# Last time

- Protection in General-Purpose Operating Systems
  - History
  - Separation vs. Sharing
  - Segmentation and Paging
  - Access Control Matrix
  - Access Control Lists vs. Capabilities

# This time

- Role-based Access Control
- User Authentication
  - Authentication Factors
  - Passwords
  - Attacks on Passwords

# Role-based Access Control (RBAC)

- In a company, objects that a user can access often do not depend on the identity of the user, but on the user's job function (role) within the company
  - Salesperson can access customers' credit card numbers, marketing person only customer names
- In RBAC, administrator assigns users to roles and grants access rights to roles
- When a user takes over new role, need to update only her role assignment, not all her access rights
- Available in many commercial databases

# RBAC Extensions

- RBAC also supports more complex access control scenarios
- Hierarchical roles
  - "A manager is also an employee"
  - Reduces number of role/access rights assignments
- Users can have multiple roles and assume/give up roles as required by their current task
  - "Alice is a manager for project A and a tester for project B"
  - User's current session contains currently initiated role
- Separation of Duty
  - "A payment order needs to be signed by both a manager and an accounting person, where the two cannot be the same person"

# User Authentication

- Computer systems often have to identify and authenticate users
  - OS when a user logs in
  - Web server before handing out confidential information, like your grades
- Identification and authentication is easy among people that know each other
  - You identify your friends 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

# Authentication Factors

- ~~Three~~ Four classes of authentication factors
- Something the user knows
  - User name and password, PIN, answer to secret question
- Something the user has
  - ATM card, badge, browser cookie, physical key, uniform
- 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

# 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
  - Remember sophisticated Phishing attacks
- “Something you have” can become “something you know”
  - If token can be easily duplicated, such as magnetic strip on ATM card. That’s why ATM fraud is so wide spread
  - Some banks distribute small devices displaying numbers that change over time. Current number needs to be input for online banking. However, knowing number does not imply physical possession of device

# Passwords

- Probably oldest authentication mechanism used in computer systems
- User enters user ID and password, maybe multiple attempts in case of error
- Usability problems
  - Forgotten passwords might not be recoverable (though this has been changing recently, see later)
  - Entering passwords is inconvenient
  - If password is disclosed to unauthorized individual, the individual can immediately access protected resource
    - Unless we use multi-factor authentication
  - If password is shared among many people, password updates become difficult

# Password Guessing Attacks

- Brute-force: Try all possible passwords using exhaustive search
- It's possible to test 350,000 Microsoft Word passwords per second on a 3-GHz Pentium 4
- For passwords of length 8 consisting only of letters, there are about $2*10^{11}$ possibilities
- It takes only 600,000 seconds or 166 hours to test all of them
  - Expected wait till success is 83 hours
- Easy to buy more hardware if payoff is worth it
- Can make attack harder by including digits and special characters in password
- However,…

# 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 observation
- For example, Password Recovery Toolkit (PRTK) assumes that a password consists of a root and a pre- or postfix appendage
  - “password1”, “abc123”, “123abc”
- Root is taken from dictionaries (names, English words,…)
- Appendage is two-digit combination, date, single symbol,…
- PRTK could have cracked 55% of 34,000 leaked MySpace passwords in 8 hours
  - Even though passwords turned out to better than passwords from previous studies

# 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?

# Choosing Good Passwords

- Use letters, numbers and special characters
- Choose long passwords
  - At least eight characters
- Avoid guessable roots
- If supported, use pass phrase
  - Mix upper and lower case, introduce misspellings and special characters
  - Avoid common phrases (e.g., advertisement slogans)

# Password Hygiene

- Writing down passwords is more secure than storing many passwords on a networked computer or re-using same password across multiple sites
  - Unreasonable to expect users to remember long passwords, especially when changed often
  - Requires physical security for password sheet, don't use sticky notes
- Change passwords regularly
  - Especially if shorter than eight characters
  - Should users be forced to change their password?
  - Leads to password cycling and similar
    - "myFavoritePwd" -> "dummy" -> "myFavoritePwd"
    - goodPwd."1" -> goodPwd."2" -> goodPwd."3"

# Password Hygiene

- 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 their 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)
  - Don't do online banking on them
  - While traveling, forward your email to a free Webmail provider and use throwaway (maybe weak) password

# 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

# Defending against Attacks

- Store only a digital fingerprint of the password (using cryptographic hash, see later) in the password file
- When logging in, system computes fingerprint of entered password and compares it with user's stored fingerprint
- Still allows guessing attacks when password file leaks
- UNIX makes these attacks harder by storing user-specific salts in the password file
  - Salt is derived from time of day and process ID of /bin/passwd
  - Salt is included when computing password fingerprint
  - Two users who happen to have the same password will 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 Attacks

- Store an encrypted version of the password in the password file
- Need to keep encryption key away from attackers
- As opposed to fingerprints, this approach allows system to (easily) re-compute password if necessary
  - E.g., have system email password to predefined email address when user forgets password
  - Has become the norm for many websites
  - In fact, some people use this reminder mechanism whenever they want to log in to a website
    - This way, they no longer have to remember passwords

# Interception Attacks

- Attacker intercepts password while it is being transmitted to website
- One-time passwords make intercepted password useless for later logins
  - In a challenge-response protocol, the server sends a random challenge to the client
  - Client uses challenge and password as an input to a function and computes 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
- Cryptographic protocols (e.g., SRP) make intercepted information useless to an attacker

# Recap

- Role-Based Access Control
- User Authentication
  - Authentication Factors
  - Passwords
  - Attacks on Passwords

# Next time

- User Authentication
  - Beyond passwords
  - Biometrics
- Security Policies and Models
  - Trusted Operating Systems and Software
  - Military and Commercial Security Policies
  - Bell La-Padula and Biba Security Models
  - Information Flow Control