Sistemi operativi

8. Sicurezza

Mauro Fiorentini

8. Sicurezza

Sistema operativo e sicurezza

- Un sistema operativo deve verificare che chi accede a una risorsa abbia i diritti per farlo.
- Il problema si presenta anche in caso di calcolatori personali, perché l'utente deve poter limitare i danni che i programmi possono causare, intenzionalmente o meno, e i furti di informazioni.

Risorse da proteggere

- Le risorse da proteggere sono:
  - I dispositivi di memoria permanente, per evitare furto o distruzione di dati;
  - Le periferiche, per evitare accessi indesiderati.
  - La CPU per impedire esecuzione indesiderata di programmi, che asserviscono un calcolatore ad altri.
- In un ambiente multiutente inoltre bisogna evitare che un utente si accaparri una quantità eccessiva di memoria, tempo di calcolo, spazio di memoria permanente.

Diritti

- I diritti fondamentali su una risorsa sono:
  - lettura;
  - scrittura;
  - esecuzione;
  - copia dei diritti, trasferendoli ad altra entità;
  - proprietà, col potere di cambiare i diritti;
  - controllo: diritto di togliere diritti detenuti da altri.

Schemi di protezione

- I principali schemi di protezione impiegati sono:
  - tabelle,
  - liste di accesso,
  - liste di capability,
  - chiavi.

Tabelle

- Idealmente costituite da una matrice, con una colonna per oggetto e una riga per entità attiva (utente o processo).
- I diritti elencati in una casella sono quelli che l'entità detiene sull'oggetto.
- Poco conveniente.
  - La maggior parte delle caselle è vuota;
  - Una colonna per ogni file occupa troppa memoria.

Liste di accesso

- Consistono nel mantenere per ogni oggetto una lista delle entità che hanno qualche diritto su di esso.
  - Difficile sintetizzare i diritti di un'entità.
  - Inefficiente: bisogna scandire una lista per ogni accesso.

Liste di capability

- Consistono nel mantenere per ogni entità una lista degli oggetti sui quali ha qualche diritto.
  - Difficile sintetizzare chi ha diritti di un oggetto.
  - Inefficiente: bisogna scandire una lista per ogni accesso.

Chiavi

- Il diritto viene dato alle entità che possiedono le chiavi di un oggetto.
- Per ogni entità si tiene una lista di chiavi possedute.
  - Verifiche semplici.
  - Impossibile elencare chi ha diritti di un oggetto.

Localizzazione delle capability

- Interne.
  - Conservate in tabelle interne al sistema.
- Esterne.
  - Conservate in area utente.
  - Servono meccanismi di crittografia per impedirne la falsificazione.
    - Lo stesso risultato si può ottenere con architetture “tagged”, che impediscono la modifica diretta di alcuni bit dell’indirizzo.

Revoca di un diritto

- Può essere:
  - immediata o ritardata;
  - selettiva o generale;
  - parziale o totale;
  - temporanea o permanente.

Revoca e meccanismi

- Con le matrici o le liste di accesso si hanno tutte le possibilità.
- Con le liste di capability la revoca è generalmente ritardata.
- Con le chiavi la revoca selettiva è impossibile, a meno di utilizzare più chiavi per lo stesso oggetto.

Revoca e liste di capability

- Ci sono vari schemi possibili.
  - **Revoca periodica**: le capability sono periodicamente revocate e devono essere riacquisite.
    - Costoso.
  - **Puntatori**: a ogni oggetto si associa una lista di puntatori a chi ne detiene le capability.
  - **Indirette**: le capability riferiscono una tabella di permessi; per revocare una capability basta rimuovere un elemento dalla tabella.

Protezioni in Unix (1)

- Basate su utente e gruppo.
- Per ogni utente esistono uno **user-id** e un **group-id**.
  - Più utenti possono appartenere allo stesso gruppo.
  - Un utente può appartenere a più gruppi.
  - Attribuiti al momento del login, tramite una tabella contenuta in un file.

Protezioni in Unix (2)

- Vi sono 3 permessi possibili:
  - lettura,
  - scrittura,
  - esecuzione.
- Ogni risorsa è protetta da 3 terne di permessi:
  - una per l'utente proprietario;
  - una per gli altri utenti dello stesso gruppo;
  - una per i restanti utenti.

Verifica dei permessi in Unix

- Al momento della connessione l'id dell'utente e del suo gruppo sono confrontati con quelli del proprietario della risorsa, per decidere la terna appropriata.
  - La verifica **non viene poi ripetuta** per ogni singolo accesso.
  - Eventuali modifiche alle protezioni non hanno effetto per gli utenti già connessi.
- Un utente privilegiato (**root**, user-id 0), ha diritti di accesso universali.

Protezioni e file in Unix

- Il diritto di esecuzione ha senso per file eseguibili e script di shell.
- Per le directory, il diritto di scrittura permette di creare, rimuovere e rinominare file, il diritto di esecuzione permette di attraversarle e cercare file.