DATA SECURITY AND PRIVACY

Week 1 Introduction to the Course and Overview of Access Control

Prof. Ninghui Li (Purdue University)

PURDUE UNIVERSITY®

Topics in the Course

- Access control (AC)
  - Operating system AC, mandatory AC, discretionary AC, role based AC, attribute-based AC, non-interference properties, integrity protection, AC in databases
- Data privacy
  - Privacy policies, data anonymization (k anonymity, t closeness, l diversity), differential privacy: concepts and algorithms, differential privacy in the local setting, membership privacy, differential privacy and machine learning
- Using crypto for data protection
  - secure multiparty computation, implementing crypto correctly, homomorphic encryption

Relationship to Other Courses

- Require basic knowledge from
  - 526 Information Security
  - 555 Cryptography
- Little overlap with
  - 527 Software Security
  - 528 Network Security
  - 523 Social Econ Legal Asp Of Sec
  - 529 Security Analytics

PURDUE UNIVERSITY®

- Time and location: TTh 10:30-11:45pm, LWSN B134
- Instructor: Ninghui Li <ninghui@purdue.edu>,
  - LWSN 2142K, office hours: after lecture and by appointment
- We will use piazza
- Teaching assistants: Zitao Li <li2490@purdue.edu>
  - Office hours, Monday 1:30-2:30pm + Tuesday 4-5pm
  - https://purdue-edu.zoom.us/j/99058485354?pwd=MDVYdDFDN1IwVnd3a2lDZUhodm9aZz09

PURDUE UNIVERSITY

Readings

- No required text, readings will be announced/distributed on course webpage.

Grading

- Homeworks (50% of grade)
  - About 6 assignments, which will be either written assignments, or small projects that require programming
  - Late policy: Three extension days to be used at your discretion
    - Must be stated explicitly in header of work being turned in
    - No fractional days
    - May not be used to extend submission past last day of class.
- Exams
  - 4 (in-class) quizs during the semester. (8% of grade)
  - Mid-term exam. (18% of grade) Final exam. (24% of grade)

Policies for Homework Cheating

- It is allowed/encouraged to discuss homework problems
- **However, if you looks at another student's program code, or (written or typed) answers , or let another student look at your program code or answers, that is considered cheating.**
- If caught for the first time, receive 0 in the assignment. For the second time, receive a failing grade in class.

PURDUE UNIVERSITY®

What is Information (Computer) Security?

- Security = Sustain desirable properties under intelligent adversaries
- Making the above statement precise requires clearly defining the following two
- What are the desirable properties?
- What the intelligent adversaries can do?
  - Needs to understand/model adversaries
  - Always think about adversaries.

Security Goals/Properties (C, I, A)

- Confidentiality (secrecy, privacy)
  - only those who are authorized to know can know
- Integrity (also authenticity in communication)
  - only modified by authorized parties and in permitted ways
  - do things that are expected
- Availability
  - those authorized to access can get access

What is (Personal) Privacy?

- **Sometimes people use privacy to refer to confidentiality.**
- **Personal privacy is complicated! It is primarily a social and legal concept.**
- Some concepts from the book “Understanding Privacy” by Daniel J. Solove:
  1. the right to be let alone
  2. limited access to the self
  3. ***secrecy–the concealment of certain matters from others;***
  4. ***control over others' use of information about oneself***
  5. personhood–the protection of one’s personality, individuality, and dignity;
  6. intimacy–control over, or limited access to, one’s intimate relationships or aspects of life.

PURDUE UNIVERSITY®

Security is Secondary

- What protection/security mechanisms one has in the physical world?
- Why the need for security mechanisms arises?
- ***Security is secondary to the interactions that make security necessary.***

> Robert H. Morris: The three golden rules to ensure computer security are: (1) do not own a computer; (2) do not power it on; and (3) do not use it.

PURDUE UNIVERSITY®

Information Security is Interesting

- The most interesting/challenging threats to security are posed by human adversaries
  - Security is harder than reliability
- Information security is a self-sustaining field
  - Can work both from attack perspective and from defense perspective
- Security is about benefit/cost tradeoff
  - Although often the tradeoff analysis is not explicit
- Security is not all technological
  - Humans are often the weakest link

PURDUE UNIVERSITY®

Information Security is Challenging

- Defense is almost always harder than attack.
- In which ways information security is more difficult than physical security?
  - adversaries can come from anywhere
  - computers enable large-scale automation
  - adversaries can be difficult to identify
  - adversaries can be difficult to punish
  - potential payoff can be much higher
- In which ways information security is easier than physical security?

PURDUE UNIVERSITY®

ACCESS CONTROL

What is Access Control?

- Quote from Security Engineering by Ross Anderson
  - “Its function is to control which principals (persons, processes, machines, …) have access to which resources in the system --- which files they can read, which programs they can execute, and how they share data with other principals, and so on.”
  - Access control is the traditional center of gravity of computer security. It is where security engineering meets computer science.

Access Control is Pervasive

- Application
  - business applications
- Middleware
  - DBMS
- Operating System
  - controlling access to files, ports
- Hardware
  - memory protection, privilege levels

Access Control is Interesting

- Has (relatively) well-developed theories
  - 40+ years history
  - some (quite involved) theory (apparently) not useful for other fields
- Many interesting and deep results
- Many misconceptions and debates

A BRIEF HISTORY OF ACCESS CONTROL RESEARCH

PURDUE UNIVERSITY®

Earlier Years: Time-Sharing Operating Systems

- Reference monitors (1972)
- Access matrix (1971)
- Discretionary access control
  - trojan horse can leak information

Confidentiality

- Bell-LaPadula Model
- Noninterference (1982)
- Nondeducibility (1986)
- Covert channel
- Proving information flow properties of systems and programs

Integrity

- Biba model
- Clark-Wilson
- Chinese Wall

Database Access Control

- System R approach: grant/revoke, view
- Ingres approach (query rewriting)
- Multilevel databases
- Object/relational databases
- Real systems
- SQL grant/revoke, view, stored procedures, fine-grained access control

Role-Based Access Control

- First in database context
- Then a generic access control approach
- Constraints
- Administration
- Extensions

PURDUE UNIVERSITY®

OPERATING SYSTEM SECURITY

PURDUE UNIVERSITY®

What Security Goals Does Operating System Provide (Past)?

- Originally: time-sharing computers: enabling multiple users to securely share a computer
  - Separation and sharing of processes, memory, files, devices, etc.
- What is the threat model?
  - Users may be malicious, users have terminal access to computers, software may be malicious/buggy, and so on
- What security mechanisms are used?
  - Memory protection
  - Processor modes
  - User authentication
  - File access control
  - Logging & Auditing

What Security Goals Does Operating System Provide (Recent Past and Current)?

- More recent past and present: Networked computers: ensure secure operation in networked environment
- New threat?
  - Attackers coming from the network. Network-facing programs on computers may be buggy. Users may be hurt via online communication.
- Additional security mechanisms
  - Secure Communication (using cryptography)
  - Remote authentication
  - Intrusion Prevention and Detection
  - Recovery

What Security Goals Does Operating System Provide (Current and Near Future)?

- Present and near future: mobile computing devices and cloud platforms
- New threat?
  - Apps (programs) may be malicious or questionable.
  - More tightly connected with personal life of the owner.
- Security mechanisms for mobile devices?
  - Isolation of each app.
  - Security scanning of apps in appstores.
  - Help users assess risks of apps.
  - Risk communication.

OPERATING SYSTEM SECURITY MECHANISM

PURDUE UNIVERSITY®

Memory Protection: Access Control to Memory

- Ensures that one user's process cannot access other's memory
  - Implemented using paging in modern operating systems
- Operating system and user processes need to have different privileges
  - This is achieved using hardware support such as CPU modes

CPU Modes (a.k.a. processor modes or privilege modes)

- System mode (privileged mode, master mode, supervisor mode, kernel mode)
  - Can execute any instruction
  - Can access any memory locations, e.g., accessing hardware devices,
  - Can enable and disable interrupts,
  - Can change privileged processor state,
  - Can access memory management units,
  - Can modify registers for various descriptor tables .

Reading: http://en.wikipedia.org/wiki/CPU_modes

PURDUE UNIVERSITY®

User Mode

- User mode
  - Access to memory is limited,
  - Cannot execute some instructions
  - Cannot disable interrupts,
  - Cannot change arbitrary processor state,
  - Cannot access memory management units
- Transition from user mode to system mode can only happen via well defined entry points, i.e., through system calls

System Calls (Guarded Gates)

- Guarded gates from user mode (space, land) into kernel mode (space, land)
  - use a special CPU instruction (often an interruption), transfers control to predefined entry point in more privileged code; allows the more privileged code to specify where it will be entered as well as important processor state at the time of entry.
  - the higher privileged code, by examining processor state set by the less privileged code and/or its stack, determines what is being requested and whether to allow it.

Kernel space vs User space

- Part of the OS runs in the kernel model
  - known as the ***OS kernel***
- Other parts of the OS run in the user mode, including service programs (daemon programs), user applications, etc.
  - they run as ***processes***
  - they form the user space (or the user land)
- When they need privileged access that only kernel can provide, they issue system calls.

Privilege Levels

- Security is often achieved by running control/protection code at a higher privilege level
- Components running at the same level can be isolated by a higher-privilege component
- If attack and defense are at the same level, then it is an arms' race and there can be no guarantee

BASIC CONCEPTS IN ACCESS CONTROL

PURDUE UNIVERSITY®

Readings for This Lecture

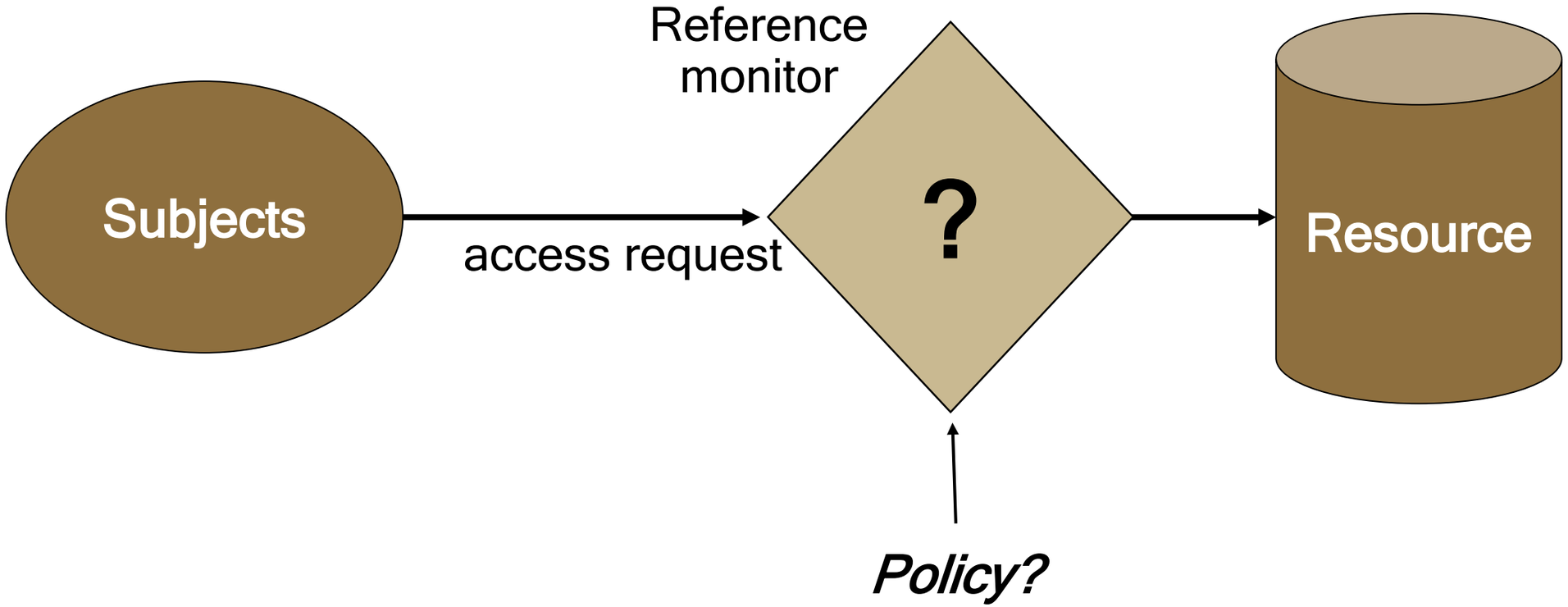
- Wikipedia
  - Filesystem Permissions
- Other readings
- Unix file permissions
  - http://www.unix.com/tips-tutorials/19060-unix-file-permissions.html

Access Control as a General Security Mechanism

- Access Control is a Protection Wall with a Guarded Gate.
  - Build a wall to prevent access.
  - Then design a guarded gate to decide what access should be allowed.

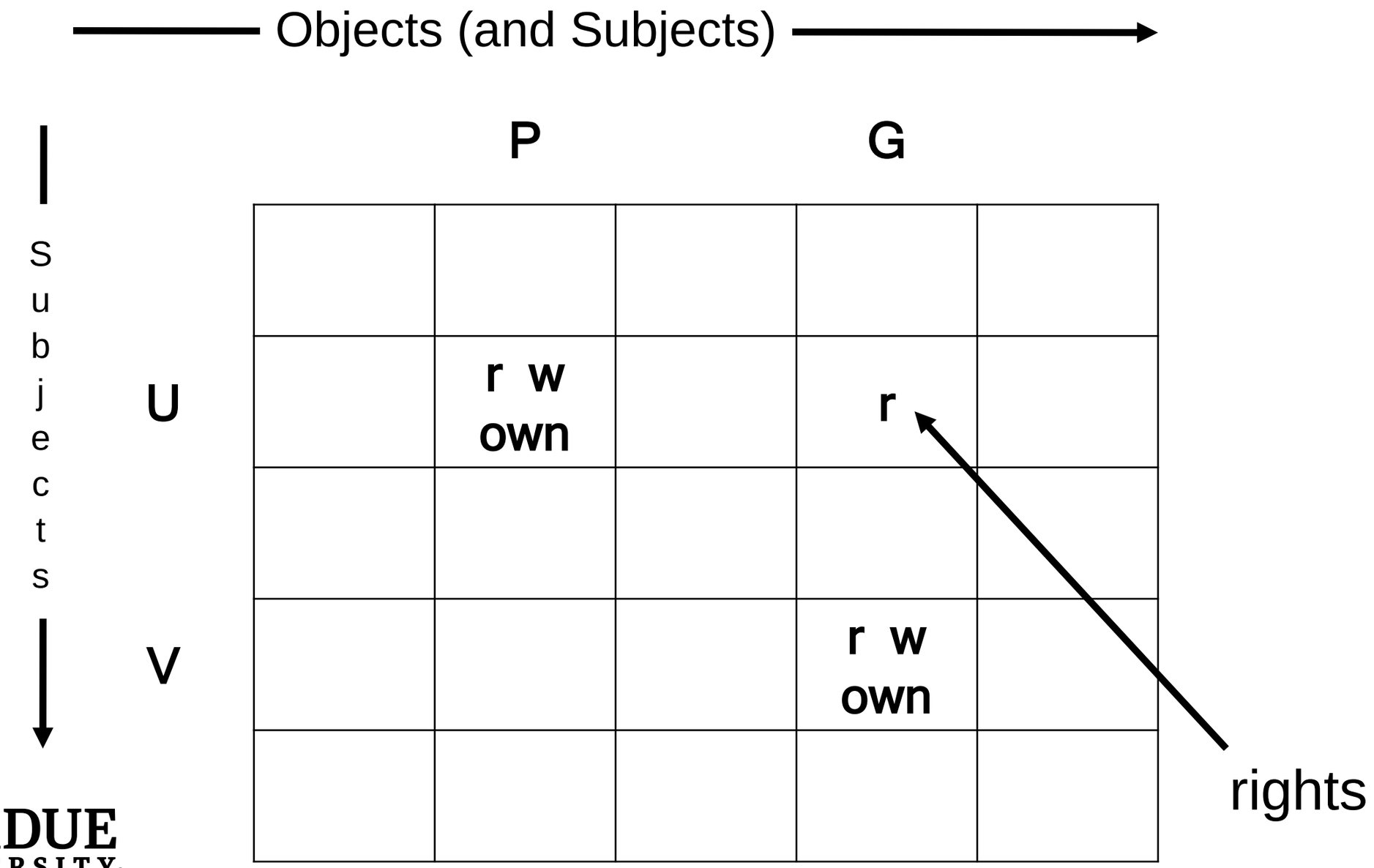
Access Control

- A reference monitor mediates all access to resources
- Complete mediation principle: control all accesses to resources



PURDUE UNIVERSITY®

Access Matrix Model for Policies



PURDUE UNIVERSITY®

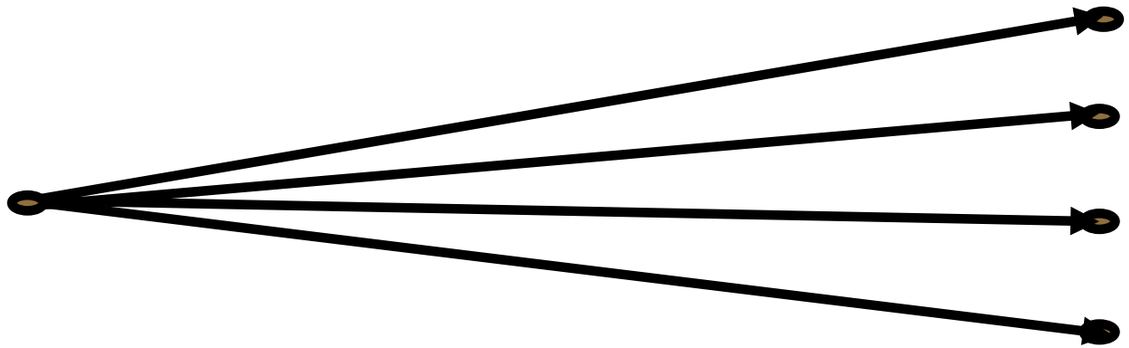
ACCESS MATRIX MODEL

- Basic Abstractions
  - Subjects
    - active entities that request accesses
  - Objects
    - entities on which accesses can be performed
  - Rights
- The rights in a cell specify the access of the subject (row) to the object (column)

Different Levels of Abstractions for Subjects

- Human Users
  - High-level policy objectives are usually regarding human users
- Principals: User Accounts in Unix
  - Policy grants access rights to principals
- Subjects: Processes in Unix
  - Processes run on behalf of principals
  - Processes initiate access requests

USERS AND PRINCIPALS



USERS

Real World User

PRINCIPALS

Unit of Access Control and Authorization

The system authenticates the human user to a particular principal

PURDUE UNIVERSITY®

USERS AND PRINCIPALS

- There should be a one-to-many mapping from users to principals
  - a user may have many principals, but
  - each principal is associated with an unique user
- This help ensures accountability
  - A system can identify which principals performed an action, and needs to uniquely identify the human for accountability

What does the above imply in practice?

Basic Concepts of UNIX Access Control: Users, Groups, Files, Processes

- Each user account has a unique UID
  - The UID 0 means the super user (system admin)
- A user account belongs to multiple groups
  - This is needed to make policy specification more succinct
- Subjects are processes
  - A sophisticated mechanism is used to determine which principal a subject/process is acting on behalf of
- Most objects are modeled as files

OBJECTS IN UNIX

PURDUE UNIVERSITY®

OBJECTS

- An object is anything on which a subject can perform operations (mediated by rights)
- Usually objects are passive, for example:
  - File
  - Directory (or Folder)
  - Memory segment
- But, subjects (i.e. processes) can also be objects, with operations performed on them
  - kill, suspend, resume, send interprocess communication, etc.

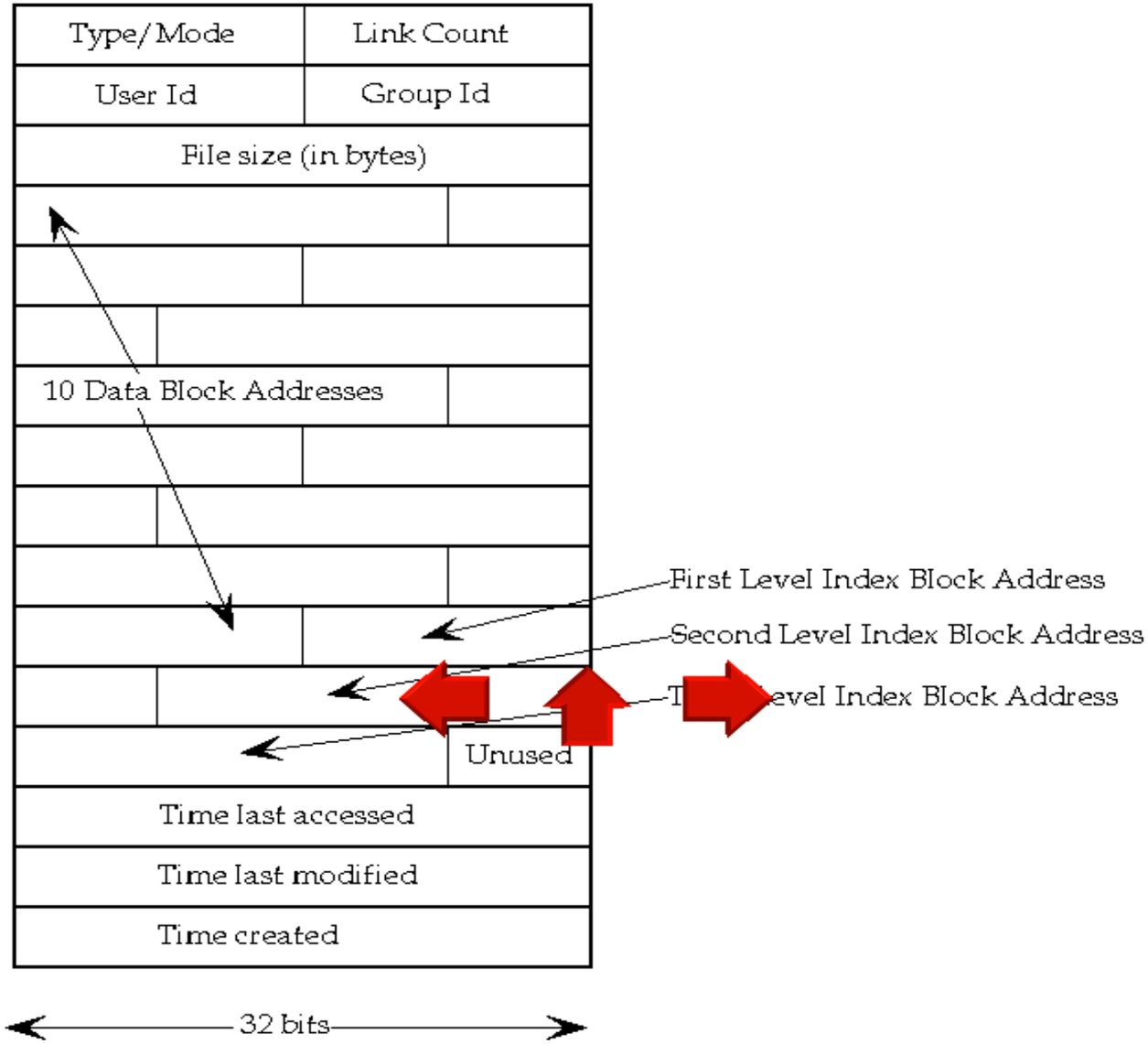
Organization of Objects

- In UNIX, almost all objects are modeled as files
  - Files are arranged in a hierarchy
  - Files exist in directories
  - Directories are also one kind of files
- Each object has
  - owner
  - group
  - 12 permission bits
    - rwx for owner, rwx for group, and rwx for others
    - suid, sgid, sticky

UNIX inodes

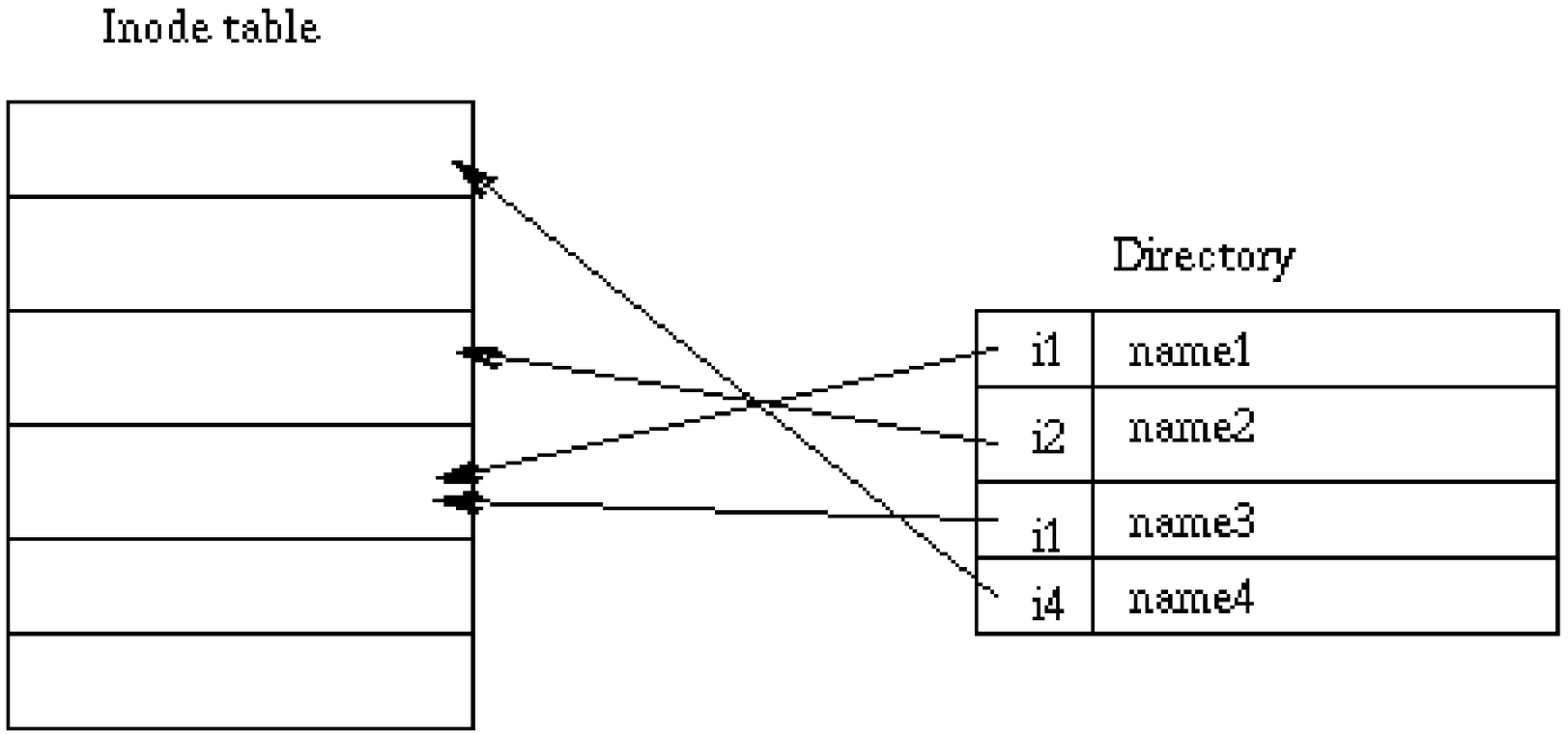
UNIX inodes:

Each file corresponds to an inode



PURDUE UNIVERSITY

UNIX Directories



A directory stores a mapping from names to inode numbers.

PURDUE UNIVERSITY

Basic Permissions Bits on Files (Non-directories)

- Read controls reading the content of a file
  - i.e., the read system call
- Write controls changing the content of a file
  - i.e., the write system call
- Execute controls loading the file in memory and execute
  - i.e., the execve system call

Permission Bits on Directories

- Read bit allows one to show all file names in a directory
- The execution bit controls traversing a directory
  - does a lookup, allows one to find inode # from file name
  - chdir to a directory requires execution bit
- Write + execution control creating/deleting files in the directory
  - Deleting a file under a directory requires no permission on the file
- Accessing a file identified by a path name requires execution to all directories along the path

Some Examples

- What permissions are needed to access a file/directory?
  - read a file: /d1/d2/f3
  - write a file: /d1/d2/f3
  - delete a file: /d1/d2/f3
  - rename a file: from /d1/d2/f3 to /d1/d2/f4
  - …
- File/Directory Access Control is by System Calls
- e.g., open(2), stat(2), read(2), write(2), chmod(2), opendir(2), readdir(2), readlink(2), chdir(2), …

The Three Sets of Permission Bits

- Intuition:
  - if the user is the owner of a file, then the r/w/x bits for owner apply
  - otherwise, if the user belongs to the group the file belongs to, then the r/w/x bits for group apply
  - otherwise, the r/w/x bits for others apply

- What are the other 3 bits? What do they control?

The suid, sgid, sticky bits

	suid	sgid	sticky bit
non-executable files	*no effect*	*affect locking (unimportant for us)*	*not used anymore*
executable files	change euid when executing the file	change egid when executing the file	*not used anymore*
Directories	*no effect*	new files inherit group of the directory	only the owner of a file can delete

PURDUE UNIVERSITY®

Other Issues On Objects in UNIX

- Accesses other than read/write/execute
  - Who can change the permission bits?
    - The owner can
  - Who can change the owner?
    - Only the superuser
- Rights not related to a file
  - Affecting another process
  - Operations such as shutting down the system, mounting a new file system, listening on a low port
    - traditionally reserved for the root user

FROM SUBJECTS TO PRINCIPALS

PURDUE UNIVERSITY®

Subjects vs. Principals

- Access rights are specified for user accounts (principals).
- Accesses are performed by processes (subjects)
- The OS needs to know on which user accounts' behalf a process is executing
- How is this done in Unix?

UNIX Access Control Overview

- Three concepts — Our terminology
  - Human Users — Humans
  - User Accounts — Users/accounts/principals
  - Process — Processes/subjects
- UNIX Access Control System has
  - A discretionary policy specification
    - determines which user accounts have access to which objects
  - A policy enforcement component
    - determines on which user's behalf a subject (process) is acting upon

PURDUE UNIVERSITY®

Process User ID Model in Modern UNIX Systems

- Each process has three user IDs
  - real user ID (ruid) — owner of the process
  - effective user ID (euid) — used in most access control decisions
  - saved user ID (suid)
- and three group IDs
  - real group ID
  - effective group ID
  - saved group ID

PURDUE UNIVERSITY®

Process User ID Model in Modern UNIX Systems

- When a process is created by ***fork***
  - it inherits all three users IDs from its parent process
- When a process executes a file by exec
  - it keeps its three user IDs unless the set-user-ID bit of the file is set, in which case the effective uid and saved uid are assigned the user ID of the owner of the file
- In addition, a process may change the user ids via system calls

The Need for suid/sgid Bits

- Some operations are not modeled as files and require user id = 0
  - halting the system
  - bind/listen on “privileged ports” (TCP/UDP ports below 1024)
  - non-root users need these privileges
- File level access control is not fine-grained enough
- System integrity requires more than controlling who can write, but also how it is written

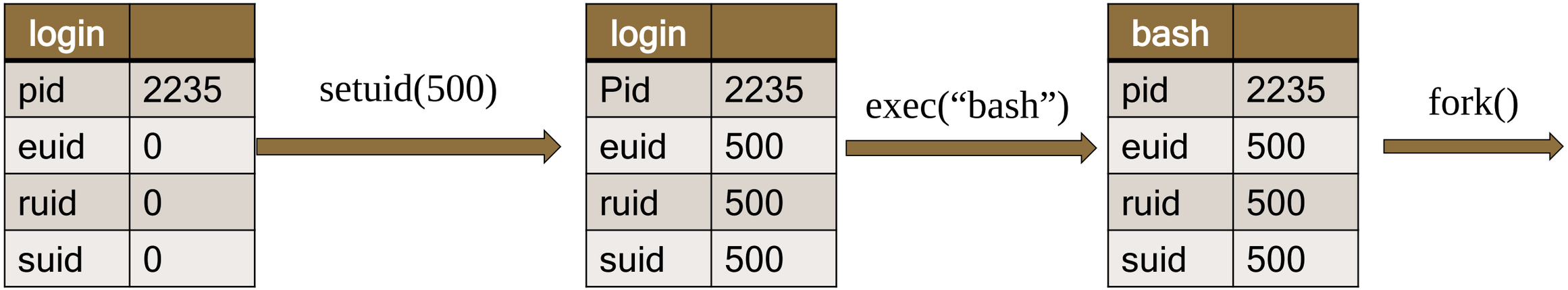
Security Problems of Programs with suid/sgid

- These programs are typically setuid root
- Violates the least privilege principle
  - every program and every user should operate using the least privilege necessary to complete the job
- Why violating least privilege is bad?
- How would an attacker exploit this problem?
- How to solve this problem?

Changing effective user IDs

- A process that executes a set-uid program can drop its privilege; it can
- drop privilege permanently
  - removes the privileged user id from all three user IDs
- drop privilege temporarily
  - removes the privileged user ID from its effective uid but stores it in its saved uid, later the process may restore privilege by restoring privileged user ID in its effective uid

What Happens during Logging in



After the login process verifies that the entered password is correct, it issues a setuid system call.

The login process then loads the shell, giving the user a login shell.

The user types in the passwd command to change his password.

PURDUE UNIVERSITY®

What Happens during Logging in (2)

bash	
pid	2235
euid	500
ruid	500
suid	500

bash	
pid	2297
euid	500
ruid	500
suid	500

exec("passwd")

password	
pid	2297
euid	0
ruid	500
suid	0

Drop privilege permanently

password	
pid	2297
euid	500
ruid	500
suid	500

Drop privilege temporarily

password	
pid	2297
euid	500
ruid	500
suid	0

The fork call creates a new process, which loads "passwd", which is owned by root user, and has setuid bit set.

PURDUE UNIVERSITY

Issues to Consider in Designing an Access Control System

- What are the objects? How are they organized?
- What are the subjects? What are the principals?
- How to relate subjects to principals?
- Whether/how to map human users to principals?
- What kinds of operations subjects can perform on objects?
- Where to store the access control policy data?
- How can access control policy data be updated? How to control the update operation?
- How to intercept access to perform the check? Are all access path covered?
- What are the limitations of the protection, i.e., what does it take to break the protection? How to deal with such residue threats?

Learning Outcomes

- Describe the evolving threats for operating systems, and summarize the main security mechanisms
- Explain the needs for protection modes and system calls for security
- Explain the differences and relationships between users, security principals, and subjects in the context of Unix
- Describe how file system access control work in Unix/Linux
- Explain the mechanism in Unix/Linux to associate security principals with subjects
- Explain potential security problems caused by setuid programs

Coming Attractions …

- How to deal with the threat of malicious and/or buggy software to enforcing access control policies?