

Securitatea bazelor de date

Introducere

Plan

- Rezumat
- Introducere
- Controlul accesului
- Accesul la aplicații
- Vulnerabilitatea
- Inferența
- Mecanismele de audit

Importanța domeniului

- Creșterea numărului de incidente raportate (pierdere sau expunere neautorizată a datelor)
- Cantitatea de date colectate, stocate și partajate crește
- ➡ Necesitatea securizării bazelor de date

Scop

- Securitatea bazelor de date trebuie:
 - să asigure accesul controlat și protejat la conținutul bazei de date
 - să păstreze integritatea, consistența și calitatea datelor
- Securitatea bazelor de date include o varietate mare de subiecte: securitatea fizică, securitatea rețelei, criptarea, autentificarea, autorizarea etc.

Concepte privind securizarea datelor

- Dintre aspectele pe care le poate cuprinde securitatea bazelor de date, un subiect important este securizarea datelor.
- Acest context presupune 3 aspecte:
 - Confidențialitatea (protecția datelor de a fi divulgate în mod neautorizat);
 - Integritatea (prevenirea accesului neautorizat la date);
 - Disponibilitatea (identificarea și recuperarea în urma erorilor hardware și software sau a acțiunilor răuvoitoare care conduc la refuzul disponibilității datelor)

Securitatea datelor

● Cele 3 aspecte precedente ale securității datelor includ, la rândul lor, următoarele subiecte:

- Controlul accesului
- Accesul la aplicații
- Vulnerabilitatea
- Inferența
- Mecanismele de audit

Securitatea datelor

- Controlul accesului este procesul prin care sunt atribuite drepturi și privilegii utilizatorilor și obiectelor bazei de date
- Accesul la aplicații se referă la necesitatea de a atribui drepturi adecvate de acces aplicațiilor externe, care solicită o conexiune la baza de date
- Vulnerabilitatea se referă la “slăbiciuni” care permit utilizatorilor răuvoitori să exploateze resurse
- Inferența se referă la utilizarea datelor legitime cu scopul de a deduce informații necunoscute, pentru a căror regăsire directă nu au fost acordate drepturi
- Auditarea bazei de date înregistrează accesul la baza de date și activitatea utilizatorilor, furnizând un mod de identificare a breșelor apărute

Introducere

- Tehnologia bazelor de date constituie o componentă a nucleului multor sisteme de calcul.
- Bazele de date permit stocarea și partajarea datelor, iar cantitatea datelor conținute în aceste sisteme crește exponențial.
- Același lucru se poate afirma despre necesitatea de a asigura integritatea datelor și de a securiza accesul la acestea.

Introducere

- *The Privacy Rights Clearing House* (2010) a raportat că mai mult de 345 milioane de înregistrări referitoare la clienți au fost pierdute sau furate din 2005 (de când investighează astfel de incidente)
- Ponemon Institute (2009) raportează costul mediu al unei infrațiuni asupra datelor: 202\$ pentru o înregistrare a unui client
- În august 2009 a avut loc cea mai mare infrațiune privind securitatea datelor: au fost furate peste 130 milioane de numere de carduri de debit și credit utilizând o vulnerabilitate binecunoscută a bazelor de date, SQL Injection

Introducere

- *Verizon Business Risk Team* raportează statistici privind infracțiunile asupra datelor din 2004; au examinat 90 de infracțiuni în cursul anului 2008
- Au raportat că mai mult de 285 milioane de înregistrări au fost compromise, iar acest număr depășește totalul celor din anii precedenți
- Studiile au avut în vedere cine comite astfel de acțiuni și cum apar ele
 - Cele mai multe infracțiuni apar din surse externe: 75% provin din afara organizației
 - 91% din înregistrările compromise aveau legătură cu grupuri de crimă organizată
 - Majoritatea infracțiunilor au avut loc prin hacking și deseori au fost facilitate de erori comise chiar de către victimă.
 - Accesul neautorizat și SQL Injection au fost găsite cele mai comune forme de hacking, lucru interesant având în vedere că acestea sunt cunoscute și adesea pot fi prevenite.

Aspecte ale securității bazelor de date

- Controlul accesului
- Accesul la aplicații
- Vulnerabilitatea
- Inferența
- Mecanismele de audit

Controlul accesului

- Metoda primară de a proteja datele este de a limita accesul la acestea, iar acest lucru poate fi realizat prin:
 - Autentificare
 - Autorizare
 - Controlul accesului
- Aceste 3 mecanisme sunt distincte, dar de obicei sunt utilizate împreună. Controlul accesului determină granularitatea cu care sunt atribuite drepturi anumitor utilizatori și obiecte.
 - Multe SGBD-uri utilizează o formă de autentificare (de exemplu, username și parolă) pentru restricționarea accesului la sistem
 - Utilizatorii sunt autorizați sau li se acordă privilegii asupra resurselor
 - Controlul accesului rafinează acest proces atribuind drepturi și privilegii obiectelor (tabele, vizualizări, linii și coloane) și seturilor de date.
 - Exemplu: StudentA poate avea drepturi de login asupra bazei de date a universității, cu privilegii de autorizare care le includ pe cele de read-only pentru tabelul Lista_cursuri. Prin acest nivel granular de control al accesului, studenții pot consulta lista de cursuri dar nu și notele primite de colegii lor.
 - Limitarea accesului la obiectele bazei de date se poate realiza cu ajutorul mecanismului de control Grant/Revoke.

Controlul accesului (Grant/Revoke)

- Concept de bază în securitate
- Limitează acțiunile (select, insert, update, delete, execute) pe care utilizatorii le pot efectua asupra obiectelor (tabele, coloane, vizualizări, proceduri stocate)
- Poate fi definit în 3 moduri:
 - Mandatory Access Control (MAC)
 - Discretionary Access Control (DAC)
 - Role Based Access Control (RBAC)
- Exemplu: Dr. Smith primește privilegii de citire asupra tabelului Student

Controlul accesului (Grant/Revoke)

- MAC și DAC furnizează privilegii utilizatorilor și grupurilor
- Regulile MAC sunt aplicate la nivelul sistemului, sunt statice și considerate mai sigure
 - Exemplu: Dr. Smith primește acces pentru citire pe tabelul Student
- Regulile DAC sunt furnizate de către utilizator, sunt dinamice și orientate către conținut
 - Exemplu: Dr. Smith primește acces pentru citire la tabelul Student, dar doar pentru studenții înscriși la un anumit curs

Controlul accesului (Grant/Revoke)

- RBAC este eficient pentru sistemele de baze de date
 - Rol = job
 - Identificarea operațiilor și a obiectelor asupra cărora operațiile necesită acces
 - Utilizatorii asociați unui rol primesc automat privilegiile asociate acestuia
 - Exemplu: Dr. Smith poate fi asociat unui rol Faculty, iar acesta conține dreptul de a citi tabelul Students, a obține data de înscriere la un anumit curs, a actualiza notele studenților asigurați cursului

Controlul accesului (Grant/Revoke)

- Identificarea utilizatorilor și determinarea necesităților de procesare și de acces la date ale acestora este un pas important în stabilirea unor bune protocoale de securitate a bazelor de date
- Identificarea și definirea role-urilor, acordarea corectă a drepturilor de acces asupra obiectelor și acțiunilor, asocierea corespunzătoare a utilizatorilor cu acele role-uri constituie dificultatea acestui proces

Controlul accesului (Grant/Revoke)

- După ce un role este creat, formatul pentru implementarea RBAC urmează modelul:
 GRANT nume_privilegiu
 ON nume_obiect TO nume_role;
- nume_privilegiu identifică drepturile care pot fi acordate; acestea pot include selectarea datelor, modificarea sau prelucrarea structurii bazei de date
- ON identifică obiectele bazei de date
- TO identifică role-urile cărora le sunt aplicate acele privilegii
- Exemplu: Lui Dr. Smith i-a fost atribuit role-ul Faculty, iar acestuia i-au fost acordate drepturi de citire asupra tabelului Students. Regula RBAC este:
 GRANT select ON students TO faculty;

Controlul accesului (Grant/Revoke)

- REVOKE – revocarea drepturilor și retragerea autorizării utilizatorilor din role-uri
- Exemplu: Ștergerea privilegiilor asupra tabelului Students din role-ul Faculty. Membrii acestui role nu vor mai putea accesa datele tabelului

Controlul accesului (Grant/Revoke)

- Sintactic, crearea role-urilor și implementarea RBAC sunt simple
- Provocarea constă în gestiunea utilizatorilor și a role-urilor asociate; aceasta include nu numai identificarea corectă a role-urilor, ci și gestiunea continuă a celor acordate
- Regula generală de securitate: atribuirea celui mai restrictiv set de privilegii necesare realizării task-urilor autorizate
- Construirea structurii organizaționale pentru un sistem RBAC poate deveni complexă, iar necesitatea de a schimba frecvent role-urile utilizatorilor înseamnă că RBAC presupune o monitorizare constantă

Controlul accesului (Grant/Revoke)

- În cartea *Security Metrics: Replacing Fear, Uncertainty and Doubts* (Jaquith, 2007):

“Today’s information security battleground is all about entitlements – who’s got them, whether they were defined properly, and how to enforce them”

Securitatea la nivel de înregistrare

- Controlul accesului la tabele sau coloane poate fi asigurat prin simpla acordare a privilegiilor asupra acestora
- Restricționarea accesului la datele conținute în înregistrări solicită pași suplimentari
- Exemplu: Un student ar trebui să poată accesa sau modifica doar înregistrările referitoare la el însuși
- Un mod comun de implementare a securității la nivel de linie este definit prin intermediul vizualizărilor SQL

Securitatea de nivel de înregistrare

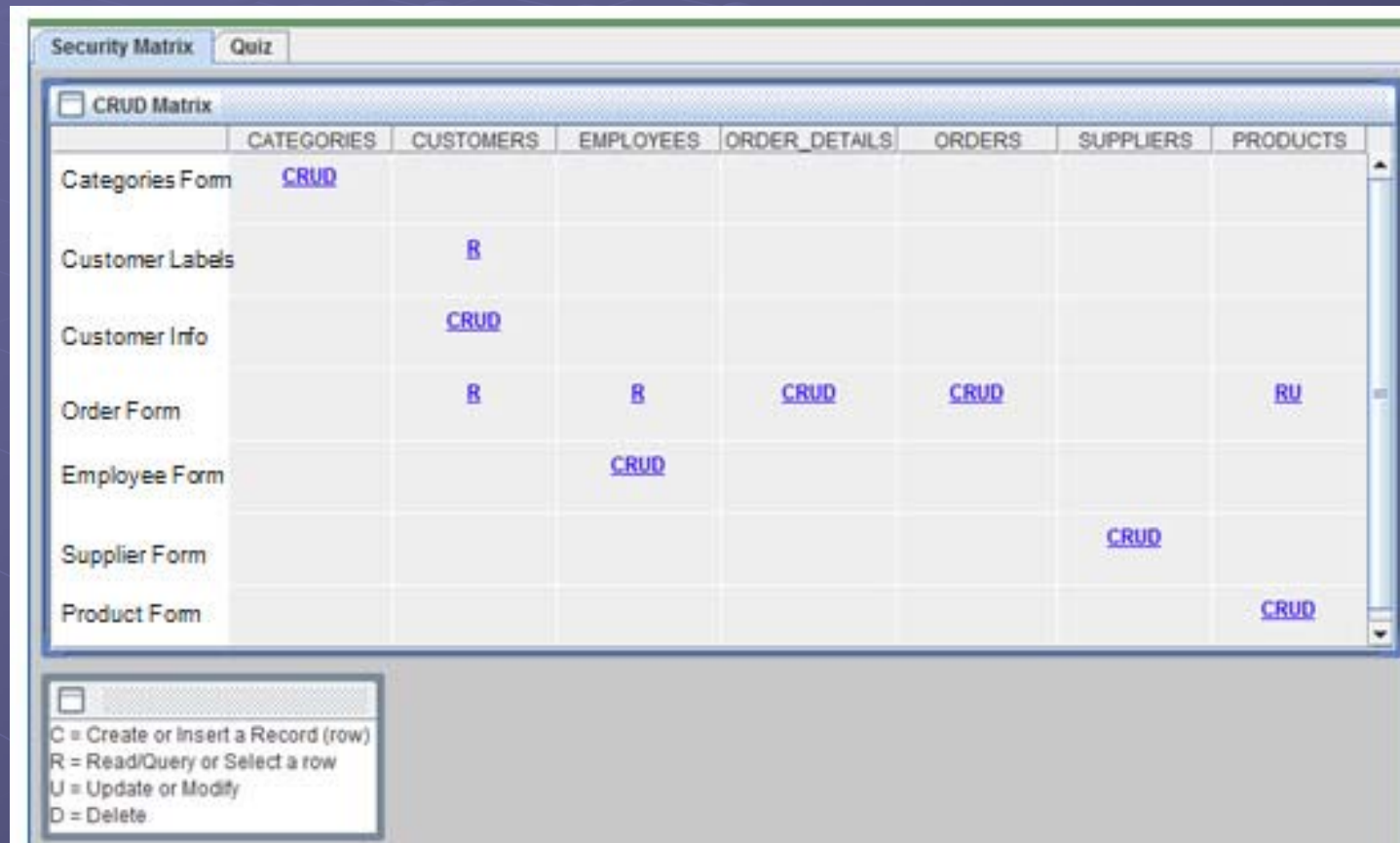
- Se poate construi o vizualizare care execută o instrucțiune SELECT care returnează anumite linii, pe baza unei anumite condiții (de exemplu, legată de utilizator)
- Exemplu:

```
CREATE VIEW viz AS  
SELECT *  
FROM tabel  
WHERE atribut = USER;
```

Accesul aplicațiilor

- Mulți utilizatori nu accesează baza de date printr-o conexiune directă la aceasta, ci prin intermediul unei aplicații
- Un instrument simplu, numit matrice de securitate (CRUD), poate fi utilizat pentru a identifica în mod explicit drepturile de acces necesare programului de aplicație
- Reprezentare vizuală a corelației dintre operații sau autorizări și sursele de intrare/ieșire (de exemplu, forme și rapoarte)

Accesul aplicațiilor



The screenshot shows a software window titled "Security Matrix" with a "Quiz" tab. Inside, there is a "CRUD Matrix" section. The matrix has 8 rows (Forms) and 7 columns (Data Objects). The rows are: Categories Form, Customer Labels, Customer Info, Order Form, Employee Form, Supplier Form, and Product Form. The columns are: CATEGORIES, CUSTOMERS, EMPLOYEES, ORDER_DETAILS, ORDERS, SUPPLIERS, and PRODUCTS. The cells contain CRUD permissions: Categories Form (C, R, U, D), Customer Labels (R), Customer Info (C, R, U, D), Order Form (R, R, C, R, U, D), Employee Form (C, R, U, D), Supplier Form (C, R, U, D), and Product Form (C, R, U, D). A legend at the bottom left defines the letters: C = Create or Insert a Record (row), R = Read/Query or Select a row, U = Update or Modify, and D = Delete.

	CATEGORIES	CUSTOMERS	EMPLOYEES	ORDER_DETAILS	ORDERS	SUPPLIERS	PRODUCTS
Categories Form	<u>CRUD</u>						
Customer Labels		<u>R</u>					
Customer Info		<u>CRUD</u>					
Order Form		<u>R</u>	<u>R</u>	<u>CRUD</u>	<u>CRUD</u>		<u>RU</u>
Employee Form			<u>CRUD</u>				
Supplier Form						<u>CRUD</u>	
Product Form							<u>CRUD</u>

☐ C = Create or Insert a Record (row)
R = Read/Query or Select a row
U = Update or Modify
D = Delete

Accesul aplicațiilor

- Alt avantaj al matricii de securitate: descrie vizual regulile de integritate
 - Simplifică identificarea tuturor aplicațiilor care pot fi afectate de o modificare adusă unui tabel al bazei de date
 - De exemplu, suprimarea unei coloane din tabelul Produse va avea efect asupra formei corespunzătoare, generând eroare la execuția acestor aplicații
 - Înaintea modificării, trebuie cunoscut impactul și astfel pot fi determinate aplicațiile care necesită actualizări

Vulnerabilitatea bazelor de date

- Breșele de securitate sunt un fenomen tot mai frecvent întâlnit
- Din ce în ce mai multe baze de date devin accesibile prin intermediul Internetului și al aplicațiilor web
- ➡ Expunerea acestora la amenințările de securitate crește
- Obiectivul este de a reduce vulnerabilitatea în fața acestor amenințări

Vulnerabilitatea bazelor de date

- Una dintre cele mai cunoscute vulnerabilități ale aplicațiilor de baze de date este SQL Injection
- SQL Injection include problematica riscurilor inerente intrărilor nevalidate
- Are loc atunci când sunt create în mod dinamic instrucțiuni SQL pe baza intrărilor furnizate de utilizator
- Amenințarea apare când utilizatorii introduc cod care conduce la executarea unor comenzi neautorizate
- Vulnerabilitatea apare din cauza trăsăturilor limbajului SQL care permite comentarii în cadrul instrucțiunilor (--), concatenarea instrucțiunilor SQL separate prin ; și capacitatea de a interoga metadate din dicționarul datelor
- Soluția: validarea intrărilor

Vulnerabilitatea bazelor de date

- Exemplu: Proces login pe o pagină web care validează un username și o parolă pe baza datelor dintr-o bază de date relațională
- Pagina web are un formular de autentificare
- Textul scris de utilizator este folosit la crearea dinamică a unei instrucțiuni SQL de căutare de înregistrări în baza de date
- Un utilizator răuvoitor poate introduce text care conduce la dobândirea accesului la date asupra cărora nu deține privilegii
- De exemplu, următorul string: ' OR 1=1 - - introdus în formularul de autentificare conduce la accesul în sistem fără a cunoaște un nume de utilizator și o parolă
- Motivul: aplicația generează o interogare dinamică formată prin concatenarea anumitor șiruri de caractere cu valorile introduse de către utilizator

Vulnerabilitatea bazelor de date

- Exemplu:

```
SELECT count(*) FROM users  
WHERE user_name = 'conținut_username_textbox'  
AND password = 'conținut_password_textbox';
```

- La introducerea unui username și a unei parole
interogarea devine:

```
SELECT count(*) FROM users  
WHERE user_name = 'user1'  
AND password = 'pass1';
```

- Dacă un utilizator introduce șirul de caractere 'OR 1=1 -
- interogarea devine:

```
SELECT count(*) FROM users  
WHERE user_name = " OR 1 = 1 - - "  
AND password = ";
```

Vulnerabilitatea bazelor de date

- Expresia $1 = 1$ este adevărată pentru fiecare linie din tabel determinând clauza OR să returneze true.
- Caracterele `--` comentează restul comenzii SQL
- Cererea va returna un rezultat mai mare decât 0, adică există cel puțin o linie în tabelul cu utilizatori, ceea ce conduce la un login cu succes

Vulnerabilitatea bazelor de date

- Alt tip de SQL Injection apare atunci când sistemul permite procesarea cererilor din stivă. Acestea presupun execuția a mai mult de o cerere în cadrul unui singur apel de funcție dintr-o aplicație
- Exemplu: Inițial, utilizatorul are permisiunea de a selecta attributele produselor din tabelul Products. Utilizatorul injectează o cerere stivuită care încorporează o cerere SQL adițională. Aceasta din urmă șterge tabelul Customers.

SELECT * FROM products;

DROP TABLE customers;

- Șirul acesta, transmis ca instrucțiune SQL, va conduce la execuția a 2 cereri. Se vor lista produsele, iar adițional va fi suprimat tabelul Customers, fiind pierdute toate datele acestuia.

Vulnerabilitatea bazelor de date

- În sistemele care nu permit cereri stivuite sau care invalidează șirurile SQL care conțin ; aceste cereri nu sunt executate.
- SQL Injection poate fi prevenit prin validarea intrărilor utilizatorului.
- Sunt utilizate 3 abordări pentru validarea șirurilor de caractere reprezentând cereri: folosirea unei liste negre, a unei liste albe sau implementarea cererilor parametrizate.
- Lista neagră parsează șirul de intrare comparând fiecare caracter cu o listă de caractere nepermise. Dezavantaj: multe caractere speciale pot fi legitime, dar vor fi respinse (exemplu: utilizarea unui apostrof în nume).
- Lista albă este similară celei negre, dar comparația se face cu o listă de caractere permise. Abordarea este preferată celei anterioare, dar trebuie făcute niște considerații speciale referitoare la apostrof.
- Cererile parametrizate utilizează parametri definiți intern pentru a completa o instrucțiune SQL pregătită anterior
- Validarea intrărilor este unul dintre mecanismele primare de apărare pentru prevenirea vulnerabilităților din baza de date, inclusiv SQL injection.

Inferența în baze de date

- O vulnerabilitate subtilă în tehnologiile bazelor de date
- Abilitatea de a deduce informație necunoscută pe baza celei regăsite
- Problema: nu există soluții ideale
- Singurele soluții acceptate includ control legat de cereri (suprimare) sau control legat de articole individuale din baza de date (ascundere)
- Datele sensibile solicitate fie nu sunt furnizate, fie răspunsurile date sunt apropiate, dar nu exacte

Inferența în baze de date

- Are loc în cazuri în care scopul inițial este acela de a genera sau vizualiza valori agregat, în condițiile în care nu au fost acordate privilegii pentru obținerea valorilor individuale
- Pe baza valorilor agregat se pot deduce uneori valorile individuale
- Exemplu: un angajat dorește să afle salariul colegului său X, care este confidențial. Angajatul are drepturi de a genera date agregat (media salariilor în funcție de anumite criterii) .
 - Presupunem că X este femeie și are 8 subalterni
 - Pe baza acestor informații, angajatul poate deduce o funcție agregat:

```
SELECT AVG(salary) FROM employees  
WHERE gender = 'F' and dependants = 8;
```
 - Se obține salariul lui X

Inferența în baze de date

- Inferența poate apărea și în cazul în care utilizatorii pot stabili informații din datele accesibile lor, la nivelul lor de securitate, chiar dacă informația este protejată la un nivel mai înalt de securitate.
- Exemplu: Anumite date (referitoare la prototipurile produselor companiei) nu sunt accesibile angajaților juniori. Acești angajați au dreptul de a actualiza tabelul Storage care înregistrează conținutul zonelor de depozitare ale companiei, dar nu pot citi liniile referitoare la prototipuri.
 - Dacă angajatul încearcă să modifice o linie protejată, va apărea un mesaj de eroare
 - Mesajul indică faptul că acea informație este ascunsă și poate deduce că un produs de tip prototip este stocat în compartimentul referit în comanda de actualizare
- Posibilă soluție: poliinstanțierea
- Aceasta permite ca baza de date să rețină mai multe înregistrări având aceeași cheie primară; ele se disting printr-un identificator al nivelului de securitate

Inferența în baze de date

- Dezvoltarea de soluții tehnologice pentru a detecta inferența este o problemă complexă
- Multe lucrări din acest domeniu propun revocarea accesului la anumite obiecte pe baza istoricului utilizatorului
- Problema cu detectarea inferenței: conduce la o întârziere semnificativă între momentul executării cererilor și cel al prezentării rezultatelor
- Alte abordări referitoare la atenuarea vulnerabilităților accentuează necesitatea compromisurilor.
- Protecția datelor sensibile necesită examinarea situațiilor care pot conduce la expunerea în fața utilizatorilor neautorizați, precum și a politicilor de monitorizare care trebuie implementate pentru a asigura obținerea unor răspunsuri adecvate.