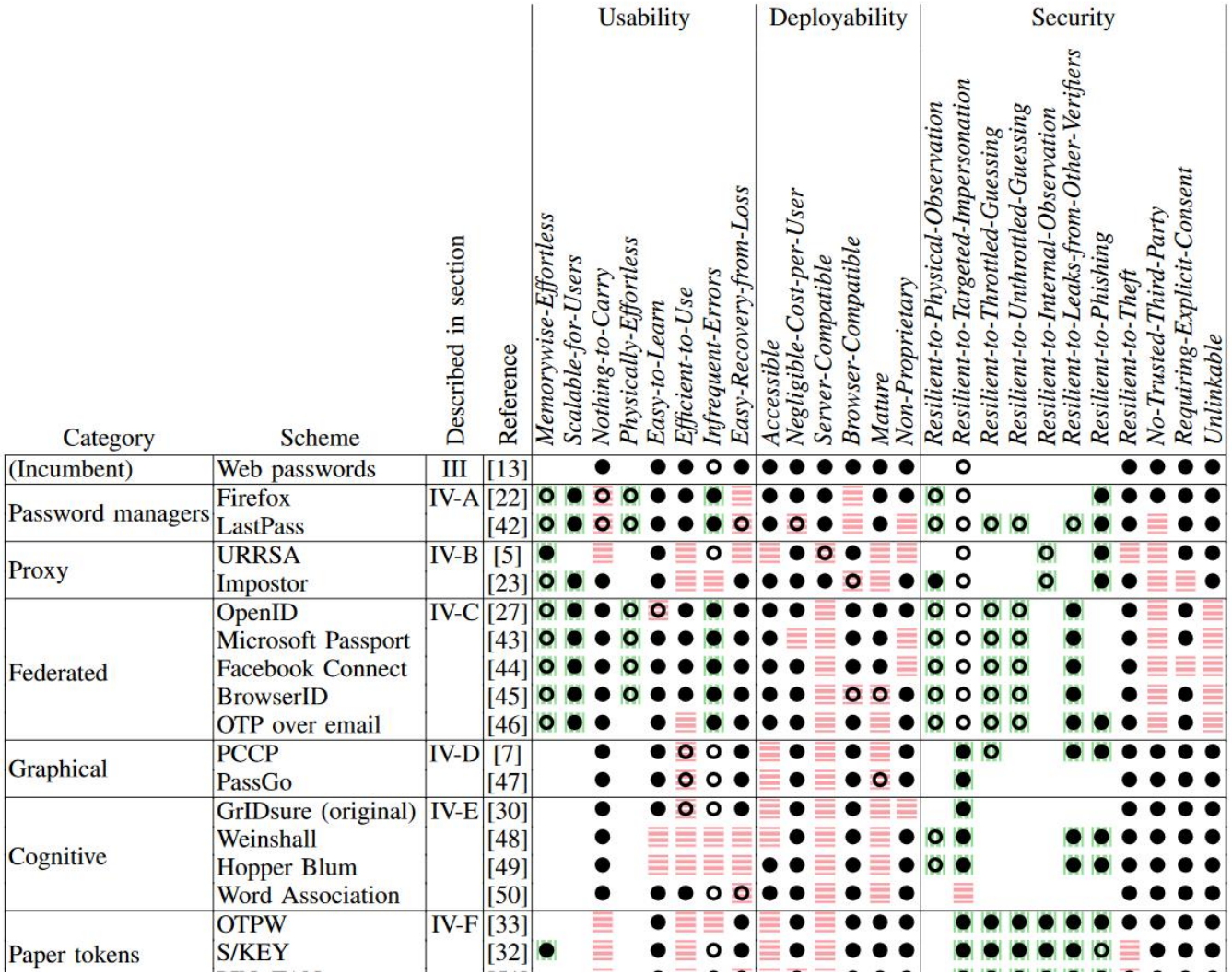
- Easy to use
- Easy to deploy
- Nothing to carry
- No “silver-bullet” alternative

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

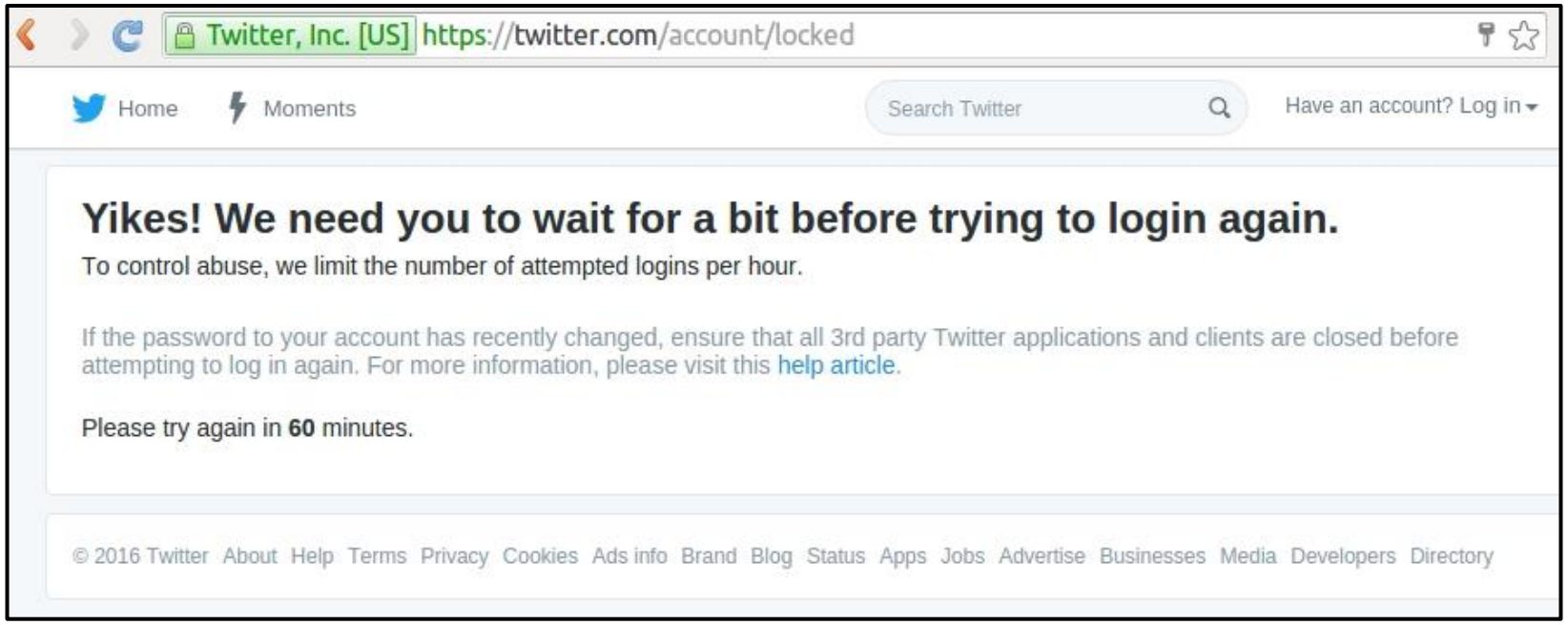
Bonneau et al. “The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes,” In *Proc. IEEE S&P*, 2012

Attacks Against Passwords

- Online attack
  - 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

Adobe

SONY

Domino's Pizza

GAWKER

000webhost.com
better than paid hosting

YAHOO!

STRATFOR
GLOBAL INTELLIGENCE

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)

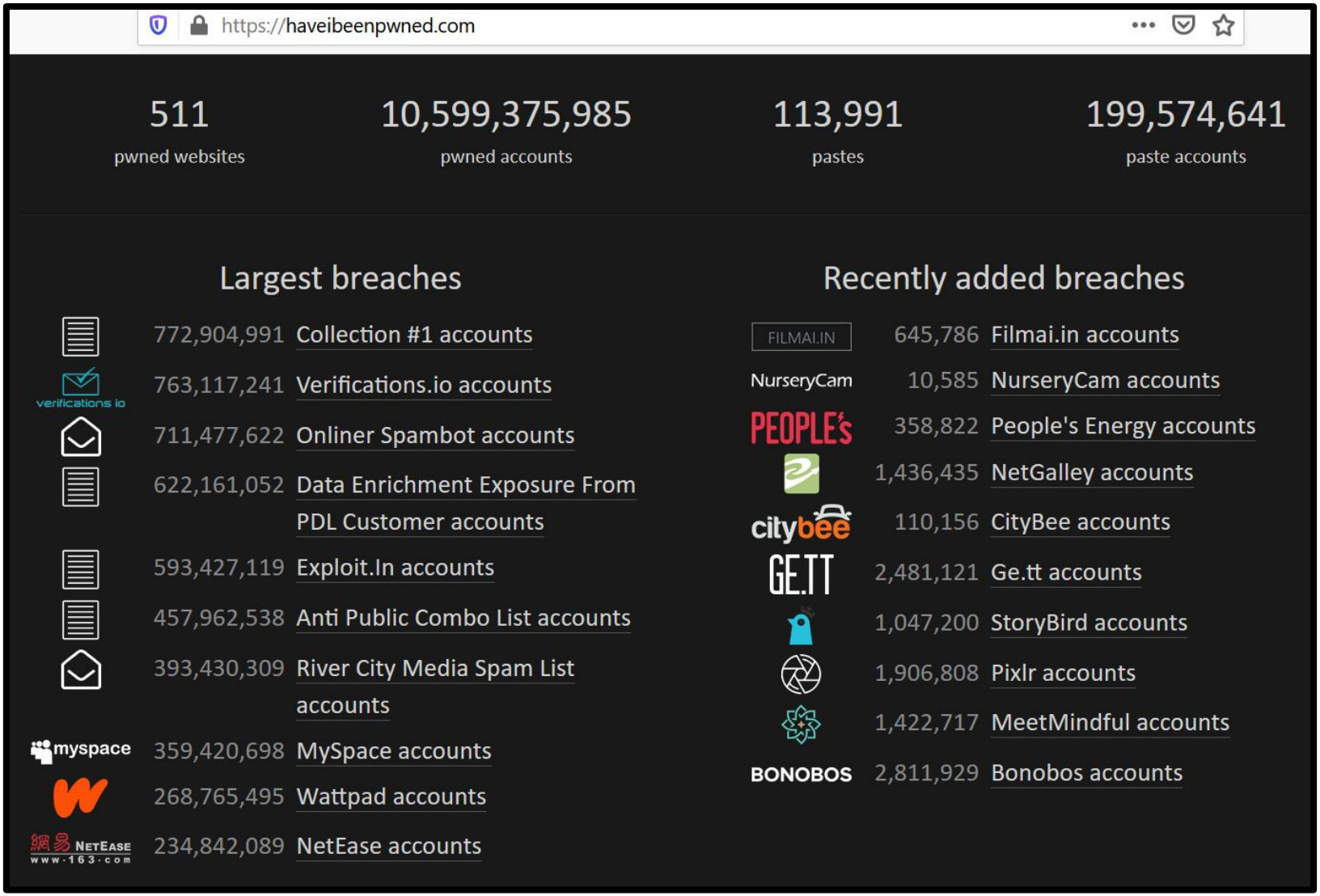
Storing Passwords

- 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)



Offline Attack

- Attacker compromises database
  - hash(“Blase”) =

`$2a$04$iHdEgkI681VdDMc3f7edau9phRwORvhYjqWAIb7hb4B5uFJO1g4zi`

- Attacker makes and hashes guesses
- Finds match → 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