

EECS 388

M Introduction to Computer Security

Lecture 14:
**Authentication and
Passwords**

October 10, 2024
Prof. Chen





Authentication

Authentication is the act of confirming the truth of an attribute of a single piece of data claimed true by an entity.

In security, we frequently authenticate the **identity** of a user or a machine to:

- enforce **access control policies**
- prevent **impersonation attacks**

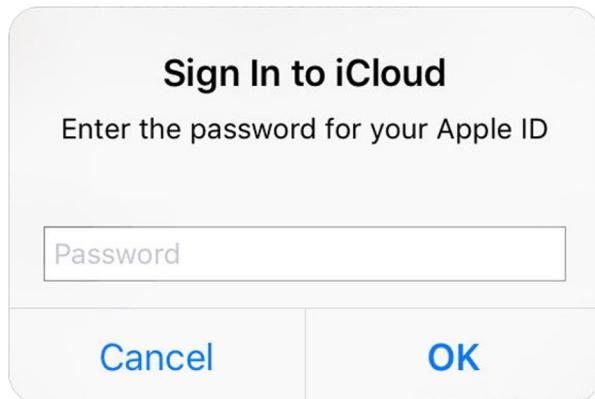
Examples:

- A **website** authenticates the identity of a **user** by demanding a **password**.
- A **phone** authenticates the identity of its **owner** by scanning a **fingerprint**.
- Your **browser** authenticates a **web server** you visit by checking a **certificate**.

Three Ways to Authenticate You



1. Something you **know**
password, PIN, secret key



2. Something you **have**
phone, security token, ID card



3. Something you **are**
biometrics



Passwords



Username: honey

Password: *****

Passwords are a **ubiquitous** but
weak form of authentication, with
numerous **usability problems**.

Problem with Passwords #1

People are bad at choosing strong passwords.

123456	12345678	abc123	654321
123456789	12345	qwerty123	555555
qwerty	iloveyou	1q2w3e4r	mynoob
Password	111111	admin	7777777
1234567	123123	qwertyuiop	welcome

Is your password on this list?

25 most common passwords of 2019 (source: [SplashData](#))



Good Password Practices



As a user:

- Never reuse passwords.
 - Use a password manager and let it generate strong passwords for you whenever possible.
 - Otherwise, `$ openssl rand -base64 15` and write it down
- Use two-factor authentication (more on this later!).

As a developer:

- Prefer outsourcing sign-on (e.g., Google/Facebook/Github OAuth) to requiring users to invent yet another password for your site.
- Avoid restrictive password complexity or rotation policies, which have been shown to do more harm than good.

Attack: Online Password Guessing



Online password guessing attack

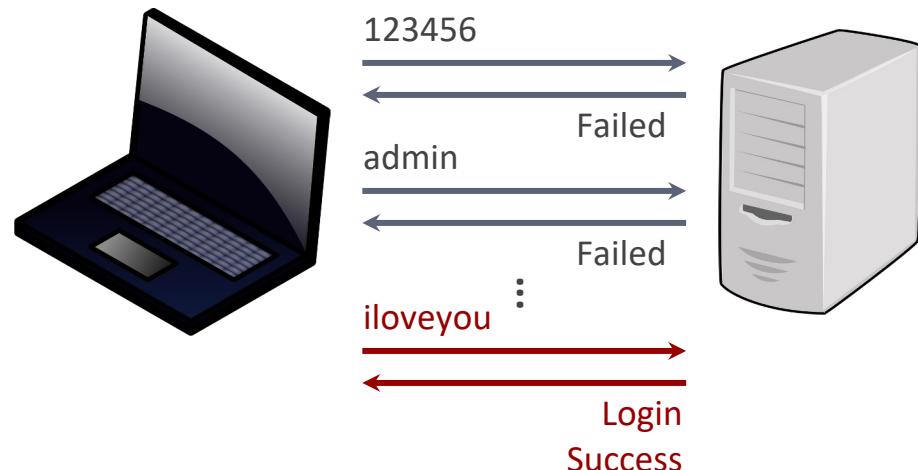
Submit guesses directly to website, try to log in.

Many guesses → single site

or Few guesses → many sites

Defenses:

- Lock account after n guesses?
- Rate-limit login attempts
- Anomaly detection
- Require solving a CAPTCHA



Defense: CAPTCHAs

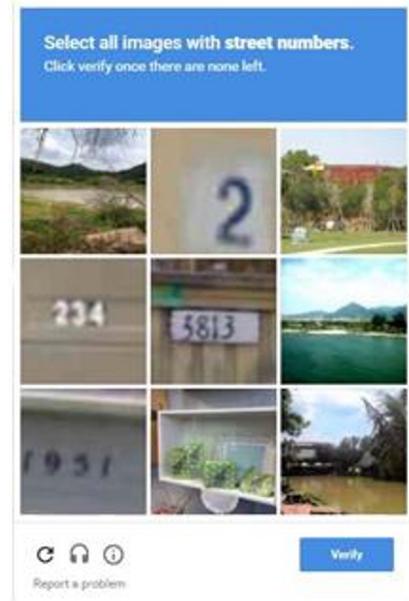
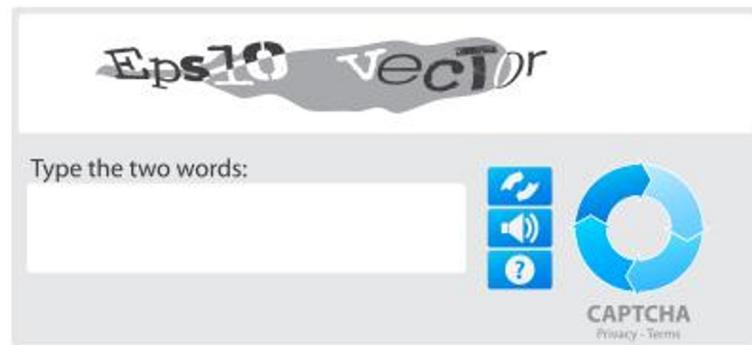
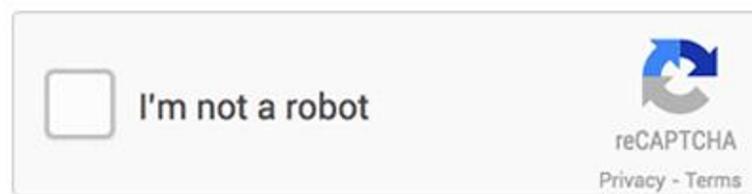


CAPTCHAs

Challenge that's easy for computers to generate, hard for them to solve, and easy for humans.

Used to make automated attacks or abuse more expensive.

"completely automated public Turing tests to tell computers and humans apart"



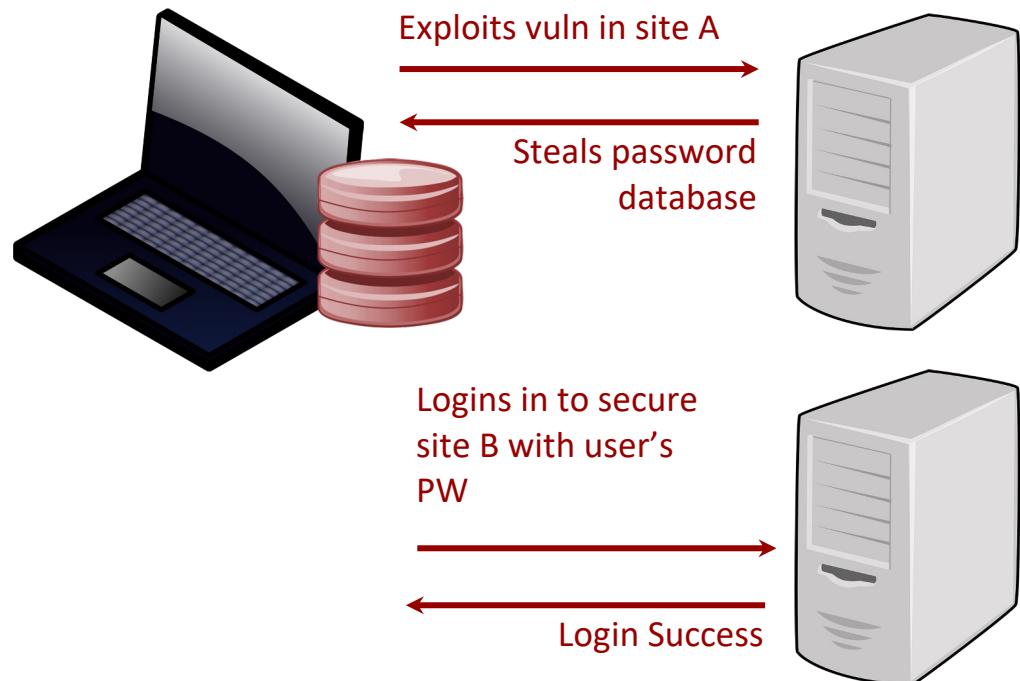
Defeating CAPTCHAs?

Problems with Passwords



People tend to use the same password for many purposes.

Why is this a problem?



Password Breaches



Companies face major financial liability, reputational loss, government sanctions.

Yahoo breach exposed more than a billion passwords, cost company \$350M in 2016.

';--have i been pwned?

Check if you have an account that has been compromised in a data breach

.....

pwned?

Oh no — pwned!

This password has been seen 5 times before

341

pwned websites

6,474,030,172

pwned accounts

89,440

pastes

100,143,728

paste accounts

Defending Against Password Breaches

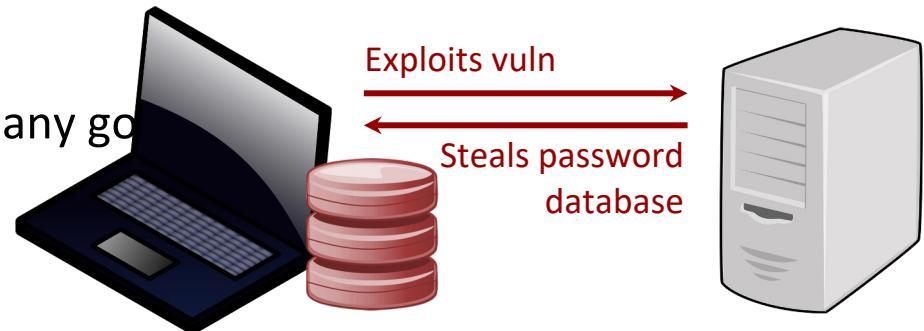


How should site store passwords to reduce risk?

Bad: Plaintext passwords

Pro: Easy.

Con: If leaked, company go



Bad: Encrypted passwords (Why?)

Better? Password hashes

Store $H(password)$ in database.

Compare $H(submitted_pw)$ to $H(password)$ to authenticate.

Pro: Site doesn't learn password.

Leaked database doesn't immediately reveal passwords.

Con: Identical passwords have identical hashes.

Attack: Offline Password Guessing

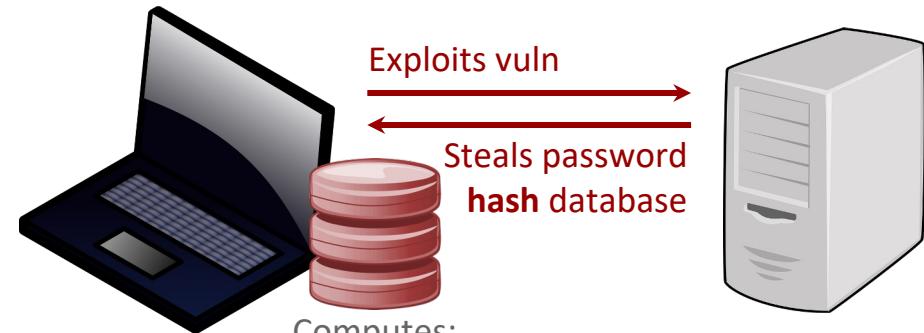


With password hashing, identical passwords result in identical hashes...

Offline password guessing

Attacker computes hashes of possible passwords and searches for them in stolen password hash database.

Brute force search of all passwords of length n
Takes exponential time.



Dictionary attack: Search corpus of previously leaked passwords and variants.

Can do massively parallel hashing on EC2, GPUs, or custom ASICs.

When searching huge dictionary against many hashes, vast speedups by using a precomputed data structure called a **Rainbow Table**.

Defending Against Offline Guessing



How should site store passwords to reduce risk?

Best: Salted password hashes

Randomly generate ***salt*** when password is set.

Store $\langle salt, H(salt \mid\mid password) \rangle$ in database.

Compare $H(salt \mid\mid submitted_pw)$ to $H(salt \mid\mid password)$.

Adversary can compromise the salt too! Is this a problem?

Pro: Leaked database doesn't reveal passwords.

Identical passwords have *different* salted hashes.

Attacker has to restart offline guessing for each stored password.

Which hash function to use? (Hint: Not anything you've seen so far.)

Something **slow**, and ideally **memory-hard**. (**Why?**)

Good Password Hash Functions: bcrypt*, scrypt, argon2

Salted password hashes

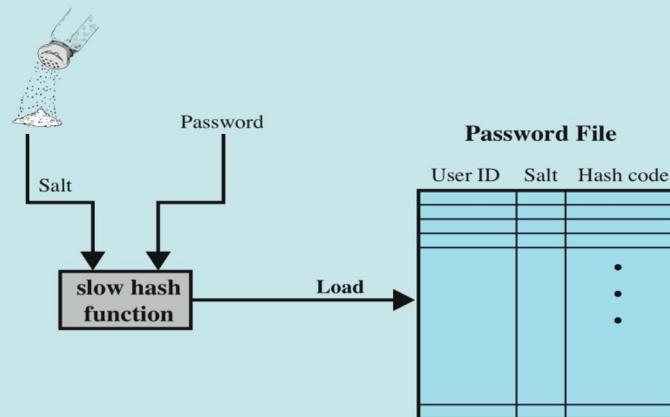
Common mistakes:

- Salt reuse

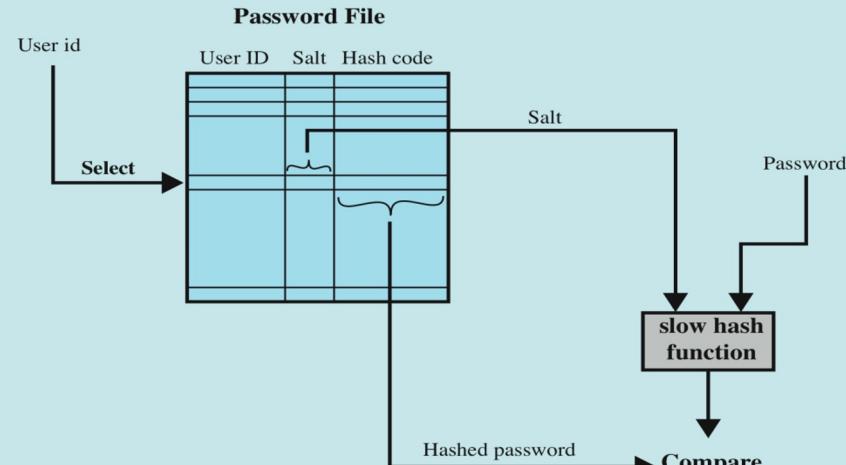
Should generate a new salt at random for each password.

- Short salt

Using a long salt ensures that a rainbow table for a database would be prohibitively large.



(a) Loading a new password



(b) Verifying a password

Defending Passwords in Hardware



An attacker who steals a traditional mobile device can perform offline guessing.

Newer devices store root password hash in a “**secure enclave**”, which functions like a tamper-resistant coprocessor.

Enclave checks passwords without exporting the password hash. Can enforce rate limit and maximum guesses.

Limits the attacker to online guessing.



Problems with Passwords

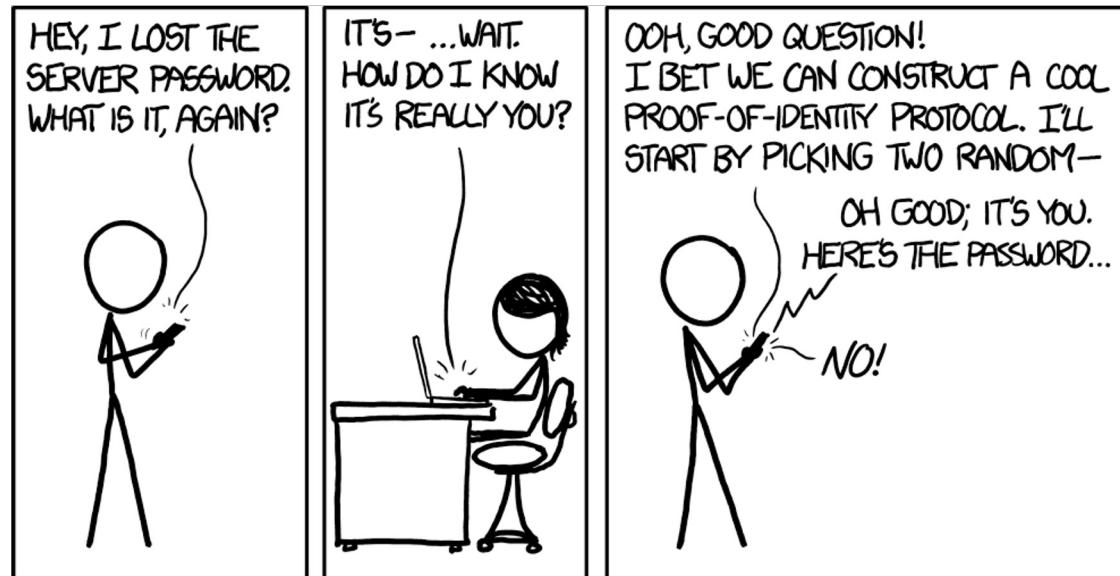


People often forget their passwords, so we still need another way to authenticate them.

Password reset mechanisms are necessary, but dangerous and often attacked. ([Discuss!](#))

A common means is **social engineering** (tricking people via psychological manipulation)

Security questions and **password hints** are often harmful and should be avoided.



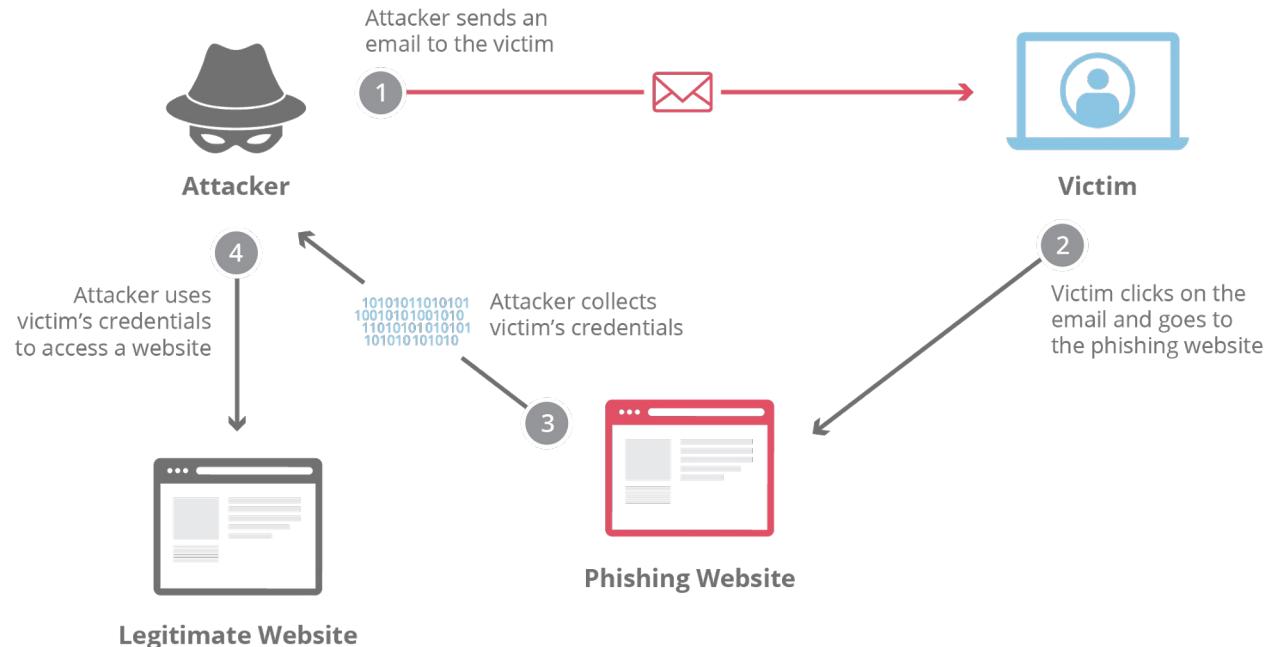
Problems with Passwords



People can be tricked into divulging their passwords.

Phishing attacks

are a form of social engineering that tricks the user into entering their password into a look-alike site controlled by the attacker.



Phishing and Spear Phishing



Typically, **phishing** is conducted by sending **forged emails** to large groups of potential victims.

Spear phishing involves more sophisticated forgery and tricks, tailored to an specific individual victim.

= WikiLeaks

Shop [Donate](#)

Search 

The Podesta Emails

WikiLeaks series on deals involving Hillary Clinton campaign Chairman John Podesta. Mr Podesta is a long-term associate of the Clintons and was President Bill Clinton's Chief of Staff from 1998 until 2001. Mr Podesta also owns the Podesta Group with his brother Tony, a major lobbying firm and is the Chair of the Center for American Progress (CAP), a Washington DC-based think tank.

[Read The Podesta Emails, Part 1: John Podesta and The Uranium One Story](#)



The Podesta
Emails

[Search by Terms in Email](#) [Search by Attached Filename](#) [Search by Email-ID](#)

Phishing and Spear Phishing



Typically, **phishing** is conducted by sending **forged emails** to large groups of potential victims.

Spear phishing involves more sophisticated forgery and tricks, tailored to an specific individual victim.



Someone has your password

Hi John

Someone just used your password to try to sign in to your Google Account
john.podesta@gmail.com.

Details:

Saturday, 19 March, 8:34:30 UTC
IP Address: 134.249.139.239
Location: Ukraine

Google stopped this sign-in attempt. You should change your password immediately.

CHANGE PASSWORD

Best,
The Gmail Team

Phishing and Spear Phishing



Typically, **phishing** is conducted by sending **forged emails** to large groups of potential victims.

Spear phishing involves more sophisticated forgery and tricks, tailored to an specific individual victim.

The screenshot shows a Google search result for "WikiLeaks". The top result is a link to a WikiLeaks page. The page displays an email message with the following header information:

```
*From:* Charles Delavan <cdelavan@hillaryclinton.com>  
*Date:* March 19, 2016 at 9:54:05 AM EDT  
*To:* Sara Latham <slatham@hillaryclinton.com>, Shane Hable <shable@hillaryclinton.com>  
*Subject:* *Re: Someone has your password*
```

The body of the email starts with "Sara," followed by a yellow-highlighted section of text:

This is a legitimate email. John needs to change his password immediately, and ensure that two-factor authentication is turned on his account.

Below this, another yellow-highlighted section of text reads:

He can go to this link: <https://myaccount.google.com/security> to do both. It is absolutely imperative that this is done ASAP.

Best Practice: Multi-Factor Authentication



Something you know (password)

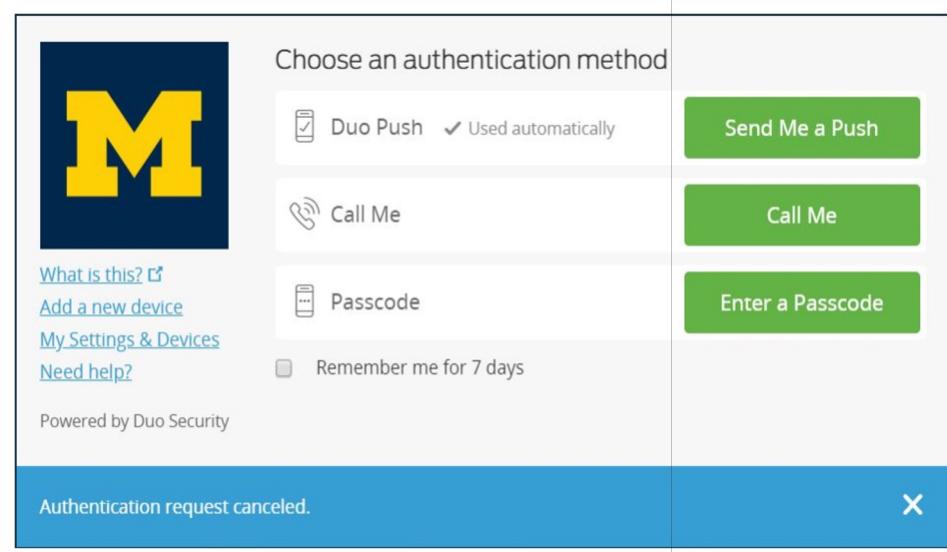
Something you have (phone/token)

Something you are (biometric)

Two-factor (2FA) or **multi-factor (MFA)** authentication use multiple of these to defend against stolen passwords.

Second factor must be:

- distinct from the password
- not just another password
- not computable from the password



Example: **Duo** (cloud-based 2FA)

Founded in A2 by CSE grads in 2009

Sold to Cisco for \$2.35B in 2018

Two-Factor Approaches



One-time passwords

“Prove knowledge” of secret k .

Counter-based: $\text{OTP}(c, k) := \text{HMAC}(k, c)$

Time-based: Use a time window index for c .

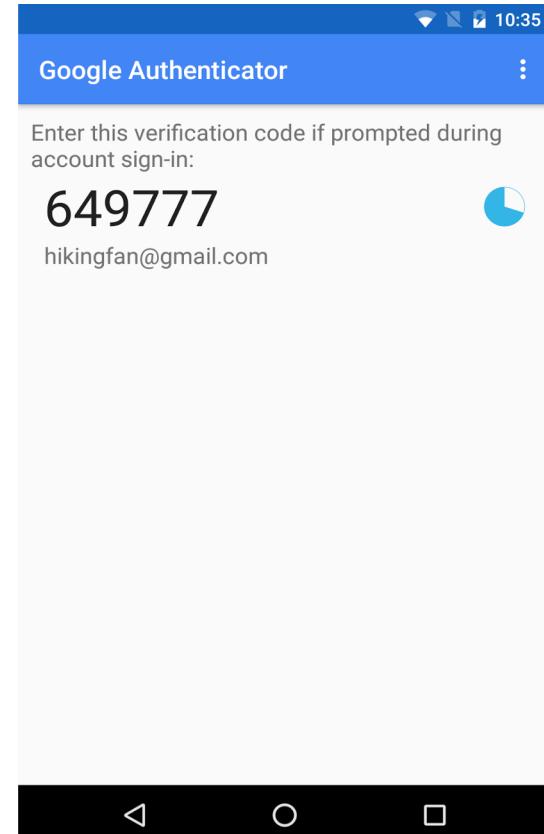
Cons: Susceptible to phishing

SMS/phone calls

Prove access to phone number.

Type in code sent over phone.

Cons: Susceptible to phishing and social engineering



Two-Factor Approaches



Universal 2nd Factor (U2F)

Open standard for authentication tokens.

USB or RFID device based on tamper-resistant hardware, containing unique secret key.

Performs challenge-response protocol with server.

Response bound to website origin,
so cannot be phished.

Cons: User needs to buy hardware.

Limited website support.



WebAuthn

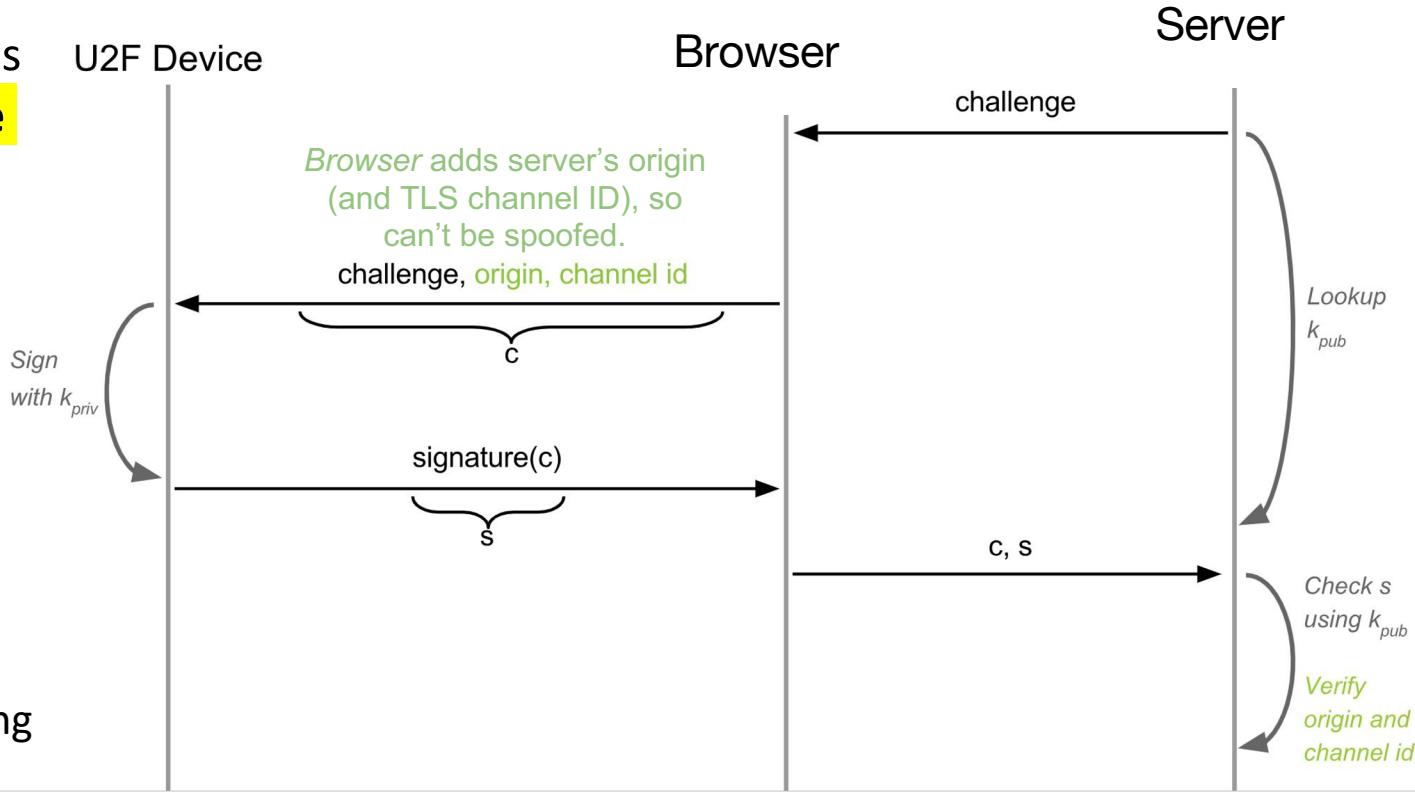


WebAuthn standard is a **challenge-response protocol** exposed to web applications.

Supports **U2F** (universal second factor) MFA devices.

Prevents **replay attacks** with random challenge.

Prevents **relay attacks** (e.g., phishing) by binding response to origin.



Biometrics



Biometrics measure something you are, **features of the body.**

Pros:

Can't be lost or lent

Inconvenient to spoof?

Cons:

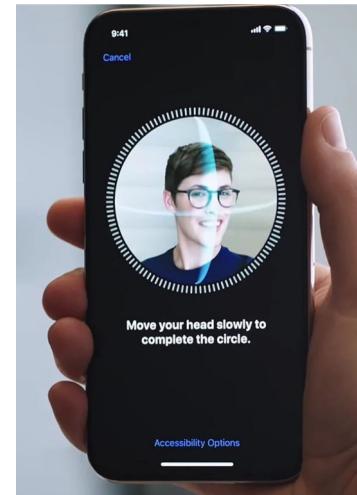
Needs trusted sensor

May not be unique

Can't be changed



Fingerprints



FaceID



Retinal Scanners

Defeating Biometrics

M



Coming Up



Reminders:

Networking Project due Thursday at 6 PM

Tuesday:

No lecture

Thursday

Midterm review