CCM 4350

Lecture 14

Security Models 2: Biba, Chinese Wall, Clark Wilson
Introduction

- Bell-LaPadula model designed to capture a specific ‘military’ security policy.
- At one time treated as ‘the model of security’.
- However, security requirements dependent on the application; many applications do not need multi-level security.
- We will now look at models for ‘commercial’ integrity policies.
- We will also examine some theoretical foundations of access control.
Agenda

- Biba model
- Chinese Wall model
- Clark Wilson Model
- Information flow models
Recap: Bell-LaPadula Model

- The Bell-LaPadula model is one of the first models that was created to control access to data.
- The properties of the Bell-LaPadula model are:
  - The simple security property (ss) which is “no read up”
  - The star property (*) which is “no write down”
- A problem with this model is it does not deal with integrity of data: it is possible for a lower level subject to write to a higher classified object
Biba Integrity Model

- The Biba integrity model was published in 1977 at the Mitre Corporation, one year after the Bell-LaPadula model was published 1976.
- The primary motivation for creating this model is the inability of the Bell-LaPadula model to deal with integrity of data.
- The Biba model addresses the problem with the star property of the Bell-LaPadula model, which does not restrict a subject from writing to a more trusted object.
Biba Model

- Integrity policies prohibit the corruption of ‘clean’ high level entities by ‘dirty’ low level entities.
  - Clean and dirty shorthand for high integrity and low integrity.
  - Concrete meaning of integrity levels is application dependent.

- Subjects and objects labelled with elements from a lattice \((L, \leq)\) of integrity levels by functions \(f_S: S \rightarrow L\) and \(f_O: O \rightarrow L\).

- Information may only flow downwards in the integrity lattice; only information flows caused directly by access operations considered.

- Biba model: state machine model similar to BLP; no single high-level integrity policy.
Integrity Levels

- Integrity levels are defined by labels, consisting of two parts:
  - a classification
  - a set of categories

- Integrity levels are given to the subjects and objects in the system

- Integrity labels tell the degree of confidence that may be placed in the data
Subjects and Objects (Recap BLP)

- Like other models, the Biba model supports the access control of both subjects and objects.
  - **Subjects** are the active elements in the system that can access information (processes acting on behalf of the users).
  - **Objects** are the passive system elements for which access can be requested (files, programs, etc.)
- Each subject and object in the Biba model will have a integrity level associated with it
Access Modes

- The Biba model consists of the following access modes:
  - **Modify**: the modify right allows a subject to write to an object. This mode is similar to the write mode in other models.
  - **Observe**: the observe right allows a subject to read an object. This command is synonyms with the read command of most other models.
  - **Invoke**: the invoke right allows a subject to communicate with another subject.
  - **Execute**: the execute right allows a subject to execute an object. The command essentially allows a subject to execute a program which is the object.
Biba With Static Integrity Levels

- **Simple Integrity Property (no write-up):** If subject $s$ can modify (alter) object $o$, then $f_S(s) \geq f_O(o)$.

- **Integrity ∗-Property:** If subject $s$ can read (observe) object $o$, then $s$ can have write access to some other object $o'$ only if $f_O(o) \geq f_O(o')$.

- **Invoke Property:** A ‘dirty’ subject $s_1$ must not touch a ‘clean’ object indirectly by invoking $s_2$: Subject $s_1$ can invoke subject $s_2$ only if $f_S(s_1) \geq f_S(s_2)$.
Simple Integrity Condition

• No Read Down
Integrity * (Star) Property

- No Write-Up
Biba: Dynamic Integrity Levels

- **Low watermark policies** automatically adjust levels (as in the Chinese Wall model):

  - **Subject Low Watermark Policy**: Subject $s$ can read (observe) an object $o$ at any integrity level. The new integrity level of $s$ is $\text{g.l.b.}(f_S(s), f_O(o))$.

  - **Object Low Watermark Policy**: Subject $s$ can modify (alter) an object $o$ at any integrity level. The new integrity level of $o$ is $\text{g.l.b.}(f_S(s), f_O(o))$. 
Subject Low Watermark Principle

(Before)  (After)
Object Low-Watermark Principle

(Before) Write (After)
Biba for Protection Rings

- **Ring Property**: A ‘dirty’ subject $s_1$ may invoke a ‘clean’ tool $s_2$ to touch a ‘clean’ object:
  
  Subject $s_1$ can read objects at all integrity levels, modify objects $o$ with $f_S(s_1) \geq f_O(o)$, and invoke a subject $s_2$ only if $f_S(s_1) \leq f_S(s_2)$.

- The ring property is the opposite of the invoke property!
- Captures integrity protection in operating systems based on protection rings.
Chinese Wall Model

- In financial institutions analysts deal with a number of clients and have to avoid **conflicts of interest**.

- Components:
  - **subjects**: analysts
  - **objects**: data item for a single client
  - **company datasets**: \( y: O \rightarrow C \) gives for each object its company dataset
  - **conflict of interest classes**: companies that are competitors; \( x: O \rightarrow P(C) \) gives for each object \( o \) the companies with a conflict of interest on \( o \)
  - **labels**: company dataset + conflict of interest class
  - **sanitized information**: no access restrictions
Chinese Wall Model – Policies

- **Simple Security Property**: Access is only granted if the object requested
  - is in the same company dataset as an object already accessed by that subject;
  - does not belong to any of the conflict of interest classes of objects already accessed by that subject.

- Formally:
  - \( N = (N_{so})_{s \in S, o \in O} \), Boolean matrix, \( N_{so} = \text{true} \) if \( s \) has accessed \( o \);
  - ss-property: subject \( s \) gets access to object \( o \) only if for all objects \( o' \) with \( N_{so'} = \text{true} \), \( y(o) = y(o') \) or \( y(o) \not\in x(o') \).
Chinese Wall: ∗ - Property

- Indirect information flow: \(A\) and \(B\) are competitors having accounts with the same \(Bank\).
- \(Analyst_A\), dealing with \(A\) and the \(Bank\), updates the \(Bank\) portfolio with sensitive information about \(A\).
- \(Analyst_B\), dealing with \(B\) and the \(Bank\), now has access to information about a competitor.
Chinese Wall: ∗ - Property

- **∗ - Property**: A subject \( s \) is permitted write access to an object only if \( s \) has no read access to any object \( o' \), which is in a different company dataset and is unsanitized.
  - subject \( s \) gets write access to object \( o \) only if \( s \) has no read access to an object \( o' \) with \( y(o) \neq y(o') \) or \( x(o') \neq \{\} \)

- Access rights of subjects change dynamically with every access operation.
Chinese Wall: * - Property

www.gammassl.co.uk/topics

blocked by *-property
Clark-Wilson Model

- Addresses security requirements of commercial applications. ‘Military’ and ‘commercial’ are shorthand for different ways of using computers.
- Emphasis on integrity
  - internal consistency: properties of the internal state of a system
  - external consistency: relation of the internal state of a system to the outside world.
Clark-Wilson: Access Control

- Subjects & objects are ‘labeled’ with programs.
- Programs serve as intermediate layer between subjects and objects.
- Access control:
  - define access operations (transformation procedures) that can be performed on each data item (data types).
  - define the access operations that can be performed by subjects (roles).
- Note the difference between a general purpose operating system (BLP) and an application oriented IT system (Clark-Wilson).
Access Control in CW

user

TP

append must be validated

Log CDI

CDIa

CDIb

UDI

authentication authorization

integrity checks, permissions checked
CW: Certification Rules

Five certification rules suggest how one should check that the security policy is consistent with the application requirements.

- **CR1**: IVPs (initial verification procedures) must ensure that all CDIs (constrained data items) are in a valid state when the IVP is run.
- **CR2**: TPs (transformation procedures) must be certified to be valid, i.e. valid CDIs must always be transformed into valid CDIs. Each TP is certified to access a specific set of CDIs.
- **CR3**: Access rules must satisfy any separation of duties requirements.
- **CR4**: All TPs must write to an append-only log.
- **CR5**: Any TP that takes an UDI (unconstrained data item) as input must either convert the UDI into a CDI or reject the UDI and perform no transformation at all.
Describe mechanisms within the computer system that should enforce the security policy:

- **ER1**: For each TP maintain and protect the list of entries \((CDI_{a}, CDI_{b}, \ldots)\) giving the CDIs the TP is certified to access.
- **ER2**: For each user maintain and protect the list of entries \((TP_{1}, TP_{2}, \ldots)\) specifying the TPs user can execute.
- **ER3**: The system must authenticate each user requesting to execute a TP.
- **ER4**: Only subjects that may certify an access rule for a TP may modify the respective list; this subject must not have execute rights on that TP.
Information Flow Models

- Similar framework as BLP: objects are labeled with security classes (form a lattice), information may flow upwards only.
- Information flow described in terms of conditional entropy (equivocation → information theory)
- Information flows from \( x \) to \( y \) if we learn something about \( x \) by observing \( y \):
  - explicit information flow: \( y := x \)
  - implicit information flow: IF \( x = 0 \) THEN \( y := 1 \)
  - covert channels
- Proving security is undecidable.
Non-interference Models

- A group of users, using a certain set of commands, is *non-interfering* with another group of users if what the first group does with those commands has no effect on what the second group of users can see.

- Take a state machine where low users only see outputs relating to their own inputs. High users are non-interfering with low users if the low users see the same no matter whether the high users had been providing inputs or not.

- Active research area in formal methods.
The more expressive a security model is, both with respect to the security properties and the systems it can describe, the more difficult it is usually to verify security properties.
Summary

- The theoretical foundations for access control are relevant in practice.
- It helps to know in which complexity class your policy language and enforcement algorithm put you in.
- Powerful description languages may leave you with undecidable enforcement problems.
- Much of current efforts on policy languages in ‘trust management’ and web services access control revolves around these issues.
Further Reading

- ESORICS 2000 (Springer Lecture Notes in Computer Science 1895): Checking secure interactions of Smart Card Applets and Verification of a Formal Security Model for Multiapplicative Smart Cards