Access control models

October 27, 2022

Overview

- Discretionary access control
- Multi-level security
  - Bell-LaPadulla model
  - Biba model
- Multi-lateral model
- Domain-type enforcement
- Role based access model
  - Compartment model
  - Chinese wall model
- Clark-Wilson model

Example of useless corporate information policy

1. This policy is approved by Management.
2. All staff shall obey this security policy.
3. Data shall be available only to those with a "need-to-know".
4. All breaches of this policy shall be reported at once to Security.

Security model

Security model provides a formal representation of the access control security policy and its working. The formalization allows the proof of properties on the security provided by the access control system being designed.

Can we prove that system is secure?

Secure system is a system that starts in an authorized state and cannot enter an unauthorized state.

Security state is a subset of state of the system that is related to the security.

State transition occurs when a command changes the state of the system.

Access control matrix is a method to precisely describe security state.

Access control matrix

	objects			
subjects	file a	file b	process a	process b
process a	rwo	rw	rwx	rwx
process b	r	rw	r	rw
alice	—	rwo	rwx	rx
bob	rwo	—	rwx	rx

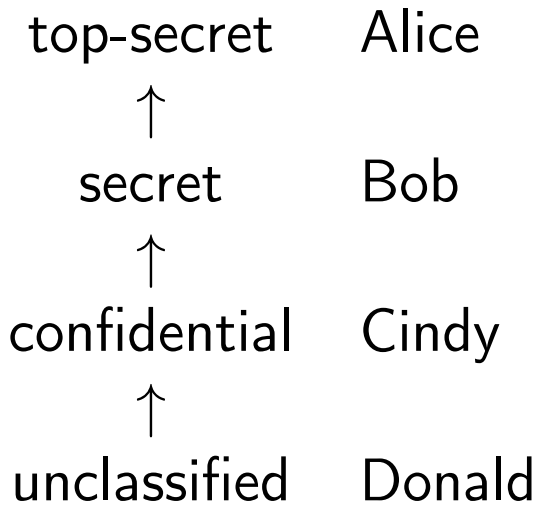
Extensions to access control matrix

1. groups and group hierarchies
2. setuid / impersonification
3. temporal access
4. open / closed default policies

Plan

Mandatory access control

Bell–LaPadulla model for confidentiality



Simple Security Condition: S can read O if and only if $I(O) \leq I(S)$ and S has discretionary read access to O.

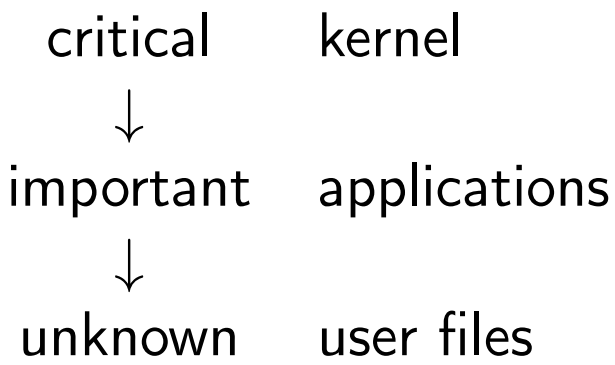
$\star$*-Property:* S can write O if and only if $I(S) \leq I(O)$ and S has discretionary write access to O.

Tranquility properties

The strong tranquility property says that Subjects and objects do not change labels during the lifetime of the system.

The weak tranquility property says that Subjects and objects do not change labels in a way that violates "spirit" of the security policy.

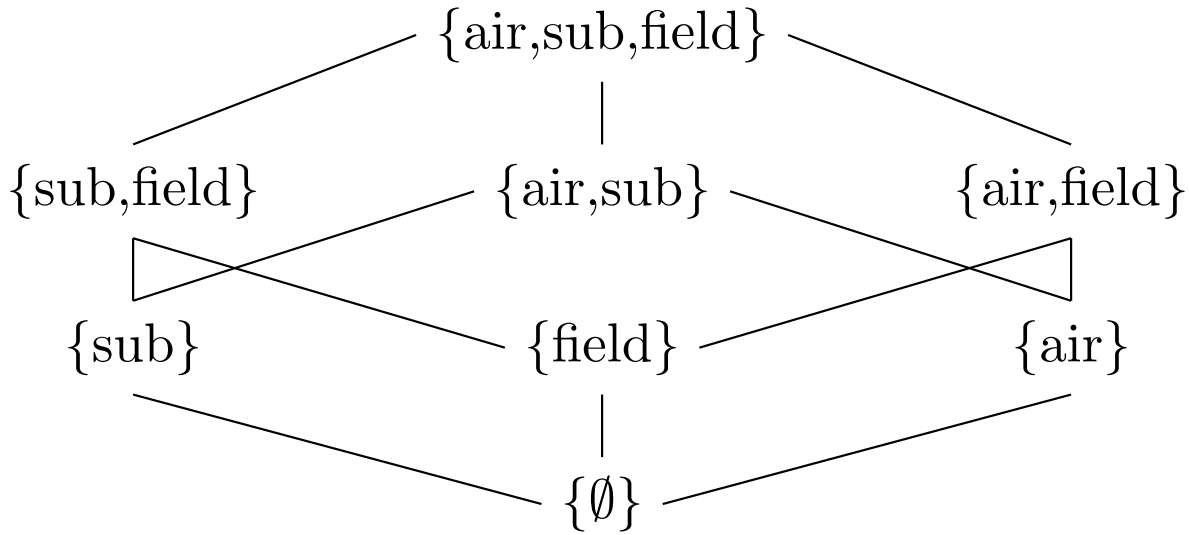
Biba model for integrity



1. $s \in S$ can read $o \in O$ if and only if $i(s) \leq i(o)$.
2. $s \in S$ can write to $o \in O$ if and only if $i(o) \leq i(s)$.
3. $s_1 \in S$ can execute $s_2 \in S$ if and only if $i(s_2) \leq i(s_1)$.

Biba model is a dual to Bell-LaPadulla model.

Compartmentation and the Lattice model



Compartmentation and the Lattice model

Expand the model by adding *categories*. Each object is assigned to multiple categories on "need to know" principle.

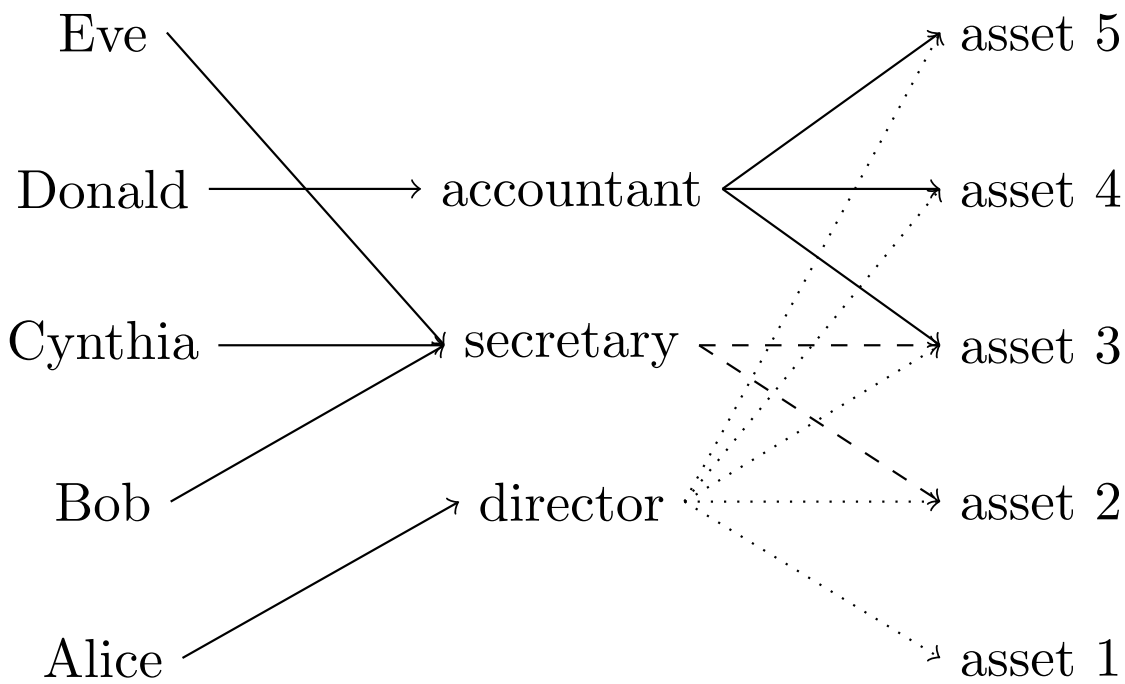
The security level (L, C) *dominates* the security level (L', C') if and only if $L' \leq L$ and $C' \subseteq C$.

Simple Security Condition: S can read O if and only if S dominates O and S has discretionary read access to O.

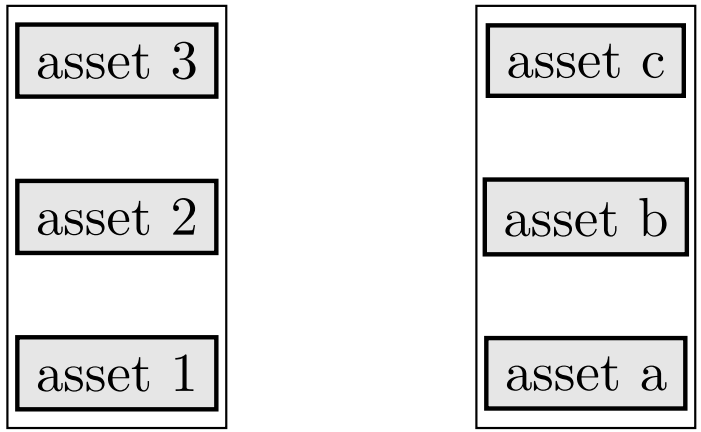
$\star$ *property:* S can write O if and only if O dominates S and S has discretionary write access to O.

Let Σ be a system with a secure initial state σ_0, and let T be a set of state transformations. If every element of T preserves the simple security condition and the $\star$-property, then every σ_i, $i \geq 0$, is secure.

Role-based access control



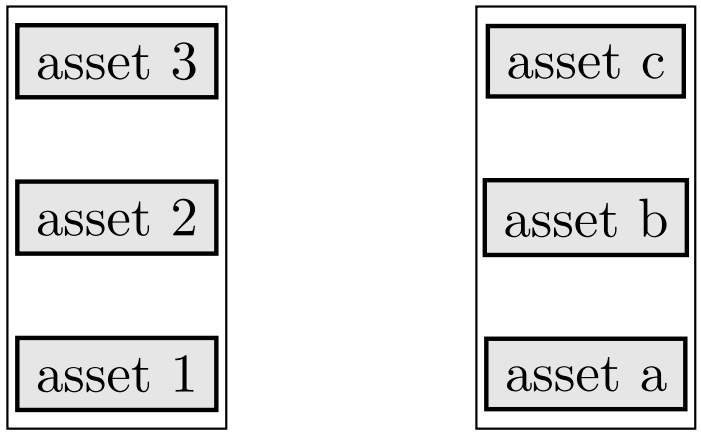
Dynamic policy access control — Chinese wall



Simple security rule: A subject s can be granted access to an object o only if the object o

1. is in the same company datasets as the objects already accessed by s, that is, "within the Wall"
2. belongs to an entirely different conflict of interest class

Dynamic policy access control — Chinese wall



$\star$-*property:* Write access is only permitted if

1. access is permitted by the simple security rule, and
2. no object can be read which i) is in a different company dataset than the one for which write access is requested, and ii) contains unsanitized information

Clark&Wilson model for integrity

Authentication All users has to authenticate before using the system.

Audit All changes of data are logged such that they can be undone.

Well-formed transactions All data manipulations must lead from consistent to consistent state.

Separation of duty The allows each user to run only those programs that reflect her working duty.

Constrained Data Items CDIs are the objects whose integrity must be ensured.

Unconstrained Data Items UDIs are objects that are not covered by the integrity policy.

Integrity Verification Procedures IVPs are verifies that CDIs conforms integrity policy.

Transformation Procedures TPs are the only procedures (well-formed procedures) that are allowed to modify CDIs or to take arbitrary user input and create new CDIs. TPs are designed to take the system from one valid state to the next.

C1 All IVPs must ensure that all CDIs are in a valid state when the IVP is run.

C2 All TPs must be certified to be valid (i.e., preserve validity of CDIs' state)

C3 Assignment of TPs to users must satisfy separation of duty

C4 The operations of TPs must be logged

C5 TPs execute on UDIs must result in valid CDIs

E1 Only certified TPs can manipulate CDIs

E2 Users must only access CDIs by means of TPs for which they are authorized

E3 The identity of each user attempting to execute a TP must be authenticated

E4 Only the agent permitted to certify entities (TP) can change the list of such entities associated with other entities (users).