modelli

conflitti e convergenza di interessi

- un sistema informatico viene tipicamente usato da molti utenti
- utenti hanno interessi diversi
  - alcuni hanno interesse a cooperare tra di loro
  - altri potrebbero trarre vantaggio da comportamenti “non cooperativi”
- conflitti e convergenza di interessi
  - alcune risorse devono essere **protette**
  - altre risorse devono essere **condivise**

modelli per il confinamento

- le politiche di sicurezza impongono un confinamento
  - gli utenti (o i soggetti) possono operare solo su certe risorse e non su altre
- i modelli hanno due obiettivi
  - esprimere politiche di sicurezza
  - imporre tali politiche
- scopi
  - applicativi
  - teorici

requisiti

per confinare bisogna…

- distinguere gli utenti (soggetti)
- identificare l'operazione richiesta
- identificare l'oggetto su cui l'operazione opera
- prendere una decisione
  - operazione ammessa o negata

- opzionalmente si può loggare

modelli “architetturali”

tali requisiti sono alla base di tre modelli

- **AAA**
  - nato in ambito telecomunicazioni con scopi non di sicurezza
- **reference monitor**
  - nato in ambito militare
  - requisito per ottenere certificazione di sistemi informatici
- **delega a controllore**
  - approccio tipico per dare ad utenti temporaneamente diritti più elevati

sono molto usati

AAA

- **Authentication, Authorization, Accounting**
  - IETF RFC 2903, anno 2000
  - orientato alle reti
  - radius è un “AAA protocol”
  - accounting: contabilizzazione del consumo delle risorse
  - ACCOUNTING ≠ gestione degli account
- lo standard focalizza sull’accesso a servizi di connettività, non su sicurezza
  - servizi dial-up, ADSL, telefonia, ecc.

AAA e la sicurezza

- AAA è ora associato soprattutto alla sicurezza
- non solo reti, ma anche sistemi sw e sistemi operativi
- chi ne parla in ambito di sicurezza spesso cambia il significato della terza A
  - Accounting → Auditing o Accountability
- infatti l'accounting non è molto rilevante in ambito sicurezza

Autenticazione (**A**AA)

- fase in cui si accerta l'identità dell'utente
- vari modi di identificazione
  - username e password
  - fingerprint
  - smartcard
  - "segreto crittografico"

user id

- l'utente è rappresentato nel sistema in qualche modo
- tipicamente a ciascun utente è associato un identificatore
  - es. in Unix: UID e login name
- tale identificatore viene passato alla successiva fase di controllo di accesso (detta anche Autorizzazione)

utenti senza utenza

- non necessariamente ciascun utente ha una “utenza” (login name)
  - es. web server viene acceduto da utenti senza utenza specifica
  - è possibile comunque distinguerli (es. session tracking con coockies e simili)
- gli utenti senza utenza **hanno tutti gli stessi diritti**
  - in RBAC diremmo che hanno tutte lo stesso ruolo “guest”

id utente ↔ utente “umano”

- ciascun utente umano dovrebbe essere rappresentato da **uno e un solo** identificatore
  - altrimenti alcune politiche configurate sui sistemi possono venir invalidate
  - **richiede una policy esterna al sistema**
    - cioè non è forzabile tecnologicamente
  - un matematico la chiamerebbe “corrispondenza biunivoca” un databasista “relazione 1 a 1”

anomalia: più user id per un utente

- policy: se un utente viola la regola R allora il suo accesso ai sistemi deve essere revocato
  - il meccanismo usato per implementare la politica è la chiusura dell'account che ha violato R
  - violabilità: se l'utente U ha due account U1 e U2 e viola R con U1 la chiusura di U1 non comporta la revoca dell'accesso ai sistemi poiché U ha ancora U2
- policy: ciascun utente o può avere diritti per l'operazione op1 o l'operazione op2 ma non per entrambe.
  - l'implementazione di questa politica verifica che ciascun account non abbia ha la possibilità di fare entrambe le operazioni
  - violabilità: se un utente U ha due account U1 e U2 dove U1 è abilitato a op1 e U2 è abilitato a op2, U ha in sostanza diritti per entrambe le operazioni in violazione della politica

accountability

è la caratteristica di poter associare un’azione o un evento a uno o più soggetti responsabili

- il responsabile è tenuto a giustificare o spiegare il suo operato
- strettamente legata a logging e auditing (la terza A di AAA)
- spesso imposta per legge

“strict accountability”

- strict accountability è la possibilità di ricondurre sempre una attività loggata (cioè una linea di log) ad un singolo soggetto del mondo reale (es. una persona fisica)
- fondamentale per assegnare la responsabilità delle azioni eseguite nel sistema

anomalia: più utenti per un user id

- in questo caso non è possibile ottenere la strict accountability

- si condivide una utenza quando i sistemi non offrono funzionalità di condivisione
  - o tali funzionalità non sono note agli utenti
- esempio tipico: più utenti conoscono la password di amministratore
  - i sistemi moderni permettono più utenze amministrative

Autorizzazione (A**A**A)

ha lo scopo di

- **negare** (deny) le richieste di operazioni **non conformi alla policy** di sicurezza
- **ammettere** (allow) le richieste di operazioni **conformi alla policy** di sicurezza

viene “eseguita” per ogni *richiesta di operazione*, detto anche *accesso*

autorizzazione

prevede tre fasi

- richiesta di accesso
  - stimolata da parte dell'utente o dell'entità che lo rappresenta nel sistema
  - parametri: **oggetto** e **operazione**
- **controllo di accesso (access control)**
  - in base alla policy di sicurezza
- autorizzazione
  - concessa (allow): l'operazione verrà eseguita
  - negata (deny): operazione non verrà eseguita

Auditing (AA**A**)

- auditing = controllo o verifica di “adeguatezza”
  - tipicamente manuale o semi-automatizzato
  - solo l’uomo ha l’adeguata flessibilità per adattarsi a variazioni di esigenze, strumenti, ambiente di lavoro, ecc.
- può essere fatto a vari livelli (varie accezioni diverse)
  - access auditing
  - log auditing
  - system security auditing
  - network security auditing
  - auditing di procedure (iso 27001)
  - auditing di competenze (di persone) (CISSP – SSCP e CISA – CISM)
  - ecc.

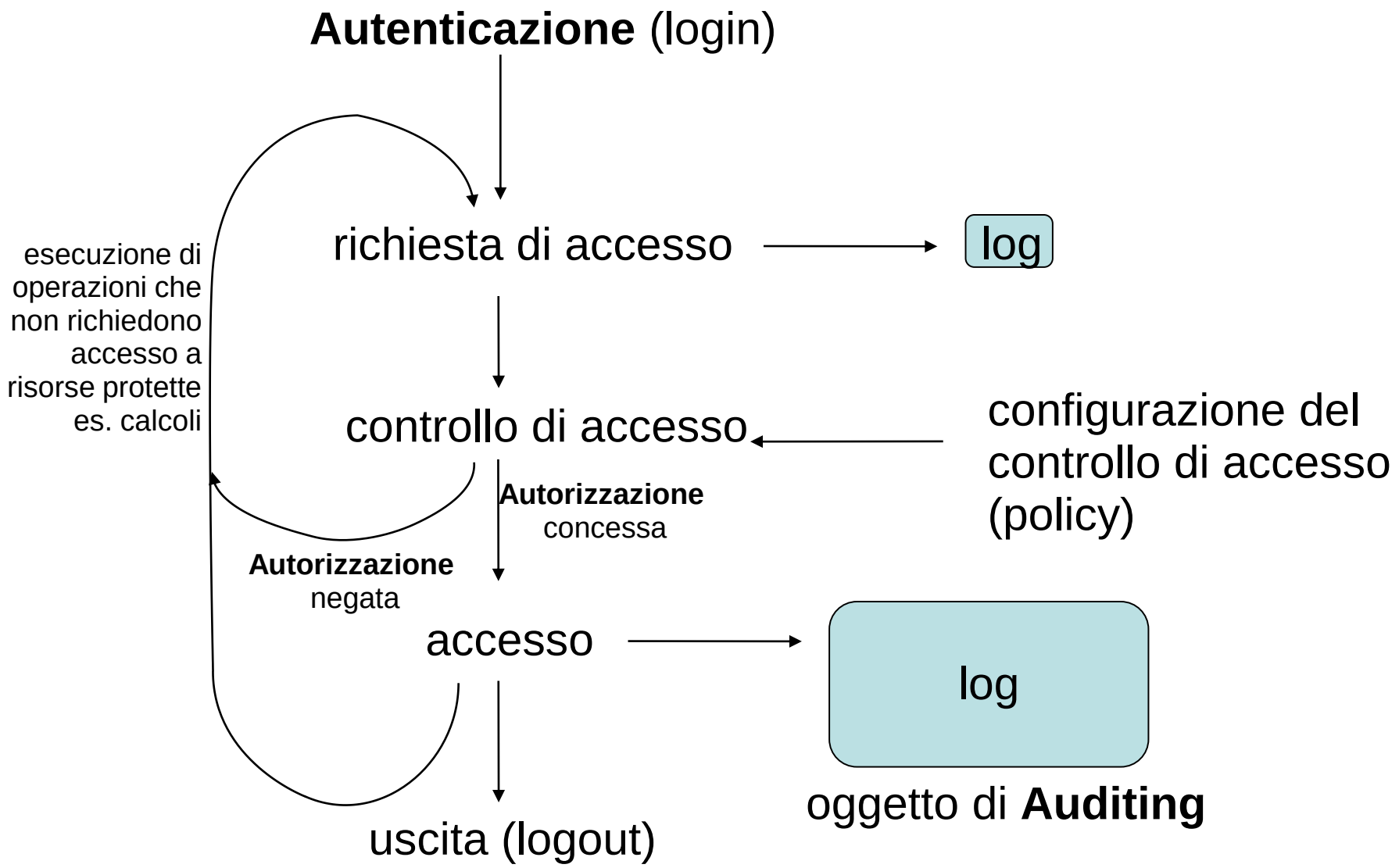
auditing

tipici eventi oggetto di auditing nel modello AAA

- autenticazione (riuscita o negata)
- richiesta di accesso
- risultato dell'access control per una richiesta di accesso (concessa o negata)
- risultato di una operazione

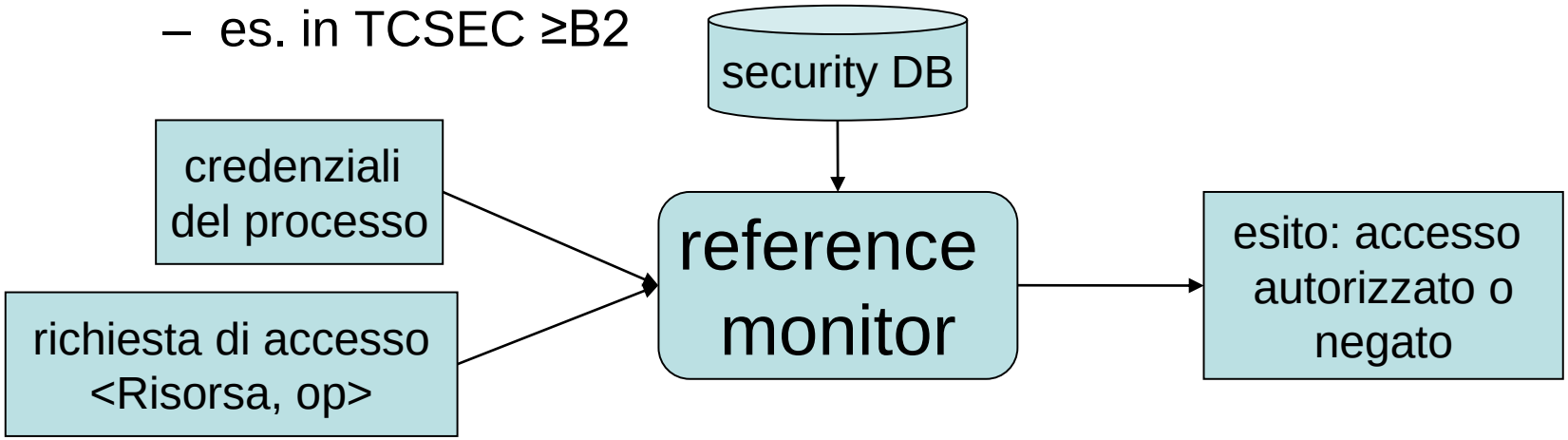
fuori dal modello AAA molti altri eventi possono essere oggetto di auditing

AAA: ciclo operativo



reference monitor (security kernel)

- parte del s.o. che effettua il controllo di accesso
  - è quindi relativo ad Authorization
- caratteristiche:
  - **invocato ad ogni richiesta di accesso** (mediazione completa)
  - **a prova di intrusione** (idealmente senza vulnerabilità)
  - **abbastanza piccolo da essere verificabile** (assurance)
- introdotto nel 1972 in James Anderson et al. - Computer Security Technology Planning Study
- richiesto in certe vecchie certificazioni
  - es. in TCSEC ≥B2



reference monitor: realizzazioni

- sistemi senza reference monitor
  - Windows 3.x, 95,98, Me
    - controllo di accesso di efficacia molto limitata
  - Linux <2.6
    - controllo di accesso “efficace” ma architettura senza reference monitor
- sistemi con reference moniotor
  - Windows NT, 2000, XP, 2003, Vista, 7, 8, 10
  - Linux >2.6 (Linux Security Modules)
    - vari modi di realizzare un security kernel
    - SELinux (Android), AppArmor, Smack, ecc.

the “impossible access” problem

- sometimes a subject needs to access a resource for which
  - it should have access rights (according to an ideal policy)
  - It has not access rights, in practice

typical reasons are

- policy is too restrictive since technology does not provide means to restrict access at the right granularity
  - e.g., a user need to modify his/her password, this implies to change the password database. Clearly, access to the whole password database *cannot be given to everybody!*
- resource is located remotely
  - e.g. web pages for a browser within a legal session

delegation

- a delegate *D* enables a subject *S* to exceptionally performs an operation *o* that is beyond its rights
- *D* perform access control before performing *o* on behalf of *S*
- *D* **can be programmed to apply any complex policy**
- *D* is **critical** from the secure programming point of view
  - it usual has very powerful rights

examples of delegations

- a user to the passwd command
- a user to the sudo command
- a browser to a web server
- a browser to a web proxy

even locally between processes, using inter process communication

- a graphic app to an X server
- a productivity app to a print server

modelli relativi alle policy di sicurezza

le policy di sicurezza sono di due tipi

- Discretionary Access Control (DAC)
- Mandatory Access Control (MAC)

la classificazione riguarda i “metadiritti”, cioè i diritti di alterare la policy di sicurezza

Discretionary Access Control

- gli utenti hanno il diritto per cambiare i diritti sulle risorse
  - es. il proprietario U1 di un file può dare, a sua discrezione, il diritto all'utente U2 di accedere a tale file
- implementazioni
  - Linux
  - Windows NT/2000/XP/2003/Vista/7/8/10
  - è il modello tipico che troviamo in tutti i sistemi

Mandatory Access Control

- gli utenti **NON hanno il diritto** di cambiare i diritti sulle risorse
  - potrebbe non esistere il concetto di proprietario di una risorsa
  - il proprietario di una risorse è, di fatto, l'organizzazione (ad es. rappresentata dall'amministratore di sistema) e non un certo utente
- i diritti sono configurati dall'amministratore o dallo sviluppatore
- non è detto che l'utenza di amministratore esista
  - se esiste è spesso tenuto a seguire procedure stringenti
- implementazioni
  - Linux: es. SELinux, AppArmor
  - Windows:(Mandatory Integrity Control, MIC)

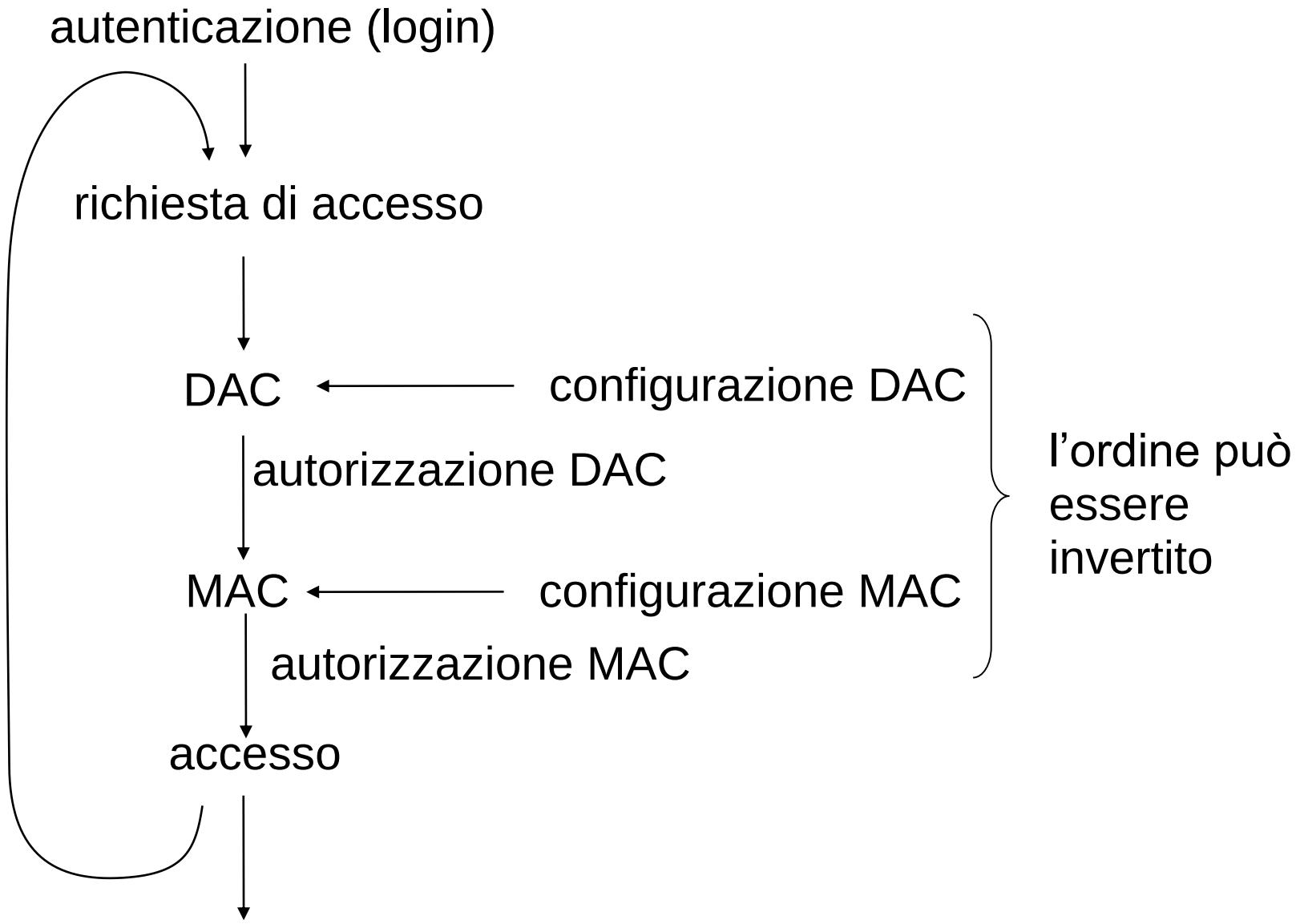
DAC vs. MAC

- DAC il più diffuso
  - flessibile
  - sicurezza “delegata” agli utenti (tipicamente al proprietario della risorsa)
    - facile per l’amministratore
- MAC considerato il più sicuro
  - molto scomodo per gli utenti
    - se due utenti devono scambiare un file devono chiamare l’amministratore
    - la riconfigurazione potrebbe essere complessa o impossibile
  - usato molto in ambito militare e sui server

MAC+DAC

- i sistemi con MAC tipicamente supportano anche DAC
- accesso consentito se sia i controlli mandatory che quelli discretionary danno autorizzazione
- permette di avere isole di “discrezionarietà” confinati da muri “mandatori”
  - es. il web server è separato dagli utenti da una configurazione MAC
  - ma MAC non isola gli utenti tra di loro, essi sono eventualmente isolati mediante DAC
    - configurazione standard per Fedora e RedHat

AAA: MAC+DAC

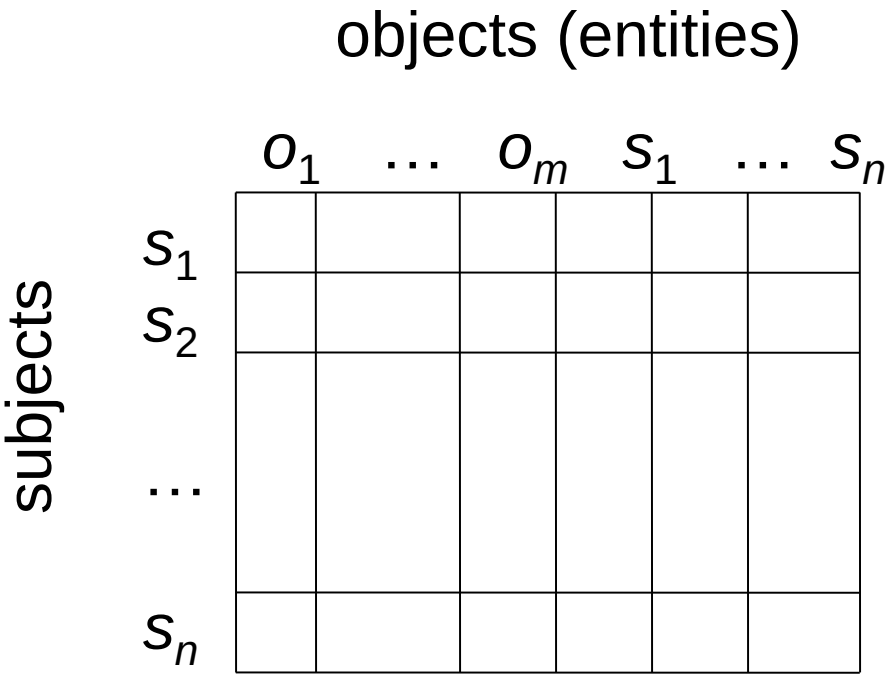


MAC e reference monitor

- il tipo di policy è indipendente dalla presenza di un reference monitor
  - storicamente i reference monitor hanno realizzato politiche MAC
  - entrambi sono spesso requisiti per alti livelli di sicurezza

access matrix
descrizione formale della policy

- matrice che descrive i diritti di ciascun soggetto sugli oggetti o soggetti



- Subjects $S = \{ s_1,\ldots,s_n \}$
- Objects $O = \{ o_1,\ldots,o_m \}$
- Rights $R = \{ r_1,\ldots,r_k \}$
- Entries $A[s_i, o_j] \subseteq R$
- $A[s_i, o_j] = \{ r_x, \ldots, r_y \}$ means subject s_i has rights $r_x, \ldots, r_y$ over object o_j

i soggetti sono anche oggetti

- nella maggioranza dei casi i soggetti sono anche oggetti
  - cioè un soggetto può operare su altri soggetti (che svolgono il ruolo di oggetti)
  - es. un processo può essere killato, debuggato, stoppato, ecc.
  - es. un utenza può essere creata, bloccata, cancellata

usi della access matrix

- per esprimere politiche
- per esprimere il “protection state”

protection state

- è la parte dello stato di un sistema che è rilevante per la sicurezza
- una transizione di stato è un cambiamento della access matrix
  - inserimento di un diritto in una cella
  - cancellazione di un diritto in una cella
  - creazione e distruzione di oggetti
    - aggiunge o cancella colonne
  - creazione e distruzione di soggetti
    - aggiunge o cancella righe e colonne
- **le transizioni di stato possono essere vincolate da particolari diritti nella matrice stessa**
  - eccezione: il diritto di creazione è eventualmente associato a un soggetto e non ad una cella della matrice

rappresentazione del protection state

- il modella ad access matrix ha un valore soprattutto teorico
  - troppo inefficiente dal punto di vista dello spazio occupato

si usano varianti ispirate alla rappresentazione delle matrici sparse in cui il default è “deny”

- access control list
  - ciascun oggetto ha associato una struttura dati che esprime quali soggetti possono agire sull’oggetto stesso e con che operazioni
  - es. usato nei filesystem
- capabilities
  - ciascun soggetto ha associato una struttura dati che esprime su quali oggetti può agire e con che operazioni

ownership

- s_i own o_j, cioè own$\in A[s_i, o_j]$
- si ha il diritto di modificare a piacimento l'elemento $A[s_i, o_j]$
  - cioè i propri diritti su o_j
  - più che un diritto su o_j è un diritto su $A[s_i, o_j]$, cioè è un meta-diritto
- esempi di vincoli sull'ownership sono…
  - un solo owner
    - come in unix e windows
  - ownership trasferibile
    - unix: chown, windows: "take ownership"
  - owner non modificabile
- spesso implica il diritto *grant*

grant e attenuazione del privilegio

- s_i grant o_j
- s_i ha il diritto di modificare gli elementi della colonna A[s, o_j] con s≠ s_i
  - cioè dare ad altri soggetti diritti su oj
- più che un diritto su o_j è un diritto su A[. , o_j]
- vincolo tipico è il cosiddetto principio di attenuazione del privilegio:
  si non può dare ad altri diritti che lui stesso non possiede
  - è un vincolo tipico nei DBMS (vedi SQL GRANT)

access matrix, DAC e MAC

- una matrice di accesso può rappresentare politiche DAC o MAC
- la presenza di diritti grant rendono la politica DAC

esempio

- Processes *p*, *q* soggetti
- Files *f*, *g* oggetti
- Rights *Read, Write, eXecute, Append, Own+grant*

	f	g	p	q
p	rwo	r	rwxo	w
q	a	ro	r	rwxo

- politica DAC

esempio

- Procedures *inc_ctr*, *dec_ctr*, *manage* soggetti
- Variable *counter* oggetti
- Rights +, –, *call*

	counter	*inc_ctr*	*dec_ctr*	*manage*
inc_ctr	+			
dec_ctr	-			
manage		*call*	*call*	*call*

- politica MAC