Stony Brook University

CSE 361: Web Security

Authentication

Nick Nikiforakis

Controlling access

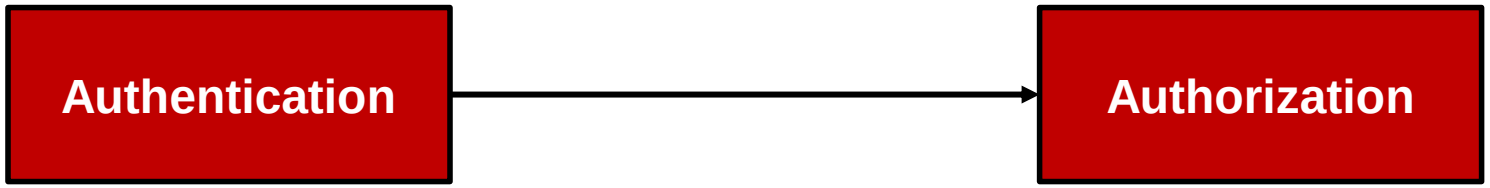
- One of the things that makes security hard is that we want fine-grained access to systems, networks, and information
  - Everyone accessing them is not okay
  - No-one accessing them is also not okay
- We only want specific users (or programs acting on behalf of users) to access specific programs, systems, and data
- So the first step is authentication
  - How do I know you are, who you say you are?

User Authentication

- Authentication based on:
  - What you know (password, answer to security question, personal image etc.)
  - What you have (smartcard, hardware token, etc.)
  - Who you are (biometrics)
  - Where you are (IP address, GPS location)
    - Typically used an extra signal in authentication

Password-based Authentication

- Passwords are key to the process of **authentication**
  - Authentication is at the heart of security



How do I know you are, who you say your are?

Are you allowed to do the thing you are asking?

Password-based Authentication

User has a secret password.

System checks it to authenticate the user.

- Security considerations
  - How is the password communicated?
    - Eavesdropping risk (We will see later how crypto can be used)
  - How is the password stored?
    - In the clear? Encrypted? Hashed?
  - How does the system check the password?
  - How easy is it to guess the password?
    - Easy-to-remember passwords tend to be easy to guess

Passwords and Computer Security

- In 2022, 82% of breaches involved a human element
  - Source: Verizon Data Breach Investigations Report
  - Stolen credentials, phishing, misuse, etc.
- Common first step after any successful intrusion: install sniffer or keylogger to steal more passwords
- Second step: run cracking tools on password files
  - Cracking needed because modern systems usually do not store passwords in the clear (how are they stored?)
- In Mitnick's "Art of Intrusion", 8 out of 9 exploits involve password stealing and/or cracking

Password Security Risks

- Keystroke loggers
  - Hardware
    - KeyGhost, KeyShark, others
  - Software (spyware)
- Shoulder surfing
- Same password at multiple sites
  - Cascading effects of a single break-in
- Social engineering

How do passwords look like?

- A lot of what we know about passwords is mainly through two ways
  - **Ethical**: Academic studies done in universities where researchers ethically investigated the password practices of their institutions
  - **Less ethical**: Hacking groups hacked and release entire databases of passwords, from their victims.

["Measuring Password Guessability for an Entire University", Mazurek et al., CCS 2013]

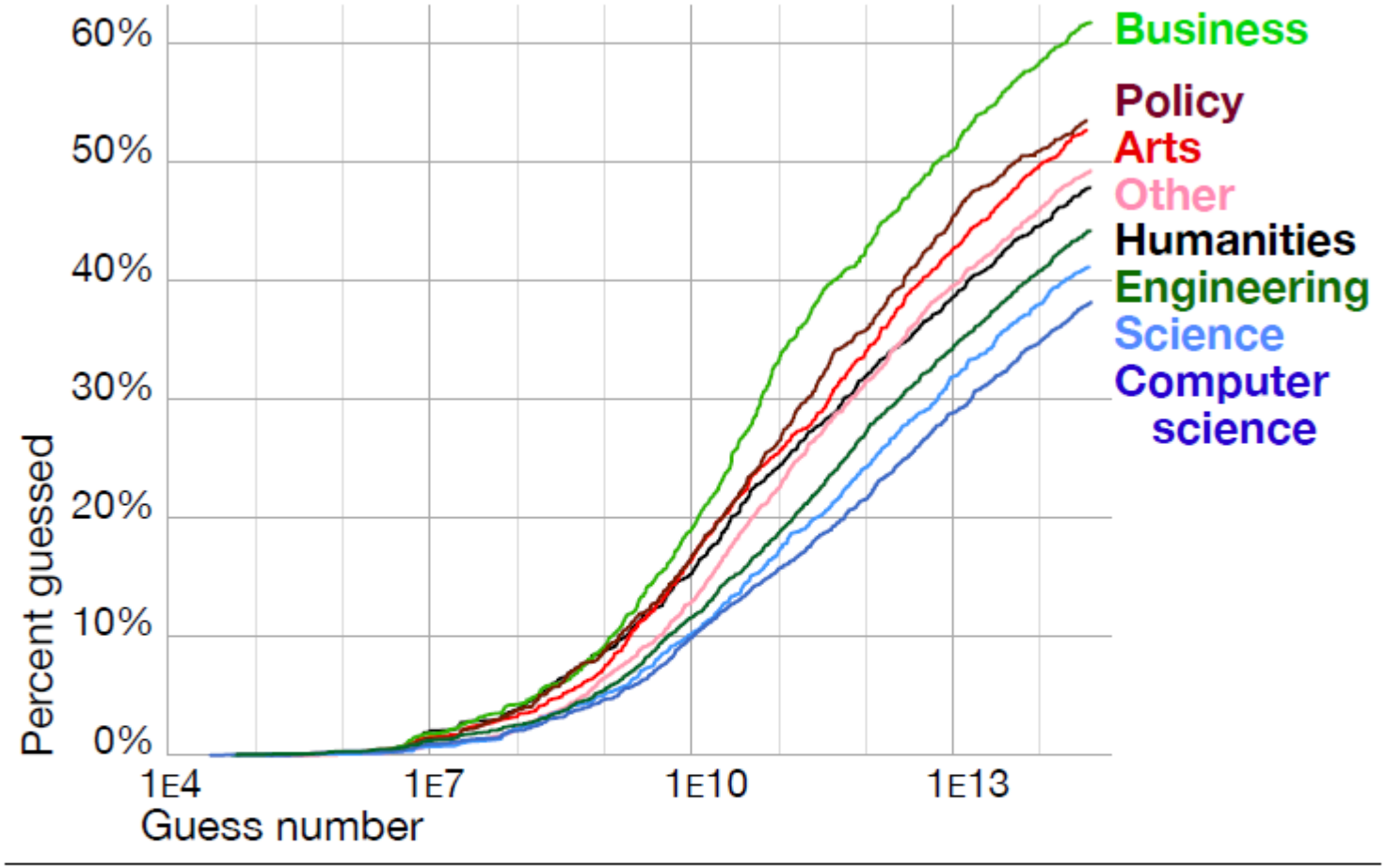


Figure 1: The percentage of passwords guessed after a given number of guesses (shown in log scale), by college within the university.

["Measuring Password Guessability for an Entire University", Mazurek et al., CCS 2013]

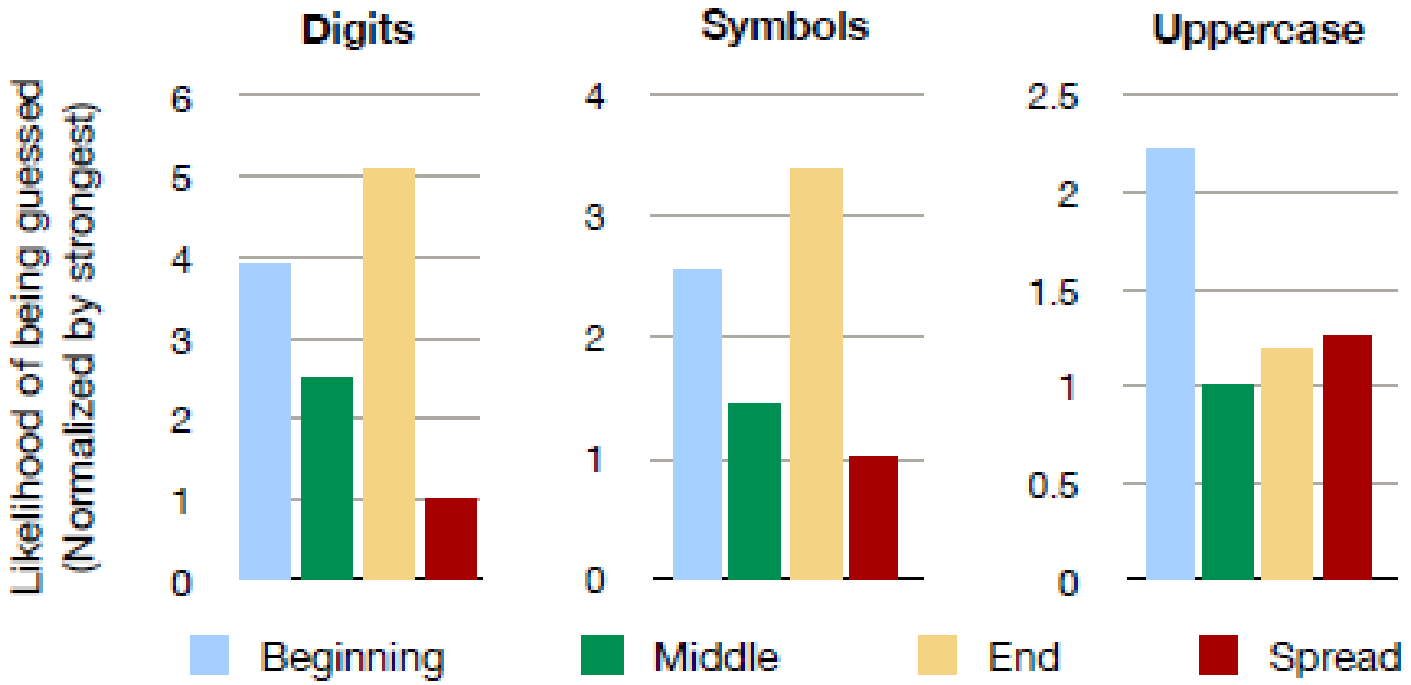


Figure 3: The relative likelihoods of passwords with digits, symbols, or uppercase letters in a given location being cracked. For example, a password with all its digits at the end is five times as likely to be cracked as a password with its digits spread throughout, other things being equal.

Adobe Passwords (2013)

- 153 million account passwords
  - 56 million of them unique
- Encrypted using 3DES in ECB mode rather than hashed (why is this important?)

Rock You Hack (2009)

- “Social gaming” company
- Database with 32 million user passwords from partner social networks
- Passwords stored in the clear
- December 2009: entire database hacked using an SQL injection attack and posted on the Internet
  - More about SQL injection attacks later

rockyou™

Passwords in RockYou Database

[Imperva]

Password Popularity – Top 20

Rank	Password	Number of Users with Password (absolute)
1	123456	290731
2	12345	79078
3	123456789	76790
4	Password	61958
5	iloveyou	51622
6	princess	35231
7	rockyou	22588
8	1234567	21726
9	12345678	20553
10	abc123	17542

Rank	Password	Number of Users with Password (absolute)
11	Nicole	17168
12	Daniel	16409
13	babygirl	16094
14	monkey	15294
15	Jessica	15162
16	Lovely	14950
17	michael	14898
18	Ashley	14329
19	654321	13984
20	Qwerty	13856

More Password Datasets

PlentyOfFish® Free Online Dating

More than 30 million passwords

eHarmony®

"#1 Most Trusted Online Dating Site"

SQL injection attack

For sale for $3000

Dropbox 2016

The Dropbox hack is real

31 AUGUST 2016

Earlier today, Motherboard reported on what had been rumoured for some time, namely that Dropbox had been hacked. Not just a little bit hacked and not in that "someone has cobbled together a list of credentials that work on Dropbox" hacked either, but *proper* hacked to the tune of 68 million records.

Article link: https://www.troyhunt.com/the-dropbox-hack-is-real/

More recently

- DailyQuiz
  - details about 12.8 million users, including plaintext passwords, emails, and IP addresses for 8.3 million accounts.
- SolarWinds
  - Network-monitoring software installed in government agencies
  - Attackers broke into SolarWind's and added a backdoor to code that went out with the next update to 18K clients
  - Password "solarwinds123" is one of the suspected ways that attackers used

Catalin Cimpanu

May 24, 2021

Cybercrime | News | Technology

8.3 million plaintext passwords exposed in DailyQuiz data breach

The personal details of 13 million DailyQuiz users have been leaked online earlier this year after a hacker breached the quiz builder's database and stole its content, which he later put up for sale.

The data, of which *The Record* has obtained copies from two different sources, contains details about 12.8 million users, including plaintext passwords, emails, and IP addresses for 8.3 million accounts.

HOME > TECH

The US is readying sanctions against Russia over the SolarWinds cyber attack. Here's a simple explanation of how the massive hack happened and why it's such a big deal

Isabella Jibilian and Katie Canales Updated Apr 15, 2021, 1:25 PM

Attackers

- What is the threat model?
  - Online attacker
    - Tries to login to a service by iteratively trying passwords and looking whether he was successful
  - Offline attacker
    - Stole password database and tries to recover the, hopefully protected, passwords
      - Also known as a “dictionary attack”
  - Against one user
  - Against all/any user

How do attackers use passwords?

- Once a database of credentials is leaked, attackers can use them in multiple ways
  - Extract emails and usernames
    - Chances are that users are reusing the same username/email address in other unrelated services
  - Learn what are the most common passwords that most users use
  - Learn what are the passwords that specific users use

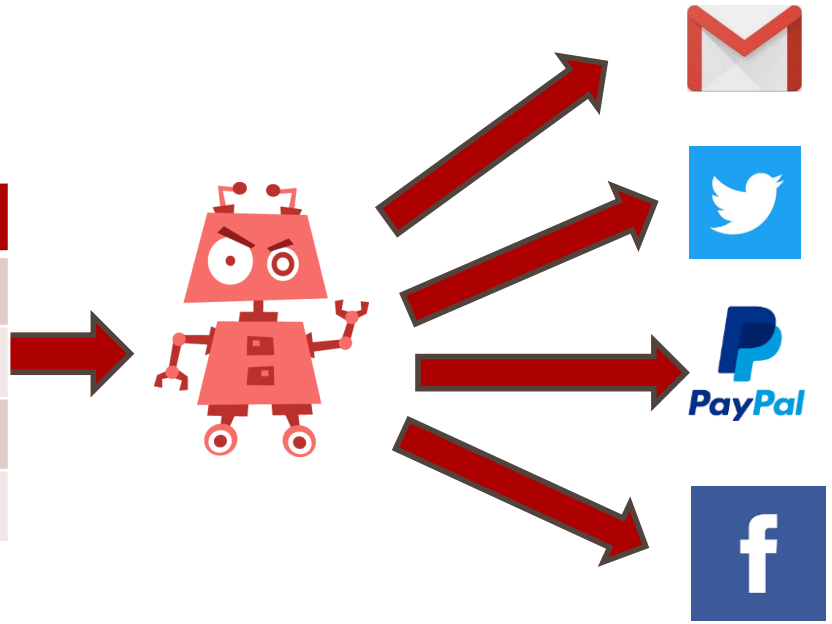
Username	Password
alice@gmail.com	ilovedogs
bob@yahoo.com	Password!
eve@outlook.com	1q2w3e4r
john@stonybrook.edu	g@rfield1

Credential stuffing

- Attackers build programs that try these credentials against other services
  - These programs act like regular users trying to log in
  - Attackers bet on users reusing their passwords

Username	Password
alice@gmail.com	ilovedogs
bob@yahoo.com	Password!
eve@outlook.com	1q2w3e4r
john@stonybrook.edu	g@rfield1

supercutecats.com



Credential stuffing is a real and growing problem

Dunkin' Donuts accounts compromised in second credential stuffing attack in three months

Hacked Dunkin' Donuts accounts are now being sold on Dark Web forums.

By Catalin Cimpanu for Zero Day | February 12, 2019 -- 01:43 GMT (17:43 PST) | Topic: Security

The gaming community is a rising target for credential stuffing attacks

Hackers have targeted the gaming industry by carrying out 12 billion credential stuffing attacks against gaming websites within the 17-month period analyzed in the report (November 2017 – March 2019) by Akamai.

Retailers have become the top target for credential stuffing attacks

Bots are being used to complete rapid-fire fraudulent purchases with very little effort from the hackers behind them.

By Charlie Osborne for Zero Day | February 27, 2019 -- 11:00 GMT (03:00 PST) | Topic: Security

DailyMotion discloses credential stuffing attack

DailyMotion falls to credential stuffing attack two weeks after Reddit had the same fate.

By Catalin Cimpanu for Zero Day | January 27, 2019 -- 12:02 GMT (04:02 PST) | Topic: Security

Online attacker

- How do we detect an online attacker?
  - Too many wrong tries
    - Distinctly different from a user who first was wrong but then was right
  - Tries multiple accounts instead of just one
    - Tradeoff to allow for NAT usage

- What can we do?
  - CAPTCHAs to differentiate between bots and humans
  - Temporarily block the IP address or rate-limit the number of requests
    - Tricky if attack is distributed
  - Temporarily lock the account that is being attacked
    - Rarely a good solution (Harms availability property)

Offline attacker

- Attacker somehow obtains the list of our passwords
  - Break-in to server
    - Credential guessing, SQL injection, Remote-command execution,…
  - Backups
  - Social engineering

- How do we store passwords?
  - **It's obvious** that the passwords should not be stored in the clear
  - How do we not store them in the clear, and still check them against users attempting to log in?

Unfortunately, not obvious for all

Plain Text Offenders

Follow plaintextoffenders

Did you just email me back my own password?!

- About
- FAQ
- Developers FAQ
- Offenders List
- 3rd Party Tools
- Reformed Offenders
- Archive
- Talk To Us
- Submit a post

NOTE: Tumblr's search feature is broken and therefore disabled. Please use the list at plaintextoffenders.com/offenders to search for any domain.

May 31st, 2021 at 6:01PM

Get Messages | Write | Chat | Address Book | Tag | Quick Filter

From Shodan <no-reply@mg.shodan.io>
Subject **Shodan Account Information**
To Me

Hi,

Somebody asked to reset your password on Shodan. If it wasn't you, you can safely ignore this email.
Log in with this information and change your password:

Account Information

URL: https://account.shodan.io/change-password

Username: [redacted]
Password:

Thank you for using Shodan!

Should we use encryption?

- How about encrypting each password with a secret key (e.g. only stored in the memory of the server) which is used to decrypt any single entry, on demand?

- Still a bad idea….
  - The attacker can steal your key and decrypt everything
  - The administrators can know users' passwords (no reason that they should)

Password Hashing

- Instead of user password, store Hash(password)
- When user enters a password, compute its hash and compare with the entry in the password file
  - System does not store actual passwords
  - Cannot go from hash to password
    - … except by guessing the password
- Hash function H must have some properties
  - Given H(password), hard to find any string X such that H(X)=H(password) - why?

Sample Cryptographic hash functions

Name	Year of release	Digest size (output size)
MD5 (Media Digest 5)	1992	128-bit
SHA-1 (Secure Hash Algorithm 1)	1995	160-bit
SHA-256 (Part of the SHA-2 family)	2001	256-bit

MD5(“helloworld”) = d73b04b0e696b0945283defa3eee4538
SHA-1(“helloworld”) = e7509a8c032f3bc2a8df1df476f8ef03436185fa
SHA-256(“helloworld”) = 8cd07f3a5ff98f2a78cfc366c13fb123eb8d29c1ca37c79df190425d5b9e424d

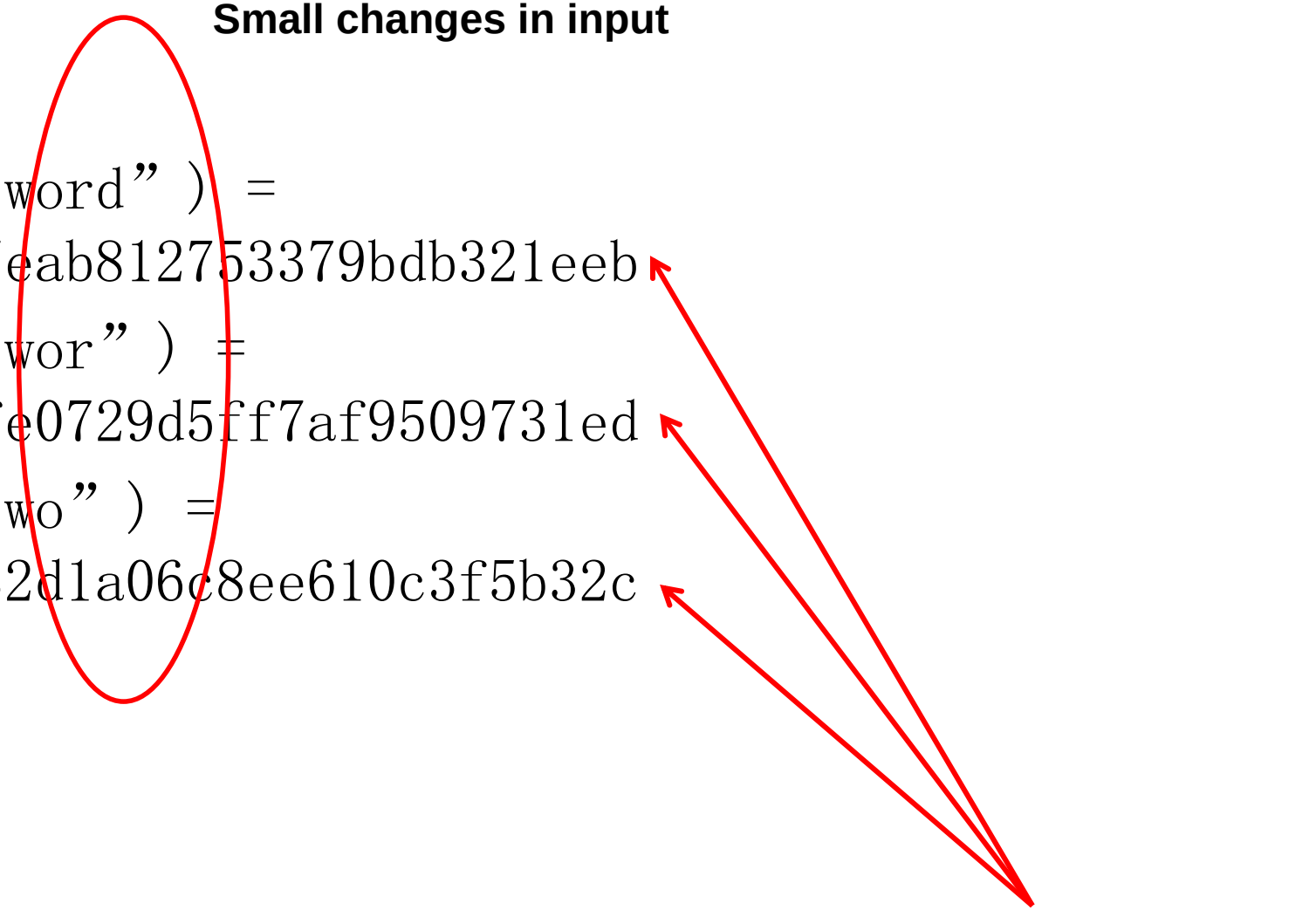
Examples

Small changes in input

SHA1(“mysecretpassword”) =
08cd9233678900096 57eab812753379bdb321eeb

SHA1(“mysecretpasswor”) =
0c894b9cd0fef7d1ccfe0729d5ff7af9509731ed

SHA1(“mysecretpasswo”) =
27c2d31b648cf7773032d1a06c8ee610c3f5b32c



Large differences in output

This is called the “avalanche” effect

Hashing vs. Encryption

- Hashing is one-way. There is no “uh-hashing”!
  - A ciphertext can be decrypted with a decryption key… hashes have no equivalent of “decryption”
- Hash(x) looks “random”, but can be compared for equality with Hash(x’)
  - Hash the same input twice → same hash value
- Cryptographic hashes are also known as “cryptographic checksums” or “message digests”

Steps for authenticating when hashing is utilized

- When an existing user is trying to login:
  1. Hash the provided password
  2. Compare it to the stored hash for that user
  3. If the hashes match then the provided password is the same as the original one

- Better than before, but still has issues
  - Same passwords of different users will have the same hash
  - Attacker can precompute hashes of popular words and try them against all accounts (rainbow tables)

Salting

- Instead of just hashing the user's password, hash the user's password when concatenated with a per-user random value

SHA256("mysecretpassword")

Username	Password
nick	94AEFB8BE78B2B7C344D11D 1BA8A79EF087ECEB19150881 F69460B8772753263

SHA256("199654mysecretpassword")

Username	Salt	Password
nick	199654	1C8622F514E7BB8B86210FE8 83D48CC55C5BEDA849DAF74 6AFFFDEC757952F77

Salting steps

- When an existing user is trying to login:
  1. Concatenate the database-stored salt to the user-provided password
  2. Hash the entire thing
  3. Compare it to the stored hash for that user
  4. If the hashes match then the provided password is the same as the original one

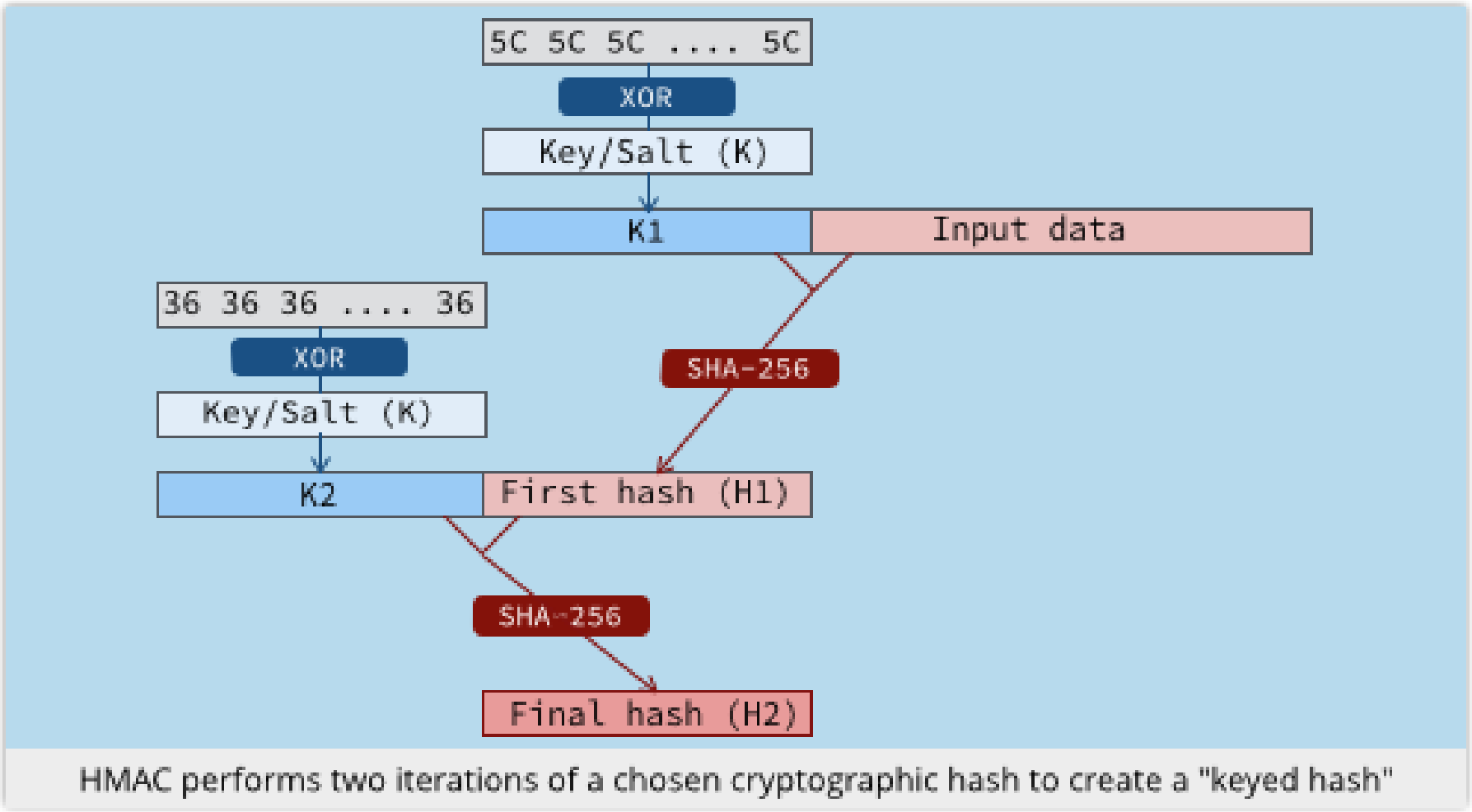
- Precomputed tables don’t work
- Same passwords for different users have different hashes
- Attacker must perform hashing “live” in order to crack passwords

HMAC

- Concatenation of keys with data can lead to some exploitable cryptographic scenarios
  - Outside the scope of this course

- HMAC (Keyed Hashed Message Authentication Code) allows us to combine the salt with the hash of the password in a more secure way

HMAC with SHA



HMAC performs two iterations of a chosen cryptographic hash to create a "keyed hash"

Image source: https://nakedsecurity.sophos.com/2013/11/20/serious-security-how-to-store-your-users-passwords-safely/

One more thing

- Our steps so far allow us the following guarantees:
  - User passwords should not be recoverable from a database
  - Identical/Similar passwords will have different hashes
  - The database does not “leak” the length of a user’s password

- The only problem remaining is that offline attackers, if they are dedicated enough, they can still brute-force their way into users with weak passwords

Password Guessing Techniques

- Dictionary with words spelled backwards
- First and last names, streets, cities
- Same with upper-case initials
- All valid license plate numbers in your state
- Room numbers, telephone numbers, etc.
- Letter substitutions and other tricks
  - If you can think of it, attacker will, too

Password Hash Cracking

- Custom GPU-based hardware
  - GPUs are great for playing games and hashing
  - Most recent numbers for Nvidia RTX 4090:
    - 300 Gigahashes per second for Windows NTLM hashes

Home > News > Nvidia RTX 4090

8 RTX 4090s could crack most of your passwords in just 48 minutes

By Dave James published October 18, 2022

A modest cracking rig would be able to go through every single possible password combination of an eight-character password in less than an hour.

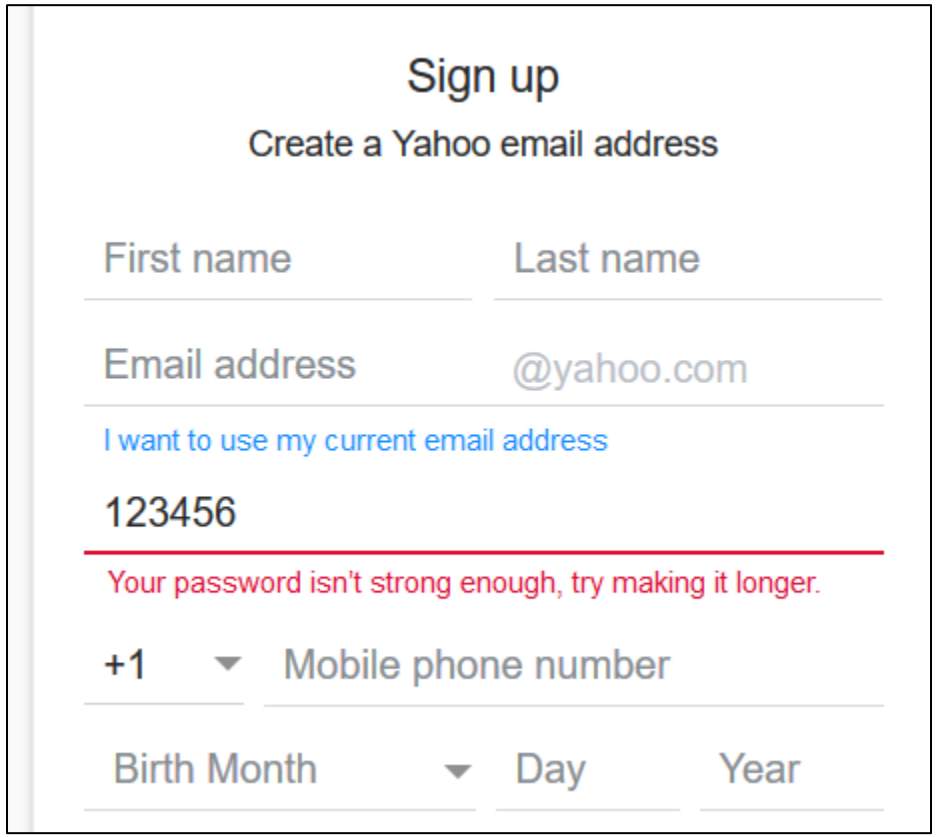
- Cloud-based cracking tools
  - Crackq
  - Password-cracking as a service

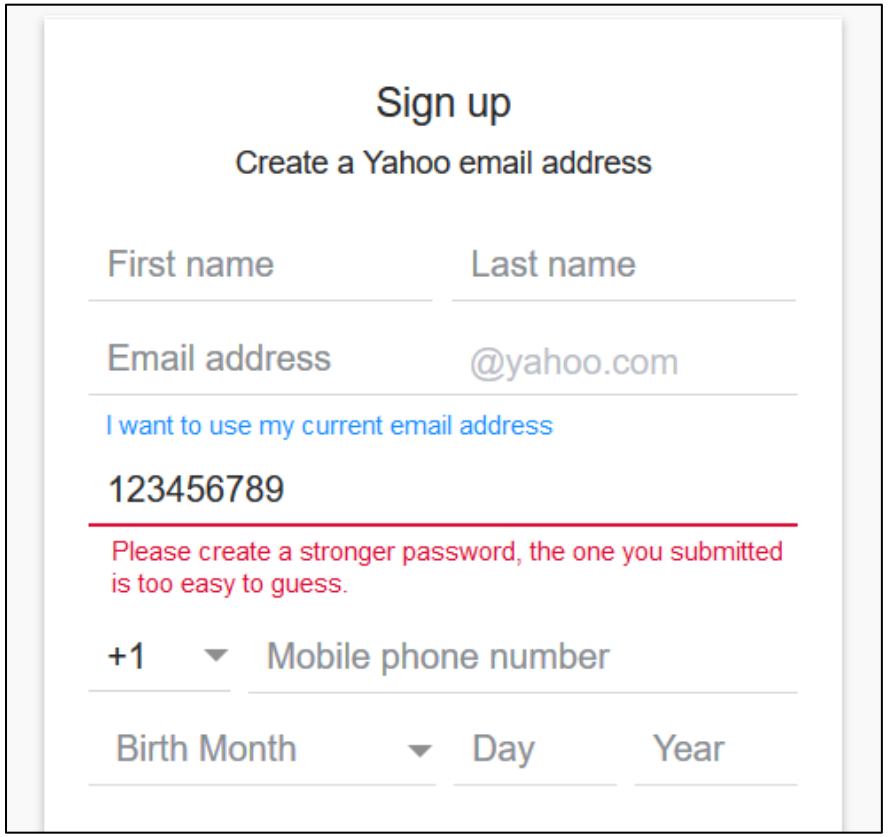
https://securityledger.com/2012/12/new-25-gpu-monster-devours-passwords-in-seconds/
https://www.pcgamer.com/just-8-rtx-4090s-could-crack-most-of-your-passwords-in-48-minutes/

Defense #1: Password requirements

- Systems can enforce password requirements when users register/change their passwords
  - Not a dictionary word
  - Must be at least X characters long
  - Must contain special characters
  - Is not part of a recently compromised database
    - Interfacing with third-party services such as haveibeenpwnd.com
- Other requirements are popular but not actually good
  - Change password every N months
    - NIST (National Institute of Standards and Technology) does not recommend forced password changes when passwords are not compromised

Examples of password-strength meters

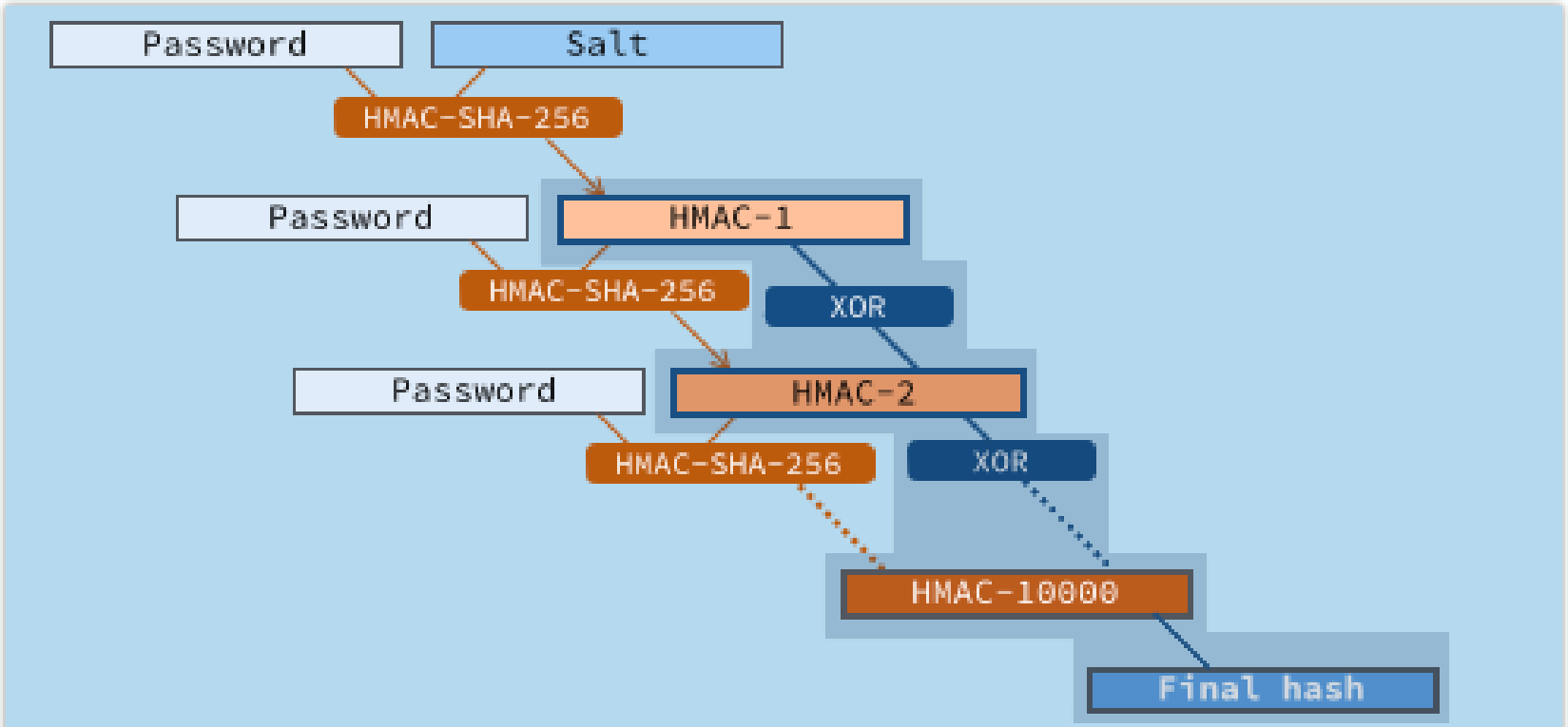




Defense #2: Hash stretching

- Why restrict ourselves to only one hash operation?
- If we perform multiple hashing rounds:
  - An attacker would need significantly more resources per cracking attempt
  - A server can still cope with the increased load because users are not authenticating all at the same time
- Standardized multi-round hashing algorithms
  - PBKDF2, brypt, scrypt

PBKDF2 + HMAC-SHA-256



PBKDF2 using XOR to combine 10,000 successive HMAC-SHA-256 outputs into a final hash

Image source: https://nakedsecurity.sophos.com/2013/11/20/serious-security-how-to-store-your-users-passwords-safely/

Demo: Cracking speed

Back to users – Password Policies

- Overly restrictive password policies…
  - 7 or 8 characters, at least 3 out of {digits, upper-case, lower-case, non-alphanumeric}, no dictionary words, change every 4 months, password may not be similar to previous 12 passwords…
- … result in frustrated users and less security
  - Burdens of devising, learning, forgetting passwords
  - Users construct passwords insecurely, write them down
    - Can't use their favorite password construction techniques (small changes to old passwords, etc.)
    - "An item on my desk, then add a number to it"
  - Heavy password re-use across systems

Password Usability

Password memorability

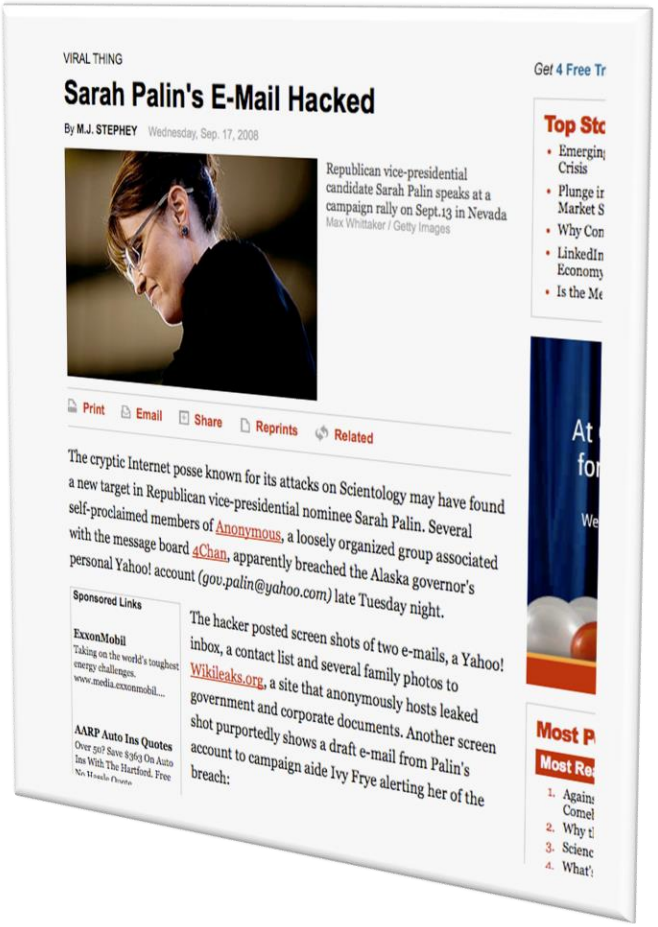
- Typically, **strength** of a password and **memorability** are working against each other
  - You can likely remember “jack123” better than “399%(mJjaweee”
- Various attempts have been made to come up with clever schemes for strong memorable passwords
  - “Abandon hope all ye who enter here” =>
  - aHaYwEh =>
  - aHaYvv3h

How People Use Passwords

- Write them down
  - Password managers attempt to make this okay
- Use a single password at multiple sites
  - Do you use the same password for Amazon and your bank account? Blackboard? Do you remember them all?
- Forget them… many services use “security questions” to reset passwords
  - “What is your favorite pet’s name?”
  - Paris Hilton’s T-Mobile cellphone hack

Sara Palin's Email Hack

- Reset password for
  gov.palin@yahoo.com
  - No secondary email needed
  - Date of birth? Wikipedia
  - ZIP code? Wasilla has 2
  - Where did you meet your spouse? Wikipedia, Google, …
- Changed pwd to "popcorn"
- Hacker sentenced to
  1 year in prison +
  3 yrs of supervised release



VIRAL THING

Sarah Palin's E-Mail Hacked

By M.J. STEPHEY Wednesday, Sep. 17, 2008

Republican vice-presidential candidate Sarah Palin speaks at a campaign rally on Sept.13 in Nevada

Max Whittaker / Getty Images

Print Email Share Reprints Related

The cryptic Internet posse known for its attacks on Scientology may have found a new target in Republican vice-presidential nominee Sarah Palin. Several self-proclaimed members of Anonymous, a loosely organized group associated with the message board 4Chan, apparently breached the Alaska governor's personal Yahoo! account *(gov.palin@yahoo.com)* late Tuesday night.

The hacker posted screen shots of two e-mails, a Yahoo! inbox, a contact list and several family photos to Wikileaks.org, a site that anonymously hosts leaked government and corporate documents. Another screen shot purportedly shows a draft e-mail from Palin's account to campaign aide Ivy Frye alerting her of the breach:

Sponsored Links

ExxonMobil

AARP Auto Ins Quotes

Problems with Security Questions

[Rabkin, "Security questions in the era of Facebook"]

- Inapplicable
  - What high school did your spouse attend?
- Not memorable
  - Name of kindergarten teacher? Price of your first car?
- Ambiguous
  - Name of college you applied to but did not attend?
- Easily guessable
  - Age when you married? Year you met your spouse? Favorite president? Favorite color?
- Automatically attackable (using public records!)

Answers Are Easy to Find Out…

- Make of your first car?
  - Until 1998, Ford had >25% of market
- First name of your best friend?
  - 10% of males: James/Jim, John, Robert/Bob/Rob
- Name of your first / favorite pet?
  - Max, Jake, Buddy, Bear…
  - Top 500 (covers 65% of names) available online
- Information available from Facebook, etc.
  - Where you went to school, college athletic rivals, favorite book/movie/pastime, high school mascot

…or Easy to Forget

- Name of the street, etc.
  - More than one
- Name of best friend
  - Friends change
- City where you were born?
  - NYC? New York? Manhattan? New York City? Big Apple?
- People lie to increase security… then forget the answers

Password Managers

- One place where all your passwords are stored
  - This place is protected with one master password
  - Flavors:
    - Online versus Offline (e.g. LastPass versus KeePass)

- Benefits
  - No need to remember any more passwords (other than the master phrase)
  - Unique password per website (no more password reuse)
  - Most password managers also have their own password generators to automatically create strong passwords
- Disadvantages
  - Single-point of failure
    - This can be easily mitigated by storing multiple copies of the database
  - Lock yourself out
    - If you forget your master password, there is no way to recover passwords
  - Cannot authenticate to services if you don’t have access to the password manager

Password managers

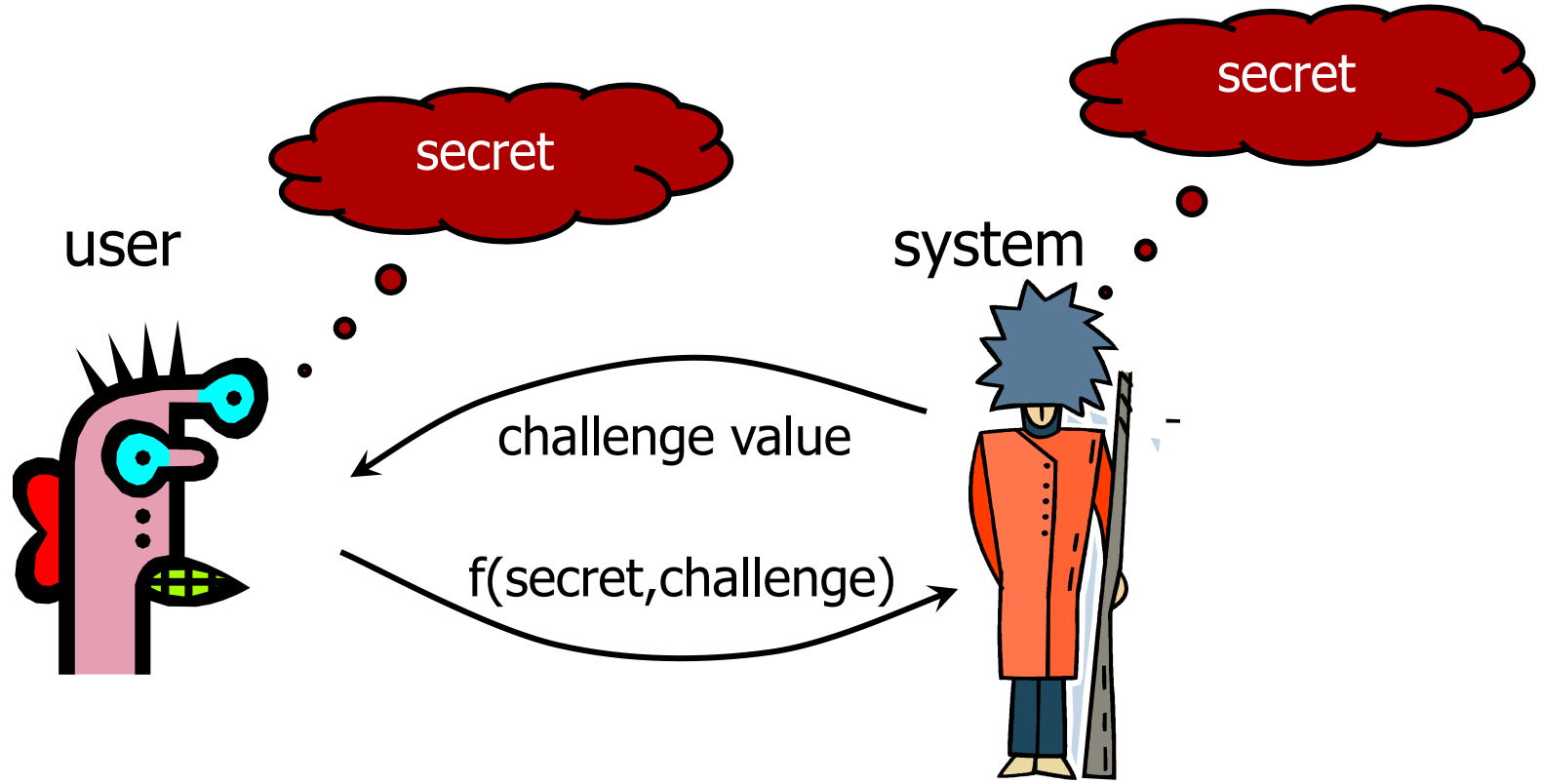
- LastPass was compromised in 2022
  - Attackers stole encrypted password vaults
  - Security is only as good as one's master password

- If malware is installed on a user's machine, all bets are off
  - Malware can try to steal both the encrypted vault and the master pasword

Replay attacks and possible solutions

- The standard, password-based authentication is vulnerable to **replay attacks**
  - A network attacker can see the password in traffic, and then later reuse to authenticate as the victim
- We can encrypt the entire channel to protect against this (explore this later in class) but we can also tackle it with **one-time passwords (OTP)**

Challenge-Response



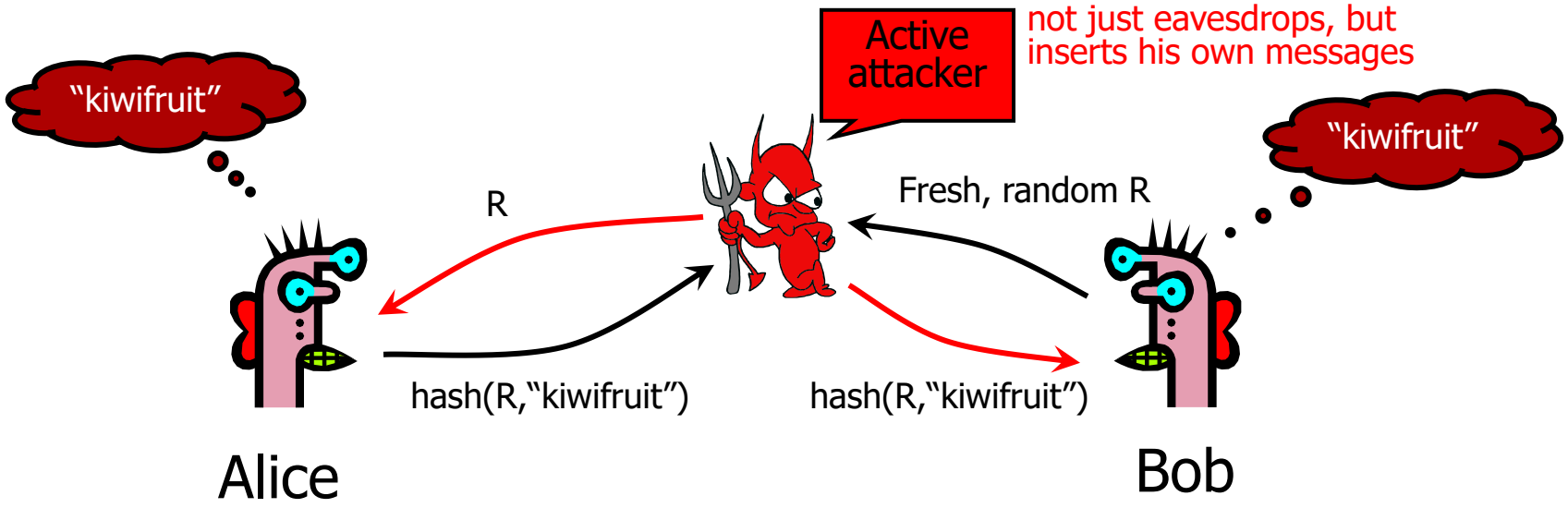
Why is this better than the password over a network?

Challenge-Response Authentication

- User and system share a secret (key or password)
- Challenge: system presents user with some string
- Response: user computes the response based on the secret and the challenge
  - Secrecy: difficult to recover secret from response
    - Cryptographic hashing or symmetric encryption work well
  - Freshness: if the challenge is fresh, attacker on the network cannot replay an old response
    - Fresh random number, counter, timestamp….
- Good for systems with pre-installed secret keys
  - Car keys; military friend-or-foe identification

Man-in-the-Middle Attack

- Man-in-the-middle attack on challenge-response
  - Attacker successfully "authenticates" as Alice by simple replay
- This is an online attack
  - Attacker does not learn the shared secret
  - Attacker cannot "authenticate" as Alice when she is offline



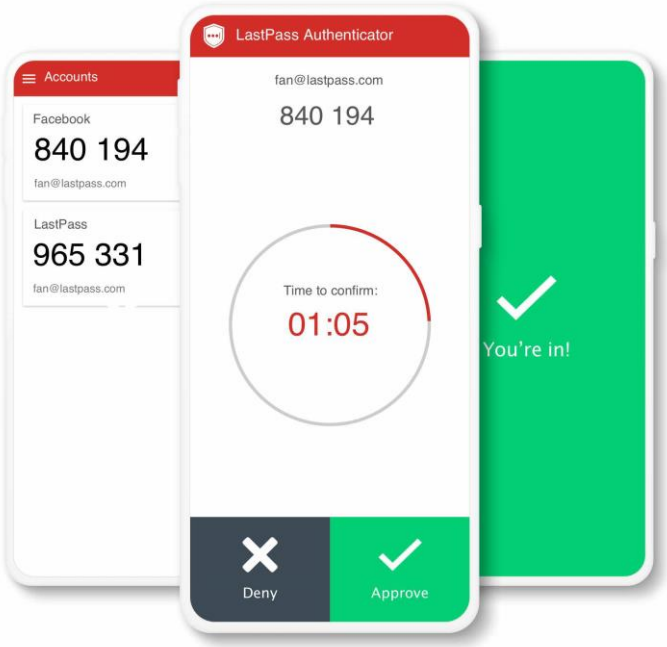
Making passwords stronger

- Passwords belong to the “what you know” category…
- Using “what you have” to strengthen the overall security of a system

- When more than techniques are used for authentication, then we have multiple-factor authentication
  - E.g. 2 Factor Authentication: password + phone

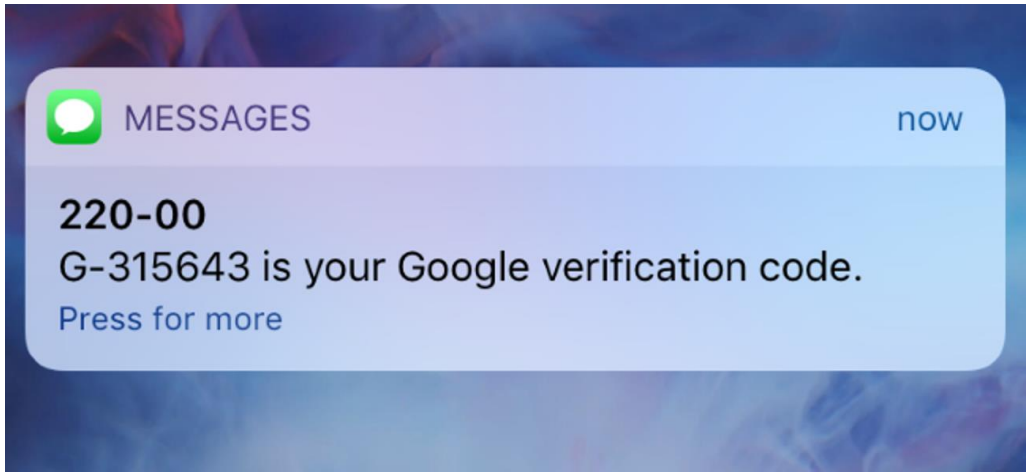
Something you have

- Things one can have
  - Access to your smartphone
    - Has gained a lot of traction recently due to popular web applications (Gmail, Twitter, etc.) supporting it
  - A bank card
  - A security token
    - A piece of hardware containing crypto that either generates one-time passwords or does a challenge-response protocol
  - A badge
- Problems
  - Stolen / forgotten / lost / duplicated
    - Higher cost to change than passwords
  - Cost of user education and support



Something you have - SMS

- Text messages (SMS) as a 2-factor authentication method is falling out of favor.
  - NIST has mentioned that it is deprecated and when possible, services should use hardware tokens or smartphone apps to deliver codes
- Reasons
  - Too many incidents of attackers social engineering phone companies into sending them SIM cards because the real owner “lost their phone”
  - Telcos in authoritarian governments can cooperate with their governments
  - Phone networks and their protocols are not exactly the most secure ones



Something you have - SMS

- SIM-swapping attacks are real
  - Particularly problematic when combined with cryptocurrencies
  - No reversal of transactions
- Moral of the story
  - Use when possible something other than SMS for 2FA
  - SMS-based 2FA is still **MUCH** better than just password-based authentication

Policy

Two US Men Sentenced for Stealing Crypto Using 'SIM Swapping'

The duo targeted "at least 10 identified victims" stealing "approximately $330,000 in cryptocurrency."

By Amitoj Singh Oct 20, 2022 at 6:19 a.m. EDT

Home / Finance / Blockchain

FBI warns: SIM-swapping attacks are rocketing, don't brag about your crypto online

FBI alert backs up Microsoft's call to avoid using phone numbers for two-factor authentication.

Written by **Liam Tung,** Contributing Writer on Feb. 9, 2022

Time-based One Time Passwords (TOTP) apps

- TOTP(K,C) = Truncate(HMAC-SHA-1(K,T))
  - K: Shared secret key
    - One copy in your app, one copy on the server
  - T: Current time (in specific steps)
    - Default time step of 30 seconds

- Resynchronization options
  - Allow for client-clocks being slightly slower / slightly faster
  - Potentially ask for additional codes

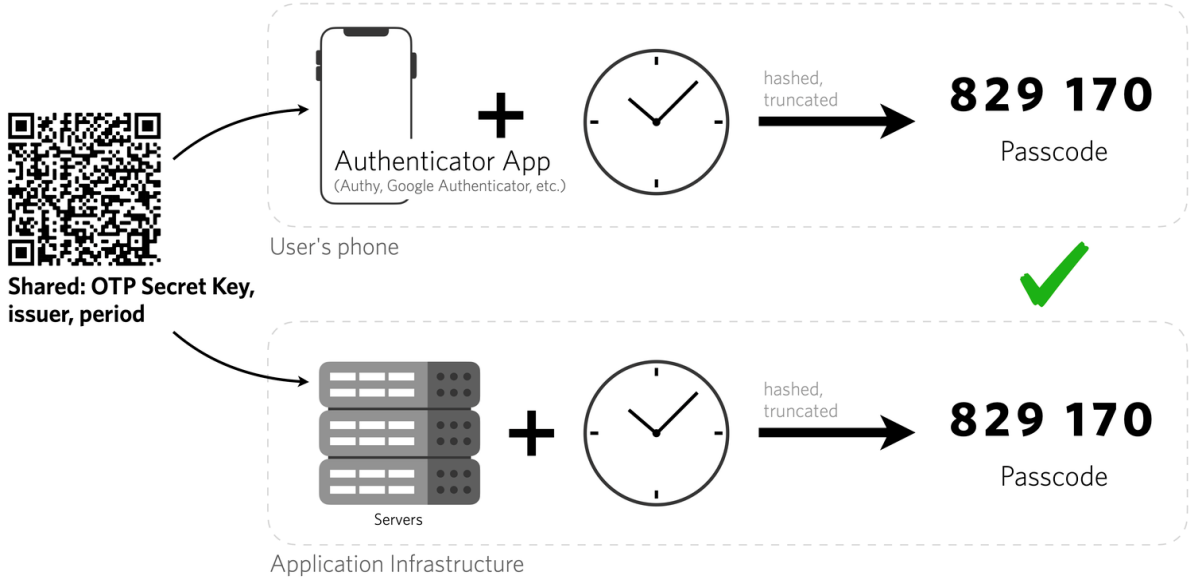
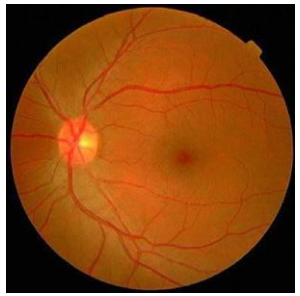
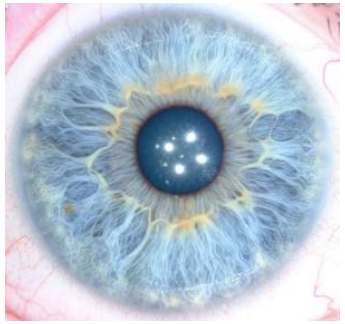


Image:https://www.twilio.com/docs/glossary/totp

Something you are

- **Biometrics**
  - Fingerprints
  - Palms
  - Face
  - Iris/Retina scanning
  - Voice
  - How you walk? How you type? How you swipe?
    - Research in continuous authentication
- Benefits
  - Nothing to remember
  - Passive (nothing to type, always carrying them around)
  - Can’t share
  - Can be fairly unique





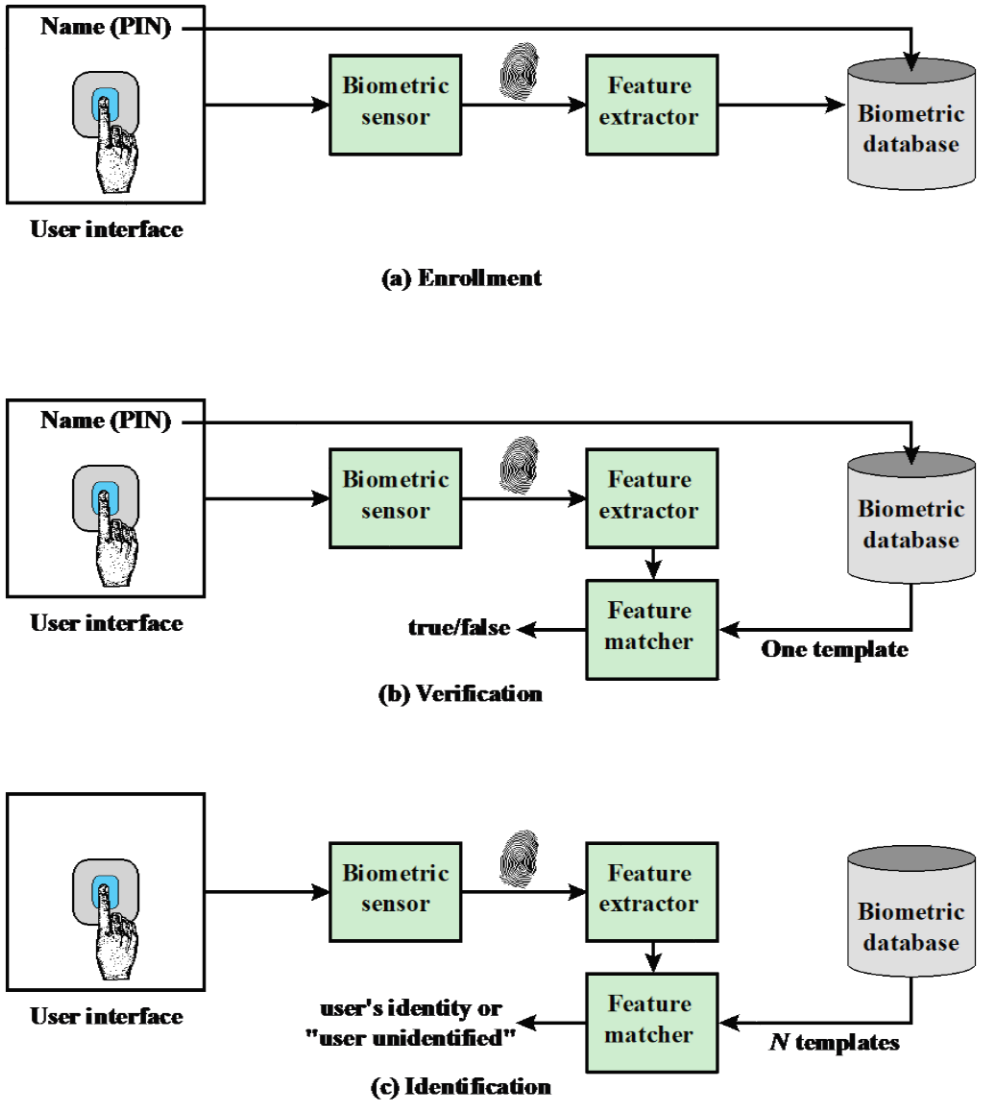


Figure 3.8 A Generic Biometric System. Enrollment creates an association between a user and the user's biometric characteristics. Depending on the application, user authentication either involves verifying that a claimed user is the actual user or identifying an unknown user.

Image Source: Computer Security: Principles and Practice

Problems

- Less socially acceptable
  - Put your eye in a large machine that blows air and shoots lasers
  - Smartphones are shifting this, at least for some biometric methods
- Revocability
  - You can change a password but how do you change your fingerprint?
- Are still spoofable
  - E.g. Pick fingerprints from objects and create molds
- Cost
  - Need special devices to read them
  - Human personnel to support them

- **Major difference with something you know/something you have?**
  - Probability of you being you, rather than certainty

Biometric Error Rates (Benign)

- “Fraud rate” vs. “insult rate”
  - Fraud = system accepts a forgery (false accept)
  - Insult = system rejects valid user (false reject)
- Increasing acceptance threshold increases fraud rate, decreases insult rate
- For biometrics, U.K. banks set target fraud rate of 1%, insult rate of 0.01% [Ross Anderson]

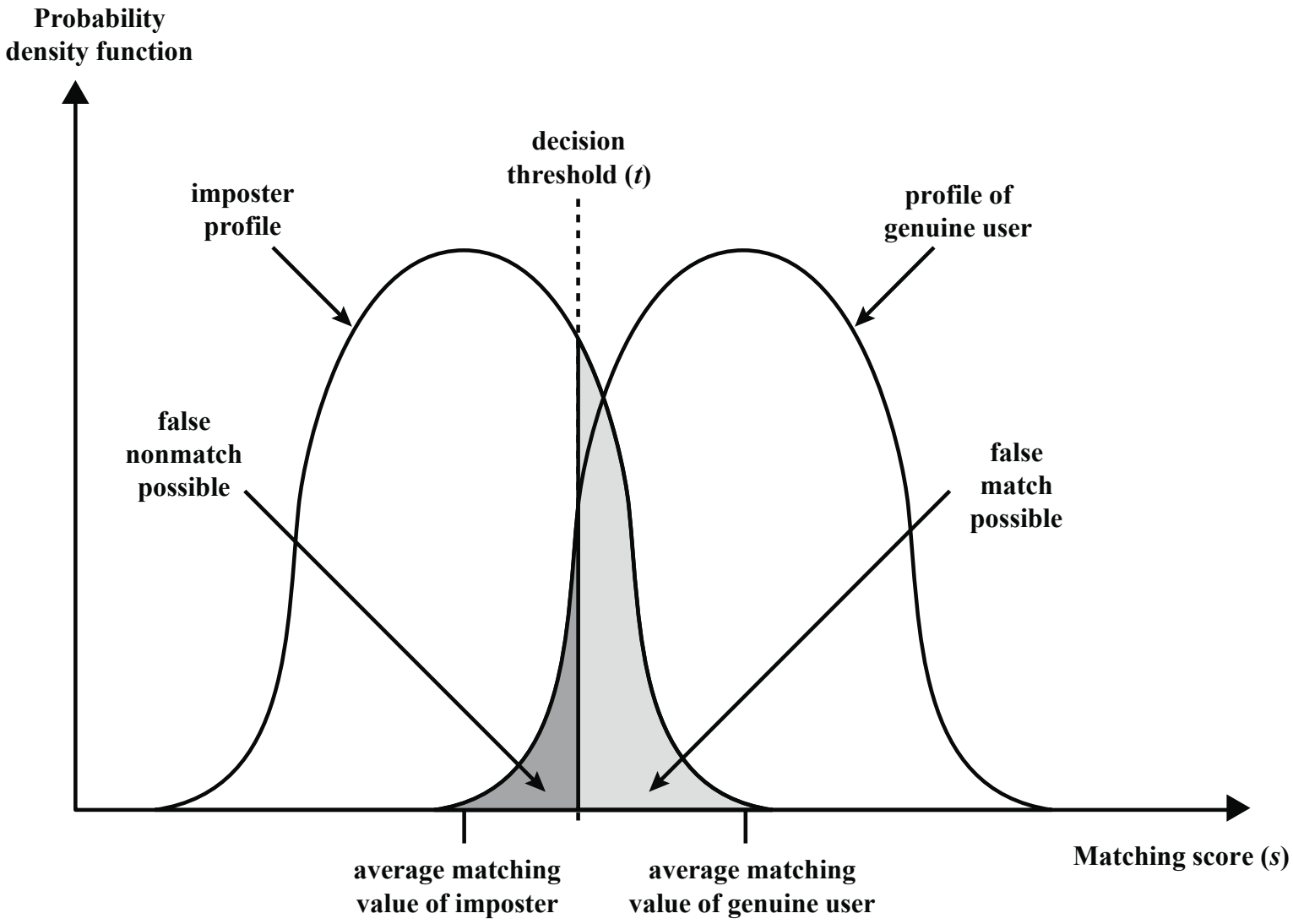


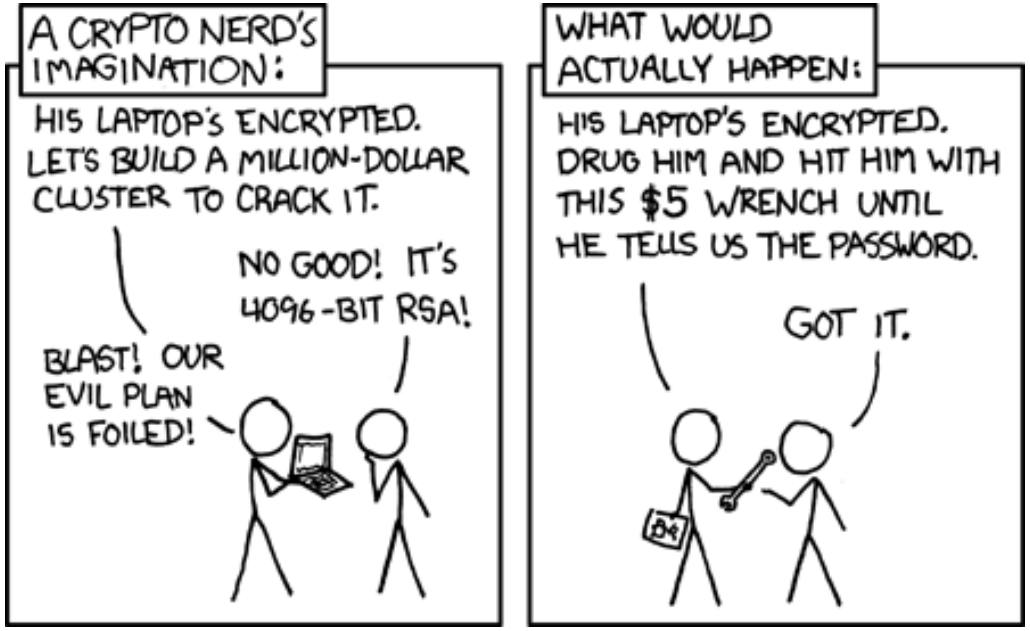
Figure 3.9 Profiles of a Biometric Characteristic of an Imposter and an Authorized Users In this depiction, the comparison between presented feature and a reference feature is reduced to a single numeric value. If the input value (*s*) is greater than a preassigned threshold (*t*), a match is declared.

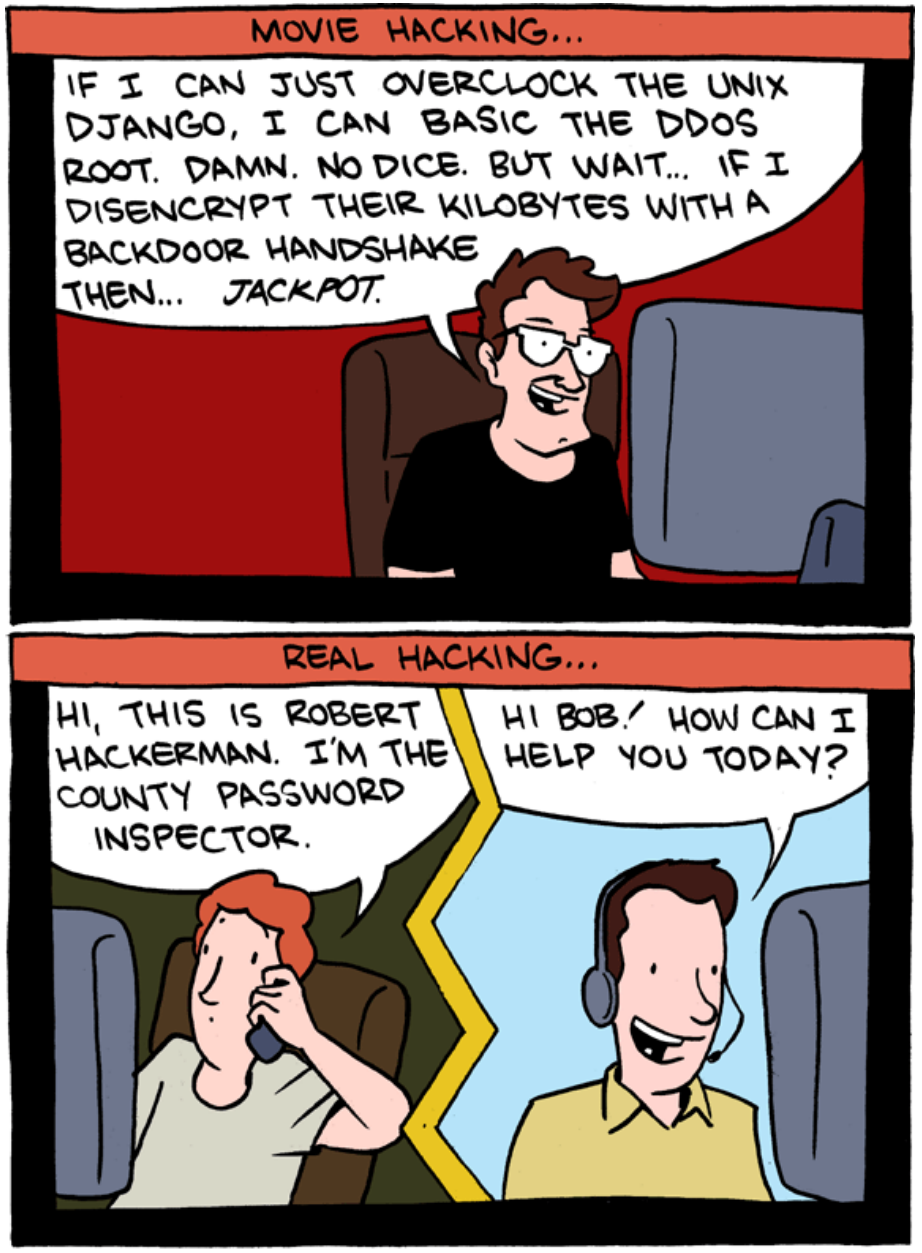
***Image Source**: Computer Security: Principles and Practice*

User-aspects

- Never forget that users are a critical part of securing an infrastructure
  - No matter how good your technology is, users can still ruin everything if someone convinces them that it is “okay”
- Abusing the trust of users: **social engineering**
- You can try to contain it by educating your personnel and setting up standard procedures
  - We will never ask you for your password over email

Social engineering





Summary

3

User Authentication

- Authentication based on:
  - What you know (password, answer to security question, personal image etc.)
  - What you have (smartcard, hardware token, etc.)
  - Who you are (biometrics)
  - Where you are (IP address, GPS location)
    - Typically used an extra signal in authentication

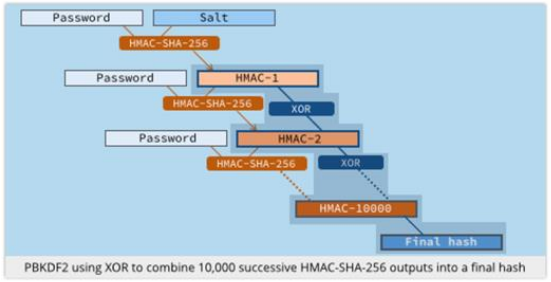
17

Attackers

- What is the threat model?
  - Online attacker
    - Tries to login to a service by iteratively trying passwords and looking whether he was successful
  - Offline attacker
    - Stole password database and tries to recover the, hopefully protected, passwords
      - Also known as a "dictionary attack"
  - Against one user
  - Against all/any user

40

PBKDF2 + HMAC-SHA-256



PBKDF2 using XOR to combine 10,000 successive HMAC-SHA-256 outputs into a final hash

Image source: https://nakedsecurity.sophos.com/2013/11/20/serious-security-how-to-store-your-users-passwords-safely/

60

Time-based One Time Passwords (TOTP) apps

- TOTP(K,C) = Truncate(HMAC-SHA-1(K,T))
  - K: Shared secret key
    - One copy in your app, one copy on the server
  - T: Current time (in specific steps)
    - Default time step of 30 seconds
- Resynchronization options
  - Allow for client-clocks being slightly slower / slightly faster
  - Potentially ask for one more code

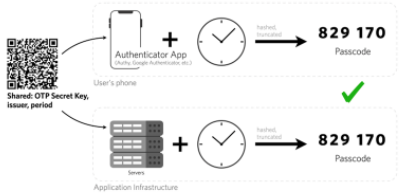


Image:https://www.twilio.com/docs/glossary/totp

Credits

- Original slide deck by Vitaly Shmatikov
- Expanded and updated by Nick Nikiforakis