Security Policy Models

- Chapters 5 & 6
- Overview
  - Confidentiality policy
  - Integrity policy
- Bell-LaPadula model
- Biba's model

Confidentiality

- X set of entities as subject, I information as object
- I has *confidentiality* property with respect to X if no $x \in X$ can obtain information from I
- I can be disclosed to others
- Example:
  - X set of students
  - I final exam answer key
  - I is confidential with respect to X if students cannot obtain final exam answer key

Confidentiality Policy

- Military (governmental) security policy
  - Policy primarily protecting confidentiality
  - Comes from the military's need to keep information
  - No disclosure of military and government information
  - Privacy act : constraints on what information a government entity can legally obtain from individuals

Confidentiality Policy

- Goal: prevent the unauthorized disclosure of information
  - Deals with information flow
- Multi-level security models are best-known examples
  - Bell-LaPadula Model basis for many, or most, of these

Integrity

- X set of entities as subject, I information as object
- I has *integrity* property with respect to X if all $x \in X$ trust information in I
- Types of integrity:
  - trust I, its conveyance and protection (data integrity)
  - I information about origin of something or an identity (origin integrity, authentication)
  - I resource: means resource functions as it should (assurance)

Integrity Policy

- Commercial security policy
  – Policy primarily protecting integrity
  – Comes from the need of commercial firms to prevent tampering of data
  – Confidentiality: disclosure of customer information is not the direct concern of a bank.
  – Integrity: the loss resulted from the disclosure is the direct concern of a bank.

Outline

- Overview
  - Confidentiality policy
  - Integrity policy
- Bell-LaPadula model
- Biba's model

Bell-LaPadula Model, Step 1

- Security levels arranged in linear ordering
  - Top Secret: highest
  - Secret
  - Confidential
  - Unclassified: lowest

Classification

Clearance

	TS	S	C	UC
	O_1	O_2	O_3	O_4
TS	RW	R-	R-	R-
S	-W	RW	R	R
C	-W	-W	RW	R
UC	-W	-W	-W	RW

- Subjects are issued *security clearance* L(s)
- Objects have *security classification* L(o)

Example

security level	*subject*	*object*
Top Secret	Tamara	Personnel Files
Secret	Samuel	E-Mail Files
Confidential	Claire	Activity Logs
Unclassified	Ulaley	Telephone Lists

- Tamara can read all files
- Claire cannot read Personnel or E-Mail Files
- Ulaley can only read Telephone Lists

Reading Information

- Information flows *up*, not *down*
  - "Reads up" disallowed, "reads down" allowed
- Simple Security Condition (Step 1)
  - Subject s can read object o iff $L(o) \leq L(s)$ and s has permission to read o
    - Note: combines mandatory control (relationship of security levels) and discretionary control (the required permission)
  - Sometimes called "no reads up" rule

Writing Information

- Information flows up, not down
  - "Writes up" allowed, "writes down" disallowed
- *-Property (Step 1)
  - Subject s can write object o iff $L(s) \le L(o)$ and s has permission to write o
    - Note: combines mandatory control (relationship of security levels) and discretionary control (the required permission)
  - Sometimes called "no writes down" rule

ACL

TS: {(TS, rw), (S, -w), (C, -w), (UC, -w)}

have group? yes

T:S {(TS, rw), (S, C, UC, -w)}

Define Procedure subject S, P, O object operation

{(TS, r), (TS, S, C, UC, w)}

CL

TS: {(TS, rw), (S, r), (C, r), (UC, r)}

Ring S, O

S < O
S = O
S > O

if (S < O) r
if (S == O) rw
if (S > O) w

ACL ✓

CL ✓

Ring ✗ original cannot work

✓ we can modify

Access Control Matrix

- Objects are sorted in descending according to their security classification.
- Subjects are sorted in descending according to their security clearance.
- What does the ACM look like?
- What mechanism can implement the ACM?
  - Ring

Problem in Step 1 Model

Sol 1: Put T_1 & T_2 on same level (can read eachother)
Sol 2: different security levels (One can read other)

- Examples
  - A general is leading two teams in a covert operation.
    - How to prevent the two teams exchange information?
  - A headquarter of a company is managing to branch offices.
    - How to prevent the two offices to exchange information?

Bell-LaPadula Model, Step 2

- Expand notion of security level to include categories
  - Security level is (*clearance*, *category set*)
  - A category corresponds to a set of information
- Examples
  - (Top Secret, { NUC, EUR, ASI })
  - (Confidential, { EUR, ASI })
  - (Secret, { NUC, ASI })

Levels and Dominance

TS, S

S_1, S_2

~~{TS, φ}~~ {TS, S_1} {TS, S_2} {TS, S_1, S_2}

~~{S, φ}~~ {S, S_1} {S, S_2} {S, S_1, S_2}

	TS S_1S_2	TS S_1	TS S_2	S S_1S_2	S S_1	S S_2
TS S_1	RW	r	r	r	r	r
TS S_1	w	RW	—	—	r	—
TS S_2	w	--	RW	—	—	r
S, S_1, S_2	w	--	--	RW	r–	r–
S, S_1	w	w	—	–w	RW	– –
S, S_2	w	--	w	–w	--	RW

- A: a level
- C: a set
- (A, C) *dom* (A', C') iff $A' \le A$ and $C' \subseteq C$
- Examples
  - (Top Secret, {NUC, ASI}) *dom* (Secret, {NUC})
  - (Secret, {NUC, EUR}) *dom* (Confidential,{NUC, EUR})
  - (Top Secret, {NUC}) ¬*dom* (Confidential, {EUR})

Levels and Ordering

- Security levels partially ordered
  - Any pair of security levels may (or may not) be related by *dom*
- "dominates" serves the role of "greater than" in step 1
  - "greater than" is a total ordering, though

Reading Information

- Information flows *up*, not *down*
  - "Reads up" disallowed, "reads down" allowed
- Simple Security Condition (Step 2)
  - Subject s can read object o iff $L(s)$ *dom* $L(o)$ and s has permission to read o
    - Note: combines mandatory control (relationship of security levels) and discretionary control (the required permission)
  - Sometimes called "no reads up" rule

Writing Information

- Information flows up, not down
  - "Writes up" allowed, "writes down" disallowed
- *-Property (Step 2)
  - Subject s can write object o iff $L(o)$ *dom* $L(s)$ and s has permission to write o
    - Note: combines mandatory control (relationship of security levels) and discretionary control (the required permission)
  - Sometimes called "no writes down" rule

Access Control Matrix

ACL CL Ring Dom, =, <, No Dom

- Levels: TS and S
- Sets: A and B
  - {A,B}, {A}, {B}
- ACM?
- Implementation?

Basic Security Theorem, Step 2

- If a system is initially in a secure state, and every transition of the system satisfies the simple security condition, step 2, and the *-property, step 2, then every state of the system is secure
  - Proof: induct on the number of transitions
  - In actual Basic Security Theorem, discretionary access control treated as third property, and simple security property and *-property phrased to eliminate discretionary part of the definitions — but simpler to express the way done here.

Problem

- Colonel has (Secret, {NUC, EUR}) clearance
- Major has (Secret, {EUR}) clearance
  - Major can talk to colonel (“write up” or “read down”)
  - Colonel cannot talk to major (“read up” or “write down”)
  - Colonel cannot issue commands

/proc
/sys

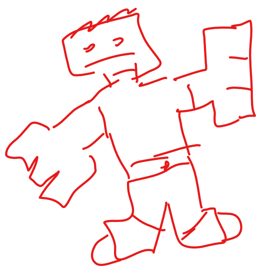
Solution

- Define maximum, current levels for subjects
  - *maxlevel*(*s*) *dom curlevel*(*s*)
- Example
  - Treat Major as an object (Colonel is writing to him/her)
  - Colonel has *maxlevel* (Secret, { NUC, EUR })
  - Colonel sets *curlevel* to (Secret, { EUR })
  - Now *L*(Major) *dom curlevel*(Colonel)
  - Colonel can write to Major without violating "no writes down"

* repromote to Major

DG/UX System

- Provides mandatory access controls
  - MAC label identifies security level Monitored Access control
  - Default labels, but can define others
- Initially
  - Subjects assigned MAC label of parent
    - Initial label assigned to user, kept in Authorization and Authentication database
  - Object assigned label at creation
    - Explicit labels stored as part of attributes
    - Implicit labels determined from parent directory



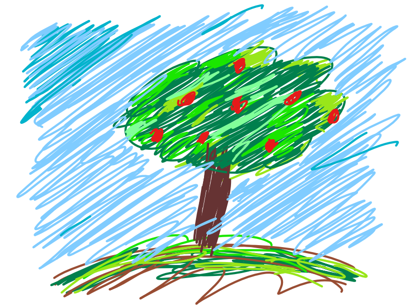
MAC Regions

- Figure 5-3

Critical Kernel Data

Hierarchy levels	Categories	Region	
	A&A database, audit log	Admin region	TS
	User data and applications	User region	S
VP�1	Site executables (OS = Kernel libs } Because everybody needs to use these)	Virus Prevention Region	C
VP�2	Trusted data		
VP�3	Executables not part of the TCB (At bottom, can be r but not w)		
VP�4	Executables part of the TCB		
VP�5	Reserved for future use		

ACM of DG/UX



B-L Model

	AA	User	Sys
AA	rw	r	r
User	w (write to audit log)	rw	r
Sys	w	w	rw

DG/UX

sanitation

	AA	User	Sys
AA	$\mathbf{r}_{\mathbf{tup}}\mathbf{w}_{\mathbf{tup}}$	$\mathbf{rw}_{\mathbf{san}}$	r
User	-	$\mathbf{r}_{\mathbf{tup}}\mathbf{w}_{\mathbf{tup}}$	r
Sys	-	w	$\mathbf{r}_{\mathbf{tup}}\mathbf{w}_{\mathbf{tup}}$

MAC Regions

- Administrative region (highest and special)
  - For logs, MAC label definitions, and so forth
  - No read up (B-L model)
  - No write up (no arbitrary alteration) from lower regions
    - This is an additional MAC to the B-L model
  - Administrative processes with MAC labels in this region can sanitize data and send data to user processes in the user region.
    - Sanitize is the key to confidentiality

MAC Regions

- Virus protection region (lowest)
  - Store system programs
  - Can be read/executed by users (B-L)
  - Cannot be modified (no write) by users (B-L)
- User program region (in the middle)
  - What if a user program is a virus

Using MAC Labels

- Simple security condition implemented
- *-property not fully implemented
  - Process MAC must equal object MAC
  - Writing allowed only at the same security level
- Overly restrictive in practice
  - Instead of one MAC level, using a range of MAC levels

MAC Tuples

- MAC range is a set of labels with upper, lower bound assigned to objects
  - Upper bound must dominate lower bound of range
  - An object has a MAC tuple.
  - A subject has a MAC label and a tuple
    - The subject can change its label within its tuple.
- Examples
  1. [(Secret, {NUC}), (Top Secret, {NUC})]
  2. [(Secret, ∅), (Top Secret, {NUC, EUR, ASI})]
  3. [(Confidential, {ASI}), (Secret, {NUC, ASI})]

MAC Tuples

- Process can read object when:
  - Object MAC range (*lr*, *hr*); process MAC label *pl*
  - *pl dom hr*
    - Process MAC label grants read access to upper bound of range
- Example
  - Peter, with label (Secret, {EUR}), cannot read paper
    - (Top Secret, {NUC, EUR}) *dom* (Secret, {EUR})
  - Paul, with label (Top Secret, {NUC, EUR, ASI}) can read paper
    - (Top Secret, {NUC, EUR, ASI}) *dom* (Top Secret, {NUC, EUR})

MAC Tuples

- Process can write object when:
  - Object MAC range (*lr*, *hr*); process MAC label *pl*
  - *pl* ∈ (*lr*, *hr*)
    - Process MAC label grants write access to any label in range
- Example
  - Peter, with label (Secret, {EUR}), can write paper
    - (Top Secret, {NUC, EUR}) *dom* (Secret, {EUR}) and (Secret, {EUR}) *dom* (Secret, {EUR})
  - Paul, with label (Top Secret, {NUC, EUR, ASI}), cannot read paper
    - (Top Secret, {NUC, EUR, ASI}) *dom* (Top Secret, {NUC, EUR})

Outline

- Overview
  - Confidentiality policy
  - Integrity policy
- Bell-LaPadula model
- Biba's model

Intuition for Integrity Levels

- The higher the level, the more confidence
  - A program will execute correctly
  - Data is accurate and dependable
- Note relationship between integrity and trustworthiness
- Important point: *integrity levels are **not** security levels*

Biba's Model

- Similar to Bell-LaPadula model
    1. $s \in S$ can read $o \in O$ iff $i(s) \leq i(o)$
    2. $s \in S$ can write to $o \in O$ iff $i(o) \leq i(s)$
    3. $s_1 \in S$ can execute $s_2 \in S$
        - iff $i(s_2) \leq i(s_1)$, when the result affects s_2 (Biba)
        - iff $i(s_1) \leq i(s_2)$, when the result affects s_1 (Locus)
- Add compartments and discretionary controls to get full dual of Bell-LaPadula model

	TS	S	C	NC
TS	rw	w	w	w
S	r	rw	w	w
C	r	r	rw	w
NC	r	r	r	rw

if s or o is affected by exec.

Access Control Matrix

- Integrity levels of subject and objects
- ACM and implementation?
- Combined B-L and Biba model
  - Security levels = integrity levels
  - ACM and implementation?

LOCUS and Biba

- Goal: prevent untrusted software from altering data or other software
- Approach: make levels of trust explicit
  - Credibility rating based on estimate of software's trustworthiness (0 untrusted, *n* highly trusted)
  - Trusted file systems contain software with a single *credibility level* (objects)
  - A user has a *risk level* (subjects)
  - A user can execute software if risk level ≤ credibility level
  - Must use *run-untrusted* command to run software at lower credibility level (objects)

Summary of Topic 1

- Security concepts
  - Confidentiality
  - Integrity
  - Availability
- Access control practices and theories
  - Access control mechanisms
  - Access control matrix
  - Security policy and models