

Security Policy

- A document that expresses clearly and concisely what the protection mechanisms are to achieve
- A statement of the security we expect the system to enforce

Formal Security Models

CS177 2013 1

Bell & LaPadula Model

Formalization and specialization of the access model

- **Security level** is a classification and a set of categories
- Subjects and Objects have security levels
- Subjects also have a **current security level**

Formal Security Models

CS177 2013 2

Bell & LaPadula Model

- **Security level** is a classification and a set of categories
- Subjects and Objects have security levels
- Subjects also have a **current security level**
- Some subjects are **trusted** to violate the basic policies of the model. This extends the model

Formal Security Models

CS177 2013 3

Security level SL1 **dominates** security level SL2 iff

- SL1's classification \geq SL2's classification
- SL1's categories are a superset of SL2's categories

Formal Security Models

CS177 2013 4

Types of Access

- Read-only
- Append
- Execute
- Read-write

Formal Security Models

CS177 2013 5

Permission Matrix

- Contains an entry for each subject-object pair
- Entries indicate the type of access that may be granted to the subject for the object

Formal Security Models

CS177 2013 6

Permission Matrix

- Contains an entry for each subject-object pair
- Entries indicate the type of access that may be granted to the subject for the object

Current Access

- List of subject, object, access triples that indicate all of the currently granted accesses

Formal Security Models

CS177 2013 7

Basic Security Theorem

A system is secure iff

- Its initial state is secure
- Each action that starts in a secure state results in a secure state

Formal Security Models

CS177 2013 8

Basic Security Theorem

A system is secure iff

- Its initial state is secure
- Each action that starts in a secure state results in a secure state

Secure State Satisfies

- SS - property
- * - property
- DS - property

Formal Security Models

CS177 2013 9

Invariants for the Model

- **Current security level**
 - every subject's current security level must be dominated by its security level
- **Simple security condition**
 - if subject S currently has read-only or read-write access to object O, then the security level of S must dominate the security level of O

Formal Security Models

CS177 2013 10

- **Star property**

- if subject S currently has read-only or read-write access to object O1 and has append or read-write access to object O2, then the security level of O2 must dominate the security level of O1 or S must be a trusted subject

- **Discretionary security property**

- if subject S currently has access A for object O, then the permission matrix entry for S,O must contain A

Formal Security Models

CS177 2013 11

A **trusted subject** can be relied on not to compromise security even if some of its current accesses violate the star property

Formal Security Models

CS177 2013 12

The model has a set of rules for changing state

Rules define the allowable ways that a system can transition from one state to another

Rule 1 Get read-only access (S,O)

- If the permission matrix entry for S,O contains read-only and the static security level of S dominates O and the (current level of S dominates the level of O or S is trusted), then the current access list is updated to include (S,O,read-only)

Formal Security Models

CS177 2013 13

Formal Security Models

CS177 2013 14

Rule 2 Get append access (S,O)

- If the permission matrix entry for S,O contains append and the (current level of S is dominated by the level of O or S is trusted), then the current access list is updated to include (S,O,append)

Rule 2 Get append access (S,O)

- If the permission matrix entry for S,O contains append and the (current level of S is dominated by the level of O or S is trusted), then the current access list is updated to include (S,O,append)

Rule 3 Get execute access (S,O)

- If the permission matrix entry for S,O contains execute, then the current access list is updated to include (S,O,execute)

Formal Security Models

CS177 2013 15

Formal Security Models

CS177 2013 16

Rule 4 Get read-write access (S,O)

- If the permission matrix entry for S,O contains read-write, the static security level of S dominates the level of O, and (the current level of S is equal to the level of O or S is trusted), then the current access list is updated to include (S,O,read-write)

Rule 5 Release access (S,O,A)

the triple (S,O,A) is removed from the current access list of subject S

Tranquility

Principle of strong tranquility – security levels do not change during the lifetime of the system

Principle of weak tranquility – security levels do not change in a way that violates the rules of a given security policy

Formal Security Models

CS177 2013 17

Formal Security Models

CS177 2013 18

Rule 10 Change subject current security level (S,L)

- if the requested level is dominated by the subject’s level and the (subject is trusted or the requested level would not violate the star property or SS property) then the subject’s current level becomes the requested level

weak tranquility

Rule 10 Change subject current security level (S,L)

- If the requested level is dominated by the subject’s level and the (subject is trusted or for all objects
 - if the current access list contains the triple (S,O, append) then the level of O dominates L
 - if the current access list contains the triple (S,O, read-write) then L equals the level of O
 - if the current access list contains the triple (S,O, read) then L dominates the level of O)
 then the subject’s current level becomes the requested level

The model also includes a tree-structured hierarchy

- This adds another invariant

Hierarchy compatibility

- the security levels of objects encountered on any path from the root node outward must be monotonically non decreasing

Rule 8 Create object (S,O1,O,L)

- if the subject currently has append or read-write access to object O, and the requested level L dominates the level of O, then the new object O1 is inserted in the hierarchy below O and with the requested security level

Rule 9 Delete object group (S,O)

- if the indicated object O is not the root and the subject S has write access to the parent, then object O and its children are removed from the hierarchy, all current accesses to object O and its children are removed, and permission matrix entries for object O and its children are set to empty

Rule 6 Give access (S1,S2,O, A)

- if subject S1 currently has write access to object O’s parent, then the permission matrix entry for subject S2 and object O will be updated to include access A

The model also has the following constraint

Rule 7 Rescind access (S1,S2,O, A)

- if subject S1 currently has write access to object O's parent, then the permission matrix entry for subject S2 and object O is updated to remove access A and if S2 currently has access A for object O it is removed from the current access list

Tranquility principle

- the static security level of a subject may not change and if the security level of an object changes then the object was inactive

System Z

Rule 11 Change object security level (S,O,L)

Reasoning About Security Models

- John McLean
IEEE Computer, January 1990

- Questioned the validity of the Bell & LaPadula model

System Z

- Initial state is secure
- Only one action
 - when a subject requests any access to any object
 - every subject and object is downgraded to the lowest level
 - permission is added to the access matrix
 - the access is recorded in the current access

System Z satisfies the Basic Security Theorem

Yet it is not secure!

Or is it?

McLean's Conclusions

- Proving that states are secure is insufficient to prove the security of a system
- One must consider both states and transitions

Formal Security Models

CS177 2013 31

Integrity Model

Integrity level is based on the sensitivity to modification of information

The higher the integrity level, the more confidence one has that a program will execute correctly

The higher the integrity level of data, the more accurate and/or reliable

Formal Security Models

CS177 2013 32

Integrity Model

Subjects and Objects

Integrity Levels

- C - crucial
- VI - very important
- I - important

Access

- O - observation
- M - modification
- E - execute
- I - invocation

Formal Security Models

CS177 2013 33

Non-Discretionary Integrity Policies

- Low-water mark policy for subjects
- Low-water mark policy for objects
- Ring policy
- Strict integrity policy

Discretionary Integrity Policies

- Access control lists
- Object hierarchy
- Rings

Formal Security Models

CS177 2013 34

Low-water mark policy for subjects

- After each observation, the integrity level of a subject is decreased to the minimum of its integrity level and the level of the object it observes
- A subject can receive modify access only to objects at an equal or lower integrity level and invoke access only to subjects at an equal or lower integrity level

Formal Security Models

CS177 2013 35

Low water mark policy for objects

- When an object is modified by a subject at a lower integrity level the object's integrity level is decreased to that of the subject

Formal Security Models

CS177 2013 36

Ring policy

- Integrity levels of both subjects and objects are fixed
- A subject may observe an object at any level
- A subject may modify an object or invoke a subject at an equal or lesser integrity level

Strict integrity policy

- Simple integrity condition
- Integrity *-property
- Invocation property

Discretionary Integrity

- Access control lists
- Object hierarchy
- Rings

Access Control Lists

- If a subject can modify an object then it can modify the object's ACL

Object Hierarchy

- To access any object a subject must have observe access for all of the object's ancestors

Rings

- Lower numbered rings have higher privilege
 - Subjects may invoke any subjects of equal or lower privilege and subjects of higher privilege within a range
 - Subjects may observe objects in an allowed range
 - Subjects may modify objects in an allowed range