Data Security

D. Denning / P. Denning
Purdue University

Computing Surverys, Vol II, No. 3 - September 1979

Jeffrey L. Welch
COP6614 – Operating Systems Techniques
UCF Fall 2002 – Computer Science Dept.
Prof. E. Montagne, Instructor

Jeffrey.L.Welch@saic.com

Overview

- Motivation
- Types of Controls
  - Definitions / Examples
- Limitations of the Controls
- Summary
- Concluding Remarks

Motivation

- Internal vs. External Security
- The unseen threats to sensitive data
  - Examples / situations
    - Money laundering from a bank
    - Political bribery / defamation
    - Leak of highly sensitive gov't data.
- Are theoretical safeguards practical?
    - They can be, but often must be tailored.

Motivation (Con't)

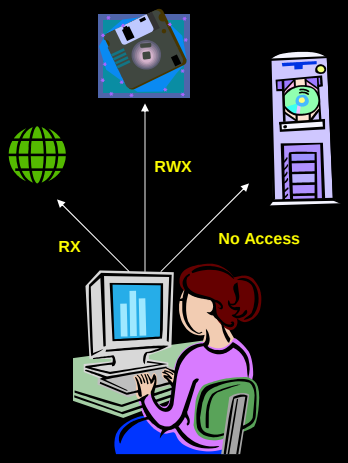
- Goal:
  - Useful / Proven internal security
  - Lessen concern for adequate hardware or software.
- Assume external security ignored
  - Protecting terminals, cypher locks on doors, effective password management, etc.

Types of Control

- 4 main types presented:
  - Access Control
    - R/W/X privilege control
  - Flow Control
    - X → Y data paths
  - Inference Control
    - Deducing protected information from summaries
  - Cryptographic Control
    - Key-based encryption

Access Control

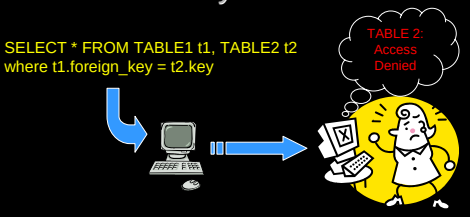
- Definition
  - Allow ownership to groups / users
  - Privileges like R W X
  - Assume:
    - Users always authenticated
    - No 3rd party screening / auditing
    - Privilege specification highly guarded from a user (perhaps known only to O/S or predefined power-users)
- 2 system examples
  - Data dependent
  - Object dependent



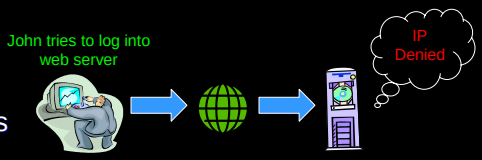
Access Control (2)

Data-dependent *Transaction-based Systems*

- Restrictions applied before execution
  - i.e. a DBMS w/limited tables to certain groups.



  - i.e. Web-based transactions
    - Have authenticated IPs

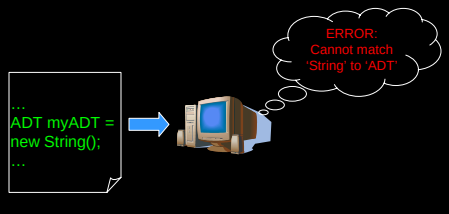


•Control can depend on current data and historically-derived.

Access Control (3)

Object-dependent *General Purpose Systems*

- Objects themselves are regulated.
  - Doesn't matter the data content
- i.e. Compilers
  - Provide type checking operations.

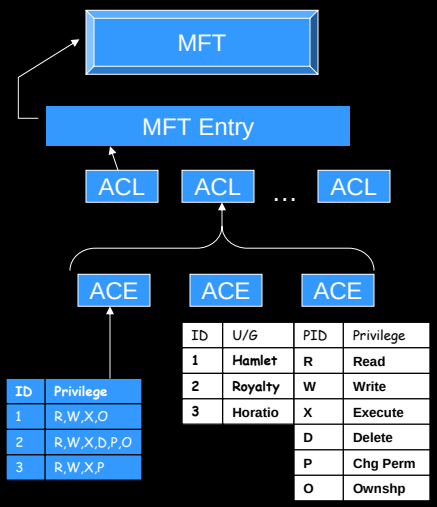


Code

Access Control (4)

***Object-dependent* Design Example 1**

- NT File System (NTFS)
  - FAT System lacking in security.
  - Security oriented to user rights
    - Based on a user's account
- Master File Table's Descriptor (SD)
  - Has "Access Control Lists" (ACL) with entries (ACE)
  - Identify user / groups permissions (R,W,X,D,P,O)



ID	Privilege
1	R,W,X,O
2	R,W,X,D,P,O
3	R,W,X,P

ID	U/G
1	Hamlet
2	Royalty
3	Horatio

PID	Privilege
R	Read
W	Write
X	Execute
D	Delete
P	Chg Perm
O	Ownshp

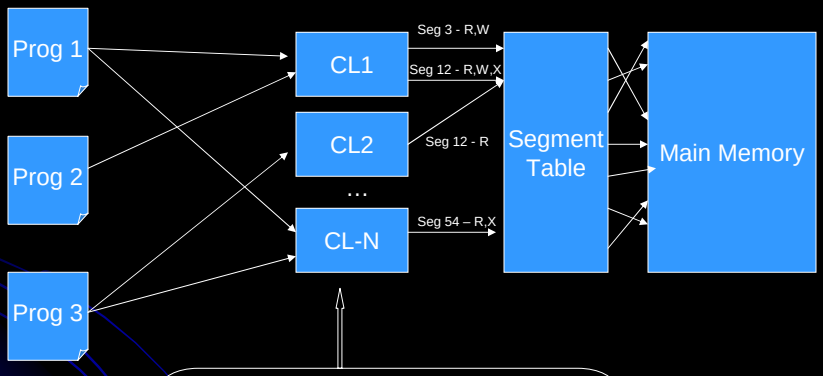
Access Control (5)

***Object-dependent* Design Example 2**

- Capability addressing in memory
  - Segments refer to portions of MM
    - Segment Table → MM
  - Each segment in memory has a "capability"
    - Capability Lists → Segment Table
  - Programs refer to their own capability list
    - Program → Capability Lists
    - Multiple programs can relate to an individual capability list.

Access Control (6)

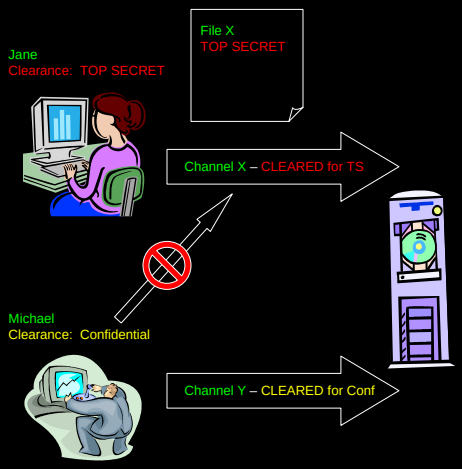
***Object-dependent* Design Example (con't)**



Capabilities define what segments can be accessed, and HOW they can be accessed

Flow Control

- "Flow" Definition
  - Object X → Object Y
    - Read X, Write Y
  - Implement a Policy
    - What data allowed on a channel?
    - What access does a user have on that channel?
    - What fidelity is the policy's model?
      - Simple
        - Confidential
        - Non-Confidential
      - Complex with many variations / classes

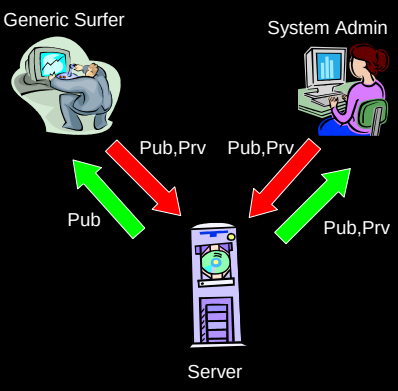


Michael cannot read File X:
No access to Channel X and only cleared for Confidential

Flow Control (2)

Example A

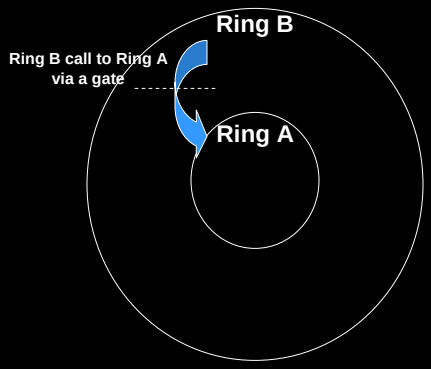
- Browse / Maintain a website
- Read
  - Any web-surfer or SA can read public data
  - Only SA can read private data.
- Write (e.g. feedback)
  - Web-surfer can leave private feedback.
  - Only SA can read collected feedback.



Flow Control (3)

Example B

- Protection Ring
  - Access Brackets define the Flow Policy b/w rings.
- Capabilities increase with ring level
  - i.e. Ring B has lower capability than Ring A, etc.



Flow Control (4)

Implementation

- Simple Flow
  - Assign a security class attached to pgm.
    - Read: Level <= Target
    - Write: Level >= Target
  - Overclassification
    - Write into increasingly higher levels that are unnecessary
    - Reduce by implementing a “ceiling”
      - Class security moves up as needed ONLY, but never down.
      - Class level stays constant longer.

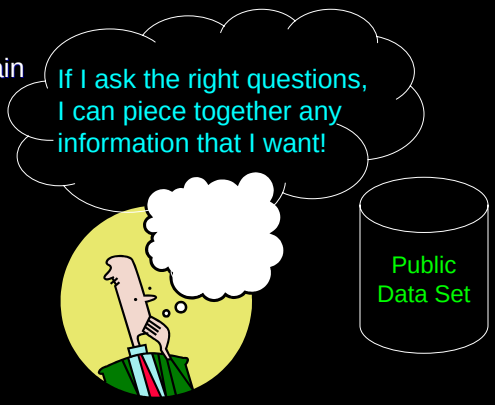
Flow Control (5)

Implementation (con’t)

- Implicit Data Flow
  - Can happen in code
    - If (x == 0) then y := 0 else y := 1
      - Say x value initially 0 or 1. Then x will always be implicitly copied to y (or partially if x not known)
  - Declare security classes for variables
    - Compiler “type-checks” according to stated flow policy
    - This is compile-time support
      - Run-time support more difficult and may require hardware assistance.

Inference Control

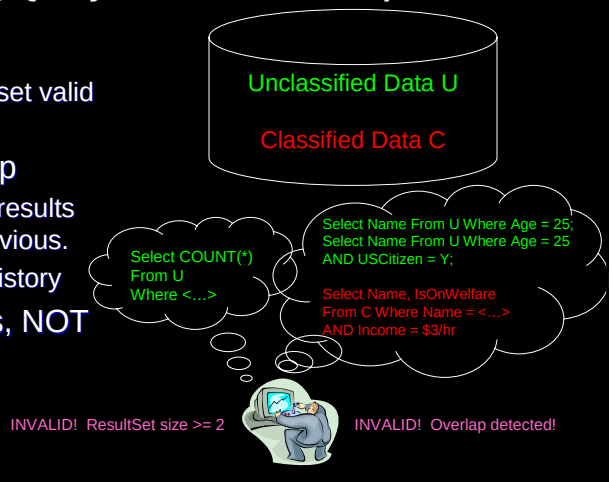
- Definition
  - Using deduction from a summary data set to obtain information that is otherwise controlled by a security policy
- Goals
  - Control *how much / what type* of data returned in response to a query
  - *Introduce error* for a response to a query



Inference Control (2)

Controlling Query Set Sizes / Overlaps

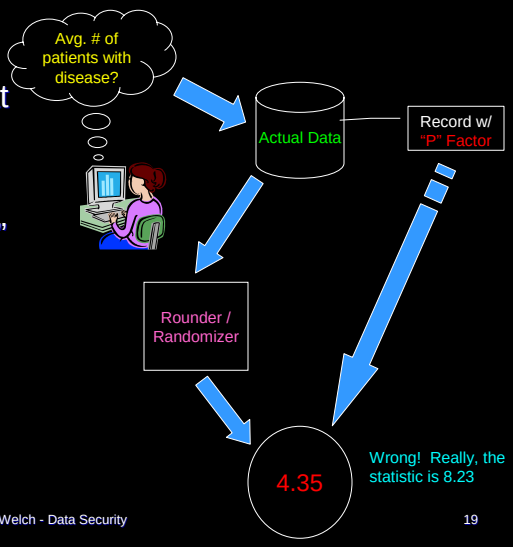
- Min / Max ranges
  - Only return result set valid within the range.
- Control Set overlap
  - Void a query with results "too similar" to previous.
  - Overhead query history
- Query to partitions, NOT individual records



Inference Control (3)

Results induced with error

- Rounding
  - Skew the results that are returned.
- Induce error
  - "Perturbation Factor"
  - Associate w/records
  - Apply before any operation on data



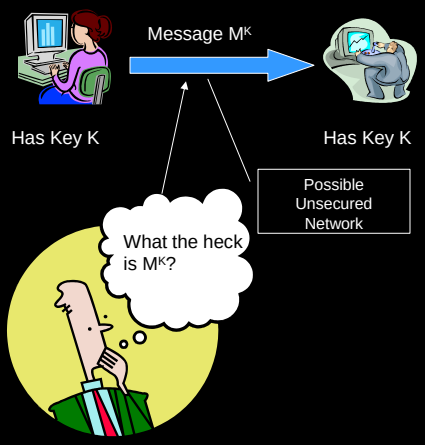
Inference Control (3)

Example: MLS DBMS

- Cerist – Software laboratory in Algeria (1997).
- MLS = Multi Level System
- DB contains data of differing classifications
  - Unclassified
  - Classified
  - Secret
  - Top Secret
- Approaches to prevent inference attacks:
  - Assign levels to each DB element
  - Knowledge Discovery
  - Data Mining Applications
  - Intelligent Agents that can use historical information to "predict" an attack.

Cryptographic Control

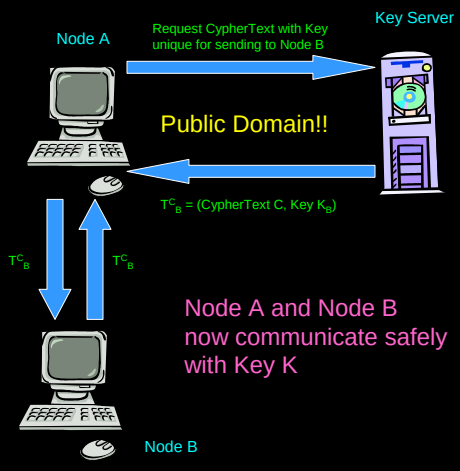
- Keep safe static or in-transit data.
  - Work with keys b/w sender and receiver
- Key management is ESSENTIAL!
  - Key Generation Server
  - DES Standard
  - PKI Standard



Cryptographic Control (2)

Key Generation Server

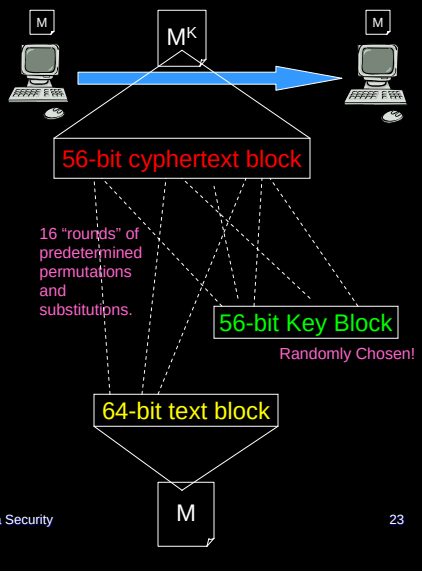
- Key Server
  - Manages unique keys for each node
- Process flow
  - Nodes first query KG for CypherText before sending (unique sender identifer included)
  - KG sends back CypherText and Key to Node A
  - Node A and Node B now communicate safely.



Cryptographic Control (3)

Data Encryption Standard (DES)

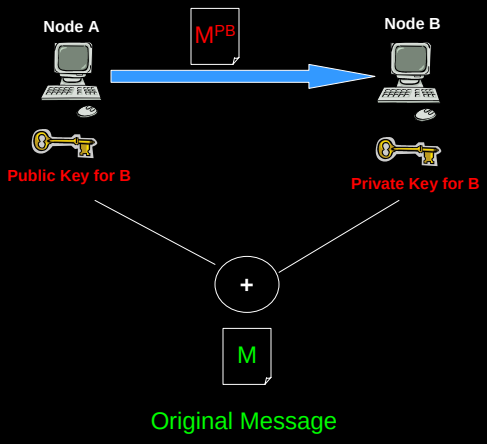
- 1st offical U.S. gov't std. intended for commercial use.
- 56-Bit private key
  - Applied to 64-bit plaintext block
  - 1 key / 72 quadrillion
- ***Goal***: Completely scramble data and key so that both are interdependent on each other.
- Key must exist on both ends (encrypt /decrypt)



Cryptographic Control (4)

Public Key Infrastructure (PKI)

- Each user has a public / private pair
- Public known to all, private only to the recipient.
- Send cyphertext encrypted with public key.
- Recipient use private key to decrypt Message.
- Private and Public keys scramble interdependently.
- Can be used like signature verifications.

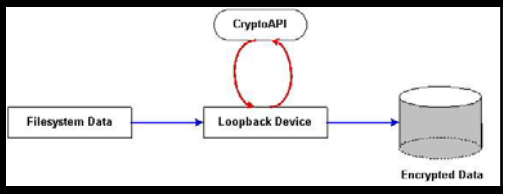


Cryptographic Control (5)

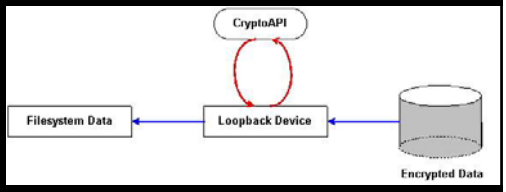
Linux CryptoAPI

- Adds cryptography framework to GNU / Linux kernel (v. 2.2 +).
- A Volume Encryptor
  - Bulk encryption only.
- User transparency
  - "loopback device" connected to the mount point.
  - System calls are intercepted for encryption / decryption.
- Transmission of data isn't considered in the purest model.
  - Just the filesystem.

Encryption



Decryption



Limitations

- Access Controls
  - Supervisory Control breaches
  - Over time, complexity increases and maintainability decreases.
- Flow Controls
  - Down-flow may lose data in "declassification" process
  - Controls need to be tightly tested and reliable
- Inference Controls
  - Subject still in its infancy
  - Goal – How much work does it take to break the system?
- Cryptographic Controls
  - Key generation server compromise?
  - PKI – What if private Key stolen?

Summary

- 4 overviews of Data Security Models.
- Access Control
  - Data Dependent
  - Object Dependent (NT File System, Capability Memory Addr.)
- Flow Control
  - Straightforward or Implicit
  - Protection Rings
- Inference Control
  - Defining Ranges / Overlap
  - Rounding / Error Induction.
- Encryption Control
  - Key Generation Server
  - PKI
  - Linux CryptoAPI example

Conclusions

- No one approach is better than the other.
- Many limitations in real life
- Nothing is 100% secure
- Find the “best fit” for the application / intention.

References

- Denning, Dorothy E. & Denning, Peter J. "Data Security." Computer Science Dept., Purdue University, West Lafayette, IN 47907. Computing Surveys, Vol. II, No. 3. September 1979
- "Access Control Lists (ACL) and Access Control Entries (ACE)". PC Guide Reference – New Technology File System (NTFS). http://www.pcguide.com/ref/hdd/file/ntfs/secAccess-c.html . Accessed on 10-23-2004.
- Khelalfa, Halim M. Basic Software Laboratory, Information Processing Dept. CERIST, Algiers, Algeria. 1997. halim@atakor.cerist.dz . http://www.unesco.org/webworld/public_domain/tunis97/com_54/com_54.html . Accessed on 10-24-2004.
- "Data Encryption Standard - A WhatIs.com definition". TechTarget.com network. http://searchsecurity.techtarget.com/sDefinition/0,,sid14_gci213893,00.html . Accessed on 10-24-2004.
- "The Data Encryption Standard". Personal website – Jeremy T. Teitelbaum (c. 1995). http://raphael.math.uic.edu/~jeremy/crypt/des.html Accessed on 10-26-2004.
- Rothman, Mike. "Public-key Encryption for Dummies." Network World. 17 May 1999. Obtained from Network World Fusion. http://www.nwfusion.com/news/64452_05-17-1999.html . Accessed on 10-28-2004.
- Dubrawsky, Ido. "Cryptographic Filesystems, Part One: Design and Implementation." Security Focus. 7 March 2003. http://www.securityfocus.com/infocus/1673 . Accessed on 10-31-2004.