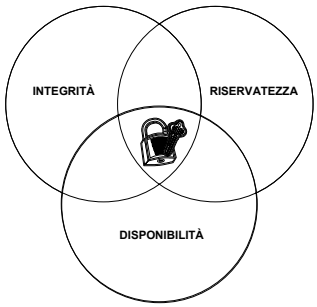
Sicurezza nei DataBase

Alessandro Ferrante / Giorgio Gamberini
Laboratorio di Basi di Dati 2

Sicurezza

- Quando si parla di *Sicurezza Informatica* vengono coinvolti tre aspetti molto importanti di qualsiasi sistema informatico: **riservatezza**, **integrità** e **disponibilità**.

Sicurezza



Sicurezza nei DataBase

Per quanto riguarda i DB possiamo identificare i seguenti requisiti di sicurezza:

- Integrità
- Verificabilità
- Riservatezza
- Autenticazione utente
- Disponibilità

Sicurezza nei DataBase

- **Integrità del DB**
  Include sia l'integrità *fisica* che *logica*. La struttura del DB deve essere conservata ed i dati devono essere protetti dal danneggiamento.
  Soluzione:
  - Backup periodici
  - Registro delle transazioni
- **Integrità dell'elemento**
  Implica la correttezza/accuratezza dell'elemento. Sia utenti che programmi possono compiere errori nell'inserimento/modifica dei dati.
  Soluzione:
  - Controlli di campo *(correttezza dell'elemento)*
  - Controllo degli accessi
  - Registro delle modifiche

Sicurezza nei DataBase

- **Verificabilità**
  Deve essere possibile rintracciare chi o che cosa ha eseguito l'accesso agli elementi del DB.
  Soluzione:
  - Generare un record di verifica di tutti gli accessi
- **Riservatezza**
  Un utente deve poter avere accesso solo ai dati autorizzati nelle modalità autorizzate
  Soluzione:
  - Controllo degli accessi

  Problema:
  - Inferenza dei dati

Sicurezza nei DataBase

- **Autenticazione utente**
  Ogni utente deve essere identificato
  Soluzione:
  - Autenticazione utente rigorosa (password, smartcard, caratteristiche biometriche)
- **Disponibilità**
  Gli utenti devono avere accesso al DB in generale e a tutti i dati di cui dispongono l'autorizzazione

Integrità

- **Aggiornamento in due fasi**
  Problema: malfunzionamento del sistema durante la modifica dei dati
  - **Fase 1 – Intento**:
    Il DBMS raccoglie le risorse necessarie per eseguire l'aggiornamento senza però apportare alcuna modifica al DB. Scrive un *flag di commit*.
  - **Fase 2 – Commit**:
    Esegue le modifiche. I *valori ombra* creati nella prima fase vengono copiati nel DB. Viene rimosso il *flag di commit*.

Integrità

- **Ridondanza / Coerenza interna**
  Molti DBMS mantengono informazioni aggiuntive per rilevare incoerenze interne dei dati:
  - Bit di parità
  - Codici di Hamming
  - Codici di ridondanza ciclica (CRC)

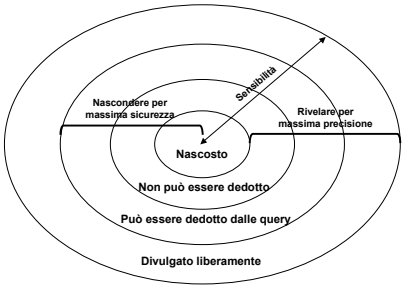
Integrità

- **Accesso concorrente**
  I DB sono sistemi multiutente: bisogna vincolare gli accessi delle persone che condividono lo stesso DB.
  Soluzione: il DBMS gestisce l'intero ciclo interrogazione/aggiornamento con una singola operazione atomica
- **Monitor**
  Utilità del DBMS responsabile dell'integrità strutturale del DB.
  - Confronto dell'intervallo
  - Vincoli di stato
  - Vincoli di transizione

Dati sensibili

- I dati sensibili sono quelli che non dovrebbero essere resi pubblici
- Decidere cosa è sensibile è complesso poiché i campi possono venire interessati in maniera indiretta
- Per ragioni di *riservatezza* sarebbe necessario divulgare solo di dati non sensibili
- *Precisione*: si mira a proteggere i dati sensibili pur rivelando la maggior quantità possibile di dati non sensibili

Sicurezza e precisione



Inferenza

- Vulnerabilità nella protezione dei DB
- Metodo per dedurre / derivare dati sensibili da dati non sensibili

Inferenza

Cognome	Sesso	Razza	Aiuto	Multe	Droghe	Dorm
Adams	M	C	5000	45	1	Holmes
Bailey	M	B	0	0	0	Grey
Chin	F	A	3000	20	0	West
Dewitt	M	B	1000	35	3	Grey
Earhart	F	C	2000	95	1	Holmes
Fein	F	C	1000	15	0	West
Groff	M	C	4000	0	3	West
Hill	F	B	5000	10	2	Holmes
Koch	F	C	0	0	1	West
Liu	F	A	0	10	2	Grey
Majors	M	C	2000	0	2	Grey

Inferenza

- Attacco diretto

```
SELECT   Cognome
WHERE    Sesso = M AND Droghe = 1
```

Tentativo di rendere meno ovvio l'attacco:

```
SELECT   Cognome
WHERE    (Sesso = M AND Droghe = 1) AND
         (Sesso != M AND Sesso != F) AND
         (Dorm = Ayres)
```


Inferenza

- Attacco indiretto

  Cerca di dedurre il risultato finale sulla base di uno o più risultati statistici intermedi

  - **Somma**

  Somma degli aiuti finanziari per dormitorio e sesso

	Holmes	Gray	West	Totale
M	5000	3000	4000	12000
F	7000	0	4000	11000
Totale	12000	3000	8000	23000

⇨ ***Liu* non riceve aiuti finanziari**

Alri attacchi indiretti sono solitamente basati su:

- **Conteggio, mediana, sistemi lineari**

Inferenza

- **Soluzioni**

  Esistono due modi per difendersi dagli attacchi di inferenza:

  - Applicare controlli alle query

    Difficile determinare se una data query divulga dati sensibili; efficaci solo contro gli attacchi diretti

  - Applicare controlli ai dati
    - Soppressione: i valori dei dati sensibili non vengono forniti
    - Occultamento: il valore fornito è *vicino* a quello effettivo
      - Combinazione dei risultati, arrotondamento, perturbazione casuale

Inferenza

- **Esempio soppressione: studenti per dormitorio e sesso**

	Holmes	Gray	West	Totale
M	1	3	1	5
F	2	1	3	6
Totale	3	4	4	11

I dati di questa tabella suggeriscono che le celle con conteggi 1 dovrebbero essere soppresse perché troppo rivelatori. Inoltre è necessario sopprimere almeno un'altra riga

	Holmes	Gray	West	Totale
M	-	3	-	5
F	2	-	3	6
Totale	-	-	-	11

Inferenza

- **Esempio occultamento: studenti per sesso e utilizzo droghe**

	0	1	2	3
M	1	1	1	2
F	2	2	2	0

Per occultare queste informazioni sensibili è possibile combinare i valori degli attributi 0 e 1, e anche 2 e 3, producendo dati meno sensibili

	0 o 1	2 o 3
M	2	3
F	4	2

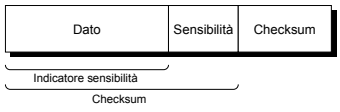
Database a più livelli

- La sicurezza di un singolo elemento può essere diversa dalla sicurezza di altri elementi dello stesso record
  → Occorre implementare la protezione per ogni singolo elemento
- Due livelli di sensibilità non sono adeguati
  → Occorre implementare vari gradi di sicurezza
- La sicurezza di un'aggregazione può essere differente dalla sicurezza dei singoli elementi

→ Modello militare di sicurezza: sensibilità di un oggetto è definita da *n* livelli

Database a più livelli

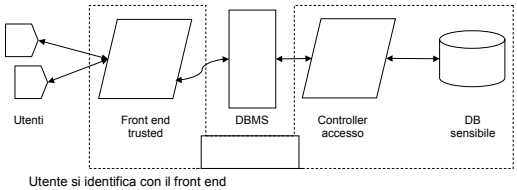
- Partizionamento
  Il DB è diviso in DB separati ciascuno con un proprio livello di sensibilità.
- Crittografia
- Blocco integrità



Front end trusted

- Il blocco dell'integrità è stato inventato come soluzione a breve termine al problema della sicurezza per i DB a più livelli. Tuttavia lo spazio per archiviare i dati deve essere aumentato per contenere l'etichetta
- Il concetto di Front End sfrutta l'esperienza e gli strumenti esistenti migliorando la sicurezza di essi con modifiche minime al sistema
- Il Front End Trusted serve come filtro unidirezionale scartando i risultati a cui l'utente non dovrebbe accedere
- Non è efficiente perché vengono recuperati più dati del necessario

Front end trusted



1. Utente si identifica con il front end
2. Utente invia query al front end
3. Front end verifica l'autorizzazione e invia la query al DBMS
4. Il DBMS interagisce col controller dell'accesso a basso livello ed accede ai dati effettivi
5. Il DBMS restituisce il risultato al Front end
6. Il Front end analizza i livelli di sensibilità dei dati nel risultato e seleziona gli elementi coerenti con il livello di protezione dell'utente

Filtri commutativi

- Semplificazione del Front end trusted
- Il filtro analizza la richiesta dell'utente, quindi la riformatta in modo da restituire solo i dati con il livello di sensibilità appropriato

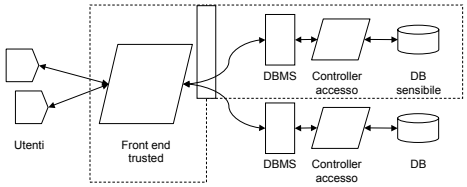
  La query:
  ```
  SELECT   Cognome
  WHERE    Sesso = M AND Droghe = 1
  ```

  Diventa:
  ```
  SELECT   Cognome
  WHERE      Sesso = M AND Droghe = 1
  FROM ALL RECORDS R
  WHERE    (Livello_segretezza_sesso(R) =<
     Livello_segretezza_utente)
           AND
           (Livello_segretezza_droghe(R) =<
     Livello_segretezza_utente)
  ```

- Il filtro fornisce quindi una seconda valutazione per selezionare solo i dati cui l'utente ha accesso

Database distribuiti

- Il Front end trusted controlla l'accesso a due DBMS commerciali: uno per i dati a bassa sensibilità e uno per quelli ad alta sensibilità
- Per un utente con livello di autorizzazione per dati ad alta sensibilità, il front end invia la query ad entrambi i DBMS



Sicurezza in Oracle

Alessandro Ferrante / Giorgio Gamberini
Laboratorio di Basi di Dati 2

Utenti

- ogni versione di oracle include nuove funzionalità a cui corrispondono nuovi utenti
- l'installazione di oracle 9i COMPLETA crea più di 20 utenti
- spesso non vengono utilizzati e le password non vengono cambiate
- alcuni di questi hanno privilegi molto alti

Utenti

- installare solo le opzioni strettamente necessarie all'utilizzo del database e blocco account non utilizzati

  ALTER USER username ACCOUNT LOCK;

- cambiare la password (che spesso è identica all'username)

ALTER USER username IDENTIFIED BY {new_password};

Privilegi utenti

Il DBA assegna a ogni utente creato dei privilegi che sono:

- privilegi di sistema
  permettono all'utente di connettersi al database e creare o manipolare oggetti
- i privilegi di oggetto
  permettono all'utente l'accesso ai dati contenuti nell'oggetto o permettono all'utente di eseguire un programma memorizzato.

Esempio privilegi

GRANT system_privilege TO {user_name | role | PUBLIC} [WITH ADMIN OPTION];
GRANT {object_priv|ALL} [(column1,...)] ON object TO {user|role|PUBLIC} [WITH GRANT OPTION]

GRANT create session TO user
(dovrebbe essere l'unico privilegio di sistema per l'utente, permette di accedere al db)
GRANT select ON oggetto to user
(permette select su oggetto per l'user)

Esempio privilegi

con l'opzione WITH ADMIN OPTION si puo concedere all'utente non solo il privilegio ma anche la possibilità di concedere lo stesso privilegio ad altri utenti

va evitata

si perde il controllo del privilegio

Meccanismi di autenticazione

Un utente in un sistema Oracle viene riconosciuto da un nome utente ed una password

Di default Oracle non attiva nessuna direttiva particolare per il controllo della password

La complessità (per esempio lunghezza) non è testata

Un utente può sbagliare la password un numero indefinito di volte senza essere bloccato

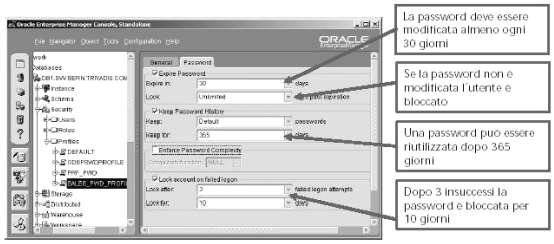
Enterprise Manager

Tool di Oracle che permette:

- cambio password periodico,
- se password non modificata l'utente viene bloccato,
- riutilizzo password solo dopo X giorni,
- dopo X insuccessi la password viene bloccata

….. oppure creare una propria funzione PL/SQL per cui `e possibile aggiungere ulteriore sicurezza

Enterprise Manager



Data dictionary

Il data dictionary è fondamentalmente una registrazione interna dello stato di tutti gli oggetti del database: tabelle, utenti, indici, sequenze, viste, link a database, ecc.

Possiamo definirlo come un dizionario di "metadati", cioè dati che descrivono gli oggetti nel database.

Data dictionary

07_DICTIONARY_ACCESSIBILITY parametro nel file di configurazione init<sid>.ora

- TRUE gli utenti con privilegio sistema %ANY% possono a seconda del privilegio leggere,modificare,eseguire gli oggetti nel D.D.
- FALSE (default da oracle9i) gli utenti con privilegio di sistema %ANY% nn hanno accesso al D.D. ma solo il SYSDBA.

  E' comunque possibile distribuire privilegi sugli oggtti del D.D.

Rete

Oracle, per l'accesso ai database, supporta diversi protocolli quali TCP/IP, SPX, DECnet e reti eterogenee, seguendo un modello di computazione client (sql*net) server (listener db).

Rete

MA….

sql*net trasferisce in "chiaro" i dati

non è possibile rilevare se un pacchetto di rete sia stato intercettato e/o modificato durante il transito in rete.

Rete

- Secure Socket Layer (SSL) (protocollo standard per la sicurezza delle connessioni di rete)
- Kerberos e CyberSafe: (protocollo di autenticazione dei servizi di rete, si serve della crittografia a chiave segreta)
- Smart Cards (dispositivi fisici simili ad una carta di credito)

Advanced Security

permette di scegliere i metodi di crittografia da utilizzare, si possono impostare indirizzi ip accettati, indirizzi rifiutati, porte utilizzate

parametri nel file protocol.ora:

tcp.validnode_checking=YES

tcp.excluded_nodes={list of ip address}

tcp.invited_nodes={list of ip address}

Auditing

Processo che offre la possibilità di monitorare e registrare le attività all'interno del database.

in Oracle la funzione di AUDIT TRAIL permette la registrazione di qualsiasi attività, ma non è possibile effettuare un monitoraggio a livello riga

… è possibile scrivere del codice PLSQL

Auditing

In init<sid>.ora :

- AUDIT FILE DEST: specifica directory in cui le informazioni saranno memorizzate
- AUDIT TRAIL: abilita o disabilita l'auditing . Valori usati:
- NONE (default)l'auditing non sarà abilitato
- OS i risultati saranno scritti in un file presente nella directory AUDIT FILE DEST.
- DB i risultati di auditing saranno conservati nella tabella SYS.AUD$.

Auditing

Oracle permette tre tipi di controllo rivolti a:

- statement SQL specifici (connect, create table,);
- privilegi (system grant, ecc);
- operazioni (select, insert, alter, ecc) sugli oggetti dei Oracle.

AUDIT [option | ALL] ON [username.]objectname
[BY [SESSION|ACCESS]]
[WHENEVER [NOT] SUCCESSFUL]

Backup

I backup possono essere divisi in due categorie principali:

- backup fisici, sono copie dei file fisici del database
- backup logici, contengono dati esportati usando comandi sql e memorizzati in un file binario.

Backup

- Export/Import: due utility di Oracle che permettono di esportare ed importare l'intero contenuto logico (tabelle,viste, ecc...) di uno o più schemi in un unico file;
- Off-line backup: si effettua quando si arresta il database. E' una copia fisica di tutti i file;
- On-line backup: con database attivo si mettono i tablespace nello stato di backup e si effettua la copia dei loro file di redo log e control

PL-SQL

Programmi scritti in PL/SQL1 possono essere memorizzati in forma compilata all'interno del databasefile funzioni sono eseguite da una chiamata esplicita (non c``e quindi trasparenza):

EXECUTE nome_funzione(param);

eseguite con i privilegi del proprietario (colui che ha creato laprocedura o funzione)
Per fare in modo che una procedura possa essere eseguita con i diritti del chiamante (invoker-rights) `e sufficiente aggiungere un parametro opzionale AUTHID CURRENT USER alla sua dichiarazione

Aggiornamenti web

Periodicamente visitare l'indirizzo web di Oracle per valutare possibili vulnerabilità ed applicare le opportune patch:

http://otn.oracle.com/deploy/security/alert

Bibliografia

- "Sicurezza in informatica" C.Pfleeger, S.Pfleeger
- http://www.oracle.com