DATA SECURITY AND PRIVACY

Week 3: Security Models: BLP, Biba, and Clark-Wilson

PURDUE UNIVERSITY

Readings for This Lecture

- Bell and La Padula: “Secure Computer System: Unified Exposition and MULTICS Interpretation”
  - Section II
- Kenneth J. Biba: "Integrity Considerations for Secure Computer Systems", MTR-3153, The Mitre Corporation, April 1977.
- David D. Clark and David R. Wilson. “A Comparison of Commercial and Military Computer Security Policies.” In IEEE SSP 1987.

PURDUE UNIVERSITY

Related Readings for This Lecture

- Other Related Papers:
  - David FC. Brewer and Michael J. Nash. "The Chinese Wall Security Policy." in IEEE SSP 1989.

PURDUE UNIVERSITY

Outline

- Overview of the Bell Lapadula Model
- Details of the Bell Lapadula Model
- Analysis of the Bell Lapadula Model
- More on Multi-level Security
- TCSEC and Common Criteria
- Biba Integrity Models
- Clark-Wilson Model and Chinese Wall Policy

PURDUE UNIVERSITY

Access Control at Different Abstractions

- Using principals
  - Determines which principals (user accounts) can access what documents
- Using subjects
  - Determines which subjects (processes) can access what resources
  - This is where BLP focuses on

Multi-Level Security (MLS) (1)

- There are security classifications or security levels
  - Users/principals/subjects have **security clearances**
  - Objects have **security classifications**
- Example of security levels
  - Top Secret > Secret > Confidential > Unclassified
- Security goal (confidentiality):
  - Ensures that information does not flow to those not cleared for that level

Multi-Level Security (MLS) (2)

- The capability of a computer system to carry information with different sensitivities (i.e. classified information at different security levels), permit simultaneous access by users with different security clearances and needs-to-know, and prevent users from obtaining access to information for which they lack authorization.
  - Discretionary access control fails to achieve MLS
- Typically use Mandatory Access Control
- Primary Security Goal: Confidentiality

Mandatory Access Control

- Mandatory access controls (MAC) restrict the access of subjects to objects based on a system-wide policy
  - denying users full control over the access to resources that they create. The system security policy (as set by the administrator) entirely determines the access rights granted

Bell-LaPadula Model: A MAC Model for Multi-level Security

- Introduce in 1973
- Air Force was concerned with security in time-sharing systems
  - Many OS bugs
  - Accidental misuse
- Main Objective:
  - Enable one to formally show that a computer system can securely process classified information

What is a Security Model?

- A **model** describes the system
  - e.g., a high level specification or an abstract machine description of what the system does
- A **security policy**
  - defines the security requirements for a given system
- **Verification techniques** that can be used to show that a policy is satisfied by a system
- System Model + Security Policy = Security Model

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 given the description of a system, one can prove that the system is secure

Outline

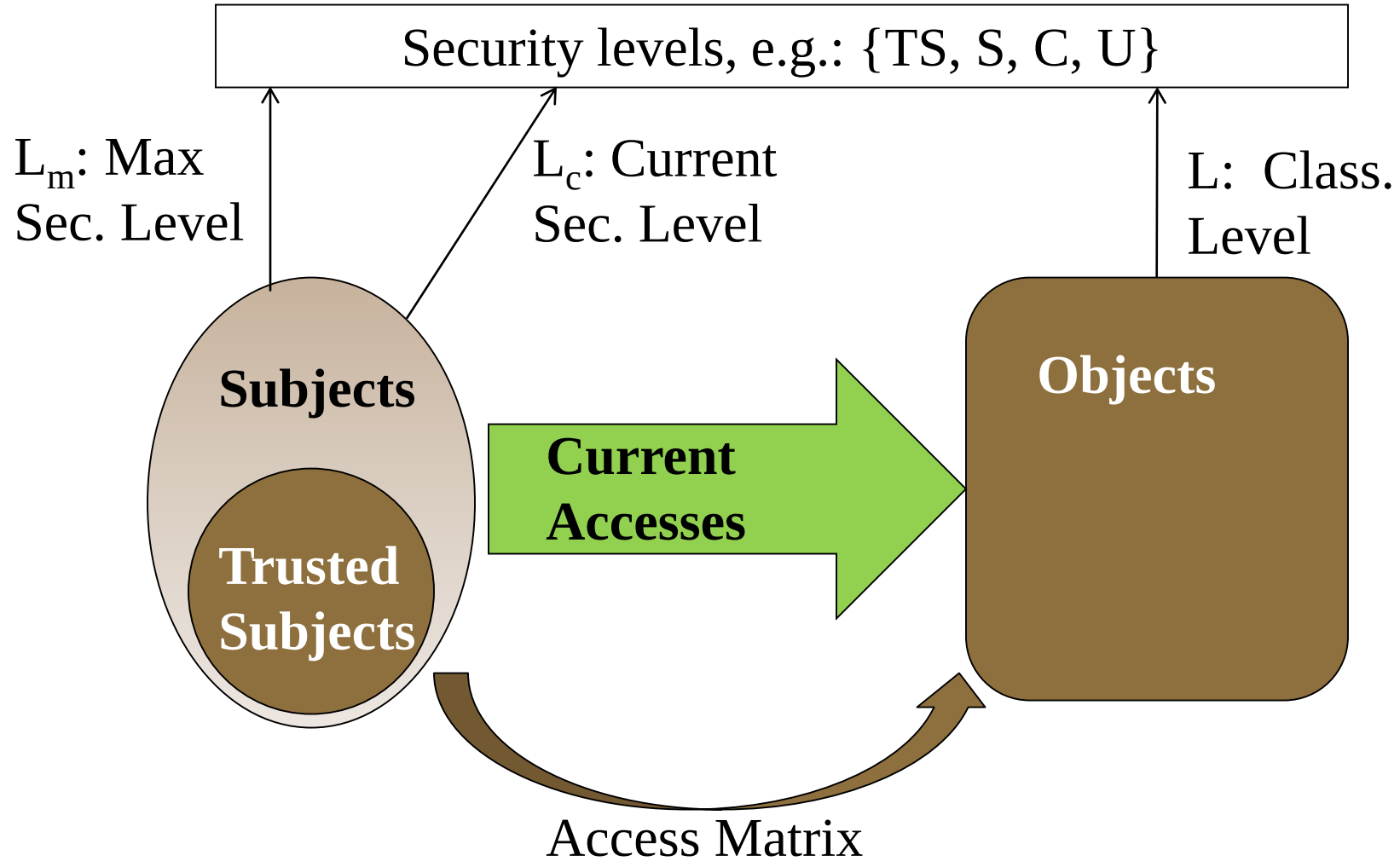
- Overview of the Bell Lapadula Model
- Details of the Bell Lapadula Model
- Analysis of the Bell Lapadula Model
- More on Multi-level Security
- TCSEC and Common Criteria
- Biba Integrity Models
- Clark-Wilson Model and Chinese Wall Policy

PURDUE UNIVERSITY

The BLP Security Model

- A computer system is modeled as a state-transition system
  - There is a set of subjects; some are designated as **trusted**.
  - Each state has objects, an access matrix, and the current access information.
  - There are state transition rules describing how a system can go from one state to another
  - Each subject s has a maximal sec level $L_m(s)$, and a current sec level $L_c(s)$
  - Each object has a classification level

Elements of the BLP Model

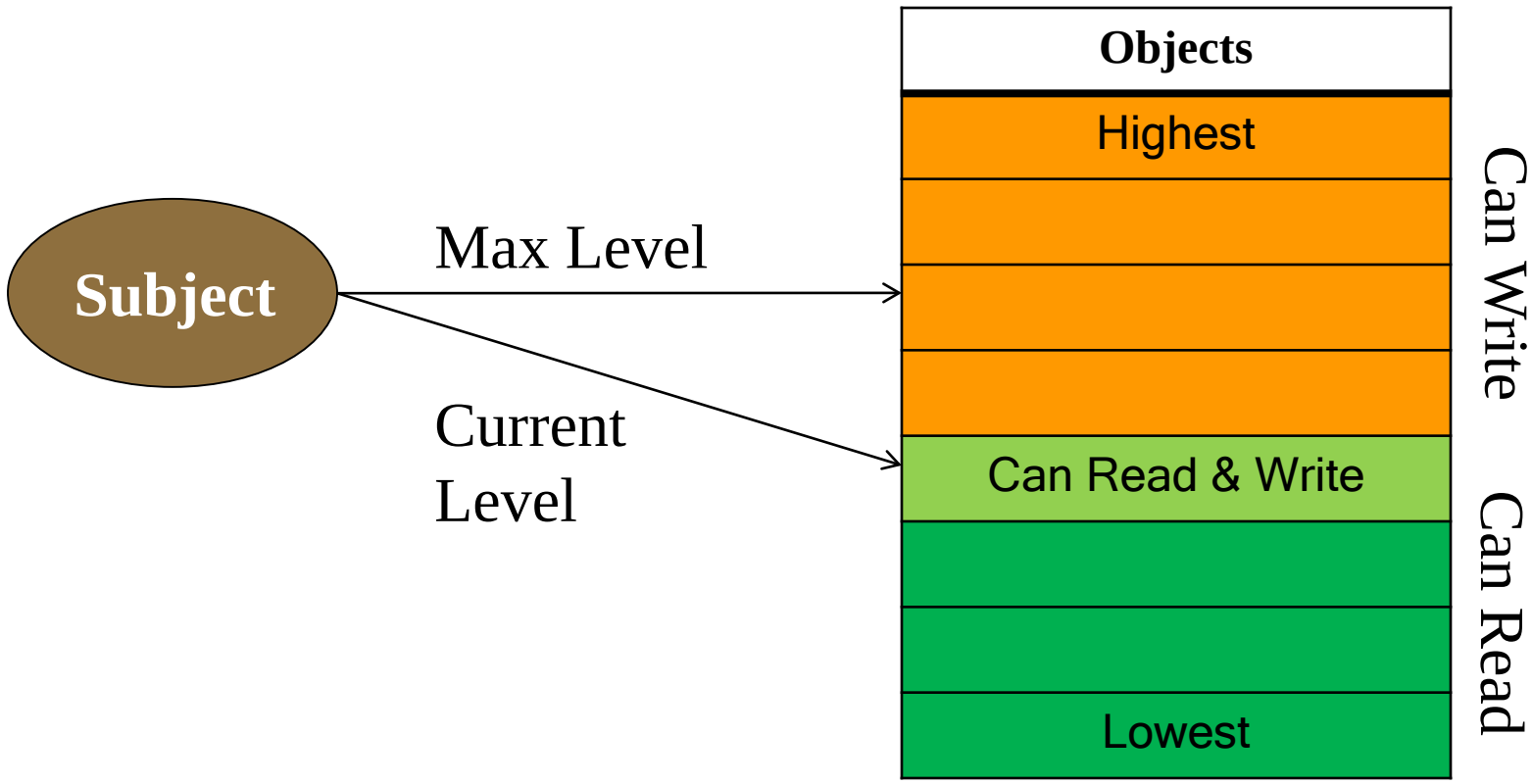


PURDUE UNIVERSITY

The BLP Security Policy

- A state is secure if it satisfies
  - Simple Security Condition (**no read up**):
    - S can read O iff $L_m(S) \geq L(O)$
  - The Star Property (**no write down**): for any S **that is not trusted**
    - S can read O iff $L_c(S) \geq L(O)$ (no read up)
    - S can write O iff $L_c(S) \leq L(O)$ (no write down)
  - Discretionary-security property
    - every access is allowed by the access matrix
- A system is secure if and only if every reachable state is secure.

Implication of the BLP Policy



PURDUE UNIVERSITY

STAR-PROPERTY

- Applies to subjects not to principals and users
- Users are trusted (must be trusted) not to disclose secret information outside of the computer system
- Subjects are not trusted because they may have Trojan Horses embedded in the code they execute
- Star-property prevents overt leakage of information and does not address the covert channel problem

Outline

- Overview of the Bell Lapadula Model
- Details of the Bell Lapadula Model
- Analysis of the Bell Lapadula Model
- More on Multi-level Security
- TCSEC and Common Criteria
- Biba Integrity Models
- Clark-Wilson Model and Chinese Wall Policy

PURDUE UNIVERSITY

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 two subjects s_1,s_2 and two objects o_1,o_2
  - $f_S(s_1) = f_C(s_1) = f_O(o_1) = \text{high}$
  - $f_S(s_2) = f_C(s_2) = f_O(o_2) = \text{low}$
- And the following execution
  - s_1 gets read 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 flow exists, assuming that a subject can store information from one state to the next
- Solution: tranquility principle: subject cannot change current levels, or cannot drop current level to below the highest level read so far

More on the BLP Notion of Security

- When a subject A copies information from high to a low object f, this violates the star-property, but no information leakage occurred yet
  - Only when B, who is not cleared at high, reads f, does leakage occurs
  - If the access matrix limits access to f only to A, then such leakage may never occur
- BLP notion of security is neither sufficient nor necessary to stop illegal information flow (through direct/overt channels)
- The state based approach is too low level and limited in expressive power

How to Fix The BLP Notion of Security (if we want to)?

- 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., define security to mean that there 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

The Basic Security Theorem

- This provides the verification techniques piece in
  - Model – Policy – Verification framework

- **Restatement of The Basic Security Theorem:** A system is a secure system **if and only if** the starting state is a secure state and **each action** (concrete state transition that could occur in an execution sequence) of the system leads the system into a secure state.

Observations of the BST

- The BST is purely a result of defining security as a state-based property.
  - It holds for any other state-based property
- The BST cannot be used to justify that the BLP notion of security is "good"
  - 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]

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

Outline

- Overview of the Bell Lapadula Model
- Details of the Bell Lapadula Model
- Analysis of the Bell Lapadula Model
- More on Multi-level Security
- TCSEC and Common Criteria
- Biba Integrity Models
- Clark-Wilson Model and Chinese Wall Policy

PURDUE UNIVERSITY

Other Limitations with BLP

- Deal only with confidentiality, does not deal with integrity at all
  - Confidentiality is often not as important as integrity in most situations
  - Integrity is addressed by models such as Biba, Clark-Wilson, which we will cover later

- Does not deal with information flow through covert channels

Overt (Explicit) Channels vs. Covert Channels

- Security objective of MLS in general, BLP in particular, is
  - high-classified information cannot flow to low-cleared users
- Illegal information flow via overt channels (e.g., read/write an object) is blocked by BLP
- Illegal information flow by covert channels can still occur
  - communication channel based on the use of system resources not normally intended for communication between the subjects (processes) in the system

Examples of Covert Channels

- Using file lock as a shared boolean variable
- By varying its ratio of computing to input/output or its paging rate, the service can transmit information to a concurrently running process
- Timing of packets being sent
- In general, shared resources can be used as covert channels
  - What is needed is one party can affect them, and another can observe the effects
- Covert channels are often noisy
- However, information theory and coding theory can be used to encode and decode information through noisy channels

More on Covert Channels

- Covert channels cannot be blocked by *-property
- It is generally very difficult, if not impossible, to block all covert channels
- One can try to limit the bandwidth of covert channels
- Military requires cryptographic components be implemented in hardware
  - to avoid trojan horse leaking keys through covert channels
- Covert channels are achieved by collaboration or high and low subjects.

More on MLS: Security Levels

- Used as attributes of both subjects & objects
  - clearance & classification
- Typical military security levels:
  - top secret $\geq$ secret $\geq$ confidential $\geq$ unclassified
- Typical commercial security levels
  - restricted $\geq$ proprietary $\geq$ sensitive $\geq$ public

Security Categories

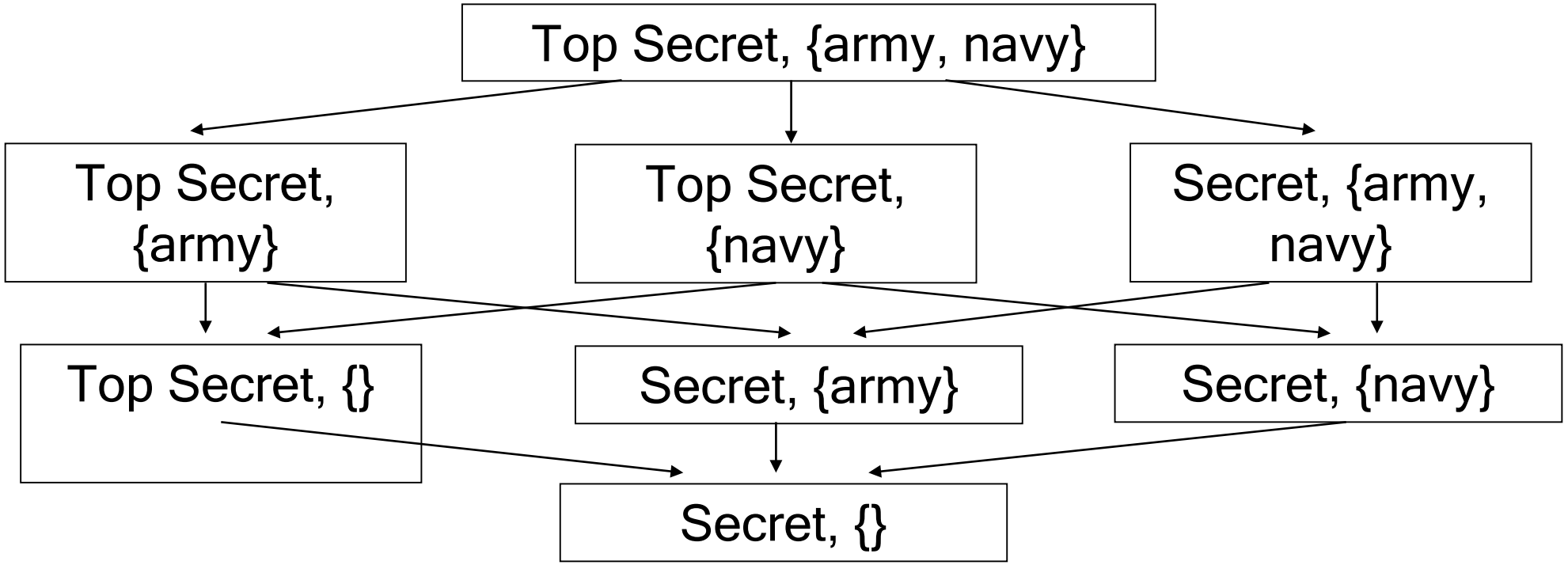
- Also known as compartments
- Typical military security categories
  - army, navy, air force
  - nato, nasa, noforn
- Typical commercial security categories
  - Sales, R&D, HR
  - Dept A, Dept B, Dept C

Security Labels

- Labels = Levels $\times$ P (Categories)
  - P (Categories) is powerset (set of all subsets) of Categories
- There is a natural partial ordering relationship among Labels
  - (e1, C1) $\leq$ (e2, C2) iff. e1 $\leq$e2 and C1 $\subseteq$ C2
- This ordering relation is a partial order
  - reflexive, transitive, anti-symmetric
  - e.g., $\subseteq$
- All security labels form a lattice

An Example Security Lattice

- levels={top secret, secret}
- categories={army, navy}



PURDUE UNIVERSITY

The need-to-know principle

- Even if someone has all the necessary official approvals (such as a security clearance) to access certain information they should not be given access to such information unless they have a *need to know*: that is, unless access to the specific information necessary for the conduct of one's official duties.

- Can be implemented using categories and/or DAC

PURDUE UNIVERSITY®

Outline

- Overview of the Bell Lapadula Model
- Details of the Bell Lapadula Model
- Analysis of the Bell Lapadula Model
- More on Multi-level Security
- TCSEC and Common Criteria
- Biba Integrity Models
- Clark-Wilson Model and Chinese Wall Policy

PURDUE UNIVERSITY

Terminology: Trusted vs. Trustworthy

- A component of a system is trusted means that
  - the security of the system depends on it
  - failure of component can break the security policy
  - determined by its role in the system
- A component is trustworthy means that
  - the component deserves to be trusted
  - e.g., it is implemented correctly
  - determined by intrinsic properties of the component

Terminology: Trusted Computing Base (TCB)

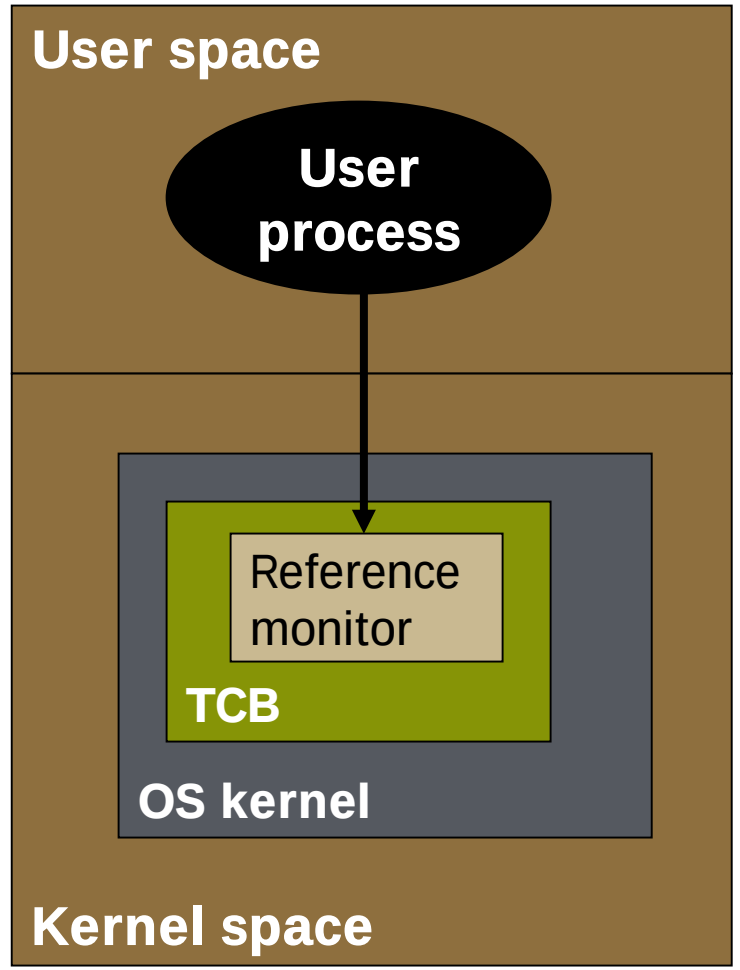
- The set of all hardware, software and procedural components that enforcing the security policy depends upon.
  - In order to break security, an attacker must subvert some part of the TCB.
  - The smaller the TCB, the more secure a system is.
- What would a Trusted Computing Base in a Unix/Linux system consists of?
  - Depends on the security objective
  - hardware, kernel, system binaries, system configuration files, setuid root programs, etc., at the minimum
- One approach to improve security is to reduce the size of TCB, i.e., reduce what one relies on for security.

Assurance

- Assurance: “estimate of the likelihood that a system will not fail in some particular way”
- Based on factors such as
  - Software architecture
    - E.g., kernelized design,
  - Development process
  - Who developed it
  - Technical assessment

Kernelized Design for High-Assurance Systems

- Uses the reference monitor concept
- Reference monitor
  - Part of TCB
  - All system calls go through reference monitor for security checking
  - Security does not depends on the whole kernel
  - Most OS not designed this way



PURDUE UNIVERSITY

Reference Monitor

- Three required properties for reference monitors in high-assurance systems
  - tamper-proof
  - non-bypassable (complete mediation)
  - small enough to be analyzable

Assurance Criteria

- Criteria are specified to enable evaluation
- Originally motivated by military applications, but now is much wider
- Examples
  - Orange Book (Trusted Computer System Evaluation Criteria)
  - Common Criteria

TCSEC: 1983–1999

- Trusted Computer System Evaluation Criteria
  - Also known as the Orange Book
  - Series that expanded on Orange Book in specific areas was called *Rainbow Series*
  - Developed by National Computer Security Center, US Dept. of Defense
- Heavily influenced by Bell-LaPadula model and reference monitor concept
- Emphasizes confidentiality

Evaluation Classes C and D

Division D: Minimal Protection

D Did not meet requirements of any other class

Division C: Discretionary Protection

C1 *Discretionary protection* : DAC, Identification and Authentication, TCB should be protected from external tampering, …

C2 *Controlled access protection* : object reuse, auditing, more stringent security testing

PURDUE UNIVERSITY®

Division B: Mandatory Protection

B1 *Labeled security protection* : informal security policy model; MAC for named objects; label exported objects; more stringent security testing

B2 *Structured protection* : formal security policy model; MAC for all objects, labeling; trusted path; least privilege; covert channel analysis, configuration management

B3 *Security domains* : satisfies three reference monitor requirements; system recovery procedures; constrains code development; more documentation requirements

PURDUE UNIVERSITY®

Division A: Verification Protection

A1 *Verified design* :
functionally equivalent to B3, but require the use of formal methods for assurance; trusted distribution; code, formal top-level specification (FTLS) correspondence

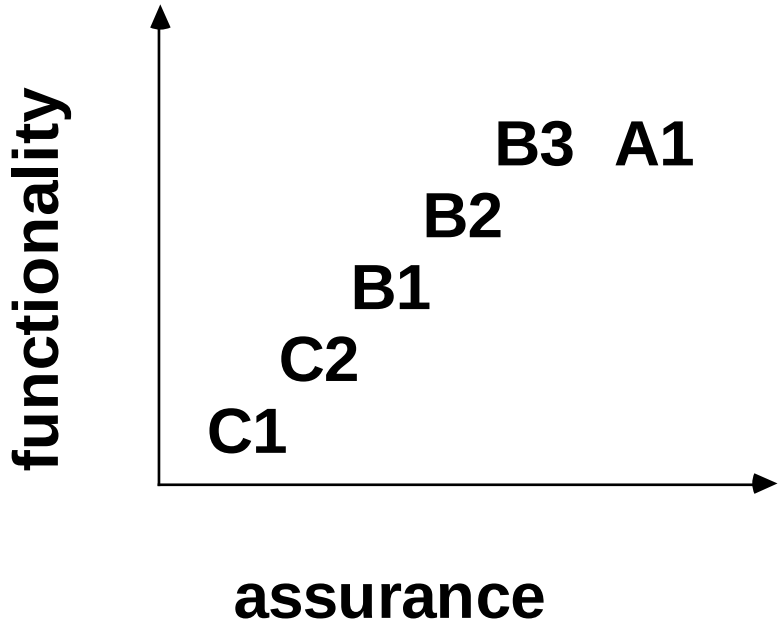
PURDUE UNIVERSITY

Limitations

- Written for operating systems
  - NCSC introduced “interpretations” for other things such as networks (*Trusted Network Interpretation*, the Red Book), databases (*Trusted Database Interpretation*, the Purple or Lavender Book)
- Focuses on BLP
  - Most commercial firms do not need MAC
- Does not address data integrity or availability
  - Critical to commercial firms
- Combine functionality and assurance in a single linear scale

FUNCTIONALITY VS ASSURANCE

- **functionality is multi-dimensional**
- **assurance has a linear progression**



PURDUE UNIVERSITY

Common Criteria: 1998–Present

- An international standard (ISO/IEC 15408)
- Began in 1998 with signing of Common Criteria Recognition Agreement with 5 signers: US, UK, Canada, France, Germany
- As of December 2015, 19 authorizing countries, and 8 consuming countries (do not evaluate, accept evaluated products)
- Standard 15408 of International Standards Organization
- *De facto* US security evaluation standard, replaces TCSEC

Common Criteria

- Does not provide one list of security features
- Describes a framework where security requirements can be specified, claimed, and evaluated
- Key concepts
  - **Target Of Evaluation (TOE)**: the product or system that is the subject of the evaluation.
  - **Security Target (ST)**: a document that identifies the security properties one wants to evaluate against
  - **Protection Profile (PP)**: a document that identifies security requirements relevant to a user community for a particular purpose.
  - **Evaluation Assurance Level (EAL)** - a numerical rating (1-7) reflecting the assurance requirements fulfilled during the evaluation.

CC Functional Requirements

- Contains 11 classes of functional requirements
  - Each contains one or more families
  - Elaborate naming and numbering scheme
- Classes: Security Audit, Communication, Cryptographic Support, User Data Protection, Identification and Authentication, Security Management, Privacy, Protection of Security Functions, Resource Utilization, TOE Access, Trusted Path
- For example, within Identification and Authentication, there are the following families
  - Authentication Failures, User Attribute Definition, Specification of Secrets, User Authentication, User Identification, and User/Subject Binding

PURDUE UNIVERSITY®

CC Assurance Requirements

- Ten security assurance classes:
  - Protection Profile Evaluation
  - Security Target Evaluation
  - Configuration Management
  - Delivery and Operation
  - Development
  - Guidance Documentation
  - Life Cycle
  - Tests
  - Vulnerabilities Assessment
  - Maintenance of Assurance

PURDUE UNIVERSITY

Protection Profiles (PP)

- “A CC protection profile (PP) is an implementation-independent set of security requirements for a category of products or systems that meet specific consumer needs”
  - Subject to review and certified
- Requirements
  - Functional
  - Assurance
  - EAL

PURDUE UNIVERSITY

Protection Profiles

- Example: Controlled Access PP (CAPP_V1.d)
  - Security functional requirements
    - Authentication, User Data Protection, Prevent Audit Loss
  - Security assurance requirements
    - Security testing, Admin guidance, Life-cycle support, …
  - Assumes non-hostile and well-managed users
  - Does not consider malicious system developers

PURDUE UNIVERSITY®

Security Targets (ST)

- “A security target (ST) is a set of security requirements and specifications to be used for evaluation of an identified product or system”
- Can be based on a PP or directly taking components from CC
- Describes specific security functions and mechanisms

Evaluation Assurance Levels 1 – 4

EAL 1: Functionally Tested
- Review of functional and interface specifications
- Some independent testing

EAL 2: Structurally Tested
- Analysis of security functions, incl. high-level design
- Independent testing, review of developer testing

EAL 3: Methodically Tested and Checked
- More testing, Some dev. environment controls;

EAL 4: Methodically Designed, Tested, Reviewed
- Requires more design description, improved confidence that TOE will not be tampered

Evaluation Assurance Levels 5 – 7

EAL 5: Semiformally Designed and Tested

- Formal model, modular design
- Vulnerability search, covert channel analysis

EAL 6: Semiformally Verified Design and Tested

- Structured development process

EAL 7: Formally Verified Design and Tested

- Formal presentation of functional specification
- Product or system design must be simple
- Independent confirmation of developer tests

Implications of EALs

- A higher EAL means nothing more, or less, than that the evaluation completed a more stringent set of quality assurance requirements.
- It is often assumed that a system that achieves a higher EAL will provide its security features more reliably, but there is little or no published evidence to support that assumption.
- Anything below EAL4 doesn't mean much
- Anything above EAL4 is very difficult to achieve for complex systems such as OS
- Evaluation is done for environments assumed by vendors

Criticism of CC

- Evaluation is a costly process (often measured in hundreds of thousands of US dollars) -- and the vendor's return on that investment is not necessarily a more secure product
- Evaluation focuses primarily on assessing the evaluation documentation, not the product itself
- The effort and time to prepare evaluation-related documentation is so cumbersome that by the time the work is completed, the product in evaluation is generally obsolete
- Industry input, including that from organizations such as the Common Criteria Vendor's Forum, generally has little impact on the process as a whole

Outline

- Overview of the Bell Lapadula Model
- Details of the Bell Lapadula Model
- Analysis of the Bell Lapadula Model
- More on Multi-level Security
- TCSEC and Common Criteria
- Biba Integrity Models
- Clark-Wilson Model and Chinese Wall Policy

PURDUE UNIVERSITY

Biba Integrity Models

- Kenneth J. Biba: "Integrity Considerations for Secure Computer Systems", MTR-3153, The Mitre Corporation, April 1977.

- Motivations
  - BLP focuses on confidentiality
  - In most systems, integrity is equally, if not more, important
  - Data integrity vs. System integrity
    - Data integrity means that data cannot be changed without being detected

What is integrity in systems?

- Attempt 1: Critical data do not change.
- Attempt 2: Critical data changed only in “correct ways”
  - Analogy: in DB, integrity constraints are used for consistency
- Attempt 3: Critical data changed only through certain “trusted programs”
- Attempt 4: Critical data changed only as intended by authorized users.

Biba: Integrity Levels

- Each subject (process) has an integrity level
- Each object has an integrity level
- Integrity levels are totally ordered
- **Integrity levels different from security levels in confidentiality protection**
  - Highly sensitive data may have low integrity
  - What is an example of a piece of data that needs high integrity, but no confidentiality?

Strict Integrity Policy (BLP reversed)

- Rules:
  - s can read o iff $i(s) \leq i(o)$
    - no read down
    - stops indirect sabotage by contaminated data
  - s can write to o iff $i(s) \geq i(o)$
    - no write up
    - stops directly malicious modification
- Fixed integrity levels
- No information path from low object/subject to high object/subject
- Too restrictive for practice. Why?

Subject Low-Water Policy

- Rules
  - s can always read o; however, after reading
    $$i(s) \leftarrow \min[i(s), i(o)]$$
  - s can write to o iff $i(s) \geq i(o)$
- Subject's integrity level decreases as reading lower integrity data
- No information path from low-object to high-object
- Dual to a form of Tranquility Principle in BLP

Object Low-Water Mark Policy

- Rules
  - s can read o; iff $i(s) \leq i(o)$
  - s can always write to o; after writing
    $i(o) \leftarrow \min[i(s), i(o)]$
- Object's integrity level decreases as it is contaminated by subjects
- In the end, objects that have high labels have not been contaminated

Low-Water Mark Integrity Audit Policy

- Rules
  - s can always read o; after reading
    $$i(s) \leftarrow \min[i(s), i(o)]$$
  - s can always write to o; after writing
    $$i(o) \leftarrow \min[i(s), i(o)]$$
- Tracing, but not preventing contamination
- Similar to the notion of taint tracking in software security

The Ring Policy

- Rules
  - Any subject can read any object
  - s can write to o iff $i(s) \geq i(o)$
- Integrity levels of subjects and objects are fixed.
- Intuitions:
  - subjects are trusted to process low-level inputs correctly
- Dual to Trusted Subjects (not subject to star-property) in BLP

PURDUE UNIVERSITY

Five Mandatory Policies in Biba

- Strict integrity policy
- Subject low-water mark policy
- Object low-water mark policy
- Low-water mark Integrity audit policy
- Ring policy

- In practice, one may be using one or more of these policies, possibly applying different policies to different subjects
  - E.g., subjects for which ring policy is applied are trusted to be able to correctly handle inputs;

Integrity Policies Options

		When high subject requests to read low object:		
		Deny	Allow, drop subject level afterwards	Allow, no change to subject level
When low subject requests to write high object:	Deny	Strict Integrity Policy	Subject Low Water Policy	Ring Policy
	Allow, drop object level afterwards	Object Low Water Policy	Low-Water Audit Policy	
	Allow, no change to object level			

Why last row is empty, but last column is not?

PURDUE UNIVERSITY

Object Integrity Levels

- The integrity level of an object may be based on
  - **Quality** of information (levels may change)
    - Degree of trustworthiness
    - Contamination level:
  - **Importance** of the object (levels do not change)
    - Degree of being trusted
    - Protection level: writing to the objects should be protected
- What should be the relationship between the two meanings, which level should be higher?

PURDUE UNIVERSITY®

Integrity vs. Confidentiality

Confidentiality	Integrity
Control reading preserved if confidential info is not read	Control writing preserved if important obj is not changed (by writing)
For subjects who need to read, control writing after reading is sufficient, no need to trust them	For subjects who need to write, one has to trust them, control reading before writing is **not** sufficient

Integrity requires trust in subjects!

Analogy

- Confidentiality violation: leak a secret
  - CAN be prevented even if I tell the secret to a person I do not trust, so long as I can lock the person up AFTERWARDS to prevent further leakage
    - The person cannot leak confidential info w/o talking
- Integrity violation: follow a wrong instruction
  - CANNOT be prevented if I follow instruction from an person I do not trust even if I lock the person up BEFOREHAND to prevent the person from receiving any malicious instruction
    - The person can invent malicious instruction without outside input

Key Difference between Confidentiality and Integrity

- For confidentiality, controlling reading & writing is sufficient
  - theoretically, no subject needs to be trusted for confidentiality; however, one does need trusted subjects in BLP to make system realistic
- For integrity, controlling reading and writing is insufficient
  - one has to trust all subjects who can write to critical data

Impacts of The Need to Trust Subjects

- Trusting only a small security kernel is no longer possible
- No need to worry about covert channels for integrity protection
- How to establish trust in subjects becomes a challenge.

Application of Integrity Protection

- Mandatory Integrity Control in Windows (since Vista)
  - Uses four integrity levels: Low, Medium, High, and System
  - Each process is assigned a level, which limit resources it can access
  - Processes started by normal users have Medium
  - Elevated processes have High
    - Through the User Account Control feature
  - Some processes run as Low, such as IE in protected mode
  - Reading and writing do not change the integrity level
    - Ring policy.

Outline

- Overview of the Bell Lapadula Model
- Details of the Bell Lapadula Model
- Analysis of the Bell Lapadula Model
- More on Multi-level Security
- TCSEC and Common Criteria
- Biba Integrity Models
- Clark-Wilson Model and Chinese Wall Policy

PURDUE UNIVERSITY

The Clark-Wilson Model

- David D. Clark and David R. Wilson. "A Comparison of Commercial and Military Computer Security Policies." In IEEE SSP 1987.
- Paper defends two conclusions:
  - There is a distinct set of security policies, related to integrity rather than disclosure, which are often of highest priority in the commercial data processing environment
    - no user of the system, even if authorized, may be permitted to modify data items in such a way that assets or accounting records of the company are lost or corrupted
  - Some separate mechanisms are required for enforcement of these policies, disjoint from those in the Orange Book

Two High-level Mechanisms for Enforcing Data Integrity (1)

- **Well-formed transaction**
  - a user should not manipulate data arbitrarily, but only in constrained ways that preserve or ensure data integrity
    - e.g., use an append-only log to record all transactions
    - e.g., double-entry bookkeeping
    - e.g., passwd

Data can be manipulated only through trusted code!

Two High-level Mechanisms for Enforcing Data Integrity (2)

- Separation of duty
  - ensure external consistency: data objects correspond to the real world objects
  - separating all operations into several subparts and requiring that each subpart be executed by a different person
  - e.g., the two-man rule

Implementing the Two High-level Mechanisms

- Mechanisms are needed to ensure
  - **control access to data:** a data item can be manipulated only by a specific set of programs
  - **program certification:** programs must be inspected for proper construction, controls must be provided on the ability to install and modify these programs
  - **control access to programs:** each user must be permitted to use only certain sets of programs
  - **control administration:** assignment of people to programs must be controlled and inspected

The Clarke-Wilson Model for Integrity

- Unconstrained Data Items (UDIs)
  - data with low integrity
- Constrained Data Items (CDIs)
  - data items within the system to which the integrity model must apply
- Integrity Verification Procedures (IVPs)
  - confirm that all of the CDIs in the system conform to the integrity specification
- Transformation Procedures (TPs)
  - well-formed transactions

Differences of Clark-Wilson from MAC/BLP

- A data item is not associated with a particular security level, but rather with a set of TPs
- A user is not given read/write access to data items, but rather permissions to execute certain programs

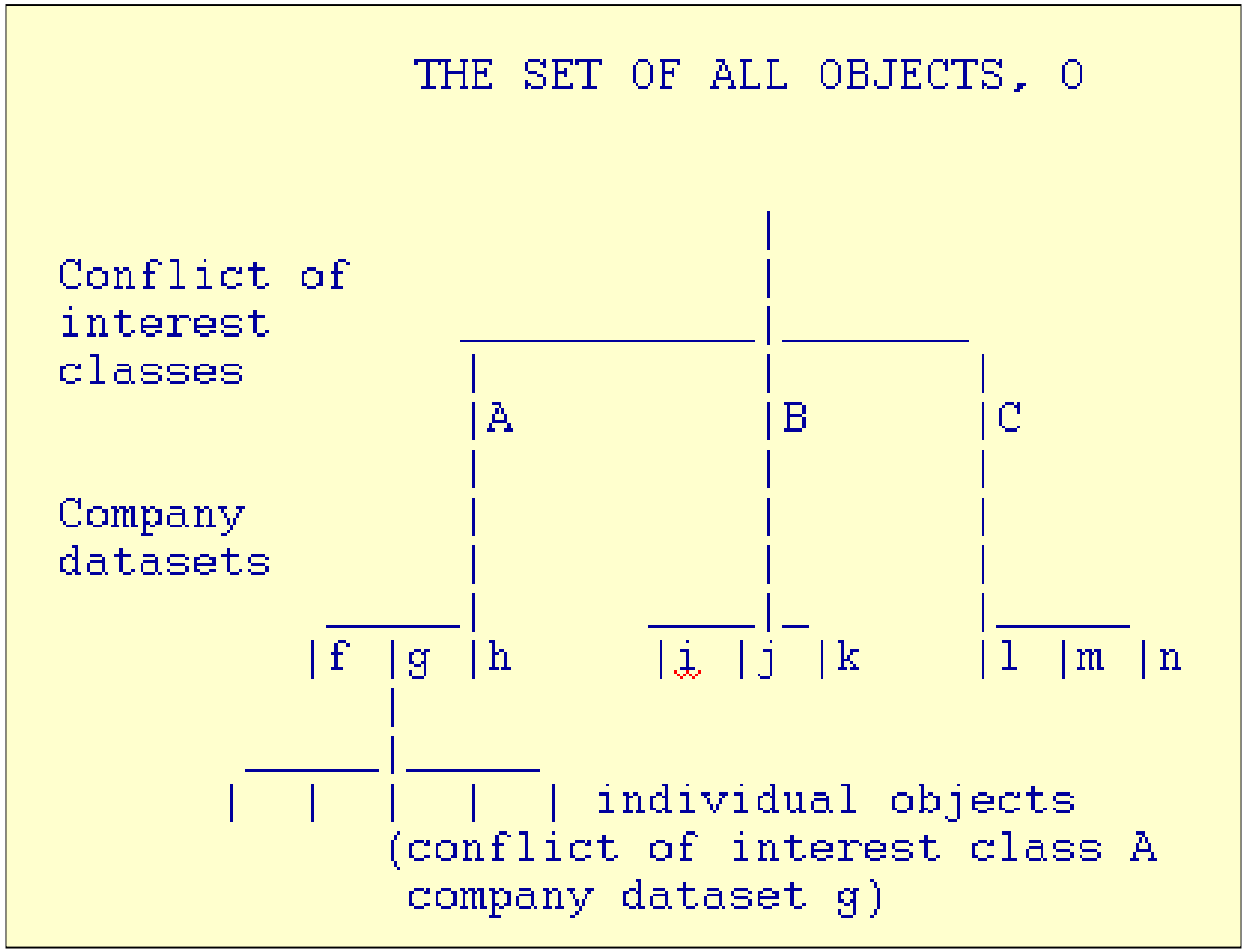
Comparison with Biba

- Biba lacks the procedures and requirements on identifying subjects as trusted
- Clark-Wilson focuses on how to ensure that programs can be trusted

The Chinese Wall Security Policy

- Goal: **Avoid Conflict of Interest**
- Data are stored in a hierarchical arranged system
  - the lowest level consists of individual data items
  - the intermediate level group data items into company data sets
  - the highest level group company datasets whose corporation are in competition

The Set of All Objects, O



PURDUE UNIVERSITY

Simple Security Rule in Chinese Wall Policy

- Access is only granted if the object requested:
  - is in the same company dataset as an object already accessed by that subject, i.e., within the Wall,

or

  - belongs to an entirely different conflict of interest class.

Summary

- Multi-level security focuses on protecting confidentiality
- Bell-Lapadula Model
- Biba Integrity Model
- Clark Wilson Model and Chinese wall policy

Next Topic

- Non-interference and non-deducibility
- Role based access control