CS458: Introduction to Information Security

Notes 2: Security Policy

Yousef M. Elmehdwi

Department of Computer Science

Illinois Institute of Technology

yelmehdwi@iit.edu

August 30, 2018

Slides: Modified from: Computer Security: Principles and Practice, 4th Edition. By: William Stallings and Lawrie Brown, Richard A. Kemmerer, Kevin Jin

Outline

- Human Factors in Security Policy

Computer Security Strategy

- Comprehensive security strategy involves three aspects
  - **Specification/Policy**: What is the security scheme supposed to do?
  - **Implementation/mechanisms**: How does it do it?
  - **Correctness/assurance**: Does it really work?

Policy and Mechanism

- Policy
  - A security policy is a statement of what is, and what is not, allowed
  - i.e., what security schemes are supposed to do
  - It defines the concept of "security" for a computer system
  - Divides the world into secure and non-secure states
  - A secure system starts in a secure state. All transitions keep it in a secure state
- Mechanism
  - A security mechanism is a method, tool, or procedure to enforce a security policy
    - Different mechanisms can be used to enforce the same policy
- For example:
  - Policy: students should not copy other students' assignments
  - Mechanism: `chmod 700 *`

Is this situation secure?

- Web server accepts all connections
  - No authentication required
  - Self-registration
  - Connected to the Internet
- Is this situation secure?
- Depends on policy
  - May be availability is really important, and the traffic is properly filtered by firewall.

Policy Example

- University computer lab has a policy that prohibits any student from copying another student's homework files
  - The computers provide mechanisms (file access controls) for preventing others from reading a user's files
- CS class has students do homework on computer
- Anne fails to use these mechanisms to protect her homework file (read-protect)
- Bill copies it
- Who cheated? Anne, Bill, both, neither?

Policy Example: Answer Part 1

- Bill cheated
  - Policy forbids copying homework assignment
  - Bill copied
  - System entered unauthorized state (Bill having a copy of Anne's assignment)

Policy Example: Answer Part 2

- Anne didn't protect her homework
  - Not required by security policy
- She didn't breach security
- If policy said students had to read-protect homework files, then Anne would have breached security
  - She didn't

Policy Example

- What if Anne posted his homework on his dorm room door?
- What if Anne did read protect her files, but Bill found a hack on the mechanism?

Goals of Security

- Security mechanisms implement functions that help prevent, detect, and respond to recovery from security attacks
- Prevention
  - Prevent attackers from violating security policy
  - Example: use of passwords
- Detection
  - Detect attackers' violation of security policy
  - Example: use of logging of sensitive operations
- Response:
  - If security mechanisms detect an ongoing attack, such as a denial of service attack, the system may be able to respond in such a way as to halt the attack and prevent further damage.
- Recovery
  - Stop attack, assess and repair damage
  - Example, use of backup systems, so that if data integrity is compromised, a prior, correct copy of the data can be reloaded.
  - Continue to function correctly even if attack succeeds

Trust and Assumptions

- A policy describes the security of a system in a certain environment and under certain assumptions
  - A policy that states that students should not copy other students' files, which is enforced by using file system access control mechanisms, is valid under the assumption that students don't share passwords and that they set file access privileges correctly

Trust and Assumptions

- Assumptions about policies
  - A policy unambiguously partitions a system?s states into secure and nonsecure
  - A policy correctly captures the security requirements of the real world
- Assumptions about mechanisms
  - The security mechanisms enforce the policy and prevent the system from entering a nonsecure state
  - The mechanisms can be trusted
    - Are implemented correctly
    - Are installed and administered correctly

Another Policy Example

- Bank officers may move money between accounts.
  - Any flawed assumptions here?

Assurance

- Assurance is a measure of how well the system meets its requirements
  - In other words, how much one can trust the system to do what it is supposed to do
- Examples: Testing, code audits, formal proofs
- Assurance is derived by analyzing the specification, design, and implementation of a system

Assurance

- Assurance: Procedures ensuring that policy is enforced
- i.e., Evidence of how much to trust a system
- Examples: Testing, code audits, formal proofs
- Assurance is what justifies our trust in a system
- Evidence can include
  - System specifications
  - Design
  - Implementation

Trust

- Trust: Belief that system or component will perform as expected or required
- Trusted: assumed to perform as expected or required
- Trustworthy: will perform as expected or required
  - Some authors use trustworthy to mean that ?there is sufficient credible evidence? that system or component will perform as expected or required

Aspirin Assurance Example

- Why do you trust Aspirin from a major manufacturer?
  - FDA certifies the aspirin recipe
  - Factory follows manufacturing standards
  - Safety seals on bottles
- Analogy to software assurance
  - Software assurance ensures integrity, security, and reliability in software

Operational Issues

- Security does not end when the system is completed. Its operation affects security.
- A “secure” system can be breached by improper operation (for example, when accounts with no passwords are created).
- The question is how to assess the effect of operational issues on security.

Operational Issues

- Cost-Benefit Analysis
  - Weighs the cost of protecting data and resources with the costs associated with losing the data
  - Is it cheaper to prevent or recover?
- Risk Analysis
  - What happens if the data and resources are compromised?
  - Should we protect something?
  - How much should we protect this thing?
- Laws and Customs
  - Are desired security measures illegal?
  - Will people do them?

Humans in the Loops

- Security would be so much easier if humans were not involved
- Generally a human is the cause of security flaw
  - improper use
  - Improper configuration
  - Improper development
  - Insufficient testing

User Education Can Help

- Actually train users to do the right thing
- Deter bad behavior by education about the penalties
- Mitigate organizational liability by showing due care
- Complying with regulations and contractual obligations

Types of Education

- Awareness: What is going on?
  - Security awareness is the knowledge and attitude that members of an organization possess regarding the protection of the physical and especially, information assets of that organization
- Literacy: How is it going on?
- Education: Why is it going on?

- Question: If you are conducting an organization's security awareness training, what topics to cover?

Types of Education

- Sensitive material and physical assets
  - e.g., trade secrets, privacy concerns and government classified info
- Responsibilities in handling sensitive information
  - e.g., review of employee nondisclosure agreements
- Proper handling of sensitive material in physical form
  - e.g., marking, transmission, storage and destruction
- Methods for protecting sensitive info on computer systems
  - e.g., password policy and use of two-factor authentication
- Workplace security
  - e.g., building access, wearing of security badges, reporting of incidents, forbidden articles, etc.
- Consequences of failure
  - e.g., potential loss of employment, damage to individuals whose private records are divulged, and possible civil and criminal penalties

Design Principles for Protection Mechanisms

- **Least Privilege**
  - Every program and every user of the system should operate using the least set of privileges necessary to complete the job.
  - Should only have the rights necessary to complete your task.
  - Default should be lack of access
  - If access needed temporarily, then it should be rescinded right after use
  - The military security rule of "need-to-know" is an example of this principle.

Design Principles for Protection Mechanisms

- **Complete Mediation**
  - Every access to every object must be checked
  - Must be efficient

Design Principles for Protection Mechanisms

- **Open Design**
  - Don't depend on secrecy of the design
  - "Security through obscurity" is a bad idea
  - Should be open for scrutiny by the community
  - Better to have a friend/colleague find an error than a foe

Design Principles for Protection Mechanisms

- **Separation of Privilege**
  - Where feasible, a protection mechanism that requires two keys to unlock it is more robust and flexible than one that allows access to the presenter of only a single key
  - i.e., Access to objects should depend on more than one condition being satisfied
    - Separation of duty
    - Two person rule

Design Principles for Protection Mechanisms

- **Fail-Safe Defaults**
  - The default is lack of access
  - Need to argue why a user *should have* access. Do not argue why a user *should not have* access
  - If action fails, system as secure as when action began

Design Principles for Protection Mechanisms

- **Least Common Mechanism**
  - Minimize the amount of mechanism common to more than one user and depended on by all users
  - Every shared mechanism is a potential information path

Design Principles for Protection Mechanisms

- **Psychological Acceptability**
  - User interface must be easy to use, so that users routinely and automatically apply the mechanisms correctly. Otherwise, they will be bypassed
  - Security mechanisms should not add to difficulty of accessing resource

Design Principles for Protection Mechanisms

- **Diebold Voting Machines**
  - Inspection of the code by John Hopkins team found [1] [2]
    - Passwords embedded in the source code
    - Unauthorized privilege escalation and other vulnerabilities
    - Incorrect use of cryptography
    - Undetected, unlimited votes by voters
    - Insider threats: company workers or election officials can alter voters' ballot choices without their knowledge

[1] Analysis of an Electronic Voting System

[2] E-lective Alarm