DOSTOPNOST IN ZAŠČITA PODATKOV PRI RELACIJSKI PODATKOVNI BAZI ORACLE

Boštjan Brumen, Tatjana Welzer
Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko
Smetanova 17, 2000 Maribor
E-pošta: Bostjan.Brumen@Uni-Mb.Si, Welzer@Uni-Mb.Si

Povzetek

Prispevek obravnava problem varovanja podatkov v informacijskih sistemih organizacij, predvsem v podatkovnih bazah, kjer se vsi podatki nahajajo. Prikazuje različne pristope k varovanju podatkov in podaja konkretne primere za okolje Oracle.

Abstract

The article describes the problem of data security in organization information systems, precisely in databases where all the data is stored. It describes various methods of data security and gives several examples for Oracle environment.

1. Uvod

Cilj zaščite podatkov je varovanje pomembnih in kritičnih informacij pred nepooblaščenim dostopom, spreminjanjem ali brisanjem. Podatki so namreč tisti del premoženja podjetja, ki jih je najtežje oceniti in ovrednotiti. Skrivnost sama po sebi nima vrednosti, lahko pa napravi veliko škode, če skrivnost preneha biti skrivnost.

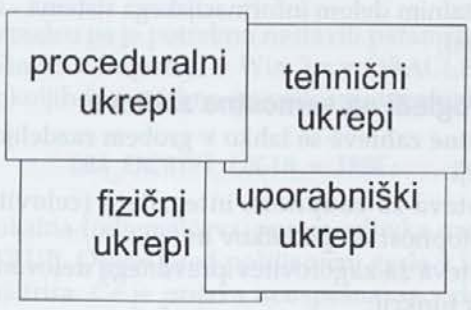
V tem prispevku prikazujemo splošne pristope k varovanju podatkov: varovalne ukrepe in politike varovanja. Sledi prikaz varovanja podatkov v podatkovnih bazah, konkretni primeri pa so predstavljeni v okolju relacijske podatkovne baze Oracle.

2. Varovalni ukrepi

Obstaja več vrst varovalnih ukrepov, ki jih mora organizacija upoštevati in uporabiti za varovanje svojih podatkov. V grobem so ti mehanizmi štirje, in sicer fizični, tehnični, proceduralni in uporabniški (slika 1).

Primer fizičnega ukrepa je shranjevanje disket in diskov v zaklenjenih omarah ali prostorih. Tehnični ukrep je večnivojsko preverjanje avtorizacije dostopa do podatkov. Proceduralni ukrep je npr. preprečevanje prepisovanja podatkov, uporabniški ukrep pa je med drugim preverjanje uporabnikove zgodovine.

Skoraj nemogoče je izdelati varovalno politiko s samo enim ukrepom. Učinkovit sistem za varovanje podatkov mora upoštevati vse štiri vidike in jih kombinirati. Fizični ukrepi so vedno potrebni za zagotovitev zaupnosti in dostopnosti. Ni namreč dovolj preprečiti tujim osebam dostop do podatkov, do njih mora biti hkrati omogočen dostop tistim osebam, ki so za to pooblaščene. Če so podatki dobro prikriti in so odtujeni, verjetno niso kaj prida uporabni odnašalcu, vendar pa so sila pomembni za organizacijo.



Slika 1: Varovalni ukrepi

Zaradi hitrega razvoja tehnologije se povečujejo možnosti varovanja s pomočjo tehničnih ukrepov in so zato drugi ukrepi lahko manj strogi. Kljub temu pa samo tehnični ukrepi niso dovolj za varnost podatkov [1].

3. Politika varovanja

Politika varovanja je množica zakonov, pravil in praktičnih napotkov, kako naj neka organizacija upravlja, ščiti in porazdeljuje občutljive informacije.

Relativna pomembnost vsake izmed zahtev je odvisna od organizacije, v kateri se ta zahteva izpolnjuje. Politike varovanja se v grobem delijo na vojaške in komercialne [3].

V vojaški organizaciji je npr. zelo pomembno varovanje zaupnih informacij in ta zahteva prevladuje pred zahtevo po celovitosti in dostopnosti. V poslovnem svetu je morda zahteva po dostopnosti pomembnejša od varovanja. V odločitvenih sistemih pa morda prevladuje celovitost podatkov.

Vsaka organizacija ima nekoliko različne zahteve in prioritete glede tehničnih varnostnih ukrepov, kakor tudi glede zahtev po fizičnem, uporabniškem in proceduralnem varovanju. Sklop vseh teh zahtev in prioritet sestavlja politiko varovanja organizacije.

Programski proizvod mora pravilno implementirati vse zahteve za uspešno varovanje, hkrati pa mora zadovoljiti vsa pravila poslovanja. Potrebna so določena prilagajanja zahtev, da lahko organizacija postavi takšno politiko varovanja, da se zadosti vsem poslovnim pravilom.

4. Pomembnost varovanja podatkovne baze

Čeprav je varovanje podatkovne baze le en del skupnih ukrepov za varovanje informacij, pa je ta ukrep eden izmed najpomembnejših. Kot centralni repozitorij organizacije je podatkovni strežnik ključna tehnična točka [2]. Operacijski sistem, mrežne storitve in prikrivne naprave prispevajo veliko k varovanju; podatkovni strežnik pa nosi glavno odgovornost za procesiranje in upravljanje z najdragocenejšim in najbolj vitalnim delom informacijskega sistema - z informacijami.

4.1. Pogledi na varnostne zahteve

Varnostne zahteve se lahko v grobem razdelijo v dve področji:

- zahteva za zaupnost, integriteto (celovitost) in dostopnost do podatkov in
- zahteva za zagotovitev pravilnega delovanja gornjih funkcij.

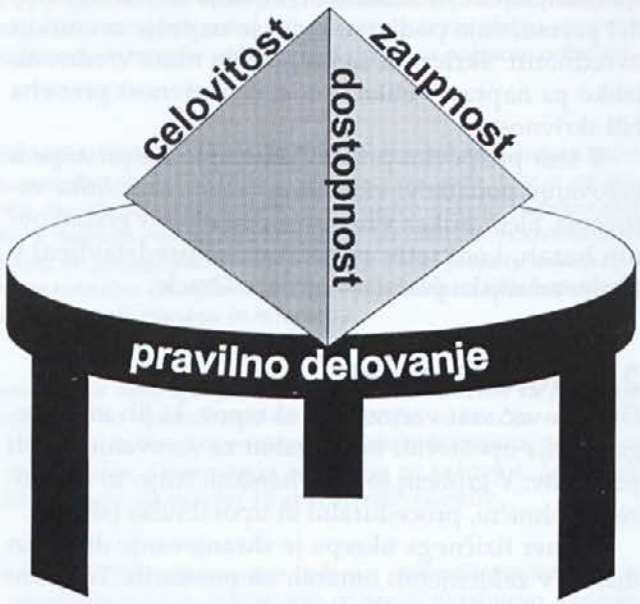
Zaupnost v informacijskem sistemu pomeni, da so podatki dostopni le tistim, ki so pooblaščeni za dostop (tistim, ki "morajo vedeti") [3]. Sistemi zagotavljajo zaupnost s pomočjo mehanizmov identifikacije in preverjanja verodostojnosti, kontrole dostopa, zapisa dnevnika dostopa ter kontrole ponovne uporabe objektov (zagotavljanje, da se zbrisani podatki ne morejo obnoviti na nepredviden način).

Celovitost podatkov pomeni, da so podatki veljavni - da v primeru spremembe niso v nasprotju z realnim stanjem. V podatkovni bazi se celovitost zagotavlja s pomočjo sistemskih mehanizmov, ki zagotavljajo celovitost podatkovne baze kot celote in s pomočjo podatkovnih mehanizmov, ki zagotavljajo skladnost podatkov z relacijskimi zahtevami (npr. referenčna integriteta) [2].

Dostopnost do podatkov pomeni, da so podatki pravočasno in na preprost način na voljo tistim, ki so jim namenjeni, hkrati pa se mora zagotoviti tolerantnost do sistemskih napak, možnost obnovitve v primeru le-teh in odpornost proti poizkusom vdora nepooblaščenim osebam [3].

Oraclov strežnik ustreza vsem pogledom na varnostne zahteve (npr. arhiviranje v času delovanja sistema, obnova in replikacija).

Zagotovitev pravilnega delovanja je osnova za pravilnost in učinkovitost celotnega sistema zaupnosti, celovitost in dostopnosti (slika 2). Zagotovitev pravilnega delovanja se doseže s pravilnimi varovalnimi tehnikami, notranjim in zunanjim testiranjem in zunanjimi ocenitvami.



Slika 2: Pogledi na varnostne zahteve

4.2. Politike varovanja znotraj podatkovne baze

Znotraj podatkovne baze se lahko izdelajo politike varovanja za naslednja področja [1]:

- varnostni skrbnik
- sistemsko varovanje
- podatkovno varovanje
- uporabniško varovanje
- zapisovanje dnevnikov

V zastavljeni politiki varovanja moramo odgovoriti predvsem na naslednja vprašanja:

- kdo so pooblaščeni uporabniki (identifikacija uporabnikov in preverjanje verodostojnosti)
- do katerih objektov smejo imeti dostop (kontrola dostopa)
- kaj lahko na teh objektih izvajajo (prav tako del kontrole dostopa)
- katere aktivnosti so se dogajale v podatkovni bazi (možnost sledenja spremembam s pomočjo dnevnika).

Dodatna vprašanja se nanašajo na podatkovno in sistemsko celovitost, zanesljivost in dostopnost ter na kontrolo pogojnega dostopa (za zagotovitev pravil poslovanja ali za posebej občutljive podatke). Zagotovitev podatkovne in sistemske celovitosti in zanesljivosti je stvar sistema za upravljanje podatkovne baze ter operacijskega sistema. Dostopnost in pogojni dostop morata biti upoštevana pri oblikovanju politike varovanja.

4.2.1. Varnostni skrbnik

Vsaka podatkovna baza ima enega ali več skrbnikov, ki so odgovorni za upravljanje vseh pogledov delovanja podatkovne baze. Če je podatkovna baza velika, je lahko skrbnik podatkovne baze druga oseba kot varnostni skrbnik. Varnostni skrbnik je odgovoren za izvajanje politike varovanja v podatkovni bazi.

4.2.2. Politika sistemskega varovanja

Del politike sistemskega varovanja je med drugim tudi odločitev, koliko uporabnikov bo imelo pravico upravljati z drugimi uporabniki podatkovne baze. Prav tako je del politike sistemskega varovanja odločitev, kje se bo izvajalo preverjanje verodostojnosti uporabnikov - na nivoju operacijskega sistema, na nivoju podatkovne baze ali pa se ne bo izvajalo. Vsaka odločitev ima seveda svoje dobre in slabe lastnosti.

V primeru, da se preverjanje izvaja na nivoju operacijskega sistema, je potrebno nastaviti zagonski parameter podatkovne baze:

```
REMOTE_OS_AUTHENT=TRUE
```

Primer:
Identifikacija na nivoju podatkovne baze:

```
CREATE USER MIHA IDENTIFIED BY GESLO;
```

Identifikacija na nivoju operacijskega sistema:

```
CREATE USER MIHA IDENTIFIED EXTERNALLY;
```

4.2.3. Politika podatkovnega varovanja

Politika podatkovnega varovanja se oblikuje glede na občutljivost podatkov. V podatkovni bazi lahko uporabniki kreirajo objekte po lastni presoji, ali pa to pravico obdrži skrbnik podatkovne baze. Prav tako se je potrebno odločiti, ali lahko uporabniki pravico do dostopa do njegovih podatkov podelijo drugim uporabnikom ali pa sme to napraviti le varnostni skrbnik. Pri politiki podatkovnega varovanja se je potrebno odločiti, ali bodo podatki šifrirani (s prikrivnim postopkom) ali ne.

Vrste prikrivanja:

- sočasno (on-line)
- prikrivanje varnostnih kopij
- prikrivanje stolpcev ali vrstic v tabelah
- prikrivanje podatkov pri pošiljanju prek omrežja.

Oracle 7 v osnovi ne podpira nobene izmed zgoraj naštetih prikrivanj. Edini podatki, ki so zapisani v podatkovni bazi v kodirani obliki, so podatki o geslih.

Pri prenosu podatkov prek nezavarovanih komunikacijskih kanalov lahko nepooblaščena oseba ugotovi geslo. Zato Oracle 7 omogoča prikrivanje gesel pred prenosom. Pri tem uporablja modificirani DES (Data Encryption Standard) [2] algoritem s 40- oz. 56-bitnim ključem [8].

Nezavarovan komunikacijski kanal je med drugim tudi prenos podatkov prek omrežja (z uporabo SQL*NET vmesnika). Prikrivanje gesel se mora vklopiti tako na odjemalčevi (client) strani, kot na strežnikovi (server) strani.

Pri strežniku je potrebno nastaviti zagonski parameter:

```
DBLINK_ENCRYPT_LOGIN = TRUE;
```

pri odjemalcu pa je potrebno nastaviti parameter okolja (v Win'95 v Registry, v Win 3.x v ORACLE.INI, v UNIX okoljih je potrebno izvoziti spremenljivko):

```
ORA_ENCRYPT_LOGIN = TRUE;
```

Če je lokalna (odjemalčeva) spremenljivka nastavljena na TRUE, Oracle pred pošiljanjem gesla v bazo letega zašifrira. Če je prijava neuspešna, se v dnevnik zapišejo podatki o neuspešnem poskusu. Nato preveri,

ali je parameter na strežniku nastavljen na FALSE. Če je, ponovno pošlje geslo, tokrat v originalni obliki. Če je tokrat poskus uspešen, se predhodni neuspeh zbriše iz dnevnika, sicer ostane.

Za prikrivanje podatkov morajo poskrbeti programske rešitve, ki uporabljajo podatke podatkovne baze. To pomeni, da program, preden zapiše podatek v podatkovno bazo, le-tega s pomočjo raznih algoritmov zašifrira. Pri tem nastane problem, ko se npr. zamenja kodirni ključ. V tem primeru je potrebno vse podatke dekodirati s starim ključem, nato pa jih na novo kodirati z novim ključem. Ta postopek je torej uporaben samo pri tistih podatkih, ki so varnostno zelo občutljivi.

Za prikrivanje podatkov pri prenosu prek varnostno nezanesljivih kanalov ponuja Oracle dodatni (komunikacijski) modul, SecureNetServices, ki podatke pred pošiljanjem prek omrežja šifrira. To opravi s pomočjo 40 bitnega ključa[1] [10] in algoritmoma RC4 (Ron's Code 4) [12] ali DES [2]. Modul se vrine med programom in omrežjem na strani odjemalca in med omrežjem in SUPB (sistem za upravljanje podatkovne baze) na strani strežnika ter skrbi za uporabniku in SUPB nevidno šifriranje podatkov pri prenosu prek omrežja.

Vendar obstajajo trije razlogi, zakaj bi želeli uporabniki razviti svoj modul za prenos varnostno občutljivih podatkov prek omrežja.

Prvi je v tem, da je potrebno komunikacijski modul SecureNetServices dokupiti, drugi pa ta, da sta uporabljena algoritma Rc4 in DES zastarela in ju je možno z napadom z uporabo grobe sile zlomiti. Uporabniki lahko v svojem modulu uporabijo trojni DES algoritem, ki uporablja 112 bitni ključ, algoritem IDEA (International Data Encrypion Algorithm) [11] ali kar kriptografsko shemo RSA (Rivest-Shamir-Adelman), ki uporablja par ključev (javnega in privatnega) [2, 12]. Vsi ti algoritmi so v tem trenutku "varni", kar pomeni, da jih z današnjo tehnologijo ni možno zlomiti (z napadom z grobo silo) [9].

Tretji razlog je ta, da lahko uporabnik prikrije le del relacijske sheme (samo določene vrstice ali stolpce) – omogočen mu je torej selektivni pristop k varovanju podatkov.

Razvoj takšnega modula je cenovno zanimiv predvsem v primeru, če ima organizacija le majhen delež občutljivih podatkov, ki se prenašajo po varnostno negotovem kanalu. Načina delovanja lastnega modula sta v principu dva:

1. Podatki se po vnosu v vnosni ekran programa kodirajo, se kodirani prenesejo po prenosnem kanalu in se kodirani tudi zapišejo v podatkovno bazo. Ta pristop je cenejši glede razvoja programskega modula, saj se vse delo opravi na strani odjemalca. Zato pa je potrebna močnejša strojna oprema.
2. Podatki se po vnosu v vnosni ekran programa kodirajo, se kodirani prenesejo po prenosnem kanalu, pred zapisom v podatkovno bazo pa se odkodirajo. Ta pristop je dražji glede razvoja programskega modula, prav tako pa še dodatno obremeni podatkovni strežnik.

Slabost prve metode je v tem, da je potrebno vsa povpraševanja v podatkovno bazo predvideti in jih implementirati, saj so podatki v podatkovni bazi prikriti in ni možno uporabiti klasičnega pristopa k povpraševanju (podatki so kodirani, edina možna primerjava med njimi je primerjava glede enakosti). Slabost je tudi v tem, da so atributi vseh tabel takšne podatkovne baze znakovnega tipa, kar dodatno otežuje naknadno vzdrževanje podatkovne baze. Prednost metode je v preprosti uvedbi in v dejstvu, da so kritični podatki ves čas prikriti.

Slabost druge metode je v tem, da je potrebno spremeniti tako programsko rešitev, s pomočjo katere se podatki vnašajo kot podatkovni strežnik (na podatkovnem strežniku je potrebno dodati prožilce in shranjene procedure, ki prestrezajo kodirane podatke, jih odkodirajo in zapišejo v tabele). Prednost metode je v tem, da je možno opraviti poljubno povpraševanje v podatkovno bazo.

V obeh primerih se poveča promet po kanalu (omrežju) med odjemalcem in strežnikom. Meritve ob uporabi DES algoritma kažejo, da povečanje prometa znaša med 100% v najboljšem primeru (ko je dolžina niza, ki ga prikrivamo, deljiva z 8 brez ostanka), v najslabšem primeru pa 1500% (ko je niz, ki ga prikrivamo, dolg le en znak) [13].

Del politike podatkovnega varovanja je tudi izbira sistema za upravljanje podatkovne baze (in tudi pripadajočega operacijskega sistema). Pri tem so v veliko pomoč razni varnostni kriteriji različnih organizacij. Oracle 7 je preizkusil ameriški "U.S. National Computer Security Center (NCSC)", po kriterijih "Trusted Computer System Evaluation Criteria (TCSEC ali Orange Book)" in ustreza razredu C2. TCSEC kriterij ima štiri nivoje (A-D), razred C ima dva podnivoja (C1, C2), razred B pa tri (B1-B3) [5], [6]. Kriterij preverja razne parametre varnostnega sistema, od načina načrtovanja sistema pa vse do varnostnih funkcij, ki jih sistem nudi. V nivoju D se nahajajo sistemi, ki ne nudijo nobene zaščite ali niso klasificirani v višjem nivoju. Na nivoju D so npr. sistemi z DOS operacijskim sistemom. V nivo C1 spadajo sistemi, ki omogočajo minimalno

1 *V ZDA in Kanadi se uporablja 128-bitni ključ, izven teh dveh držav pa 40-bitni zaradi prepovedi izvoza kriptografske tehnike*

zaščito z gesli, v nivo C2 pa sistemi, ki že omogočajo natančnejši dostop do objektov sistema (samo pooblaščeni uporabniki lahko dodeljujejo pravice). Nivo B zahteva od sistema, da omogoča večnivojski dostop do objektov sistema (vsak objekt lahko ima določeno varnostno nalepko). Nivo A zahteva od sistema, da so poti dostopa do objektov zaprte in natančno določene in da so varnostno kritične metode formalno verificirane.

Evropski preizkus je opravil "European Information Technology Security Center" po kriterijih "European Information Technology Security Evaluation Criteria (ITSEC)". Oracle 7 ustreza razredu zaupanja E3 in razredu funkcionalnosti F-C2.

Evropski kriterij je po funkcionalnosti podoben ameriškemu, le da ima šest nivojev (E1-E6). Nivo E1 je ekvivalenten ameriškemu nivoju D, E6 pa je še malo bolj strog od ameriškega A nivoja [7].

Podatkovna baza "Trusted Oracle 7" ustreza še višjim kriterijem – nahaja se na ameriškem nivoju B [8].

4.2.4. Politika uporabniškega varovanja

Politika uporabniškega varovanja mora upoštevati različne vrste uporabnikov: končne uporabnike, razne skrbnike in razvijalce programov. Pri vseh mora poskrbeti za varnost uporabniških gesel in upravljanje s pravicami uporabnikov.

Varnost uporabniških gesel je ključnega pomena za varovanje podatkov, tako pred izgubo, brisanjem, spreminjanjem kot nepooblaščenim dostopom. Od uporabnikov lahko zahtevamo, da redno spreminjajo svoja gesla, da v geslih zahtevamo posebne znake in števila in da uporabnik spremeni geslo, če ga nekdo odkrije.

Pri upravljanju s pravicami uporabnikov je potrebna odločitev, ali bodo vsakemu uporabniku izrecno določene pravice, ali pa bodo uporabniki porazdeljeni v skupine in se bodo pravice dodeljevale skupini. Pri tem je pomembno število vseh uporabnikov podatkovne baze.

Pri dodeljevanju pravic mora uporabnik dobiti vse pravice, ki jih potrebuje za nemoteno delo, hkrati pa nobene, ki je ne potrebuje. Oracle zagotavlja veliko stopnjo varnosti. Ob prvi namestitvi se namreč izdela skupek pravic, ki jih novi uporabniki potrebujejo za delo. Vsak na novo vključen uporabnik dobi le te pravice in nobene druge.

4.3. Dodeljevanje pravic

Pravice se pri Oraclovi bazi delijo na sistemske in na objektne. Objekti v podatkovni bazi so: tabela, pogled (view), sinonim, shranjena procedura, sekvenca, posnetek (snapshot) [1], [4].

Za vsak objekt so lahko dodeljene naslednje pravice):

Objekt:	Pravice:
tabela	SELECT, INSERT, UPDATE, DELETE, REFERENCES, ALTER, INDEX, ALL PRIVILEGES
pogled	SELECT, INSERT, UPDATE, DELETE
sekvenca	SELECT, ALTER
procedura	EXECUTE
posnetek	SELECT

Razen pravic za delo z objekti ima lahko uporabnik še določene sistemske pravice.

Sistemske pravice so naslednje:
ALTER (anything), ANALYZE ANY, AUDIT ANY, AUDIT SYSTEM, BECOME USER, COMMENT, CREATE (anything), DELETE (ANY) TABLE, DROP (anything), EXECUTE, FORCE TRANSACTION, GRANT (anything), INSERT ANY TABLE, LOCK, MANAGE TABLESPACE, READUP, RESTRICTED SESSION, SELECT ANY SEQUENCE, SELECT ANY TABLE, UNLIMITED TABLESPACE, UPDATE ANY TABLE, WRITEDOWN in WRITEUP

Vsak uporabnik lahko pogleda svoje sistemske pravice z ukazom:

```
SELECT * FROM session_privs;
```

Objektne pravice lahko uporabnik preveri z ukazom:

```
SELECT table_name, privilege, grantable FROM
sys.dba_tab_privs WHERE grantee='USER';
```

Če se uporabniku doda določena sistemska pravica z dodatkom WITH ADMIN OPTION, lahko ta uporabnik le-to po lastni presoji doda drugemu uporabniku. Podobno je z objektnimi pravicami, kjer se mora prenos pravice opraviti z dodatkom WITH GRANT OPTION.

Primer:

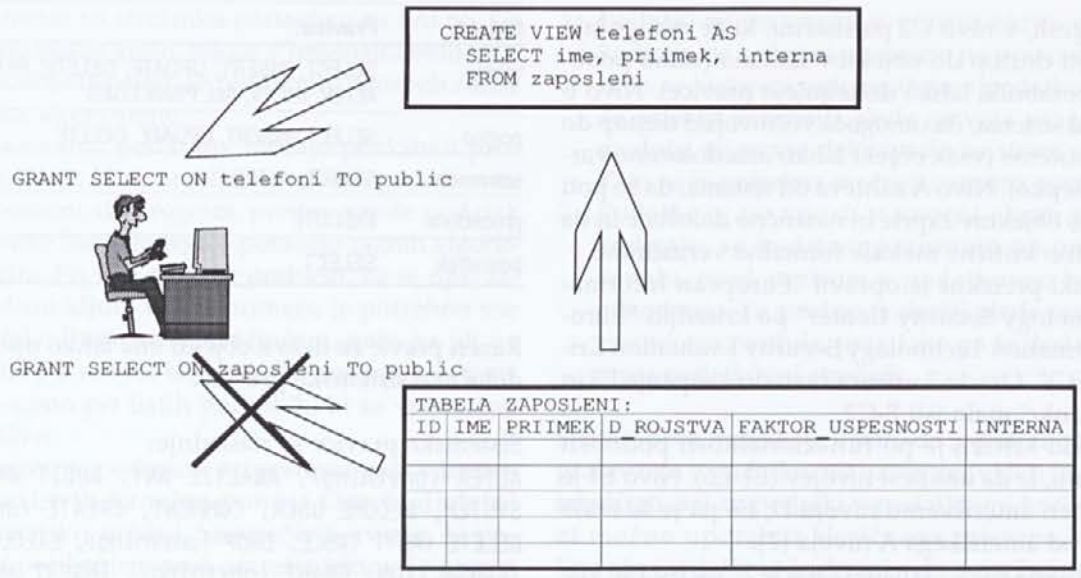
```
GRANT SELECT ON STRANKE TO MIHA;
```

Uporabniku Mihi je dodeljena pravica, s katerim lahko izpisuje podatke iz tabele "STRANKE", ne more pa omogočiti drugemu uporabniku izvajanja te operacije.

```
GRANT SELECT ON STRANKE TO MIHA WITH GRANT OPTION;
```

Sedaj lahko uporabnik Miha dodeli ta isto pravico (možnost izpisovanja podatkov iz tabele "STRANKE") drugim uporabnikom po lastni presoji.

Uporabnik ne more dobiti le del objektne pravice - na primer pravico SELECT le na dveh stolpcih ali vrsticah



Slika3: Omejevanje dostopa do podatkov s pomočjo pogledov

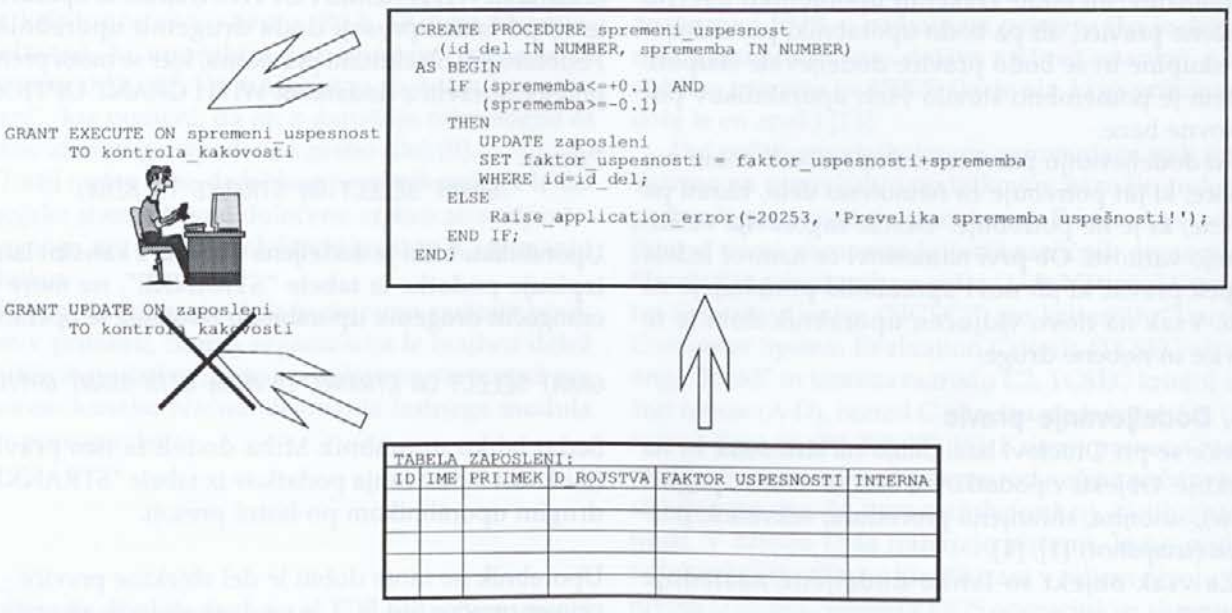
celotne tabele. Ta problem je rešljiv z uporabo pogledov (angl. view). Skrbnik izdela pogled na tabelo in vanj vključi le tiste stolpce (ali vrstice), do katerih uporabnik potrebuje dostop. Uporabniku nato dodeli potrebne pravice samo na ta pogled. Uporabniku ni potrebno dodeliti kakršnih koli pravic do osnovne tabele (primer na sliki 3).

Pravice se lahko dodajajo neposredno uporabnikom ali pa skupinam (pri Oraclu se skupine imenujejo "roles"). Vsak uporabnik pripada skupini "PUBLIC" in poljubnemu številu ostalih skupin. Skupina je poimenovan skupek pravic. Vsak uporabnik dobi pravice skupine, katere član je. Skupina je lahko zaščitena z geslom.

4.3.1. Natančnejša (pogojna) kontrola dostopa do podatkov

Dostop do podatkov lahko še bolj precizno nadziramo s pomočjo shranjenih procedur (angl. stored procedures) [1].

Shranjene procedure in funkcije so množice PL/SQL (Oraclov proceduralni jezik) ukazov, ki so shranjeni v podatkovni bazi v prevedeni obliki. Procedure in



Slika 4: Shranjena procedura za opravljanje poslovne funkcije

funkcije lahko povežemo v pakete. Proceduro lahko definiramo tako, da izvede določeno poslovno opravilo (npr. sprememba uspešnosti delavca). Uporabniku nato dodelimo pravico izvajanja te procedure.

Prednost je v tem, da uporabniku ni potrebno dodeliti nobenih dodatnih pravic za dostop do podatkov, ki jih procedura potrebuje, niti do morebitnih ostalih procedur, ki jih osnovna procedura kliče. S tem omejimo uporabnika, da izvaja le tiste operacije, ki so določene le znotraj konteksta opravljanja poslovnih funkcij.

Kot je prikazano na sliki 4, uporabniku podatkovne baze 'kontrola kakovosti' ni potrebno dodeliti pravic za spreminjanje celotne tabele "ZAPOSLENI". Lahko bi sicer tega uporabnika omejili na pogled, kjer bi imel dostopne podatke o faktorju kakovosti posameznega delavca, vendar s tem ne bi rešili poslovnega pravila, ki zahteva, da sprememba faktorja ni večja od 0.1.

Prožilci (angl. triggers) so posebni tipi shranjenih procedur, ki se samodejno izvedejo ob določenih dogodkih (npr. brisanje iz tabele, pisanje v tabelo). S pravicami lahko določimo le, ali naj ima neki uporabnik dostop do podatkov ali ne. S prožilci lahko omejimo dostop do podatkov še glede na druge pogoje, ne samo glede na pravice. Ti pogoji so najpogosteje poslovna pravila, ki jih organizacija uporablja za delovanje.
Primer:

Prožilec 'preveri_izplačilo' izvede preverjanje, ali se izplačilo izvaja v časovnem okviru, ki je določen s pravili poslovanja. Podatkovna baza zavrne dostop, če se poskuša npr. odobriti izplačilo izven obratovalnega časa.

```
CREATE TRIGGER preveri_izplačilo
       before delete or insert or update
       ON računovodstvo.izplačila
       BEGIN
/* Če je sobota ali nedelja, zavrni dostop*/

IF(TO_CHAR(SYSDATE, 'DY') = 'SAT; OR
TO_CHAR(SYSDATE, 'DY') = 'SUN')
THEN
raise_application_error(-20501,
'Izplačil ni možno izvajati v soboto ali nedeljo!');
ENDIF;
/*Če je trenuten čas pred 8.00 zjutraj ali po
13 popoldan, zavrni dostop*/

IF (TO_CHAR(SYSDATE,'HH24') < 8 OR
TO_CHAR(SYSDATE, 'HH24') >=18)
THEN
raise_application-error( -20502,
'Izplačila je možno izvesti samo med 8.00 in 13.00!');
ENDIF;
END;
```

4.4. Zapisovanje podatkov o dostopu

Zapisovanje podatkov o dostopu do podatkov v dnevnik (angl. Database Auditing) je pri sistemu za upravljanje s podatkovno bazo Oracle 7 avtomatizirano [1].

Čeprav je zapisovanje dnevnika relativno nezahtevno, tako prostorsko kot časovno, je potrebno omejiti število dogodkov, ki se v dnevnik zapisujejo. Prav tako je potrebno natančno določiti, katere aktivnosti v podatkovni bazi se bodo preverjale ali opazovale.

V primeru, da se ugotovijo kršitve politike varovanja, je potrebno upoštevati naslednja priporočila:

- Ni dovolj ugotovitev, da nekdo npr. nepooblaščeno dostopa do podatkov. Potrebno je natančneje določiti, kateri podatki (v katerih tabelah) so kritični in po možnosti kateri uporabniki so sumljivi. Če to ni možno, je potrebno sprva spremljati vse uporabnike in vse kritične podatke. Postopoma je potrebno na podlagi analize dnevnika zoževati opazovanje na tiste podatke in uporabnike, ki so sumljivi, vse dokler niso na voljo vsi dokazi o nepooblaščenem dostopu.
- Potrebna je zaščita dnevnika. Sled mora biti zaščitena tako, da je ni možno spreminjati ali brisati. Če to nekdo poskusi, mora ta poskus biti zabeležen.

Primer zaščite sledi:

```
AUDIT INSERT, UPDATE, DELETE ON sys.aud$ BY ACCESS;
```

Pravico DELETE ANY TABLE naj ima le oseba, odgovorna za varovanje podatkovne baze. Vse aktivnosti, ki jih opravljajo uporabniki z skrbniškimi (sistemskimi) pravicami, se naj zabeležijo v dnevnik operacijskega sistema.

Zapisovanje dogodkov v dnevnik je potrebno vklopiti, saj po inštalaciji podatkovne baze zapisovanje ni vklopljeno. Vklop se opravi z nastavitvijo inicializacijskega parametra podatkovne baze:

```
AUDIT_TRAIL=DB
```

Možna je še nastavitev AUDIT_TRAIL=OS, pri čemer se podatki o dostopu zapisujejo v dnevnik operacijskega sistema.

V dnevnik se zmeraj zapišejo naslednji dogodki:

- zagon podatkovne baze (angl. DB instance startup)
- ustavitev podatkovne baze (angl. DB instance shutdown)
- povezava s podatkovno bazo z skrbniškimi pravicami

V dnevnik se lahko poljubno zapisujejo naslednji podatki:

- ime uporabnika
- identifikator seje (session ID)

- identifikator terminala
- ime objekta
- izvedena ali poskušena operacija
- datum in čas
- sistemske pravice, uporabljene za izvršitev akcije

Ob spremljanju dogodkov se v dnevnik zapišejo vsaj naslednji podatki:
- uporabnik, ki je izvedel ukaz
- koda akcije, ki določa, kateri ukaz je bil izveden (ali poskušen)
- objekt ali objekti, ki so bili (bi naj bili) udeleženi v akciji
- datum in čas izvedbe akcije.

Zapis v dnevniku ne vsebuje informacij o vrednostih, ki so bile udeležene ob izvajanju akcij. Zapis vrednosti lahko dosežemo z uporabo prožilcev.

Spremljanje akcij lahko poteka na treh nivojih:
- nivo ukaza: zapis temelji na uporabi tipa SQL ukaza, npr. uporaba SELECT, DROP
- nivo pravice: zapis temelji na uporabi določene sistemske pravice, npr. CREATE TABLE
- nivo objekta: zapis temelji na uporabi ukazov na določenem objektu, npr. ALTER TABLE na tabeli "ZAPOSLENI".

Zapis lahko omejimo z dvema pogojema:
- v primeru uspeha / neuspeha (WHENEVER SUCCESSFUL / NOT WHENEVER SUCCESSFUL)
- glede na sejo / glede na dostop (BY SESSION / BY ACCESS)

Zapis podatkov v dnevnik se začne ob prijavi v podatkovno bazo. Če skrbnik v času, ko je neki uporabnik prijavljen, spremeni opcije zapisovanja, le-te za trenutno prijavljene uporabnike niso relevantne. Spremembe bodo postale aktivne šele ob ponovni prijavi uporabnika v bazo.

Primeri zapisovanja podatkov v dnevnik:

1. *Zapisovanje podatkov glede na sistemske pravice*

```
AUDIT SESSION BY miha, janez;
```

V dnevnik se zapišejo podatki o (uspešni in neuspešni) prijavi in odjavi uporabnikov Mihe in Janeza v bazo.

```
AUDIT EXECUTE ANY PROCEDURE BY ACCESS WHENEVER NOT
SUCCESSFUL;
```

V dnevnik se zapišejo podatki o neuspešni rabi pravice izvajanja katerekoli procedure, kateregakoli uporabnika, glede na dostop.

2. *Zapisovanje podatkov glede na ukaz*

```
AUDIT SELECT TABLE, INSERT TABLE, DELETE TABLE BY
ACCESS WHENEVER SUCCESSFUL;
```

V dnevnik se zapišejo podatki o uspešni izvedbi ukazov SELECT, INSERT, DELETE na katerikoli tabeli kateregakoli uporabnika.

3. *Zapisovanje podatkov glede na objekt*

```
AUDIT DELETE ON ZAPOSLENI;
```

V dnevnik se zapišejo vsi podatki o uspešnem in neuspešnem brisanju iz tabele "ZAPOSLENI".

Preklic zapisovanja v dnevnik izvedemo z ukazom NOAUDIT, vendar se pri tem ne uporablja par BY SESSION / BY ACCESS.

Primer preklica:

```
NOAUDIT SESSION BY Miha, Janez;
```

Preklic zapisovanja podatkov v dnevnik o prijavah in odjavah uporabnikov Mihe in Janeza.

Kot je bilo že prej omenjeno, lahko vrednosti, ki so udeležene ob izvajanju določene akcije, zapisujemo v dnevnik s pomočjo prožilcev. Prožilci se uporabljajo takrat, ko potrebujemo bolj natančne podatke.

Primer prožilca:

```
CREATE TRIGGER spremljaj_zaposlene
AFTER INSERT OR DELETE OR UPDATE ON zaposleni
FOR EACH ROW
BEGIN
    INSERT INTO dnevnik_zaposleni VALUES
    (:old.ID, :new.ID, :old.faktor_uspesnosti,
    new.faktor_uspesnosti, user, sysdate);
END;
```

Ta prožilec zapiše v prej definirano in kreirano tabelo "SPREMLJAJ_ZAPOSLENE" stare in nove podatke o številki delavca, njegovi plači in delovnem mestu, uporabniku, ki je ukaz izvedel, o času ter datumu izvedbe ukaza.

5. Zaključek

Podatki so zelo pomemben del lastnine neke organizacije, zato jih je potrebno varovati tako, kot drugo lastnino. Število in učinkovitost varnostnih mehanizmov naj bo sorazmerna vrednosti podatkov. Odločitev, katere

mehanizme izbrati, je del celotne politike varovanja podjetja.

Podatki se shranjujejo v podatkovni bazi, ki je srce informacijskega sistema organizacije, zato je še posebej pomembno, kateri sistem za upravljanje s podatkovno bazo se uporabi. Izbiro nam olajšajo razni kriteriji, ki ocenjujejo komercialne proizvode glede stopnje varnosti, ki jo ponujajo.

V članku smo predstavili nekatere mehanizme in lastnosti sistema Oracle 7, ki omogoča zelo veliko stopnjo varnosti podatkov. Selektivno dodeljevanje pravic, prikrivanje podatkov o geslih med prenosom preko omrežja, preverjanje verodostojosti uporabnikov, zapisovanje podatkov o aktivnostih v dnevnik in mnogi drugi mehanizmi pa sami niso dovolj. Pri varnosti in varovanju podatkov ima zelo velik vpliv človeški faktor. Pri zmanjševanju stopnje tveganja igrajo zelo veliko vlogo vodilni delavci v organizaciji, ki se morajo zavedati pomena in vrednosti podatkov in od svojih podrejenih zahtevati dosledno spoštovanje politik varovanja, pri tem pa morajo biti sami drugim za vzgled.

Literatura in viri:

[1] J. Fee, V. Kane et al.:
Oracle7 Server Administrator's guide, Release 7.2, Oracle Press 1995

[2] C. Pfleeger:
Security in Computing, Prentice Hall, New Jersey, 1997

[3] S Castano et al.:
Database security, Addison-Wesley, New York, 1995

[4] M. Rennhackkamp:
Server Side Database Security, DBMS, Vol. 10, No. 2, pp. 67, Miller-Freeman Inc., San Mateo-California, 1997

[5] De Montfort University Bedford UK,
Orange book summary,
http://www.dmu.ac.uk/~chl/orange.html

[6] US Department of Defense,
Trusted Computer System Evaluation Criteria,
http://www.radium.ncsc.mil

[7] ITSEC,
http://www.itsec.gov.uk/itsechtml/welcome.htm

[8] ORACLE Corporation,
http://www.oracle.com/products/oracle7/server/whitepapers/security/html/security.html

[9] W. Stallings:
Practical Cryptography for Data Internetworks, IEEE Computer Society Press, New Jersey, 1996

[10] ORACLE Corporation,
Secure Networking Option, http://www.oracle.com/st/o8collateral/html/xanostwp.html

[11] CERN, IDEA algorithm,
http://www.r3.ch/products/idea/index.html

[12] RSA Laboratories, RC4,
http://www.rsa.com/rsalabs/newfaq/q87.html

[13] M. Veber:
Diplomsko delo: Varovanje podatkov: prikrivanje podatkov in analiza sklepanja, FERI Maribor, 1997

◆

Dr. Tatjana Welzer je docentka na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru. Poučuje predmete "Podatkovne baze I", "Podatkovne baze II", "Dostopnost in zaščita podatkov" in "Baze podatkov in ekspertni sistemi" tako na univerzitetnem kot tudi na visokošolskem strokovnem programu. Diplomirala je iz elektrotehnike l. 1984, se zaposlila v industriji (Metalna) in se kasneje spet vrnila k raziskovalnemu delu. Magistrirala je l. 1989 in doktorirala l. 1995. Seznam pomembnejših člankov se nahaja na http://lisa.uni-mb.si/osebje/welzer.

◆

Boštjan Brumen je asistent na Fakulteti za elektrotehniko, računalništvo in informatiko Univerze v Mariboru. Diplomiral je iz računalništva l. 1996. Trenutno je vpisan na podiplomski študij računalništva. Seznam pomembnejših člankov se nahaja na http://lisa.uni-mb.si/osebje/brumen.

◆