Foundations of Computer Security

Lecture 9: MLS Example: Part IV

Dr. Bill Young
Department of Computer Sciences
University of Texas at Austin

MLS Thought Experiment

We introduced the following rule, which appears to capture our intuition about when a subject can read an object.

The Simple Security Property: Subject S with clearance (L_S, C_S) may be granted *read* access to object O with classification (L_O, C_O) only if $(L_S, C_S) \geq (L_O, C_O)$.

Is it all we need? What about other types of access?

Do We Need Secure Writing?

The Simple Security property codifies restrictions on *read* access to documents. What about *write* access?

Suppose someone with access to a Top Secret document copies the information onto a piece of paper and sticks it into an Unclassified folder.

Has Simple Security been violated? *No!* Has confidentiality been violated? *Clearly.*

Secure Writing

In general, subjects in the world of military documents are *persons* trusted not to write classified information where it can be accessed by unauthorized parties.

Subjects in the world of computing are often *programs* operating on behalf of a trusted user (and with his or her clearance).

Some program I run may have embedded malicious logic (a "trojan horse") that causes it to "leak" information without my knowledge or consent.

The *-Property

We restrict *write* access according to the following rule:

> **The *-Property:** *Subject S with clearance (L_S, C_S) may be granted write access to object O with classification (L_O, C_O) only if $(L_S, C_S) \leq (L_O, C_O)$.*

This is pronounced "the star property." *How does it help?*

The *-Property

Does this rule make sense? Is it too restrictive? Is it too lax?

- Can a commanding general with a Top Secret clearance email marching orders to a foot soldier with no clearance? *No!*
- Can a corporal with no clearance overwrite the war plan? *Nothing in our rules stops it, but that's an integrity problem!*

Simple security and the *-property are sometimes characterized as "read down" and "write up," respectively. Alternatively, they're characterized as "no read up" and "no write down."

Lessons

- Control over read *and* write operations is needed to prevent confidentiality breaches.
- The *-property uses dominates to decide whether a write access should be allowed.
- Controlling write access is especially crucial for computers because the accessing subject may be a *program* executing on behalf of a user. The user has been cleared; the program has not.

Next lecture: Tranquility and BLP