

**Introduction to Operating Systems**  
**Prof. Chester Rebeiro**  
**Department of Computer Science and Engineering**  
**Indian Institute of Technology, Madras**

**Week - 08**  
**Lecture - 36**  
**Information Flow Policies**

Hello. In the previous video we had looked about Access Control Techniques, and in particular we had looked into the DAC or the Discretionary Access Control. So, we have seen that DAC is very efficient in hack up through giving certain access rights to a particular subject, with respect to set of objects. However, what we seen was the drawback of DAC was that it was incapable of preventing information flow.

So, in this video, what we are going to see is information flow techniques and the MAC that is the Mandatory Access Control.

(Refer Slide Time: 01:03)

## Information Flow Policies

- Every object in the system assigned to a security class (SC)

Security classes (SC)

high

A ● ● ●

B ● ● ●

C ● ● ●

low

↑ Information flow

→ object

– Information flow triple :

$\{SC, \rightarrow, \oplus\}$

$\rightarrow$  is the can flow relation

- $B \rightarrow A$  : Information from  $B$  can flow to  $A$
- $C \rightarrow B \rightarrow A$  : Information flow
- $C \leq B \leq A$  : Dominance relation

$\oplus$  is the join relation

- defines how to label information obtained by combining information from two classes
- $\oplus : SC \times SC \rightarrow SC$ .

$SC, \rightarrow,$  and  $\oplus$  are fixed and do not change with time.  
The  $SC$  of an object may vary with time

Ravi Sandhu, *Lattice Based Access Control Models*, 199326

In information flow policies, every object in the system is assigned to us particular security class. As such the system is divided into fixed number of security classes and each security class is then given a particular category.

For example you have security class which is high, security class which is low, and so on. Now each object in the system is assigned to one of the security class. Similarly each subject in the system that is the entities which actually operate or access these objects they also are given a particular security class. Next we will define how information flows between the classes.

Note that here about information flow, we are not concerned about how information about one object flows to another object or flows to a subject. But rather a concern is more about classes. So, what we are going to see is that how information from one class, flows to another class.

So, formally this is represented by this following triple. So, this triple is a 3 triple it contains the SC that is a security class. It has this arrow operator which shows the flow relation, and it has the join operator which is the join relation. So, an operator like this B arrow to A implies that information from B can flow to A similarly in this example C arrow B to arrow a shows that information from C flows to B, and information from B flows to A. In other words information from C can also flow to A. Now based on this there is another technique of representing, which is by this particular symbol the greater than or equal to symbol. So, what it means is that A dominates B then B dominates C.

Now, let us look at the join relation. So, essentially the join relation is used to determine how to label information after information from 2 classes is combined. So, for instance, let us say we are creating a new object see a new file, and this file is for instance created by concatenating and object present in the security class C and an object in security class B.

Now, the question that arises is to which class should the, new file that we have just created be present in, should it be in security class C or security class B or in the A security class. So, in order to formally write this the join operator is used. So, join is defined as a function between 2 classes. So, it takes the security class and another security class and it will actually give us what is the resulting security class.

So, we will see understand more of for this with an example. Now what we will see is that for the system, the security classes are fixed. So, during the design time itself for

instance we would say that our system has for instant just the security class A, B and C also the flow operations is fixed at the design time itself. So, we can design various scenarios saying that information can flow to C and from C to B and from B to A and so on.

This is also a design time construct or also the join operation is fixed at design time. So, what changes over time is the position of the objects, for instance it may have object which is present in the security class A, and after a while this object becomes for instance less important or it becomes public domain and it can be moved to the security class C. To take an example from real life, what we see is that some top secret documents would be initially categorized as highly secure documents, but over a period of time this becomes public information and therefore, that particular document could then go to a lower security class.

So, you see that while the security classes are fixed as well as the flow of information is fixed among the security classes, the objects as such could move between security classes over a period of time.

(Refer Slide Time: 06:35)

**Examples**

- Trivial case (also the most secure)
  - No information flow between classes

- $SC = \{A_1(\text{low}), A_2, \dots, A_n(\text{high})\}$
  - $A_i \rightarrow A_i$  (for  $i = 1 \dots n$ )
  - $A_i \oplus A_i = A_i$
- Low to High flows only

- $SC = \{A_1(\text{low}), A_2, \dots, A_n(\text{high})\}$
- $A_j \rightarrow A_i$  only if  $j \leq i$  (for  $i, j = 1 \dots n$ )
- $A_i \oplus A_j = A_i$

27

Let us take some examples about information flow let us start with a very trivial case, which also is the most secure example for information flow, essentially because it does not allow information flow between classes. So, let us take this example of a system where the security classes are defined by this particular set  $S$   $C$  and it has  $n$  security classes, out of these the security class  $A_1$  is the lowest while security class  $A_n$  is the highest, then we need to define the flow operator in this case it is  $A_i$  flows to  $A_i$ .

In other words, what it means for every value from 1 to  $n$  information can only flow within the class. In other words, it is not possible for information to flow from one class to another class. The third requirement is to define the join operator. So, in this particular case it is quite trivial to know that if you combine particular information from one class, with another information from the same class we will result in a new object which also belongs to the same class. Now let us look at a less stringent example, which is less secure than the previous case which allows information only to flow from the low to high and not anywhere else.

The security classes are defined as in the previous case. So, we had  $n$  security classes and  $A_1$  is a lowest  $A_n$ , is the highest and information can only flow from  $A_j$  to  $A_i$  where  $j$  is less than equal to  $i$ . So, what this means is that information can flow only from lower class to a higher class, but the opposite direction from a high to low is not possible, also while defining the join operator that is when we take information from a low class and join combine it with information from a higher class, that is  $A_i$  with  $A_j$  we will get some new information which also belongs to the higher class that is the  $A_i$  class.

So, it will repeat that if we take some information from a low class and combine it with information in a higher class for instance  $A_2$  then the new object that we create will also be in the  $A_2$  security class think of what would happen, if instead of  $A_i$  over here we had  $A_j$ . So, I leave that to you to think about.

(Refer Slide Time: 09:34)

## Mandatory Access Control

- Access based on regulations set by a central authority
- Most common form is **multilevel security (MLS)** policy
  - Access Class
    - Objects need a **classification level**
    - Subjects needed a **clearance level**
  - A subject with X clearance can access all objects in X and below X but not vice-versa
  - Information only flows upwards and cannot flow downwards



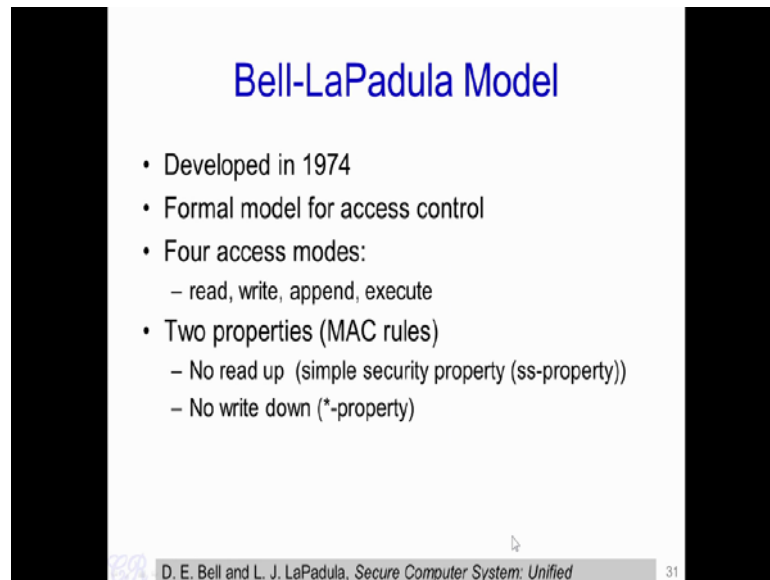
30

Now, let us come to the Mandatory Access Control Mechanism. So, essentially over here the access mechanism or policies are based on regulations which are sent by a centralized authority. So, the most common form is the MLS or Multilevel Security policy essentially in this policy we have several access classes like the unclassified, confidential, secret and top secret. So, every object in the system needs a particular classification level. So, every object could be classified as either one of these 4. Similarly every subject in the system also needs a clearance level. So, that clearance levels are also unclassified confidential secret and top secret.

So, one of these 4, so now, subject with clearance x can only access all objects in x and below x and not vice-versa that is information can only flow upwards, but cannot flow downwards. To take an example suppose we have a particular subject that is a user, who has a clearance level secret, now what it means is that this particular user in access all the secret objects, all the objects which are classified as secret all objects which are classified as confidential as well as unclassified, but this user will not be able to access any top secret objects.

In other words information about a top secret object cannot flow to a user with clearance secret. While the upward direction of information flow is possible, now there are 2 types of MAC control techniques.

(Refer Slide Time: 11:36)



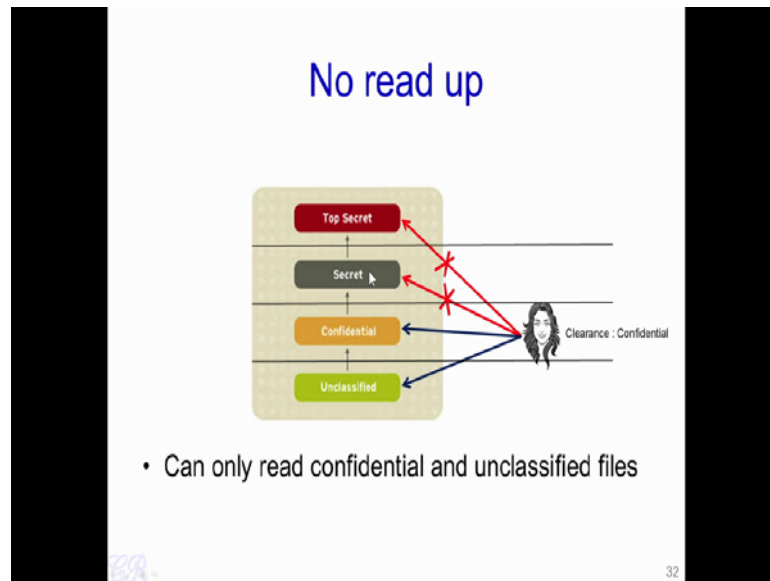
## Bell-LaPadula Model

- Developed in 1974
- Formal model for access control
- Four access modes:
  - read, write, append, execute
- Two properties (MAC rules)
  - No read up (simple security property (ss-property))
  - No write down (\*-property)

D. E. Bell and L. J. LaPadula, *Secure Computer System: Unified* 31

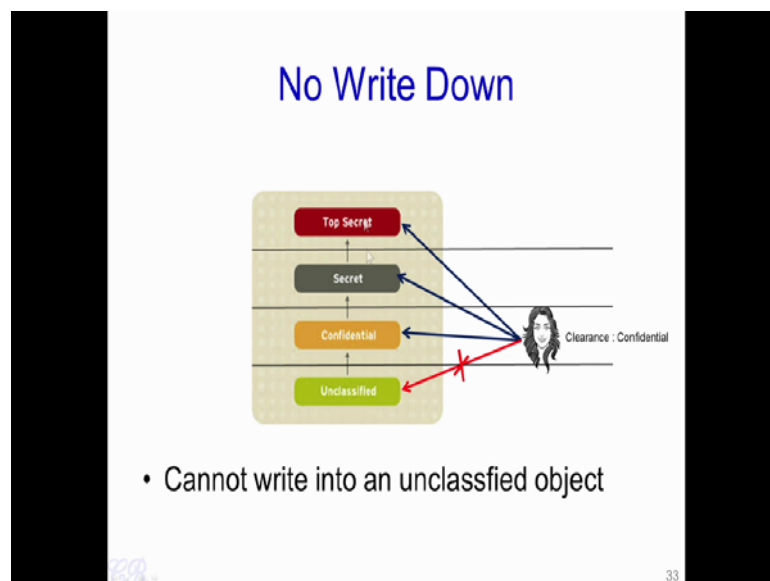
And the first we will see is the Bell LaPadula model. So, this was developed in 1974 and has is a formal model for access control. It gives four access modes read, write, append and execute. And it has 2 MAC properties that is no read up which is also known as the SS property or simple security property, and no write down which is the star property. So, let us look at what these two properties are?

(Refer Slide Time: 12:09)



So, what this means that is, the no read up that is the first property what we seen here, is that if we have a user with clearance confidential then that user can read all the objects which are confidential as well as all the objects which are unclassified, in the sense she cannot read any object which is classified as secret and any object which is classified as top secret. So, this particular mechanism does not allow reader.

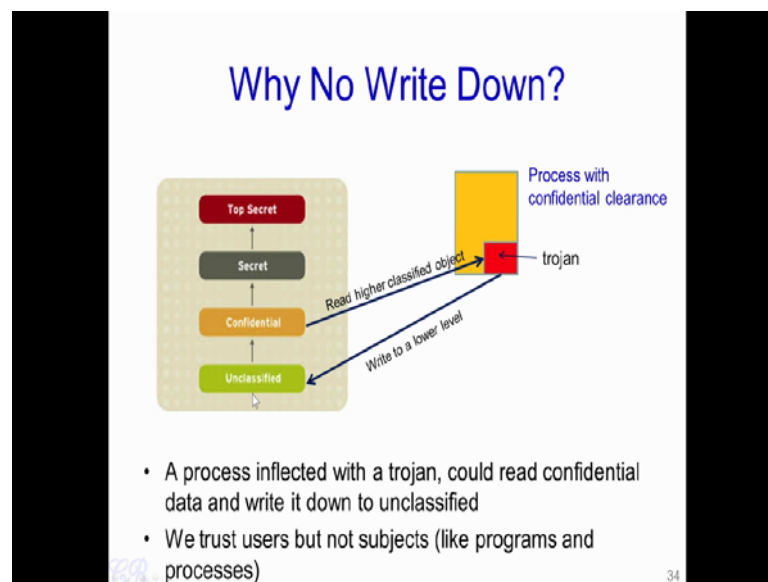
(Refer Slide Time: 12:46)



On the other hand, the second policy states that there is no write down, in other words this particular user with clearance confidential is allowed to write or modify an object which is classified as confidential as well as this user can modify objects of classified as secret as well as top secret. She is able to write upwards, but what is not possible is that she is not allowed to write downwards.

So, she is not allowed to change or modify any object which is unclassified. Now this would seem very strange. So, let us see why such a mechanism was present.

(Refer Slide Time: 13:35)



Essentially let us consider this particular scenario, where we have a process which is confidential process with clearance confidential which is executing therefore, as we know it could read data or read an object which is also classified as confidential, now let us assume that there is a trojan host that is present. So, what we have seen is that when a trojan executes it is going to inherit all the clearance levels of the process.

In this case the trojan is also going to get a clearance confidential. Now assume that write down was allowed, in such a case it will not be very difficult for the trojan to read at particular file or an a particular object which is not confidential and write it down to the unclassified, that is it could be made as a public domain information. So, you see that

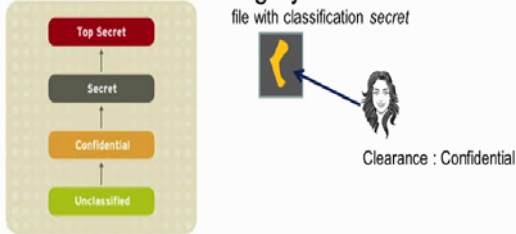


this causes a flow of information from high confidential level to a low confidential level, essentially through the particular Trojan. Therefore, this particular model does not permit write down.

(Refer Slide Time: 14:58)

### Limitations of BLP

- Write up is possible with BLP
- Does not address Integrity Issues



User with clearance can modify a secret document  
BLP only deals with confidentiality. Does not take care of integrity.

35

So, we look the limitations of the Bell-LaPadula model, essentially what the Bell-LaPadula model does not prevent is that it allows a user with clearance confidential to write data upwards. So, this particular user with clearance confidential could essentially modify particular top secret document or a secret document. In other words the BIP model does not address the integrity issues that are present. So, in order to cater to the integrity issues, there was another model which was used.

(Refer Slide Time: 15:46)



## Biba Model

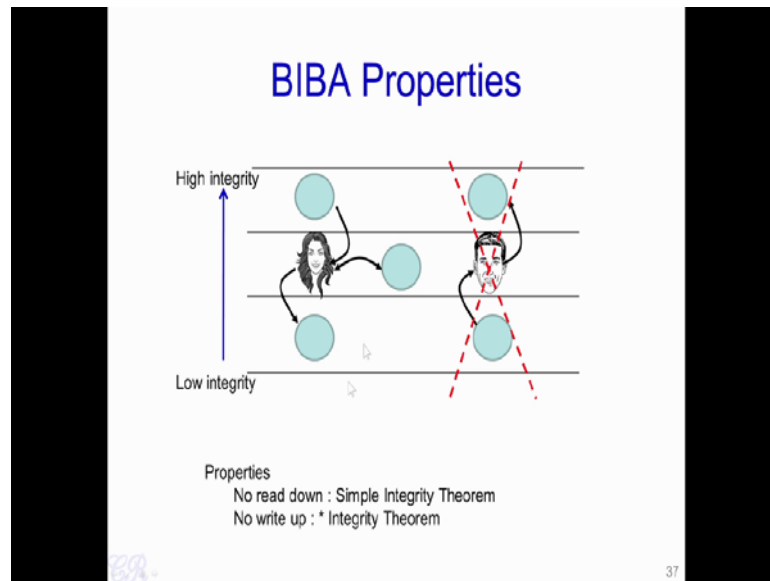
- Bell-LaPadula upside down
- **Ignores confidentiality and only deals with integrity**
- Goals of integrity
  - Prevent unauthorized users from making modifications in a document
  - Prevent authorized users from making improper modifications in a document
- Incorporated in Microsoft Windows Vista

36

So, this is the Biba model. So, Biba model is the Bell-LaPadula model upside down. So, while the Bell-LaPadula model just focuses on confidentiality, ensures that no information flows from high to a low.

The Biba model on the other hand ignores confidentiality all together and deals only with integrity. So, the main goal of the Biba model is to prevent unauthorized users from making modifications to particular document, also it prevents authorized users from making improper modifications in a document. So, this Biba model is incorporated in Microsoft windows vista operating system.

(Refer Slide Time: 16:38)

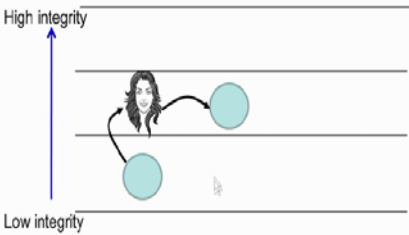


So, what the Biba model defines is that the user can first of all read and write to any object within the same security class, and the user could write to object in a lower security class, and read from an object present in a higher security class. This particular object can be read, this particular object can be modified because it is in a lower security class, this can be read because it is in a higher security class. While objects in the same security class can be read and written, please follow the arrows in this particular case. So, what it does not allow is that the lower security class B read from, thus essentially user cannot read from the lower security class and cannot write to a higher security class.

So, you see this is exactly the opposite or what the Bell-LaPadula model tells us. So, the property of the Biba model is that there is no read down and no write up this is the star integrity theorem. And where the no read down is a simple integrity theorem. So, with respect to the security classes this is the low integrity and this is the high integrity class.

(Refer Slide Time: 18:05)

### Why no Read Down?



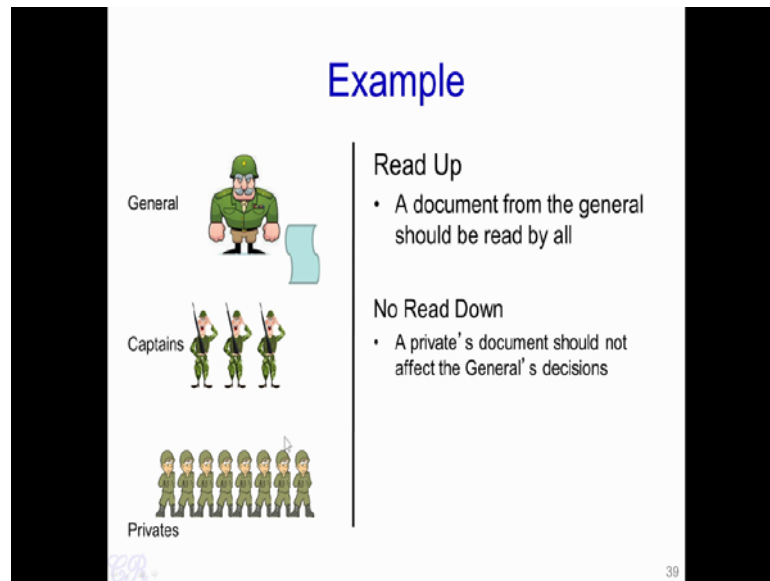
- A higher integrity object may be modified based on a lower integrity document

38

So, why does the Biba model not support read down, why cannot user with particular security class read from an object from a lower security class? So, why is this particular direction for information flow not permitted? The reasoning behind that is that higher integrity object such as this one may be modified based on lower integrity document.

For instance since, this particular user with particular security class is capable of writing to this secure this particular object present in the same security class. Now if she is able to read from a lower security class, and get at there is a flow of information upwards what it means? Is that she can then get influenced by this information or she can then copy this information on to this particular object in the higher security class. So, this means lower integrity object is affecting higher integrity object, and therefore the Biba model prevents such flow of information.

(Refer Slide Time: 19:24)



To take an example, let us say the hierarchy in the military where you have a general right on top; then, the captains and the privates who are right at the bottom of the hierarchy.

Now, the Biba model allows read up meaning document which is prepared by the general should be read by all that is a document which is created by the general, should be read by the captain as well as the privates. However, no read down is permitted, that is document written or modified by the privates at the lower end of the hierarchy should not affect the generals decision.

(Refer Slide Time: 20:08)

**Threats**

- Control flow hacking
  - Example : Buffer overflows
- Covert Channels

...next

40

So, we see that in spite of having such flow mechanisms, where information can be restricted in how they flow that is we can prevent information flow between security classes; however, due to various reasons on due to bugs or other flaws in the design, it may be possible for information to actually flow between security classes, in spite of having such robust measures like the access control measures that we just discussed.

So, what we will do in the next video is that, we will look at such techniques where information can actually flow in spite of having such mechanisms in place. So, we will look at control flow hacking, essentially buffer overflows and how they could be used to allow and unauthorized user gaining information from a system.

Thank you.