

# Security Principles, Policies, and Tools

CS 236

On-Line MS Program

Networks and Systems Security

Peter Reiher

# Outline

- Security design principles
- Security policies
  - Basic concepts
  - Security policies for real systems
- Classes of security tools
  - Access control

# Design Principles for Secure Systems

- Economy
- Complete mediation
- Open design
- Separation of privileges
- Least privilege
- Least common mechanism
- Acceptability
- Fail-safe defaults

# Economy in Security Design

- Economical to develop
  - And to use
  - And to verify
- Should add little or no overhead
- Should do only what needs to be done
- Generally, try to keep it simple and small

# Complete Mediation

- Apply security on every access to a protected object
  - E.g., each read of a file, not just the open
- Also involves checking access on everything that could be attacked

# Open Design

- Don't rely on "security through obscurity"
- Assume all potential attackers know everything about the design
  - And completely understand it
- This doesn't mean publish everything important about your security system
  - Though sometimes that's a good idea
- Obscurity can provide *some* security, but it's brittle
  - When the fog is cleared, the security disappears
  - And modern attackers have good fog blowers

# Separation of Privileges

- Provide mechanisms that separate the privileges used for one purpose from those used for another
- To allow flexibility in security systems
- E.g., separate access control on each file

# Least Privilege

- Give bare minimum access rights required to complete a task
- Require another request to perform another type of access
- E.g., don't give write permission to a file if the program only asked for read



# Least Common Mechanism

- Avoid sharing parts of the security mechanism
  - among different users
  - among different parts of the system
- Coupling leads to possible security breaches

# Acceptability

- Mechanism must be simple to use
- Simple enough that people will use it without thinking about it
- Must rarely or never prevent permissible accesses

# Fail-Safe Designs

- Default to lack of access
- So if something goes wrong or is forgotten or isn't done, no security lost
- If important mistakes are made, you'll find out about them
  - Without loss of security
  - But if it happens too often . . .

# Thinking About Security

When considering the security of any system, ask these questions:

1. What assets are you trying to protect?
2. What are the risks to those assets?
3. How well does the security solution mitigate those risks?
4. What other security problems does the security solution cause?
5. What tradeoffs does the security solution require?

(This set of questions was developed by Bruce Schneier, for his book *Beyond Fear*)