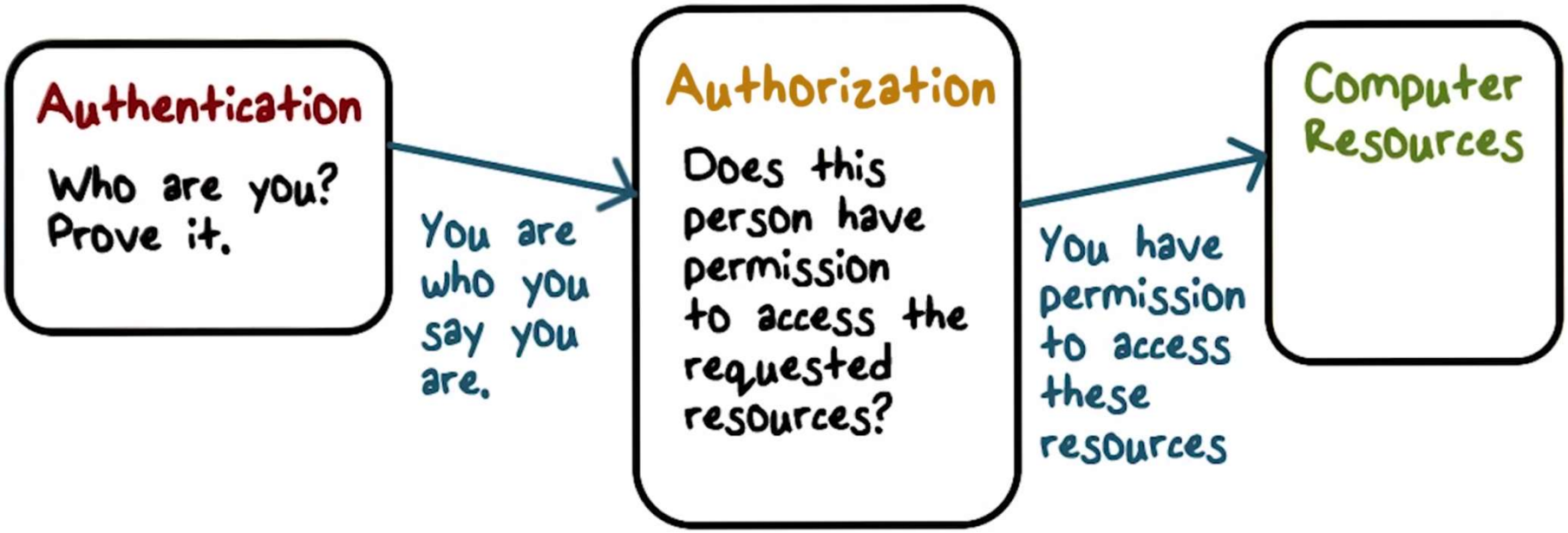
Authentication
Lesson Introduction

- Understand the **importance of authentication**
- Learn **how authentication can be implemented**
- Understand **threats to authentication**

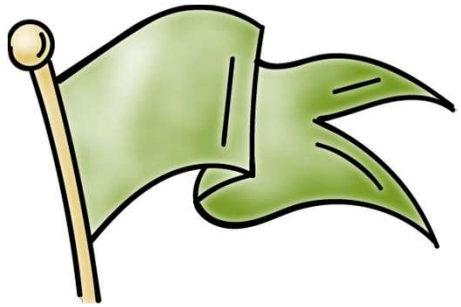
What is Authentication?



What is Authentication?

- OS (TCB) needs to know **who makes a request** for a protected resource
- A process that makes the request does it **on behalf of a certain user**, subject or principal
- Authentication helps us answer the question: **on whose behalf the requesting process runs?**
- Includes claims about an identity and verification of the claimed identity of **the user who wants to gain access to system and resource**

Authentication Goals



User/principal associated with an identity
should be able to successfully authenticate itself

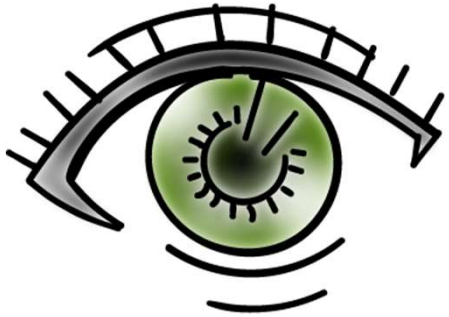
User/principal not associated with the identity should not be able to authenticate itself

How is Authentication Implemented?

Three basic methods:

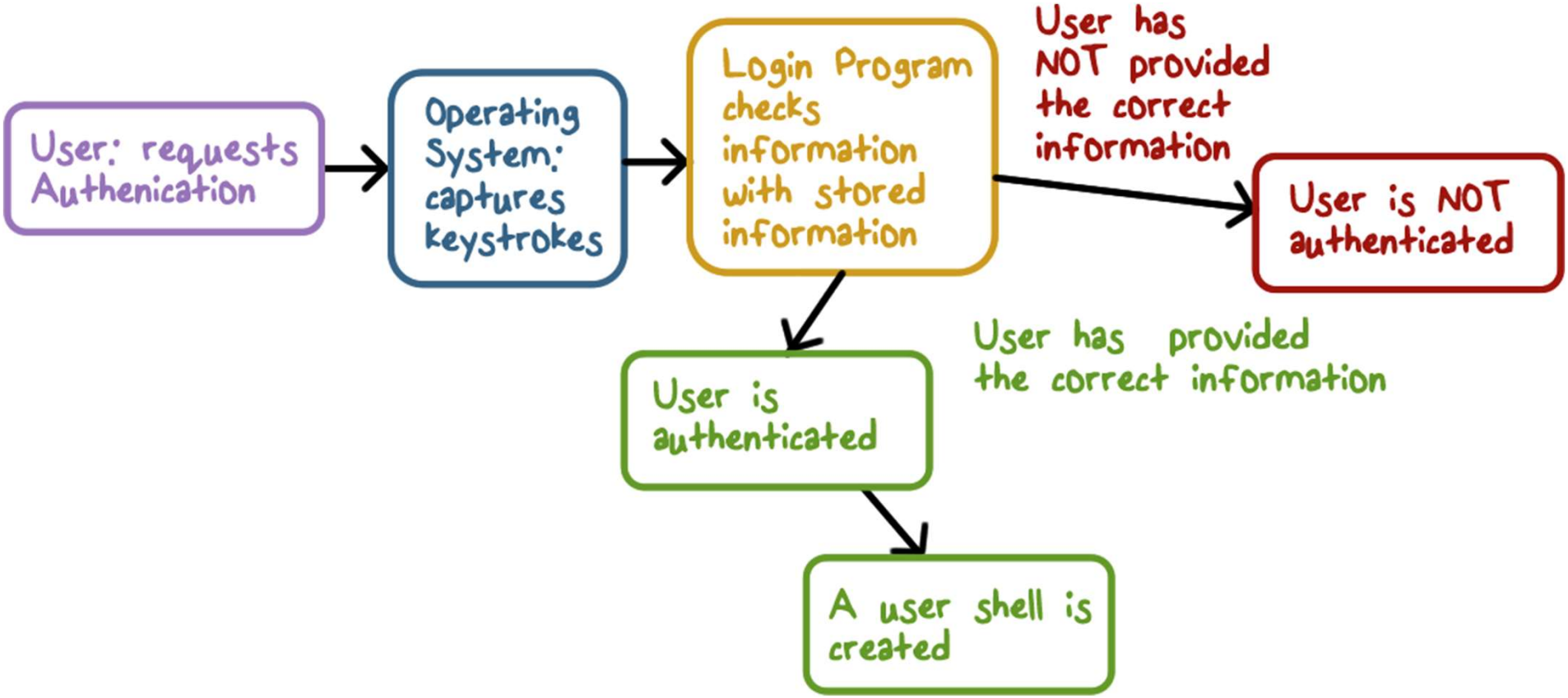
- Something a user **knows**

- Something a user **has**

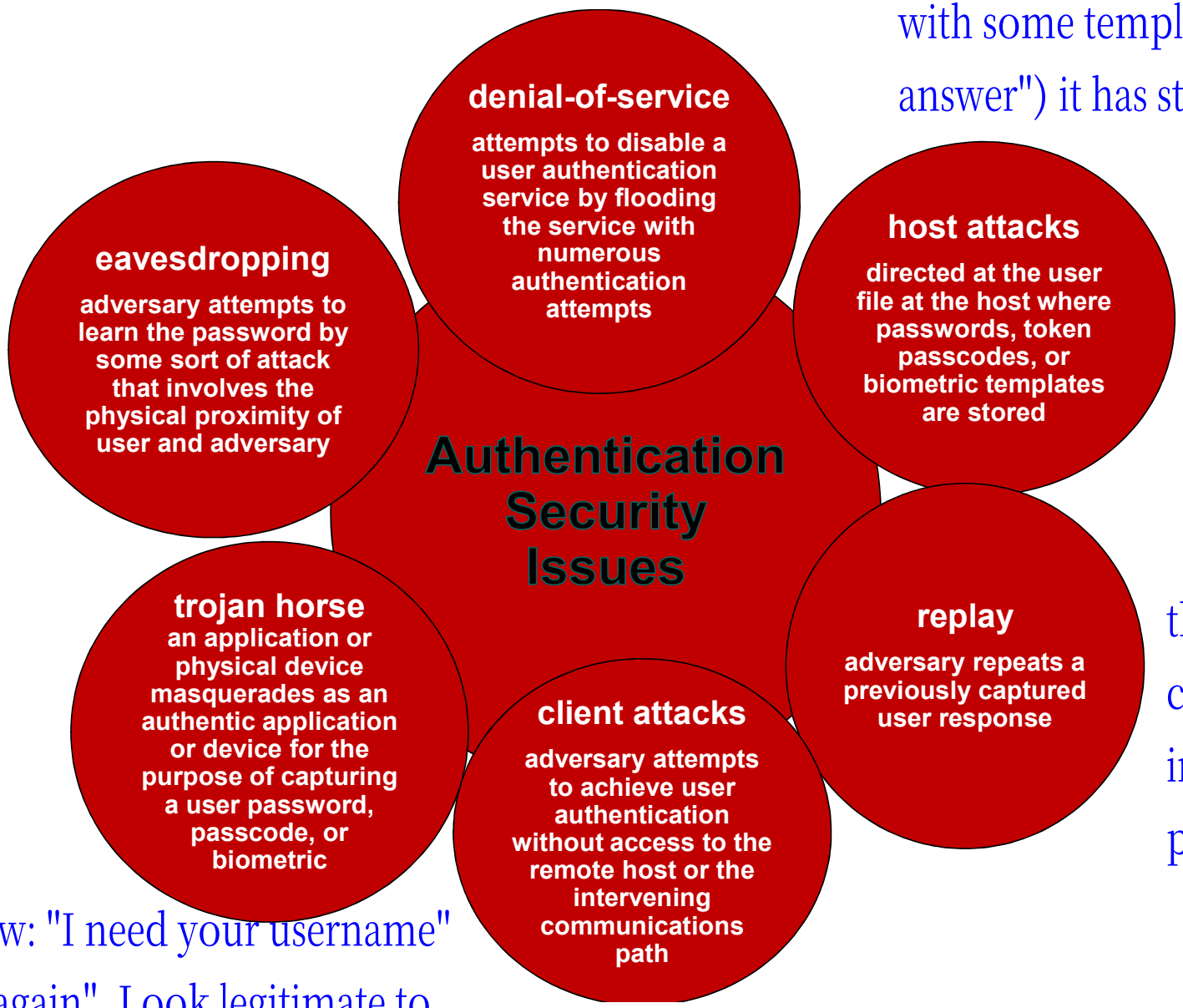


- Something a user **is**

How is Authentication Implemented?



the host compare what you input with some templates ("standard answer") it has stored



through some ways copy your login information and paste it back.

pop-up window: "I need your username" and password again". Look legitimate to the users, but it was someone else trying to get information from you.

attempting to be you

Threat Modeling of the Password Method

- **Guessing the password** for a given user allows impersonation
- **Impersonating** a real login program
- **Keylogging** to steal a password

Implementing Password Authentication

How do we check the password supplied with a user id?

Method 1 - store a list of passwords, one for each user in the system file.

- The file is readable only by the root/admin account
- What if the permissions are set incorrectly?
- Why should admin know the passwords?
- If security is breached, the passwords are exposed to an attacker.

Implementing Authentication

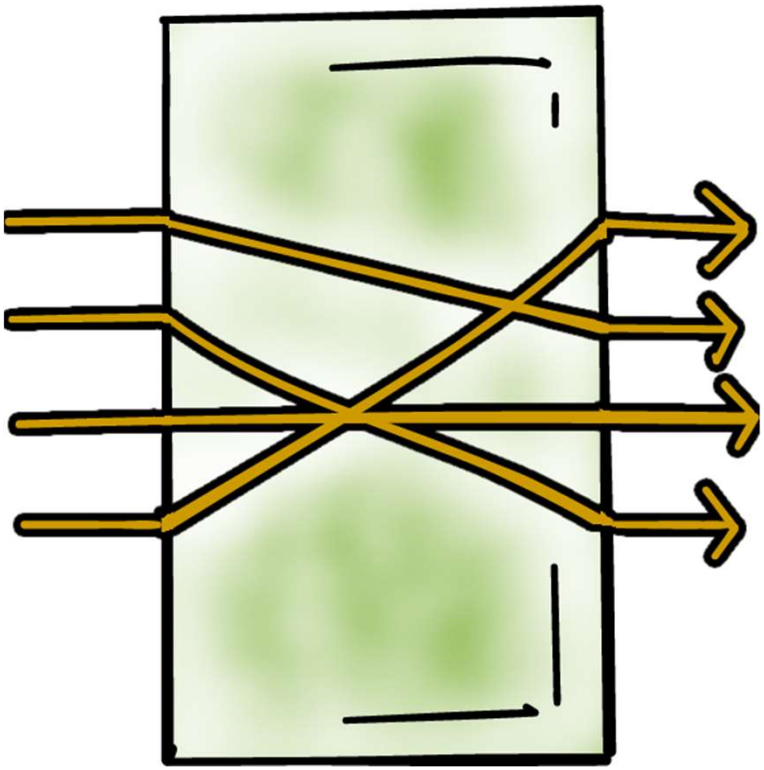
How do we check the password supplied with a user id?

Method 2 - do not store passwords, but store something that is derived from them

- Use a one-way hash function and store the result
- The password file is readable only for root/admin

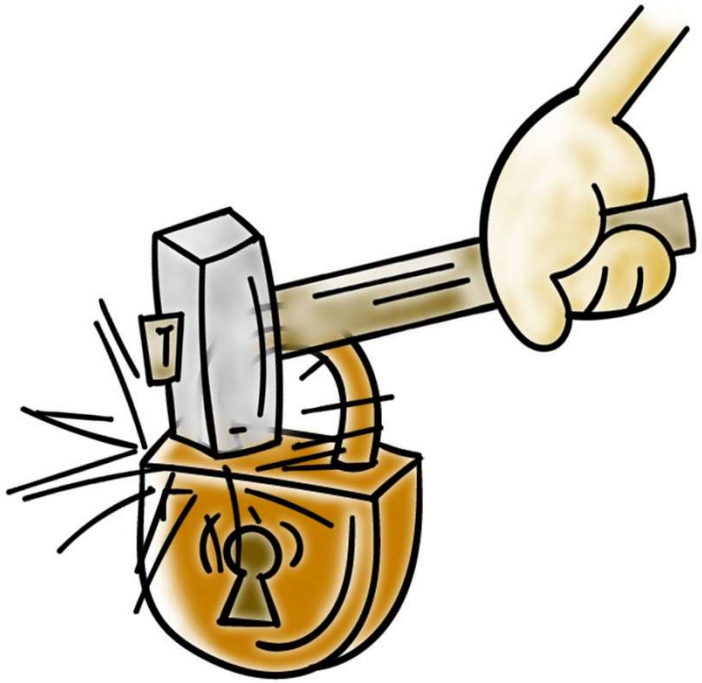
Hash Functions

Hash Functions & Threats



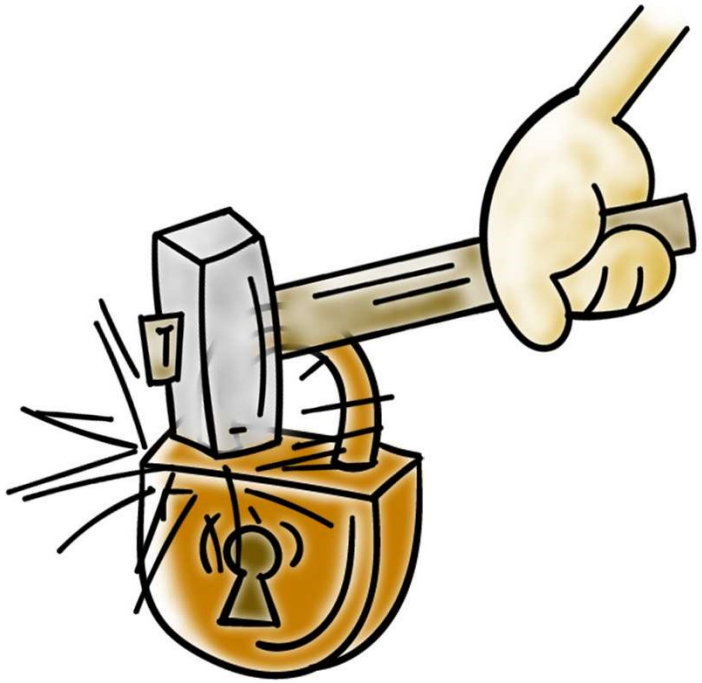
- We **assume a one-way property** for hash functions
- If we **know common passwords**, we can determine their hash
- For dictionary and offline attacks, we have the **hash values and plenty of time** *to test* for matches

Brute Force Guessing of Passwords



- Publicly available software can do **10^8 MD5 hashes/sec on a GPU**
- Six random upper case/lower case/digits then 62^6 possible passwords, **about *10 minutes***
- Eight random characters increases it to about **six days**

Brute Force Guessing of Passwords

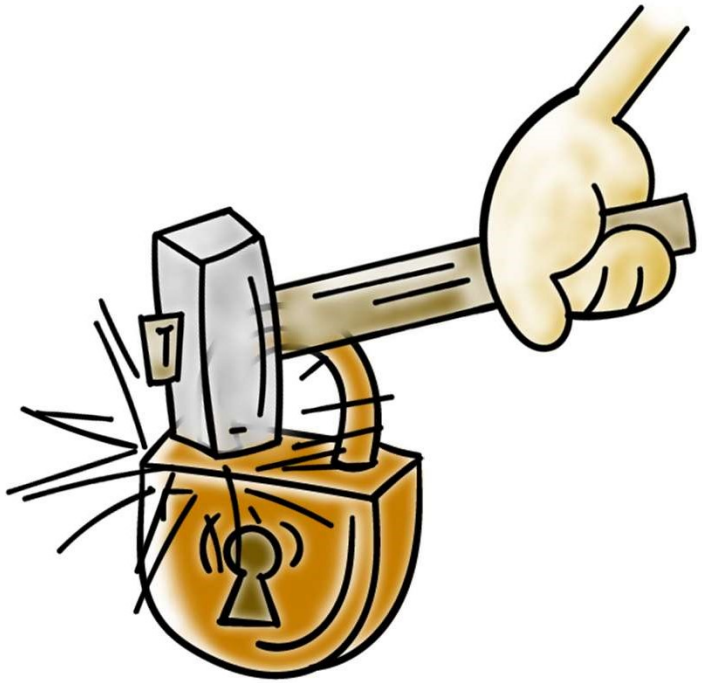


Passwords are not really random

To reduce the work required for a brute force attack:

- Try the popular passwords first
- Create a rainbow table

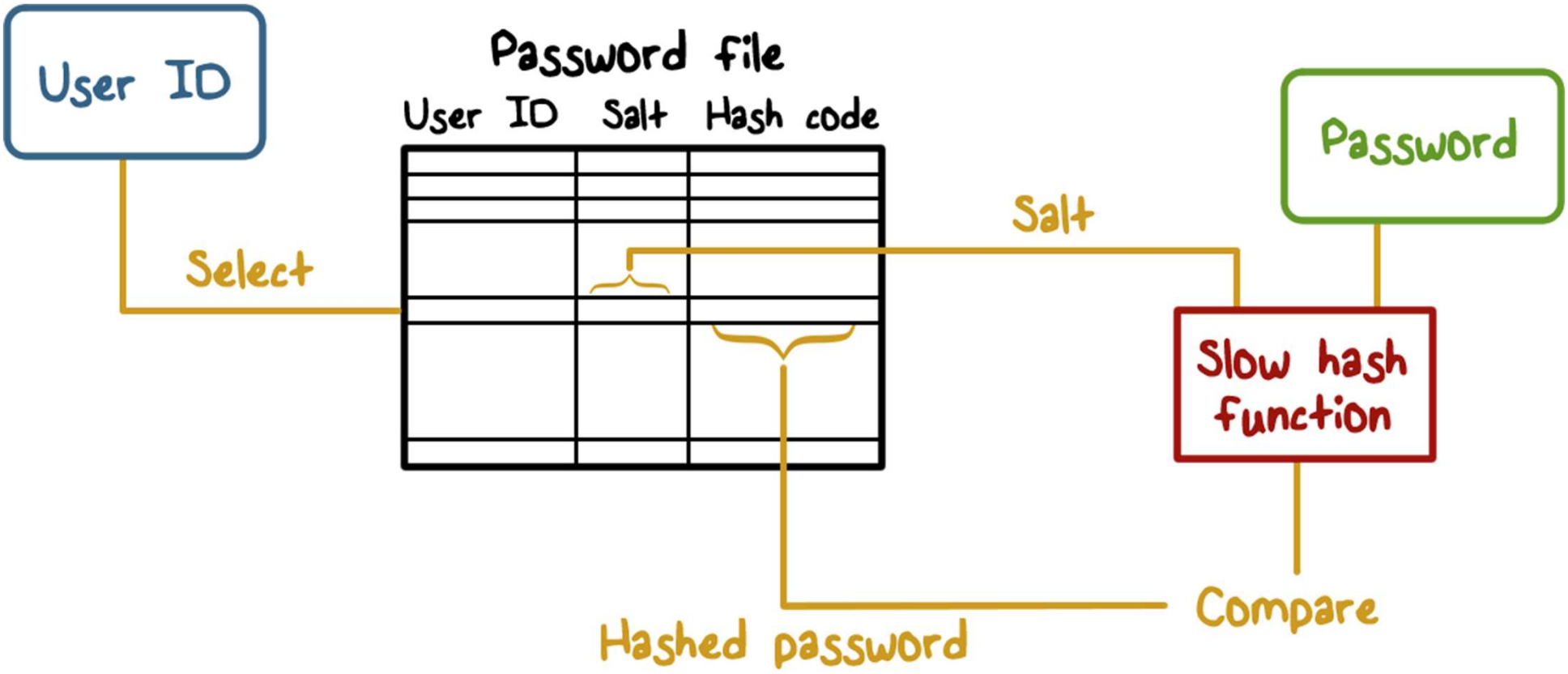
Brute Force Guessing of Passwords



What if two users pick the same password?

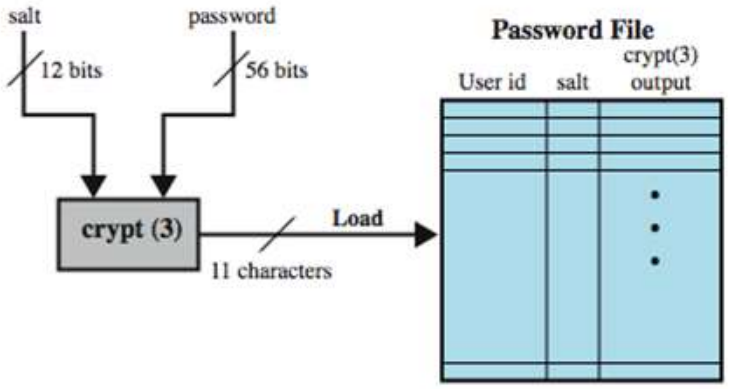
- **Add a random salt** before hashing
- **Store the salt** with the hashed value
- **Check** by using the salt with the typed password

Brute Force Guessing of Passwords

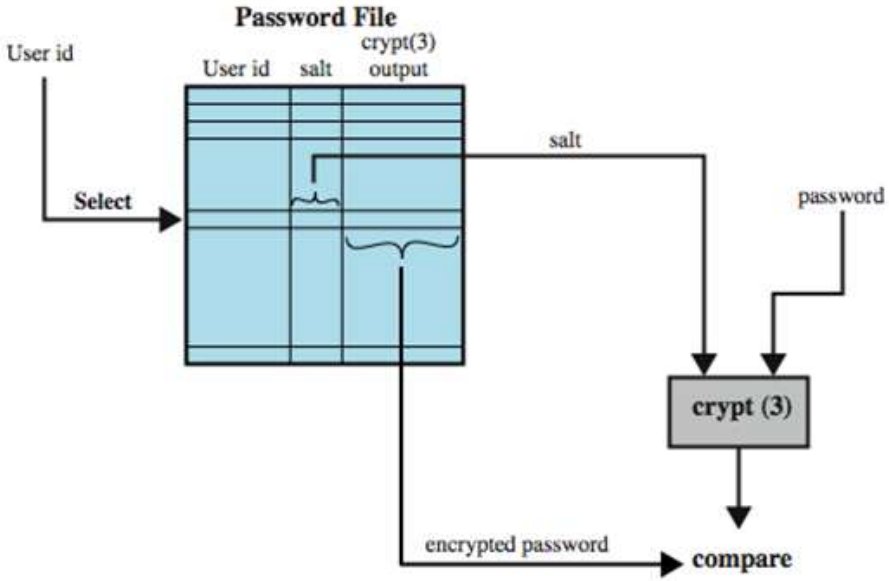


Use of Hashed Passwords

- prevents duplicate passwords from being visible in the password file.
- greatly increases the difficulty of offline dictionary attacks. For a salt of length *b bits, the number of possible passwords is increased by a factor of* 2^b.
- becomes nearly impossible to find out whether a person with passwords on two or more systems has used the same password on all of them.



(a) Loading a new password



(b) Verifying a password

Improved Password Implementations

- much stronger hash/salt schemes available for Unix
- recommended hash function is based on MD5
  - salt of up to 48-bits
  - password length is unlimited
  - produces 128-bit hash
  - uses an inner loop with 1000 iterations to achieve slowdown
- OpenBSD uses Blowfish block cipher based hash algorithm called Bcrypt

Problems with Passwords

- As password length and complexity increases, **usability suffers**
- Phishing and social engineering – **users do not authenticate who is asking for a password**.
- Once a password is stolen, **it can be used many times**
  - This is why there are policies that say passwords be changed frequently
- **Humans have a hard time remembering** lots of passwords. Usable passwords are easy to guess.

Problems with Passwords

Sys Administrators:

- Never store passwords in the clear
- Store only hashed values generated with a random salt and limit access to them
- Avoid general purpose fast hash functions

Users:

- Use password managers

Other Authentication Methods

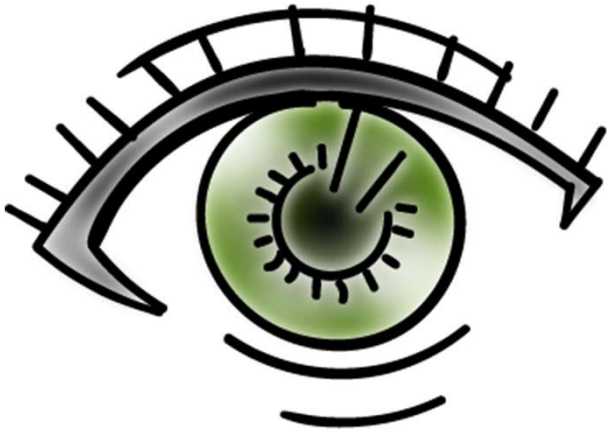
Something you have:

Tokens, smart cards

- You must have them
- May require additional hardware (e.g., readers)
- How does it implement authentication (challenge/response)
- Cost and misplaced trust (RSA SecureID master key breach)

Other Authentication Methods

Something You Are:

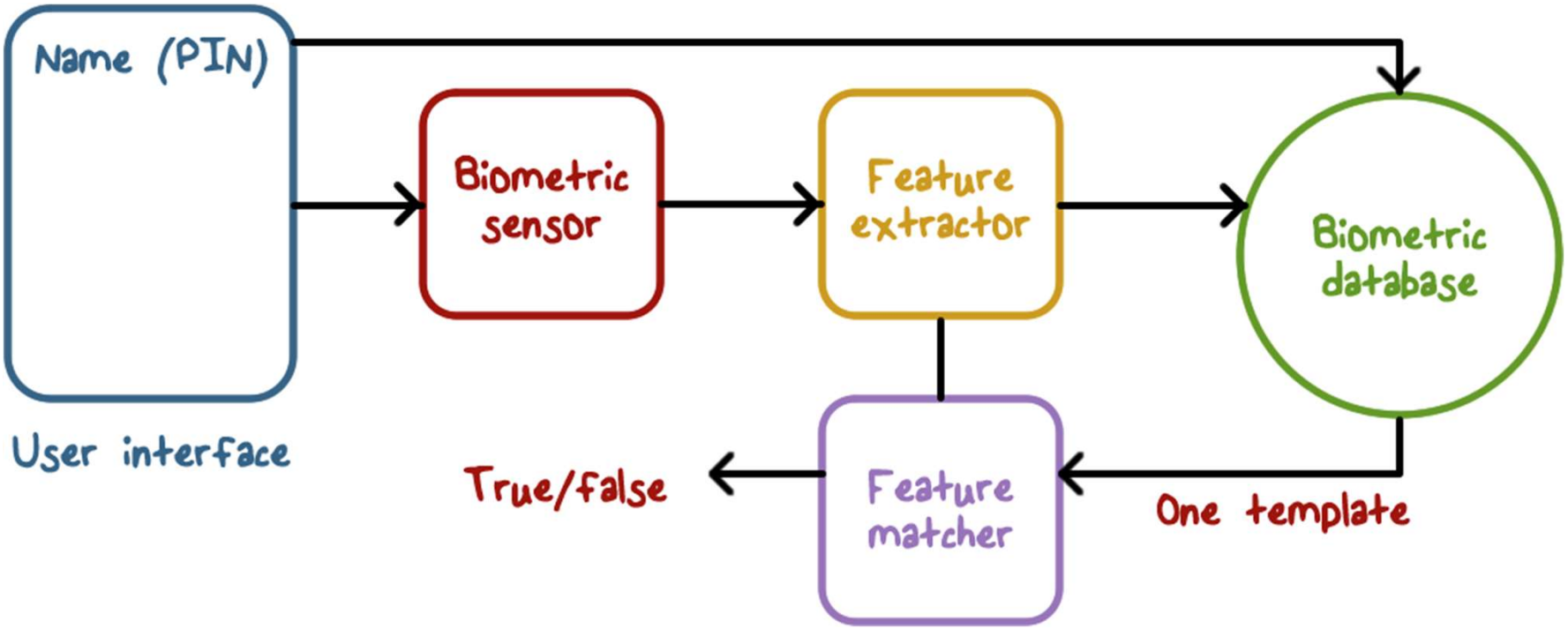


- **Various biometrics**
  - Fingerprints (finger swipes)
  - Keystroke dynamics
  - Voice
  - Retina scans

Do you get the same biometric measurement each time?

- Probability distribution or a range for feature values
- False positives and negatives

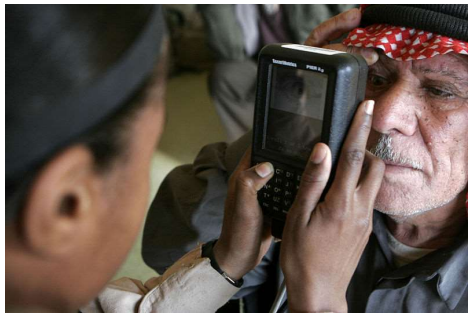
Implementing Biometric Authentication



Candidates for Biometric IDs

- Fingerprints
- Retinal/iris scans
- DNA
- “Blue-ink” signature
- Voice recognition
- Face recognition
- Gait recognition

Public domain image from
http://commons.wikimedia.org/wiki/File:Fingerprint_Arch.jpg

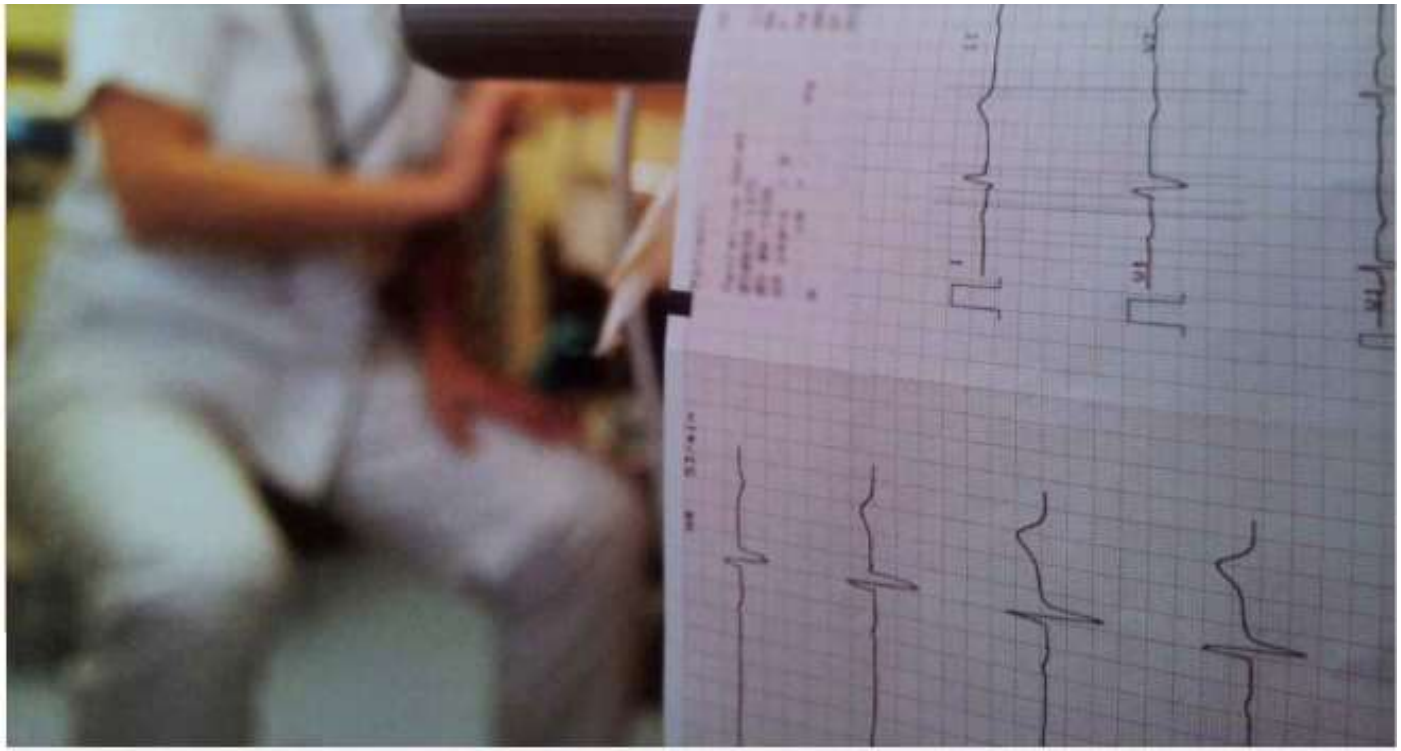


Public domain image from
http://commons.wikimedia.org/wiki/File:Retinal_scan_securimetrics.jpg

Public domain image from
http://commons.wikimedia.org/wiki/File:CBP_chemist_reads_a_DNA_profile.jpg

Ear Ear Travis Isaacs/Flickr, CC BY 2.0

You heard it here: The shape of your ear is just as distinguishing as your fingerprints; no two ears, even on the same person, are alike. Startup Descartes Biometrics has come up with an app that can identify smartphone users by the way they press their phone to their ear and cheek—though its less-than-consistent recognition means that perhaps this particular app isn't yet ready for prime time.

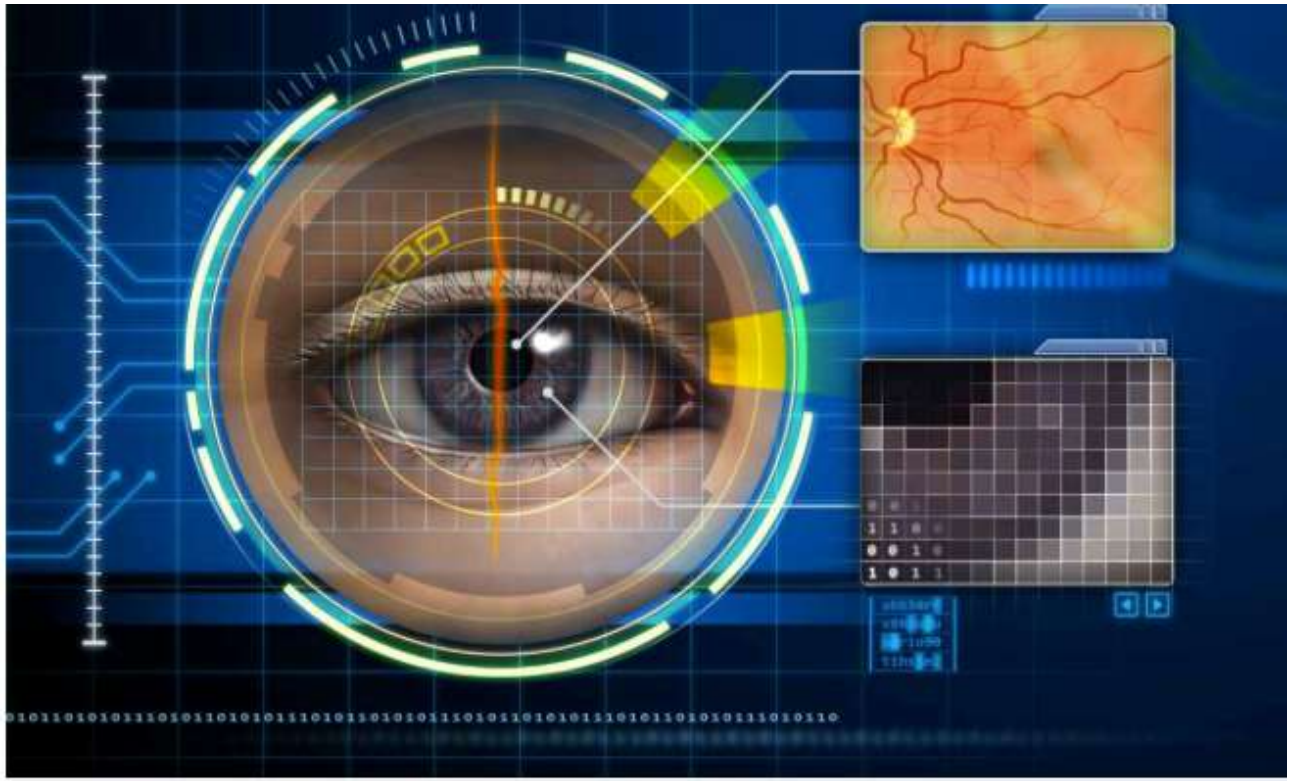


Follow Your Heart Helge V. Keitel/Flickr, CC BY 2.0

They say the heart always knows the truth, so it shouldn't be surprising that someone's come up with a way to prove your identity based on it. The Nymi is an in-development wristband that takes an electrocardiogram (ECG)—measuring the electrical signal generated by your heart's activity—and uses it to authenticate your identity. You can then use the Nymi as a secure token for unlocking access to other devices, such as smartphones and computers. To date, identifying people by ECG is less proven than fingerprints or iris/retina recognition, but given the burgeoning popularity of smart devices that measure your heart rate, it could end up being a convenient method of authentication.

Butt Biometrics Advanced Institute of Industrial Technology

I suppose you could say there's just *one* 'but' about this biometric authentication method—and it's your posterior. Turns out your keister—or, more specifically, the way you sit—can be used to identify you. One team of researchers has created a prototype of a car seat that can tell who's sitting in it. It's not only great for making sure that only you (or, presumably, your family) can start your car, but also potentially handy for ensuring that your seat, mirrors, and other preferences are automatically adjusted for you.



The Eye (Movements) Have It Dreamstime

Authentication via parts of the eye, like the retina or iris, has been around for a while, but an Israeli company wants to use the unique *movements* of your eyes to identify you. It seems that we move our eyes in predictable patterns when doing certain tasks, such as following an icon across a screen. The advantages of the system are that it's tough to fool, since it requires a real-time response to a stimulus, rather than a static factor like a fingerprint, and it's fairly easy to implement. The downside, I imagine, is that it requires eye contact (which may not be easy when you're driving, for instance) and is probably a little slower than using something like a fingerprint.

The Nose Knows Eden, Janine and Jim/Flickr. CC BY 2.0

Not only is your olfactory organ good for smelling, but British researchers have established that it's also a handy way to tell you apart from your neighbor. Like your ears, your nose is distinct—probably belonging to one of six common nose types—and is unlikely to be mistaken for anybody else's. It's also easy to recognize, though changing your nose is hardly as tough as changing, say, your eyes. Hollywood can vouch for that.

You're So Vein West Midlands Police/Flickr, CC BY-SA 2.0

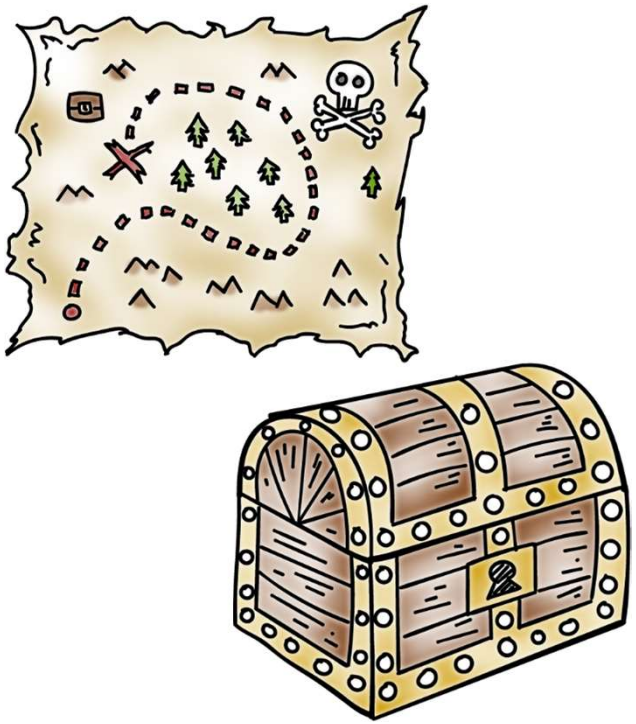
While your fingerprints may be the biometric standby these days, there are some issues with relying on them too heavily. For one, they're fairly easy to copy. Second, if someone is truly invested in breaking into your accounts, that may provide enticement to (*gulp*) remove a finger. Vein matching, on the other hand, can also use a finger or a palm, but provides a few additional benefits—most notably that the veins must be from a living person in order to work, and that they're very hard to fake.

The Sniff Test GM/Flickr. https://creativecommons.org/licenses/by/2.0/

When that grade school bully taunted "Smell ya later," he probably didn't realize that he was predicting another potential biometric method. That's right, your distinct body odor—and we're making no judgments here—can be used to identify you. Researchers at the Polytechnical University of Madrid have studied how scents differ among people and built an artificial nose, which they say can differentiate between two people by their smell, like a bloodhound. The U.S. Army is interested in similar technology, which it would like to use to help suss out potential threats. It's still early days, though: the artificial nose can filter out smells like hand cream or changes in odor caused by diet and disease, but the Madrid team's technology still has failure rate of around 10 percent.

Other Authentication Methods

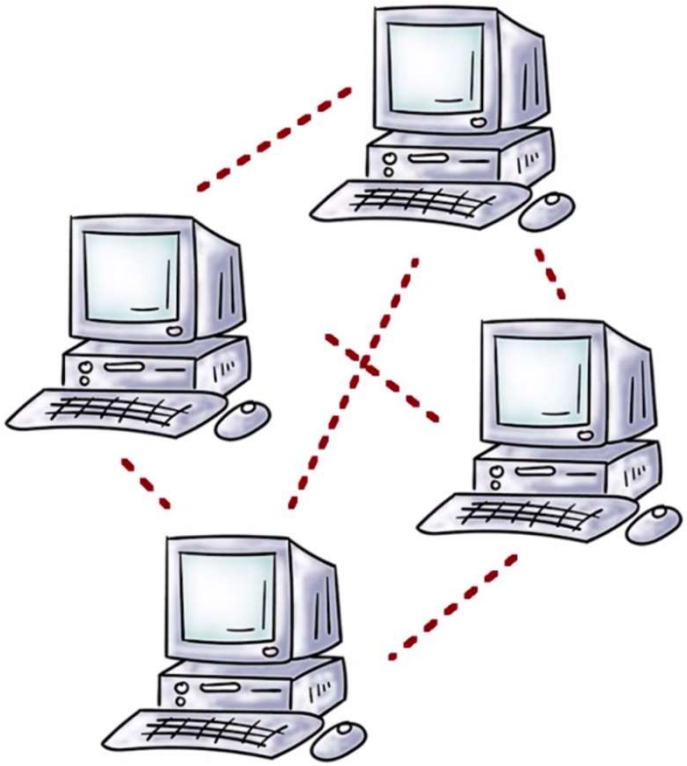
Multi-factor authentication



- Uses more than one method
- Type password but also send a code via SMS
  - It goes to your phone (something you have)
  - Gmail implements this
- ATM card and a PIN
- Other things like your location
- **Attacker must defeat both to compromise authentication**

Other Authentication Methods

Authentication over a network:



- Do we always have a trusted path to the OS we need to authenticate to?
  - **Remote services**
- Network authentication **introduces new problems**
- Need crypto to secure network communication
- **Other attacks** (man-in-the-middle)

Authentication
Lesson Summary

- Authenitcation is a **key requirement for securing access** to resources
- All methods present a **number of tradeoffs** that need to be balanced
- Understand how **various types of authentication is implemented**
- Security mindset requires that we do **careful threat modeling**