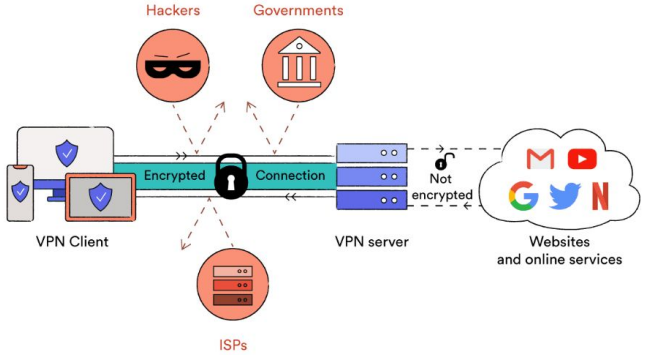
Defense: Virtual Private Networks

Issue: How to provide security for non-encrypted protocols across the public Internet?

A **virtual private network (VPN)** creates an encrypted channel that "tunnels" IP packets to a distant network location.



Provides confidentiality and integrity of packets *inside the tunnel*, authentication of endpoints

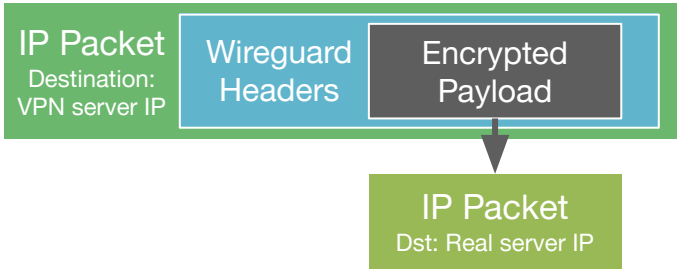
But…VPN can't protect packets traveling beyond endpoints (i.e., from VPN server to destination)

Broad applications of VPNs:

- Allow a remote device (e.g., a traveling employee) to access a corporate network
- Bridge two private networks via the Internet
- Provide Internet access from a distant ISP (to bypass local censorship or surveillance)

Common VPN protocols:

IPsec: complicated legacy protocol
OpenVPN: open-source, TLS-based
AnyConnect: proprietary, TLS-based
Wireguard: modern, high-performance



EECS 388

Introduction to Computer Security

Lecture 14:

Authentication and Passwords

October 12, 2023
Prof. Ensafi

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:

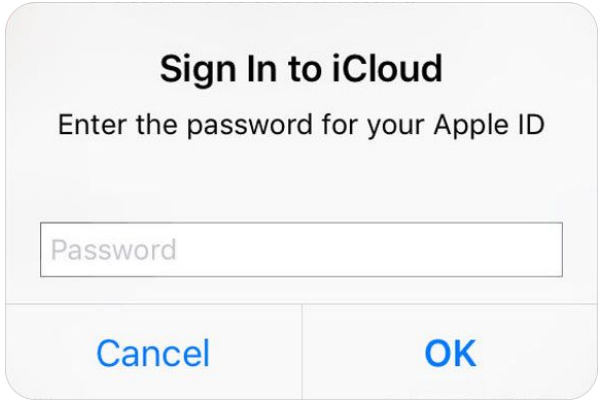
- A **website** authenticates the identity of a **user** by demanding a **password**.
- A **phone** authenticates the identity of its **owner** by scanning a **fingerprint**.
- Your **browser** authenticates a **web server** you visit by checking a **certificate**.

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: ensafi
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	888888
123456789	12345	qwerty123	555555	princess
qwerty	iloveyou	1q2w3e4r	mynoob	dragon
Password	111111	admin	7777777	password1
1234567	123123	qwertyuiop	welcome	123qwe

Is your password on this list?

25 most common passwords of 2019 (source: SplashData)

Good Password Practices

As a user:

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

As a developer:

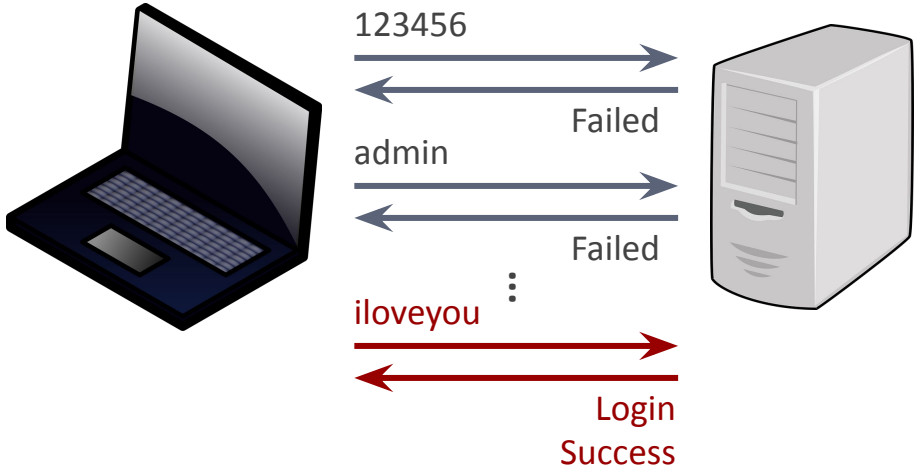
- Prefer outsourcing sign-on (e.g., Google/Facebook/Github OAuth) to requiring yet another password for your site.
- Avoid restrictive password complexity or rotation policies (they've been shown to do more harm than good).

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

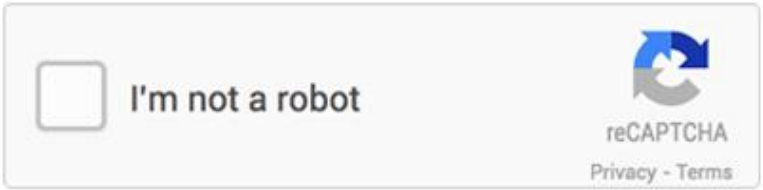
Defense: CAPTCHAs

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"

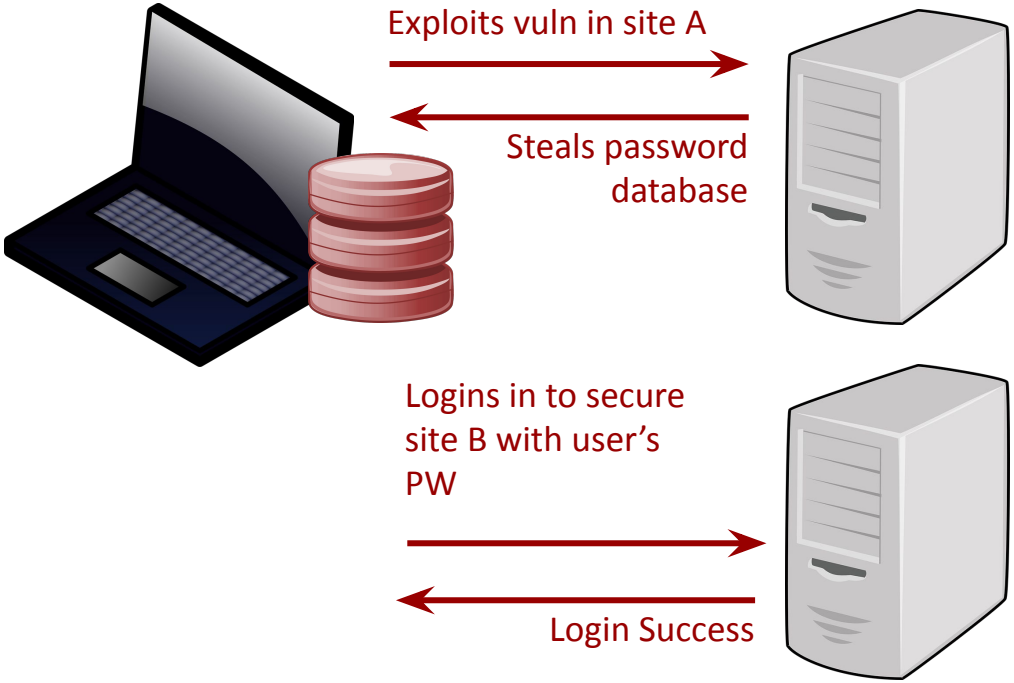


Defeating CAPTCHAs?

Problems with Passwords

People tend to use the same password for many purposes.

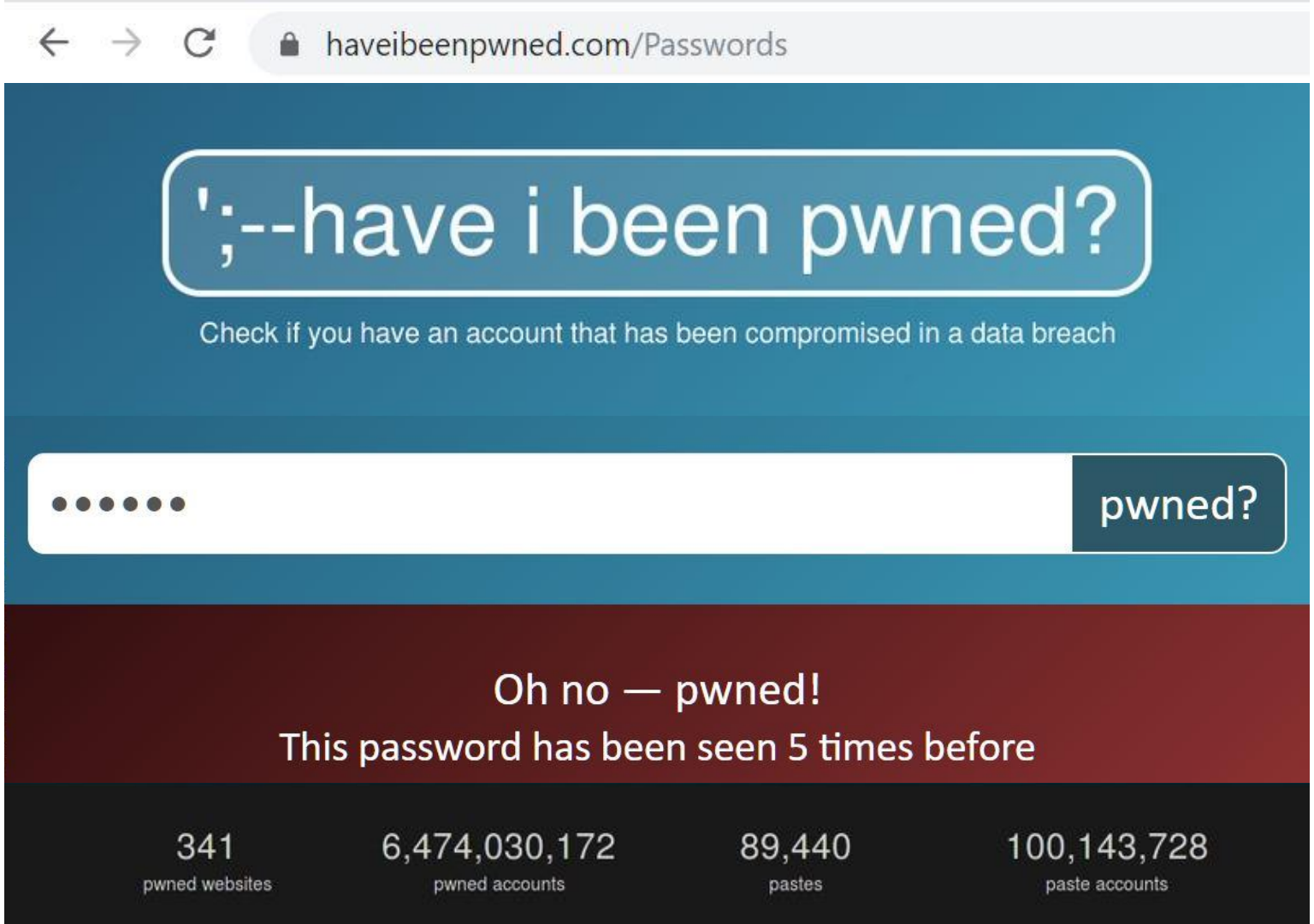
Why is this a problem?



Password Breaches

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

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


haveibeenpwned.com/Passwords
';--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

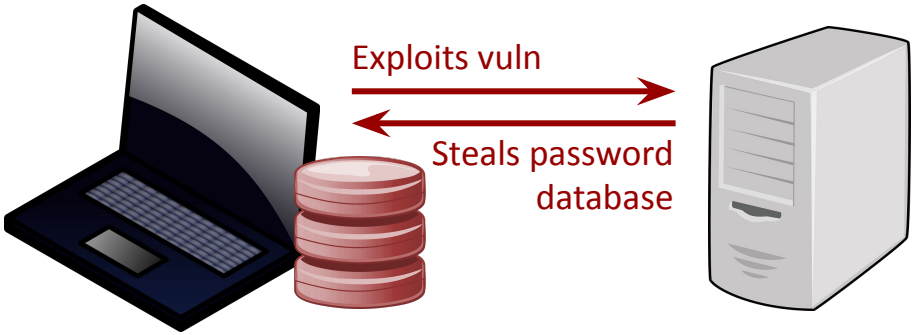
Defending Against Password Breaches

How should site store passwords to reduce risk?

Bad: **Plaintext passwords**

Pro: Easy.

Con: If leaked, company goes bust.



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.

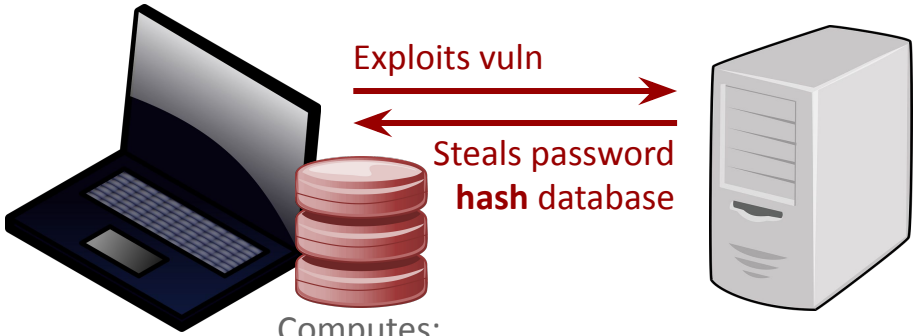
Con: Identical passwords have identical hashes.

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 <*salt*, H(*salt* || *password*)> in database.
Compare H(*salt* || *submitted_pw*) to H(*salt* || *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

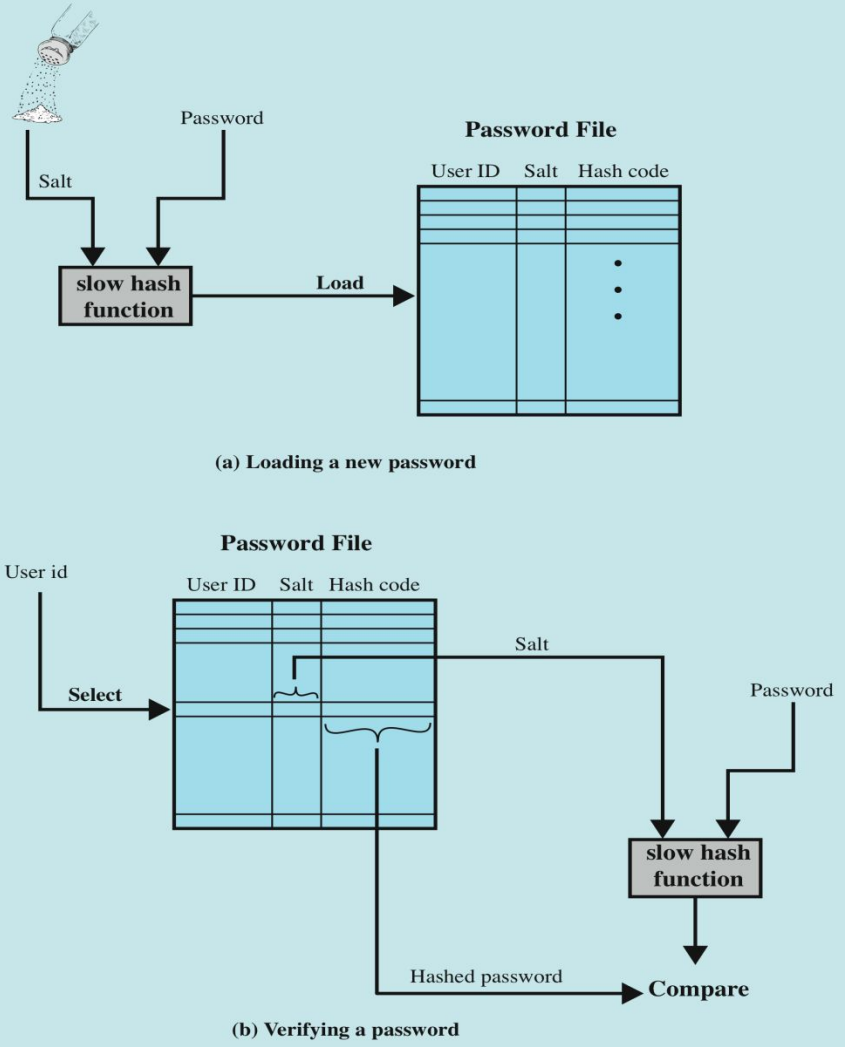
Common mistakes:

- 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.



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.

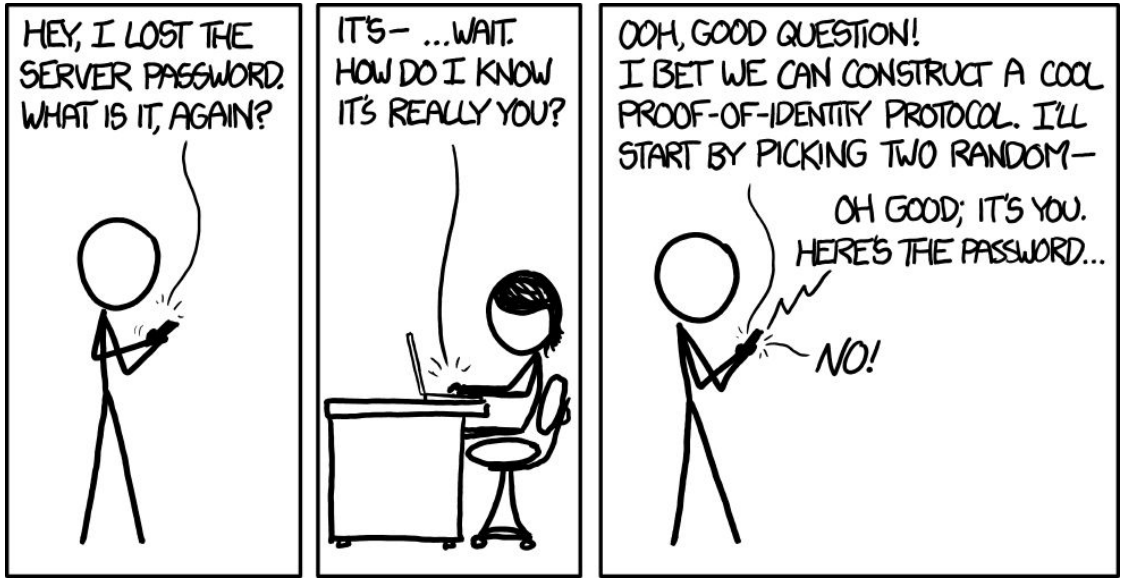
Problems with Passwords

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)

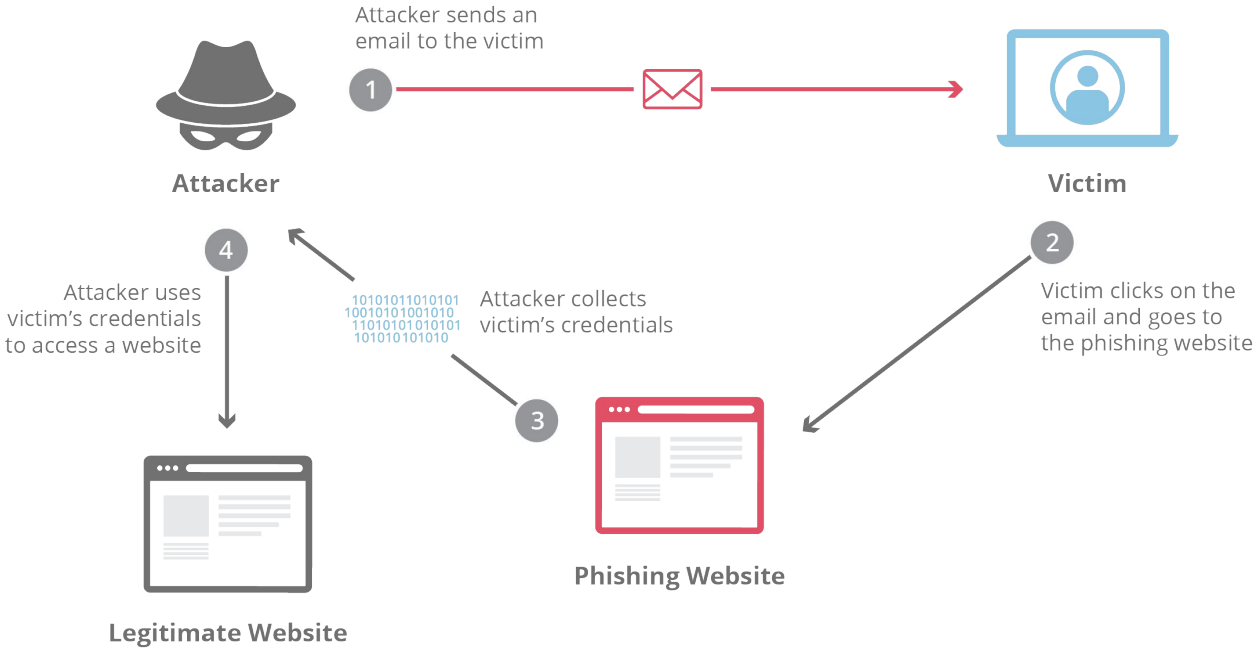
Security questions and **password hints** are often harmful and should be avoided.



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

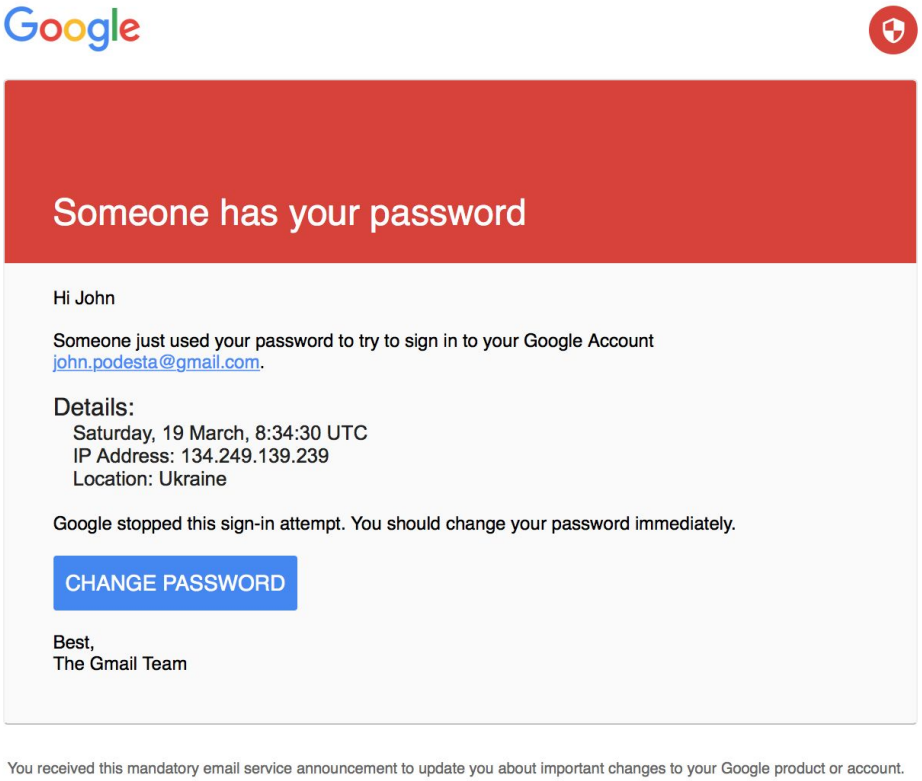
Typically, **phishing** is conducted by sending **forged emails** to large groups of potential victims.

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

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.



Google

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

You received this mandatory email service announcement to update you about important changes to your Google product or account.

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.

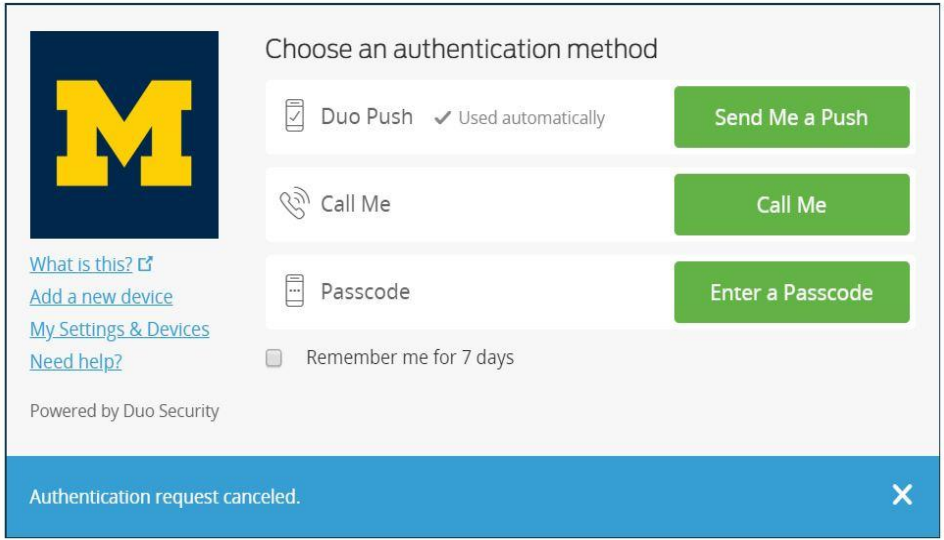
Best Practice: Multi-Factor Authentication

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: OTP(c, k) := HMAC(k, c)

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

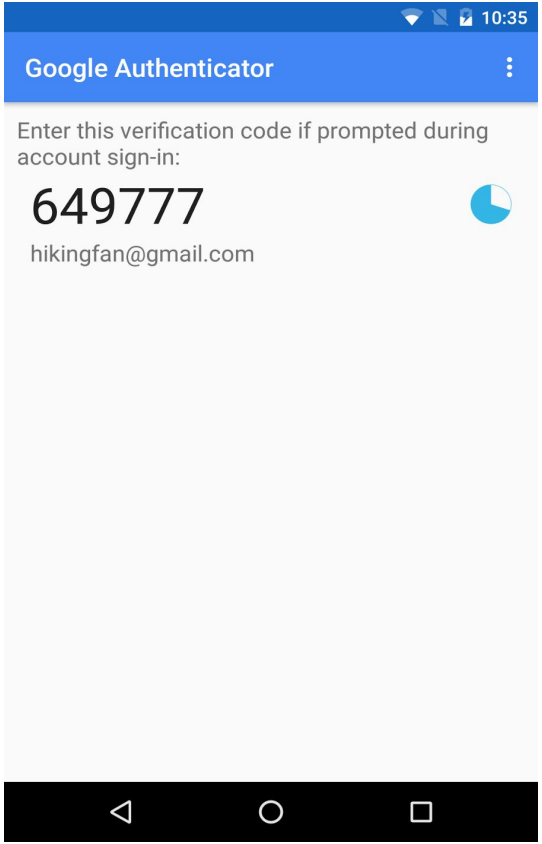
Cons: Susceptible to phishing

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.

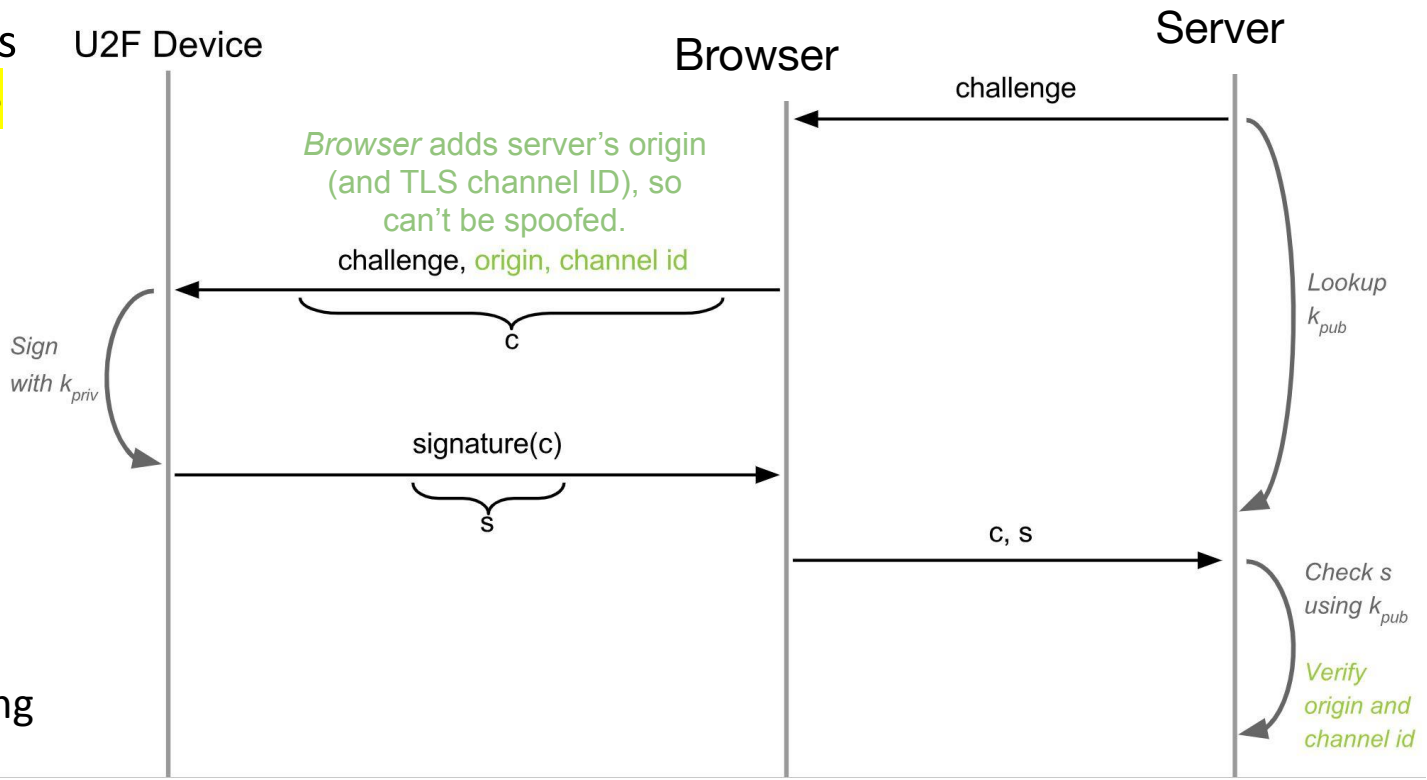
WebAuthn

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.



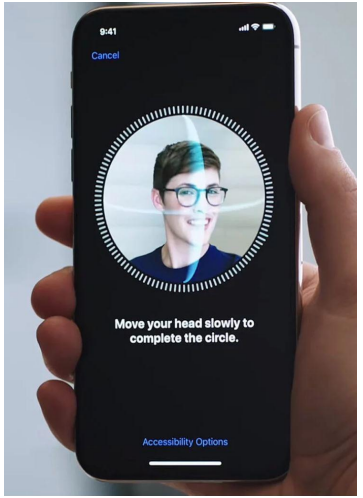
Biometrics

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

Coming Up

Reminders:

Lab Assignment 3 due TODAY at 6 PM

Midterm Exam is Oct. 20, 7–8:30 PM

No labs this week or next week

Tuesday:

Study break!

(No lecture)

Thursday:

Midterm review session during lecture