Michael E. Locasto

Department of Computer Science
UofC
CPSC 525/625

January 28, 2013
Agenda

Announcements (10 minutes)
Learning Objectives for Today (1 minute)
Three Security Models (45 minutes)
→ I expect the reading load to ease off and spread out.
→ Bug Hunting Project is released. Let’s glance at it quickly.
→ Sketch major properties of Bell-LaPadula and Biba
→ Understand logical relationship between these (how the concepts of confidentiality and integrity interplay)
→ Define the Chinese Wall model
Last session, we looked at the underpinnings of MLS-style security models. We seemed to notice a few shortcomings.
This model deals with integrity and confidentiality in the sense of contamination and conflict: conflict of interest.

What if a subject needs to access an object, but should not thereafter see objects of different classification?
Objects are mapped to conflict classes. Subjects are allowed to access anything in a conflict class $C$.

Event: subject $S$ accesses object $O \in C$.

After this, $S$ cannot access any object $O'$ in $C$ where $O' \neq O$. 
This model primarily concerns itself with the preservation of confidentiality. Subjects and objects are labeled with the category mechanism we saw from last time.

No read-up. (subjects at lower classification levels should not read higher levels)

No write-down. (subjects at high classification levels should not disclose sensitive information)
Simple Security Property / no-read-up
The *-Property / no-write-down
Discretionary Security Property
Simple Security Property / no-read-up: *cannot read data from an object above you in the lattice*. class of subject must dominate or equal class of the object.

The *-Property / no-write-down

Discretionary Security Property
Simple Security Property / no-read-up
The *-Property / no-write-down: *If a subject has read access to Object 1 and write access to Object 2, then the class of Object 2 must dominate or equal the class of Object 1.*
Discretionary Security Property
Simple Security Property / no-read-up
The *-Property / no-write-down
Discretionary Security Property: A subject can only access an object if the ACM table entry for that relationship permits it.
Practical Considerations

How would you write a compiler to implement these rules?

How would you get an OS to enforce such rules?

What kind of hardware do you need to enable the enforcement of these rules efficiently?
In contrast to Bell-LaPadula, Biba focuses on providing a notion of integrity. In this context, the definition of integrity focuses on avoiding contamination of trusted or high-level information by lower-level or untrusted information.

No write-up. (subjects cannot write objects with greater integrity)

No read-down. (subjects cannot read objects with lower integrity)
These readings are the two papers on Bell-LaPadula and Biba. You can also read the textbook section 2.4.