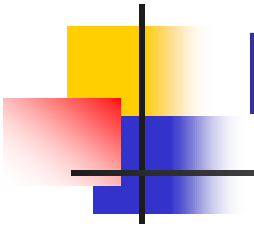
CS590U

Access Control: Theory and Practice

Lecture 3 (Jan 17)

& Lecture 4 (Jan 19)

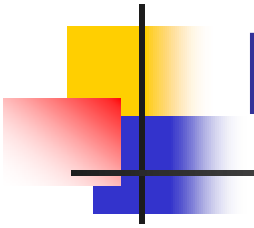
The Bell-LaPadula Model



Main Objective of BLP

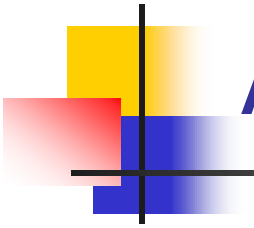
- Enable one to show that a computer system can securely process classified information

- Main reference
  - Bell and LaPadula: Secure Computer Systems: Unified Exposition and Multics Interpretation. MITRE Tech Report.



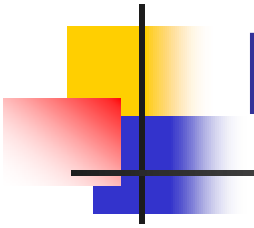
Methodology in BLP

- Define an abstract model that can be used to describe computer systems.
  - the model
- Define what does it mean for a system in the model to be secure.
  - the policy
- Develop techniques to prove that a system in the model is secure



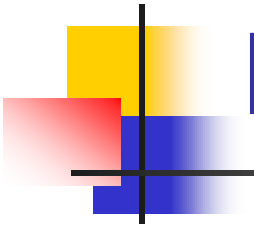
Approach of BLP

- Use state-transition systems to describe computer systems
- Define a system as secure iff. every reachable state satisfies 3 properties
  - simple-security property, *-property, discretionary-security property
- Prove a Basic Security Theorem (BST)
  - so that one can prove a system is secure by proving things about the system description



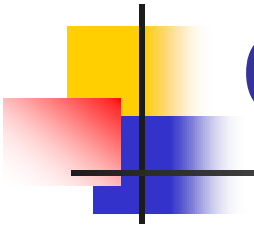
Main Contributions of BLP

- The overall methodology to show that a system is secure
  - adopted in many later works
- The state-transition model
  - which includes an access matrix, subject security levels, object levels, etc.
- The introduction of *-property
  - ss-property is not enough to stop illegal information flow



Main Technical Flaws of BLP

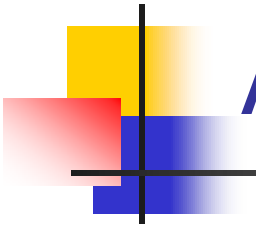
- The BLP notion of security is neither necessary nor sufficient to stop illegal information flows
- That BLP defines security as a state-based property is too low level and limited in expressive power
- The BST fails to provide *necessary and sufficient* conditions for verifying a system is BLP-secure



Outline

- Overview of BLP
- The system model in BLP
- The BLP notion of security
- The Basic Security Theorem
- Related work
- Conclusions

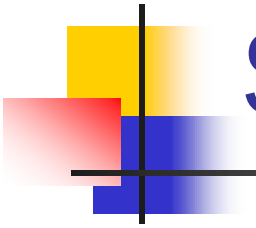
Alphabet



- A set S of subjects
- A set S_T of trusted subjects
- A set A of access modes
  - A={ execute, read, append, write }
    - observation and alteration
- A set of ***O*** of object identifiers
- A partially ordered set ħ L, · i of security levels

States Z

- Each state z2 Z is a 4-tuple h O, b, M, F i
  - O ½ $\boldsymbol{O}$ — objects in current state
  - b µ S £ O£ A — current access set
  - M: S £ O ! 2^A — an access matrix
  - F=h f_S, f_O, f_C i — security level functions
    - f_S: S ! L — subject maximal level
    - f_O: O ! L — object level
    - f_C: S ! L — subject current level



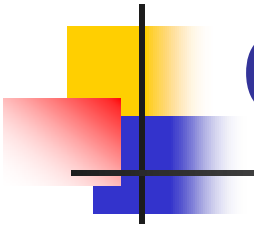
State Transitions

- A set R of requests
  - get-access(s,o,a)
  - release-acess(s,o,a)
  - give-access(s_1,s_2,o,a)
  - rescind-access(s_1,s_2,o,a)
  - create-object(s,o,*l*)
  - reclassify-object(s,o, *l*)
  - destruct-object(s,o)
  - change-current-level(s, *l*)

BLP Systems

- Definition: A system is given by (z_0, W)
  - z_0 is the initial state
  - W½ R£ D £ Z £ Z D={ yes, no }
    - (req, d, z', z)2W defines one state transition

- Definition: An appearance of the system (z_0, W) is a sequence ħ z_0, (req_1, d_1, z_1), (req_2, d_2, z_2), ···, (req_t, d_t, z_t) i where
  - t is a natural number
  - 8 i s.t. 1 · i · t $(req_i, d_i, z_i, z_{i-1})$2 W

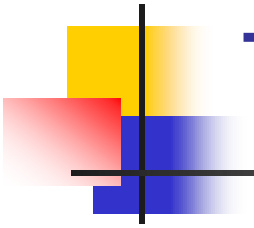


Outline

- Overview of BLP
- The system model in BLP
- The BLP notion of security
- The Basic Security Theorem
- Related work
- Conclusions

BLP: Secure States

- Definition: $z=\hbar O, b, M, F=\hbar f_S, f_O, f_C \imath \imath$ is a secure state if and only if
  - z satisfies the ss-property,
    - i.e., $8\ (s,o,a) 2\ b\ (a 2\ \{read,write\}\ )\ f_S(s)\ ¸\ f_O(o)$
  - z satisfies the *-property,
    i.e., $8\ (s,o,a) 2 b$ where $s \notin S_T$
    - $a 2\ \{read,write\}\ )\ f_C(s)\ ¸\ f_O(o)$ no read up
    - $a 2\ \{append,write\}\ )\ f_C(s)\ \cdot\ f_O(o)$ no write down
  - z satisfies the ds-property,
    - i.e., $8\ (s,o,a) 2\ b\ \ a 2\ M[s,o]$



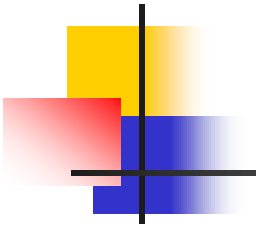
The *-property

- Does *-property imply ss-property? No.
  - The ss-property uses maximal level.
  - The *-property applies only to untrusted subjects
- Can one say *-property is just no-write-down? No.
- The original BLP model doesn't require that $f_C(s) \cdot f_S(s)$
  - setting one's current level higher only gets less access right

BLP: Secure Systems

- Definition: A system (z_0,W) is secure iff. every state in every appearance of the system is secure.

- State-based definition is limited in expressive power
  - cannot express a policy that says a state z_2 occurs after a state z_1 in an appearance is not acceptable

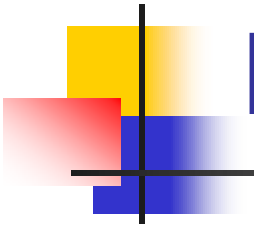


Is BLP Notion of Security Good?

- The objective of BLP security is to ensure
  - a subject cleared at a low level should never read information classified high
- The ss-property and the *-property are sufficient to stop such information flow at any given state.
- What about information flow across states?

BLP Security Is Not Sufficient!

- Consider a system with s_1, s_2, o_1, o_2
  - $f_S(s_1)=f_C(s_1)=f_O(o_1)=$high
  - $f_S(s_2)=f_C(s_2)=f_O(o_2)=$low
- And the following execution
  - s_1 gets access to o_1, read something, release access, then change current level to low, get write access to o_2, write to o_2
- Every state is secure, yet illegal information exists



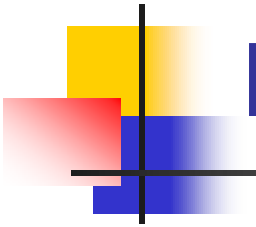
But People Already Know This.

- The following have been proposed:
  - subject cannot change current levels
  - require a subject to “forgot” everything when changing levels
- But the original BLP security is wrong!
- And all the fixes limit the applicability of the model
- It is not the model that is wrong, it is the definition of security that is wrong.

BLP Security Is Not Necessary!

- Consider a system with only s_1, s_2, o_1, o_2
  - $f_S(s_1)=f_C(s_1)=f_O(o_1)=$high
  - $f_S(s_2)=f_C(s_2)=f_O(o_2)=$low
- And an access matrix s.t. s_2 cannot access o_2
- And the following execution
  - s_1 gets access to o_1, and get write access to o_2, then the state violates *-property
- Why is this system bad?

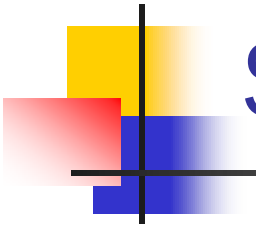
Summary of Issues with BLP Notion of Security



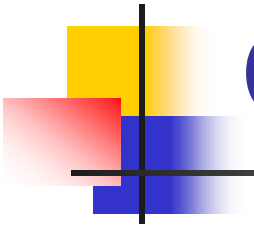
- BLP notion of security is neither sufficient nor necessary to stop illegal information flow (through overt channels)

- The state based approach is too low level and limited in expressive power

How to Fix The BLP Notion of Security?



- May need to differentiate externally visible objects from other objects
  - e.g., a printer is different from a memory object
- State-sequence based property
  - e.g., exists no sequence of states so that there is an information path from a high object to a low externally visible object or to a low subject



Outline

- Overview of BLP
- The system model in BLP
- The BLP notion of security
- The Basic Security Theorem
- Related work
- Conclusions

The Basic Security Theorem

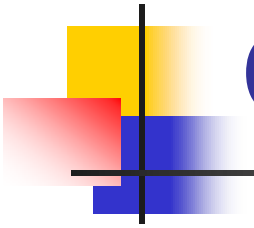
- Corollary A1 [BL76]: A system (z_0,W) is a secure system iff. z_0 is a secure state and W satisfies the conditions of theorems A1, A2, and A3 for each action.

Theorem A1

- Theorem A1 [BLP76]: A system (z_0, W) satisfies the ss-property iff. z_0 satisfies the ss-property and W satisfies the following conditions for each action $\in$ req, d, (O',b',M',F'), (O,b,M,F)i
  - each (s,o,a)2 b'-b satisfies the ss-property wrt. F'
  - each (s,o,a)2 b which doesn't satisfy ss-property wrt. F' is not in b'

Basic Security Theorem

- Restatement of The Basic Security Theorem: A system (z_0,W) is a secure system if and only if z_0 is a secure state and each action of the system leads the system into a secure state.
- Given a system (z_0,W), $\sigma 2W$ is an action of the system iff. there is an appearance of the system that uses σ



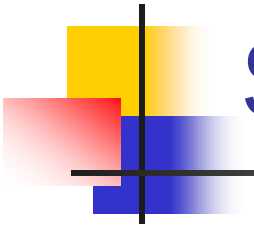
Observations of the BST

- The BST is a result of defining security as a state-based property.
- The BST cannot be used to justify the BLP notion of security
  - This is McLean's main point in his papers
    - "A Comment on the Basic Security Theorem of Bell and LaPadula" [1985]
    - "Reasoning About Security Models" [1987]
    - "The Specification and Modeling of Computer Security" [1990]

Observations of the BST

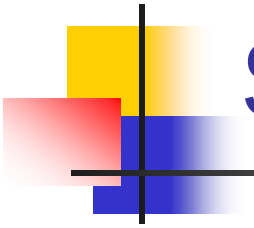
- The BST intends to provide a necessary and sufficient condition for verifying that a system is secure without running the system
  - [McLean 90]: “The most notable theorem known about BLP-security is called the `Basic Security Theorem (BST), which gives necessary and sufficient conditions for a system starting in a secure state to never reach a non-secure state.”

BST and Static Verification of Security



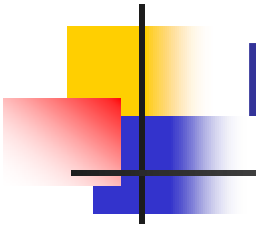
- Can one use BST to verify whether a system is secure or not without running the system?
  - Repeat of BST: A system (z_0,W) is a secure system if and only if z_0 is a secure state and each action of the system leads the system into a secure state.

BST and Static Verification of Security



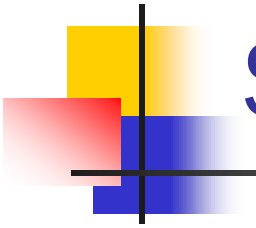
- Yes and No.
  - if every $\sigma 2W$ leads the system into a secure state, then the system is secure
  - if some $\sigma 2W$ leads the system into an insecure state, then we don't know whether the system is secure
    - as we don't know whether σ is an action or not
- BST provides effectively only sufficient (but not necessary) conditions.

Analogy with Safety Analysis in HRU



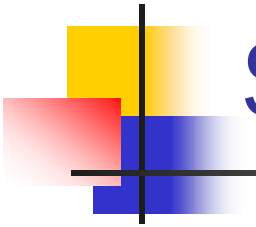
- Safety analysis in HRU is undecidable.
  - [Harrison, Ruzzo, Ullman 1976]
- Nonetheless, we can state a BST Theorem: A protection system is a secure system if and only if each action leads the system into a secure state.
  - an instance of a command is an action if can appear in one of the system runs
- The theorem is trivially true, but useless.

On The Inductive Nature of Security

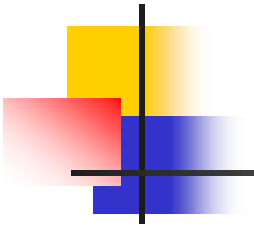


- Bell and LaPadula say
  - We say that the BST establishes the “inductive nature of security” in that it shows that the preservation of security from one state to the next guarantees total system security.
  - The importance of this result should not be underestimated. Other problems of seemingly comparable difficult are not of an inductive nature. The problems of data- and resource-sharing, for example, are not inductive.

On The Inductive Nature of Security

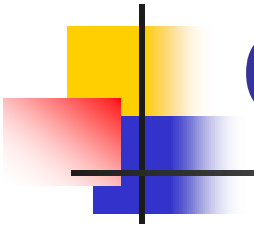


- In fact, the most trivial example of deadlock can arise in any nontrivial sharing system that decides immediately to grant or deny a request for access. Resolution of this problem requires knowledge of future possibilities, queues of requests, and process priorities. The result, therefore, that security (as defined in the model) is inductive establishes the relative simplicity of maintaining security: the minimum check that the proposed new state is ``secure'' is both necessary and sufficient for full maintenance of security.



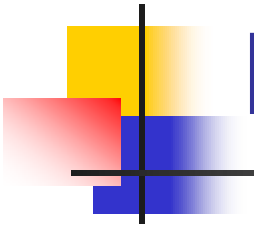
Is Security of “Inductive Nature”?

- No. Some counter arguments are:
  - BLP notion of security doesn’t capture security.
  - If one define deadlock freeness as a state-based property, rather than the inability to progress to the next state, then deadlock freeness is also “inductive”
  - When a subject requests to access something that is not allowed, then it is rejected. This may also cause deadlock.



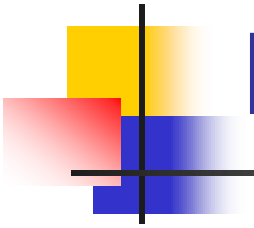
Outline

- Overview of BLP
- The system model in BLP
- The BLP notion of security
- The Basic Security Theorem
- Related work
- Conclusions



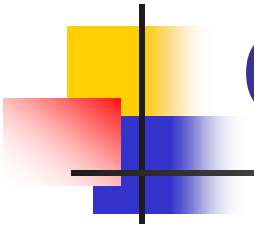
McLean's Criticism of BLP

- BST cannot be used to justify BLP security
  - [McLean 1985] If one define security to be any other state-based property, BST still holds
    - Defense [Bell 1988]: exactly what is security is outside the model
  - [McLean 1987] System Z, defines a state change that downgrade everything
    - Defense 1: Tranquility principle disallows that
    - Defense 2: If such state change is desired, then fine.



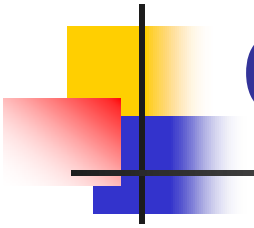
McLean's Criticism of BLP

- In [McLean 1990], McLean try to justify BLP security using an alternative notion of secure transition, but admitted failure
  - We believe that BLP notion of security is inherent problematic
- Tranquility principle
  - the classification of active objects will not change during the normal operation.



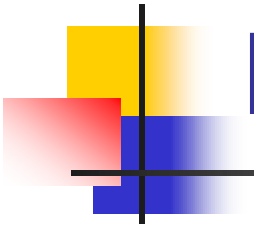
Other Issues with BLP

- Often discussed in the textbooks
  - BLP concerns only with confidentiality, but not integrity
  - Blind writes
  - Trusted subjects are a necessary evil
    - processes like device drivers and memory management software have to be trusted subjects



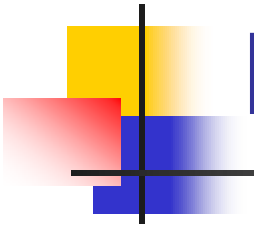
Outline

- Overview of BLP
- The system model in BLP
- The BLP notion of security
- The Basic Security Theorem
- Related work
- Conclusions



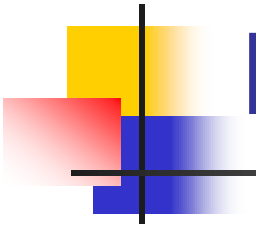
Main Contributions of BLP

- The overall methodology to show that a system is secure
  - adopted in many later works
- The state-transition model
  - which includes an access matrix, subject security levels, object levels, etc.
- The introduction of *-property
  - ss-property is not enough to stop illegal information flow



Main Technical Flaws of BLP

- The BLP notion of security is neither necessary no sufficient to stop illegal information flows
- That BLP defines security as a state-based property is too low level and limited in expressive power
- The BST fails to provide *necessary and sufficient* conditions for verifying a system is BLP-secure



End of Lecture 3 & 4

- Next lecture:
  - Non-interference and non-deducability