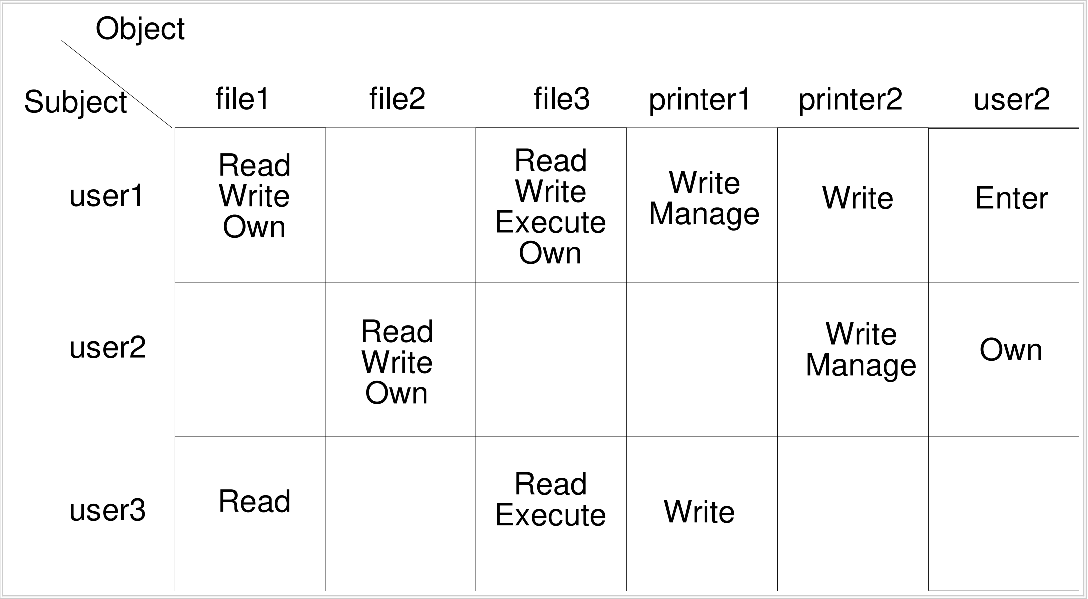
INFORMATION SECURITY

Authorization and Access Control (2)
- Definitions (3)
- Accessing (acting on) an object (7)
- Securing (or Protecting) a system (10)
  - Protection Domain (10)
  - Protection Matrix (12)
  - Authorization & control mechanisms (13)
  - Access Control Lists, ACL (14)
  - Access Capabilities (15)
  - Hybrid Method (16)
- Other facets of access control mechanisms (18)

Authorization and Access Control



Object Subject	file1	file2	file3	printer1	printer2	user2
user1	Read Write Own		Read Write Execute Own	Write Manage	Write	Enter
user2		Read Write Own			Write Manage	Own
user3	Read		Read Execute	Write		

Definitions

- Authorization and access control: security mechanisms that (try to) enforce system's access policy
  - will be distinguished shortly

Entity (subject)

- physical person, “active” object or type of task (*role*)

Object

- resource that undergoes actions[1] from subjects

Action

- utilization, manipulation, transformation...

[1] or accesses

...Definitions (cont.)

Right, permission, attribute or access mode, ...

- related to operations/actions performed by subjects on objects
- confusing terminology that varies with computer system or theoretical model
  - Ex.: in Unix, (access) *permissions* to files – reading writing, etc. - are part of the files' *attributes*. Other attributes are: ownership, size, creation date…
- we will try to separate:
  - right, permission
    - capacity (of subject) to perform some action on resource (object)
  - attribute, access mode
    - capacity (of object) to sustain some action from subject
  - separation of terms is fuzzy, as they are intertwined[1]
  - sometimes we will intermix the use of the terms, if confusion is unlikely

[1] if an user has the *right to read* a file, implicitly, the file has the *read access mode set*!

...Definitions (cont.)

Examples of access rights to an object

- read: be able to know (the content of) the object
- create: be able to make new objects of a certain type
- execute: be able to use (activate or invoke) the object
- modify permissions: be able to change the access rights of the object[1]

Examples of subjects to whom the access rights apply

- user: ordinary worker of system
- administrator: controller or installer of the system
- auditor: verifier or analyzer of the system

[1] here, the Principle of Attenuation of Privilege should apply: «*A subject can only give rights it possesses*.»

...Definitions (cont.)

Authorization

- concession to authenticated entities of rights to objects

Access Control

- verification of rights upon usage of resource[1]

Nomenclature's disclaimer

- in practice, the distinction between authorization and access control is seldom made: both name the process of control of authenticated entities' actions
- also, many times authorization and control are run in simultaneity

[1] sometimes, the controller is named *reference monitor*

Accessing (acting on) an object

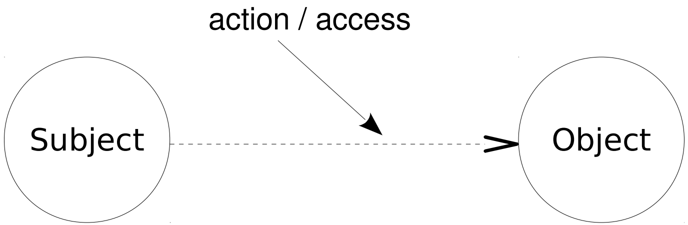


Fig. Acting on an unprotected object.

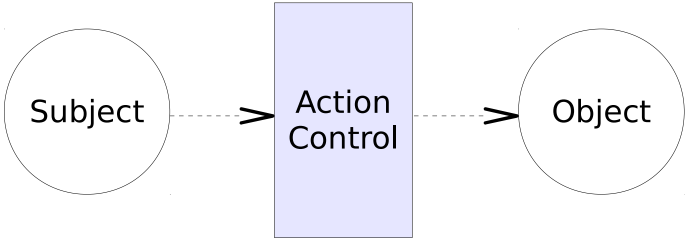


Fig. Acting on a protected object.

Correct sequence of operations

- authentication of entity
- retrieval of authorization for accessing the object
- control of the subject's access to the object
  - the object can be accessed directly or through a resource's server

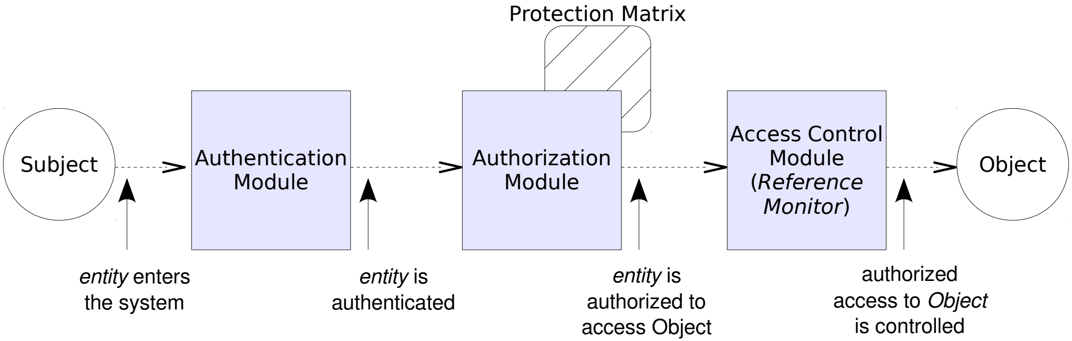

Protection Matrix
Subject
Authentication Module
Authorization Module
Access Control Module (*Reference Monitor*)
Object
entity enters the system
entity is authenticated
entity is authorized to access Object
authorized access to *Object* is controlled

Fig. Recommended procedure for a *Subject* to access an *Object*.

...Accessing (acting on) an object (cont.)

More typical procedure[1]

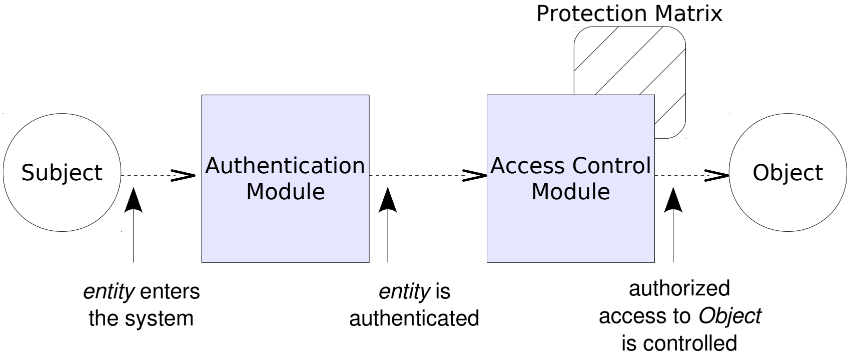

Protection Matrix
Subject
Authentication Module
Access Control Module
Object
entity enters the system
entity is authenticated
authorized access to *Object* is controlled

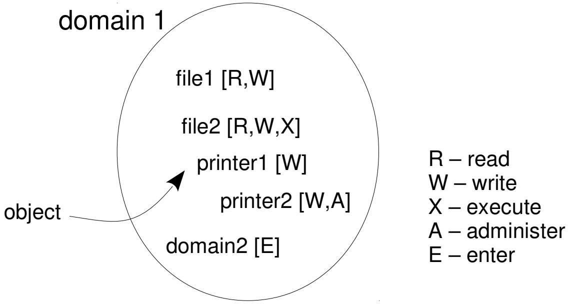
Fig. Typical procedure for a *Subject* to access an *Object*.

[1] As said before, in practice, the distinction between *authorization* and *access control* is seldom made.

Securing (or Protecting) a system

- needs complete knowledge of subjects and objects of system:
  - users’ identification, current and possible activities
  - objects’ identification, their possible operations and who can operate them
- all the time!...

Protection Domain



- data structure[1] associated with a **subject** specifying the objects the subject can access and how (operations allowed)
- set of pairs (object, rights) relating to a **subject**
- can be associated with any type of **subject** (user, process, procedure...)

[1] more correctly, domain is an "abstraction"; it can be implemented by a "data structure".

...Securing (or Protecting) a system (cont.)

Examples of protection domains from Unix

- represented by active users:
  - pairs (UID, GID)[1] ↔ objects and rights associated with (UID, GID)
- represented by type of users:
  - groups GID ↔ objects and rights associated with GID
- represented by execution mode:
  - kernel level procedures ↔ can do and access practically everything
  - user level procedures ↔ can do and access some things

[1] UID/GID: User/Group IDentifier

Protection Matrix

- table representing access information for every domain and object in the system
  - each table row lists the rights of a domain over an object
  - each column lists the access information of each domain over the object

Object Domain	file1	file2	file3	print1	print2	dom2
1	Read Write Own		Read Write Execute Own	Write	Write	Enter
2		Read Write Own			Write	Own
3	Read		Read Execute	Write		

Fig. Example of a Protection matrix.

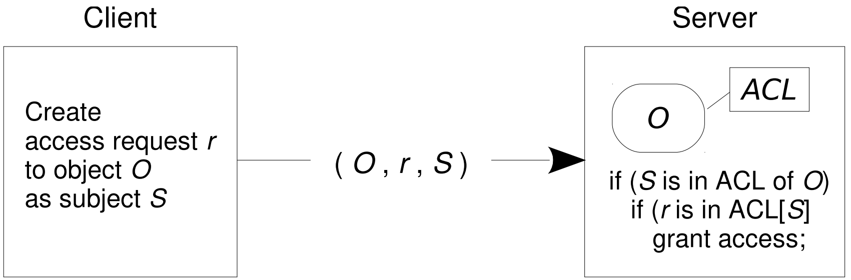
Authorization & control mechanisms

- access control lists (ACL)
  - object “knows” which users will be able to access it and how
- capability (or permission) lists[1]
  - user knows which object will be able to access and how
- hybrid methods
  - some part of the system use ACL, other use capability lists
  - access has two phases, each controlled by one of the techniques

[1] PT: listas de permissões (ou de credenciais) de acesso

Access Control Lists, ACL

- each object (or its manager) keeps a list of the related protection domains' information;
- the protection matrix is stored column-wise.



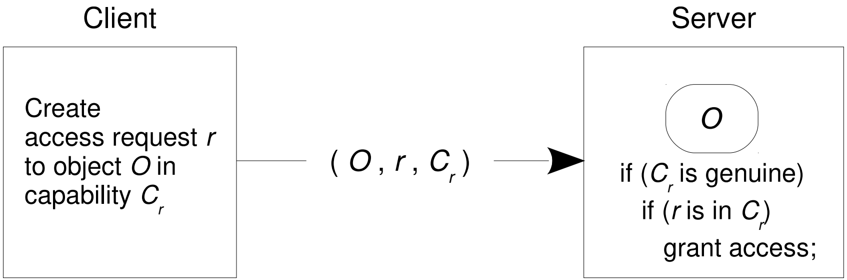
Example (*see previous Protection Matrix****)***

- ACL (`file1`) - `Dom1`: Read, Write, Own; `Dom3`: Read

Problem: when an entity wants to know which objects can access and with which rights...

Access Capabilities

- each subject keeps own protection domain data structure (*capabilities list*, *C-list*);
- each pair (object, rights) is a *capability*;
- the matrix is stored row-wise.



Example (*see previous Protection Matrix****)***

- *C-list* (`Dom2`) – `file2`: Read, Write, Own; `print2`: Write; `Dom2`: Own

Problem: when an entity wants to change the access rights to an object...

Hybrid Method

- use access control lists before the opening of a session (of utilization of the object) and after the closing of the session
  - access permissions to objects are easily changed
- use capabilities during the session (temporary capabilities)
  - access permissions are easily verified
- [FIG]

...Securing (or Protecting) a system (cont.)

Example of (kind of) hybrid access control: cookies on the Web

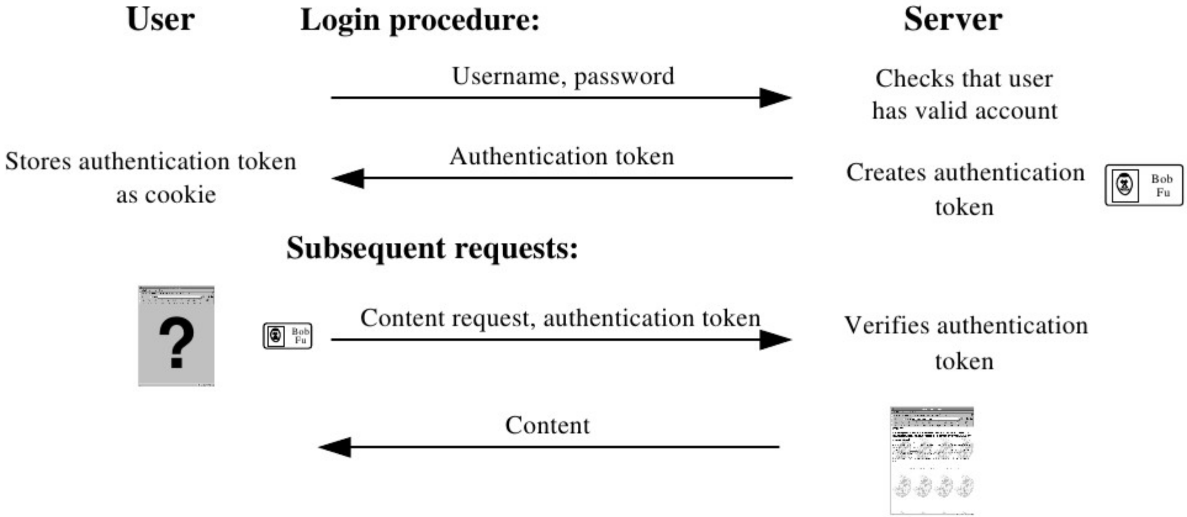
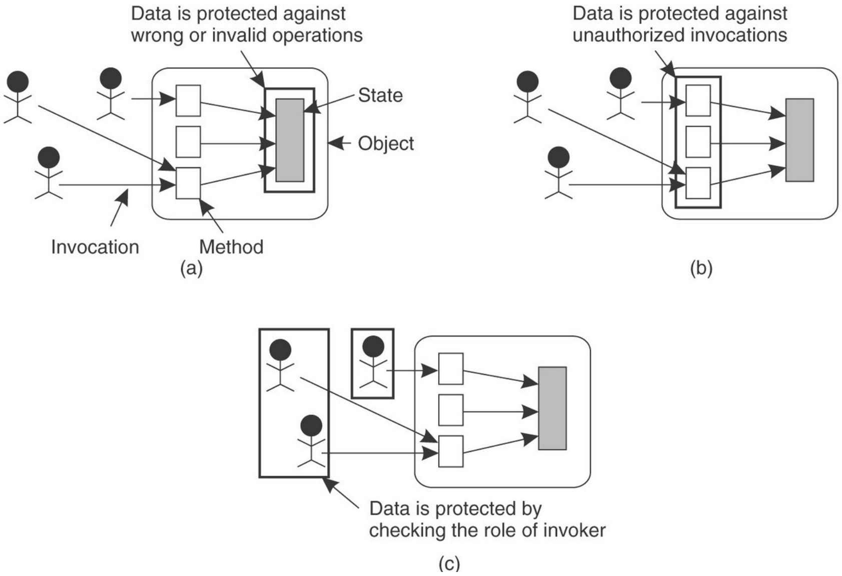

User
Login procedure:
Server
Username, password
Checks that user has valid account
Stores authentication token as cookie
Authentication token
Creates authentication token
Bob Fu
Subsequent requests:
?
Bob Fu
Content request, authentication token
Verifies authentication token
Content

Fig. Hybrid access control: common usage of Web cookies.
(*in* Fu et al., “Client Authentication on the Web”, 2001)

Other facets of access control mechanisms

Acting on several levels

- Control of: [FIG]
  - data (a)
  - invocation (b)
  - user (c)


Data is protected against wrong or invalid operations
State
Object
Invocation
Method
(a)
Data is protected against unauthorized invocations
(b)
Data is protected by checking the role of invoker
(c)

Restricting the execution environments

- Forcing code to run in special areas

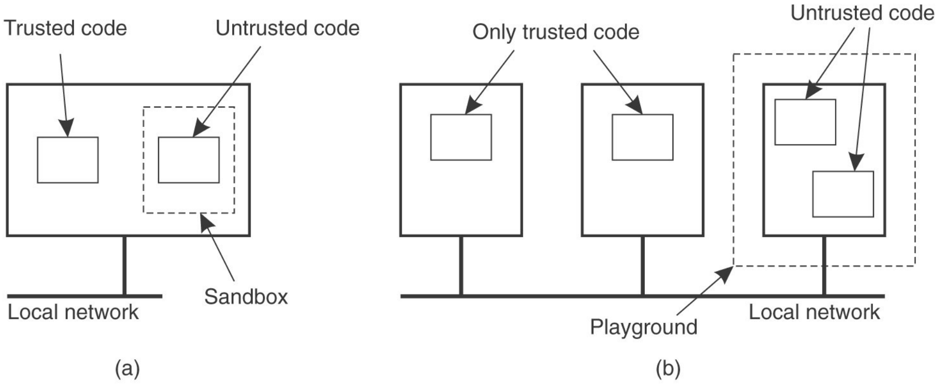


Fig. Use of *sandboxes*, a), or of *playgrounds*, b).

...Other facets of access control mechanisms (cont.)

Cryptography!

- Ubiquitous, in association with other mechanisms
- Several facets (techniques, algorithms...), already surfaced