Sicurezza

A. Ferrari

Sicurezza nelle Basi di Dati

- Nelle basi di dati è fondamentale il concetto di autorizzazione
- Si parla di sicurezza logica dei dati
  - procedure che assicurano che l'accesso di dati avvenga solo da parte di soggetti autorizzati secondo le modalità (lettura, scrittura, …) autorizzate

Protezione di una base di dati

- Protezione da accessi impropri
  - Autenticazione utenti
  - Accountability (capacità di identificare un utente)
- Individuazione e trattamento di dati sensibili
- Backup e ripristino
- Integrità semantica (ridondanza)

Controlli di accesso

- Controllo dell'identità dei soggetti
- Controllo della modalità di accesso (scrittura, lettura …) e dell'oggetto cui il soggetto chiede di accedere
- Le modalità di accesso vengono definite privilegi
- Il Data Base Administrator ha la possibilità di concedere e revocare privilegi di accesso alle risorse
  - GRANT (autorizzazione)
  - REVOKE (revoca dell'autorizzazione)

Viste utente

- Il DBA determina quale parte di un database è accessibile ad un utente
- CREATE VIEW
  - Crea una tabella virtuale il cui contenuto (colonne e righe) è definito da una query.
  - E' possibile utilizzare una vista
    - per semplificare e personalizzare la visualizzazione del database per ogni utente
    - come meccanismo di sicurezza grazie al quale è possibile consentire agli utenti di accedere ai dati tramite una vista, senza concedere loro le autorizzazioni di accesso alle tabelle di base sottostanti

Stored procedure

- Una stored procedure è un insieme di istruzioni SQL che vengono memorizzate nel server con un nome che le identifica
- E' possibile in seguito rieseguire l'insieme di istruzioni facendo semplicemente riferimento a tale nome
- Esempio

```
CREATE PROCEDURE nomeProc (IN param1 INT, OUT param2 INT)
SELECT COUNT(*) INTO param2
FROM tabella
WHERE campo1 = param1;
```

Trigger

- I trigger sono oggetti associati a tabelle, che vengono attivati nel momento in cui un determinato evento si verifica relativamente a quella tabella
- Quando definiamo un trigger, stabiliamo per quale evento deve essere attivato (inserimento di righe, modifiche o cancellazioni) e se deve essere eseguito prima o dopo tale evento
  - BEFORE INSERT
  - BEFORE UPDATE
  - BEFORE DELETE
  - AFTER INSERT
  - AFTER UPDATE
  - AFTER DELETE

```
CREATE DATABASE ProvaTrigger
USE ProvaTrigger

CREATE TABLE Studenti(
    nome varchar(50) NOT NULL,
    cognome varchar(50) NOT NULL
)

INSERT INTO Studenti VALUES ('Giuseppe', 'Verdi')
INSERT INTO Studenti VALUES ('Alberto', 'Ferrari')

-- Questa tabella conterra' gli studenti eliminati dalla tabella Studenti
CREATE TABLE StudentiAnnoPrecedente(
    nome varchar(50) NOT NULL,
    cognome varchar(50) NOT NULL,
    dataInserimento datetime
)

-- ******** Recupera gli studenti eliminati inserendoli in altra tabella ******
CREATE TRIGGER CancellazioneStudenti ON Studenti FOR DELETE
AS
    INSERT StudentiAnnoPrecedente
    SELECT nome, cognome, GETDATE() AS dataInserimento
    FROM deleted
-- ********************************************************************************

DELETE FROM Studenti Where cognome='Ferrari'

SELECT * FROM Studenti

SELECT * FROM StudentiAnnoPrecedente
```

Caratteristiche Password

- https://howsecureismypassword.net/
- 7 caratteri minuscoli (2 secondi)
  - Length: 7 characters
  - Character Combinations: 26
  - Calculations Per Second: 4 billion
  - Possible Combinations: 8 billion

Password utenti

- Per gestire gli account degli utenti è necessario salvare la password di ciascun utente all'interno del database
- Le password non andrebbero mai salvate in chiaro, ma mediante criptazione
- "L'amministratore del database può conoscere la mia password?"
  - Se è in chiaro sì
  - Se è criptata no
    - nemmeno noi possiamo recuperarla
    - Il DBA può cambiarla ma non conoscerla

Inserimento Password

- INSERT INTO utenti (id, nomeUtente, password) VALUES ('1','utente 1',md5('password1'));
- La password nel database risulta criptata
- Per controllare l'accesso da parte dell'utente è necessario conoscere la password originale
- SELECT id
  FROM utenti
  WHERE nomeUtente=utente1
  AND password=md5('password1')

MD5

- MD5 (Message Digest algorithm 5) indica un algoritmo crittografico di hashing realizzato da Ronald Rivest nel 1991
- La codifica prende in input una stringa di lunghezza arbitraria e ne produce in output un'altra a 128 bit. La codifica avviene molto velocemente e l'output (noto anche come "MD5 Checksum" o "MD5 Hash") restituito è tale per cui è altamente improbabile ottenere con due diverse stringhe in input uno stesso valore hash in output
- Ad esempio, viene utilizzata per controllare che uno scambio di dati sia avvenuto senza perdite, semplicemente attraverso il confronto della stringa prodotta dal file inviato con quella prodotta dal file ricevuto. Con lo stesso metodo si può verificare se il contenuto di un file è cambiato (funzione utilizzata dai motori di ricerca per capire se una pagina deve essere nuovamente indicizzata). È diffuso anche come supporto per l'autenticazione degli utenti attraverso i linguaggi di scripting Web server-side (PHP in particolare): durante la registrazione di un utente su un portale internet, la password scelta durante il processo verrà codificata tramite MD5 e la sua firma digitale verrà memorizzata nel database (o in qualsivoglia contenitore di dati). Successivamente, durante il login la password immessa dall'utente subirà lo stesso trattamento e verrà confrontata con la copia in possesso del server, per avere la certezza dell'autenticità del login.

wikipedia

Decriptare md5

- E' possibile decriptare?
- La risposta ufficiale è "no", ma con qualche riserva
- Se per decriptare intendiamo il procedimento di trovare la password originale partendo dalla chiave hash, purtroppo la risposta diventa – "sì, è possibile decriptare"
- Chi ha accesso al database può leggere il valore della chiave hash, e se proprio ha tempo da perdere, può lanciare qualche software che si metta a provare tutte le combinazioni di caratteri fino a che il risultato della funzione hash coincide

SQL injection

- SQL injection è una tecnica dell'hacking mirata a colpire le applicazioni che si appoggiano su un DBMS SQL
- Sfrutta l'inefficienza dei controlli sui dati ricevuti in input ed inserisce codice maligno all'interno di una query SQL
- L'SQL injection permette al malintenzionato di autenticarsi anche senza essere in possesso delle credenziali d'accesso e di visualizzare e/o alterare dati sensibili

SQL injection (esempio)

- `stringaSQL = "SELECT * FROM utenti WHERE user= '"+nomeUtente+"' AND pwd='"+pwd+"'"`
- `Se nomeUtente= "Giuseppe" e pwd= "Verdi" …`
- `SELECT * FROM utenti WHERE user= 'Giuseppe' AND pwd='Verdi'`
- `e se pwd = "Rossi' OR user='Giuseppe" …`
- `… è sufficiente conoscere il nome utente per accedere al sistema`
- http://bobby-tables.com/