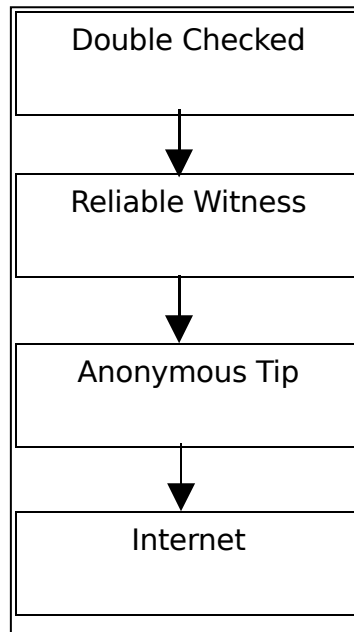


Notes: 9th Feb, 2009

BIBA Integrity Model, ACMs, ACLs

In last class we discussed Bell-LaPadula Model and it only addresses 'Data confidentiality'. In this class BIBA Integrity Model was discussed, which addresses 'Data Integrity'.

BIBA Integrity Model:-



- 1) This 'Integrity Model' contains 4 compartments viz., Double Checked, Reliable Witness, Anonymous Tip, Internet.
- 2) Double Checked contains the 'most reliable data'.
- 3) Giving the rules of 'Read' and 'Write':

Consider the following pairs (Lo, Co) and (Ls, Cs):

Lo – Label of object

Ls – Label of Subject

Co – Compartment of object

Cs - Compartment of Subject

Read access:

Rules:

- i) The 'Label of object' should be greater than or equal to the 'Label of Subject'.
- ii) The 'Compartment of object' should be superset of the 'Compartment of Subject'.

The rules are represented as:

- i) $L_o \geq L_s$
- ii) C_o is superset of C_s

Write access:

Rules:

- i) The 'Label of Object' should be less than or equal to the 'Label of Subject'.
- ii) The 'Compartment of object' should be subset of the 'Compartment of Subject'.

The rules are represented as:

- i) $L_o \leq L_s$
- ii) C_o is subset of C_s

Role Based Access Control:

It can be explained as follows:

- 1) Consider an example of an 'Enterprise'. An Enterprise may have thousands of employees.
- 2) Since there are large number of employees, there will be hundred or dozens of roles.
- 3) In such system we can represent the 'Access Control Matrix' in terms of roles.
- 4) E.g., Auditor, Teller, Branch, Manager, CEO, CSR.

- 5) The roles can be written as follows: Auditor (is subset of) Teller (is subset of) Branch Manager (is subset of) CEO (is subset of) CSR i.e., Teller should be able to perform the tasks of Auditor too.
- 6) The duties should be statically separated or dynamically separated.
- 7) Problem with static separation is: The tasks are separated initially itself. Hence, there can be a conflict. E.g., Auditor's tasks conflicting with CEO's tasks.
- 8) Dynamically separation of tasks involves switching between roles.

Chinese wall Model:

It can be explained as follows:

- 1) Objects in the system are partitioned into domains: D1, D2...Dn.
- 2) Initially the user can access any domain.
- 3) As soon as user accesses some domain Di then:
 - i) The user loses access to all other conflicting domains.

Example:

This model can be explained with the example of 'Untrusted Code':

It says that as soon as an applet reads Hard-drive it loses Network Write Access permanently.

The ways of communication between processes are:

- i) Shared Memory
- ii) Inter Process communication
- iii) Input
- iv) Fans

Once the process reads the Hard-drive then it won't be able to communicate with other processes through the above mentioned communication ways.

The illegal means through which the processes can communicate is **Covert channels**.

Covert Channels can be defined as 'leaking information' between processes. Examples: CPU usage, Memory, Temperature, etc.

Implementation of Access Control Models:

- 1) Consider the example of UNIX operating System. In Unix OS, the metadata of the file contains information like access controls and policies.

Metadata:

The total number of bits for the access permission is 41bits. These 41bits are classified as:

Owner – 16bits

Group – 16bits

Others – 9bits

The access permissions are Read, Write, and Execute.

The access permissions can be granted to Owner, Group, and Others.

The Metadata is stored in a file.

Access Control Lists:

- 1) ACLs are stored with the object.
- 2) They indicate who can access the object and how it can be accessed.
- 3) Ex:
 - i) GET ACL: It is a Linux Command to get file access control lists.
 - ii) SET ACL: It is a Linux Command to set file access control lists.
- 4) For ACLs we can have Positive and Negative Rules.
- 5) Positive Rule: If we have to give an access to a person a positive entry is made. It is simple to understand.
- 6) Negative Rule: If we have to deny the access then a negative entry is made. Ex: Windows NTFS uses Negative rules.