CS 858 - User Authentication

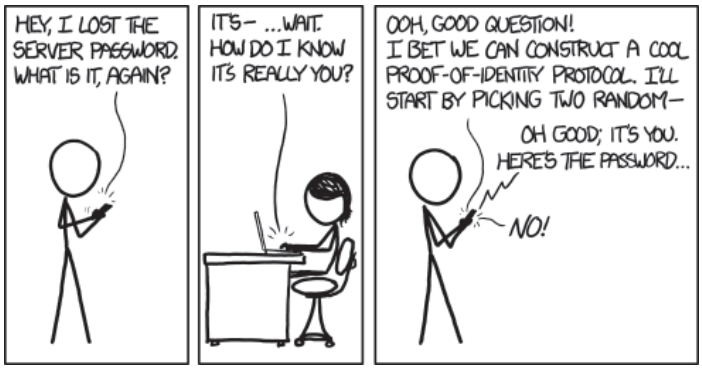
The Basics

Fall 2022

User authentication

- Computer systems often have to identify and authenticate users before authorizing them
- Identification: Who are you?
- Authentication: Prove it!
- Identification and authentication is easy among people that know each other
  - For your friends, you do it based on their face or voice
- More difficult for computers to authenticate people sitting in front of them
- Even more difficult for computers to authenticate people accessing them remotely

User authentication



https://xkcd.com/1121/

Authentication factors

- Three classes of authentication factors
- Something the user knows
  - Password, PIN, answer to "secret question"
- Something the user has
  - ATM card, badge, browser cookie, physical key, uniform, smartphone
- Something the user is
  - Biometrics (fingerprint, voice pattern, face,...)
  - Have been used by humans forever, but only recently by computers

Authentication factors

- Four classes of authentication factors
- Something the user knows
  - Password, PIN, answer to "secret question"
- Something the user has
  - ATM card, badge, browser cookie, physical key, uniform, smartphone
- Something the user is
  - Biometrics (fingerprint, voice pattern, face,...)
  - Have been used by humans forever, but only recently by computers
- Something about the user's context
  - Location, time, devices in proximity

Combination of auth. factors

- Different classes of authentication factors can be combined for more solid authentication
  - Two- or multi-factor authentication
- Using multiple factors from the same class might not provide better authentication
- "Something you have" can become "something you know"
  - Token can be easily duplicated, e.g., magnetic strip on ATM card
  - SMS message

Passwords

- Probably oldest authentication mechanism used in computer systems
- User enters user ID and password, maybe multiple attempts in case of error
- Many usability problems, such as
  - Entering passwords is inconvenient, in particular on small screens
  - Password composition/change rules
  - Forgotten passwords might not be recoverable
  - If password is shared among many people, password updates become difficult

Security problems with passwords

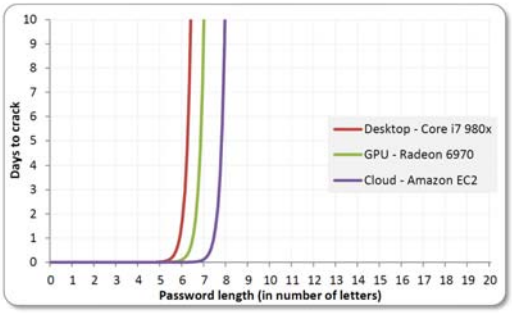
- If password is disclosed to unauthorized individual, the individual can immediately access protected resource
  - Unless we use multi-factor authentication
- Shoulder surfing
- Keystroke logging
- Interface illusions / Phishing
- Password re-use across sites
- Password guessing

Password guessing attacks

- Brute-force: Try all possible passwords using exhaustive search
- Can test 350 billion Windows NTLM passwords per second on a cluster of 25 AMD Radeon graphics cards
- Can try 95^8 combinations in 5.5 hours
- Enough to brute force every possible eight-character password containing upper- and lower-case letters, digits, and symbols

Brute-forcing passwords is exponential

http://erratasec.blogspot.ca/2012/08/common-misconceptions-of-password.html



Password guessing attacks

- Exhaustive search assumes that people choose passwords randomly, which is often not the case
- Attacker can do much better by exploiting this
- For example, assume that a password consists of a root and a pre- or postfix appendage
  - "password1", "abc123", "123abc"
- Root is from dictionaries (passwords from previous password leaks, names, English words, ...)
- Appendage is combination of digits, date, single symbol, ...
- >90% of 6.5 million LinkedIn password hashes leaked in June 2012 were cracked within six days

Password guessing attacks

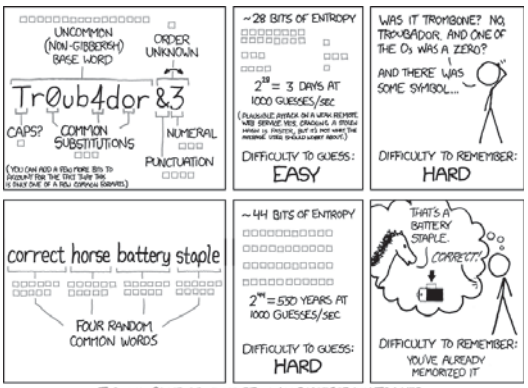
- So should we just give up on passwords?
- Attack requires that attacker has encrypted password file or encrypted document
  - Offline attack
- Instead, attacker might want to guess your banking password by trying to log in to your bank's website
  - Online attack
- Online guessing attacks are detectable
  - Bank shuts down online access to your bank account after n failed login attempts (typically $n \leq 5$)
  - But! How can an attacker circumvent this lockout?

Password hygiene

- Use a password manager to create and store passwords
  - At least for low- and medium-security passwords
  - All (most) eggs are now in one basket, so keep your computer's software up to date
  - Prevents password re-use across sites
- Use a pass phrase
  - Phrase of randomly chosen words, avoid common phrases (e.g., advertisement slogans)

Password strength

https://xkcd.com/936/



Password hygiene

- Have site-specific passwords
- Don't reveal passwords to others
  - In email or over phone
    - If your bank really wants your password over the phone, switch banks
  - Studies have shown that people disclose passwords for a cup of coffee, chocolate, or nothing at all
    - Caveat of these studies?
- Don't enter password that gives access to sensitive information on a public computer (e.g., Internet café) or over public networks.
  - Don't do online banking (or anything sensitive) on them

Advice for developers (NIST 2017)

- No password composition rules
  - Otherwise everybody uses the same simple tricks to follow rule
- At least 8 characters minimum length
- At least 64 characters maximum length
- Allow any characters, including space, Unicode, and emoji
- Black list frequently used or compromised passwords (from password leaks)
- Avoid password hints or "secret questions"

Advice for developers (NIST 2017)

- Don't ask users to periodically change passwords
  - Leads to password cycling and similar
    - "myFavoritePwd" -> "dummy" -> "myFavoritePwd"
    - goodPwd."1" -> goodPwd."2" -> goodPwd."3"
- Allow passwords to be copy-pasted into password fields
- Use two-factor authentication (but avoid SMS-based second factor)

Attacks on password files

- Website/computer needs to store information about a password in order to validate entered password
- Storing passwords in plaintext is dangerous, even when file is read protected from regular users
  - Password file might end up on backup tapes
  - Intruder into OS might get access to password file
  - System administrator has access to file and might use passwords to impersonate users at other sites
    - Many people re-use passwords across multiple sites

Cryptographic Tools

The following cryptographic tools are useful for storing information about passwords (see Module 5 for details):

- Cryptographic hash: Compute a fixed-length, deterministic output value from a variable-length input value. Given an output value, it is hard to find an input value with this output value, i.e., a cryptographic hash is not reversible.
- MAC: Same as a cryptographic hash, but it takes a secret key as another input value. Still deterministic and not reversible. Changing the secret key will change the output value.

Cryptographic Tools

- (Symmetric) encryption: Compute a non-deterministic output value that is an encryption of the input value under a secret key. Encryption is reversible if we know the secret key ("decryption").

Storing password fingerprints

- Store only a digital fingerprint of the password (using a cryptographic hash) in the password file
- When logging in, system computes fingerprint of entered password and compares it with user's stored fingerprint
- Still allows offline guessing attacks when password file leaks

Defending against guessing attacks

- UNIX makes guessing attacks harder by including user-specific salt in the password fingerprint
  - Salt is initially derived from time of day and process ID of /bin/passwd
  - Salt is then stored in the password file in plaintext
- Two users who happen to have the same password will likely have different fingerprints
- Makes guessing attacks harder, can't just build a single table of fingerprints and passwords and use it for any password file

Defending against guessing attacks

- Don't use a standard cryptographic hash (like SHA-1 or SHA-512) to compute the stored fingerprint
- They are relatively cheap to compute (microseconds)
- Instead use an iterated hash function that is expensive to compute (e.g., bcrypt) and maybe also uses lots of memory (e.g., scrypt)
  - Hundreds of milliseconds
- This slows down a guessing attack significantly, but is barely noticed when a users enters his/her password

Defending against guessing attacks

- An additional defense is to use a MAC, instead of a cryptographic hash
- A MAC mixes in a secret key to compute the password fingerprint
- If the fingerprints leak, guessing attacks aren't useful anymore
- Can protect the secret key by embedding it in tamper resistant hardware
- If the key does leak, the scheme remains as secure as a scheme based on a cryptographic hash

Password Recovery

- A password cannot normally be recovered from a hash value (fingerprint)
- If password recovery is desired, it is necessary to store an encrypted version of the password in the password file
- We need to keep encryption key away from attacker

Password Recovery

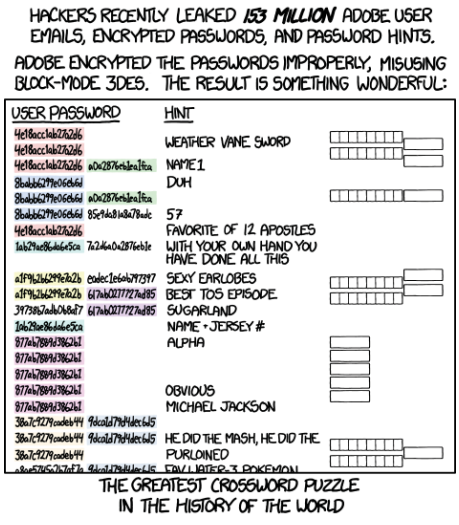
- As opposed to fingerprints, this approach allows the system to (easily) re-compute a password if necessary
  - E.g., have system email password in the clear to predefined email address when user forgets password
- There are many problems with this approach!
- Password reset is more common now.

The Adobe Password Hack (November 2013)

- In November 2013, 130 million encrypted passwords for Adobe accounts were revealed.
- The encryption mechanism was the following:
  1. First a NUL byte was appended to the password.
  2. Next, additional NUL bytes were appended as required to make the length a multiple of 8 bytes.
  3. Then the padded passwords were encrypted 8 characters at a time using a fixed key. (This is called ECB mode and it is the weakest possible encryption mode.)
- The password hints were not encrypted.
- It turns out that many passwords can be decrypted, without breaking the encryption and not knowing the key.

The Adobe Password Hack (cont.)

Adobe password data			Password hint
110edf2294fb8bf4	->	numbers 123456	
110edf2294fb8bf4	->	==123456	❶ 123456
110edf2294fb8bf4	->	c'est "123456"	
8fda7e1f0b56593f e2a311ba09ab4707	->	numbers	
8fda7e1f0b56593f e2a311ba09ab4707	->	1-8	❷ 12345678
8fda7e1f0b56593f e2a311ba09ab4707	->	8digit	
2fca9b003de39778 e2a311ba09ab4707	->	the password is password	
2fca9b003de39778 e2a311ba09ab4707	->	password	❸ password
2fca9b003de39778 e2a311ba09ab4707	->	rhymes with assword	
e5d8efed9088db0b	->	q w e r t y	
e5d8efed9088db0b	->	ytrewq tagurpidi	❹ qwerty
e5d8efed9088db0b	->	6 long qwert	
ecba98cca55eabc2	->	sixxone	
ecba98cca55eabc2	->	1*6	❺ 111111
ecba98cca55eabc2	->	sixones	



http://xkcd.com/1286

Interception attacks

- Attacker intercepts password while it is in transmission from client to server
- One-time passwords make intercepted password useless for later logins
  - Fobs (e.g., RSA SecurID), Authenticator apps
  - Challenge-response protocols

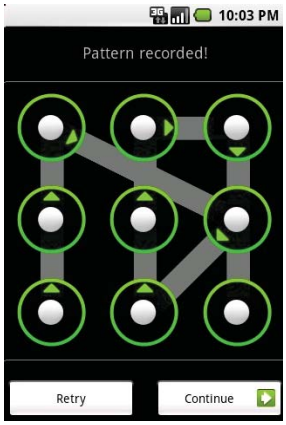
Challenge-response protocols

- Server sends a random challenge to a client
- Client uses challenge and password to compute a one-time password
- Client sends one-time password to server
- Server checks whether client's response is valid
- Given intercepted challenge and response, attacker might be able to brute-force password

Interception attacks

- On the web, passwords may still be transmitted in plaintext
  - Sometimes, digital fingerprint of them
  - Encryption (TLS, see later) protects against interception attacks on the network
- There are cryptographic protocols (e.g., SRP) that make intercepted information useless to an attacker
- Alternative solutions are difficult to deploy
  - Patent issues, changes to HTTP protocol, hardware
- And don't help against interception on the client side
  - Malware

Android unlock patterns



Graphical passwords

- Graphical passwords are an alternative to text-based passwords
- Multiple techniques, e.g.,
  - User chooses a picture; to log in, user has to re-identify this picture in a set of pictures
  - User chooses set of places in a picture; to log in, user has to click on each place
- Issues similar to text-based passwords arise
  - E.g., choice of places is not necessarily random
- Shoulder surfing becomes a problem
- Ongoing research

Server authentication

- With the help of a password, system authenticates user (client)
- But user should also authenticate system (server) else password might end up with attacker!
- Classic attack:
  - Program displays fake login screen
  - When user "logs in", programs prints error message, sends captured user ID/password to attacker, and ends current session (which results in real login screen)
  - That's why Windows trains you to press <CTRL-ALT-DELETE> for login, key combination cannot be overridden by attacker
- Today's attack:
  - Phishing

Biometrics

- Biometrics have been hailed as a way to get rid of the problems with password and token-based authentication
- Unfortunately, they have their own problems
- Idea: Authenticate user based on physical characteristics
  - Fingerprints, iris scan, voice, handwriting, typing pattern,...
- If observed trait is sufficiently close to previously stored trait, accept user
  - Observed fingerprint will never be completely identical to a previously stored fingerprint of the same user

Local vs. remote authentication

- Biometrics work well for local authentication, but are less suited for remote authentication or for identification
- In local authentication, a guard can ensure that:
  - I put my own finger on a fingerprint scanner, not one made out of gelatin
  - I stand in front of a camera and don't just hold up a picture of somebody else
- In remote authentication, this is much more difficult

Authentication vs. identification

- Authentication: Does a captured trait correspond to a particular stored trait?
- Identification: Does a captured trait correspond to any of the stored traits?
  - Identification is an (expensive) search problem, which is made worse by the fact that in biometrics, matches are based on closeness, not on equality (as for passwords)
- False positives can make biometrics-based identification useless
  - False positive: Alice is accepted as Bob
  - False negative: Alice is incorrectly rejected as Alice

Biometrics-based identification

- Example (from Bruce Schneier's "Beyond Fear"):
  - Face-recognition software with (unrealistic) accuracy of 99.9% is used in a football stadium to detect terrorists
    - 1-in-1,000 chance that a terrorist is not detected
    - 1-in-1,000 chance that innocent person is flagged as terrorist
  - If one in 10 million stadium attendees is a known terrorist, there will be 10,000 false alarms for every real terrorist
  - Remember "The Boy Who Cried Wolf"?
  - Another example of the base rate fallacy (see Module 2)

Other problems with biometrics

- Privacy
  - Why should my employer (or a website) have information about my fingerprints, iris,..?
    - Aside: Why should a website know my date of birth, my mother's maiden name,... for "secret questions"?
  - What if this information leaks? Getting a new password is easy, but much more difficult for biometrics
- Accuracy: False negatives are annoying
  - What if there is no other way to authenticate?
  - What if I grow a beard, hurt my finger,...?

Other problems with biometrics

- Secrecy: Some of your biometrics are not particularly secret
  - Face, fingerprints,...
- Legal protection: The law may allow the police to put your finger on your phone's fingerprint reader (or simply hold your phone's camera in front of you). But the law may protect you from you having to reveal your password (depending on the country).