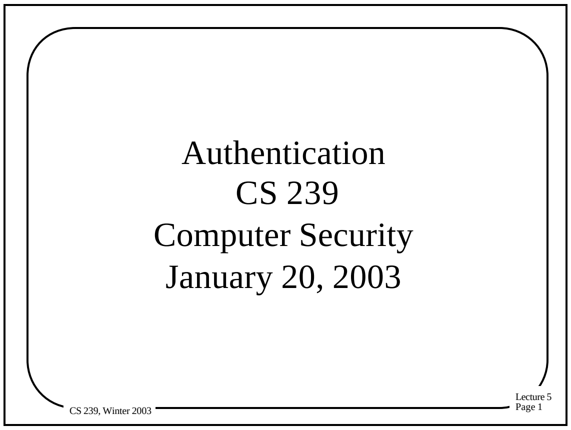
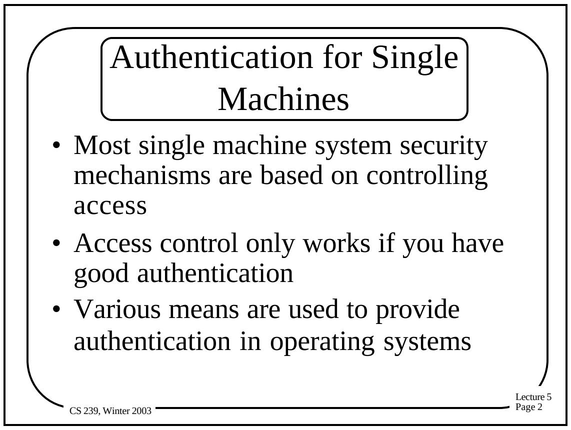
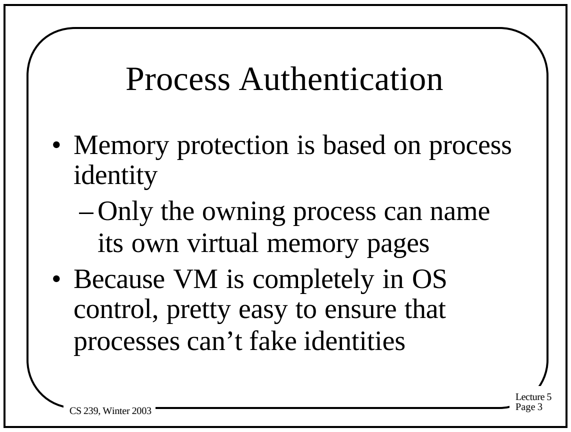


Authentication
CS 239
Computer Security
January 20, 2003



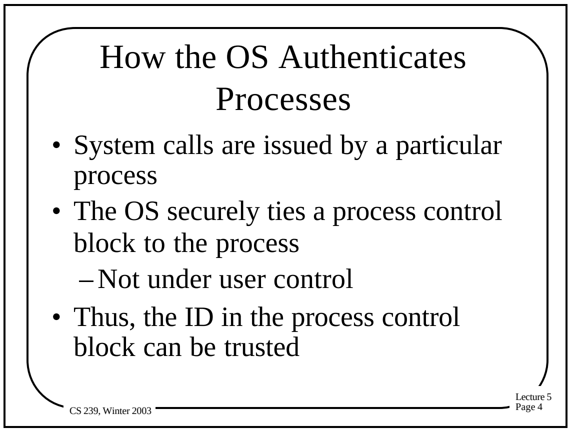
Authentication for Single Machines

- Most single machine system security mechanisms are based on controlling access
- Access control only works if you have good authentication
- Various means are used to provide authentication in operating systems



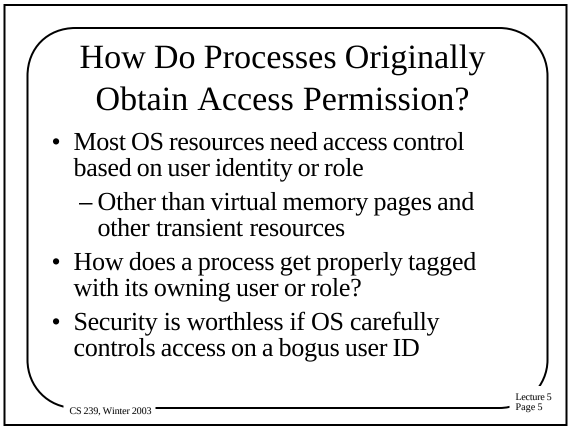
Process Authentication

- Memory protection is based on process identity
  - Only the owning process can name its own virtual memory pages
- Because VM is completely in OS control, pretty easy to ensure that processes can’t fake identities



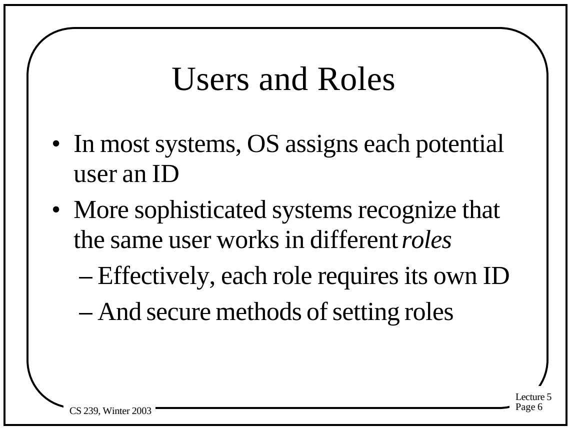
How the OS Authenticates Processes

- System calls are issued by a particular process
- The OS securely ties a process control block to the process
  - Not under user control
- Thus, the ID in the process control block can be trusted



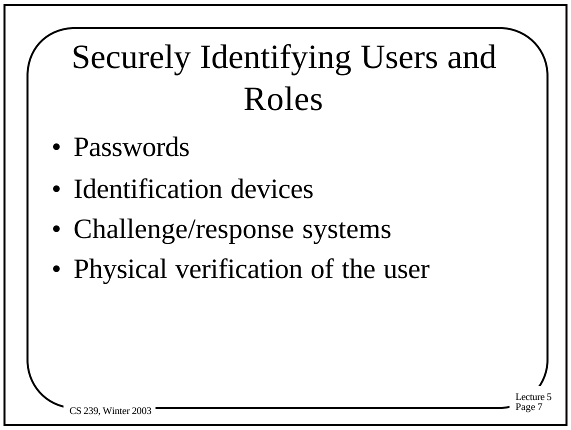
How Do Processes Originally Obtain Access Permission?

- Most OS resources need access control based on user identity or role
  - Other than virtual memory pages and other transient resources
- How does a process get properly tagged with its owning user or role?
- Security is worthless if OS carefully controls access on a bogus user ID



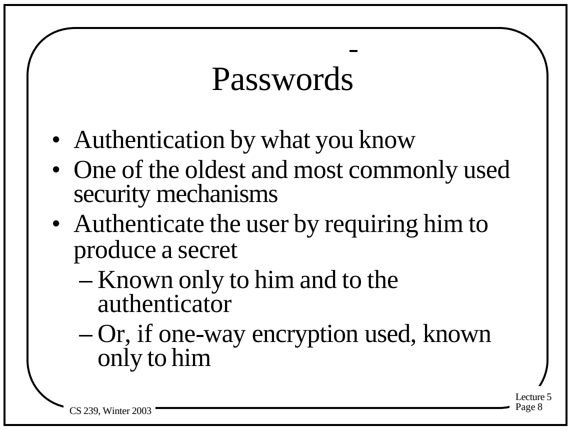
Users and Roles

- In most systems, OS assigns each potential user an ID
- More sophisticated systems recognize that the same user works in different *roles*
  - Effectively, each role requires its own ID
  - And secure methods of setting roles



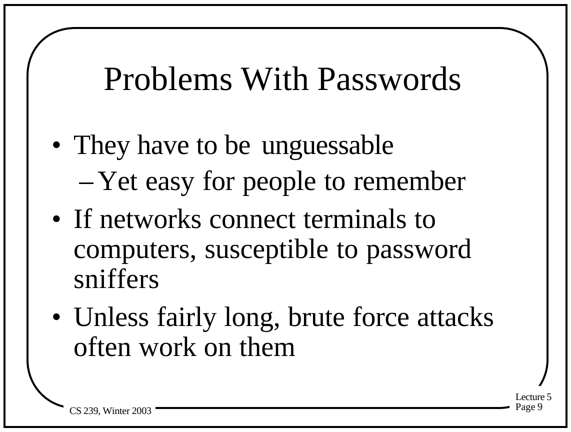
Securely Identifying Users and Roles

- Passwords
- Identification devices
- Challenge/response systems
- Physical verification of the user



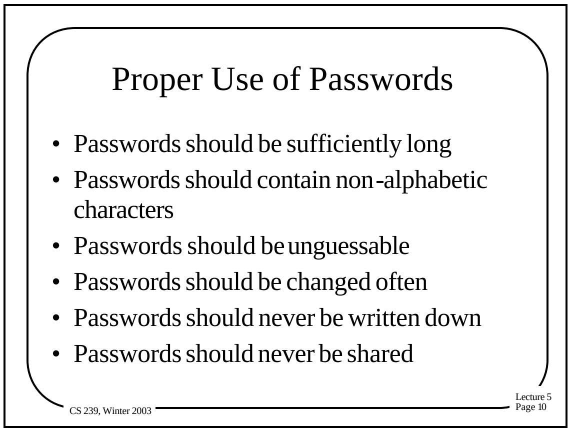
Passwords

- Authentication by what you know
- One of the oldest and most commonly used security mechanisms
- Authenticate the user by requiring him to produce a secret
  - Known only to him and to the authenticator
  - Or, if one-way encryption used, known only to him



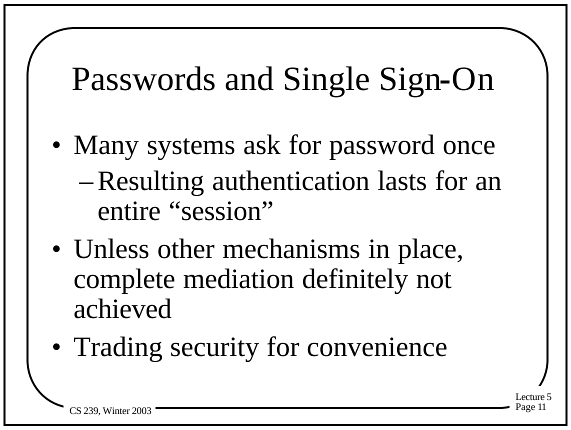
Problems With Passwords

- They have to be unguessable
  - Yet easy for people to remember
- If networks connect terminals to computers, susceptible to password sniffers
- Unless fairly long, brute force attacks often work on them



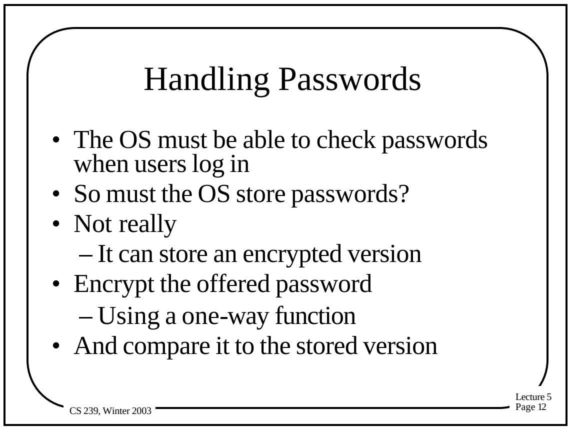
Proper Use of Passwords

- Passwords should be sufficiently long
- Passwords should contain non-alphabetic characters
- Passwords should be unguessable
- Passwords should be changed often
- Passwords should never be written down
- Passwords should never be shared



Passwords and Single Sign-On

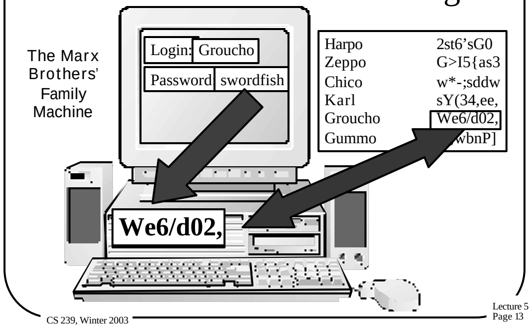
- Many systems ask for password once
  - Resulting authentication lasts for an entire "session"
- Unless other mechanisms in place, complete mediation definitely not achieved
- Trading security for convenience



Handling Passwords

- The OS must be able to check passwords when users log in
- So must the OS store passwords?
- Not really
  - It can store an encrypted version
- Encrypt the offered password
  - Using a one-way function
- And compare it to the stored version

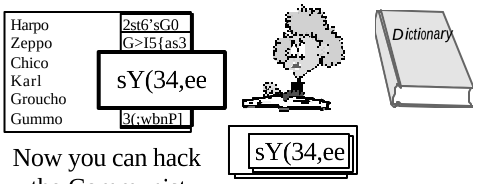
Standard Password Handling



Is Encrypting the Password File Enough?

- What if an attacker gets a copy of your password file?
- No problem, the passwords are encrypted
  - Right?
- Yes, but . . .

Dictionary Attacks on an Encrypted Password File



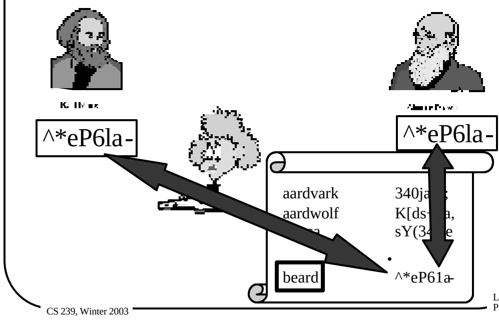
Now you can hack the Communist Manifesto!

Rats!!!!

A Serious Issue

- All Linux machines use the same one-way function to encrypt passwords
- If someone runs the entire dictionary through that function,
  - Will they have a complete list of all encrypted dictionary passwords?

Illustrating the Problem



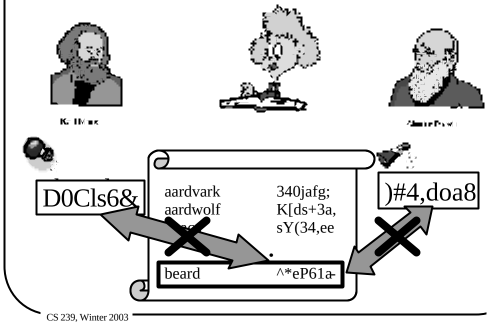
The Real Problem

- Not that Darwin and Marx chose the same password
- But that anyone who chose that password got the same encrypted result
- So the attacker need only encrypt every possible password once
- And then she has a complete dictionary usable against anyone

Salted Passwords

- Combine the plaintext password with a random number
  - Then run it through the one-way function
- The random number need not be secret
- It just has to be different for different users

Did It Fix Our Problem?



Protecting the Password File

- So it’s OK to leave the encrypted version of the password file around?
- No, it isn’t
- Why make it easy for attackers?
- Dictionary attacks against single accounts can still work
- Generally, don’t give access to the encrypted file, either

Identification Devices

- Authentication by what you have
- A smart card or other hardware device that is readable by the computer
- Authenticate by providing the device to the computer

Problems With Identification Devices

- If lost or stolen, you can’t authenticate yourself
  - And someone else can
  - Often combined with passwords to avoid this problem
- Unless cleverly done, susceptible to sniffing attacks
- Requires special hardware

Challenge/Response Authentication

- Authentication by what questions you can answer correctly
- The system asks the user to provide some information
- If it’s provided correctly, the user is authenticated

Differences From Passwords

- Challenge/response systems ask for different information every time
- Or at least the questions come from a large set
- Best security achieved by requiring what amounts to encryption of the challenge
  - But that requires special hardware
  - Essentially, a smart card

Problems With Authentication Through Challenge/Response

- Either the question is too hard to answer without special hardware
- Or the question is too easy for intruders to spoof the answer
- Still, commonly used in real-world situations
  - E.g., authenticating you by asking your mother’s maiden name

Authentication Through Physical Verification

- Authentication based on who you are
- Things like fingerprints, voice patterns, retinal patterns, etc.
- To authenticate to the system, let it measure the appropriate physical characteristics

Problems With Physical Verification

- Requires very special hardware
  - Possibly excepting systems that examine typing patterns
- May not be as foolproof as you think
- Many characteristics vary too much for practical use
- Generally not helpful for authenticating programs or roles
- What happens when it’s cracked?
  - You only have two retinas, after all

Authenticating Across the Network

- What new challenges does this add?
- You don’t know what’s at the other end of the wire
- So, when does that cause a problem?
- And how can you solve it?