# Protezione e Sicurezza

La protezione riguarda l'insieme di attività che si preoccupano di garantire all'interno di un sistema di calcolo il controllo dell'accesso alle risorse logiche e fisiche da parte degli utenti.

La sicurezza garantisce l'autenticazione degli utenti impedendo accessi non autorizzati al sistema, tentativi dolosi di alterazione e distruzione dei dati.

# Protezione

Il controllo degli accessi è suddivisibile in tre livelli concettuali: modelli, politiche, meccanismi.

- modelli
- politiche
- meccanismi

# Modelli

• Un modello di protezione definisce i soggetti, gli oggetti ai quali i soggetti hanno accesso ed i diritti di accesso, cioè le operazioni con le quali si può accedere agli oggetti.

• soggetti sono la parte attiva di un sistema, cioè i processi che agiscono per conto degli utenti per accedere a determinati oggetti.

• oggetti costituiscono la parte passiva (risorse fisiche e logiche).

•Un soggetto può avere diritti di accesso sia per gli oggetti che per altri soggetti (un processo può controllarne un altro).

• Un **soggetto** può essere considerato come una coppia (processo, dominio), dove dominio è l'ambiente di protezione nel quale il soggetto sta eseguendo (insieme dei diritti di accesso posseduti dal processo).

• Un dominio di protezione è unico per un soggetto, mentre un processo può cambiare dominio durante la sua esecuzione.

• Il soggetto Si può rappresentare il processo P mentre esegue in un dominio di protezione Di ed il soggetto Sj può rappresentare lo stesso processo P mentre esegue in un altro dominio Dj.

# Politiche

Le politiche di protezione definiscono le regole con le quali i soggetti possono accedere agli oggetti:

- Discretional access control (DAC). Il creatore di un oggetto controlla i diritti di accesso per quell'oggetto ( UNIX).
- Mandatory access control (MAC). I diritti di accesso vengono gestiti centralmente. Installazioni di alta sicurezza (es., enti governativi).
- Role Based Access Control (RABC). Ad un ruolo sono assegnati specifici diritti di accesso sulle risorse. Gli utenti possono appartenere a diversi ruoli.

Caratteristica comune delle politiche di protezione:
Principio del privilegio minimo. Ad un soggetto sono garantiti i diritti di accesso solo agli oggetti strettamente necessari per la sua esecuzione.

# Meccanismi

• I meccanismi di protezione sono gli strumenti messi a disposizione dal sistema di protezione per imporre una determinata politica.

•Separazione tra meccanismi e politiche. La politica definisce cosa va fatto ed il meccanismo come va fatto.

•Flessibilità del sistema di protezione: i meccanismi di protezione devono essere sufficientemente generali per consentire l'applicazione di diverse politiche di protezione

## Separazione tra meccanismi e politiche

Esempio:

- UNIX fornisce un meccanismo per definire per ciascun file i tre bit di *read, write e execute* per il proprietario del file, il gruppo e gli altri.

- L'utente (o l'amministratore) definisce il valore dei bit (la politica).

# Dominio di protezione

• Un dominio definisce un insieme di oggetti ed i tipi di operazioni che si possono eseguire su ciascun oggetto (diritti di accesso):

*<nome oggetto, insieme diritti di accesso>*

• Un soggetto può accedere solo agli oggetti definiti nel dominio.

- **Domini disgiunti o domini con diritti di accesso in comune**.

( Corrisponde alla possibilità di due o più soggetti di effettuare alcune operazioni comuni su un oggetto condiviso)

| D1 | D2 | D3 |
|---|---|---|
| <File1, ( read, write)> | <File1, (execute) > | <File2, (read) > |
| <File3, (execute) > | <File2, (write) > | <File3, (read) > |

## Associazione tra un processo ed un dominio

- Statica: l'insieme delle risorse disponibili ad un processo rimane fisso durante il suo tempo di vita.
- Non adatta nel caso si voglia limitare l'uso delle risorse ad un processo a quello strettamente necessario.
- L'insieme globale delle risorse che un processo potrà usare può non essere un'informazione disponibile prima dell'esecuzione del processo.
- L'insieme minimo (politica del minimo privilegio) delle risorse necessarie ad un processo cambia dinamicamente durante l'esecuzione.

- Dinamica: associazione tra processo e dominio varia durante l'esecuzione del processo.

- Occorre un meccanismo per consentire il passaggio da un dominio all'altro del processo che, in questo modo, acquisisce diritti di accesso diversi.

## Esempi di cambio di dominio

Standard dual mode (monitor-user mode)

• Due domini di protezione:quello dell'utente (user mode) e quello del kernel (monitor o kernel mode)

• Cambio di dominio associato alle system call

• Quando un processo deve eseguire una istruzione privilegiata (accesso ai file, alle funzioni di rete, generazione dei thread etc.) avviene un cambio di dominio.

•Non consente la protezione tra utenti. Insufficiente per la multiprogrammazione

# Unix

• Dominio associato con l'utente. Il cambio di dominio corrisponde al **temporaneo** scambio di identità tra utenti.

• Ad ogni file sono associati l'**identificazione** del proprietario (*user-id*) e un bit di dominio (*set-uid)*.

• Quando un utente *A (user-id=A*) inizia l'esecuzione di un file il cui proprietario è B (*user-id=B*) ed il file ha *set-uid=on*, *user-id* di *A* è posto uguale a B temporaneamente. Quando l'esecuzione termina sono ripristinate le condizioni iniziali.

• Problema. Se un utente crea un file con *user-id=root* e con *set-uid=on* , l'utente può diventare *root* ed avere controllo del sistema.

# Il modello matrice degli accessi

- Un sistema di protezione può essere rappresentato utilizzando il modello matrice degli accessi.
- Il modello mantiene tutta l'informazione che specifica il tipo di accessi che i soggetti hanno per gli oggetti (stato di protezione) e consente di:
  - Rappresentare lo stato di protezione.
  - Garantire il rispetto dei vincoli di accesso (come specificati dallo stato di protezione) per ogni tentativo di accesso di un soggetto ad un oggetto.
  - Permettere la modifica controllata delle stato di protezione determinando una transizione di stato.

# Matrice degli accessi

| | O1 | O2 | O3 |
|---|---|---|---|
| S1 | read,write | execute | write |
| S2 | | execute | read,write, |

Il meccanismo associato al modello :

• ha il compito di verificare se una richiesta di accesso che proviene da un processo che opera in un determinato dominio è consentita oppure no.

• consente di modificare dinamicamente il numero degli oggetti e dei soggetti

• consente ad un processo di cambiare dominio durante l'esecuzione

• consente di modificare in modo controllato il cambiamento dello stato di protezione (transizione di stato)

Il meccanismo consente di assicurare che un processo che opera nel dominio *Dj* può accedere solo agli oggetti specificati nella riga *i* e solo con i diritti di accesso indicati.

• Quando un'operazione *M* deve essere eseguita nel dominio *Di* sull'oggetto *Oj*, il meccanismo consente di controllare che *M* sia contenuta nella casella *(i,j).* In caso affermativo l'operazione può essere eseguita. In caso negativo si ha una situazione di errore.

•Normalmente sono gli utenti a decidere il contenuto degli elementi della matrice di accesso. Se un utente crea un nuovo oggetto $F_j$, questo viene aggiunto nelle colonne della matrice di accesso e l'utente decide quali diritti inserire nella colonna j-sima in corrispondenza dei vari domini.

# Modifica dello stato di protezione

• La modifica controllata dello stato di protezione può essere ottenuta tramite un opportuno insieme di comandi (differenti sistemi di protezione).

•Definizione di comandi per passare i diritti tra i soggetti (Graham e Denning).

• Tipi di comandi:

- Propagazione dei diritti di accesso
- Aggiunta e rimozione di diritti di accesso per gli oggetti.

## Propagazione dei diritti di accesso

• La possibilità di copiare un diritto di accesso per un oggetto da un dominio ad un altro della matrice di accesso è indicato con un asterisco (* ) (copy flag).

• Un soggetto Si può trasferire un diritto di accesso $\alpha$ per un oggetto X ad un altro soggetto Sj solo se Si ha accesso a X e $\alpha$ ha il copy flag.

• S1, ad es., può propagare a S2 il diritto *read* per O1. L'operazione di propagazione può avvenire in due modi: viene copiato solo read (propagazione limitata di un diritto) oppure read* (S2 può copiare il diritto per un altro soggetto).

•Trasferimento di un diritto: il soggetto che trasferisce perde il diritto per l'oggetto.

# Matrice degli accessi

| | O1 | O2 | O3 | S1 | S2 |
|---|---|---|---|---|---|
| S1 | read*, write | execute | write | | control |
| S2 | | execute, owner | read, write, owner | | |

• Assegnazione di un diritto di accesso per un oggetto X ad un soggetto Sj da parte di un soggetto Si:

L'operazione è consentita solo se il diritto owner appartiene a A[Si,X].

Esempio, S2 può garantire il diritto write su O2 a S1

• Eliminazione da un soggetto Sj di un diritto di accesso per un oggetto X da parte di Si.

L'operazione è consentita solo se il diritto control appartiene a A[Si,Sj], oppure owner appartiene a A[Si,X]

Esempio, S2 può revocare a S1 il diritto execute su O2.

Graham e Denning mostrano che le regole definite precedentemente (assieme a poche altre) danno luogo ad un sistema di protezione in grado di risolvere problemi come:

- propagagazione limitata e controllata dei diritti di accesso (confinement)

- prevenire la modifica indiscriminata dei diritti di accesso di un processo (sharing parameters)

- uso non corretto dei diritti di accesso di un processo da parte di un altro (trojan horse)

# Realizzazione della matrice degli accessi

Problemi:

- Dimensione della matrice
- Matrice sparsa

Access Control List (ACL).

Memorizzazione per colonne: per ogni oggetto è associata una lista che contiene tutti i soggetti che possono accedere all'oggetto e per ogni soggetto i diritti di accesso per l'oggetto.

Capability List.

Memorizzazione per righe: ad ogni soggetto è associata una lista che contiene gli oggetti accessibili dal soggetto ed i relativi diritti di accesso

# Lista degli accessi

• La lista degli accessi per ogni oggetto è rappresentata dall'insieme delle copie ordinata

<soggetto, insieme dei diritti >

limitatamente ai soggetti con un insieme non vuoto di diritti per l'oggetto.

• Quando deve essere eseguita un'operazione M su un oggetto Oj da parte di Si, si cerca nella lista degli accessi

<Si,Rk> con M appartenente a Rk

La ricerca può essere fatta preventivamente in una lista di default contenente i diritti di accesso che, per la loro generalità, sono applicabili a tutti gli oggetti ( es. destroy object, copy object..)

• Se in entrambi i casi la risposta è negativa l'accesso è negato.

•ACL è stata descritta per utenti singoli. Molti sistemi hanno il concetto di gruppo di utenti. I gruppi hanno un nome e possono essere inclusi nella ACL.

•Siano UID(user identifier) e GID (group identifier) gli identificatori di un soggetto. L'entry in ACL ha la forma:

$UID_1, GID_1$: < insieme di diritti>
$UID_2, GID_2$: < insieme di diritti>

•Concetto di ruolo. Uno stesso utente può appartenere a gruppi diversi e quindi con diritti diversi.

•Quando accede deve specificare il gruppo di appartenenza (oppure ci sono differenti copie *login-password* per ogni gruppo)
Obiettivo: tenere separati i diritti di accesso.

• L'utente può accedere a certi oggetti **indipendentemente** dal gruppo cui appartiene. Ad esempio per il file F1:

UIDi,* : <diritti di accesso>

• Possibilità di bloccare selettivamente uno specifico utente:

UIDk,* : <insieme vuoto>
*,* : <insieme di diritti>

Tutti possono accedere al file tranne UIDk (le entry nella ACL sono esaminate in sequenza).

# Capability list

- La lista delle capability, per ogni soggetto, è la lista di oggetti assieme con gli accessi consentiti su di essi, cui il soggetto può accedere.

- Ogni elemento della lista prende il nome di capability. Ogni capability garantisce al proprietario (soggetto) certi diritti su un oggetto.

- La capability si compone di un nome fisico o indirizzo che identifica l'oggetto ed una sequenza di bit per i vari diritti.

- Quando S intende eseguire un'operazione M su Oj il meccanismo di protezione controlla se nella lista delle capability associata a S ne esiste una relativa ad Oj che abbia tra i suoi diritti M.

| | F1 | F2 | F3 | F4 | F5 | F6 | PR1 | PR2 |
|---|---|---|---|---|---|---|---|---|
| **soggetto** | - | - | R | RWE | RW | - | W | - |

| **Tipo** | **Diritti** | **Oggetto** |
|---|---|---|
| File | R | Puntatore a F3 |
| File | RWE | Puntatore a F4 |
| File | RW | Puntatore a F5 |
| Printer | W | Puntatore a stampante |

**capability list per il soggetto**

• Le liste di capability devono essere protette da manomissioni degli utenti. Ciò si può ottenere:

• Architettura etichettata, un progetto hardware in cui ogni parola ha un bit extra (*tag*) che dice se la parola contiene o meno una capability.
Il bit tag non è utilizzato dall'aritmetica, dai confronti e da altre istruzioni normali e può essere modificato solo da programmi che agiscono in modo kernel (S.O.) . Es IBM AS/400.

• La lista delle capability viene gestita solo dal S. O.
L'utente fa riferimento ad una capability specificando la sua posizione nella lista (soluzione simile all'uso dei descrittori di file in UNIX).

# Revoca dei diritti di accesso

· In un sistema di protezione dinamica può essere necessario revocare i diritti di accesso per un oggetto.

· La revoca può essere:
- selettiva o generale, cioè valere per tutti gli utenti che hanno quel diritto di accesso o solo per un gruppo.

- parziale o totale, cioè riguardare un sottoinsieme di diritti per l'oggetto o tutti .

- temporanea o permanente, cioè il diritto di accesso non sarà più disponibile, oppure può essere successivamente riottenuto.

- In un sistema a liste di accesso la revoca risulta semplice. Si fa riferimento alla ACL associata all'oggetto e si cancellano i diritti di accesso che si vogliono revocare.

- L'operazione risulta più complessa in un sistema a lista di capability. E' necessario infatti verificare per ogni dominio se contiene la capabilty con riferimento all'oggetto considerato.

# Confronto

Un sistema di protezione realizzato esclusivamente con ACL o capability list può presentare alcuni problemi di efficienza:

- **ACL**. L'informazione di quali diritti di accesso possieda un soggetto è sparsa nelle varie ACL relative agli oggetti del sistema.
Ogni accesso allo stesso oggetto da parte di un soggetto comporta una ricerca nella lista.

- **Capability list**. La rimozione di un oggetto con diritti di accesso per più soggetti comporta la ricerca in tutte le capability list relative.

- La soluzione che viene adottata in generale è di usare una combinazione dei due metodi.

# Soluzione mista

Soggetto tenta di accedere ad un oggetto per la prima volta:

- Si analizza la ACL; se esiste una entry contenente il nome del soggetto e se tra i diritti di accesso c'è quello richiesto dal soggetto, viene fornita la capability per l'oggetto.

- Ciò consente al soggetto di accedere all'oggetto più volte senza che sia necessario analizzare la ACL.

- Dopo l'ultimo accesso la capability è distrutta.

## Esempio: S.O. Unix. Apertura di un file

fd=open(<nome file>, <diritti di accesso>)

- Si cerca il file nel direttorio e si verifica se l'accesso è consentito (ACL).

- In caso affermativo viene creata una nuova entry nella tabella dei file aperti associata al processo, costituita da **fd e dai diritti di accesso** (*capability* ).

- Viene ritornato al processo **fd**, cioè *l'i-node* corrispondente al nuovo file aperto.

- Tutte le successive operazioni sul file sono eseguite utilizzando direttamente **fd** (capability) e verificando che il diritto di accesso sia tra quelli consentiti.

# Sicurezza multilivello

• La maggior parte dei sistemi operativi permette a singoli utenti di determinare chi possa leggere e scrivere i loro file ed i loro oggetti
*(DAC,* controllo discrezionale degli accessi).

• In alcuni ambienti è richiesto un *più stretto controllo sulle regole di accesso alle risorse* (ambiente militare,ospedali, aziende..). Vengono stabilite *regole* su chi può vedere cosa e non possono essere modificate senza aver ottenuto permessi speciali
(MAC, controllo degli accessi obbligatorio)

**Obiettivo**: assicurarsi che le politiche di sicurezza stabilite siano rispettate dal sistema.

- **Modello Bell-La Padula**

Progettato per gestire la sicurezza in ambiente militare.
Livelli di sicurezza (*sensibilità*) dei documenti:

- non classificato
- confidenziale
- segreto
- top secret

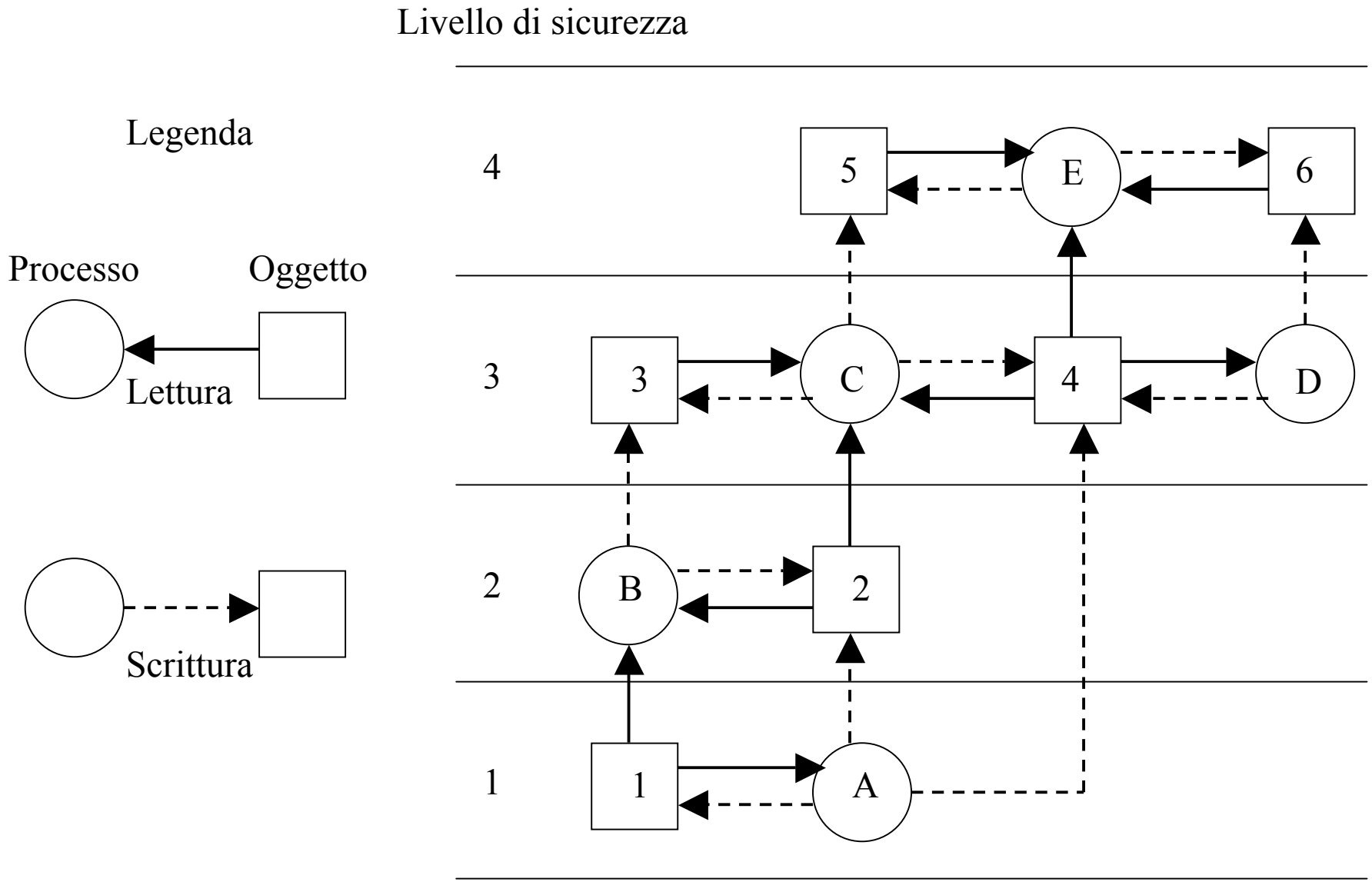
•Le persone sono assegnate ai livelli a seconda dei documenti che è loro consentito esaminare

Regole su come le informazioni possano circolare:

•Proprietà di semplice sicurezza: un processo in esecuzione al livello di sicurezza k può leggere solo oggetti al suo livello o a livelli inferiori.

•Proprietà*: un processo in esecuzione al livello di sicurezza k può scrivere solamente oggetti al suo livello o a quelli superiori

•I processi possono leggere verso il basso e scrivere verso l'alto, ma non il contrario.


Livello di sicurezza
Legenda
Processo
Oggetto
Lettura
Scrittura
4
3
2
1
5
E
6
3
C
4
D
B
2
1
A

•Il modello Bell-La Padula è stato concepito per mantenere i segreti non per garantire l'integrità dei dati. E' possibile infatti sovrascrivere l'informazione appartenente ad un livello superiore.

•**Modello Biba**

proprietà di semplice sicurezza: un processo in esecuzione al livello di sicurezza k può scrivere solamente oggetti al suo livello o a quelli inferiori (nessuna scrittura verso l'alto).

Proprietà di integrità*: un processo in esecuzione al livello k può leggere solo oggetti al suo livello o a quelli superiori (nessuna lettura verso il basso)

•I due modelli sono in conflitto tra loro e non si possono realizzare contemporaneamente.

# Reference Monitor

**Sistemi fidati:** sistemi per i quali è possibile definire formalmente dei requisiti di sicurezza.

**Reference monitor:** E' un *elemento di controllo* realizzato *dall'hardware e dal S.O.* che regola l'accesso dei soggetti agli oggetti sulla base di parametri di sicurezza del soggetto e dell'oggetto.

• Ha accesso a una base di calcolo fidata (Trusted Computing Base, o **base di dati del nucleo di sicurezza**-*security kernel data-base*) che contiene:

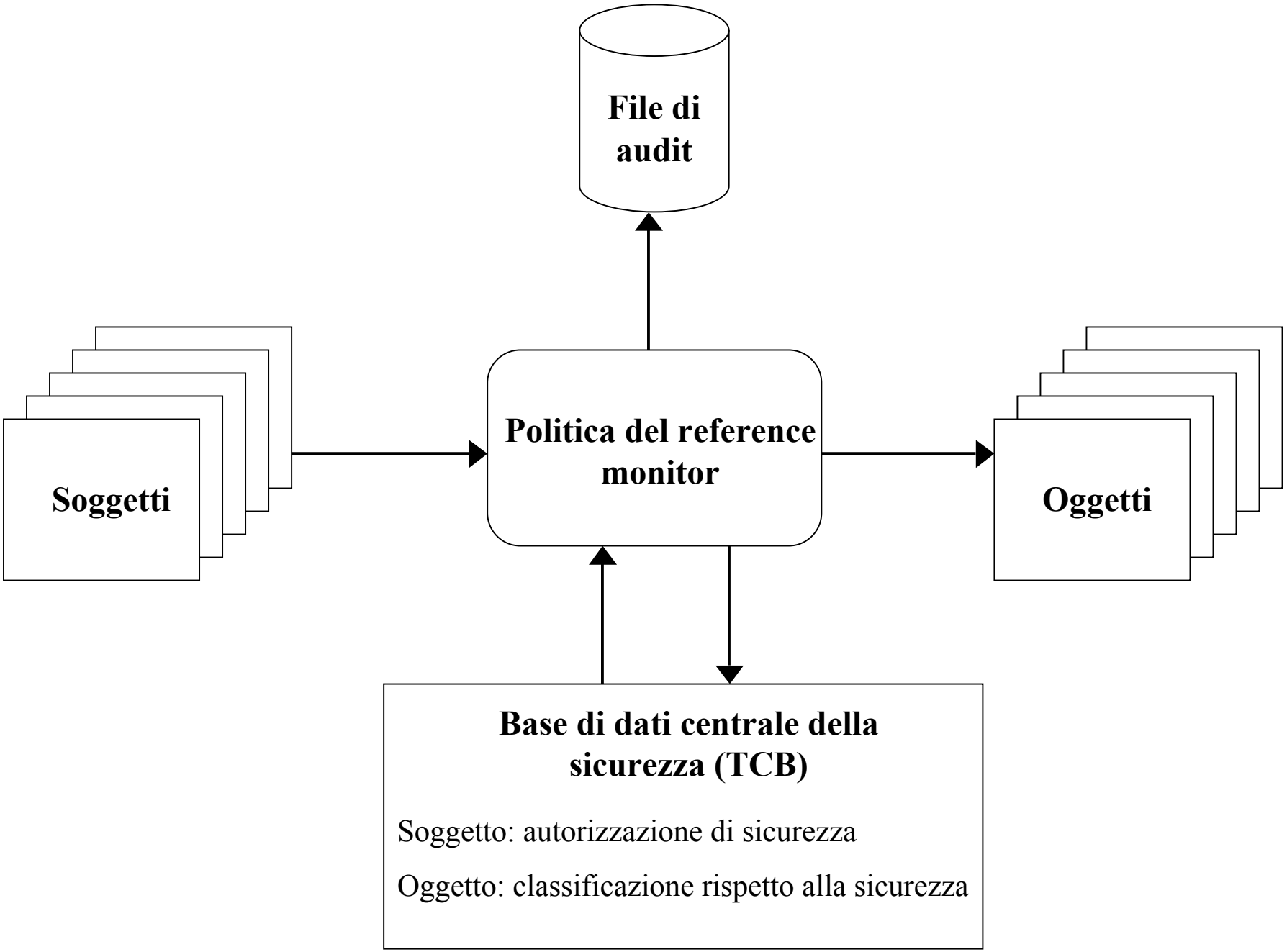
- Privilegi di sicurezza (*autorizzazioni di sicurezza*) di ogni soggetto.
- Attributi di protezione (*classificazione rispetto alla sicurezza*) di ciascun oggetto.

Il monitor dei riferimenti *impone le regole di sicurezza* (*no read-up*, *no-write down*) ed ha le seguenti proprietà:

- Mediazione completa: le regole di sicurezza vengono applicate *ad ogni accesso* e non solo, ad esempio, quando viene aperto un file.

- Isolamento: il monitor dei riferimenti e la base di dati *sono protetti* rispetto a modifiche non autorizzate.

- Verificabilità: la correttezza del monitor dei riferimenti deve esser provata, cioè deve esser possibile dimostrare *formalmente* che il monitor impone le regole di sicurezza ed fornisce mediazione completa ed isolamento


File di audit
Soggetti
Politica del reference monitor
Oggetti
Base di dati centrale della sicurezza (TCB)
Soggetto: autorizzazione di sicurezza
Oggetto: classificazione rispetto alla sicurezza

• Il requisito di *mediazione completa* rende preferibile, per motivi di efficienza, che la soluzione debba essere almeno parzialmente hardware.

• Il requisito *dell'isolamento* impone che non sia possibile per chi porta l'attacco, modificare la logica del reference monitor o il contenuto della base di dati centrale della sicurezza.

• Il requisito della *dimostrazione formale* è difficile da soddisfare per un sistema general-purpose.

**Audit file**

Vengono mantenuti in questo file gli *eventi importanti per la sicurezza*, come le violazioni alla sicurezza che sono state scoperte e le modifiche autorizzate alla base di dati del nucleo di sicurezza.

## Difesa dai cavalli di Troia

• Nell'esempio viene usato un cavallo di Troia per aggirare un meccanismo di controllo basato sulle liste di controllo degli accessi (ACL).

• Un utente, Paolo, ha creato un file F, contenente la stringa di caratteri riservati "CPE1704TKS", con i permessi di *lettura/scrittura* solo per i processi che appartengono a lui.

• Un utente ostile, Piero, ottenuto l'accesso al sistema installa sia il *cavallo di Troia* sia un file privato che verrà utilizzato come "tasca posteriore".

• Piero ha permessi di lettura e scrittura per il suo file e dà a Paolo il permesso di scrittura.

• Piero induce Paolo ad attivare il cavallo di Troia (per esempio, spacciandolo come un programma di utilità).

• Il programma, eseguito da Paolo copia la stringa dei caratteri riservati nel file "tasca posteriore" di Piero: sia l'operazione di lettura che quella di scrittura soddisfano i vincoli imposti da ACL.

**Utilizzo di un S.O. sicuro:** Vengono fissati due livelli di sicurezza, *riservato e pubblico*. Ai processi ed al file dati di Paolo viene assegnato il livello di sicurezza "riservato". A quelli di Piero il livello "pubblico".

• Quando Paolo attiva il cavallo di Troia, questo acquisisce il livello di sicurezza di Paolo e può vedere la stringa di caratteri riservata. Quando il programma tenta di memorizzarla in un file pubblico (file della tasca posteriore) la proprietà * verrebbe violata ed il tentativo non viene consentito dal *reference monitor* (anche se la ACL lo permetterebbe).

• La politica di sicurezza *ha la precedenza sul meccanismo delle ACL*

# Classificazione della sicurezza dei sistemi di calcolo

- **Orange Book**. Documento pubblicato dal Dipartimento della Difesa americano (D.O.D). Sono specificate quattro categorie di sicurezza:A,B,C,D(in ordine decrescente).

- **Categoria D**. Non ha livelli di sicurezza. Esempio MS-DOS, Windows 3.1.

- **Categoria C**. Suddivisa in C1 e C2.

*C1*. La TCB consente:

- Autenticazione degli utenti (password). I dati di autenticazione sono protetti rendendoli *inaccessibili* agli utenti non autorizzati.
- Protezione dei dati e programmi *propri di ogni utente*.
- Controllo degli accessi a *oggetti comuni* per gruppi di utenti definiti

Esempio: Unix

• *C2*. La TCB consente, oltre a quanto definito per la C1, il controllo degli accessi su una *base individuale*.
Esempio UNIX, Windows NT e 2000.

• **Categoria B**. Suddivisa in B1, B2 e B3
*B1*. La TCB consente, oltre a quanto definito in C2, l'introduzione dei livelli di sicurezza (modello Bell-La Padula). Almeno due livelli.

*B2*. La TCB estende l'uso di etichette di riservatezza ad ogni risorsa del sistema, compresi i canali di comunicazione.

*B3*. La TCB consente la creazione di liste di controllo degli accessi in cui sono identificati utenti o gruppi *cui non è consentito* l'accesso ad un oggetto specificato.

• **Categoria A**. Suddivisa in A1 e classi superiori
*A1*. E' equivalente a B3, ma con il vincolo di essere progettato e realizzato utilizzando metodi formali di definizione e verifica.
Un sistema appartiene ad una classe superiore ad A1 se è stato progettato e realizzato in un impianto di produzione affidabile da persona affidabile