Week 2 - Wednesday

COMP 4290

Last time

- What did we talk about last time?
- Authentication
- Challenge-response
- Passwords
- Started biometrics

Questions?

Project 1

Adam Garantche Presents

Biometrics

Other biometrics

- Hand geometry readers measure the shape of your hand
- Keystroke dynamics are the patterns that you use when typing
  - Users are quite distinctive, but distractions and injuries can vary patterns a lot
- Combinations of different biometrics are sometimes used
- DNA sequencing is not (yet) fast enough to be used for authentication
- Researchers are finding new biometrics to use

Problems with biometrics

- People assume that they are more secure than they are
- Attacks:
  - Fingerprints can be lifted off a champagne glass
  - Voices can be recorded
  - Iris recognition can be faked with special contact lenses
- Both false positives and false negatives are possible
- Disabilities can prevent people from using some kinds of biometrics
- It's possible to tamper with transmission from the biometric reader
- Biometric characteristics can change
- Identical twins sometimes pose a problem

False positives and false negatives

	Is the Person Claimed	Is Not the Person Claimed
Test is Positive	***a***	***b***
Test is Negative	***c***	***d***

- **Sensitivity** is positive results among correct matches
  - $a / (a + c)$
- **Specificity** is negative results among people who are not sought
  - $d / (b + d)$
- **Accuracy** is how often the test is correct
  - $(a + d) / (a + c + b + d)$
- **Prevalence** is how common a condition is
  - $(a + c) / (a + c + b + d)$

Tokens

- Tokens are physical objects you possess
  - Keys
  - Badges
  - Cell phones
  - RFIDs
- **Passive tokens** take no action and do not change
  - Example: photo ID
- **Active tokens** change or interact with surroundings
  - Examples: RFID or magnetic card

Static and dynamic tokens

- The value of a **static token** does not change
  - Examples: Keys, passports, RFIDS
  - Static tokens are better for onsite authentication and may be easy to forge for remote authentication
- **Dynamic tokens** have values that change
  - Examples: RSA SecurID, Battle.net Authenticator
  - Every 60 seconds, it displays a different code

Multifactor authentication

- More than one form of authentication may provide increased security
  - You may need to sign on with your password and with a code generated by an RSA SecurID
  - They often need two forms of ID when you're getting a driver's license
- Two-factor authentication is now common for many platforms
  - Often they only ask for the second form of authentication if the computer has not logged on before
- Multifactor authentication is probably more secure, but it adds complexity and possibly annoyance

Federated identity management

- It's annoying to sign on to lots of different services with lots of different authentication mechanisms
- **Federated identity management** schemes connect a variety of different services with one authentication method
  - Example: free access to journals because you're logged onto Otterbein computers
- **Single sign-on** is similar, allowing you to log in once, with services sharing authentication information
  - Examples: logging onto Meetup.com with Facebook or Google credentials

Access Control

Access control

- **Subjects** are human users or programs that are executing on their behalf
- **Objects** are things that actions can be performed on
  - Files
  - Database fields
  - Directories
  - Hardware devices
- **Access modes** are the different ways that access can be done: read, write, modify, delete, etc.
- **Access control** is the process of managing the access modes that subjects can have on objects

Access control goals

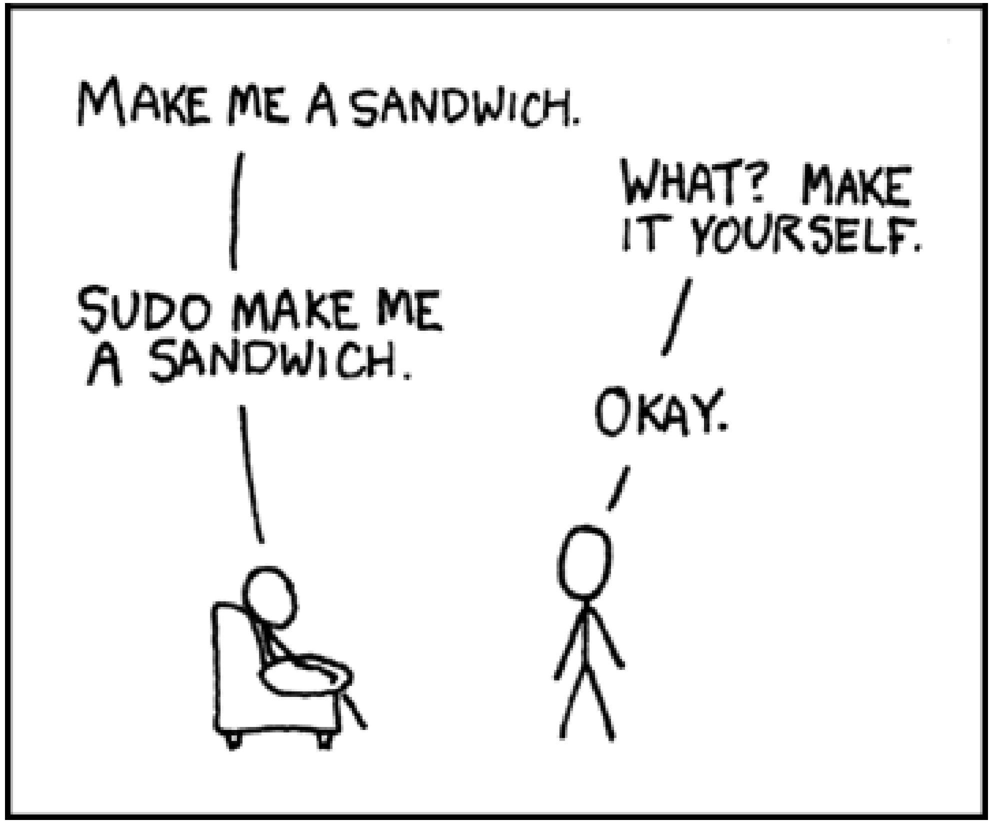
- Check every access
  - The user may no longer have rights to a resource
  - The user may have gained rights
- Enforce least privilege
  - **Least privilege** means you get the bare minimum to get your job done
- Verify acceptable usage
  - Access to an object is not enough: Some actions might be legal and others illegal

Access control issues

- Many issues come up with access control
- Do the correct people have the correct rights? Have statuses changed?
- **Granularity** is the how specifically you can control rights
  - Maybe you can only give complete rights to an object, not read-only rights
- An audit log tracks who performed what kinds of accesses
- Limited privilege tries to keep accesses from doing big damage
  - Example: **`sudo`** in Linux

sudo

- It is possible to temporarily use another user's permissions in Unix using the command **`sudo`**
- Users can be given special access to files or commands they normally could not access
- An administrator can run at a normal privilege level and only occasionally run commands using higher privileges
- This strategy prevents the whole system from being corrupted if the administrator gets a virus



Directory based approaches

- Create a directory that lists all the objects a given user can access and their associated rights:
  - Examples: read, write, execute, own
- The own right gives the user the ability to grant others rights to that object
- Problems:
  - Directories can become large
  - How is access revoked?
  - What if two files in different locations in the system have the same name?

Access control lists

- Listing all the objects a user can access can take up too much space
- An alternative is to list all the users that have rights for a specific object
- Most objects only have a few legal users
- Wild cards can make the situation easier
    - Read access can be granted to everyone

Access control matrices

- Both directories and access control lists are equivalent
- Different implementations are used for different kinds of efficiency
- We can also imagine a matrix that holds all subjects and all objects
- Although it is far too inefficient for most systems to be implemented this way, security researchers sometimes use this model for theoretical purposes
  - Can you determine if some sequence of operations could leak read access to your file?
  - Nope, it's impossible!

Access control matrix example

	Objects			
Subjects	file 1	file 2	process 1	process 2
process 1	*read, write, own*	*read*	*read, write, execute, own*	*write*
process 2	*append*	*read, own*	*read*	*read, write, execute, own*

Rights

- A few possible rights:
  - Read
  - Write
  - Execute
  - Own
  - Anything else that is useful?
- Some rights allow users to change the rights of others

Brightspace system

- What would the access control matrix look like for the Brightspace gradebook system?

Ticket out the Door

Upcoming

Next time…

- Finish access control
- Cryptography basics

Reminders

- Read Section 2.3
- Work on Project 1
- Work on Assignment 1