Biba model of Integrity

Simon Foley

January 25, 2016

Biba Model of Integrity

BLP/MLS model is concerned with confidentiality.

Example, secret may read from unclassified; secret may write to topsecret.

Simple Interpretation of Integrity: preventing unauthorized access.

Example, the firefox browser may not modify private files.

Biba Model

- A dual of BLP/MLS model.
- Partially ordered set of integrity levels.
- Subjects and objects bound to integrity levels.
- Implemented by Security Kernel Reference Monitor.

Biba Axioms

Biba
▷ Axioms
LOMAC
MIC

Intuition: high integrity data may not in any way be based on/influenced by low-integrity data.

Simple Integrity. No Write Up. Subject S may write an object O only if the integrity level of the subject dominates the integrity level of the object.

$$W \in M[S,O] \Rightarrow \underline{O} \leq \underline{S}$$

Integrity Confinement. No Read Down. Subject may read object only if the integrity level of the subject is dominated by the integrity level of the object.

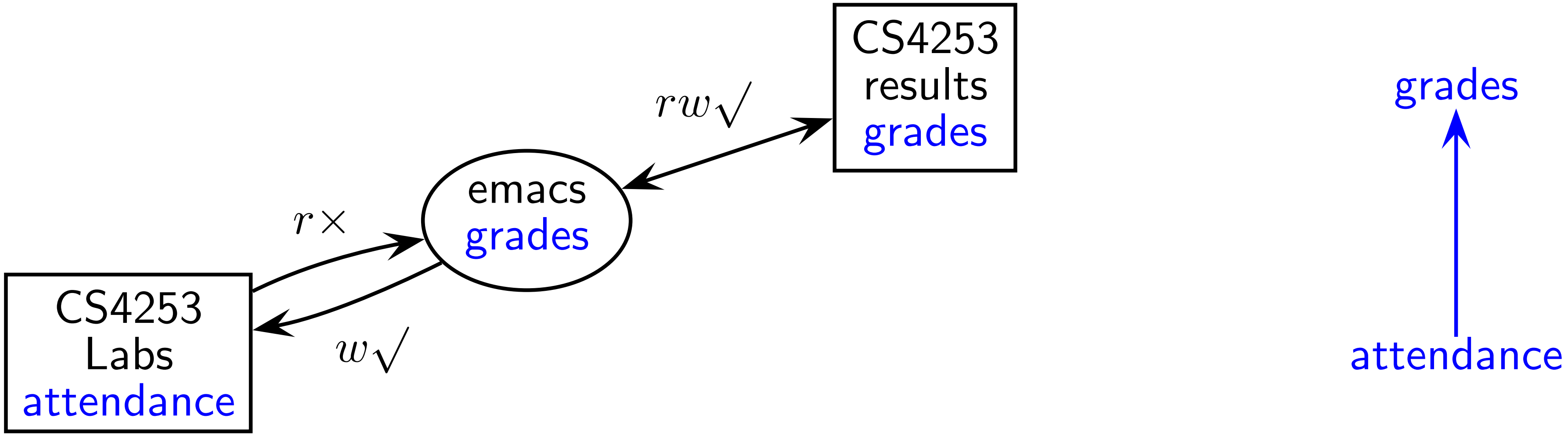
$$R \in M[S,O] \Rightarrow \underline{S} \leq \underline{O}$$

Biba Example: No read down

We have two classes of data:

- □ grades: high integrity data calculated and maintained by Simon.
- □ attendance: low-integrity data maintained by Tutor.

Simon is cleared to grades; Tutor cleared to attedance
Simon uses untrusted emacs editor to process grades.

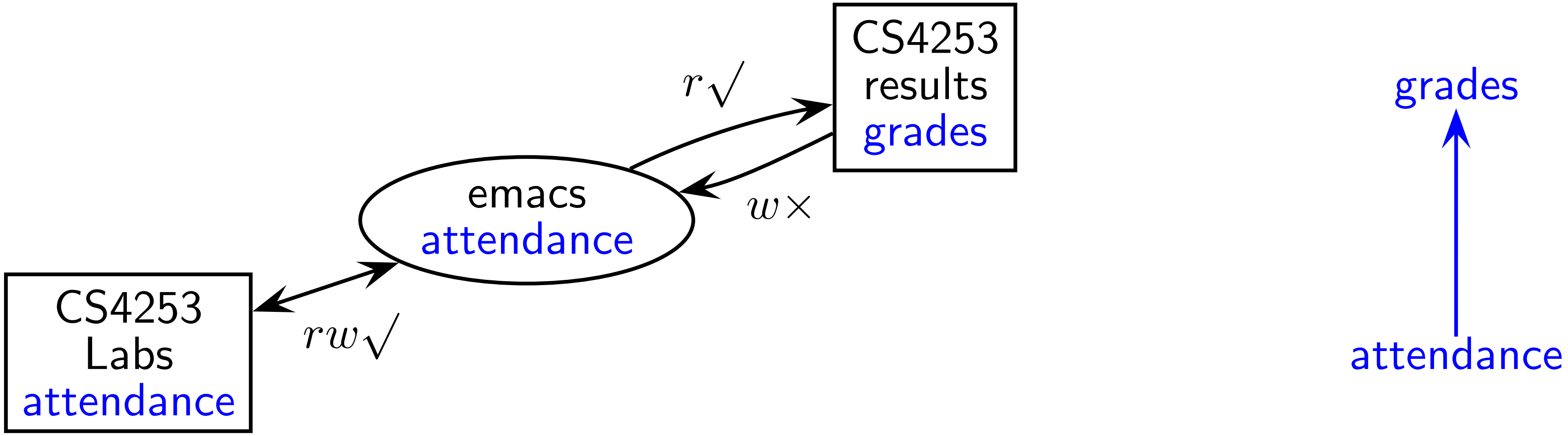


Trojan Horse is not permitted to copy low-integrity data to high integrity: grades may not be ‘polluted’ with low-integrity data.

Biba Example: No write up

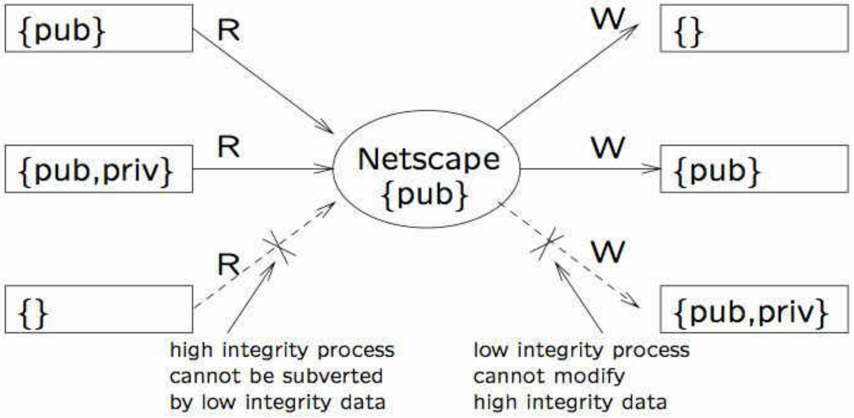
Biba
▷ Axioms
LOMAC
MIC

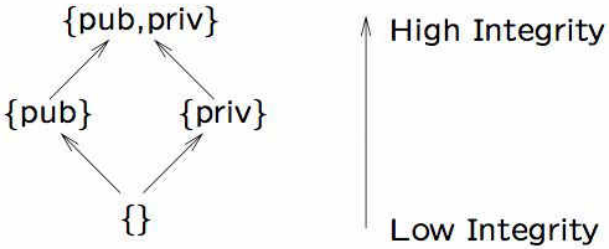
Tutor runs emacs to process lab attendance.



OK for high integrity data to flow down to low integrity, but not vice-versa.

Biba Model of Integrity





Biba Integrity Axioms.

Simple Integrity. No Write Up. Subject may write an object only if the integrity level of the subject dominates the integrity level of the object.

Integrity Confinement. No Read Down. Subject may read object only if the integrity level of the subject is dominated by the integrity level of the object.

Biba and Low-Water-Mark labels

Object levels remain fixed.

Subject levels may change over time. Whenever a subject reads an object with a level lower than its own, the level of the subject is reduced to that of the object (so long as the Biba axioms are not violated).

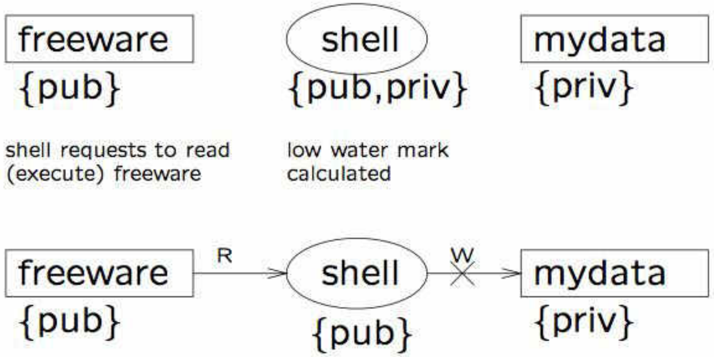
Subject's level is determined by the lowest-leveled object it read in the past (lwm).

Download program contains Trojan horse.

User shell has initial level {pub,priv}.

To execute freeware, security kernel lowers level of shell to {pub}.

Shell may not request write to mydata.



Should the integrity level of a subject be allowed to float upwards?

Can you come up with a comparable mechanism for BLP/MLS?

LOMAC Low Water Mark Mandatory Access Control

Integrity mechanism for freebsd Unix.

Loadable kernel module that implements Biba with low water marks

Supports only a total ordering of integrity levels $1 < 2 < 3 < \cdots$.

A MAC protection mechanism; MAC policy is also applied to 'root' user, so a super user who can do *anything* does not exist.

Objects are anything mapped to an inode: files, pipes, FIFOs, sockets, etc.

Subjects are processes or jobs.

All processes in a process group have same integrity level.

Windows MIC Mandatory Integrity Control

MIC implements a version of the Biba model in Windows (Vista onwards).

For any access, Windows first checks that MAC (MIC/Biba) policy is upheld and if OK, then checks the DAC policy.

Associate integrity level with each process (subject), each file/object has minimum (low water mark) integrity level. (Roughly) 3 integrity levels:

- □ `low`: untrusted access to temporary internet folders and low-privilege sections of current users registry; eg, Internet Explorer runs at `low`.
- □ `medium`: user access to own documents folder and section of registry.
- □ `high`: administrative privilege to install to Program Files folder, write system registry entries, etc.

Usually, child processes inherit integrity level of parent unless executable program running in child process has lower integrity level (for example, Internet Explorer runs at `low`).

Interpreting MIC: User Interface Privilege Isolation

A lower privilege application cannot:

- □ SendMessage or PostMessage to higher privilege application windows (blocks Shatter attacks, whereby one low-authority process can send code to another high-authority process for execution).
- □ Use thread hooks to attach to a higher privilege process
- □ Use journal hooks to monitor a higher privilege process
- □ Perform DLL injection to a higher privilege process