Authentication: Password & Biometrics

Shuai Wang

香 港 科 技 大 學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

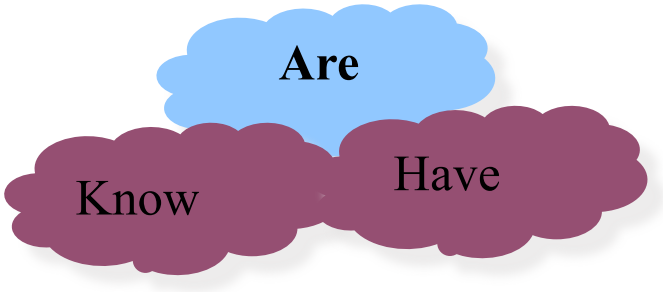
Some of the slides are from Mark Stamp.

Access Control

- Two parts to access control
- Authentication: are you who you say you are?
  - Determine whether access is allowed
  - Authenticate human to machine
  - Or authenticate machine to machine
- Authorization: are you allowed to do that?
  - Once you have access, what can you do?
  - Enforces limits on actions

Are You Who You Say You Are?

- Authenticate a human to a machine?
- Can be based on…
  - Something you **know**
    - For example, a password
  - Something you **have**
    - For example, a smartcard
  - Something you **are**
    - For example, your fingerprint



Something You Know

- Passwords
- Lots of things act as passwords!
  - PIN
  - Social security number
  - Mother’s maiden name
  - Date of birth
  - Your first boss’s name
  - Name of your pet, etc.

Trouble with Passwords

- “Passwords are one of the biggest practical problems facing security engineers today.”
- “Humans are incapable of securely storing high-quality cryptographic keys, …, They are also large, expensive to maintain, difficult to manage, and they pollute the environment. It is astonishing that these devices continue to be manufactured and deployed.”

How to solve the problem?

Why Passwords?

- Why is “something you know” more popular than “something you have” and “something you are”?
- **Cost**: passwords are free
- **Convenience**: easier for sysadmin to reset pwd than to issue a new thumb

Keys vs Passwords

- **Crypto keys**
- Spse key is 64 bits
- Then 2^{64} keys
- Choose key at random…
- …then attacker must try about 2^{63} keys

- **Passwords**
- Spse passwords are 8 characters, and 256 different characters
- Then $256^8 = 2^{64}$ pwds?!
- But users do not select passwords at random
- Attacker has far less than 2^{63} pwds to try (**dictionary attack**)

dictionary attack: tries only those possibilities which are deemed most likely to succeed, typically derived from a list of words such as in a dictionary

Good and Bad Passwords

- Bad passwords
  - frank
  - Fido
  - Password
  - incorrect
  - Pikachu
  - 88195277
  - AustinStamp
- Good Passwords?
  - jflej,43j-EmmL+y
  - 09864376537263
  - 0nceuP0nAt1m8 ← Passphrase
  - P0kem0n

Passphrase:

- longer (10~30 letters) to make brute force attacks difficult
- if well chosen, they will not be found in any phrase or quote dictionary
- **Structured to be more easily memorable** than passwords

Password Experiment

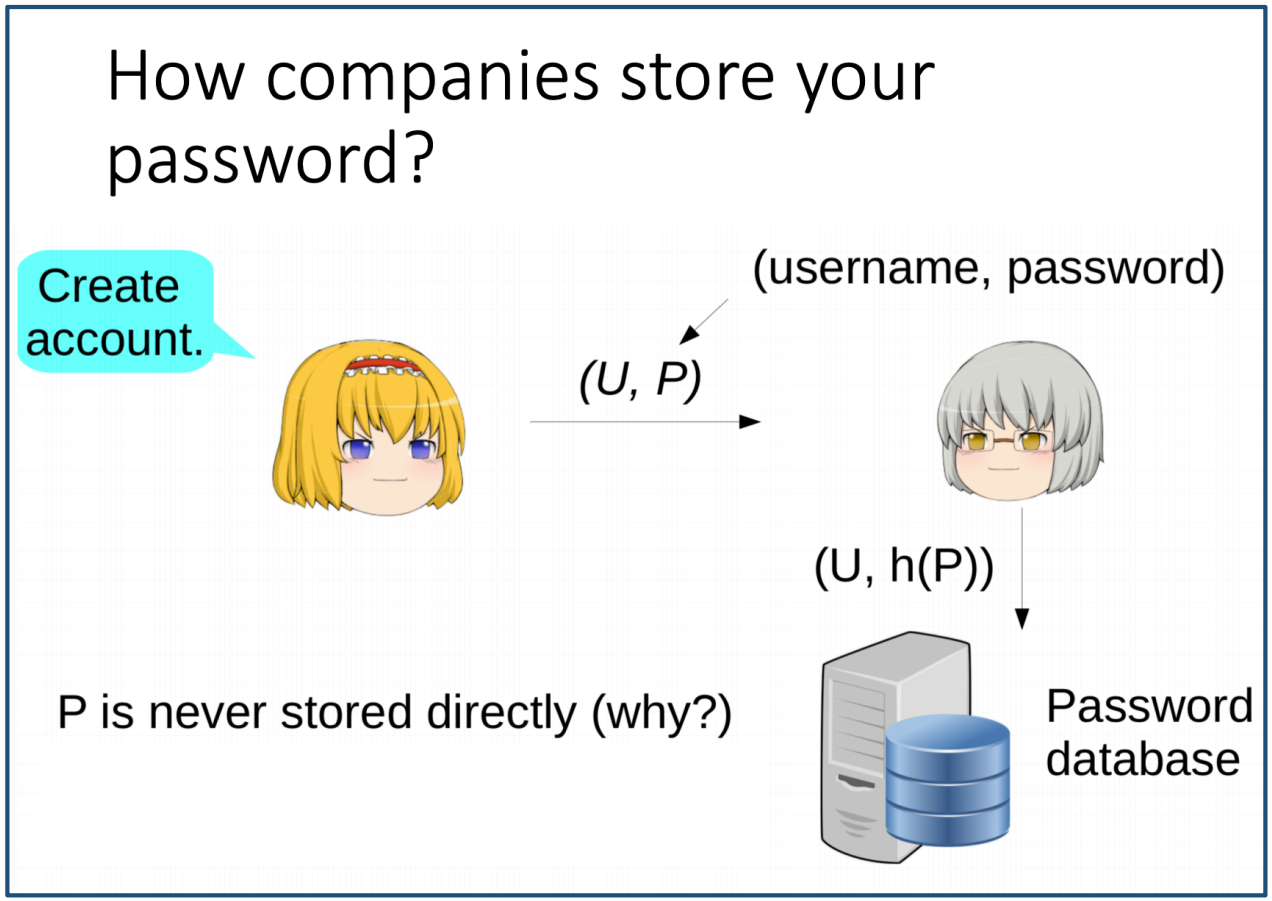
- Three groups of users — each group advised to select passwords as follows
  - **Group A:** At least 6 chars, 1 non-letter
  - winner → **Group B:** Password based on passphrase
  - **Group C:** 8 random characters
- Results
  - **Group A:** About 30% of pwds easy to crack
  - **Group B:** About 10% cracked
    - Passwords easy to remember
  - **Group C:** About 10% cracked
    - Passwords hard to remember

Attacks on Passwords

- Attacker could…
  - Target one particular account
  - Target any account on system
  - Target any account on any system
  - Attempt denial of service (DoS) attack
    - What's the connection here?
    - Think about the ATM machine…

Password File?

- Bad idea to store passwords in a file
- But we need to verify passwords
- Solution?



Dictionary Attack towards a File of hashed passwords

- Attacker pre-computes $h(x)$ for all x in a **dictionary** of common passwords
- Suppose the attacker gets access to password file containing hashed passwords
  - She only needs to compare hashes to her pre-computed dictionary
  - After one-time work of computing hashes in dictionary, actual attack is trivial
- Can we prevent this search-based attack?
  - Or at least make it more difficult?

Salt (Random Chosen Value)

- Hash password with **salt**
- Choose random salt s and compute

$$y = h(\text{password}, s)$$

  and store (s,y) in the password file
- Note that the salt s is not secret
- Still easy to verify salted password
- But lots more work for the attacker
  - Why?
  - Must recompute hash for each user

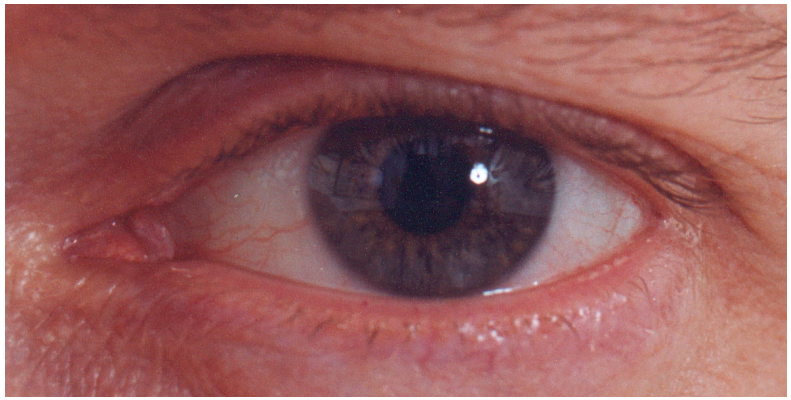
Other Password Issues

- Too many passwords to remember
  - Results in password reuse
  - Why is this a problem?
- Who suffers from bad password?
  - Login password vs ATM PIN
- Failure to change default passwords
- Social engineering…
- Error logs may contain “almost” passwords
- Bugs, keystroke logging, spyware, etc.

Passwords

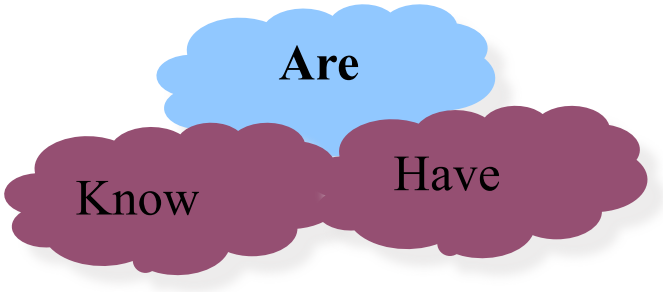
- **Password attacks are too easy**
  - Often, one weak password will break security
  - Users choose bad passwords
  - Social engineering attacks, etc.
- Passwords are a **BIG** security problem
  - And will continue to be a problem
- Popular password cracking tools
  - Password Crackers
  - Password Portal
  - L0phtCrack and LC4 (Windows)
  - John the Ripper (Unix)
- Admin should use these tools to test weak password.

Biometrics



Something You Are

- Biometric
  - **"You are your key"**
- Examples
  - Fingerprint
  - Handwritten signature
  - Facial recognition
  - Speech recognition
  - Walking (gait) recognition
  - Many more!



Why Biometrics?

- May be better than passwords
- But, cheap and reliable biometrics needed
  - Today, an active area of research
- Biometrics **are** used in security today
  - Fingerprint to unlock car door
  - Palm print for secure entry

Ideal Biometric

- **Universal** — applies to (almost) everyone
  - In reality, no biometric applies to everyone
- **Distinguishing** — distinguish with certainty
  - In reality, cannot hope for 100% certainty
- **Permanent** — physical characteristic being measured never changes
  - In reality, OK if it to remains valid for long time
- **Collectable** — easy to collect required data
  - Depends on whether subjects are *cooperative*
- Also, safe, user-friendly

Enrollment vs Recognition

- Enrollment phase
  - Subject's biometric info put into database
  - Must carefully measure the required info
  - OK if slow and repeated measurement needed
  - Must be very precise
  - May be a weak point in real-world use
- Recognition phase
  - Biometric detection, when used in practice
  - Must be quick and simple
  - But must be reasonably accurate

Biometric Errors

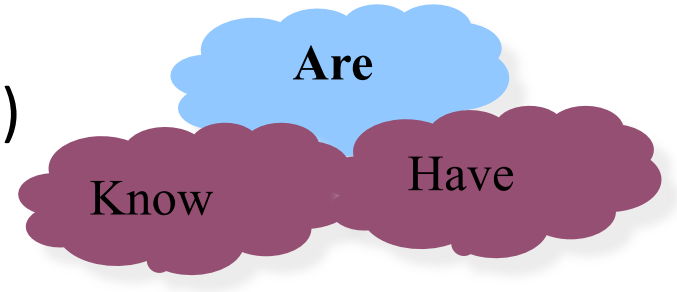
- **Fraud rate** versus **insult rate**
  - Fraud **(false negative)** — Attacker mis-authenticated as Alice
  - Insult **(false positive)** — Alice not authenticated as Alice
- For any biometric, generally speaking, can decrease fraud or insult, but other one will increase
- For example
  - 99% voiceprint match ⇒ low fraud, high insult
  - 30% voiceprint match ⇒ high fraud, low insult
- **Equal error rate:** rate where fraud == insult
  - A way to ***compare*** different biometrics

Biometrics: The Bottom Line

- Biometrics are hard to forge
- But attacker could
  - Steal Alice's thumb
  - Photocopy Bob's fingerprint, eye, etc.
  - Subvert software, database, "trusted base" …
- And how to revoke a "broken" biometric?
- **Biometrics are not foolproof**

Are You Who You Say You Are?

- Something in your possession
- Examples include following…
  - Car key
  - Laptop computer (or MAC address)
  - Password generator (next)
  - ATM card, smartcard, etc.



2-factor Authentication

- Requires any 2 out of 3 of
  - Something you **know**
  - Something you **have**
  - Something you **are**

Examples

- Credit card: Card and signature
- ATM: Card and PIN