MSc - Cybersecurity

CMT310: Developing Secure Systems and Applications

Host Security and Access Control

Dr Neetesh Saxena
saxenan4@cardiff.ac.uk

Outline

- Host Security – security principles
- Access Control – DAC, MAC, RBAC, ABAC
- Backup, Logs
- Integrity protection strategies
- UNIX file permissions

Security Principles

- Defense In Depth
  - placing security mechanisms in place at many different levels in your system architecture
- It means doing security in your network
  - on your operating systems
  - on your applications
  - at all levels where possible.

Cont.

- Risk Analysis
  - identifying things within your organisation that are valuable and should be protected.
- Security Policy
  - a set of rules that you put in place to make sure your systems stay secure.
- Identity Management
  - rules about who gets accounts, what they can access, how long they get to keep the accounts.
- Security Incident
  - What constitutes an incident, responsibilities of each party, who gets notified and when, legal implications, etc.

Cont.

- End-to-End Security
  - in contrast to host security, is security which is implemented at the application level.
  - In this case, even the host itself need not be trusted, since the application itself supplies the security.
    - For example, PGP encrypted e-mail

Host Security

- Host Security
  - refers to securing the operating system, filesystem and the resources of the Host from unauthorized access or modification or destruction.

- This is in contrast to things like:
  - firewalls and VPNs (network security) or Apache or Oracle penetration testing (application security).

Vulnerabilities

- Common Vulnerabilities are the areas where you are most likely to get break-ins.

- These are areas to focus on:
  - Public Services: FTPD, Apache, MySQL, PHP
  - User Accounts: default accounts, weak passwords, automatic logins
  - Old Software: BIND/named
  - Phishing Attacks via E-mail and the Web

*BIND or named (short for name daemon), is the most widely used Domain Name System (DNS) software on the Internet

Host Security Tools

- Processes:
  - ps - non-interactive, view information concerning a selection of running processes.
  - top - interactive, used to show the Linux processes and their activities.
  - acct - manages user actives (processes and information)

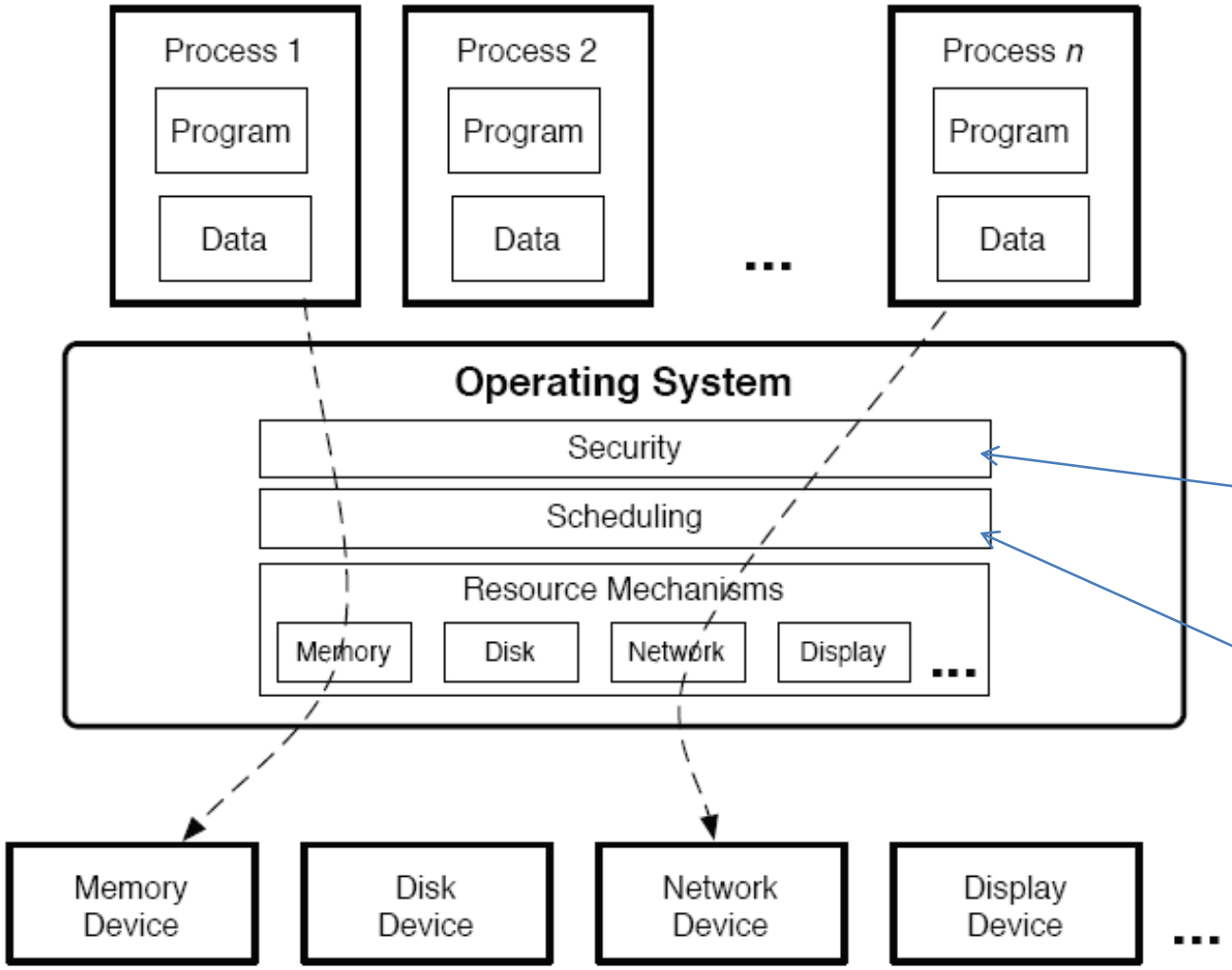
- Open Files: lsof (stands for List Open Files)

- Changed Files:
  - incrond - listen for events that affect a filesystem, such as opening, creating, or deleting a file; accessing a directory; or changing an attribute.
  - debsums - tool for verification of installed package files against MD5 checksums
  - fcheck - monitor directories, files or complete filesystems for any additions, deletions, and modifications.

Cont.

- Network:
  - Netstat - (network statistics), connections for TCP, routing tables, and a number of network interface and network protocol statistics.
  - nmap – (network scanner), used to discover hosts and services on a computer network by sending packets and analyzing the responses.
- Suid: setuid is Unix access rights flags that allow users to run an executable with the permissions of the owner and to change behaviour in directories.
  - find
  - /etc/sudoers: The sudoers file is a file Linux and Unix administrators use to allocate system rights to system users.
- Logins:
  - last – command reviews recent logins.
  - lastcomm - command prints the information of previously executed commands of user.
  - sa - command summarizes information of previously executed commands.
  - acct - monitoring users activities.
- Updates:
  - apt - installation of new software packages, upgrade of existing software packages, updating of the package list index.

Working of Computer System



Internet-scale sharing mechanisms - users may share anything with anyone in the world.

Access to resources should be controlled, such that one process cannot inadvertently or maliciously impact the execution of another.

Switch among the processes fairly

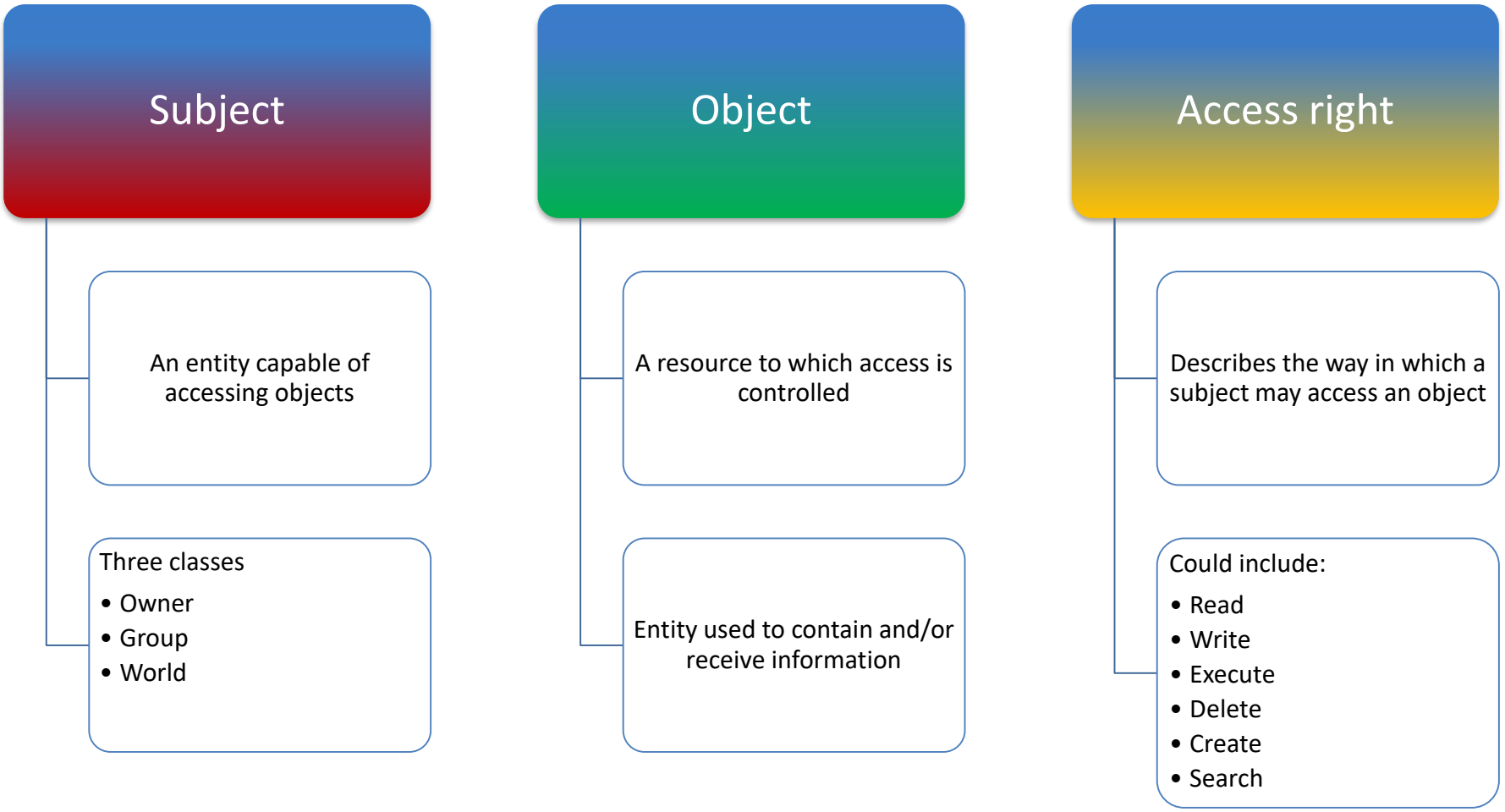
An operating system runs *security*, *scheduling*, and *resource mechanisms* to provide *processes* with access to the computer system's resources (e.g., CPU, memory, and devices).

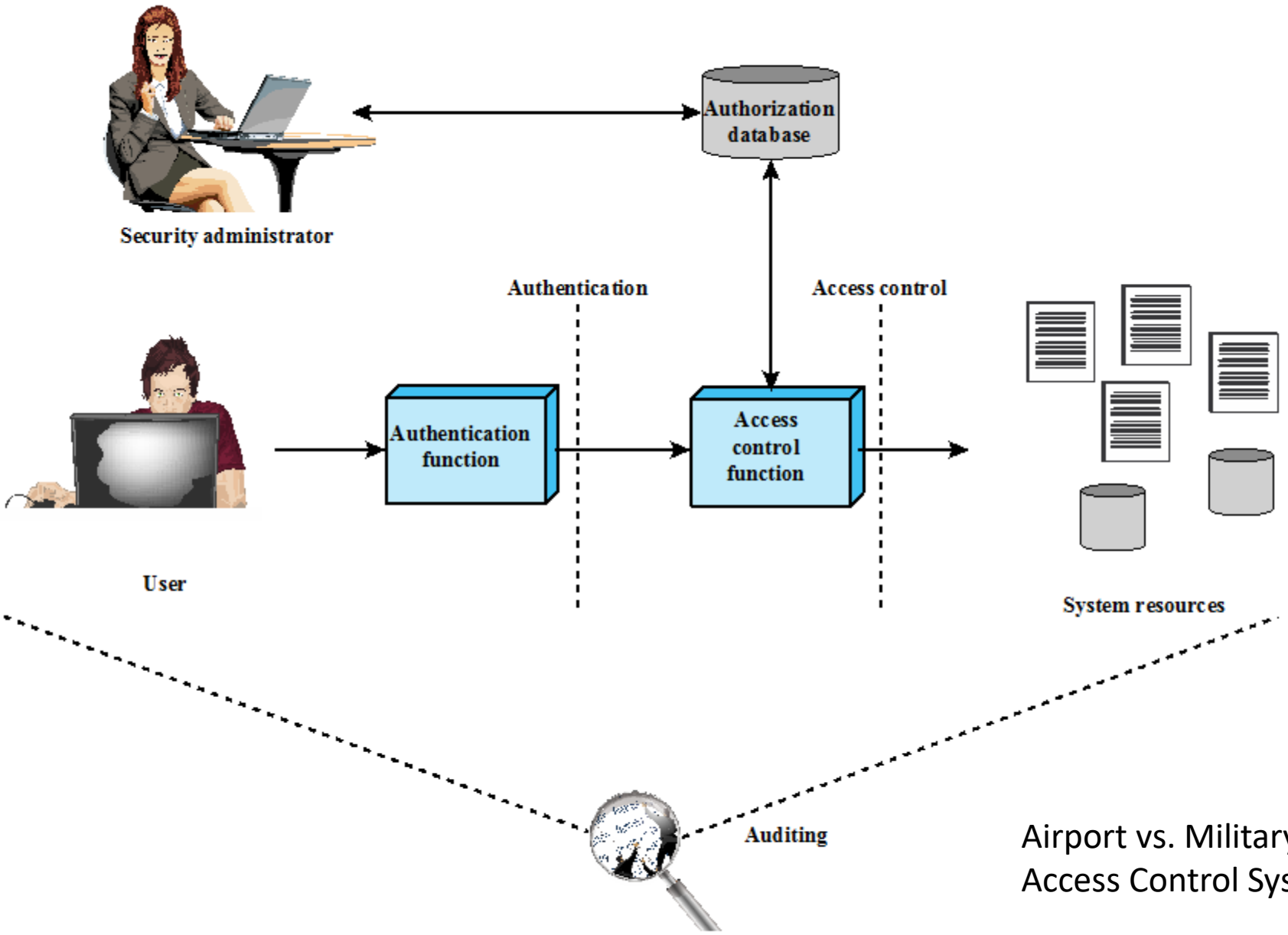
Access Control Principles

RFC 4949 defines computer security as:

"Measures that implement and assure security services in a computer system, particularly those that assure access control service."

Subjects, Objects, and Access Rights





Relationship Among Access Control and Other Security Functions

Airport vs. Military Access Control System

Access Control

- An access enforcement mechanism authorizes requests from multiple subjects (e.g. users, processes, etc.) to perform operations (e.g., read, write, etc.) on objects (e.g., files, sockets, etc.).

- An operating system provides an access enforcement mechanism.

- Two fundamental concepts of access control:
  - a *protection system* that defines the access control specification and
  - a *reference monitor* that is the system's access enforcement mechanism that enforces this specification.

Protection System

- A protection system consists of a protection state, which describes the operations that system subjects can perform on system objects, and a set of protection state operations, which enable modification of that state.

- The access matrix is used to define the *protection domain* of a process.

	File 1	File 2	File 3
Process 1	Read	Read, Write	Read, Write
Process 2	-	Read	Read, Write

Lampson's Access Matrix

Reference Monitor

- A reference monitor is the classical access enforcement mechanism.
- It verifies the nature of the request against a table of allowable access types for each process on the system.
- 3 guarantees:
  - Completeness: must mediate all access
  - Isolation: must be protected from modification
  - Verifiability: must be verifiable for correctness

- Reference monitor: an abstract concept for access control
- Security kernel: implementation of the reference monitor
- Trusted computing base (TCB): kernel + other protection mechanisms

Access Control Policies

- **Discretionary access control (DAC)**
  - Controls access based on the identity of the requestor and on access rules (authorizations) stating what requestors are (or are not) allowed to do.
- **Mandatory access control (MAC)**
  - Controls access based on comparing security labels with security clearances.
- **Role-based access control (RBAC)**
  - Controls access based on the roles that users have within the system and on rules stating what accesses are allowed to users in given roles.
- **Attribute-based access control (ABAC)**
  - Controls access based on attributes of the user, the resource to be accessed, and current environmental conditions.

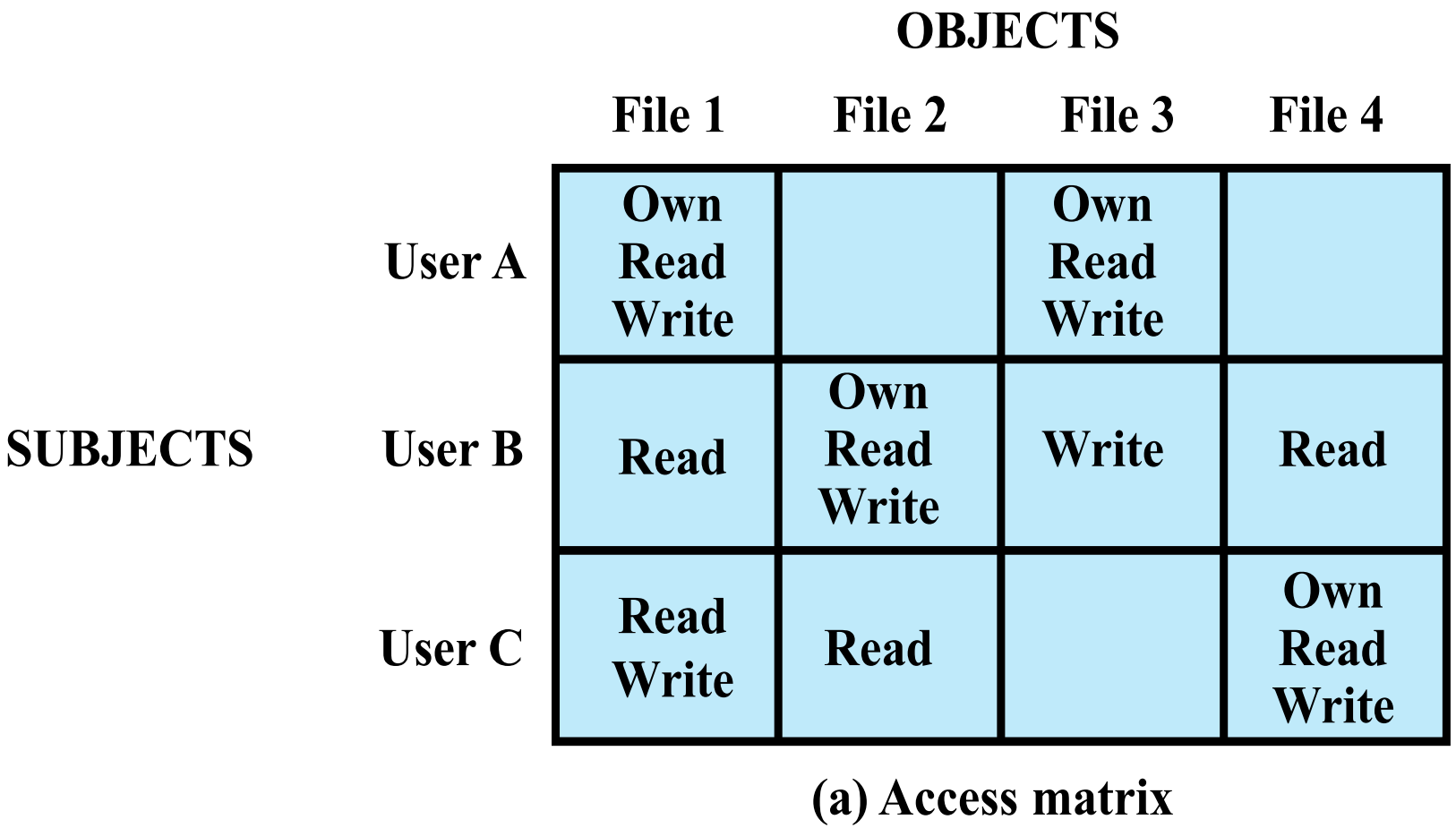
Discretionary Access Control (DAC)

- Scheme in which an entity may enable another entity to access some resource.

- Often provided using an access matrix
  - One dimension consists of identified subjects that may attempt data access to the resources
  - The other dimension lists the objects that may be accessed.

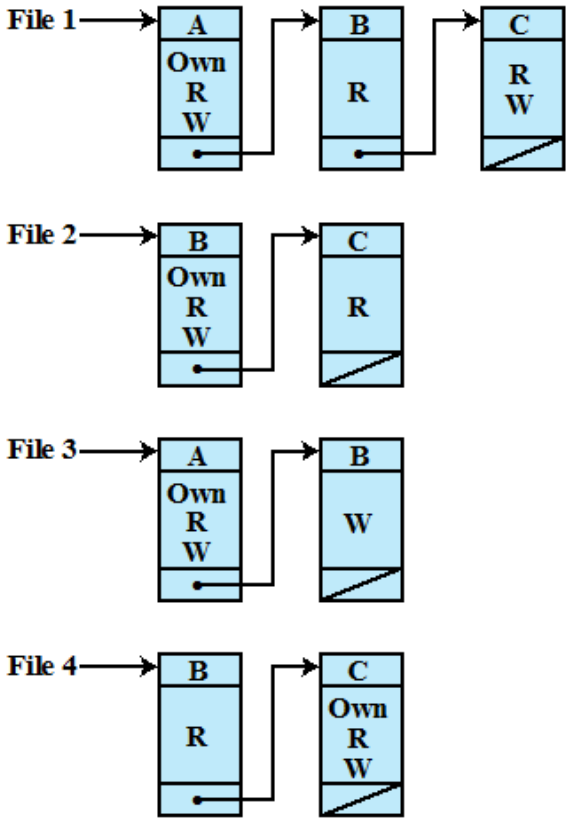
- Each entry in the matrix indicates the access rights of a particular subject for a particular object.

Cont.

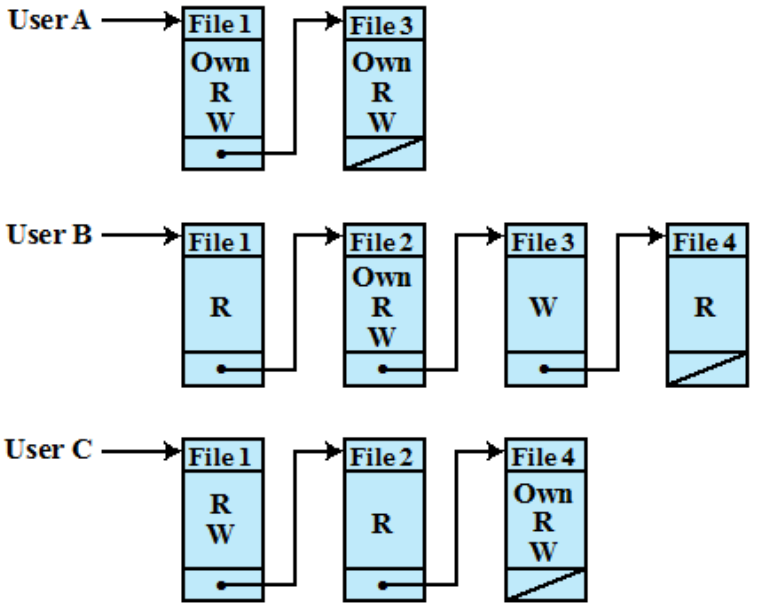


(a) Access matrix

Cont.



(b) Access control lists for files of part (a)



(c) Capability lists for files of part (a)

Example of Access Control Structures

The standard access control lists are ranged from 1 to 99 and from 1300 to 1999. In devices, ACL is based on source address and wildcard mask.

Read Wildcard mask here: https://www.cbtnuggets.com/blog/technology/networking/networking-basics-what-are-wildcard-masks-and-how-do-they-work.

Authorization Table for Files

Subject	Access Mode	Object
A	Own	File 1
A	Read	File 1
A	Write	File 1
A	Own	File 3
A	Read	File 3
A	Write	File 3
B	Read	File 1
B	Own	File 2
B	Read	File 2
B	Write	File 2
B	Write	File 3
B	Read	File 4
C	Read	File 1
C	Write	File 1
C	Read	File 2
C	Own	File 4
C	Read	File 4
C	Write	File 4

Mandatory Access Control

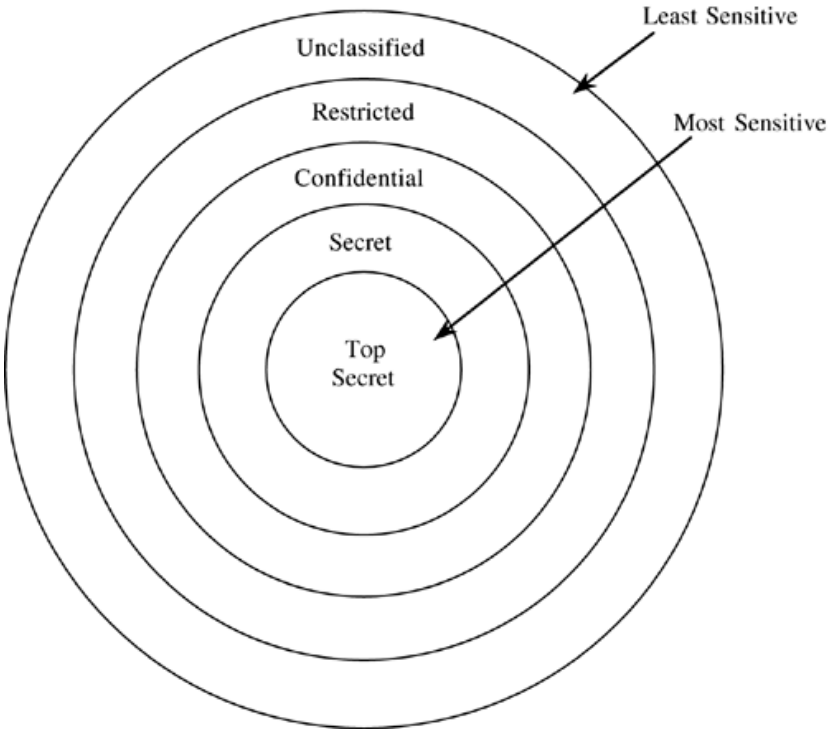
- Owner of the resource does not decide who gets to access it
  - but instead access is decided by a group or individual who has the authority to set access on resources.

- Implemented in government organizations
  - access to a given resource is largely dictated by the sensitivity label applied to it (secret, top secret, etc.),
  - by the level of sensitive information an individual is allowed to have access.

- Traditional MAC mechanisms have been tightly coupled to a few security models.
  - systems supporting flexible security models start to appear (e.g., SELinux, Trusted Solaris, TrustedBSD, etc.)
  - e.g., MAC Implementation in Windows Vista

Military Security Model

- Information is ranked:
  - Unclassified
  - Confidential
  - Secret
  - Top Secret



- Least Privilege: Subject should have access to fewest objects needed for successful work

Examples

- DAC - works by checking the properties of a file on my MSWindows machine, the file has attributes and the owner of the file can do almost anything with it like making it available for everyone to read, transfer the ownership to an other user or even delete it.

- MAC - getting a Windows 8 machine and trying to modify files within a windows 8 store programs installation directory (under the hidden directory c:\program files\windowsapps). Even as an administrative user you will be prevented from changing these files via standard OS tools even after you have "taken ownership" of the file.

MAC vs. DAC

Mandatory Access Control (MAC)

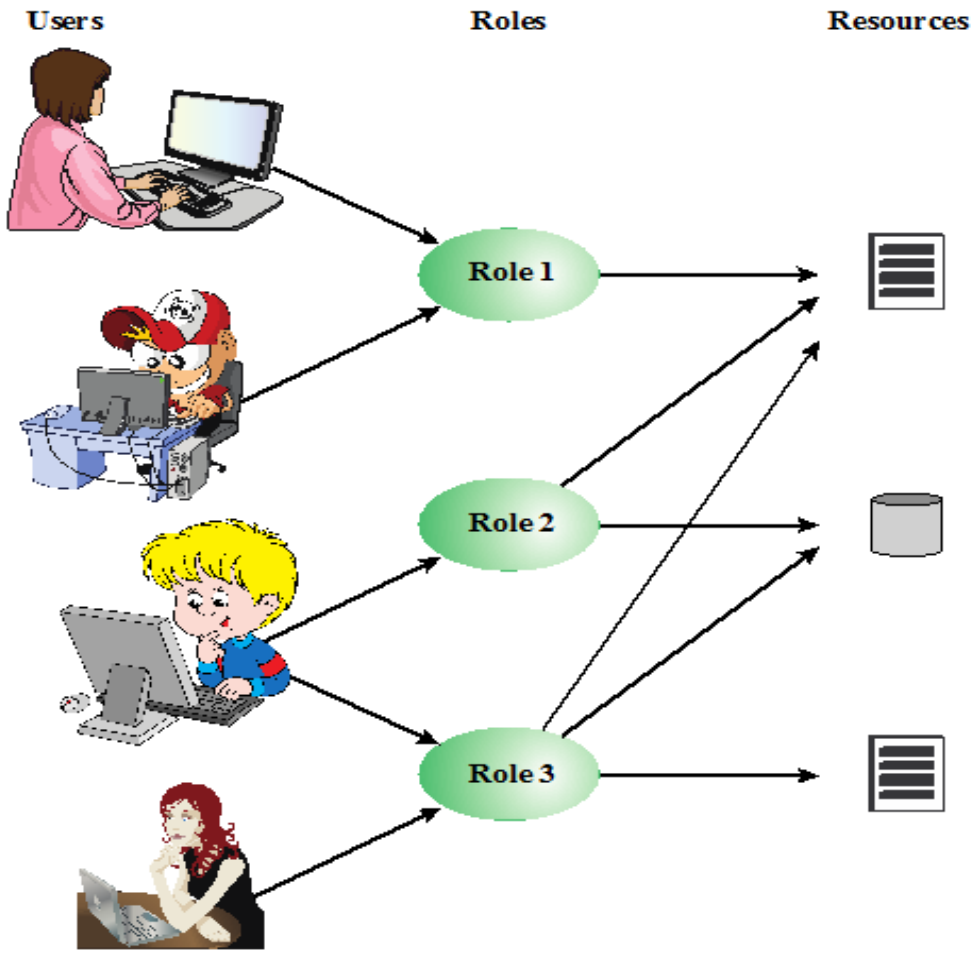
- Access not controlled by owner of object
- Example: User does not decide who holds a **TOP SECRET** clearance

Discretionary Access Control (DAC)

- Owner of object determines access
- Example: UNIX/Windows file protection

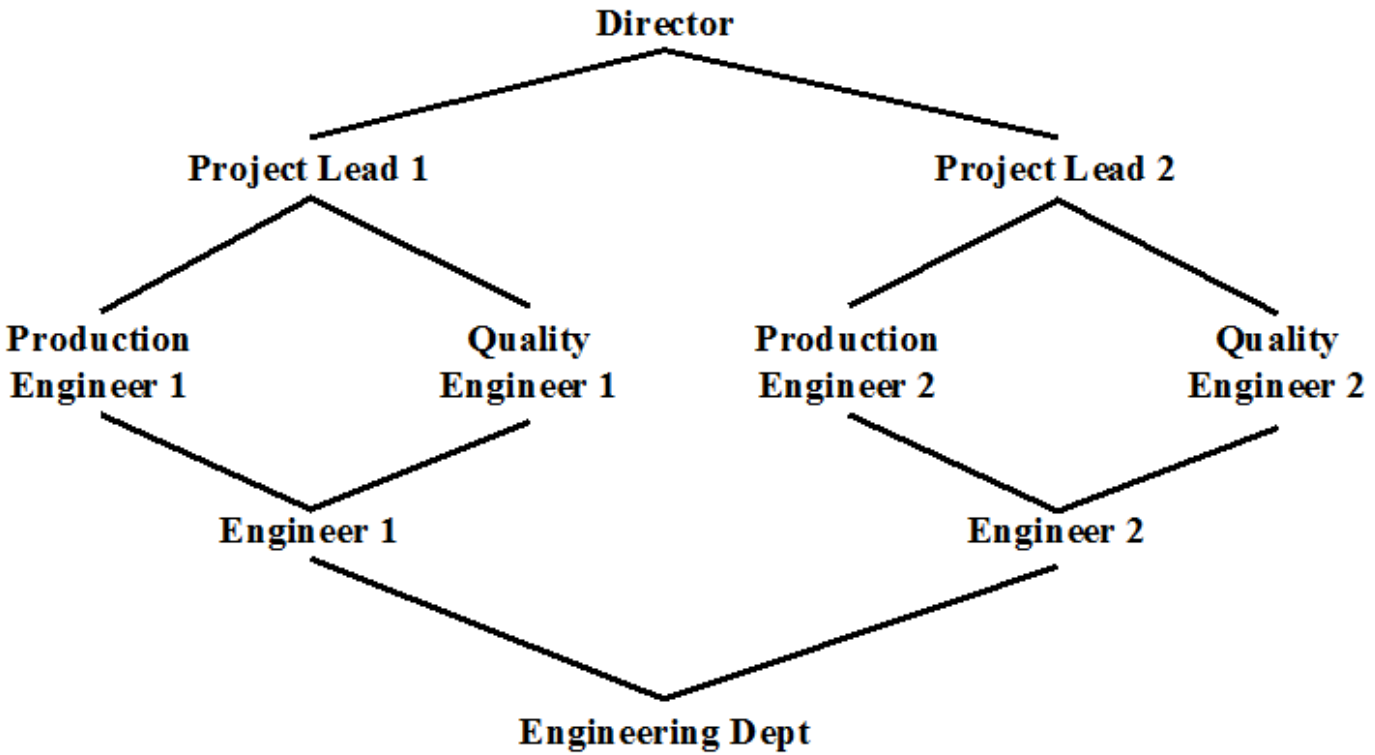
If DAC and MAC both apply, MAC wins

Role Based Access Control (RBAC)

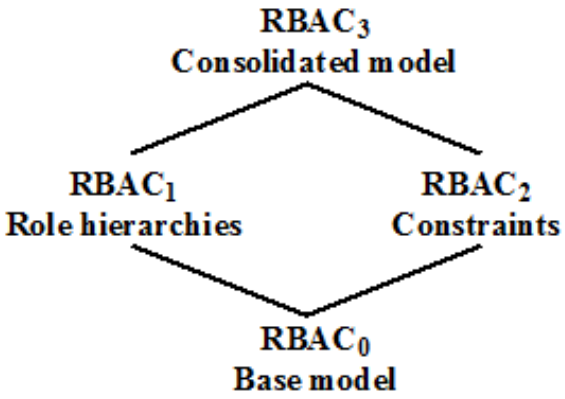


Users, Roles, and Resources

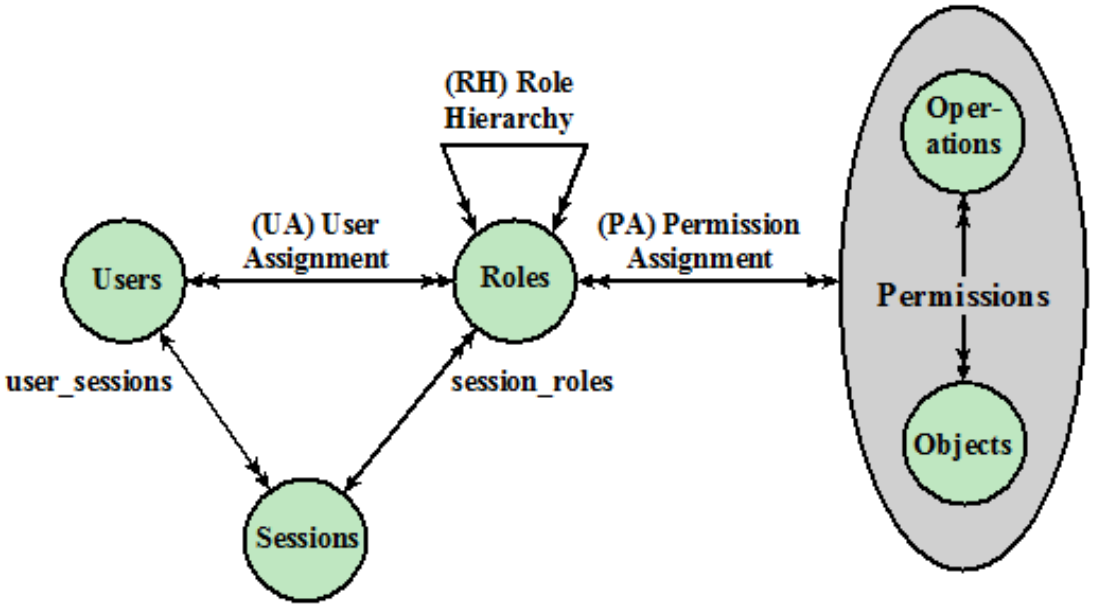
Cont.



Example of Role Hierarchy



(a) Relationship among RBAC models



(b) RBAC models

A Family of Role-Based Access Control Models.

Constraints - RBAC

- Provide a means of adapting RBAC to the specifics of administrative and security policies of an organization.
- A defined relationship among roles or a condition related to roles.
- Types:

Mutually exclusive roles

- A user can only be assigned to one role in the set (either during a session or statically)
- Any permission (access right) can be granted to only one role in the set

Cardinality

- Setting a maximum number with respect to roles

Prerequisite roles

- Dictates that a user can only be assigned to a particular role if it is already assigned to some other specified role

Attribute-Based Access Control (ABAC)

Subject attributes

- A subject is an active entity that causes information to flow among objects or changes the system state
- Attributes define the identity and characteristics of the subject

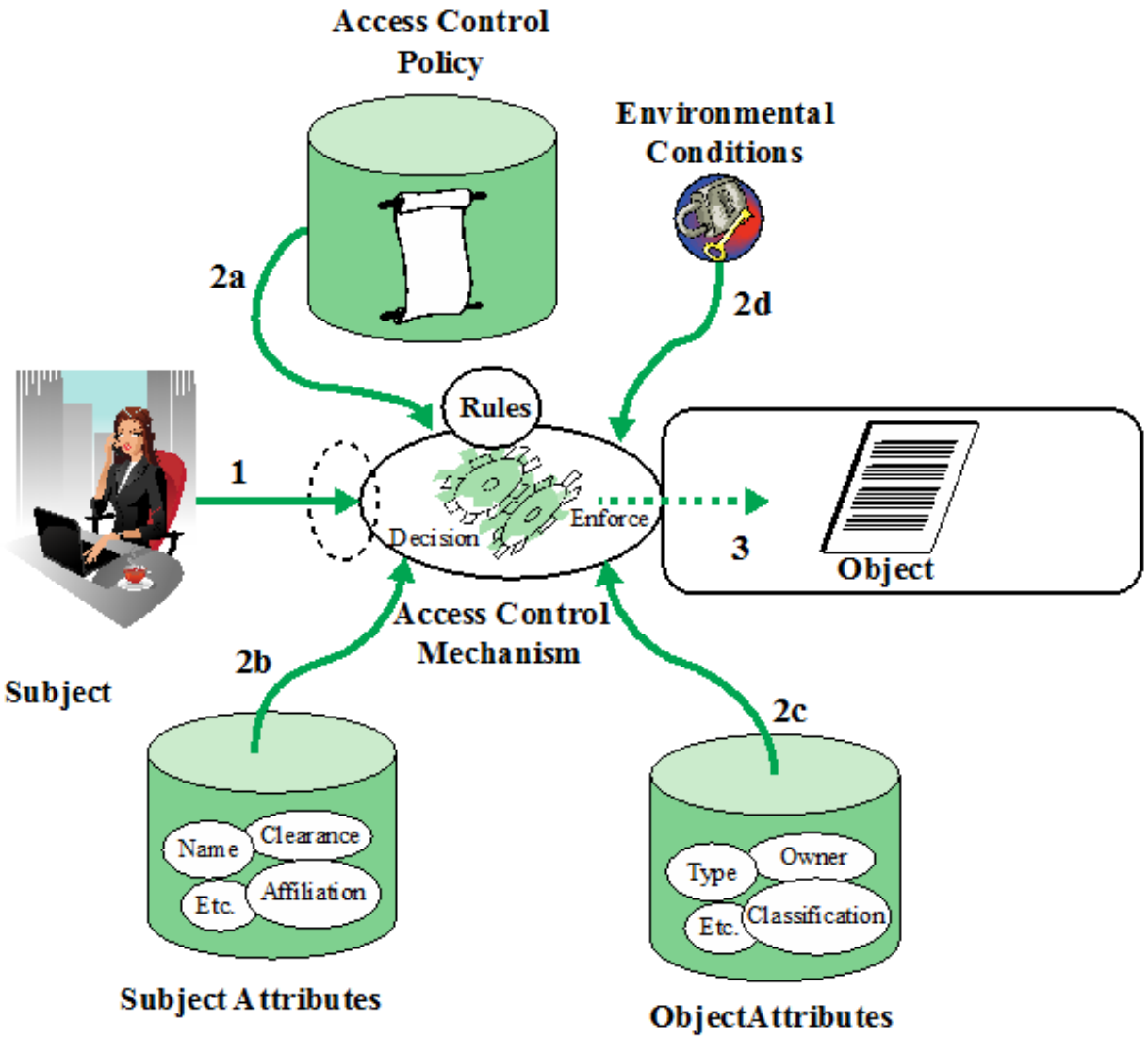
Object attributes

- An object (or resource) is a passive information system-related entity containing or receiving information
- Objects have attributes that can be leverages to make access control decisions

Environment attributes

- Describe the operational, technical, and even situational environment or context in which the information access occurs
- These attributes have so far been largely ignored in most access control policies

Current time and location -> type of communication channel, such as protocol or encryption strength, or client type (PC, smart phone, etc.)



Simple ABAC Scenario

Protection Domains

- In terms of the access matrix, a **row** defines a protection domain.

- In **user mode**, certain areas of the memory are **protected from use** and certain instructions may not be executed.

- In **kernel mode**, privileged instructions may be executed and protected areas of the memory may be accessed.
  - e.g., when the user process calls a system routine, that routine executes in a system mode.

UNIX File Access Control

UNIX files are administered using inodes (index nodes)

- File attributes, permissions and control information are sorted in the inode.
- Several file names may be associated with a single **inode**.
- An **inode table** or **inode list** contains inodes of all the files in the file system.
- When a file is opened, its inode is brought into main memory and stored in a memory resident inode table.

Directories are structured in a hierarchical tree

- May contain files and/or other directories.
- Contains **file names** plus **pointers** to associated inodes.

Users and Accounts

Users and groups

- UNIX user -> username -> uid
- mapping is in /etc/passwd
- User - one or more groups
- Group - zero or more users

Dangerous Accounts

- root – superuser
- open, guest, play, nobody - courtesy accounts
- Accounts without passwords
- Group accounts

Bell-LaPadula Properties

Simple Security Property:

- Subject may have read access only if object classified at same level or lower.
- Subject may have write access only if all objects read are at same level or higher than object to be written.

Secure Operating Systems

- Basic Features of a Multiprogramming OS
  - Authentication of users.
  - Protection of memory.
  - File and I/O device access control.
  - Allocation and access control to general objects.
  - Enforcement of sharing.
  - Interprocess communication and synchronization.

Backup Policy

- One backup volume per partition
- Maintain physical security on backups
- Maintain logical security on backups
- Be careful about legal issues on backups

Integrity

- Integrity threats:
  - Change permissions to allow modification/reading
  - Change password file
  - Change device / interface configurations
  - Move files
  - Replace log files with sanitized versions
- 95% of UNIX security incidents result of misconfiguration
- Integrity Protection Strategies
  - Prevention
  - Detection
  - Recovery

Prevention Strategies

- Software Controls:
  - File permissions
  - Directory permissions
  - Restrictions on root access
- Low-level operating system controls:
  - Immutability (only change in single-user mode)
  - append (only add to file, except single-user mode)
- Hardware controls:
  - Read-only file systems (CD ROM)
  - Write-protect options

Detection Strategies

- Comparison copies:
  - On read-only media
  - On remote storage
  - Large space, slow, expensive
- Metadata:
  - Stored list of files
  - Path to files
  - Modification times
  - Easy to fool
- Digital Signature
  - Encrypt with private key of modifier
  - Fast, small, hard to fool, requires extra work

Recovery Strategies

- Restore from backup - Rollback (Data Loss)
- If data problem, may be able to replay changes - Selective Rollback (some data loss)
- If redundant file system, vote file versions - Masking
- If specific changes found - correct - Roll forward

Auditing

- Need to monitor systems
- Monitoring methods: Audits and Logs
  - Audit - active scanning of current state of system
  - Log - record of actions taken in operation of system
- Audits often use logs, and do more

Log File Vulnerabilities

- Alteration
  - Append mode
- Deletion
  - Non-rewritable media
  - to restricted log host
- Flooding
  - Ensure large storage
  - Reduce before logging (look for repeating patterns)

Syslog

- General purpose logging utility
- Any program can generate syslog messages
  - Socket connect to syslogd process TCP port
- Messages to files, devices or computers
  - Dependent on severity and service
- Messages marked with authentication level
  - kern, user, mail, auth, demon, uucp
- Messages marked with priority
  - emerg, alert, err, warning, notice, info, debug

UNIX Command - chmod

#	Permission	rwx
7	read, write and execute	rwx
6	read and write	rw-
5	read and execute	r-x
4	read only	r--
3	write and execute	-wx
2	write only	-w-
1	execute only	--x
0	none	---

Mode	Name	Description
r	read	**r**ead a file or list a directory's contents
w	write	**w**rite to a file or directory
x	execute	e**x**ecute a file or recurse a directory tree

Operator	Description
+	adds the specified modes to the specified classes
-	removes the specified modes from the specified classes
=	the modes specified are to be made the exact modes for the specified classes

Subject	Class	Description
u	owner	file's owner
g	group	users who are members of the file's group
o	others	users who are neither the file's owner nor members of the file's group
a	all	all three of the above, same as ugo

UNIX File Permissions Notation

Textual Representation

- It consists of 10 characters.
- The first character shows the file type.
- Next 9 characters are permissions, consisting of 3 groups: owner, group, others.
- Each group consists of three symbols: **rwx** (in this order),
- if some permission is denied, then a dash "-" is used instead.

Example: **-rwxr--r–**

- Symbol in the position 0 ("-") is the type of the file. It is either "d" if the item is a directory, or "-" if the item is a regular file.
- Symbols in positions 1 to 3 ("rwx") are permissions for the owner of the file.
- Symbols in positions 4 to 6 ("r--") are permissions for the group.
- Symbols in positions 7 to 9 ("r--") are permissions for others.

r	Read access is allowed
w	Write access is allowed
x	Execute access is allowed
-	Replaces "r", "w" or "x" if according access type is denied

-rwxr-xr-x	File, owner has read, write, execute permissions, group: only read and execute permissions, others: only read and execute permissions.
dr-x------	Directory, owner has read and execute access, group and others have no access

http://www.zzee.com/solutions/unix-permissions.shtml

UNIX File Permissions Notation

Numeric (Octal) Representation

- If a numeric representation is used (like in chmod), then
  - it is in the octal format (with the base of 8), and digits involved are 0 to 7.
  - every octal digit combines read, write and execute permissions together.
- Respective access rights for owner, group and others (in this order) are the last three digits of the numeric file permissions representation.

Example: "644".

- Here the second digit ("6") stands for rights of the owner,
  - the third digit ("4") stands for rights of the group,
  - the fourth digit ("4") stands for rights of others.

Octal digit	Text equivalent	Binary value	Meaning
0	---	000	All types of access are denied
1	--x	001	Execute access is allowed only
2	-w-	010	Write access is allowed only
3	-wx	011	Write and execute access are allowed
4	r--	100	Read access is allowed only
5	r-x	101	Read and execute access are allowed
6	rw-	110	Read and write access are allowed
7	rwx	111	Everything is allowed

644	owner: read and write permissions, group: only read permissions, others: only read permissions.
755	owner: read, write and execute permissions, group: read and execute permissions, others: read and execute permissions.

http://www.zzee.com/solutions/unix-permissions.shtml

Example

chmod usage example

Using chmod to change myfile.txt's permissions

```
$ chmod 777 myfile.txt
```

"-rwxrwxrwx" being "myfile.txt's" new permissions, which permit the following:

```
Owner: rwx - 7
Group: rwx - 7
Other: rwx - 7
```

chmod usage example

Using chmod to change myfile.txt's permissions

```
$ chmod 142 myfile.txt
```

"---xr---w-" being "myfile.txt's" new permissions, which permit the following:

```
Owner: --x - 1
Group: r-- - 4
Other: -w- - 2
```