SIGURNOST BAZE PODATAKA

Dumančić, Robert

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni studij

SIGURNOST BAZE PODATAKA

Završni rad

Robert Dumančić

Osijek, 2019.

Sadržaj

1. UVOD

Baza podataka se može definirati kao organizirani skup podataka spremljen na tvrdom disku računala. Omogućuju svakom ovlaštenom korisniku brz i lagan pristup, unos i analizu podataka. Uzimajući u obzir značaj podataka unutar tvrtke ili organizacije, neophodno je osigurati podatke unutar njene baze podataka.

Sigurnost baza podataka obuhvaća metode zaštite baze podataka od neovlaštenog pristupa, preinaka ili uništenja podataka. Uz zaštitu samih podataka, mora se voditi računa i o privatnosti pojedinaca čije se podatke sprema.

Cilj ovog rada je dati uvid u najčešće sigurnosne propuste pri uspostavi baze podataka, pregled sigurnosnih metoda kojima se sprječavaju takvi propusti i primjenu sigurnosnih metoda u nekim zastupljenijim sustavima za upravljanjem bazama podataka.

Za sigurnosne prijetnje proći će se kroz temeljne sigurnosne ciljeve (kao i probleme koji nastaju kada nisu ispunjeni) kao i najčešće oblike namjernog i nenamjernog proboja sigurnosti baze podataka.

Pri pregledu sigurnosnih metoda proći će se kroz razne metode osiguravanja metoda kako ne bi došlo do navedenih propusta, kao i posebne sigurnosne metode pojedinih sustava za upravljanje bazama podataka.

1.1. Zadatak završnog rada

Zadatak ovog rada dati uvid u ranjivosti baza podataka kao i metode zaštite baza podataka kao što su dodjeljivanje primjerenih ovlasti i dozvola pristupa, korisnički računi i zaporke i korištenje enkripcije. Na kraju se analiziraju i dodatne metode koje koriste neki sustavi za upravljanje bazama podataka kao što su Oracle i Microsoft Access.

2. SIGURNOSNE PRIJETNJE

Prijetnja je svaki događaj, namjeran ili nenamjeran, koji može utjecati na sustav i organizaciju. Bilo da je prijetnja namjerna ili ne, utjecaj može biti jednak. Ranjivost je slaba točka u sustavu, poput neprimjerene kontrole pristupa ili rupa u vatrozidu, koja omogućava pojavu prijetnje. Šteta koju prijetnja proizvede može biti materijalna poput štete sklopovlja, programske opreme ili podataka, ali može biti i nematerijalna kao što je gubitak vjerodostojnosti i klijentskog povjerenja.

Plan zaštite baze podataka treba sadržavati proceduru procjene rizika koja identificira prijetnje i ranjivosti i uspostavlja prikladne kontrole kako bi se ostvarili sigurnosni ciljevi.

2.1. Sigurnosni ciljevi

Sigurnosni ciljevi su definirani po CIA (engl. *Confidentiality Integrity Availability*) modelu. Kršenjem ijedne stavke CIA modela baza podataka postaje nesigurna.

Gubitak povjerljivosti - povjerljivost baze podataka se odnosi na zaštitu podataka od nedozvoljenog razotkrivanja. Rezultat nedozvoljenog razotkrivanja podataka može biti u rasponu od kršenja zakona o zaštiti osobnih podataka do nacionalne sigurnosti. Takvi propusti u sigurnosti mogu dovesti do gubitka povjerenja javnosti ili čak sudskog postupka protiv organizacije.

Gubitak integriteta – integritet baze podataka se odnosi na zahtjev da su podatci zaštićeni od neprikladnih izmjena. Izmjene podataka uključuju stvaranje, umetanje, ažuriranje, mijenjanje stanja podataka i brisanje. Integritet se gubi ako dođe do neodobrenih izmjena. Ako se gubitak sustava ili integritet podataka ne ispravi, daljnja uporaba sustava može dovesti do netočnosti, pogrešnih odluka ili prevare (ako je netko namjerno netočno izmijenio podatke).

Gubitak raspoloživosti – raspoloživost baze podataka se odnosi na pružanje usluge i podataka osobi ili programu koji imaju prava na nju. Ponekad zvan uskraćivanje usluge (engl. *denial of service*). Kada baza podataka nije dostupna nastaje gubitak, pa se svaka prijetnja koja povećava vrijeme tijekom kojeg je baza podataka nedostupna izbjegava.

2.2. Nenamjerne sigurnosne prijetnje

Pod nenamjernim sigurnosnim prijetnjama spadaju proboji sigurnosti koji su nastali slučajno, bez namjere da se nanese zlo organizaciji.

Primjeri takvih proboja su:

- korisnik može nenamjerno napraviti zahtjev za objektom ili operacijom za koji ne bi trebao imati odobrenje, ali se zahtjev odobrava zbog propusta u dodjeli ovlasti
- pogreška u komunikacijskom sustavu može spojiti korisnika u sjednicu koja pripada drugom korisniku s drugačijim ovlastima pristupa.
- osoba može slučajno primiti poruku koja je namijenjena drugom korisniku što može dovesti do gubitka povjerljivosti u bazi podataka.
- operacijski sustav može slučajno prepisati preko podataka i uništiti dio baze podataka, dohvatit krive podatke i onda ih nesvjesno poslati korisniku ili zakazati u brisanju podataka koje je trebalo uništiti.

2.3. Namjerne sigurnosne prijetnje

Namjerne sigurnosne prijetnje nastaju kada korisnik namjerno dobije neovlašten pristup i izvodi neovlaštene operacije nad bazom podataka. Takvi napadi se izvode s ciljem prouzrokovanja štete organizaciji.

Primjeri takvih proboja su:

- nezadovoljni zaposlenik upoznat s organizacijskom bazom podataka želi nauditi organizaciji
- industrijski špijun koji pokušava pristupiti podatcima za konkurenciju
- ovlašteni korisnici poput administratora baze podataka pristupljuju privatnim podatcima krajnjih korisnika za koje ni bi trebali imati ovlasti

2.4. Tipovi sigurnosnih prijetnji

Iako bi se za sigurnost baze podataka trebalo krenuti od fizičkog aspekta sigurnosti kao što su sigurnost zgrade organizacije i provjera zaposlenika i osoba koja pristupaju organizaciji, ovaj rad se neće baviti stvarima koji su daleko izvan opsega dizajnera baze podataka.

1. Prekomjerne ovlasti

   Davanje prekomjernih ovlasti stvara dva problema. Jedan je da većinu napada na podatke neke organizacije obavlja trenutni zaposlenik ili bivši zaposlenik. Dodjeljivanje prekomjernih ovlasti ili ne uklanjanje takvih na vrijeme samo nepotrebno olakšava počiniteljima prijestup. Drugi je da bi štetne radnje korisnik mogao obaviti nesvjesno bez znanja da čini nešto što ne bi trebao.

2. SQL ubrizgavanje

Ubrizgavanje naredbi je napad kojemu je cilj iskriviti originalnu namjenu SQL naredbe tako što je zamjeni s SQL naredbama napadača.

Termin ubrizgavanje se odnosi na činjenicu da se kroz korisnički unos na sučelju aplikacije mogu zlouporabiti ranjivosti povezane s dinamičkim stvaranjem SQL upita. Koristeći SQL ubrizgavanje, napadač može unijeti svoje linije koda u upit koje se onda mogu iskoristiti za dohvaćanje podataka za koje napadač nema ovlasti, brisanje ili izmjenu podataka ili unos podataka koje bi napadaču dalo neograničen pristup bazi podataka.

```
SELECT *
FROM users
WHERE username = 'X' AND password = 'Y' OR 'a'='a';
```

Na isječku koda 2.1 se nalazi primjer SQL ubrizgavanja gdje čak i s netočnim korisničkim podatcima i zaporkom ovaj će se upit smatrati točnim zbog *OR* uvjeta koji će uvijek biti ispunjen i time će upit vratiti podatke o svim korisnicima.

Isječak koda 2.1. Primjer SQL upita kojim se vrši ubrizgavanje

3. Zlonamjerni programi (engl. *Malware*)

   Tipovi napada poput *spear phishing* elektroničkih poruka pokušavaju prodrijeti unutar organizacije i ukrasti njihove osjetljive podatke. Mogu sadržavati crve i viruse koji ometaju normalan rad baze podataka ili joj prouzroče štetu. Nesvjesni da je zlonamjerni program zarazio njihove uređaje, korisnici postaju sredstvo prijenosa takvog programa.

4. Slab revizijski trag

   Automatizirano praćenje transakcija baze podataka koje uključuje osjetljive podatke bi trebalo biti uključeno u implementaciju bilo koje baze podataka. Neuspješno skupljanje detaljnih revizijskih zapisa aktivnosti baze podataka stvara ozbiljnu organizacijsku prijetnju. Većina revizijskih mehanizama nema podatke o krajnjem korisniku već samo njihovo korisničko ime. Takvi mehanizmi otežavaju pronalazak napadača jer ne postoji poveznica sa osobom odgovornom za napad. Jednako tome, korisnici s administrativnim privilegijima mogu isključiti reviziju baze podataka kako

bi prikrili svoje nelegalne aktivnosti. Revizijski trag je potreban kako bi se znalo tko je napravio što, tko je pokušao napraviti što i gdje i kada se sve to dogodilo.

5. Izloženost sigurnosne kopije

   Često su mediji za pohranu sigurnosne kopije podataka potpuno izložene napadima. Takvi propusti mogu dovesti do krađe kompletne baze podataka sa sigurnosnih kopija a ne sa same baze podataka. Zbog toga su primjene sigurnosnih metoda na sigurnosne kopije ne samo poželjne već i potrebne.

6. Slaba provjera autentičnosti

   Slaba provjera autentičnosti omogućava napadaču da preuzme identitet ovlaštenog korisnika baze podataka. Implementacija korisničkih zaporki i autentifikacije u dva koraka su obavezne.

7. Pogrešna konfiguracija baze podataka

   Često se mogu pronaći baze podataka koje nisu nadograđene ili koje još uvijek koriste zadane korisničke račune i konfiguracijske parametre. Napadači znaju iskoristiti takve ranjivosti kako bi lansirali napad na organizaciju. Ponekada čak i kada su nadogradnje dostupne, organizacije ih ne uspijevaju implementirati. Problemi nastaju zbog složenih dugotrajnih zahtjeva za testiranje nadogradnji i pronalaska vremenskog prozora u kojemu se sustav može isključiti i nadograditi. Rezultat je takav da organizacijama ponekad treba i nekoliko mjeseci kako bi se baze podataka nadogradile tijekom kojih su ranjive.

8. Nekontrolirani osjetljivi podatci

   Mnoge organizacije ne održavaju točan inventar svojih baza podataka i ključnih podataka unutar njih. Zaboravljene baze podataka mogu sadržavati osjetljive podatke i nove baze podataka (kao što su one koje nastaju tijekom testiranja) mogu nastati bez znanja sigurnosnog odjela. Osjetljivi podatci u takvim bazama podataka mogu predstavljati prijetnje ako ih se ne kontrolira.

9. Uskraćivanje usluge

   Uskraćivanje usluge je generalizirana kategorija u kojoj je pristup mreži ili podatcima odbijen korisniku. Takvi napadi su izvedeni tako da se ciljani uređaj preplavi suvišnim zahtjevima kako bi se sustav preopteretio.

10. Ograničena edukacija i stručnost

Mnoge organizacije su slabo opremljene za rješavanje sigurnosnog proboja kao što im i unutrašnje sigurnosne kontrole ne drže korak s tempom kojim im podatci rastu. Često je ovo rezultat manjka znanja o implementaciji sigurnosnih kontrola, provođenju strategija ili provođenju reakcija na incidente.

3. KONTROLA PRISTUPA

Kontrola pristupa bazi podataka je proces osiguravanja podataka i ostalih resursa od neovlaštenog pristupa. Bazira se na davanju i oduzimanju privilegija nad resursima. Privilegije omogućuju korisniku stvaranje ili pristup (poput pisanja, čitanja ili izmjene) objektu baze podataka ili korištenje usluga sustava za upravljanje bazom podataka. Sustav za upravljanje bazom podataka bilježi svo oduzimanje i davanje privilegija korisnicima i osigurava da će samo ovlašteni korisnici moći pristupiti objektu baze podataka.

Pri planiranju baze podataka, administrator može koristit alate poput matrice kontrole pristupa prikazane tablicom 3.1. Stupci u takvoj matrici predstavljaju objekte baze podataka poput tablica, pogleda, modula i sličnih kategorija ovisnih o modelu baze podataka i sustavu za upravljanje. Redovi predstavljaju uloge, pojedince ili grupe korisnika. U samim ćelijama se nalaze privilegiji dani određenoj grupi nad pripadajućim objektom. Tipovi privilegija ovise o korištenom sustavu, ali uglavnom sadrže vrijednosti poput *READ*, *INSERT*, *UPDATE*, *DELETE*, *EXECUTE* i *CREATE*.

Tablica 3.1. Primjer matrice kontrole pristupa

	STUDENT TABLE	**GRADES TABLE**	**FACULTY TABLE**
STUDENT	READ	READ	
ASSISTANT	READ	READ, INSERT	READ
PROFESSOR	READ	READ, INSERT, UPDATE, DELETE	READ

Administrator baze podataka može stvarati i mijenjati strukturu baze podataka, kao i upravljati privilegijima. Neki sustavi omogućuju administratoru i delegaciju svojih privilegija, što omogućava i drugim korisnicima upravljanje bazom podataka. Takvo ponašanje može dovesti do velikog sigurnosnog propusta, jer takvi korisnici mogu odobriti i drugim korisnicima upravljanje bazom podataka. Broj korisnika s prevelikim ovlastima se tako brzo može povećati što administratoru otežava oduzimanje privilegija.

3.1. Provjera autentičnosti

Provjera autentičnosti je proces utvrđivanja identiteta osobe kako bi se uvjerilo da je korisnik onaj za koga tvrdi da je. Započinje na razini operacijskog sustava, kada se korisnik prijavljuje, unosi svoje korisničko ime koje se provjerava za autentičnost. Sustav ima korisnički profil za to korisničko ime koji pruža detaljne informacije o samom korisniku. Uz korisničko ime profil sadrži i zaporku. Zaporka se treba čuvati tajnom i mijenjati često. Sustav nikada ne bi trebao prikazati zaporku pri prijavljivanju i korisnički profili se trebaju čuvati sigurnima u kriptiranom obliku. Još jedna sigurnosna mjera bi bila zaključavanje korisnika od pristupa nakon nekoliko neuspješnih pokušaja prijavljivanja. Time se sprječava pokušaj pogađanja zaporke ili njeno otkrivanje na silu.

Iako su zaporke najraširenija metoda provjere autentičnosti, nisu vrlo sigurne, pošto ih korisnici često zapisuju negdje ili biraju jednostavne zaporke koje se daju brzo razbiti ili se mogu logički zaključiti (poput datuma rođenja). Tada se može primijeniti provjera autentičnosti s više parametara. Od korisnika se onda zahtjeva dvije ili više metode provjere autentičnosti kao što su zaporka i nekakav oblik biometrije. Iako se provjera autentičnosti većinom odvija na razini operacijskog sustava, poželjno ju je provoditi i na razini baze podataka. U najmanju ruku uvesti dodatnu zaporku koja se unosi pri pristupu bazi podataka.

Provjera autentičnosti ne daje nikakve privilegije korisnicima. Ona samo uspostavlja povjerenje da je korisnik koji pokušava pristupiti sustavu za upravljanje bazom podataka onaj za kojeg tvrdi da je i da je sustav kojem korisnik pokušava pristupiti ispravan. Provjera autentičnosti je samo preduvjet pristupu bazi podataka.

3.2. Korisničke ovlasti

Korisničke ovlasti nad bazom podataka su određene privilegijama koje su dane samom korisniku ili ulozi kojoj pripada. Privilegija je radnja poput stvaranja, uređivanja ili brisanja koju korisnik smije izvesti na objektu baze podataka. U standardnom SQL-u, administrator baze podataka ima sve privilegije i može ih prenijeti na druge korisnike. Forma za davanje privilegija je prikazana na isječku koda 3.1. Ako se u ON izjavi nalazi tablica, onda *ALL PRIVILEGES* sadrži *SELECT*, *DELETE*, *INSERT*, *UPDATE* i *REFERENCES* privilegije. Ako je naveden pogled, onda se provjerava je li pogled moguće ažurirati. Za poglede koje se ne može ažurirati daje se samo *SELECT* privilegije, dok za one koje mogu daju se *SELECT*, *INSERT*, *DELETE* i *UPDATE*

```
GRANT {ALL PRIVILEGES | privilege list}
ON {object name}
TO {PUBLIC | user-list | role-list} [WITH GRANT OPTION];
```

privilegije.

Isječak koda 3.1. Forma za davanje privilegija

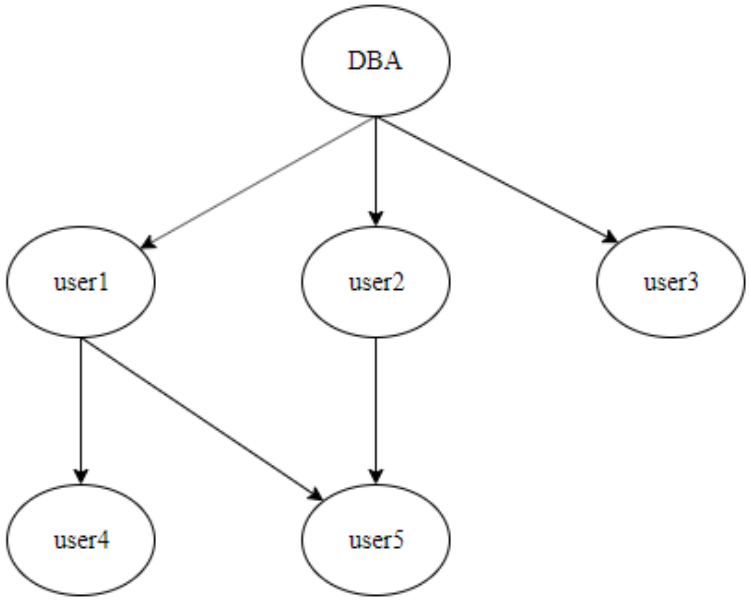
U *TO* izjavi se navode sve uloge, korisnici ili grupe kojima se predaju privilegije. Neobavezna *WITH GRANT OPTION* opcija označava da će navedeni korisnici moći dobivene ovlasti predati

```
GRANT UPDATE ON Employees TO user1, user2, user3 WITH GRANT
OPTION;
```

drugim korisnicima. Na isječku koda 3.2. se vidi primjer davanja trima korisnicima privilegije za izmjenu podataka u tablici uz mogućnost predaje tih ovlasti drugim korisnicima.

Isječak koda 3.2. Primjer predaje privilegija trima korisnicima

Navedena tri korisnika mogu proslijediti svoje privilegije drugima, pa sustav mora voditi računa o svim dodijeljenim ovlaštenjima preko dijagrama autorizacije prikazanog na slici 3.1.



Slika 3.1. Primjer dijagrama autorizacije

Korijen takvog grafa je uvijek administrator baze podataka. Čvorovi grafa predstavljaju kombinaciju privilegije i korisnika. Strelice označavaju dodjeljivanje privilegija od jednog korisnika do drugog. Pri dodjeljivanju drukčije privilegije korisniku u graf se dodaje novi čvor koji predstavlja drugu privilegiju.

Ovi podatci su stalno dostupni administratoru baze podataka kako bi mogao poništiti dodijeljene privilegije po potrebi.

Dodjeljivanje korisničkih ovlasti se može znatno olakšati uporabom uloga. Jednom kada administrator baze podataka definira ulogu, niz privilegija joj se pridružuju i korisnički računi joj se dodjeljuju.

```
CREATE ROLE EmployeeRole;
GRANT SELECT ON Employees TO EmployeeRole;
GRANT EmployeeRole TO user1;
```

Korisnički računi nisu ograničeni na samo jednu ulogu već ih mogu imati nekoliko. Uloge se mogu i dodijeliti drugoj ulozi kako bi uspostavili nasljeđivanje privilegija kroz uloge. Prikaz stvaranja uloge, dodjeljivanja privilegija i korisnika se nalazi na isječku koda 3.3.

Isječak koda 3.3. Rad s ulogama

Uklanjanje privilegija se vrši pomoću *REVOKE* naredbe. Navedena naredba se ne koristi kako bi korisniku nešto zabranili, već kako bi mu uklonili privilegije koje su mu prethodno dane. Na isječku koda 3.4 se nalazi forma *REVOKE* naredbe a na isječku koda 3.5 primjer izvedbe

```
REVOKE {ALL PRIVILEGES | privilege-list}
ON {object-list}
FROM {PUBLIC | user-list | role-list} [CASCADE | RESTRICT];
```

naredbe kojom poništavamo privilegiju danu jednom korisniku iz isječka koda 3.2.

Isječak koda 3.4. Forma REVOKE naredbe

Isječak koda 3.5. Primjer uporabe REVOKE naredbe

Ako se korisniku oduzme određeni privilegij, svim korisnicima koji su dobili taj privilegij od tog

```
REVOKE UPDATE ON Employees FROM user1;
```

korisnika će ga izgubiti zajedno s njim. To se događa zbog *CASCADE* opcije, koja je podrazumijevana ne navede li se drukčije. Tako će u primjeru sa slike 3.1, uklonimo li privilegij prvom korisniku, taj privilegij izgubiti i četvrti korisnik dok će ga peti korisnik zadržati jer ga je dobio i od drugog korisnika. Koristi li se *RESTRICT* opcija pri oduzimanju privilegija, sustav provjerava hoće li ta radnja oduzeti privilegije i korisnicima koji nisu navedeni u samom pozivu

naredbe. Ako hoće, naredba vraća grešku i ne izvodi se. Pri uklanjanju privilegija, dijagram autorizacije se automatski izmjenjuje brisanjem odgovarajućih čvorova.

3.3. Pogledi

Pogled je virtualna relacija koja ne postoji u bazi podataka već se stvara kada određeni korisnik napravi upit nad tablicom. Korisnik će imati odobren pristup pogledu, ali ne i izvornoj tablici po kojoj je pogled baziran. Pogled skriva dijelove baze podataka od određenih korisnika i oni nisu ni svjesni o postojanju dodatnih atributa ili unosa koji nedostaju iz pogleda. Tako se pojednostavljuje prikaz korisniku i štite se osjetljivi podatci od razotkrivanja.

Pogledi mogu biti ovisni o atributu ili neovisni o atributima. Navođenjem ograničenja unutar *WHERE* linije u *SELECT* izjavi se stvara pogled ovisan o atributu. Izostavljanjem *WHERE* linije

```
CREATE VIEW NightShiftView AS
     SELECT firstName, surname
     FROM Employees
     WHERE shift= 'Night';
```

nastaju pogledi neovisni o atributima. Takvi pogledi se vide na primjerima 3.6. i 3.7.

Isječak koda 3.6. Pogled ovisan o atributu

Isječak koda 3.7. Pogled neovisan o atributu

Pogled neovisan o atributu će ispisati sve zaposlenike iz tablice dok će pogled ovisan o atributu sadržavati samo one koji zadovoljavaju zadano ograničenje.

```
CREATE VIEW EmployeeView AS
     SELECT firstName, surname
     FROM Employees;
```

3.4. Revizijski trag

Revizija je proces praćenja i bilježenja svih operacija nad bazom podataka od strane svih korisnika. Revizijski trag je popis svih korisničkih radnji nad bazom podataka. Ako korisnici uspješno dokažu autentičnost i pokušaju pristupiti resursu, sve uspješne i neuspješne pokušaje pristupa i njihov status prati sustav i zapisuje ih u datoteke revizijskog traga. Kada se neovlaštena radnja otkrije, administrator baze podataka može odrediti korisnički račun s kojega

se radnja odvila. Uz to, zapis će sadržavati i podatke o korisniku koji je pokušao pristup podatku, samu radnju koju je pokušao izvesti, korištenu radnu stanicu, vrijeme pokušaja, informacije o podatku, njegovu staru vrijednost i novu vrijednost ako je pokušaj bio uspješan. Revizijski trag je iznimno bitan za osjetljive baze podataka koje ažuriraju brojni korisnici poput bankovnih ili bolničkih baza podataka.

Okidači se mogu postaviti nad tablicom u svrhu praćenja svih promjena, vremena kada su uspostavljene i identiteta osobe koji ih je napravio. Na primjeru 3.8. se nalazi jednostavan

```
TableAudit( datetimeOfUpdate, userID, oldValue, newValue)
```

primjer uspostavljanja tablice koja će držati revizijske zapise.

Isječak koda 3.8 Uspostavljanje tablice za držanje revizijskih zapisa

Okidač bi trebao unijeti novi zapis u *TableAudit* tablicu kada korisnik pokuša promijeniti

```
CREATE OR REPLACE TRIGGER TableAuditTrigger
     BEFORE UPDATE OF Value ON Table
     FOR EACH ROW
     BEGIN
          INSERT INTO TableAudit
               VALUES(SYSDATE,USER,              :OLD.Value,
:NEW.Value);
```

vrijednost *Value* atributa u izvornoj tablici. Prikaz uspostave okidača je dan na isječku koda 3.9.

Isječak koda 3.9. Uspostavljanje okidača

Primjer koristi *SYSDATE* i *USER*, koji su određeni kao pseudo atributi u Oracle. Oba djeluju kao funkcije koje vraćaju pripadne vrijednosti. *SYSDATE* vraća trenutni datum i vrijeme, dok *USER* vraća identifikacijski broj trenutnog korisnika.

3.5. Kontrolne mjere statističkih baza podataka

Statističke baze podataka su dizajnirane za pružanje podataka o raznim populacijama kako bi se na njima mogla vršiti statistička obrada. Sami podatci mogu sadržavati podatke o pojedincima, ali se podatci ne smiju prikazati za jednu osobu. Korisnicima je dopušteno pretraživati statističke podatke o populaciji kao što su minimum, maksimum, prosjek i slično. Kontrolne mjere

```
SELECT COUNT(Employees)
FROM Employees
WHERE shift = 'Night';
```

statističkih baza podataka moraju onemogućiti dohvaćanje podataka o specifičnoj osobi. Primjer dozvoljenog upita je na isječku koda 4.1. dok je na isječku koda 4.2. prikaz upita koji je potrebno zabraniti.

Isječak koda 4.1. Prikaz dopuštenog upita nad statističkom bazom podataka

Isječak koda 4.2. Prikaz nedopuštenog upita nad statističkom bazom podataka

To se može postići zabranom upita koji vraćaju vrijednosti atributa i samo dozvoljavanjem upita

```
SELECT Salary
FROM Emplyoees
WHERE surname = 'James';
```

koji sadržavaju statističke agregatne funkcije poput *COUNT*, *SUM*, *MIN*, *MAX*, *AVERAGE* i *STANDARD DEVIATION*. Takvi upiti se nazivaju statistički upiti.

Posebne mjere opreza se moraju primijeniti kako bi se spriječila mogućnost logičke dedukcije podataka o korisnicima. Ako ne postoje druga ograničenja osim tog da se smiju postavljati samo statistički upiti, korisnici mogu dodavati uvjete u upit dok se rezultat ne svede na samo jednu osobu. Primjer takvog upita je dan na isječku koda 4.3.

```
SELECT SUM(Salary)
FROM Employees
WHERE shift = 'Night' AND job = 'Programmer' AND dateHired >
'2019-01-01';
```

Isječak koda 4.3. Prikaz upita kojim se sužava rezultat na jednu osobu

Mogućnost izvlačenja informacija o pojedincu je smanjena ako se sustav podesi na način na koji odbija prikazati rezultate upita kojima je rješenje jedan zapis. Iako, takvo ograničenje se lagano zaobiđe. Tada osoba može zatražiti prikaz svih plaća u određenom odjelu i onda isti taj prikaz bez tražene osobe. Oba upita vraćaju rezultat s više rješenja, ali se iz njih lagano da zaključiti plaća određene osobe.

Kako bi spriječili zaključivanje informacija o pojedincu sustav može ograničiti upite zahtijevanjem da broj rezultata mora biti iznad određene granice. Ujedno se može i primijeniti zabrana niza upita koji stalno vraćaju iste rezultate. Moguće je i namjerno dodati neznatne netočnosti u rezultate kako bi se otežalo izdvajanje pojedinaca iz populacije. Prva nabrojana

tehnika se može primijeniti kao particije. Particije podrazumijevaju da su unosi spremljeni u grupama određene minimalne veličine. Tada upiti mogu vraćati cijelu grupu ili više grupa, ali nikada pojedinačne unose sadržane u grupi.

3.6. Kontrola toka

Kontrola toka regulira distribuciju ili tijek informacija između dostupnih objekata. Tok se odvija između objekta A i objekta B kada program očita vrijednost iz objekta A i upisuje vrijednosti u objekt B. Kontrola toka provjerava da informacije sadržane unutar nekih objekata ne teku implicitno ili eksplicitno u manje sigurne objekte. Većina sustava za kontrolu toka koristi neku vrstu sigurnosne klase. Prijenos informacija je dozvoljen samo ako je primateljska sigurnosna klasa na razini pošiljatelja ili iznad. Tako sprječava tok informacija do neovlaštenih korisnika.

Pravila o protoku određuju kanale kroz koje informacije smiju teći. Najjednostavniji pravilnik o protoku određuje samo dvije klase informacija, povjerljivu i nepovjerljivu, i dopušta tok informacija u svim smjerovima osim iz povjerljive klase u nepovjerljivu.

Razlikujemo dva tipa tokova. Eksplicitne tokove koji se stvaraju kao posljedica naredbi dodjeljivanja i implicitne tokove koje stvaraju uvjetne naredbe. Sustav za kontrolu toka mora kontrolirati da se samo ovlašteni tokovi, implicitni i eksplicitni, izvršavaju. Niz pravila mora biti zadovoljen kako bi se osigurao siguran tok informacija. Pravila se mogu izraziti promatrajući odnose između klasa. Ti odnosi mogu definirati niz klasa kojima informacije mogu teći ili mogu odrediti posebne odnose između klasa kako bi omogućili tok informacija iz jedne klase u drugu, ali ne i suprotno.

3.6.1. Skriveni kanali

Skriveni kanali omogućuju prijenos informacija koji krši sigurnosna pravila, to jest, omogućuju prijenos informacija iz više sigurnosne klase u nižu. Skriveni kanali se mogu podijeliti u dvije glavne kategorije, kanale tempiranja i kanale pohrane. Glavna razlika između ta dva tipa je što kanali pohrane ne zahtijevaju vremensku sinkronizaciju kao kanali tempiranja već se informacije prenose pomoću sustava koji su inače nedostupni samom korisniku.

Metode zaključavanja u sustavima za bazu podataka sprječavaju istodobno upisivanje informacija u isti objekt od strane korisnika s različitim sigurnosnim razinama što onemogućava gubitak podataka kroz kanale pohrane. Operacijski sustavi i distribucijske baze podataka nude kontrolu nad multi-programskim operacijama koje dozvoljavaju dijeljenje resursa bez mogućnosti zadiranja jednog programa ili procesa u memoriju drugog sprječavaju kanala tempiranja.

4. ENKRIPCIJA PODATAKA

Prethodne metode kontrole pristupa i toka, iako sadrže snažne sigurnosne mjere, neće zaštititi podatke od nekih prijetnji. Kako bi se onemogućila mogućnost krađe podataka ili pristup informacijama direktno kroz operacijski sustav, podatci se mogu spremiti u bazu podataka u kriptiranom obliku. Samo sustav za upravljanje bazom podataka može dešifrirati podatke pa će svi koji dođu do podataka na drugi način primiti ispremiješane podatke. Kada ovlašteni korisnik pristupi podatcima, sustav za upravljanje bazom podataka dohvaća i dešifrira podatke automatski. Enkripcija se može primijeniti i tijekom slanja samih podataka kako bi u slučaju preusmjeravanja komunikacije poruka ostala skrivena. Tako u slučaju gubitka podataka ili krađe, neovlaštene osobe neće moći razumjeti kriptirane podatke.

Enkripcija zahtjeva sustav za šifriranje koji se sastoji od:

- algoritma za enkripciju koji uzima normalan tekst i od njega proizvodi kriptirani tekst
- ključa za enkripciju koji zajedno s porukom ulazi u algoritam za enkripciju
- algoritma za dekriptiranje koji vrši operacije na kriptiranom tekstu i iz njega daje normalan tekst
- ključa za dekriptiranje koji zajedno s kriptiranom porukom prolazi kroz algoritam za dekriptiranje

Enkripcija se sastoji od primjene algoritma za enkripciju nad podatcima koristeći prethodno određeni ključ za enkripciju. Dobiveni podatci se moraju dekriptirati koristeći ključ za dekriptiranje kako bi se povratili izvorni podatci.

4.1. Standardi za kriptiranje

Standard za enkripciju podataka (engl. *Data Encryption Standard* – DES) je standard razvijen za opću uporabu 1977. U DES standardu, sam algoritam za enkripciju je poznat, dok je ključ sakriven. Slika 6.1. daje pregled DES procesa. DES algoritam koristi 56-bitni ključ na 64-bitni blok teksta, što proizvodi 64-bitni blok kriptiranog teksta. Zbog toga se podatci pri kriptiranju razdvajaju na 64-bitne blokove. Unutar svakog bloka, znakovi se zamjenjuju i premještaju u skladu s vrijednosti ključa. Zbog relativno kratkog ključa, enkripcija se može poništiti unutar razumnog vremena što čini DES slabije sigurnom metodom.

Slika 6.1. pregled rada DES enkripcije

Trostruki DES ili 3DES je osmišljen 1999 kao zamjena za stariji DES standard. 3DES sustav koristi tri ključa i izvodi DES enkripciju tri puta nad podatcima, jednom za svaki ključ.

Napredni standard za enkripciju (engl. *Advanced Encryption Standard* – AES) je nastao 2001. Koristi simetričnu shemu koja je sofisticiranija od DES sheme i podržava tri različite veličine ključa od 128 bita, 192 bita ili 256 bita ovisno o potrebnoj razini sigurnosti. Podatci se razdvajaju na 128-bitne blokove i prolaze kroz četiri procesa transformacije, svaki s nekoliko koraka određenih pomoću ključa. Zbog većih ključeva, probiti shemu je mnogo zahtjevnije.

4.2. Enkripcija simetričnim ključem

Enkripcija simetričnim ključem je oblik enkripcije u kojemu se jedan ključ koristi za enkripciju i dekriptiranje. Korištenje simetričnog ključa omogućava brzu enkripciju i dekriptiranje za rutinsku uporabu osjetljivih podataka u bazi podataka. Poruke kriptirane s tajnim ključem se mogu dekriptirati samo s istim tajnim ključem. Ovakve metode enkripcije se uglavnom koriste za enkripciju samog sadržaja poruke.

Veliki propust vezan za enkripciju simetričnim ključem je potreba za dijeljenjem samog ključa. Ključ se mora posebno čuvati inače je enkripcija beskorisna, bilo tko s simetričnim ključem ima pristup podatcima. Što više osoba ima pristup samom ključu to je veći rizik otkrivanja ključa neovlaštenim osobama.

4.3. Enkripcija javnim ključem

Enkripcija javnim ključem koristi dva različita ključa za enkripciju, gdje je jedan javni a drugi privatan. Algoritmi vezani za enkripciju javnim ključem se baziraju na matematičkim funkcijama umjesto na operacijama nad bitovima. Takav oblik enkripcije rješava sigurnosni problem prijašnjeg oblika enkripcije, siguran prijenos tajnog ključa. Javni ključ nije potrebno slati na siguran način, a privatni ključ se ne šalje. Dva ključa su matematički povezana, pošto se jedan ključ koristi za enkripciju a drugi za dekriptiranje. Međutim, postupak izvođenja privatnog ključa iz javnog je vrlo kompliciran zbog sigurnosnih razloga. Osnovni koraci enkripcije javnim ključem su:

1. Svaki korisnik generira svoj par ključeva koji će se koristiti za enkripciju i dekriptiranje
2. Svaki korisnik daje jedan od ključeva u javnost dok se drugi drži privatnim
3. Kada pošiljatelj želi poslati privatnu poruku primatelju, kriptira svoju poruku javnim ključem primatelja
4. Kada primatelj primi poruku, dekriptira ju svojim privatnim ključem. Nitko drugi ne može dekriptirati poruku jer samo primatelj zna svoj privatni ključ

Primjer za ključeve pri enkripciji bi bila dva velika prosta broja za privatni ključ i umnožak ta dva broja za javni ključ. Pošto nema brzog načina za otkrivanje prostih faktora velikog broja, napadaču je otežan pronalazak određena dva koja čine privatni ključ. Ova metoda je sigurna koliko i sam privatan ključ pa korisnici moraju čuvati sam privatan ključ od razotkrivanja.

5. SIGURNOST BAZA PODATAKA NA INTERNETU

Trajnim rastom interneta raste i opseg podataka koji su dostupni korisnicima. Ukoliko se ne koriste sigurnosni programi, sve se poruke poslane preko interneta mogu otkriti odgovarajućim programima za detekciju podatkovnih paketa što predstavlja veliki sigurnosni rizik. Kako bi se osigurala sigurna komunikacija preko interneta potrebno je osigurati:

- da je komunikacija dostupna samo pošiljatelju i primatelju
- da se podatci neće mijenjati tijekom slanja
- da primatelj može biti siguran da su podatci stigli od pošiljatelja
- da pošiljatelj može biti siguran da je šalje podatke pravom primatelju
- da pošiljatelj ne može poreći slanje podataka

U slučaju slanja izvršivih sadržaja poput ActiveX, VBScript ili Java programa, potrebno je osigurati da pokretanje takvih sadržaja neće dovesti do gubitka podataka ili štete na bazi podataka ili samom sustavu.

5.1. Proxy server

Proxy server je računalo ili program koji se nalazi između korisnika i web servera i djeluje kao posrednik pri komunikaciji, upravljajući porukama u oba smjera. Presreće sve zahtjeve poslane web serveru i obavlja ih sam. Ukoliko ne može sam obaviti zahtjev, filtrira ga i prosljeđuje ga serveru. Odgovor se tada sprema kako bi proxy server ubuduće znao sam odgovoriti na taj zahtjev.

Proxy server se može koristiti za skrivanje IP adrese servera, poboljšanje performansi, zaštitu servera od zlonamjernih programa, zaštitu podataka skeniranjem odlaznih poruka kako bi pronašli curenje podataka ili za filtriranje podataka. Tako organizacija može upotrijebiti proxy server kako bi spriječili svojim zaposlenicima pristup određenim štetnim web stranicama.

5.2. Vatrozid

Vatrozid je sustav koji štiti privatne mreže od neovlaštenog pristupa u ili iz mreže. Implementiraju se sklopovski, programski ili kombinacijom obje opcije. Sve poruke i zahtjevi u mreži prolaze kroz vatrozid koji ih pregledava i blokira one koji ne zadovoljavaju navedene sigurnosne karakteristike.

Vatrozidi baza podataka štite bazu podataka od specifičnih napada koji pokušavaju dati pristup bazi podataka i osjetljivim podatcima unutar nje napadaču. Ujedno omogućuju pregled svih pristupa bazi podataka kroz svoj zapis. SQL ubrizgavanje i napadi uskraćivanja usluge su dva česta tipa napada na bazu podataka koja vatrozid baze podataka može zaustaviti. Ukradene korisničke zaporke i imena mogu dovesti do pokušaja proboja u bazu podataka, ali pošto vatrozid baze podataka stalno nadzire za iregularne aktivnosti, takvi se pokušaji mogu identificirati.

5.3. Digitalni potpisi

Digitalni potpisi koriste enkripciju javnim ključem kako bi osigurali sigurnu komunikaciju koja se ne može poreći. Poput rukom pisanog potpisa, digitalni potpis je način povezivanja oznake specifične određenoj osobi s porukom. Oznaka je nezaboravljiva što znači da i druge osobe mogu provjeriti stiže li poruka od točnog pošiljatelja.

Digitalni potpis se sastoji od niza simbola. Da je nečiji digitalni potpis uvijek isti za svaku poruku lagano bi se mogao falsificirati kopiranjem niza simbola. Zbog toga potpisi moraju biti različiti za svaku uporabu. To se postiže stvaranjem potpisa koji je nastao kao funkcija originalne poruke zajedno s vremenskim oznakom. Kako bi svaki potpis ostao unikatan jednom pošiljatelju i nemoguć za falsificirati, svaki digitalni potpis ovisi o tajnom broju koji je unikatan pošiljatelju. Tako svaki pravilan digitalan potpis ovisi o samoj poruci kao i o tajnom broju. Primatelj ne mora znati tajni broj kako bi potvrdio pošiljatelja.

Jedna metoda korištenja digitalnih potpisa je da pošiljatelj izvrši enkripciju poruke svojim privatnim ključem i onda s javnim ključem primatelja. Primatelj dekriptira poruku svojim privatnim ključem pa onda s javnim ključem pošiljatelja. Dvostruka enkripcija osigurava

autentičnost oba sudionika pošto se poruka ne može dešifrirati bez njihovih javnih i privatnih ključeva. Također osigurava i da je poruka nepromijenjena jer bi promjene na poruci poništile potpis i poruka bi bila nemoguća za dešifrirati.

5.4. Digitalni certifikati

Digitalni certifikati se koriste kako bi povezali javni ključ s identitetom osobe ili organizacije koje drže navedeni javni ključ u digitalnu potpisanu izjavu. Certifikate izdaje i potpisuje autoritet za certifikate (engl. *Certificate Authority* – CA) poput Verisign. Osoba ili organizacija koja prima certifikat od CA je subjekt tog certifikata. Primatelj kriptirane poruke koristi javni ključ CA kako bi dekodirao digitalni certifikat priložen uz poruku.

Digitalni certifikat sadrži razne informacije kao što su identitet CA i podatci o vlasniku certifikata. Svi certifikati sadrže:

- informacije o vlasniku certifikata koje su predstavljene jedinstvenim identifikatorom kao i ime, organizacija kojoj pripada i ostale informacije o samom vlasniku
- javni ključ vlasnika
- datum izdaje certifikata
- period valjanosti
- jedinstveni identifikator izdavača certifikata
- digitalni potpis CA

5.5. Sigurnosne mjere pojedinih SUBP

5.5.1. Oracle

Oracle nudi razne sigurnosne metode koje stupaju na snagu uz standardne SQL autorizacijske komande. Neke od takvih mjera uključuju:

- VPD iliti Virtualna privatna baza podataka je dodatna razina sigurnosti koja dodaje predikate na korisničke izjave kako bi im ograničila pristup na transparentan način. VPD se može postaviti za kontrolu pristupa određenim objektima ili zapisima u bazi podataka, odnosno određenim stupcima ili redovima.
- Redakcija podataka je metoda kojom se sakrivaju podatci pri izvršavanju upita. Neki ili svi znakovi su sakriveni u rezultatu upita. Tako se, na primjer, prikažu samo zadnje četiri znamenke JMBG-a, broja telefona ili broja kreditne kartice u rezultatu.

- Enkripcija podataka na internetu se može izvršiti automatski ili manualno koristeći DBMS_CRYPTO PL/SQL paket. Oracle Net usluge se mogu podesiti kako bi pružale enkripciju i integritet podataka na serveru i na klijentskoj strani.
- Informacije kao što su korisničko ime i lokacija se mogu prikupljati automatski kako bi se kontrolirao korisnički pristup kroz aplikaciju.

Pri instalaciji, Oracle nudi nekoliko prethodno definiranih administratorskih korisničkih računa koji uključuju SYS, SYSTEM i DBSNMP. Ti korisnički računi automatski imaju ulogu administratora baze podataka koja im dozvoljava stvaranje uloga i korisničkih profila, davanje privilegija i ovlasti za stvaranje. brisanje i uređivanje objekata u bazi podataka. Privilegije se dijele na privilegije nad objektima i privilegije nad sustavom. Privilegije nad objektima su obrađene u ranijem poglavlju, a privilegije nad sustavom uključuju naredbe nad samim Oracle resursima kao i stvaranje korisničkih profila.

5.5.2. Microsoft Access

SQL naredbe GRANT i REVOKE nisu dostupne u Microsoft Office Access-u. Sigurnosna mjera baze podataka je zaporka za pristup bazi podataka. Tako samo korisnici s točnom zaporkom mogu otvoriti bazu podataka, ali jednom kada se pristup ostvari, svi objekti u bazi podataka postaju dostupni korisniku. Zbog toga se preporuča redovito mijenjanje zaporke.

Uz to, preporuča se spremanje baze podataka na server koji upravlja svojim zasebnim sigurnosnim mjerama. Tada se Access koristi za stvaranje formi i upita za povezivanje sa serverom. Pri takvoj komunikaciji treba primijeniti sigurnosne mjere pri komunikaciji na internetu.

6. ZAKLJUČAK

Razina sigurnosti baze podataka varira od jedne do druge baze podataka. Za lokalnu bazu podataka koju će koristiti samo jedan korisnik i koja ne sadržava osjetljive podatke nije potrebno koristiti ijednu sigurnosnu metodu navedenu u ovom radu. Dok se za velike baze podataka s mnogo osjetljivih podataka poput baza podataka banke ili bolnice treba uspostaviti maksimalna sigurnost.

Tehnike korištene za osiguravanje baze podataka imaju još puno potencijala za nadogradnju. Međutim, ne postoji standard za dizajniranje takvih sigurnosnih modela. Ovaj rad daje osnovne prikupljene informacije potrebne za upoznavanje s opasnostima i sigurnosnim problemima baze podataka. Po njima se može proširiti, dizajnirati i implementirati prikladan sigurnosni model prilagođen bazi podataka koju štiti.

LITERATURA

[1] M. Connolly, C. Begg, Database systems – A practical approach to design,implementation and management (4th ed.), Addison-Wesley, England, 2005.

[2] R. Elmasri, S.B. Navathe, Fundamentals of database systems (6th ed.), Addison-Wesley, England, 2010.

[3] J. Hoffer, M. Prescott, F. McFadden, Modern database management (8th ed.), New Jersey: Prentice-Hall, 2007.

[4] M. Malik, T. Patel, Database security – attacks and control measures, International Journal of Information Sciences and Techniques, Vol.6, No.1/2, March 2016

[5] D. A. Schultz, Decentralized Information Flow Control for Databases, Massachusetts Institute of Technology, 2012.

[6] Jones and Bartlett Learning, Introduction to Database Security, https://samples.jbpub.com/9781284056945/DBICHAP8.pdf [Kolovoz 2019]

[7] K. Rajesh, What are Database Firewalls, why are they required & how do they protect databases?, excitingip, 2011, https://excitingip.com/1933/what-are-database-firewalls-why-are-they-required-how-do-they-protect-databases [Kolovoz 2019]

SAŽETAK

Ovaj rad pruža uvid u najčešće sigurnosne prijetnje bazama podataka kao i pogled na kontrolne metode za suzbijanje takvih prijetnji. Navedene sigurnosne prijetnje ne pokrivaju sve moguće prijetnje i opasnosti, ali obuhvaća one najčešće i najlakše za izvesti napadaču. Prolaz kroz sigurnosne mjere počinje od same baze podataka i nastavlja se širiti na okolinu koja utječe na bazu podataka kao što su komunikacija i internet. Na kraju se obrađuje nekoliko sigurnosnih mjera koje implementiraju popularniji sustavi za upravljanje bazom podataka.

Ključne riječi: baza podataka, internet, kontrola pristupa, kontrola toka, prijetnje, sigurnost, statističke baze podataka, sustav za upravljanje bazom podataka

ABSTRACT

Database security

This thesis gives an overview of the most common threats to databases, as well as an insight into the control measures used for preventing such threats. Previously mentioned security threats do not cover all the possible threats and dangers, but they do cover the most frequently used ones as well as ones that are the easiest to perform. Overview of the security measures starts from the security measures concerning the database itself and then proceeds to those concerning its surroundings such as the internet and networking. In the end, some of the security measures implemented by popular database management systems are mentioned.

Keywords: access control, database, database management system, flow control, internet, security, statistical databases, threats

ŽIVOTOPIS

Robert Dumančić rođen je 14. Kolovoza 1997. godine u Osijeku. Nakon završenog osnovnoškolskog obrazovanja, upisuje Elektrotehničku i prometnu školu Osijek, smjer tehničar za računarstvo. 2016. godine upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek, preddiplomski smjer računarstvo.

.....