

CIS/TCOM 551

Computer and Network Security

Spring 2005

Lecture 14

Announcements

- Final project is due on Friday, April 22nd.
 - Check web site for updates
 - Ask questions on the newsgroup (and check there for answers)

- Final exam:
 - Thurs. April 28th 11am - 1pm
 - Chemistry B13

Outline

- Access Control Concepts
 - Matrix, ACL, Capabilities
- OS Mechanisms
 - Multics
 - Ring structure
 - Amoeba
 - Distributed, capabilities
 - Unix
 - File system, Setuid
 - Windows
 - File system, Tokens, EFS
 - SE Linux
 - Role-based, Domain type enforcement
- Secure OS
 - Methods for resisting stronger attacks
- Assurance
 - Multi-level security (MLS)
 - Orange Book, TCSEC
 - Common Criteria
 - Windows 2000 certification
- Some slides courtesy of John Mitchell

Secure Operating Systems

- Extra mechanisms for extra security
- Follow design and implementation procedures
- Review of design and implementation
- Maintenance procedures

Will discuss

- Mechanisms associated with secure OS
- Standards for certification
 - Mostly used by government, some commercial interest

Sample Features of Trusted OS

- Discretionary access control (DAC)
 - Access control decisions are made by owner of the object
- Mandatory access control (MAC)
 - Access control decisions are made by a global administrative policy
 - takes precedence over DAC
- Object reuse protection
 - Write over old data when file space is allocated
- Complete mediation
 - Prevent any access that circumvents monitor
- Audit
 - See next slide
- Intrusion detection
 - Anomaly detection
 - Learn normal activity, Report abnormal actions
 - Attack detection
 - Recognize patterns associated with known attacks

Audit

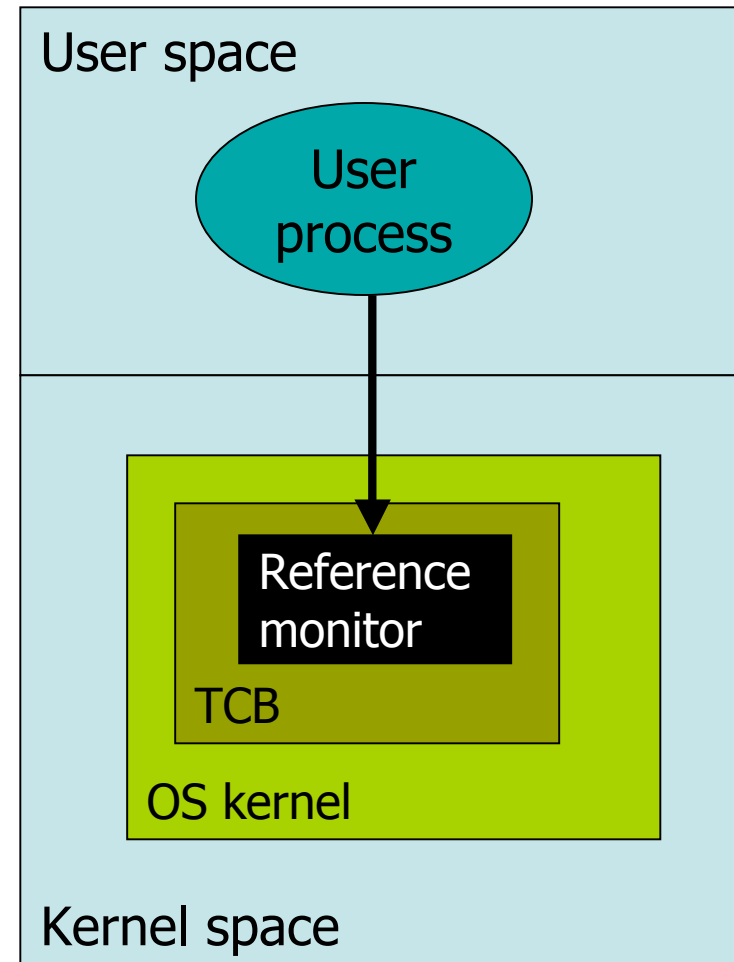
- Log security-related events
- Protect audit log
 - Write to write-once non-volatile medium
- Audit logs can become huge
 - Manage size by following policy
 - Storage becomes more feasible
 - Analysis more feasible since entries more meaningful
 - Example policies
 - Audit only first, last access by process to a file
 - Do not record routine, expected events
 - E.g., starting one process always loads library X ...

Trusted path

- Spoofing
 - Fool user/process into thinking they are communicating with secure part of system
 - Intercept communication
- Trusted path
 - Mechanisms to prevent spoofing
 - Special key sequence for passwd command intercepted by trusted kernel (e.g, ctrl-alt-delete)
 - Allow some actions only at boot time, before user processes loaded

Kernelized Design

- Trusted Computing Base
 - Hardware and software that must function correctly to enforce security policy
- Reference monitor
 - Part of TCB
 - All system calls go through reference monitor for security checking
 - Most OS are *not* designed this way



Encrypted File Systems

- Store files in encrypted form
 - Key management: user's key decrypts file
 - Useful protection if someone steals disk
- Windows – EFS
 - User marks a file for encryption
 - Unique file encryption key is created
 - Key is encrypted, can be stored on smart card
- Unix – CFS [Matt Blaze]
 - Transparent use
 - Local NFS server running on "loopback" interface
 - Key protected by passphrase
- Unix - Zero Interaction Authentication [Noble et al.]
 - Data is kept in encrypted form on disk
 - Users authenticate by possession of a cryptographic wireless fob
 - Whenever fob leaves vicinity of computer, applications are flushed to disk

Q: Why use crypto file system?

- General security questions
 - What properties are provided?
 - Against what form of attack?
- Crypto file system
 - What properties?
 - Secrecy, integrity, authenticity, ... ?
 - Against what kinds of attack?
 - Someone steals your laptop?
 - Someone steals your removable disk?
 - Someone has network access to shared file system?

Depends on how file system configured and used

SELinux

- Security-enhanced Linux system (NSA)
 - Enforce separation of information based on confidentiality and integrity requirements
 - Mandatory access control incorporated into the major subsystems of the kernel
 - Limit tampering and bypassing of application security mechanisms
 - Confine damage caused by malicious applications

<http://www.nsa.gov/selinux/>

SELinux Security Policy Abstractions

- Security-Encanced Linux
 - Built by NSA
- Type enforcement
 - Each process has an associated domain
 - Each object has an associated type
 - Configuration files specify
 - How domains are allowed to access types
 - Allowable interactions and transitions between domains
- Role-based access control
 - Each process has an associated role
 - Separate system and user processes
 - configuration files specify
 - Set of domains that may be entered by each role

Why Linux?

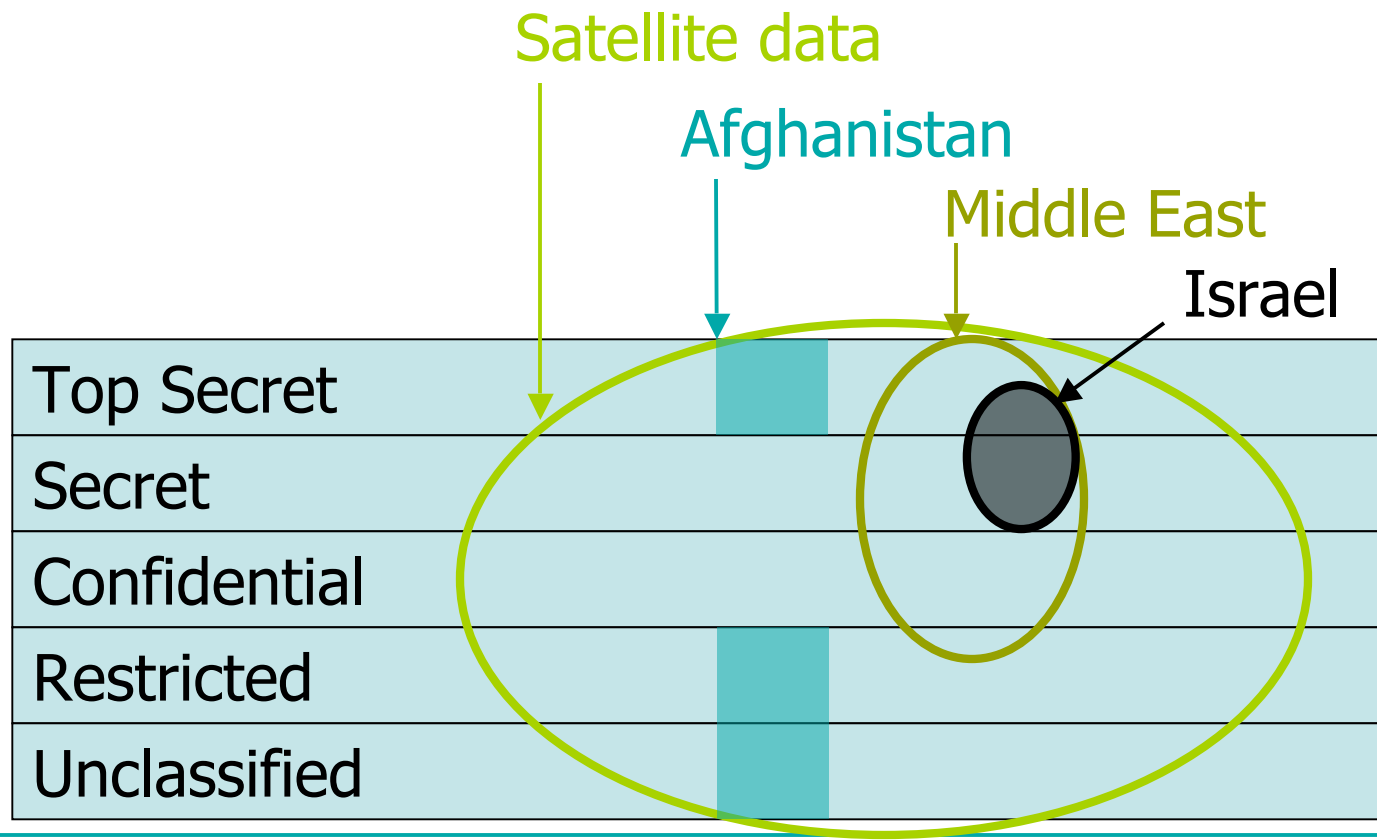
- Open source
 - Already subject to public review
 - This by itself does not guarantee security ...
 - NSA can review source, modify and extend
 - Hope to encourage additional operating system security research
 - Released under the same terms and conditions as the original sources.
 - includes documentation and source code

Multi-Level Security (MLS) Concepts

- Policies about **information-flow**
- Military security policy
 - Classification involves sensitivity levels, compartments
 - Do not let classified information leak to unclassified files
- Group individuals and resources
 - Use some form of hierarchy to organize policy
- Other policy concepts
 - Separation of duty
 - “Chinese Wall” Policy

Military security policy

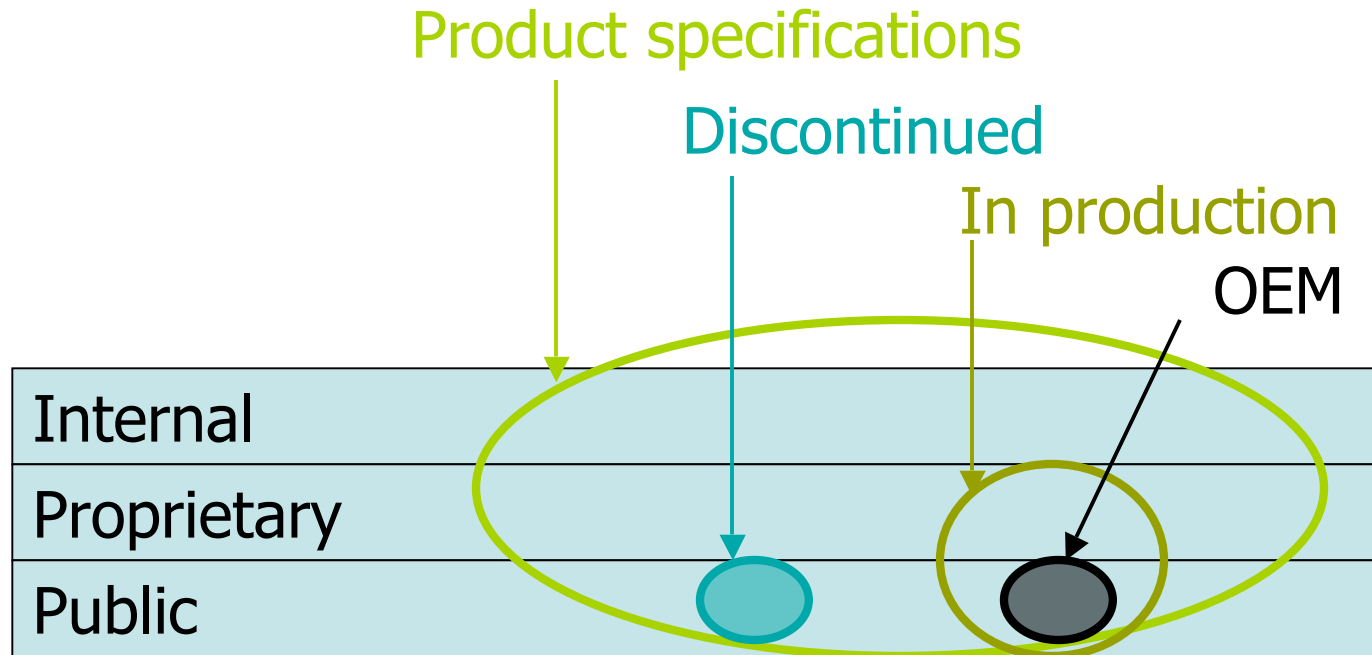
- Sensitivity levels
- Compartments



Military security policy

- Classification of personnel and data
 - Class = $\langle \text{rank, compartment} \rangle$
- Dominance relation
 - $D_1 \leq D_2$ iff $\text{rank}_1 \leq \text{rank}_2$
and $\text{compartment}_1 \subseteq \text{compartment}_2$
 - Example: $\langle \text{Restricted, Israel} \rangle \leq \langle \text{Secret, Middle East} \rangle$
- Applies to
 - Subjects – users or processes: $C(S)$ = "clearance of S"
 - Objects – documents or resources: $C(O)$ = "classification of O"

Commercial version

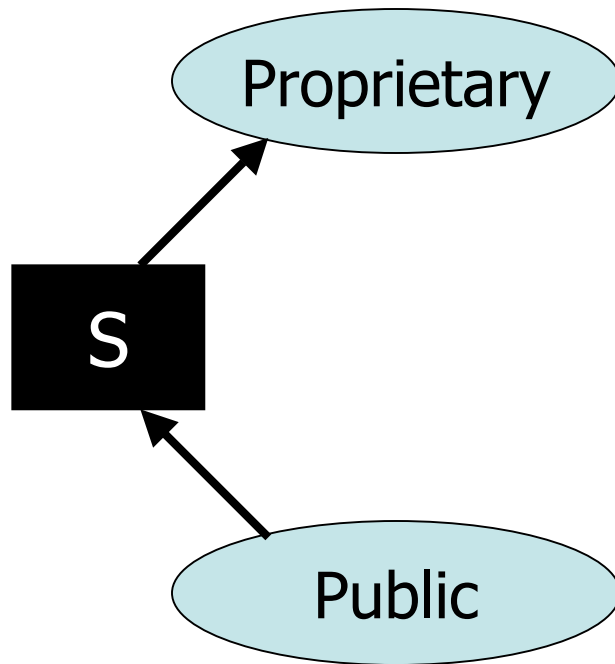


Bell-LaPadula Confidentiality Model

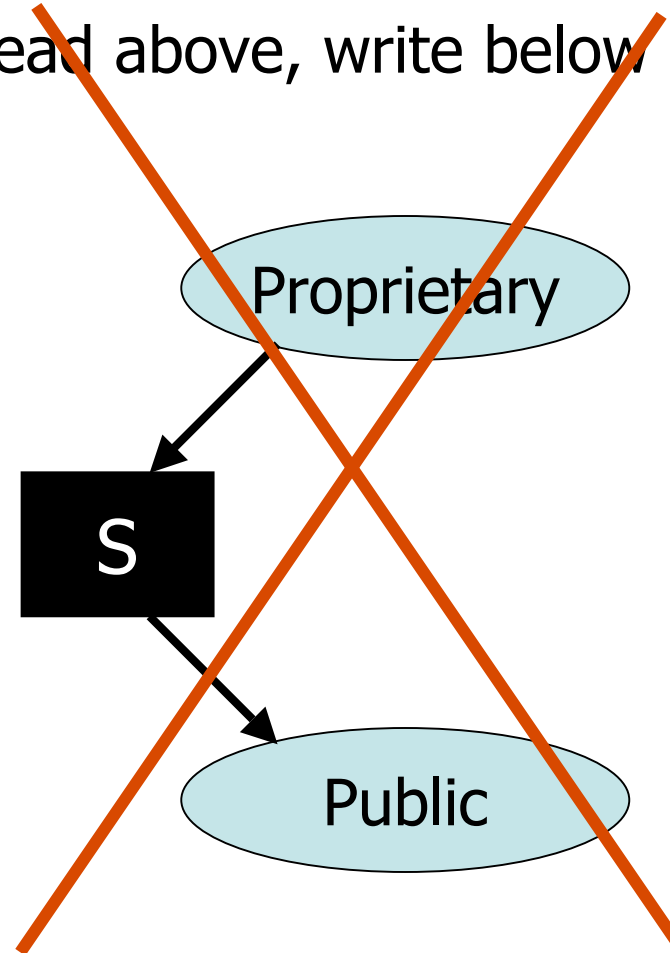
- When is it OK to release information?
- Two Properties (with silly names)
 - Simple security property
 - A subject S may read object O only if $C(O) \leq C(S)$
 - *-Property
 - A subject S with read access to O may write object P only if $C(O) \leq C(P)$
- In words,
 - You may only *read below* your classification and only *write above* your classification

Picture: Confidentiality

Read below, write above



~~Read above, write below~~

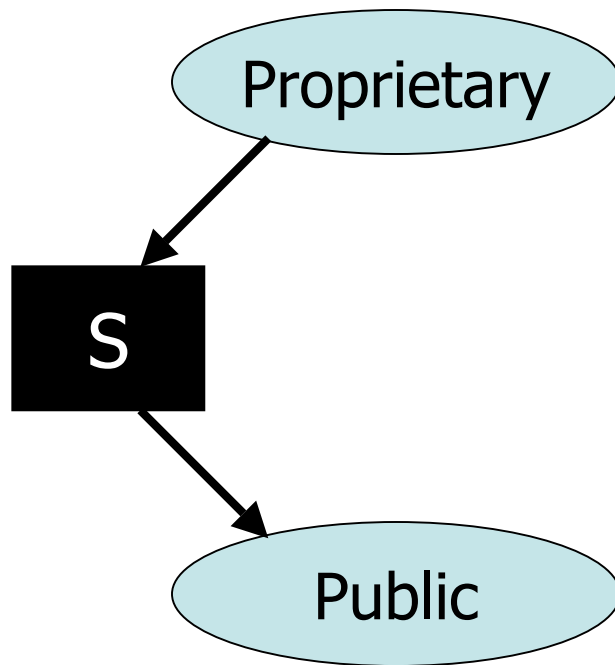


Biba Integrity Model

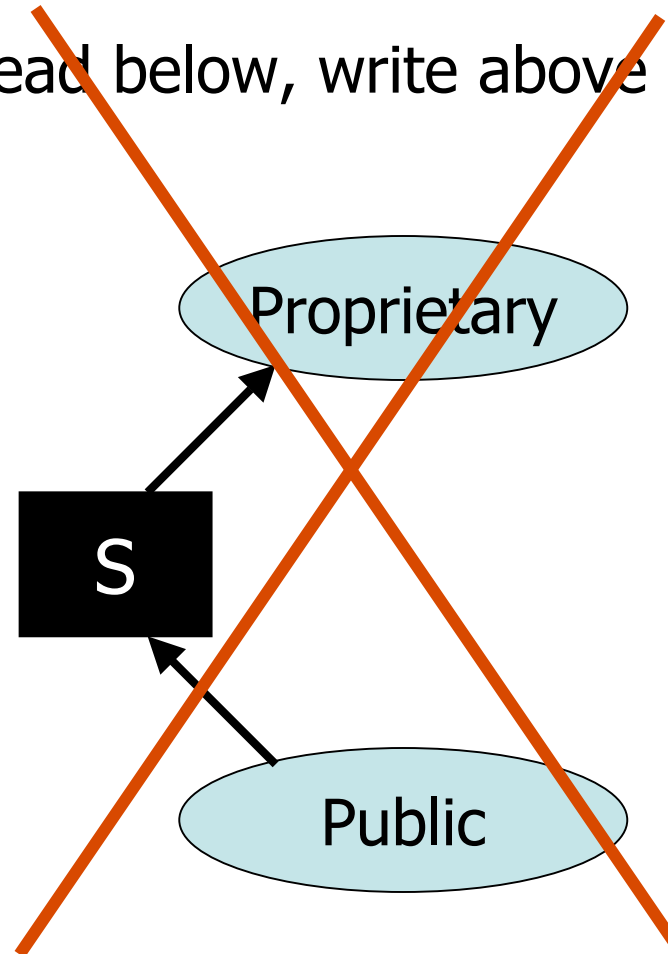
- Rules that preserve integrity of information
- Two Properties (with silly names)
 - Simple integrity property
 - A subject S may write object O only if $C(S) \geq C(O)$
(Only trust S to modify O if S has higher rank ...)
 - *-Property
 - A subject S with read access to O may write object P only if $C(O) \geq C(P)$
(Only move info from O to P if O is more trusted than P)
- In words,
 - You may only *write below* your classification and only *read above* your classification

Picture: Integrity

Read above, write below



Read below, write above



Problem: Models are contradictory

- Bell-LaPadula Confidentiality
 - Read down, write up
- Biba Integrity
 - Read up, write down
- Want both confidentiality and integrity
 - Example: May use Bell-LaPadula for some classification of personnel and data, Biba for another
 - Otherwise, only way to satisfy both models is only allow read and write at same classification

In reality: Bell-LaPadula used more than Biba model
Example: Common Criteria

Other policy concepts

- Separation of duty
 - If amount is over \$10,000, check is only valid if signed by two authorized people
 - Two people must be *different*
 - Policy involves role membership and ability to distinguish equality of principles.
- Chinese Wall Policy
 - Lawyers L1, L2 in Firm F are experts in banking
 - If bank B1 sues bank B2,
 - L1 and L2 can each work for either B1 or B2
 - No lawyer can work for opposite sides in any case
 - Permission depends on use of other permissions

These policies cannot be represented using access matrix