



# Design of Trusted Operating Systems


CS165, B. D. Fleisch
1


## Major Underpinnings of Trusted Operating Systems

- *Policy* – A set of well defined, consistent and implementable rules that have been clearly and unambiguously expressed
- *Model* – OS designer constructs a model of the environment to be secured and studies different ways of enforcing the security
- *Design* – what trusted OS is and how it will be constructed
- *Trust* – rules for believing it will meet our expectations. Depends on:
  - ◆ *Features* – OS must have functionality to implement the expected security policy
  - ◆ *Assurance* – OS is implemented in such a way that it will enforce the security policies


CS165, B. D. Fleisch
2


## Qualities of Security and Trustedness

|   |   |
|---|---|
| <p><b>Secure</b></p> <ul style="list-style-type: none"> <li>■ Either-or: something either is or is not secure</li> <li>■ Property of presenter</li> <li>■ Asserted: based on product characteristics</li> <li>■ Absolute: not qualified as to how, where, when or by whom used</li> <li>■ A goal</li> </ul> | <p><b>Trusted</b></p> <ul style="list-style-type: none"> <li>■ Graded: there are degrees of trustedness</li> <li>■ Property of receiver</li> <li>■ Judged: based on evidence or analysis</li> <li>■ Relative: viewed in context of use</li> <li>■ A characteristic</li> </ul> |
|---|---|


CS165, B. D. Fleisch
3

## Trusted Objects

- **Trusted Process** – process that can affect system security. A malicious execution is capable of violating system security policies
- **Trusted Product** – evaluated or approved products
- **Trusted Software** – software that can be relied upon to enforce a security policy
- **Trusted Computing base** – set of all protection mechanisms including hardware, firmware and software
- **Trusted system** – system employs sufficient hardware and software integrity measures to allow its use for processing sensitive information


CS165, B. D. Fleisch
4

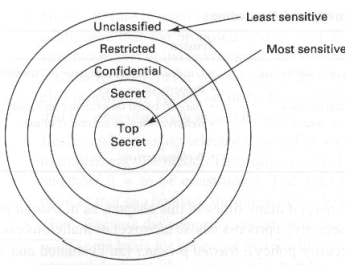
## Security Policies

- Policy is a statement of security we expect the system to enforce
- An OS is trusted with respect to the security policy
- Military policy
  - ◆ Basis of much work in operating systems
  - ◆ Can be stated precisely

University of Colorado Boulder CS165, B. D. Fleisch 5

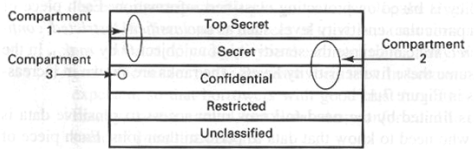
## Military Security Policy

- Information is classified:
  - ◆ Unclassified
  - ◆ Restricted
  - ◆ Confidential
  - ◆ Secret
  - ◆ Top Secret
- Sensitivity of an object  $O$  by rank $_O$
- Sensitivity requirements known as hierarchical requirements



University of Colorado Boulder CS165, B. D. Fleisch 6

## Information Access in Military Security Model

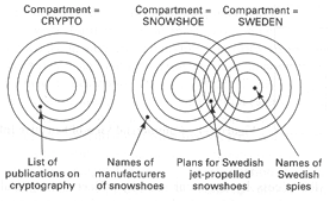


- Need-to-know rule: access to sensitive data is allowed only to subjects who need to know that data to perform their duties
- Each piece of classified information may be associated with one or more projects called *compartments*
- Compartments may include information at one sensitivity level only or more than one sensitivity level

University of Colorado Boulder CS165, B. D. Fleisch 7

## Information and Compartments


- $\langle \text{rank, compartments} \rangle$  is called the *class* or *classification* of a piece of information
- Example: one piece of information is list of pubs on cryptography another describes development of snowshoes in Sweden
- **Compartment:**
  - ◆ {crypto}
  - ◆ {snowshoe, Sweden}



University of Colorado Boulder CS165, B. D. Fleisch 8


## Clearances

- Person is trusted to access information up to a certain level of sensitivity
- The person needs to know certain categories of information
- Clearance: <rank; compartments>
- Dominance: for a subject  $s$  and object  $o$ :
  - ◆  $s \leq o$  iff
    - $rank_s \leq rank_o$  and
    - $compartments_s$  subset of  $compartments_o$
  - ◆  $o$  dominates  $s$  if  $s \leq o$
- A subject can read an object if
  - ◆ The clearance level of the subject is at least as high as that of the information
  - ◆ The subject has a need to know about all the compartments for which the information is classified

 CS165, B. D. Fleisch 9

## Examples

- Information classified <secret; {Sweden}>
- Could be read by
  - ◆ <top secret; {Sweden}>
  - ◆ <secret; {Sweden}> or <secret; {Sweden, crypto}>
- Not by:
  - ◆ <top secret; {crypto}> or
  - ◆ <confidential; {Sweden}>

 CS165, B. D. Fleisch 10

## Commercial Security Policies

- Introduction
- Clark-Wilson Commercial Security Policy
- Separation of Duty
- Chinese Wall Security Policy


 CS165, B. D. Fleisch 11

## Introduction

- Commercial policies may be less rigidly structured than military ones
- Corporate-level responsibilities tend to overlay projects and departments
  - ◆ People throughout the corporation may need accounting or personnel data
  - ◆ Data may have different degrees of sensitivity
  - ◆ Sensitivity due to projects: old standby have not need to know about new-product but reverse may not be true

|            |
|------------|
| Accounting |
| Personnel  |

|             |             |           |
|-------------|-------------|-----------|
| Old standby | New project | Project A |
|-------------|-------------|-----------|

 CS165, B. D. Fleisch 12

## Military vs Commercial

- **Commercial: no formalized notion of clearances**
  - ◆ Employee allowed access to internal example
  - ◆ Rules less regularized
  - ◆ No dominance function as in military systems



CS165, B. D. Fleisch

13

## Clark-Wilson Commercial Security Policy

- Integrity is important in commercial applications
- Clark-Wilson proposed a policy for *well-formed transactions*
  - ◆ Must be performed in correct order
  - ◆ Must be formed exactly as specified
  - ◆ Must be performed by authenticated individuals



CS165, B. D. Fleisch

14

## Example of well-formed transaction in Clark-Wilson Model

1. Purchasing clerk creates an order for a supply, sends copies of the order to both supplier and receiving depts
2. Supplier ships goods, which arrive at receiving dept, ensures that the correct quantity and right item has been received, and signs a delivery form. The delivery form and the original go to accounting
3. Supplier sends an invoice to the accounting dept. Accounting clerk compares invoice with the original order and the delivery form



CS165, B. D. Fleisch

15

## Rationale in Example

- Receiving clerk will not sign a delivery form without receiving a matching order
  - ◆ Prevents deliveries from suppliers of unwanted/unordered goods
- Accounting clerk will not issue a check for payment without having received a matching order and delivery form
  - ◆ Suppliers should not be paid for goods not ordered or received
- Both order and delivery form must be signed by authorized individual



CS165, B. D. Fleisch

16

### Clark Wilson Formal Notation

- Policy specified in terms of *constrained data items* which are processed by *transformation procedures*
- *Transformation procedure* is like a monitor because it performs only particular operations on specific kinds of data items
- These data items (CDIs) are manipulated only by transformation procedures
- Transformation procedures maintain the integrity of the data items by validating the processing to be performed
- Access triples:  $\langle \text{userid}, TP_i, \{CDI_j, CDI_k, \dots\} \rangle$ 
  - ◆ User
  - ◆ Transformation procedure
  - ◆ Constrained data items

CS165, B. D. Fleisch 17

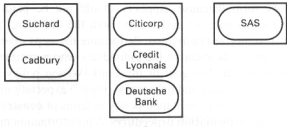
### Separation of Duty

- May not want the same person issuing the order for goods, receiving the goods, and writing checks
  - ◆ potential for abuse
- Division of responsibilities of this kind is called separation of duty
- Commonly accomplished with dual signatures
  - ◆ Clark-Wilson triples are stateless
  - ◆ Triple doesn't have a context of prior operations and thus incapable of passing information to other triples
  - ◆ If one person is authorized to do TP1 and TP2, Clark-Wilson cannot prevent same person from performing both TP1 and TP2
  - ◆ So must be stated as a policy requirement

CS165, B. D. Fleisch 18

### Chinese Wall Security Policy

- Concerns conflict of interest
- Builds on three level of abstraction:
  - ◆ Objects: e.g. files each file concerns only one company
  - ◆ Company groups: all objects concerning one company are grouped together
  - ◆ Conflict classes: all groups of objects for competing companies are clustered



Initial Options

CS165, B. D. Fleisch 19

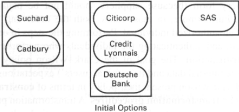
### Chinese Wall Security Policy (2)

- Access policy: A person can access any information as long as the person has never accessed information from a different company in the same conflict class
- Object must be part of the same company group as the object that has been previously accessed or the object belongs to a company conflict class that has never before been accessed

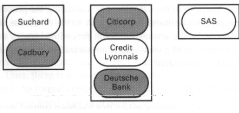
CS165, B. D. Fleisch 20

### Example

- Initially can access any objects. Suppose you read a file from Suchard
- A subsequent request for access to any bank or to SAS would be granted, but a request to Cadbury would be denied
- Access to SAS will not affect future accesses
- Why is it interesting?
  - ◆ Access permissions change dynamically as a subject accesses some objects, other objects that would have been accessible now are denied



Initial Options



University of California Berkeley CS165, B. D. Fleisch 21

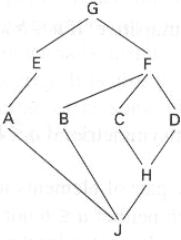
### Multilevel Security

- Military has developed extensive procedures for securing information
- Generalization of military model of information security has been adopted as a model of data security within an operating system
- Bell and La Padula describe in formal notation
- Denning formalized the structure of this model. The generalized model is called the lattice model of security

University of California Berkeley CS165, B. D. Fleisch 22

### Lattice Model

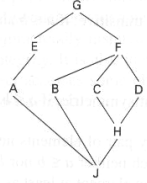
- Military model is a representative of a more general scheme called a lattice
- Lattice: elements form a mathematical structure under a relational operator
- Elements of a lattice are ordered under a partial ordering  $\leq$
- A partial ordering is a relation  $\leq$  that is
  - ◆ Transitive (if  $a \leq b$  and  $b \leq c$  then  $a \leq c$ )
  - ◆ Antisymmetric (if  $a \leq b$  and  $b \leq a$  then  $a = b$ )



University of California Berkeley CS165, B. D. Fleisch 23

### Lattices have non-comparable elements

- There may be elements for which neither  $a \leq b$  nor  $b \leq a$
- There will be an upper bound element even though some elements are not comparable s.t.  $a \leq u$  and  $b \leq u$
- There will be a lower bound element  $l$  dominated by both  $a$  and  $b$  s.t.  $l \leq a$  and  $l \leq b$
- EXAMPLE: B and H



University of California Berkeley CS165, B. D. Fleisch 24

## Military Model as Lattice

- The relation  $\leq$  is defined in the military model is the relation for the lattice.
- The relation  $\leq$  is transitive and antisymmetric
- Largest element is <top secret; all compartments>
- Smallest element is <unclassified; no compartments>
- Commercial securities policies can be lattice models as well



CS165, B. D. Fleisch

25

## Bell-La Padula Confidentiality Model

- Goal: Identify allowable communication where it is important to maintain security
- Formal description of allowable information flow paths
- Set of subjects  $S$  and objects  $O$
- Each subject  $s$  in  $S$  and each object  $o$  in  $O$  has a fixed security class  $C(s)$  and  $C(o)$ .
- Security classes ordered by relation  $\leq$



CS165, B. D. Fleisch

26

## Bell-La Padula Confidentiality Model

- Simple security: A subject  $s$  may have read access to an object  $o$  iff  $C(o) < C(s)$ 
  - ◆ Person must have clearance at least as high as the information
- \*-property: A subject  $s$  who has read access to an object  $o$  may have write access to an object  $p$  only if  $C(o) \leq C(p)$ 
  - ◆ Contents of an object can only be written to objects with at least the same level of sensitivity
- \*-property prevents write-down to lower levels and disclose



CS165, B. D. Fleisch

27

## Biba Integrity Model

- Biba model addresses *integrity* whereas Bell-La Padula concerns *disclosure* of information
- Subjects and objects are ordered by an integrity scheme denoted  $I(s)$  and  $I(o)$
- Simple Integrity Property: Subject  $s$  can modify (or have write access to) object  $o$  iff  $I(s) \geq I(o)$ 
  - ◆ Person must have clearance to write data (and only to a lower level)
- Integrity \*-property: If subject  $s$  has read access to object  $o$  with integrity level  $I(o)$ ,  $s$  can have write access to object  $p$  iff  $I(o) \geq I(p)$ 
  - ◆ Cant write up, prevents write-up to higher levels and thus downgrading the sensitivity (reducing the integrity of the higher integrity data)



CS165, B. D. Fleisch

28

## Graham-Denning Access Model

Table 7-2 Protection System Commands

| Command   | Pre-Condition  | Effect   |
|---|--|--|
| Create object <i>o</i>  | —  | Add column for <i>o</i> in <i>A</i> ; place <i>owner</i> in <i>A</i> [ <i>s</i> , <i>o</i> ] |
| Create subject <i>s</i>   | —  | Add row for <i>s</i> in <i>A</i> ; place <i>control</i> in <i>A</i> [ <i>s</i> , <i>o</i> ]  |
| Delete object <i>o</i>  | <i>Owner</i> in <i>A</i> [ <i>s</i> , <i>o</i> ]   | Delete column <i>o</i>   |
| Delete subject <i>s</i>   | <i>Control</i> in <i>A</i> [ <i>s</i> , <i>s</i> ]   | Delete row <i>s</i>  |
| Read access right of <i>s</i> on <i>o</i>                           | <i>Control</i> in <i>A</i> [ <i>s</i> , <i>s</i> ] or <i>owner</i> in <i>A</i> [ <i>s</i> , <i>o</i> ] | Copy <i>A</i> [ <i>s</i> , <i>o</i> ] to <i>s</i>  |
| Delete access right <i>r</i> of <i>s</i> on <i>o</i>                | <i>Control</i> in <i>A</i> [ <i>s</i> , <i>s</i> ] or <i>owner</i> in <i>A</i> [ <i>s</i> , <i>o</i> ] | Remove <i>r</i> from <i>A</i> [ <i>s</i> , <i>o</i> ]  |
| Grant access right <i>r</i> to <i>s</i> on <i>o</i>                 | <i>Owner</i> in <i>A</i> [ <i>s</i> , <i>o</i> ]   | Add <i>r</i> to <i>A</i> [ <i>s</i> , <i>o</i> ]   |
| Transfer access right <i>r</i> or <i>rs</i> to <i>s</i> on <i>o</i> | <i>rs</i> in <i>A</i> [ <i>s</i> , <i>o</i> ]  | Add <i>r</i> or <i>rs</i> to <i>A</i> [ <i>s</i> , <i>o</i> ]                                |

- Model operates on a set of subjects, objects, rights and an access matrix
- One row for each subject, one column for each object
- Elements of the matrix show rights for subject or object in that cell
- Owner has special rights
- Controller has special rights
- Transferable right denoted *r\**, non-transferable right denoted *r*



## Graham-Denning Access Model

|           | BIBLOG | TEMP | F   | HELP.TXT | C_COMP | LINKER | SYS_CLOCK | PRINTER |
|-----------|--------|------|-----|----------|--------|--------|-----------|---------|
| USER A    | ORW    | ORW  | ORW | R        | X      | X      | R         | W       |
| USER B    | R      | —    | —   | R        | X      | X      | R         | W       |
| USER S    | RW     | —    | R   | R        | X      | X      | R         | W       |
| USER T    | —      | —    | —   | R        | X      | X      | R         | W       |
| SYS_MGR   | —      | —    | —   | RW       | OX     | OX     | ORW       | O       |
| USER_SVCS | —      | —    | —   | O        | X      | X      | R         | W       |

Figure 6-13 Access Control Matrix

