

Overview

- We take a look at one particularly influential confidentiality model for multilevel systems: the **Bell-La Padula** model
- It models confidentiality policies suitable for use in military and government systems where secrecy is paramount
- The goal is to prevent users (subjects) from accessing information (objects) for which they have not been cleared
- Material has been drawn from:
 - *Introduction to Computer Security* by Bishop
 - *Computer Security: Principles and Practice* by Stallings
 - *Access Control: Policies, Models and Mechanisms* by Samarati and di Vimercati

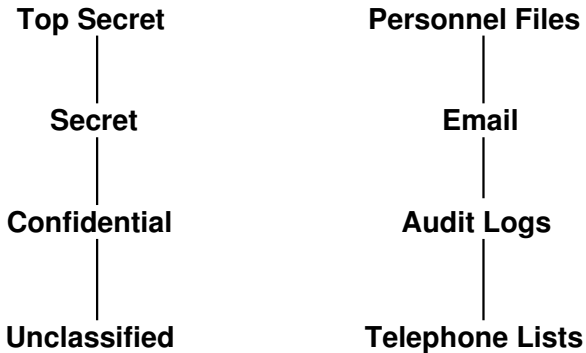
Bell-La Padula

- In the early 70s the US military wanted to avail of the time-sharing computer systems that were coming available
- An access control system that enforced confidentiality across multiple users of varying rank and data of varying sensitivity and category was required
- The success of “tiger teams” in breaking existing systems meant a new approach to secure design and implementation was required
- Much effort was invested in developing models that could be used in proving a security policy was both sound and faithfully enforced
- One such model was Bell-La Padula

Security Classification, Clearance and Level

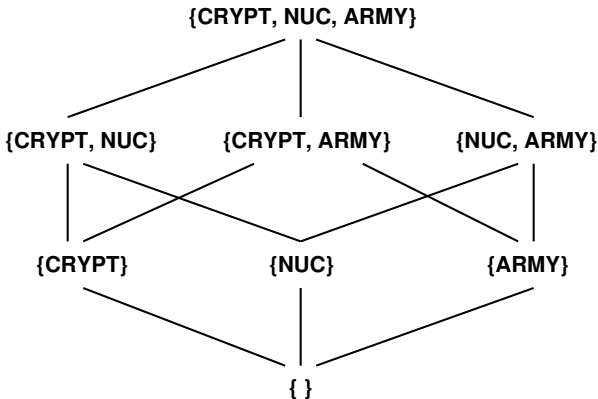
- In the Bell-La Padula model each subject and object are assigned a **security classification**
- The classification of a subject is also referred to as that subject's **security clearance**
- Subjects and objects may also be assigned to zero or more **categories** where categories describe a type of information and capture the “need to know” principle
- Together, the pair {classification, set of categories} is known as a **security level**

Security Classification



Security Categories

The set of categories a subject may have access to is the power set of categories and they form a lattice under the operation of \subseteq , e.g.



Informal Description

- **No read-up:** A subject can only read an object of lower or equal security level, in the model this is referred to as the **simple security property** or **ss-property**
- **No write-down:** A subject can only write into an object of greater or equal security level, in the model this is referred to as the ***-property**
- **DAC:** A subject may grant/revoke access to objects as long as doing so does not violate the MAC rules above, in the model this is referred to as the **ds-property**

Why No Write-Down?

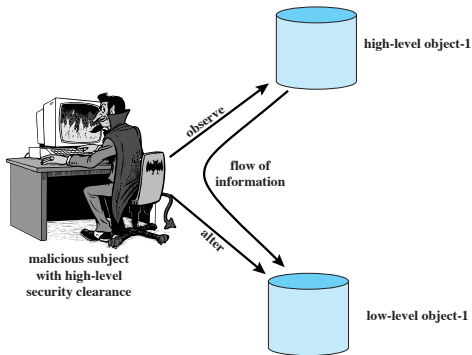


Figure 10.1 Information Flow Showing the Need for the *-property

Access Modes

In the original description of the model, for each object four access modes may apply:

- **read**: The subject is allowed read access to the object
- **append**: The subject is allowed only write access to the object
- **write**: The subject is allowed both read and write access to the object
- **execute**: The subject is allowed invoke the object for execution

Dominance

- Security level $\{Class_1, CategorySet_1\}$ **dominates** level $\{Class_2, CategorySet_2\}$ if and only if:
 1. $Class_1 \geq Class_2$ and
 2. $CategorySet_1 \supseteq CategorySet_2$
- Or to put it another way:
 $(C_1, CS_1) \succeq (C_2, CS_2) \Leftrightarrow C_1 \geq C_2 \wedge CS_1 \supseteq CS_2$
- Dominance is **reflexive** so $\forall x \in SL : x \succeq x$
- Dominance is **transitive** so
 $\forall x, y \in SL : x \succeq y, y \succeq z \Rightarrow x \succeq z$

Formal Description

- The current state of the system, Σ , is described by the 4-tuple (b, M, f, H)
- **Current access set b** : A set of triples of the form (subject, object, access mode) where a triple (S, O, A) means that that subject S is **currently exercising** access to O in access mode A
- **Access matrix M** : The access matrix has the now familiar structure where element M_{ij} records the permitted access rights of S_i over O_j

Formal Description

- **Level function f :** This function assigns a security level to each subject and object and consists of 3 mappings:
 1. $f_o(O_j)$ is the security level assigned to object O_j
 2. $f_s(S_i)$ is the maximum security clearance of subject S_i
 3. $f_c(S_i)$ is the current security level of subject S_i
- The maximum security clearance of a subject is the maximum security level at which they may operate
- The ability to log in at lower levels facilitates role based access control
- **Hierarchy H :** A rooted, directed tree where the nodes are objects and the security level of an object dominates that of its children

Bell-La Padula Properties

ss-property

- Every triple of the form $(S_i, O_j, read)$ in the current access set b has the property $f_c(S_i) \succeq f_o(O_j)$

*-property

- Every triple of the form $(S_i, O_j, append)$ in the current access set b has the property $f_c(S_i) \preceq f_o(O_j)$
- Every triple of the form $(S_i, O_j, write)$ in the current access set b has the property $f_c(S_i) = f_o(O_j)$

Bell-La Padula Properties

ds-property

- If (S_i, O_j, A_x) is in the current access set b then access mode A_x is recorded in the (S_i, O_j) element of M , or,
 $(S_i, O_j, A_x) \implies A_x \in M[S_i, O_j]$

Basic Bell-La Padula Security Theorem

These three properties can then be used to define a confidentiality secure system:

- Let Σ be a system with secure initial state σ_0 and T be a set of state transformations: if every element of T preserves the three conditions then every state $\sigma_i, i \geq 0$ is secure
- Thus, given an actual design and implementation it is theoretically possible to prove it secure by showing that any action will lead to a new state that satisfies the three properties

Bell-La Padula Abstract Operations

1. **Get access:** add a triple (S, O, A) to b
2. **Release access:** remove a triple (S, O, A) from b
3. **Change object level:** change the value of $f_o(O_j)$ for some object O_j
4. **Change current level:** change $f_c(S_i)$ for some subject S_i
5. **Give access permission:** add access mode A to M_{ij}
6. **Rescind access permission:** remove access mode A from M_{ij}
7. **Create object:** add an object to the hierarchy H
8. **Delete object:** remove an object from the hierarchy H

Practicalities

- It may be necessary to occasionally allow access to content created at a high security level to subjects operating at a lower security level by writing the data to a lower security level
- Ordinarily this would violate the ***-property** so a security administrator operating outside the model must be assigned downgrading authority so that the release of information can proceed in a controlled manner

Biba Integrity Model

- The Bell-La Padula model deals with confidentiality while the Biba model deals with integrity and is concerned with the unauthorised **modification** of data
- Biba is intended to deal with the case where there is data that must be visible to users at multiple levels but should only be modifiable by authorised agents
- As with Bell-La Padula, classifications and categories impose a dominance relation across levels but we speak of **integrity levels** rather than security levels
- As an integrity model, Biba's goal is to ensure content can be trusted

Biba Access Modes

- **Read:** To read information in an object
- **Write:** To write or update information in an object
- **Execute:** To execute an object
- **Invoke:** To communicate with another subject

The Rules

- **Simple integrity:** A subject can modify an object only if the integrity level of the subject dominates the integrity level of the object or **no write-up**
- **Integrity confinement:** A subject can read an object only if the integrity level of the subject is dominated by the integrity level of the object or **no read-down**
- **Invocation property:** A subject can invoke (send a message to) another subject only if the integrity level of the first subject dominates the integrity level of the second

The Rules

- The **simple integrity rule** prevents contamination of high integrity data by low security level subjects
- The **integrity confinement rule** prevents a Trojan copying low integrity data into a high integrity file
- The rules are analogous to those met in Bell-La Padula but reverse the significance of read and write illustrating the incompatibility of maintaining confidentiality and integrity across multilevel security systems