



## Confidentiality Policies

---

- Overview
  - What is a confidentiality model
- Bell-LaPadula Model
  - General idea
  - description of rules
- MAC tuples

1



## Confidentiality Policy

---

- Goal: prevent the unauthorized disclosure of information
  - Deals with information flow
  - Integrity incidental
- Multi-level security models are best-known examples
  - Bell-LaPadula Model basis for many, or most, of these

2



## Bell-LaPadula Model (BLP)

---

- Subjects and objects labeled with **security levels** that form a partial ordering.
- **The policy: No information flow from 'higher' security levels down to 'lower' security level (confidentiality).**
- Only considers information flows that occur when a subject observes or alters an object.
- Access permissions defined through an **access control matrix** and **security levels**.

3



## Constructing the State Set

---

All current access operations:

- an access operation is described by a triple  $(s, o, a)$ ,  $s \in S$ ,  $o \in O$ ,  $a \in A$   
e.g. (Alice, fun.com, read)
- The set of all current access operations is an element of  $P(S \times O \times A)$   
e.g.  $\{(Alice, fun.com, read), (Bob, fun.com, write), \dots\}$

4



## Constructing the State Set

---

### Current assignment of security levels:

- maximal security level:  $f_S: S \rightarrow L$  ( $L$  ... labels)
  - current security level:  $f_C: S \rightarrow L$
  - classification:  $f_O: O \rightarrow L$
- 
- The security level of a user is the user's clearance.
  - Current security level allows subjects to be down-graded temporarily (more later).
  - $F \subseteq L^S \times L^S \times L^O$  is the set of security level assignments;  $f = (f_S, f_C, f_O)$  denotes an element of  $F$ .

5



## Constructing the State Set

---

### Current permissions:

- defined by the access control matrix  $M$ .
  - $\mathcal{M}$  is the set of access control matrices.
- 
- The state set of BLP:  $V = B \times \mathcal{M} \times F$ 
    - $B$  is our shorthand for  $P(S \times O \times A)$
    - $b$  denotes a set of current access operations
    - a state is denoted by  $(b, M, f)$

6



## BLP Policies

---

- **Discretionary Security (ds)-Property :**  
Access must be permitted by the access control matrix: if  $(s,o,a) \in b$ , then  $a \in M_{so}$ .
- **Simple Security (ss)-Property (no read-up):**  
if  $(s,o,a) \in b$  and access is in observe mode, then  $f_s(s) \geq f_o(o)$ .
- The ss-property is a familiar policy for controlling access to classified paper documents.

7



## Example

---

<i>security level</i>	<i>subject</i>	<i>object</i>
Top Secret	Tamara	Personnel Files
Secret	Samuel	E-Mail Files
Confidential	Claire	Activity Logs
Unclassified	Ursula	Telephone Lists

- Tamara can read all files
- Claire cannot read Personnel or E-Mail Files
- Ursula can only read Telephone Lists

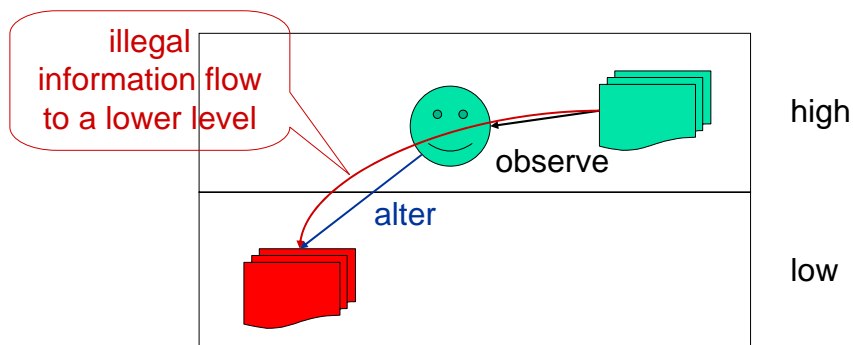
8

## On Subjects

- In the ss-property, subjects act as observers.
- In a computer system, subjects are *processes* and have no memory of their own.
- Subjects have access to memory objects.
- Subjects can act as channels by reading one memory object and transferring information to another memory object.
- In this way, data may be declassified improperly.

9

## Subjects as Channels



10

## Star Property

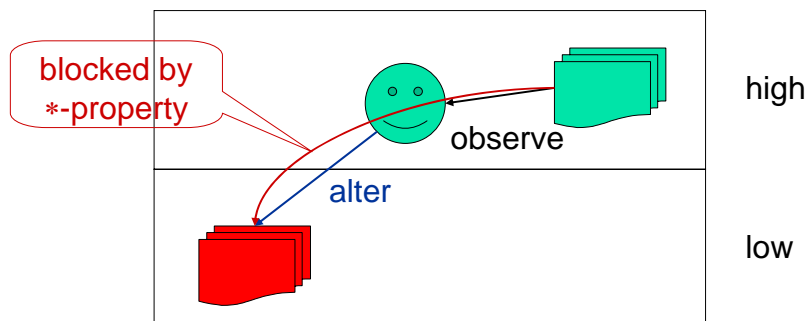
- **\*-Property (star property) (no write-down):**  
if  $(s,o,a) \in b$  and access is in alter mode, then  $f_C(s) \leq f_O(o)$  ; also, if subject  $s$  has access to object  $o$  in alter mode, then  $f_O(o') \leq f_O(o)$  for all objects  $o'$  accessed by  $s$  in observe mode.

( first version of BLP did not have \*-property)

- Mandatory BLP policies: ss-property and \*-property.

11

## Blocking the Channel



12



## No Write-Down

---

- The \*-property prevents high level subjects from sending legitimate messages to low level subjects.
- Two ways to escape from this restriction:
  - Temporarily downgrade high level subject; hence the current security level  $f_C$ ; BLP subjects have no memory of their own!
  - Exempt trusted subjects from the \*-property.
- Redefine the \*-property and demand it only for subjects that are not trusted.

13



## Trusted Subjects

---

**Trusted subjects may violate security policies! Distinguish between trusted subjects and trustworthy subjects.**

14



## Basic Security Theorem

---

- A state is **secure**, if all current access tuples  $(s,o,a)$  are permitted by the *ss-*, *\*-*, and *ds-* properties.
- A *state transition is secure* if it goes from a secure state to a secure state.

**Basic Security Theorem:** If the initial state of a system is secure and if all state transitions are secure, then the system will always be secure.

15



## BLP & Security

---

- Construct system with operation *downgrade*:
  - downgrades all subjects and objects to system low.
  - enters all access rights in all positions of the access control matrix.
- As a result, any state is secure in the BLP model.
- Should such a system be regarded secure?
  - McLean: no, everybody is allowed to do everything.
  - Bell: yes, if *downgrade* was part of the system specification.

16





## Example

---

- Expand notion of security level (topsecret, secret, confidential, unclassified) to include categories
- Security level is (*clearance, category set*)
  - $(A, C)$  dominates  $(A', C')$  iff  $A' \leq A$  and  $C' \subseteq C$
  - Examples
    - (Top Secret, {Aus,Asi}) *dom* (Secret, {Aus})
    - (Secret, {Aus, Eur}) *dom* (Confidential, {Aus,Eur})
    - (Top Secret, {Aus}) *not dom* (Confidential, {Eur})

"greater than" is a total ordering on clearance, "dominates" is not

17



## Levels and Lattices

---

- Security levels partially ordered
  - Any pair of security levels may (or may not) be related by *dom*
- If  $C$  is the set of classifications, and  $K$  the powerset of the set of categories, then the set of security levels  $L = C \times K$  with partial order *dom* forms a lattice
  - $\text{lub}(L) = (\text{max}(C), \text{set of categories})$
  - $\text{glb}(L) = (\text{min}(C), \emptyset)$


18

## DataGeneral B2 UniX System

- Provides mandatory access controls
  - MAC label identifies security level
  - Default labels discussed here, but can define others
- Initially
  - Subjects assigned MAC label of parent
    - Initial label assigned to user, kept in Authorization and Authentication database
  - Object assigned label at creation
    - Explicit labels stored as part of attributes
    - Implicit labels determined from parent directory

19

## The three MAC Regions

Hierarchy levels ↑		A&A database, audit	Administrative Region
		User data and applications	User Region
	VP-1	Site executables	
	VP-2	Trusted data	Virus Prevention Region
	VP-3	Executables not part of the TCB	
VP-4	Executables part of the TCB		
VP-5	Reserved for future use		
	Categories		

IMPL\_HI is “maximum” (least upper bound) of all levels

IMPL\_LO is “minimum” (greatest lower bound) of all levels

20



## Directory Problem

---

- Process  $p$  at MAC\_A tries to create file  $/tmp/x$
- $/tmp/x$  exists but has MAC label MAC\_B
  - Assume MAC\_B dom MAC\_A
- Create fails
  - Now  $p$  knows a file named  $x$  with a higher label exists (covert channel)
- Fix: only programs with same MAC label as directory can create files in the directory
  - Now compilation will not work (access to  $/tmp$ ), mail cannot be delivered ( $/var/mail$ )

21



## Multilevel Directory

---

To solve previous problem

- Directory with a set of subdirectories, one per label
  - Not normally visible to user
  - $p$  creating  $/tmp/x$  actually creates  $/tmp/d/x$  where  $d$  is directory corresponding to MAC\_A
  - All  $p$ 's references to  $/tmp$  go to  $/tmp/d$

22



## Using MAC Labels

---

- Simple security condition implemented
- \*-property not fully implemented
  - Process MAC must equal object MAC
  - Writing allowed **only** at same security level
- Overly restrictive in practice  
So ... assign range

23



## MAC Tuples

---

- Up to 3 MAC ranges (one per region)
- MAC range is a set of labels with upper, lower bound
  - Upper bound must dominate lower bound of range
- Examples of range
  1. [(Secret, {NUC}), (Top Secret, {NUC})]
  2. [(Secret,  $\emptyset$ ), (Top Secret, {NUC, EUR, ASI})]
  3. [(Confidential, {ASI}), (Secret, {NUC, ASI})]

24



## MAC Ranges

---

1. [(Secret, {NUC}), (Top Secret, {NUC})]
  2. [(Secret,  $\emptyset$ ), (Top Secret, {NUC, EUR, ASI})]
  3. [(Confidential, {ASI}), (Secret, {NUC, ASI})]
- (Top Secret, {NUC}) in ranges 1, 2
  - (Secret, {NUC, ASI}) in ranges 2, 3
  - [(Secret, {ASI}), (Top Secret, {EUR})] not valid range
    - as (Top Secret, {EUR})  $\neg dom$  (Secret, {ASI})

25



## Objects and Tuples

---

- Objects must have MAC labels
  - May also have MAC tuple
  - If both, tuple overrides label
- Example
  - Paper has MAC range:  
[(Secret, {EUR}), (Top Secret, {NUC, EUR})]

26



## MAC Tuples

---

- Process can read object when:
  - Object MAC range  $(lr, hr)$ ; process MAC label  $pl$
  - $pl \text{ dom } hr$ 
    - Process MAC label grants read access to upper bound of range
- Example
  - Peter, with label (Secret, {EUR}), cannot read paper
    - (Top Secret, {NUC, EUR})  $\text{dom}$  (Secret, {EUR})
  - Paul, with label (Top Secret, {NUC, EUR, ASI}) can read paper
    - (Top Secret, {NUC, EUR, ASI})  $\text{dom}$  (Top Secret, {NUC, EUR})

27



## MAC Tuples

---

- Process can write object when:
  - Object MAC range  $(lr, hr)$ ; process MAC label  $pl$
  - $pl \in (lr, hr)$ 
    - Process MAC label grants write access to any label in range
- Example
  - Peter, with label (Secret, {EUR}), can write paper
    - (Top Secret, {NUC, EUR})  $\text{dom}$  (Secret, {EUR}) and (Secret, {EUR})  $\text{dom}$  (Secret, {EUR})
  - Paul, with label (Top Secret, {NUC, EUR, ASI}), cannot write paper
    - (Top Secret, {NUC, EUR, ASI})  $\text{dom}$  (Top Secret, {NUC, EUR})

28



## Key Points

---

- Confidentiality models restrict flow of information
- Bell-LaPadula models multilevel security
  - Cornerstone of much work in computer security