Trusted Computing and Multilevel Security

First Edition
by William Stallings and Lawrie Brown
Lecture slides by Lawrie Brown
Trusted Computing and Multilevel Security

• present some interrelated topics:
  - formal models for computer security
  - multilevel security
  - trusted systems
  - mandatory access control
  - security evaluation
Formal Models for Computer Security

- two fundamental computer security facts:
  - all complex software systems have flaw/bugs
  - is extraordinarily difficult to build computer hardware/software not vulnerable to attack
- hence desire to prove design and implementation satisfy security requirements
- led to development of formal security models
  - initially funded by US DoD
- Bell-LaPadula (BLP) model very influential
Bell-LaPadula (BLP) Model

- developed in 1970s
- as a formal access control model
- subjects and objects have a **security class**
  - top secret > secret > confidential > unclassified
  - subject has a **security clearance** level
  - object has a **security classification** level
  - class control how subject may access an object
- applicable if have info and user categories
Multi-Level Security
BLP Formal Description

- based on current state of system \((b, M, f, H)\):
  (current access set \(b\), access matrix \(M\), level function \(f\), hierarchy \(H\))
- three BLP properties:
  - ss-property: \((S_i, O_j, \text{read})\) has \(f_c(S_i) \geq f_o(O_j)\).
  - \*-property: \((S_i, O_j, \text{append})\) has \(f_c(S_i) \leq f_o(O_j)\) and \((S_i, O_j, \text{write})\) has \(f_c(S_i) = f_o(O_j)\)
  - ds-property: \((S_i, O_j, A_x)\) implies \(A_x \in M[S_i]\)
- BLP give formal theorems
  - theoretically possible to prove system is secure
  - in practice usually not possible
BLP Rules

1. get access
2. release access
3. change object level
4. change current level
5. give access permission
6. rescind access permission
7. create an object
8. delete a group of objects
BLP Example
BLP Example cont.
BLP Example cont.
MULTICS Example

- `current-process`
- `DSBR`
- `descriptor segment`
- `root`
Biba Integrity Model

- various models dealing with integrity
- strict integrity policy:
  - simple integrity: $I(S) \geq I(O)$
  - integrity confinement: $I(S) \leq I(O)$
  - invocation property: $I(S_1) \geq I(S_2)$
Clark-Wilson Integrity Model
Chinese Wall Model

(a) Example set

(b) John has access to Bank A and Oil A

(c) Jane has access to Bank A and Oil B
Reference Monitors
Trojan Horse Defence
Multilevel Security (MLS)

- a class of system that has system resources (particularly stored information) at more than one security level (i.e., has different types of sensitive resources) and that permits concurrent access by users who differ in security clearance and need-to-know, but is able to prevent each user from accessing resources for which the user lacks authorization.
MLS Security for Role-Based Access Control

- rule based access control (RBAC) can implement BLP MLS rules given:
  - security constraints on users
  - constraints on read/write permissions
  - read and write level role access definitions
  - constraint on user-role assignments
RBAC MLS Example
MLS Database Security

(a) Classified by table

(b) Classified by column (attribute)
### MLS Database Security

#### (c) Classified by row (tuple)

<table>
<thead>
<tr>
<th>Department Table</th>
<th>Employee</th>
</tr>
</thead>
<tbody>
<tr>
<td>Did</td>
<td>Name</td>
</tr>
<tr>
<td>4</td>
<td>accts</td>
</tr>
<tr>
<td>8</td>
<td>PR</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### (d) Classified by element

<table>
<thead>
<tr>
<th>Department Table</th>
<th>Employee</th>
</tr>
</thead>
<tbody>
<tr>
<td>Did</td>
<td>Name</td>
</tr>
<tr>
<td>4 - U</td>
<td>accts - U</td>
</tr>
<tr>
<td>8 - U</td>
<td>PR - U</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
MLS Database Security
Read Access

- DBMS enforces simple security rule (no read up)
- easy if granularity entire database / table level
- inference problems if have column granularity
  - if can query on restricted data can infer its existence
    - SELECT Ename FROM Employee WHERE Salary > 50K
    - solution is to check access to all query data
- also have problems if have row granularity
  - null response indicts restricted/empty result
- no extra concerns if have element granularity
MLS Database Security
Write Access

- enforce *-security rule (no write down)
- have problem if a low clearance user wants to insert a row with a primary key that already exists in a higher level row:
  - can reject, but user knows row exists
  - can replace, compromises data integrity
  - can polyinstantiation and insert multiple rows with same key, creates conflicting entries
- same alternatives occur on update
- avoid problem if use database / table granularity
Trusted Platform Module (TPM)

- concept from Trusted Computing Group
- hardware module at heart of hardware / software approach to trusted computing
- uses a TPM chip on
  - motherboard, smart card, processor
  - working with approved hardware / software
  - generating and using crypto keys
- has 3 basic services: authenticated boot, certification, and encryption
Authenticated Boot Service

- responsible for booting entire O/S in stages
- ensuring each is valid and approved for use
  - verifying digital signature associated with code
  - keeping a tamper-evident log
- log records versions of all code running
- can then expand trust boundary
  - TPM verifies any additional software requested
    - confirms signed and not revoked
- hence know resulting configuration is well-defined with approved components
Certification Service

- once have authenticated boot
- TPM can certify configuration to others
  - with a digital certificate of configuration info
  - giving another user confidence in it
- include challenge value in certificate to also ensure it is timely
- provides hierarchical certification approach
  - trust TPM then O/S then applications
Encryption Service

• encrypts data so it can be decrypted
  - by a certain machine in given configuration

• depends on
  - master secret key unique to machine
  - used to generate secret encryption key for every possible configuration only usable in it

• can also extend this scheme upward
  - create application key for desired application version running on desired system version
TPM Functions

- Cryptographic co-processor
- HMAC engine
- SHA-1 engine
- Opt-In
- Non-volatile memory
- Key generation
- Random number generator
- Power detection
- Execution engine
- Volatile memory
Protected Storage Function
Trusted Systems

• security models aimed at enhancing trust
• work started in early 1970’s leading to:
  - Trusted Computer System Evaluation Criteria (TCSEC), Orange Book, in early 1980s
  - further work by other countries
  - resulting in Common Criteria in late 1990s
• also Computer Security Center in NSA
  - with Commercial Product Evaluation Program
  - evaluates commercially available products
  - required for Defense use, freely published
Common Criteria (CC)

- ISO standards for security requirements and defining evaluation criteria to give:
  - greater confidence in IT product security
  - from formal actions during process of:
    - development using secure requirements
    - evaluation confirming meets requirements
    - operation in accordance with requirements
- evaluated products are listed for use
CC Requirements

- have a common set of potential security requirements for use in evaluation
- target of evaluation (TOE) refers to product / system subject to evaluation
- functional requirements
  - define desired security behavior
- assurance requirements
  - that security measures are effective and correct
- have classes of families of components
CC Profiles and Targets
CC Security Paradigm

Diagram showing the Target of Evaluation (TOE) and the TOE security functions interface (TSFI). The diagram includes the Target of evaluation (TOE) with TOE security functions (TSF) and Enforces TOE Security Policy (TSP). The diagram also includes User, Human, and remote IT product interactions with Security attributes, Object/Information, Resource, Process, and Subject.
Smartcard PP

- simple PP example
- describes IT security requirements for smart card use by sensitive applications
- lists threats
- PP requirements:
  - 42 TOE security functional requirements
  - 24 TOE security assurance requirements
  - IT environment security requirements
- with rationale for selection
Assurance

- “degree of confidence that the security controls operate correctly and protect the system as intended”
- applies to:
  - product security requirements, security policy, product design, implementation, operation
- various approaches analyzing, checking, testing various aspects
CC Assurance Levels

- EAL 1 - functionally tested
- EAL 2: structurally tested
- EAL 3: methodically tested and checked
- EAL 4: methodically designed, tested, and reviewed
- EAL 5: semiformally designed and tested
- EAL 6: semiformally verified design and tested
- EAL 7: formally verified design and tested
Evaluation

- ensure security features correct & effective
- performed during / after TOE development
- higher levels need greater rigor and cost
- input: security target, evidence, actual TOE
- result: confirm security target satisfied for TOE
- process relates security target to some of TOE:
  - high-level design, low-level design, functional spec, source code, object code, hardware realization
- higher levels need semiformal / formal models
Evaluation Parties & Phases

- **evaluation parties:**
  - sponsor - customer or vendor
  - developer - provides evidence for evaluation
  - evaluator - confirms requirements satisfied
  - certifier - agency monitoring evaluation process

- **phases:**
  - preparation, conduct of evaluation, conclusion

- government agency regulates, e.g. US CCEVS

- have peering agreements between countries
  - saving time / expense by sharing results
Summary

- Bell-LaPadula security model
- other models
- reference monitors & trojan horse defence
- multilevel secure RBAC and databases
- trusted platform module
- common criteria
- assurance and evaluation