THE UNIVERSITY OF BRITISH COLUMBIA

Principles of Designing Secure Systems

EECE 412

learning objectives

- recognize the principles
- explain which should (have been) be applied

What Do you Already Know?

- What principles of designing secure systems do you already know?
- What anti-principles do you know?
  - "security through obscurity"
  - m&m security

source: candyrific.com

Principles

1. Least Privilege
2. Fail-Safe Defaults
3. Economy of Mechanism
4. Complete Mediation
5. Open Design
6. Separation of Privilege
7. Least Common Mechanism
8. Psychological Acceptability
9. Defense in depth
10. Question assumptions

Overarching Goals

- Simplicity
  - Less to go wrong
  - Fewer possible inconsistencies
  - Easy to understand
- Restriction
  - Minimize access
    - “need to know” policy
  - Inhibit communication to minimize abuse of the channels

Principle 1: Least Privilege

Every program and every user of the system should operate using the least set of privileges necessary to complete the job

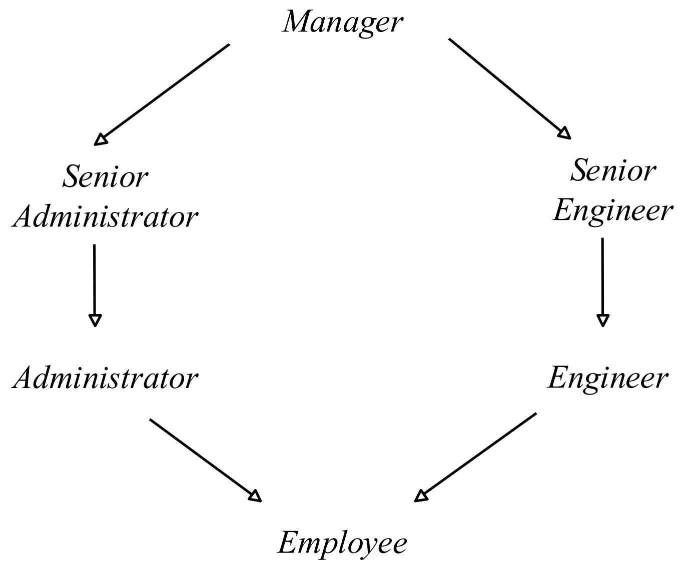
- Rights added as needed, discarded after use

- Limits the possible damage
- Unintentional, unwanted, or improper uses of privilege are less likely to occur
- Guides design of protection domains

Example:
Privileges in Operating Systems

- Until Windows NT, all privileges for everybody
- Separate admin (a.k.a., root) account on Windows and Unix
  - Ways to switch between accounts
- IIS account in Windows Server 2003

Example: role-based access control



Differentiation between assigned and activated roles

Example: IIS in Windows Server 2003

- before -- all privileges
- in Windows Server 2003 and later -- low-priveleged account

Principle 2: Fail-Safe Defaults

Base access decisions on permission rather than exclusion.

suggested by E. Glaser in 1965

- Default action is to deny access
- If action fails, system as secure as when action began

Example: IIS in Windows Server 2003

crashes if attacked using buffer overflow

Principle: Economy of Mechanism

Keep the design as simple and small as possible.

- KISS Principle
- Rationale?
  - Essential for analysis
  - Simpler means less can go wrong
    - And when errors occur, they are easier to understand and fix

Example:
Trusted Computing Base
(TCB)

- temper-proof
- non-bypassable
- small enough to analyze it

Principle 4:
Complete Mediation

Every access to every object must be checked for authority.

If permissions change after, may get unauthorized access

Example: .rhosts mechanism abused by Internet Worm

Access to one account opened unchecked access to other accounts on different hosts

Example:
Multiple reads after one check

- Process rights checked at file opening
- No checks are done at each read/write operation
- Time-of-check to time-of-use

Kerckhoff's Principle

"The security of a cryptosystem must not depend on keeping secret the crypto-algorithm. The security depends only on keeping secret the key"

Auguste Kerckhoff von Nieuwenhof

Dutch linguist

1883

Principle 5:
Open Design

Security should not depend on secrecy of design or implementation

P. Baran, 1965

- no "security through obscurity"
- does not apply to information such as passwords or cryptographic keys

Example:
Content Scrambling System

DVD content

- SecretEcrypt(K_D,K_{p1})
- …
- SecretEcrypt(K_D,K_{pn})
- Hash(K_D)
- SecretEcrypt(K_T,K_D)
- SecretEcrypt(Movie,K_T)

1999

- Norwegian group derived SecretKey by using K_{Pi}
- Plaintiff’s lawyers included CSS source code in the filed declaration
- The declaration got out on the internet

Principle 6: Separation of Privilege

Require multiple conditions to grant privilege

R. Needham, 1973

Separation of duty

example: SoD constraints in RBAC

- static SoD
  - if a user is assigned role “system administrator” then the user cannot be assigned role “auditor”
- dynamic SoD
  - a user cannot activate two conflicting roles, only one at a time

Principle 7: Least Common Mechanism

Mechanisms should not be shared

- Information can flow along shared channels in uncontrollable way
- Covert channels
- solutions using isolation
  - Virtual machines
  - Sandboxes

example: network security

- switches vs. repeaters
- security enclaves

Principle 8: Psychological Acceptability

Security mechanisms should not add to difficulty of accessing resource

- Hide complexity introduced by security mechanisms
- Ease of installation, configuration, use
- Human factors critical here

example: Switching between user accounts

- Windows NT -- pain in a neck
- Windows 2000/XP -- “Run as …”
- Unix -- “su” or “sudo”

Principle 9: Defense in Depth

Layer your defenses

example: Windows Server 2003

Potential problem	Mechanism	Practice
Buffer overflow	defensive programming	check preconditions
Even if it were vulnerable	IIS 6.0 is **not** up by default	no extra functionality
Even if IIS were running	default URL length 16 KB	conservative limits
Even if the buffer were large	the process crashes	fail-safe
Even if the vulnerability were exploited	Low privileged account	least privileged

Principle 10: Question Assumptions

Frequently re-examine all the assumptions about the threat agents, assets, and especially the environment of the system

Example: Assumtpions, Assumptions, …

- ident
- finger protocol

Principles

1. Least Privilege
2. Fail-Safe Defaults
3. Economy of Mechanism
4. Complete Mediation
5. Open Design
6. Separation of Privilege
7. Least Common Mechanism
8. Psychological Acceptability
9. Defense in depth
10. Question assumptions