Sigurnosni aspekti baza podataka

Bursać, Đorđe

UVOD

U današnje vrijeme kada je internet potreban i važan resurs u svim organizacijama potrebno je naročito obratiti pozornost računalnoj sigurnosti i računalnim sigurnosnim mehanizmima, te zaštiti podataka.

Pritom je važno imati u vidu; brisanje povijesti pregledavanja, antivirusnu zaštitu, optimalne lozinke i kriptiranje te fizičku zaštitu, jer štite računalni sustav od mnogih zlonamjernih korisnika. Zbog stalne prisutnosti na internetu, najviše su izloženi korisnici *DSL*[1] veze, kabelskog interneta i stalnih veza, ali ni ostali korisnici interneta nisu izvan opasnosti. Zaštita sigurnosnom stijenkom postoji u različitim oblicima pa se zbog toga preporuča odabrati rješenje u skladu s potrebama.

Cilj ovog rada je ukazati na moguće opasnosti koje postoje, prikazati mogućnosti zaštite osobnih podataka i drugih podataka u elektroničkom obliku.

U početnom dijelu naglasak je dat na upoznavanju same baze podataka i definiranju njenih osnovnih pojmova. Zatim je rečeno o osnovnim sigurnostima računalnih sustava, osnovama sigurnosti baza podataka, te sigurnosnim procedurama baza podataka (u nastavku teksta BP).

Naposlijetku je osvrt stavljen na upravljanje sigurnošću unutar sustava upravljanja baza podataka (u nastavku teksta SUBP), načinu osiguranja dostupnosti, te zaštiti BP.

[1] DSL - *Digital Subscriber Line* - digitalna pretplatnička linija kod koje je brzina prijenosa podataka u smjeru prema korisniku veća od brzine u suprotnom smjeru.

1. BAZA PODATAKA

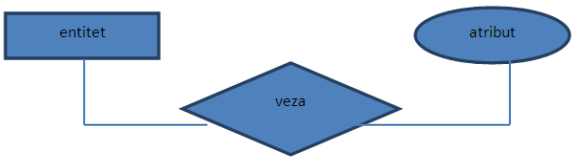
Baza podataka (eng. Database) je termin koji se koristi za skupove podataka i metoda koje služe da se ti podaci organiziraju, pohrane, obrađuju, koriste, te ažuriraju. Termin „baza podataka“ koristi se i za razvojni softver koji omogućava kreiranje i korištenje baze podataka. Baze podataka su informacije koje opisuju stanje nekog sustava u realnom svijetu.(CIS.hr str.2/20).

Baze podataka dijele se na:

**relacijsko–tablične*-strukture koje su definirane tako da se informacijama može pomoću logičkih izraza jednostavno pristupiti i izdvojiti tražene informacije,
**objektno orijentirane*-koriste se pri programiranju, odnosno sadržavanju informacije koje pripadaju objektima, razredima i podrazredima te
**distribuirane*-baze podataka koje se mogu rasporediti na više točaka u računalnoj mreži ili na Internetu,od kojih su najviše u primjeni relacijske baze podataka.
Alati za uređivanje, stvaranje i upravljanje bazama podataka objedinjeni su pod zajedničkim nazivom SUBP-sustavi za upravljanje bazama podataka (eng. SUBP-Database Management System). Neki od najpoznatijih SUBP-a su: Oracle, DB2, MySQL, Informix, PostgreSQL i SQL server.
Relacijske baze podataka su baze kod kojih su podaci smješteni u jednu ili više tablica koje su međusobno povezane. Takve baze mogu biti na jednom računalu za samo jednog korisnika ili na jednom od računala u računalnoj mreži kojoj može pristupiti više korisnika. Postoje baze gdje se podaci nalaze raspoređeni na više računala u mreži. U tom slučaju svi korisnici mogu imati pristup podacima bez obzira gdje se nalaze, naravno u skladu sa svojim ovlastima. Svaki od korisnika ne mora imati ista ovlaštenja. Neki korisnici neke podatke ne smiju niti vidjeti, dok ih drugi smiju viđati, ali ih ne smiju mijenjati, a najviša razina ovlaštenja, osim pristupa, dozvoljava i promjenu podataka. (Čičin-Šain, 2007 : 5)

Baze koje se koriste u praksi obično su vrlo složene, te se zato koriste prototipi modela, koji su pojednostavljeni modeli, na kojima se mogu objasniti glavne značajke građe i funkcije modela. Da bi se kod prototipa objasnilo funkcioniranje baze, ne uzimaju se svi podaci koji se pojavljuju u bazi, nego samo oni posebno važni i uglavnom razumljivi. Prototip se kasnije može doraditi do pravog modela ili se pravi model gradi ispočetka na osnovi prototipa.

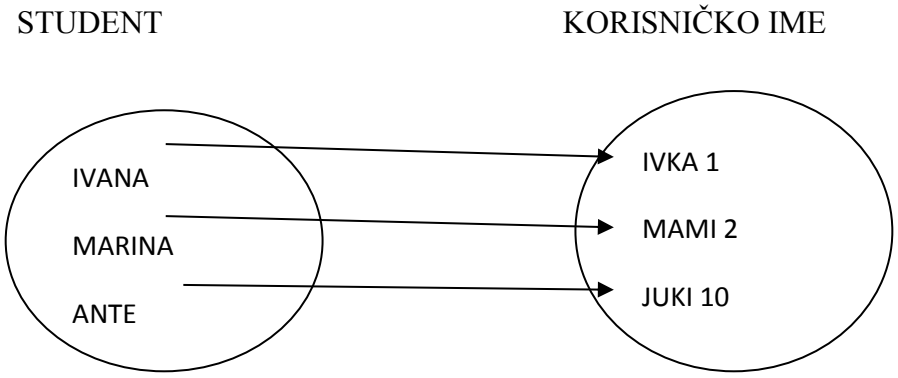
Kod pripreme za izradu baze koriste se grafički prikazi baze ili dijagrami, npr.dijagram objekt-veza. (Čičin-Šain, 2007 : 6)



1. slika – dijagram objekt – veza, Čičin-Šain, 2007, str.6, - Baze podataka.

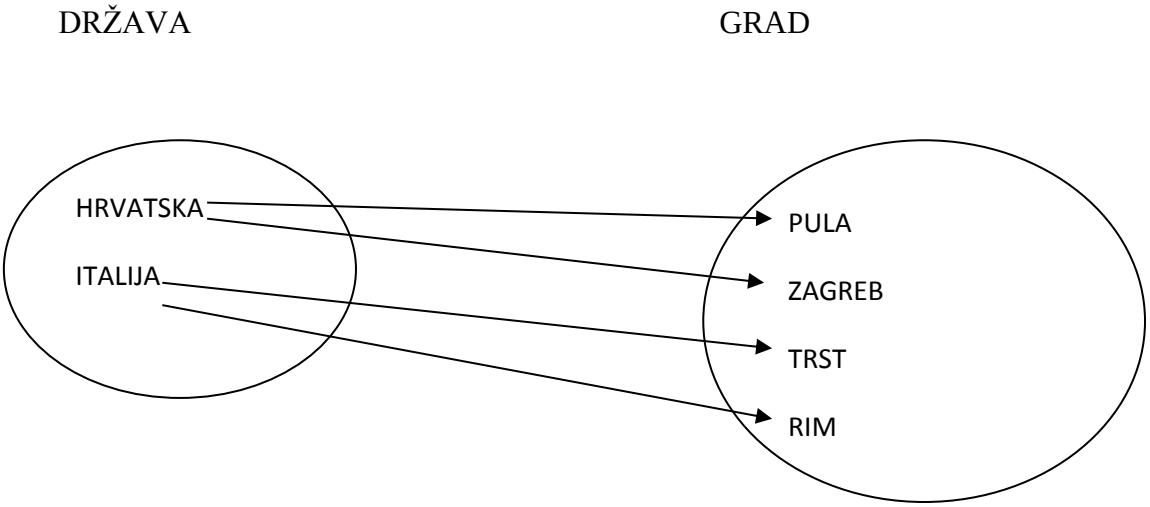
Za dijagrame objekt-veza koriste se osnovni gradbeni elementi. Ovaj se dijagram koristi kada imamo podatke koje želimo čim svrsihodnije spremiti u onoliko datoteka koliko je najbolje za tu bazu. Koriste se simboli: pravokutnik za entitet, elipsa za atribut, te romb za vezu. Podaci se skupljaju o nekom objektu ili entitetu. Entitet može biti bilo što o čemu se prikupljaju podaci. Podatak koji opisuje entitet se zove atribut, a korist se da bi se razlikovalo općeniti podatak od pojedinačnog podatka. Pr. jedan od atributa je Naziv kupca, a Brionka je naziv jednog od kupca. Među entitetima se uspostavlja veza, a ovisno o kardinalnosti može biti : 1:1,1:n, n:1, n:n. U slučaju 1:1 veze, ona se uspostavlja putem primarnog ključa oba entiteta, a možda je poželjno da se ta dva entiteta spoje u jedan. U slučaju 1:n i n:1 veze, ista se uspostavlja putem primarnog ključa na strani 1 i stranog na strani n. U slučaju n:n veze, veza se uspostavlja pomoću vezne tablice. U dijagram se upisuje kardinalnost koja se ponekad i ucrta u dijagram posebnim simbolima.(Čičin-Šain, 2007 : 7).Baza podataka Trgovina-objekt-veza.

Veza između entiteta STUDENT i KORISNIČKO IME, je ta da svaki student može imati samo jedno korisničko ime, dok korisničko ime može pripadati samo jednom studentu.



1. Slika – jednostavna veza 1:1 , http://e-student.fpz.hr/Predmeti/B/Baze_podataka/Materijali/Auditorne_vjezbe_2.pdf
2.

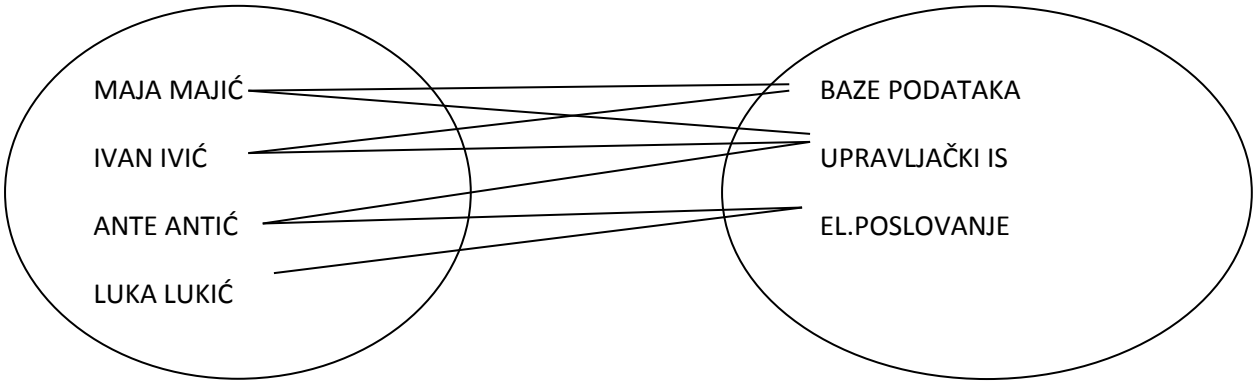
Veza između entitea DRŽAVA i GRAD, svaka država može imati više gradova, dok svaki grad pripada samo jednoj državi.



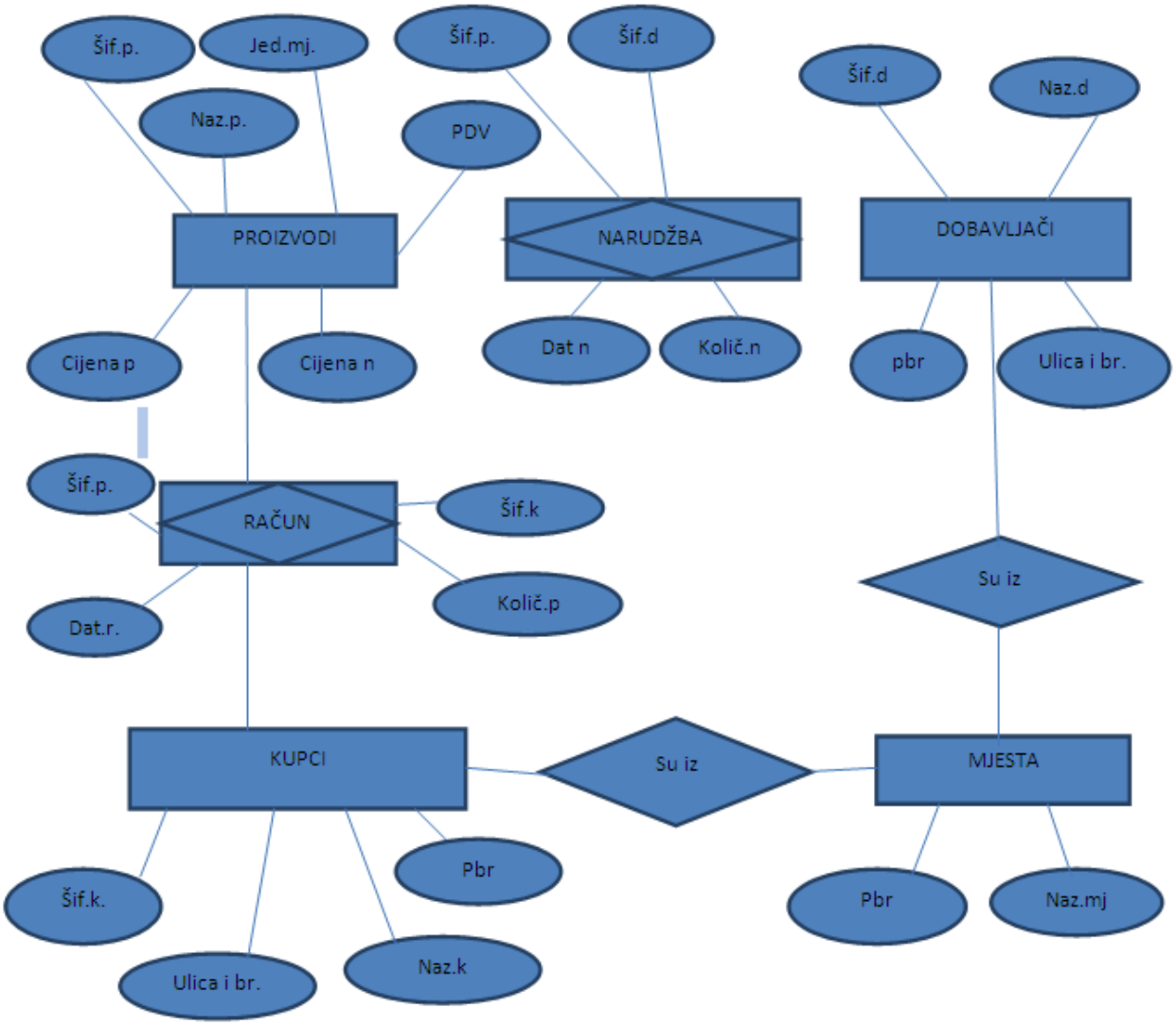
3. Slika – jednostavna veza 1: N, http://e-student.fpz.hr/Predmeti/B/Baze_podataka/Materijali/Auditorne_vjezbe_2.pdf

Veza između entiteta STUDENT i PREDMET. Svaki student može biti upisan na više predmeta, dok svaki predmet može slušati više studenata.

STUDENT PREDMET



4. Slika – jednostavna veza M:N, http://e-student.fpz.hr/Predmeti/B/Baze_podataka/Materijali/Auditorne_vjezbe_2.pdf



5. *Slika ER model-dijagram objekt-veza – Čičin-Šain, 2007 str.7., Cybernetica*

Osnovni entiteti ove baze su: proizvodi, dobavljači, kupci i mjesta. Po kardinalnosti veze, entitet Kupac i Mjesta znači da svaki kupac može imati sjedište u jednom mjestu, a sjedište može imati n kupaca, pa je stoga ta veza n:1.

Svaki kupac može imati n raznih proizvoda, a svaki proizvod može kupiti n raznih kupaca, ta veza entiteta Kupaca i Proizvoda po kardinalnostijeste veza n:n.

Iz dijagrama objekt-veza može se vidjeti koliko će baza imati tablica, a imati će ih onoliko koliko je u dijagramu entiteta.
Baza Trgovina ima šest entiteta: proizvodi, kupci, dobavljači, narudžbe, računi i mjesta, a to znači da će u bazi biti šest tablica s tim imenima. Iz dijagrama se može vidjeti da će od tih šest

tablica imati onoliko stupaca koliko odgovarajući entitet ima atributa, npr. entitet proizvodi ima šest atributa, pa će tablica imati šest stupaca. Ne može se vidjeti koliko će tih šest tablica imati redaka, jer svaka tablica će imati onoliko redaka koliko bude u praksi potrebno, npr. tablica kupci će imati redaka koliko ima registiranih kupaca, a tablica proizvodi onoliko redaka koliko u bazi ima raznih proizvoda. Svaki se redak iz tablice zove n-torka ili slog, a sadrži n podataka, odnosno onoliko koliko ima atributa. Iz dijagrama je vidljivo koliko tablice imaju stupaca, a redaka će biti onoliko koliko trenutno ima proizvoda, kupaca, dobavljača, narudžbi, računa i poštanskih brojeva, što se na temelju dijagrama ne vidi.

Relacijske baze podataka posjeduje osnovna svojstva koja su:

- Podaci se korisniku predstavljaju kao zasebni entiteti,
- Svaki entitet je opisan svojstvima,
- Entiteti se prikazuju tablicama,
- Entiteti su međusobno povezani – postoje relacije među entitetima,
- Podaci su jednostavno dohvatljivi – SQL.

(http://www.unizd.hr/portals/1/primjena_rac/brodostrojarstvo/predavanje_4.pdf).

Osnovna ideja ovog modela leži u činjenici da korisnik ne može unaprijed znati sve moguće načine korištenja podataka u bazi, ne postoje predefinirani putovi kretanja kroz podatke.
Upitni jezici, kao npr, SQL, operiraju sa skupom zapisa, a ne samo s jednim u danom trenutku, a to je karakteristika hijerarhijskog i mrežnog modela.
Podaci se na strukturnom nivou prikazuju kao dvodimenzinalne tablice uz osiguranje fizičke i logičke nezavisnosti.

2. OSNOVNE SIGURNOSTI RAČUNALNIH SUSTAVA

Sigurnost računalnih sustava je ime za skup alata, procedura, pravila i rješenja, čija je namjena da umreženi sistem obrani od napada. Da bi se efikasno procijenile sigurnosne potrebe neke organizacije,te da bi se odabrali različiti sigurnosni proizvodi, pravila, procedure i rješenja,rukovodiocu u firmi koji je zadužen za sigurnost,potreban je sistematski način definiranja zahtjeva u pogledu sigurnosti i kategorizacije pristupa koji osiguravaju da se ti zahtjevi zadovolje.
Trebaju se razmotriti tri aspekta sigurnosti informacija:

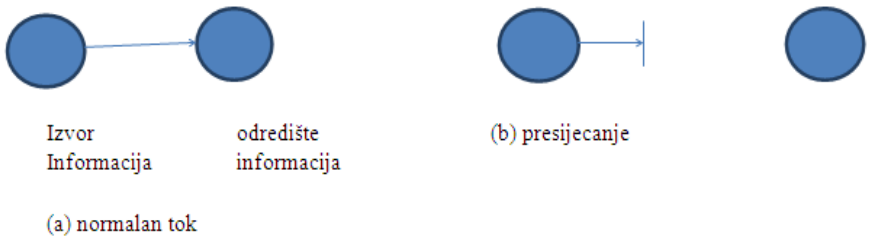
**Povjerljivost* – zaštita svih podataka koji nisu za javnost, te spriječavanje curenja informacija.
**Integritet* – zaštita koja osigurava cijelovitost podataka, tj. Zaštita od neovlaštenog, nepredviđenog ili nenamjernog modificiranja podataka.

**Raspoloživost* – zaštita koja osigurava dostupnost podataka i raspoloživost sistema koji pruža nene usluge.

Sigurnosna usluga podrazumijeva uporabu jednog ili više sigurnosnih mehanizama.

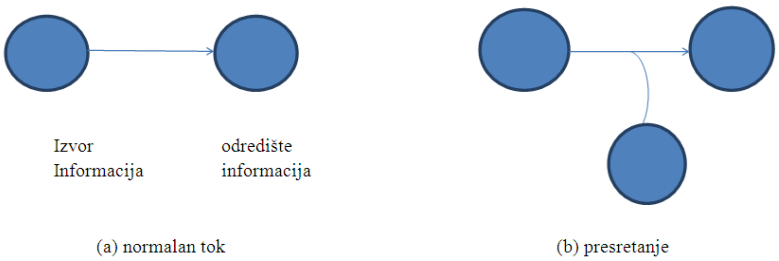
U osnovi,napadi su akcije koje su usmjerene na ugrožavanje sigurnosti informacija, računalnih sustava i mreža. (Pleskonjić i suradnici, 2007 : 2)

Postoje različite vrste napada, te se oni mogu klasificirati u četiri osnovne kategorije.
-*Presijecanje, tj. prekidanje* predstavlja napad na raspoloživost. Presijecanjem se prekida tok informacija, tj. onemogućava se pružanje neke usluge ili funkcioniranje nekog sistema.
Ovakav napad je aktivan.



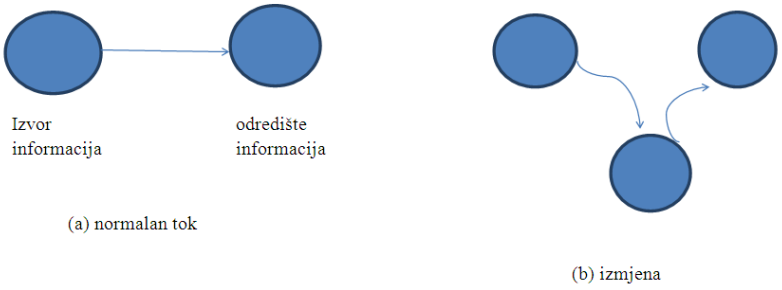
6.slika – Presijecanje - Pleskonjić i suradnici, 2007, str.2, Sigurnost računarsih sistema i mreža

-*Presretanje* predstavlja napad na povjerljivost. Presretanje može biti u praksi sprovedeno kao prisluškivanje prometa, nadziranje njegovog intenziteta, uvid u osjetljive informacije ili slično. Kao pasivan napad, teško se otkriva jer ne mijenja podatke tj. ne utječe na unutrašnje funkcioniranje sustava. Ovakav tip napada ponekad je pripremna faza za neku drugu vrstu napada.



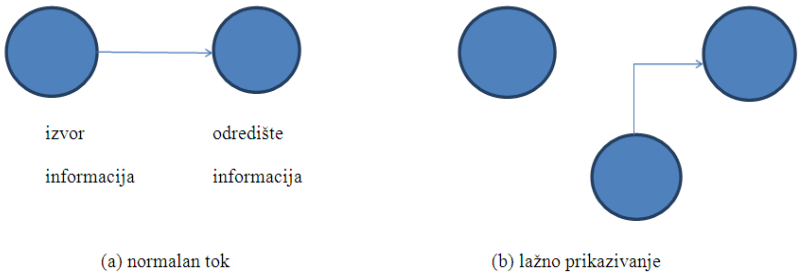
7. slika – Presretanje - Pleskonjić i suradnici, 2007, str.3, Sigurnost računarskih sistema i mreža

-*Izmjena* predstavlja napad na integritet. Po svojoj prirodi to je aktivan napad. Napad se može obaviti unutar nekog računalnog sustava – u tom slučaju radi se o izmjeni podataka, pristupnih prava, načina funkcioniranja programa ili sustava i slično. Iako mijenja podatke ili sustav, često ostaje neprimjećen neko izvjesno vrijeme, kako zbog nepažnje, tako i zbog složenih tehnika koje se pri ovom napadu koriste.



8. slika – Izmjena - Pleskonjić i suradnici, 2007, str.3, Sigurnost računarskih sistema i mreža

-*Lažno prikazivanje* - predstavlja napad na autentičnost. Napadač izvodi ovaj napad tako što priozvodi lažne podatke, lažni promet ili izdaje neovlaštene naredbe. Vrlo često se koristi i lažno predstavljanje korisnika, usluge, servera, web strane ili nekog drugog dijela sustava.



9. slika – Lažno prikazivanje - Pleskonjić i suradnici, 2007, str.3, Sigurnost računarskih sistema i mreža

2.1. RIZIK

Rizik je u kontekstu sigurnosti računalnih sustava i mreža, mjera opasnosti, tj. mogućnost da nastane oštećenje ili gubitak neke informacije, hardvera, intelektualnog vlasništva, prestiža ili ugleda. Obično se izražava u obliku jednadžbe rizika, gdje je :

Rizik = Prijetnja x Ranjivost x Vrijednost imovine (Pleskonjić i suradnici, 2007 : 5)

Prijetnja je protivnik, situacija ili splet okolnosti s mogućnošću ili namjerama da se eksploatira ranjivost. Prijetnja može biti strukturirana ili nestrukturirana. Strukturirane prijetnje su protivnici s formalnom metodologijom, financijskim sponzorom i definiranim ciljem. Takve prijetnje su karakteristične za ekonomsku špijunažu, organizirani kriminal, strane obavještajne službe i tzv informatičke ratnike. Prijetnje se dijele na aktivne i pasivne.

-*Pasivne* prijetnje ne utječu neposredno na ponašanje sistema i njihovo funkcioniranje. Tu spadaju otkrivanje sadržaja poruka (npr. prisluškivanje) i analiza prometa.

-*Aktivne* prijetnje mogu utjecati na ponašanje i funkcioniranje sustava ili sadržaj podataka. Tu spadaju: maskiranje, tj. pretvaranje, lažiranje, reprodukcija, tj. ponavljanje mrežnog prometa, izmjena sadržaja poruke i odbijanje usluge.

2.2. RANJIVOST

Predstavlja slabost u nekoj vrijednosti, resursu ili imovini koja može biti iskorištena tj. eksploatirana. Ranjivosti su posljedica lošeg projektiranja, implementacije ili „zagađenja".

-*Loše projektiranje* je greška projektanta sustava. Proizvođač koji piše loš kod,je kod koji sadrži greške, kao što je prekoračenje bafera na steku ili u dinamičkoj memoriji – pravi osjetljivi proizvod koji se može lakše „razbiti". Pametni napadači će iskoristiti slabosti u arhitekturi softvera.

-*Implementacija* je odgovornost klijenata koji instalira proizvod. Iako proizvođači trebaju pripremiti dokumentaciju o sigurnom korištenju svojih proizvoda, korisnik mora biti vrlo oprezan.

-*„Zagađenje"* se odnosi na mogućnost da se dostigne stupanj „iza" predviđene upotrebe proizvoda. Dobro projektiran softverski proizvod treba obavljati funkciju i ništa više od toga. Na primjer, ne smije postojati mogućnost da se iz mrežne usluge ili aplikacije koja se izvršava s privilegijama korisnika *root* na Linux sustavu, otvori instanca komandnog interpretera , jer će u tom slučaju korisnik dobiti na „poslužavniku"komandni interpreter sa svim pravima administratora sistema. Odluke koje ponekad donesu proizvođači i korisnici, mogu da prouzroče „zagađenje" tj. sa stvore mogućnost za prekoračenje predviđene upotrebe proizvoda.

Vrijednost imovine je mjera vremena i resursa potrebnih da se neka imovina zamijeni ili vrati u prethodno stanje. Zato se kao ekvivalentan termin može koristiti i „cijena zamjene". Server baze podataka na kome se čuvaju informacije o kreditnim karticama klijenata, podrazumijevano je vrijedniji, tj. ima veću cijenu zamjene nego radna stanica u nekom laboratoriju za ispitivanje softverskih proizvoda.

3. OSNOVNE SIGURNOSTI BAZA PODATAKA

Sigurnost je proces održavanja prihvatljivog nivoa rizika,što znači da nije završno stanje, tj. konačan proizvod. Organizacija ili institucija ne može se smatrati „sigurnom" u niti jednom trenutku nakon izvršene poslijednje provjere usklađenosti s vlastitim pravilima. To možemo zaokružiti jednom pričom, kad bi nas npr. šef upitao da li

smo sigurni, trebali bismo odgovoriti „Pričekajte da provjerim“, ili ako bi njegovo pitanje bilo „Da li ćemo biti sigurni sutra?“, naš odgovor bi glasio „ Ne znam.“ Takvi iskreni odgovori nisu popularni, ali uz takvo poimanje stvarnosti – poduzeća ili organizacije biti će uspješnije zaštićene. Rukovodioci koji shvaćaju koncept po kome je sigurnost proces održavanja prihvatljivog, tj. razumnog nivoa rizika, vjerojatno će odrediti vrijeme i resurse koji su potrebni da se ti zahtjevi i odgovornosti ostvare. Nerijetko se događa da velike svjetske grupacije, reklamiraju u raznim medijima svoje proizvode kao svemoćna rješenja, oni koji vjeruju da sigurnost može biti jednom dostignuta, i da će nakon toga sustav ostati siguran, voljni su da kupe proizvode i usluge koji se na taj način reklamiraju. Treba vrlo oprezno razmotriri tako oglašenu ponudu. (Pleskonjić i suradnici, 2007 : 9)

Kada se kaže da je sigurnost proces, onda se misli na činjenicu da se sigurnost ne može kupiti kao proizvod ili usluga, već je to proces u kome se koriste različiti proizvodi i usluge, procedure i pravila, ali se smatra i to da postoje drugi bitni elementi kao što su edukacija, podizanje svijesti i stalno praćenje stanja u ovoj oblasti. Ostvarivanje sigurnosti podrazumijeva održavanje sustava u stanju prihvatljivog rizika, tj. kompromis između potrebnih ulaganja i smanjenja mogućnosti da nastane šteta koja se tim ulaganjem postiže.

Kada se govori o sigurnosti i zaštiti informacijskih sistema i mreža, imamo nekoliko važnih principa:
**Sigurnost je proces*. Sigurnost nije proizvod, usluga ili procedura, već skup koji ih sadrži – uz još mnogo elemenata i mjera koje se stalno sprovode.
**Ne postoji apsolutna sigurnost*
**Uz različite metode zaštite*, treba imati u vidu i ljudski faktor, sa svim slabostima.

Veće ulaganje u sigurnost smanjuje izloženost sustava ili računske mreže riziku. S druge strane ono izlaže vlasnika sustava ili računske mreže većim troškovima i smanjenju profitabilnosti. Zato je veoma značajno da se odredi točka u kojoj se postiže ravnoteža između ulaganja u sigurnost i postignutih efekata. Treba imati u vidu da kao i u drugim sustavima i oblastima, sigurnosni mehanizmi ili procedure vrlo često smanjuju udobnost rada ili pogoršavaju performanse sistema. Kratkoročno gledano, to može negativno utjecati na opće efekte rada, a dugoročno ove mjere pozitivno utječu na uspjeh u radu, to jest, na profit komercijalnih organizacija. To se gleda i kroz materijalne pokazetelje, i kroz pokazatelje koji direktno nisu materijalni, kao što su rast ili gubitak ugleda, zavisno od toga da li se događaju incidenti ili ne.
Najvažniji faktori uspjeha su:
a) Aktivnosti koje se odnose na cijeli sigurnosni proces moraju biti zasnovane na zahtjevima posla i moraju ih voditi poslovna rukovodstva.
b) Neophodno je dobro razumijeti rizike od potencijalnih prijetnji i ranjivosti sustava.
c) Osnovni koncepti zaštite moraju biti izloženi svim rukovodiocima i zaposlenima kako bi svi shvatili koliko je zaštita bitna.
d) Institucionalna uputstva za primjenu pravila i standarada zaštite moraju se dostaviti svim zaposlenima i svim suradnicima koji nisu stalno zaposleni.

Sigurnost kao proces zasniva se na četiri osnovna koraka:

1. Procjena – smatra se posebnom akcijom, jer je u vezi sa pravilima, procedurama, pravnom i drugom regulativom, određivanjem budžeta i drugim upravljačkim dužnostima, i još je povezana s tehničkom procjenom stanja sigurnosti. Greška u procjeni bilo kojeg od ovog elementa, može naškoditi svim operacijama koje slijede.

2. Zaštita, tj. sprečavanje ili prevencija, podrazumijeva pripremu protiv mjera kako bi se smanjila mogućnost ugrožavanja sustava. Ukoliko zaštita zakaže, primjenjuje se slijedeći korak – otkrivanje.

3. Otkrivanje – ili detekcija predstavlja proces identifikacije upada, tj. povrede sigurnosnih pravila ili incidenata koji se odnose na sigurnost. Može se definirati kao svaki „nezakonit", neovlašten ili neprihvatljiv postupak koji je poduzet, a odnosi se na računarski sustav ili mrežu.

4. Odgovor – ili reakcija predstavlja proces oporavka, tj. liječenja posljedica upada. Ranije su se na prvo mjesto stavljalo popravljivanje funkcionalnosti oštećenih resursa, kao što je korištenje rezervnih kopija podataka za vraćanje sustava u stanje prije izvršenog napada. U novije vrijeme sve češće se koriste pravna sredstva (sudski procesi protiv onoga tko ugrožava sigurnost), među koja spada prethodno prikupljanje dokaza metodama digitalne forenzike pomoću kojih se potkrepljuje tužba.

3.1. KONTROLA PRISTUPA

Objekat baze podataka može se smatrati kao entitet koji zauzima određeni prostor u bazi podataka. ***Tabele*** jesu objekti koji služe za skladištenje podataka.
Indeksi – specijalne tabele – omogućavaju brz pristup podacima u tabeli (ukoliko je tabela indeksirana po atributu po kom se baza pretražuje.
Pogledi – omogućavaju izdvajanje podskupa informacija iz tabele ili grupe tabela (podskup redova ili kolona). Pogledi određuju koji podaci će biti prikazani i kako. U osnovi pogled predstavlja filter koji omogućava korisnicima da vide samo podskup redova i / ili kolone iz jedne ili više tabela. Pogledi su zapravo kvazi objekti jer ne skladište podatke. (Pleskonjić i suradnici, 2007 : 480).
Kontrola pristupa može se ostvariti na dva načina:
**Provjerom identiteta korisnika*
**Davanjem posebnih privilegija i prava*

Korisnici se na bazu prijavljuju pomoću korisničkih naloga koje priprema administrator baze podataka. Administrator korisnicima zadaje inicijalnu lozinku čiji se heš čuva u bazi, a korisnik može promijeniti svoju lozinku kada god želi. Korisnik mora unijeti ispravnu lozinku pri povezivanju s bazom da bi se spriječila neovlaštena upotreba. Korištenjem informacija

smještenih u bazi, provjerava se identitet korisnika. Administrator baze podataka može odrediti pravila za složenost lozinke, kojim bi se odredila minimalna dužina, ili obavezna upotreba malih slova, velikih slova i brojeva.
Lozinke za prijavljivanje korisnika na bazu ne treba čuvati u bazi podataka u obliku otvorenog teksta. Ukoliko se jedna tabela u bazi podataka koristi za čuvanje informacija o korisnicima nekog sistema (na primjer, atribut tabele su *korisničko_ime i lozinka*), poželjno je da i ove lozinke budu kriptografski zaštićene.
Neki sustavi za upravljanje bazama podataka osiguravaju više metoda za provjeru identiteta korisnika. Na primjer, Microsoft SQL Server podržava klasičan režim provjere identiteta (korisnik se prijavljuje pomoću korisničkog imena i lozinke koje SQL server uspoređuje s podacima u sistemskoj tabeli *sysxlogins*baze podataka *master*) i režim zasnovan na Windowa NT mehanizmu za provjeru identiteta. SQL Server se može konfigurirati tako da vjeruje Windows NT mehanizmu za provjeru identiteta. U tom slučaju, korisniku nije potreban poseban nalog za prijavljivanje na bazu podataka, već se prijavljuje automatski (pod uvjetom da mu je dozvoljen pristup bazi). U slučaju da se koristi mješani režim provjere identiteta, SQL server najprije provjerava da li korisnik postoji u sustavnoj tabeli *sysxlogins*, ukoliko ne postoji, SQL server će iskoristiti LSA podsistem da provjeri akreditive korisnika (Windows NT mehanizam za provjeru identiteta). (Pleskonjić i suradnici, 2007 : 481)

Korisnicima se dodjeljuju ovlaštenja za povezivanje na bazu i rad s njenim objektima. Ovlaštenja korisnicima može dodjeljivati administrator baze, vlasnik objekata ili neki drugi ovlašteni korisnik kome je dato to pravo. Ovlaštenja omogućuju korisnicima da obavljaju određene akcije nad bazom (sistemska ovlaštenja) ili objektima baze (objektna ovlaštenja).
**Sistemska ovlaštenja* najčešće dodjeljuje administrator baze podataka. U ova ovlaštenja spadaju, na primjer: CREATE DATEBASE, CREATE PROCEDURE, CREATE TABLE, CREATE VIEW i CREATE USER., koja dozvoljavaju korisniku da napravi novu bazu podataka, uskladištenu proceduru, tabelu, pogled i novi korisnički nalog.
**Objektna ovlaštenja* korisniku omogućavaju da izvrši operacije nad konkretnim objektima baze (kao što su tabele, pogledi i uskladištene procedure). Ako korisnik treba da vidi podatke u nekoj tabeli, potrebno mu je dodjeliti SELECT ovlaštenje nad tom tabelom (isto važi za INSERT, UPDATE, DELETE, ...). Ovaj vid zaštite podrazumijeva da se za svaku tabelu koja se nalazi u bazi posebno odrede prava pristupa za svakog korisnika. Na primjer, korisnik koji unosi podatke može imati samo pravo upisa (INSERT), dok drugi korisnik može samo vršiti izmjene (UPDATE).(Pleskonjić i suradnici, 2007 : 482)

Ovlaštenja se dodjeljuju korisnicima pomoću SQL naredbe GRANT, a oduzimaju pomoću naredbe REVOKE. Tabela koju uređuje neki korisnik je njegova, on je njen vlasnik. Drugi korisnik je načelno ne može koristiti ukoliko mu vlasnik eksplicitno ne dodjeli prava korištenja pomoću naredbe GRANT.

```
GRANT  privileges
[ON relation ]
```

```
To users
[WITH GRANTOPTION ]
```

```
REVOKE  privileges
[ON relation ]
FROM users
[WITH GRANTOPTION ]
```

Ove naredbe spadaju u grupu naredba za upravljanje podacima. Ako je korisnik dobio od vlasnika (ili ovlaštenog korisnika) neko objektno ovlaštenje sa opcijom (WITH GRANT-OPTION), onda i on može dodjeliti drugim korisnicima prava korištenja tog objekta,ali samo ista ili manja od onih koja je on dobio od vlasnika. Važno je shvatiti kako se ovlaštenja dodjeljuju korisnicima i šta koji korisnik može s njima uraditi. Ako nismo sigurni kako su ovlaštenja dodjeljena, možemo si zadati SQL naredbu SHOW GRANT – i ona će prikazati koja su ovlaštenja dodjeljena korisnicima baze. Nakon toga, možemo određenim korisnicima dodjeliti ovlaštenja koja su im potrebna ili ukinuti suvišna ovlaštenja. (http://www.cis.hr/files/dokumenti/CIS-DOC-2012-08-059.pdf)

3.2. INTEGRITET

U okviru sigurnosti informacija, koncept integriteta osigurava sljedeće:

a) Podatke ne smiju mijenjati neovlaštena lica ili procesi,
b) Ovlaštena lica ili procesi ne smiju obavljati neovlaštene promjene podataka,
c) Podaci su interno i eksterno konzistentni, što znači da su interni podaci međusobno konzistentni u svim dijelovima, kao i s realnim svijetom, tj. Vanjskim okruženjem.

Integritet kao usluga osigurava cjelovitost podataka, tj. Osigurava da napadač ne može izmjeniti podatke, a da to ostane neprimjećeno. Dakle, integritet je usluga zaštite od neovlaštenog, nepredviđenog ili nenamjernog modificiranja. Što se tiče podataka, oni moraju biti zaštićeni od neovlaštenih izmjena tokom skladištenja, obrade ili transporta, a sustav treba neometano izvršavati predviđene operacije (usluge) bez neovlaštenog manipuliranja. (Pleskonjić i suradnici, 2007 : 13)
Na primjer, jednosmjerna hash funkcija osigurava integritet dokumenata, i ukoliko netko izmjeni makar jedan znak u dokumentu, izmjeniti će i hash. Samim time će korisnici postati svjesni da je dokument izmijenjen.

3.3. AUTENTIFIKACIJA I AUTORIZACIJA

Da bi korisnik mogao pristupiti bazi i izvesti u njoj neku akciju, SUBP ga mora autentificirati i autorizirati. Pod autentifikacijom se podrazumijeva provjera identiteta korisnika. Upisivanjem korisničkog imena i zaporke korisnik se predstavlja SUBP – u, koji

potom pretražuje svoj katalog korisnika i pokušava pronaći zapis s identičnim imenom i zaporkom. Ako se takav zapis pronađe, upisani podaci su vjerodostojni, tj. korisnik je potvrdio svoj identitet. Osim upisivanja korisničkog imena i zaporke, SUBP – i obično podržavaju i neke druge metode autentifikacije. Tako se SUBP može osloniti na operativni sustav i autentificirati korisnika na osnovi ranijeg autentificiranja od operativnog sustava. Novije verzije SUBP – a obično podržavaju i mehanizme autentificiranja preko digitalnig certifikata. Kad SUBP autentificira korisnika, to samo znači da ga je „prepoznao“ i da se korisnik uspješno prijavio na SUBP. Ako autentificirani korisnik želi izvesti određenu akciju unutar SUBP – a, treba za to dobiti autorizaciju. Prilikom pokušaja izvođenja neke akcije, SUBP će provjeriti popis dopuštenja za tog korisnika, ustanoviti je li on za tu akciju ovlašten i prema tome dopustiti ili zabraniti njezino izvođenje. (Korbar, 2010: 74)

Korisničko ime i zaporka, nužni prilikom spajanja na SUBP u procesu autentifikacije, čine korisnički račun koji se naziva server login. SUBP – i mogu koristiti korisničke račune definirane u katalogu unutar operativnog sustava ili svoj poseban katalog. Kod konfiguracije SUBP – a odabere se s kojim od ta dva kataloga će se raditi, a obično postoji i mogućnost da se radi s oba istovemeno. Zaporke trebaju biti jake i trebaju se redovito mijenjati da bi se postigla što bolja zaštita od hakerskih napada i provala u SUBP. Logine možemo autorizirati za obavljanje određenih akcija na razini SUBP – a. Takve su akcije, na primjer, kreiranje novih baza u SUBP instanci, izvođenje sistemskih procedura ili kreiranje i upravljanje drugim loginima. Da bismo krajnjem korisniku dopustili da izvede bilo kakve akcije u određenoj bazi, u nekim SUBP – ima je pored logina potrebna još jedna vrsta korisničkog računa, nazvana **database user**. On je definiran na razini baze i preko njega se vrši autorizacija za izvođenje akcija u bazi. Svaki **database user** u bazi je povezan na neki login. Kad se login autentificira i preko njega se pokuša ući u bazu, njegov će se identitet prenijeti na usera s kojim je povezan. Useri nemaju zaporke i njih se ne autentfiicira, nego se samo autorizira za izvođenje akcija u bazi. (http://www.cis.hr/files/dokumenti/CIS-DOC-2012-08-059.pdf)

4. SIGURNOSNE PROCEDURE BAZE PODATAKA

Serveri baza podataka su među najvažnijim serverima svake tvrtke. Na njima se čuvaju podaci o proizvodima, klijentima, kao i razne financijske i računovodstvene informacije, dakle sve ono što tvrtku održava u poslu. Zbog toga baze podataka treba zaštititi od napada. Nekad prije zaštita ovih servera je bila relativno jednostavna, zato jer su oni bili fizički odvojeni i pristupalo im je samo ovlašteno osoblje tvrtke pomoću odgovarajućeg klijentskog softvera. Međutim, razvoj Interneta i troslojne klijent/server arhitekture dovelo je do toga da su mnogi serveri baza podataka praktično javno dostupni, tj.korisnici mogu da im pristupe preko posebne aplikacije koja se izvršava na Web serveru, a da pritom koriste samo čitač Weba. Web je jako pogodno i jeftino sredstvo za objavljivanje informacija. Treba imati u vidu činjenicu da se na jednome serveru baze podataka nalaze različiti podaci – na primjer, opis proizvoda i usluga koje neka tvrtka nudi i evidencija o plaćama zaposlenih u toj kompaniji. Koliko god da je za tvrtku bitno da informacije o proizvodima objavi na Internetu, na dinamički proizvedenimWeb stranicama koje koriste informacije iz baze, isto toliko je

bitno da se ostale podatke u bazi zaštiti, tj. da sprovede pravilne mjere kontrole pristupa bazi i spriječi krađu i zlouporabu informacija koje nisu namijenjene javnosti. Ovaj aspekt sigurnosti spada u tzv. sivu zonu odgovornosti. Punu odovornost za zaštitu baze podataka administrator mreže najvjerojatnije neće preuzeti, jer provjeravanje da li postoje SQL upiti u podacima koje unosi korisnik, zaista i nije njegov posao. (Pleskonjić i suradnici, 2007: 479). S druge strane, od projektanata i administratora baze podataka ne možete očekivati da vam konfiguriraju mrežnu barijeru i formiraju šifrirani tunel ka serveru baze podatka.

4.1. SIGURNOSNE KOPIJE

Za razliku od kopiranja nekih „običnih“ datoteka, poput slika, multimedijalnih ili tekstualnih datoteka, izrada sigurnosnih kopija baza je složeniji zadatak. Baze mogu biti velike i njihovo potpuno kopiranje može trajati jako dugo. Zbog toga će se ponekad izrađivati kopije samo onih podataka koji su se promijenili. Budući da se baze sastoje od podtaktovnih i log datoteka, treba znati kako uskladiti kopiranje tih dviju vrsta datoteka da bi se baze iz svojih kopija mogle vratiti u konzistentno stanje. (Korbar, 2010: 58)

Da bi se omogućilo vrijeme oporavka unutar granica koje zahtijeva poslovanje, a da se zbog izrade kopija ne remeti normalan rad sustava, treba znati koji tipovi kopija nam stoje na raspolaganju, koje su njihove karakteristike i kako se koriste. Tada možemo napraviti dobru strategiju izrade sigurnosnih kopija baza koje administriramo. Svaki SUBP ima svoje specifičnosti pa se tako i proces izrade sigurnosnih kopija i vrste kopija razlikuju od jednog do drugog SUBP –a .

Za vrijeme rada SUBP – a razne pogreške mogu uzrokovati nedostupnost baza ili oštećenje podataka u njima. Iako korištenje UPS sustava, zrcaljenje diskova, failover ili neke druge tehnike mogu povećati pouzdanost, i dalje se može dogoditi da zbog hardverske greške otkaže disk pa zbog toga podaci postanu nedostupni. Moguć je i kvar na memorijskim čipovima, matičnoj ploči ili nekoj drugoj hardverskoj komponenti, što za posljedicu može imati i pad cijelog sustava.

Softverske pogreške mogu uzrokovati probleme u funkcioniranju SUBP –a. To mogu biti interne greške u samom SUBP –u, greške u operativnom sustavu ili u nekom drugom softveru koji je povezan s SUBP – om. Iako one obično ne uzrokuju oštećenje podataka, može se dogoditi da zbog njih cijela SUBP instanca postane nedostupna ili da pojedini dijelovi sustava ne rade ispravno.

Najčešće su ljudske pogreške, kao na primjer pogrešno ažurirane ili nehotično brisanje podataka od strane krajnjih korisnika, ali i od strane administratora baze. Administrator može pokrenuti neke skripte u pogrešno vrijeme ili s krivim parametrima pa na taj način pokvariti podatke. Što se prije nastale pogreške uoče i isprave, šteta će biti manja.

Zadatak administratora baza je vraćanje baze u stanje prije nastanka problema i omogućavanje normalnog nastavka rada. Taj se postupak naziva oporavkom. Da bi se oporavak baza mogao izvesti, moramo imati kopije baza iz vremena prije nego se greška dogodila. Takve kopije, iz kojih se baze u slučaju potrebe mogu oporaviti, zovu se sigurnosne kopije baza.

Sigurnosne kopije se mogu izrađivati dok je baza on-line, tj. za vrijeme dok su korisnici spojeni na bazu i izvode po njoj čitanja ili zapisivanja. Kod takvog kopiranja potebno je imati zabilježene i promjene koje su se nad podacima događale dok je kopiranje trajalo. Te su promjene zapisane u log datoteci. Prilikom oporavka baze iz on-line sigurnosne kopije, SUBP će nakon vraćanja podataka iz podtaktovnih datoteka morati pročitati kopirane log podatke i ponoviti transakcije koje su potvrđene, odnosno poništiti one nepotvrđene. Na taj se način osigurava konzistentnost podataka nakon oporavka. Ovaj tip sigurnosne kopije se još naziva *hot backup*. Sigurnosna kopija može se napraviti i nakon što se baza stavi off-line. Tada se SUBP pobrine da podaci u tako „spuštenoj" bazi budu konzistentni. Nakon toga, dovoljno je kopirati samo podaktovne datoteke, jer za vrijeme kopiranja nema nikakvih promjena nad bazom. Oporavak off-line kopija je brži, ali ponekad poslovanje zahtijeva stalnu dostupnost pa takva kopiranja nisu izvediva.(Korbar, 2010: 59). Ovakve kopije ponekad se nazivaju *cold backups*.

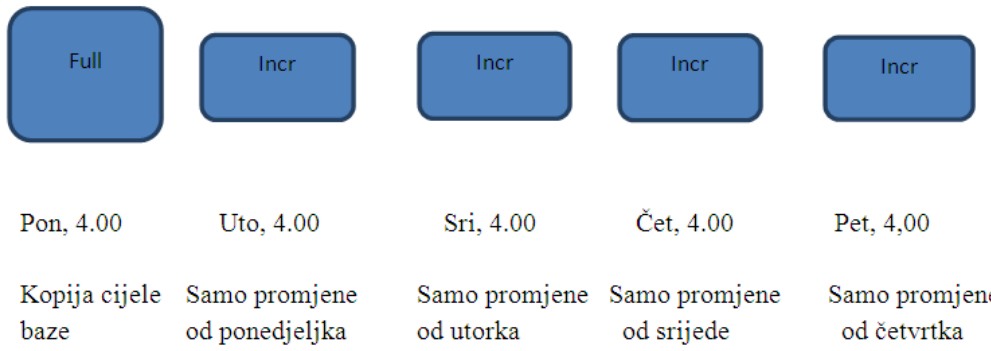
Potpuna sigurna kopija (full backup) sadrži sve stranice svih podtaktovnih datoteka baze. Za razliku od nje, inkrementalna kopija sadrži samo stranice podtaktovnih datoteka promijene nakon kreiranja posljednje potpune ili inkrementalne kopije.

SUBP – i se razlikuju po tome koje mogućnosti kod kreiranja inkrementalnih kopija nude. Razlikuju se i nazivi koji se upotrebljavaju. U Oracleu, na primjer, postoje dvije vrste inkrementalnih kopija – *diferencijalne i kumulativne*. Diferencijalne sadrže samo promjene nastale nakon zadnje inkrementalne kopije, a kumulativne sadrže promjene nastale nakon zadnje potpune kopije. Za razliku od toga, SQL Server ima samo jednu vrstu inkrementalnih kopija koje Microsoft nazova diferencijalnima, a one sadrže sve promjene nastale od zadnje potpune kopije.

Kad su baze velike, obično se pojavljuje problem s trajanjem potpunog kopiranja. Da bi se on zaobišao, izrađuju se inkrementalne kopije, što može trajati znatno kraće. Ali ako treba vratiti bazu koja je bila inkrementalno kopirana, takav će oporavak duže trajati jer je potrebno prvo restaurirati bazu iz potpune kopije, a onda na nju primijeniti promjene iz inkrementalne kopije.

U primjeru sa slike 7. Potpuna kopija radi se ponedjeljkom u 4.00. nakon toga, utorkom, srijedom i četvrtkom izrađuju se inkrementalne kopije u isto vrijeme. Ako u četvrtak u 11.00 baza padne, oporavak će se izraditi prvo restauriranjem baze iz potpune kopije od ponedjeljka, a nakon toga će se primjeniti promjene iz inkrementalne kopije od četvrtka (ovisno o tipu inkrementalnih kopija, može biti potrebno restaurirati promjene iz svih inkrementalnih kopija redom). (Korbar, 2010: 59)

Raspored izrade potpunih i inkrementalnih kopija.



7. slika - Raspored izrade potpunih i inkrementalnih kopija, Korbar, 2010, str.59, Administriranje baza podataka

Inkrementalno kopiranje pogodno je kad su baze velike, ali u njima nema puno promjena podataka. Tada ušteda na vremenu kopiranja može biti puno važnija nego neznatno povećanje vremena za oporavak. Kod malih baza, kod kojih potpuno kopiranje ne traje dugo, nije potrebno raditi inkrementalne kopije jer se na taj način komplicira procedura kopiranja, a pogodnosti nisu velike.

Neki SUBP – i imaju mogućnost analiziranja količine promjena u bazi pa mogu sami odlučiti hoće li se napraviti potpuna ili inkrementalna kopija. Administratori mogu podesiti graničnu vrijednost, tj. postotak promijenjenih podataka iznad kojega će SUBP raditi potpunu kopiju. Općenito, kad se promijeni 40 % ili više podataka, inkrementalne kopije nisu dobar izbor.

Kod nekih SUBP – a postoje pomoćni programi za spajanje inkrementalnih kopija. Više inkrementalnih kopija može biti spojeno u jednu ili se one mogu spojiti s potpunom kopijom, stvarajući na taj način novu potpunu kopiju. Takvo spajanje je dobro pokrenuti odmah nakon izrade inkrementalne kopije. Pri spajanju tih kopija nemamo problema s ometanjem normalnog rada baze, a također smanjujeno vrijeme potrebno za oporavak. (Korbar, 2010: 60)

Kod velikih baza, gdje su izrade sigurnosnih kopija dugotrajne, mogu biti korisne mogućnosti kopiranja samo pojedinih dijelova baze. Obično se to odnosi na skup fizički povezanih tablica. Tako u Oracleu postoji mogućnost da se naprave sigurnosne kopije pojedinih tablespaceova. U SQL Serveru se skup tablica može smjestiti u različite datoteke, a svaka datoteka se može zasebno kopirati.

Ako imamo baze u kojima postoji puno arhivskih podataka, koji se samo čitaju i više se ne ažuriraju, onda je tablice dobro razdvojiti tako da se one arhivske ne nalaze u istim datotekama kao i one nad kojima se događaju promjene. Tada možemo uspostaviti strategiju u kojoj jednom napravimo potpunu kopiju baze, a nakon toga radimo kopiranje samo onih datoteka s tablicama čiji podaci se mijenjaju. Podaci arhivskih tablica se ne trebaju kopirati često.
Kod kreiranja rasporeda po kojem će se sigurnosne kopije za neku bazu kreirati, važno je napraviti dobar balans između dva suprotna zahtijeva: onog da se kopije izrađuju što češće kako bi se smanjilo kasnije vrijeme oporavka i onog da izrade kopija što manje ometaju normalan rad sustava. Nisu svi podaci u bazi jednako važni. Neki mogu biti presudni za poslovanje tvrtke, dok su drugi sporedni ili su dostupni i na nekim drugim mjestima. Prije nego se uspostavi strategija i raspored za izradu sigurnosnih kopija svakako bi bilo dobro analizirati poslovanje te prirodu podataka i njihovu važnost. U toj analizi treba dati odgovore na pitanja:

**Koliko se često podaci mijenjaju?*
**Koliko su podaci važni za poslovanje?*
**Mogu li se određeni podaci lako rekreirati iz nekih drugih izvora?*
**Koliko je loše ako određene podatke izgubimo?*
**Postoji li vrijeme u kojem nitko ne treba raditi s bazom?* Je li potrebna dostupnost 24/7?
**Koliko je skupo vrijeme nedostupnosti sustava zbog oporavka*?(Korbar, 2010: 61)

4.1.1. PREPORUKE ZA IZRADU SIGURNOSNIH KOPIJA

Kad se napravi sigurnosna kopija baze, dobro ju je još jednom kopirati da ne bismo ostali bez ikakve kopije u slučaju oštećenja diska ili trake. Preporučljivo je čuvati barem dvije generacije sigurnosnih kopija. Ako se ispostavi da se iz nekog razloga baza ne može oporaviti iz zadnje sigurnosne kopije, možda ćemo biti u stanju vratiti bazu iz prethodne kopije (ili barem ublažiti gubitak podataka). Ako se sigurnosna kopija sprema izravno na traku, to je sporije nego da se spremi na disk. Sigurnosnu kopiju možemo kasnije kopirati na traku, ali je poželjno zadržati je i na disku jer će u slučaju potrebe za oporavkom biti brže uzeti je s diska, nego s trake. Nakon izrade sigurnosne kopije, obavezno treba provjeriti je li ispravna i može li se iz nje napraviti oporavak. U tu svrhu u SUBP – ima postoje pomoćni alati, a može se napraviti i pravi oporavak na za to prilagođenoj okolini.
Podaci koji nisu pohranjeni u bazi, ali ih aplikacije koriste, također se trebaju kopirati paralelno s bazom. To se odnosi na dokumente ili neke druge datoteke koje su pohranjene na disku, a u bazi je navedena samo njihova putanja.
U plan izrade sigurnosnih kopija treba uključiti i kopiranje sistemskih baza i datoteka. Iako korisničke baze mogu biti ispravne, greške u sistemskim datotekama mogu prouzročiti da cijela SUBP instanca prestane raditi. Oporavak sistemskih baza u takvim situacijama obično je bolji izbor, nego uspostava novog okruženja s jednakim podešenjima i istim bazama kao u starom okruženju. Prije nego se namjerava napraviti promjena u sistemskom katalogu (na primjer, prije kreiranja nove ili brisanja postojeće baze), preporučljivo je kopirati ga. Nakon

što se napravi point-in-time recovery, dobro je napraviti potpunu kopiju baze. To će osigurati da sljedeći oporavak možemo izvesti brže ako se pad baze ubrzo ponovi. (Korbar, 2010: 62)

4.2. OPORAVAK BAZA PODATAKA

Da bi se shvatilo što se događa za vrijeme oporavka baze, treba razumijeti na kojem principu rade transakcijski logovi. Moderni SUBP – i imaju takozvani write-ahead log, što znači da se niti jedna promjena ne zapisuje u podtaktovnu datoteku prije nego se zabilježi u logu. Da bi SUBP napravio promjenu nad nekim podatkom, on prvo mora učitati stranicu s tim podatkom u međuspremnik. Promjena podatka se događa najprije u međuspremniku, a zatim se mora propagirati na disk. Kad se dogodi promjena u međuspremniku, odgovarajući zapis koji opisuje tu promjenu dodaje se u transakcijski log. Ali promjena se ne bilježi istog trena u podaktovnoj bazi. Zbog toga stranica u međuspremniku ima drugačije podatke nego na disku. Takve stranice u međuspremniku nazivaju se prljavim stranicama.(Korbar, 2010: 63)
SUBP u intervalima okida posebne akcije koje se nazivaju checkpoints. Kad se dogodi checkpoint, sve prljave stranice iz međuspremnika pohranjuju se na disk u podtaktovnu datoteku. Taj proces ažuriranja stranica na disku naziva se flushing.

Ako u nekom trenutku baza padne – tada u podtaktovnoj datoteci mogu postojati neke stranice koje su u procesu flushinga ažurirane na disku, ali pripadajuće transakcije su prekinute padom baze pa tako nisu potvrđene. S druge strane, iako u transakcijskom logu mogu biti zabilježene potvrde nekih transakcija, podaci se u podtaktovnoj datoteci nisu ažurirali jer se zbog pada baze nije dogodio novi checkpoint. To ostavlja podatktovnu datoteku u nekozistentnom stanju.

Pod oporavkom baze podrazumijevamo vraćanje baze i podataka u njoj u stanje kakvo je bilo u nekom trenutku u prošlosti. Taj trenutak nazivamo točkom oporavka. Ako se dogodio pad baze ili sustava, obično se treba vratiti u trenutak neposredno prije pada. Ponekad se mogu dogoditi pogrešna ažuriranja podataka pa se treba vratiti u trenutak prije nego je počela transakcija koja je obavila ta ažuriranja. Važno je istaknuti da oporavak mora završiti dovođenjem baze u konzistentno stanje, što znači da integritet podataka mora biti sačuvan.

Oporavak se ne mora nužno izvoditi vraćanjem podataka iz sigurnosne kopije, nego i iz logičkih kopija ili nekim drugim metodama. Ako koristimo sigurnosne kopije, tada govorimo o restauriranju baza. Tu podrazumijevamo prvu fazu u procesu oporavka (vraćanja podataka), nakon koje slijedi uspostava konzistentnosti. Oporavak ne mora značiti da se dogodio neki problem s bazom. Proces u kojem se baza automatski dovodi on-line nakon restartanja SUBP instance također je oporavak baze.

Da bi se baza u procesu oporavka uspješno dovela u konzistentno stanje, čita se i analizira transakcijski log. One transakcije koje su potvrđene, ali iza kojih nije slijedio niti jedan checkpoint, moraju se ponovno izvesti. Promjene nakon tih transakcija moraju se zapisati u

podtaktovnu datoteku. Taj se postupak zove roll forard. Kod transakcija koje nisu potvrđene može se dogoditi da je neki checkpoint već pohranio dio „prljavih“ podataka na disk pa efekte takvih transakcija treba poništiti – to se naziva roll back.

U procesu oporavka baze obično se poduzimaju sljedeće aktivnosti:

*Ustanovi se da postoji problem. Ako se bazi ne može pristupiti kroz administrativne alate ili se u aplikaciji pojavljuju poruke o nedostupnosti baze, očito je došlo do problema. Ponekad se čini da baza normalno radi, a zapravo se događaju teže uočljive greške zbog kojih će bazu trebati vratiti u neko ranije, ispravno stanje.
**Utvrdi se uzrok problema*. Treba saznati što je uzrokovalo problem u bazi, a to će obično odrediti i metodu opoavka. Ta aktivnost uzima najviše vremena.
**Određuju se podaci i dijelovi baze koje treba oporaviti*. Administrator utvrđuje koji su objekti baze pogođeni i zahtijevaju oporavak. Za to može biti potrebno mnogo vremena.
**Utvrđuju se dodatni objekti koji ovise o onima koje treba opora*viti. Iako se greška može dogoditi na samo jednoj tablici, treba provjeriti kako će oporavak utjecatina ostale tablice koje su s njom povezane.
**Lociraju se potrebne sigurnosne kopije*. Treba pronaći sigurnosne kopije iz koji će se baza oporaviti u najkraćem vremenu. U obzir treba uzeti i mogućnost da se kopije nalaze na nekoj udaljenoj lokaciji.
**Baze se restauriraju iz sigurnosnih kopija*. Restauracija se izvodi preko pomoćnih programa operativnog sustava ili samog SUBP-a.
**Baze se dovedu u konzistentno stanje*. Preko pomoćnih programa, nad transakcijskim logom se naprave roll forward back operacije. U nekm će se SUBP-ima restauracija i dovođenje u konzistentno stanje napraviti u jednom koraku.

4.2.1. TIPOVI OPORAVKA BAZA PODATAKA

Oporavak do točke pada – ovaj tip oporavka obično se radi nakon pada baze. Za njega je značajno da se baza treba vratiti u stanje kakvo je bilo u trenutku neposredno prije pada sustava. Bazu oporavljamo tako da napravimo restauraciju iz najsvježije potpune kopije koju imamo, nakon toga vraćamo podatke iz inkrementalnih kopija (ako postoje), zatim radimo roll forward iz arhiviranih transakcijskih logova i na kraju iz aktivne log datoteke. Da bi se baza dovela u konzistentno stanje, na kraju se još napravi roll back transakcije koje nisu potvrđene.
Ako nemamo potpunu kopiju baze, najvjerojatnije nećemo moći napraviti oporavak. U nekim bi se iznimnim situacijama oporavak bez potpune kopije mogao napraviti samo na osnovi kopija log datoteka, čuvanih od početka rada baze. Dogodi li se da baza padne jer je uništen aktivni log, nećemo je moći vratiti u trenutak pada sustava, nego u neki dalji trenutak u prošlosti (u onaj kad je zadnji put arhiviran transakcijski log).

Oporavak do odabranog trenutka – podrazumijeva vraćanje baze u stanje kakvo je bilo u odabranom trenutku u prošlosti. Primjenjuje se u situacijama kad se zbog softverske ili

ljudske pogreške naprave kriva ažuriranja ili brisanje podataka, koja nije lako na drugačiji način ispraviti. Tada se obično baza vraća u stanje neposredno prije izvođenja akcije koja je prouzročila grešku. Točka oporavka može se odrediti kao datum i vrijeme ili kao adresa u transakcijskom logu do koje se treba vratiti. Ako SUBP dozvoljava da se u transakciji log zabilježe posebne oznake, onda prilikom oporavka možemo zadati točku oporavka navodeći upravo tu oznaku. U točku oporavka možemo doći restauriranjem baze iz potpune sigurnosne kopije, a nakon toga se uzimaju transakcije iz arhivskih logova ili aktivnog loga te se provode roll forward i roll back kako bi se baza vratila u konzistentno stanje.

Oporavak transakcije – za razliku od prethodna dva tipa, gdje je potrebno vratiti cijelu bazu i kod kojih je za vrijeme oporavka baza nedostupna, ovdje se može u staro stanje vratiti samo dio baze. Dok teče oporavak tog dijela, ostatak baze može biti dostupan drugim procesima.
Oporavak transakcije koristi se u situacijama u kojima se žele ukloniti efekti određenih transakcija koje se se dogodile unutar vremenskog okvira. To su transakcije sa stajališta korisnika, a ne transakcije kako ih vidi SUBP. Npr. sve aktivnosti koje je izveo određeni korisnik od jučer u 16.00 sati mogu činiti jednu korisničku transakciju.
Problemi koji se mogu pojaviti i zbog kojih bismo mogi poželjeti napraviti oporavak transakcije u osnovi su isti kao kod oporavka do odabranog trenutka, dakle obično se radi o softverskom bugu ili ljudskoj greški, zbog koje u bazi nastanu krivi podaci. Za razliku od oporavka od odabranog trenutka, oporavak transakcije je pogodniji ako zbog testiranja želimo nad nekim podacima izvesti određenu proceduru, a onda dobiveni efekt poništiti i pokušati sve iznova nakon što proceduru malo modificiramo.
Oporavak transakcije možemo izvesti na dva načina: da uklonimo samo efekte loših transakcija ili da uklonimo sve transakcije nakon zadane točke i onda ponovno izvedemo samo dobre transakcije. (Korbar, 2010: 67)

Ponekad se može dogoditi da se podaci u stranicama pokvare. Prilikom zapisivanja podataka u stranicu, na osnovi njihovih vrijednosti izračuna se kontrolna suma i ona se upiše u zaglavlje stranice. Kod čitanja, ta se kontrolna suma ponovno izračuna i usporedi s onom koja je već zapisana u zaglavlju. Ako te dvije sume nisu jednake, to znači da su podaci u toj stranici pokvareni. U zaglavlju može biti pogrešan identifikator ili dio podatka iz stranice može biti izbrisan. Ako za vrijeme rada SUBP naiđe na takvu stranicu, najčešće prijavi grešku s identifikatorom pokvarene stranice i prekine izvođenje započete akcije. Pokvarene stranice mogu se detektirati i pokretanjem posebnih pomoćnih programa koji služe za tu svrhu. Da bi se problem s pokvarenim stranicama otklonio, ponekad će možda biti dovoljno samo restartati SUBP instancu. Neki SUBP-i imaju ugrađene pomoćne programe za popravak pokvarenih stranica. Prije pokretanja takvih alata treba pročitati upute za njihovo korištenje i razumijeti što oni točno rade da se pogrešnom upotrebom ne bi učinila još veća šteta. SUBP proizvođači preporučuju izvođenje takvih radnji samo pod nadzorom njihovih stručnjaka.
Najizglednije je da će se trebati napraviti restauriranje tih stranica iz sigurnosnih kopija. Može se uzeti zadnja potpuna kopija baze i krenuti u restauriranje, s naznakom da se restaurira samo određena stranica. Nakon toga se primijene i inkrementalne kopije te logovi. Ovakav oporavak može se izvoditi on-line, ali oni objekti baze, koji se nalaze u istoj datoteci kao i

pokvarena stranica, ne mogu biti dostupni drugim korisnicima za vrijeme oporavka.(Korbar, 2010: 68)

Za plan oporavka baze potrebno je učiniti sljedeće:

** Razraditi detaljno svaki korak oporavka za sve baze i dokumentirati ga*
** Uključiti u dokumentaciju sve skripte koje su potrebne za oporavak*
** Usuglasiti plan sa svim osobama koje će morati sudjelovati u oporavku i zabilježiti njihove podatke za kontakt.*
** Ažurirati po potrebi plan nakon što se dogode promjene poput dodavanja ili brisanja baza.*

Situacije u kojima treba oporavljati produkcijske baze ipak nisu tako česte. Ali, kad se dogode, trebaju se napraviti hitno pa je vjerojatno prisutan veliki pritisak i stres. Zbog toga treba učiniti sve da u takvim situacijama stvari teku glatko. Testiranje plana oporavka nužno je da bi se provjerilo rade li sve skripte kako treba i mogu li se baze na taj način zaista oporaviti. Testiranje je poželjno provoditi redovno jer će to administratoru služiti kao svojevrstan trening koji će mu omogućiti da bude što spremniji kad se dogodi potreba za oporavkom. (Korbar, 2010: 69).

5. UPRAVLJANJE SIGURNOŠĆU UNUTAR SUSTAVA UPRAVLJANJA BAZA PODATAKA

Administratori mogu kontrolirati sigurnost u bazama koristeći naredbe GRANT i REVOKE iz DCL – a . Pomoću njih se logira ili userima daju prava da izvode određene akcije nad bazom, odnosno objektima u bazi. Naredbom GRANT dodjeljujemo neko pravo, a sa REVOKE poništavamo prethodno dodijeljeno pravo. Ako za nekog korisnika nije dano eksplicitno pravo GRANT naredbom da izvodi neku akciju, onda SUBP primjenjuje svoje predefinirano ponašanje. To znači da korisnik neće imati pravo izvršiti tu akciju. No, kako korisnici mogu biti članovi grupa, moguće je da je određenoj grupi korisnika dano neko pravo. Tako korisnik može preko pripadnosti grupi dobiti dopuštenje da nešto napravi u bazi, bez obzira što za njega samoga to pravo nije dodijeljeno. Microsoft je u svoj T-SQL uveo i naredbu DENY kojom se može eksplicitno zabraniti izvođenje određene akcije.

Svaki SUBP ima nekoliko tipova prava, kao što su pravo na pristup podacima, kreiranje objekata u bazi ili izvođenje sistemskih procedura. Pored toga, SUBP – i mogu imati i neke dodatne tipove prava, ovisno o funkcionalnostima koje podržavaju. Uobičajeni tipovi prava su:
* *Prava na tablice*: za kontrolu pristupa i modifikacije podataka u tablicama – ovaj tip prava daje se da bi se korisnicima omogućio pristup tablicama ili pogledima. Za tablice i poglede mogu se postaviti prava na izvođenje sljedećih naredbi: SELECT – za čitanje podataka iz tablica/pogleda, INSERT – za upis novih slogova u tablice/poglede, UPDATE – za ažuriranje podataka u tablicama/pogledima, DELATE – za brisanje zapisa iz tablica/pogleda, ALL – za

izvođenje svih gore navedenih naredbi. Obično se prava na tablice daju programerima zbog razvojnih potreba. Krajnjim korisnicima ne daju se prava izravno na tablice, nego se pristup tablicama kontrolira preko pogleda ili pohranjenih procedura.
* *Prava na bazu*: za kontrolu kreiranja, modoficiranja i brisanja objekata unutar baze. Preko prava na bazu može se kontrolirati kreiranje objekata unutar baze, poput tablica, indeksa, okidača, pohranjenih procedura, funkcija ili korisnički definiranih tipova podataka. Prava na kreiranje objekata u bazi najčešće imaju samo administratori baza podataka. Ako se ta prava dodijele i drugima, može postati teško kontrolirati njihovo nastajanje. Tako se može lako upasti u situaciju da imamo previše tablica ili pohranjenih procedura u bazi, a da ne znamo koja je njihova uloga i koriste li se uopće. Zbog toga je preporučljivo ostaviti prava za kreiranje objekata u bazi samo aministratorima i, prema potrebi, programerima koji su dovoljno stručni na području baza podataka. (Korbar, 2010: 78)
* *Prava na procedure*: kontroliraju tko smije izvršavati određene procedure i funkcije. Ova se prava odnose na kontrolu izvođenja pohranjenih procedura i drugih programskih objekata unutar baze. Naredba za izvršavanje procedure je EXECUTE. Ako je potrebno korisniku dopustiti mijenjanje podataka u nekim tablicama, bolje je napisati procedure koje to rade, zabraniti korisniku direktno mijenjanje tablice, a dati mu pravo na izvođenje procedure.
* *Sistemska prava*: kontroliraju tko može obavljati razne sistemske aktivnosti na razini SUBP – a . Njima se kontrolira korištenje određenih značajki SUBP – a i izvođenje sistemskih naredbi. O samom SUBP – u ovisi koje će to konkretne naredbe biti, a najčešće su to naredbe poput kopiranja baza i arhiviranje transakcijskih logova, gašenja i restartanja SUBP – a, pokretanja automatiziranih zadataka, kreiranja baza te davanja prava drugim korisnicima. Sistemska prava ne dodjeljuju se userima na razini baze, nego loginima na razini cijelog SUBP – a. Sistemska prava trebaju se dodjeljivati s velikim oprezom i općenito bi trebala biti rezervirana samo za administratore SUBP – a.

Da bi korisnik mogao koristiti DCL naredbe nad nekim objektom u bazi, on mora biti vlasnik tog objekta ili mora biti u nekoj od sistemskih grupa korisnika s visokim privilegijama (na primjer, u grupi sistemskih administratora SUBP –a). Postoji i mogućnost da se nekome dodijeli dopuštenje da odobri drugima izvođenje određene naredbe na određenom objektu u bazi.

Prema tome tko sve ima pravo na izvođenje DCL naredbi, administraciju sigurnosti baza možemo podijeliti na centraliziranu i decentraliziranu. Ako se često koristi WITH GRANT OPTION, imat ćemo decentraliziranu administraciju. U takvoj situaciji administrator je raterećen jer i obični korisnici mogu davati prava ostalima, ali je tada teže uspostaviti kontrolu nad davanjem prava pa se može dogoditi da neki korisnici daju drugima određena prava premda sam administrator to ne bi učinio. U centraliziranoj administraciji pravo dodjele prava drugima ima samo manji broj korisnika. Najčešće su to korisnici koji se nalaze u grupi administratora SUBP – a. U takvom načinu rada lako je kontrolirati kome se daju kakva prava, ali je negativno to što velik posao administracije sigurnosti pada samo na jednu ili nekimanji broj osoba, koje su onda preopterećene.

Prava se osim pojedinačnim korisnicima, mogu dodjeljivati i grupama korisnika. Kad se neko pravo dodijeli grupi PUBLIC, onda je ono dodijeljeno svakom korisniku koji se može prijaviti na SUBP.

Administriranje sigurnosti u bazama podataka može biti vrlo složeno. Korištenje prečica, poput dodjeljivanja nekih prava svima odjednom korištenjem grupe PUBLIC, može ponekad značajno olakšati posao. Može bit korisno i u nekim drugim slučajevima, na primjer, ako imamo pohranjenu proceduru koja se koristi samo iz aplikacije, možemo pravo izvođenja te procedure dodijeliti grupi PUBLIC, a aplikaciji prepustiti da se pobrine o tome hoće li neke korisnike ograničiti u korištenju procedure. U dodjeljivanju prava grupi PUBLIC treba biti oprezan da korisnici ne bi zloupitrijebili svoje pravo pristupa određenim objektima u bazi. Stoga ovu metodu treba primjenjivati samo u rijetkom situacijama, u kojima smo zaista sigurni da ništa ne može proći po zlu.
Informacijski sustav se mijenja pa se uvode nove aplikacije ili radnici prelaze u druge odjele. To zahtijeva i promjenu prava za pojedine korisnike, a tada se pojavljuje potreba da se pregleda tko trenutačno ima kakva prava u SUBP – u. Te se informacije spremaju u sistemskom katalogu, zato je zapravo jedna ili više tablica iz kojih možemo pročitati podatke SELECT naredbom. Postoje i pogledi ili sistemske procedure da olakšaju dohvaćanje tih podataka.

Da bi se olakšala administracija sigurnosti u bazama, SUBP – i imaju mogućost grupiranja korisnika kojima je potrebno dodjeliti ista prava. Takve se grupe nazivaju autorizacijskim grupama. Administratori mogu kreirati specifične grupe prilagođene trenutačnim potrebama (korisnički definirane grupe), a postoje i ugrađene grupe koje nije moguće mijenjati. Ako administrator treba većem broju korisnika baze dati ista prava, bit će naporno za svakog od njih davati ta prava jedno po jedno. Da bi si olakšao posao, on može kreirati grupu i sva potrebna prava dodijeliti njoj. Nakon toga, sve korisnike učlani u tu grupu i tako jednim potezom svima dodijeli jednaka prava.

U svakom SUBP – u postoje ugrađene grupe korisnika kojima su dodijeljena određena prava izvršavanja akcija unutar SUBP – a. Iako se te grupe korisnika značajno razlikuju i po nazivu, i po pravima od jednog do drugog SUBP – a, ipak se mogu izdvojiti neki tipovi grupa, koji svugdje postoje:
* *Administrator sistema* – to je grupa s najvećim ovlastima unutar SUBP – a. Korisnik kojem su dodijeljena prava obično može izvršavati sve naredbe i pristupati svim bazama i objektima unutar njih. Najčešće je onvlasnik sistemskih resursa i sistemskih tablica.
* *Administrator baze* – ova grupa ima sva prava nad bazom na koju se odnosi, zajedno s pravom da čita, ali ne i modificira podatke u tablicama iz te baze. Korisnici u ovoj grupi imaju pravo obrisati i promijeniti bilo koji objekt u bazi (tablicu ili indeks).
* *Održavanje baze* – ova grupa ima specifična prava vezana za održavanje objekata unutar baze (pokretanje naredbi ili pomoćnih programa za poslove održavana baze, ažuriranje statistike i sl.). Definirana je na razini baze i njezina se prava odnose na bazu u kojoj se nalazi.
* *Administrator sigurnosti* – ova grupa sadrži skup prava potrebnih za uspostavu sigurnosti untar SUBP – a. Korisnici iz te grupe mogu administrirati korisnike i njihove zaporke, davati

prava i postavljati zabrane koristeći GRANT, DENY i REVOKE, nadgledati korištenje objekata u bazi i korištenje prava.
* *Operacijski zadaci* – ova grupa ima prava za izvođenje operativnih poslova poput izrade sigurnosnih kopija, oporavka baza ili terminiranja problematičnih procesa. (Korbar, 2010: 81)

Korisnici administrator sistema imaju sva prava na SUBP – u, te je poželjno ograničiti broj takvih korisnika na iskusne administratore baza podataka. Krajnji korisnici, voditelji odjela ili programeri aplikacija nikako ne bi trebali biti uključeni u tu grupu. SUBP – i imaju mogućnost da određenu razinu pristupa bazama postave na cijelu aplikaciju koristeći aplikacijske role. U tom slučaju korisnici ne mogu bazi pristupiti izravno, nego samo kroz aplikaciju, odnosno aplikacijsku rolu. Tada se aplikacijskoj roli daju sva prava potrebna za izvođenje aplikacije, a aplikacija mora sama implementirati sigurnosne mehanizme da bi se odredila prava svakog pojedinog korisnika.

5.1. ALTERNATIVNI SIGURNOSNI MEHANIZMI

Korištenje pogleda za uspostavu sigurnosti je jedan od alternativnih sigurnosnih mehanizma. Ako pretpostavimo da u bazi imamo tablicu Zaposlenici, koja sadrži podatke o zaposlenicima neke tvrtke, a u njoj se nalaze kolone Ime i Prezime, Adresa, Broj Telefona, Šifra Odjela i Plaća, te ako određenim korisnicima želimo dopustiti da vide sve podatke o zaposlenicima, ali da ne vide iznos njihove plaće, tada možemo koristiti pogled. Pogled može biti kreiran tako da se temelji na tablici Zaposlenici, ali da ne sadrži kolonu s osjetljivim podatkom o plaći. Korisnicima koji smiju vidjeti sve podatke osim plaće nećemo dati nikakvo pravo na tablicu, ali ćemo im dati SELECT pravo na pogled. Na taj način radimo restrikciju podataka za te korisnike u kojoj je određena kolona iz tablice nedostupna za pregled. Takva vrsta restrikcije se naziva ***vertikalna restrikcija***. Možemo imati i situaiciju da određeni korisnik smije vidjeti sve podatke o zaposlenicima, ali samo o onima koji rade u istom odjelu kao i on sam.
Niti ovdje neće biti dozvoljeno korisnicima da pristupe tablici, nego samo pogledu. Tu vidimo sve kolone, ali ne vidimo sve zapise, nego samo one iz odjela sa šifrom 3. Restrikcija koja je napravljena na taj način naziva se ***horizontalna restrikcija***. Na kraju pogleda dodana je opcija WITH CHECK OPTION. Ona onemogućava pokušaj promjene podataka preko pogleda ako nove vrijednosti ne zadovoljavaju postavljeni WHERE uvjet. U ovom slučaju je moguće da se preko pogleda upišu samo zaposlenici iz odjela sa šifrom 3, što predstavlja dodatni sigurnosni mehanizam.

Da bi neki korisnik mogao izvesti pohranjenu proceduru, mora mu se eksplicitno dodijeliti EXECUTE pravo za nju, čak kad bi on imao sva potrebna prava na sve objekte koji se unutar procedure koriste. Vrijedi i obrnuto: korisnik može imati zabranu korištenja objekata iz procedure, ali ako je dobio EXECUTE pravo na proceduru, moći će je uredno izvesti. (Korbar, 2010: 82)
Ako imamo potrebu korisnicima dopustiti modifikaciju podataka u nekoj tablici, najbolje bi bilo napisati procedure koje će raditi te modifikacije. Korisnicima ne bi bilo dopuštena

izravna promjena na tablicama, nego im se može dati prava da izvode procedure. U nekim situacijama sigurnost se treba implementirati prema nekom algoritmu, na primjer, korisnici koji rade na određenoj lokaciji prijepodne ne smiju imati pristup nekim podacima, a poslijepodne im taj pristup treba omogućiti. Tako bi se mogla kreirati pohranjena procedura koja bi implementirala tu poslovnu logiku i davala, odnosno ukidala prava korisnicima s obzirom na trenutačno vrijeme.

5.2.VANJSKA SIGURNOST

Ako se nekim SUBP resursima može pristupiti „izvana“, tj.bez korištenja SUBP-a ili SQL naredbi, onda sam SUBP nema sigurnosnog mehanizma da kontrolira takav pristup. To se odnosi na datoteke koje SUBP koristi i kojima se može pristupiti izravno iz operativnog sustava, a to uključuje:

* *Podtatktovne datoteke*
* *Log datoteke*
* *Sigurnosne kopije podatktovnih i log datoteka*
* *Zapisnike s rezultatima auditinga*(ugrađene mogućnost praćenja događaja koje svaki SUBP ima)
* *Zapisnike s rezultatima nadgledanja performansi*
* *Skripte i izvršne datoteke*

Ako ove datoteke nisu zaštićene, osobe koje posjeduju visoko stručno znanje, a imaju zle namjere, mogle bi doći u njihov posjed, pronaći način da ih pročitaju i na taj način neovlašteno pristupe podacima koji su u njima sadržani. Da bi se datoteke zaštitile od takve vrste neovlašteno pristupa, treba koristiti sigurnosne mehanizme za pristup datotekama, koji su ugrađeni u operativni sustav. Za to je potrebno imati visoka prava u operativnom sustavu, te treba odlučiti da li će to biti posao administratora baza, koji mora dobiti prava sistem administratora ili će sistemski administrator preuzeti taj posao na sebe. U oba slučaja sistemski administrator i administrator baza moraju usko surađivati da bi mogli efikasno zaštititi SUBP resurse od neovlaštenog vanjskog pristupa.

6. NAČINI OSIGURANJA DOSTUPNOSTI

Tehnike kojima se može povećati dostupnost baza podataka uključuju:
* *Korištenje alata* koji omogućuju da baza na vrijeme održavanja ostane dostupna
* *Korištenje alata* koji ubrzavaju izvođenje aktivnosti kod kojih baza mora biti off-line
* *Korištenje značajki SUBP* – a koje mogu povećati dostupnost

Mnoge aktivnosti koje je potrebno obavljati u sklopu održavanja baza mogu negativno utjecati na njihovu dostupnost. Stoga su potrebni alati koji omogućuju da te aktivnosti izvode dok je baza on-line. Neke takve alate proizvođači uključuju unutar samog SUBP – a, postoje i

alati nezavisnih proizvođača. Alati koji dolaze s SUBP – om ne moraju se posebno plaćati jer su uključeni u cijenu SUBP – a. Alati nezavisnih proizvođača mogu biti skupi, ali obično imaju bolje performanse od onih ugrađenih u SUBP – e.

Na tržištu postoje alati koji omogućuju on-line obavljanje ovih aktivnosti:
* *Reorganizacija baza*
* *Kreiranja sigurnosnih kopija*
* *Oporavka baza*
* *Provjere integriteta podataka* (Korbar, 2010: 120)
On-line obavljanje takvih aktivnosti može trajati znatno duže i ukupne performanse mogu zbog toga pasti.

Neki procesi su takvi da dijelove baze moraju učiniti nedostupnima da bi se mogli izvesti. Da bismo povećali dostupnost u takvim slučajevima trebamo te procese što prije ubrzati, da što prije završe. Promjena strukture tablica u produkcijskoj bazi obično je zahtjevan zadatak. Ako želimo promjeniti uzlaznu sortiranost klasteriranog indeksa u silaznu, morat ćemo kreirati novu tablicu sa željenim indeksom, prekopirati podatke i obrisati staru tablicu. Dok ta promjena traje, podaci će biti nedostupni, a trajanje promjene će ovisiti o količini podataka u tablici, vezama prema drugim tablicama, ostalim indekcima i drugim objektima povezanima sa starom tablicom.
I druge promjene nad tablicama zahtijevaju kreiranje nove tablice s drugačijim svojstvima, a potom brisanje stare. Npr. dodavanje nove kolone u tablicu, ali ne na kraj, već na neko mjesto „ispred“. I brisanje kolone iz tablice zahtijeva brisanje cijele tablice i stvaranje nove. Trebat će ponovno kreirati strane ključeve na novoj tablici te možda ponovno definirati i prava pristupa novoj tablici.
Da bi napravili neku takvu promjenu, administratori moraju najprije napraviti analizu i zaključiti koji će se objekti kaskadno obrisati nakon brisanja tablice. Nakon toga trebaju napisati skripte da bi sve te objekte mogli ponovno kreirati. (Korbar, 2010: 121).To može biti dugotrajan posao, može se lako pogriješiti ili napraviti neki propust.
Zbog toga je dobro imati alat kojemu se samo zada kakva se promjena nad tablicom treba napraviti, a on napravi potrebnu analizu, kreira sve potrebne skripte i izvede ih. Upotreba takvog alata ima velike prednosti – administratorima štedi vrijeme, skripte će se vjerojatno brže izvesti nego da ih administratori izvode jednu po jednu, manja je mogućnost pogreške, a alat može imati i mogućnost da izvede povratak na staro stanje. Alati takvog tipa spadaju u alate za upravljanje promjenama nad bazom, a mogu pridonijeti u skraćivanju vremena nedostupnosti za vrijeme promjena strukture baze.
Ako se baza sruši i treba napraviti oporavak iz sigurnosnih kopija, važno je da strategija oporavka dobro implementira. Administratori tada analiziraju stanje, zaključuju koje objekte baze treba oporaviti, koje su sigurnosne kopije potrebne, te moraju izraditi skripte za oporavak baze. U brzom oporavku mogu pomoći alati, tako što navedene aktivnosti od analize stanja do izrade skripti za oporavak , mogu obaviti sami. To značajno ubrzava oporavak baze i smanjuje njezino vrijeme nedostupnosti.
U transferima podataka obično se zahtijeva da se kod unloada podaci u izvorišnoj tablici ne smiju mijenjati da bi se u odredište mogli prebaciti konzistentni podaci. Ako podataka ima

mnogo, unload može dugo trajati i za to vrijeme korisnicima podaci neće biti u potpunosti dostupni. U tu svrhu postoje alati koji ubrzavaju transfere podataka pa se pomoću njih može skratiti vrijeme nedostupnosti podataka. (Korbar, 2010: 121)

6.1. KORIŠTENJE ZNAČAJKI SUSTAVA UPRAVLJANJA BAZA PODATAKA ZA VISOKU DOSTUPNOST

Budući da potrebe za povećanom dostupnošću rastu, proizvođači SUBP-a tome se nastoje prilagoditi, pa obično u svakoj novoj verziji dodaju neke nove funkcionalnosti vezane uz visoku dostupnost.
Osnovna načela na kojima se te funkcionalnosti zasnivaju su:

- KLASTERI

Većina SUBP – a ima podršku za rad u klasteriranom okruženju. Klasteri su grupe od dva ili više povezana poslužitelja koji djeluju kao jedan. Postoje različiti načini kako da se oni implementiraju. Možemo imati klaster od samo dva poslužitelja spojena na isti set mrežnih diskova ili pak desetke poslužitelja koji mogu podijeliti radno opterećenje cijelog sustava i time povećati performanse. (Korbar, 2010: 121)
Klasteri mogu osigurati visoku dostupnost tako da u slučaju otkazivanja jednog poslužitelja preostali poslužitelji preuzmu njegovu ulogu. Baze koje je nadgledao pokvareni poslužitelj, nalaze se na vanjskim diskovima dostupnima ostalima, te oni mogu preuzeti konrolu nad njegovim bazama. Postupak prebacivanja svih potrebnih funkcionalnosti na druge poslužitelje događa se automatski. Oni tada zbog povećanog opterećenja mogu imati slabije performanse, ali baze cijelo vrijeme ostaju dostupne.
Klasteri na ovaj način mogu izvrsno poslužiti u nadogradnji poslužitelja, poput instaliranja zakrpa za operativni sustav ili SUBP, odnosno kod hardverske nadogradnje kao što je dodavanje memorije. Zbog takvih postupaka poslužitelj određeno vrijeme ne može biti funkcionalan. U klasteriranom okruženju ulogu tog poslužitelja možemo prebaciti na ostale poslužitelje i odraditi nadogradnju bez ugrožavanja dostupnosti.

- PRIČUVNI SUSTAVI

Visoka dostupnost baza često se nastoji povećati uspostavljanjem pričuvnih, stand-by sustava. podaci se iz aktivnog sustava kopiraju na pričuvni sustav, koji se obično nalazi na drugoj lokaciji. u slučaju otkazivanja aktivnog sustava, njegovu ulogu preuzima pričuvni. (Korbar, 2010: 121)
Postoji metoda za stvaranje pričuvnih sustava – zrcaljenje, kod koje se svaka promjena na bazi automatski primjenjuje i na njezinoj zrcalnoj kopiji. U nekim implementacijama se prebacivanje na pričuvni sustav u slučaju pada aktivnog sustava može izvesti automatski, dok se kod drugih to treba obaviti ručno. Postoje različiti stupnjevi sinkroniziranosti između aktivne i pričuvne baze. Promjena podataka se ponekad na aktivnom sustavu može potvrditi

tek nakon što pričuvni sustav pošalje obavijest da je i on napravio tu promjenu, a u drugima se pak ta obavijest ne čeka.
Još jedna metoda uspostave pričuvnog sustava je log-shipping. Za razliku od zrcaljenja, gdje se svaka promjena automatski prosljeđuje na pričuvni sustav, kod log-shippinga se propagiranje promjena događa u zadanim vremenskim intervalima. Iz aktivne baze šalju se nove promjene zabilježene u njezinom transakcijskom logu, a na pričuvnoj se bazi radi restauriranje tog trnsakcijakog loga. Log-shipping je jeftiniji za implementiranje nego zrcaljenje, ali ne daje mogućnost automatskog prebacivanja na pričuvni sustav. Svi klijenti se kod prelaska na pričuvni sustav moraju ručno preusmjeriti na novu adresu.

- OPORAVAK OD POGREŠNIH TRANSAKCIJA

U slučajevima da je nedostupnost baza uzrokovana ljudskim greškama, od velike pomoći mogu biti značajke koje omogućuju pregled tranakcijskog loga i poništavanje krivih transakcija. Oracle je sa svojom flashback tehnologijom donio mnogo dobrih značajki, poput pregleda različitih verzija podataka kroz vrijeme, olakšanog poništavanja transakcija te jednostavnijeg point-in-time oporavka.

- ON-LINE PROMJENA SISTEMSKIH PARAMETRA

U mnogim SUBP – ima postoje sistemski parametri čija promjena zahtijeva ponovno pokretanje SUBP – a. Kako bi povećali dostupnost baza, mnogi proizvođači SUBP – a sve više rade na tome da omoguće on-line promjenu sistemskih parametara. (Korbar, 2010: 122)

7. ZAŠTITA BAZE PODATAKA OD NEOVLAŠTENOG I ISTOVREMENOG PRISTUPA

Baze podataka moraju se zaštititi od neovlaštenog pristupa. Osim fizičke zaštite postoji i softverski način zaštite kod kojeg se određeni softver ugrađuje u sustav za upravljanje bazom podataka. Njime se ograničava rad s bazom podataka, tj. rad korisnika koji imaju fizički pristup računalu ili računalnim terminalima. Načini zaštite baze podataka mogu biti identifikacijom korisnika, raznim mehanizmima zaštite, te ovlaštenjima. Kod zaštite baze podataka identifikacijom korisnika, poznato je da svaki korisnik baze ima svoje korisničko ime (username) i samo njemu poznatu korisničku lozinku (pasword). Korisnik se mora predstaviti svojim korisničkim imenom i lozinkom u sustav za upravljanje bazom podataka, čime dokazuje svoj identitet. Velikom brojem sustava za upravljanje bazom podataka možemo upravljati preko interneta. Bazi podataka možemo pristupiti i s udaljenog računala koje može biti locirano bilo gdje u svijetu. Ova prednost je ujedno i nedostatak, jer otvara vrata zlonamjernim osobama da neovlašteno pristupaju određenim podacima. (Varga i suradnici, 2012: 83)

Bazu podataka kojoj se može pristupiti preko interneta možemo privremeno zaštititi isključivanjem mrežnih mogućnosti sustava za upravljanje bazom podataka. Tako će samo lokalni korisnici (programeri) moći pristupiti bazi, tj. klijenti lokalne mreže. Može se dopustiti pristup bazi podataka samo nekim vanjskim korisnicima, uz identifikaciju klijenata korištenjem šifrirane komunikacije, npr. ssl/ssh, dvostruki ključevi itd. Što se tiče pogleda (views), korisniku se može dati pravo pristupanja samo određenom dijelu podataka. Drugi dijelovi baze podataka su za određenog korisnika nedostupni.

Zaštita baze davanjem ovlaštenja određuje što korisnik može raditi s podacima iz baze koji su mu na raspolaganju. Ovlaštenja za zaštitu mogu biti read/select, update, insert i delete. Sustav za upravljanje bazom podataka mora „pamtiti“, tj. imati pohranjen popis ovlaštenja za svakog korisnika i svaku relaciju iz točno određenog pogleda. Ako korisnik pokuša obaviti aktivnost za koju nije ovlašten, sustav za upravljanje bazom podataka neće je izvršiti, nego će ispisati upozorenje da korisnik nije ovašten za obavljanje navedene aktivnosti. Za zaštitu baze podataka se u većini slučajeva brine programer, odnosno administrator baze podataka. On ima popis korisika, te daje mogućnost pogleda i regulira ovlaštenja. Treba ispravno nadzirati bazu podataka, analizirati promjene nad podacima po razdobljima te snimati postupke osobe koja želi nanijati štetu.
Funkcionalnost sustava za upravljanje bazom podataka može biti pokazatelj njegove sigurnosti. Backupom ili sigurnosnom kopijom mogu se zaštititi podaci iz baze. U današnje vrijeme se mogu koristiti sustavi koji automatski stvaraju sigurnosnu kopiju i oporavak sustava, te omogućavaju arhiviranje podataka na više od 5 godina i uklanjaju troškove koji su izazvani ljudskom greškom povezanom s pohranom podataka. Ovakav način zaštite bolji je od stvaranja sigurnosne kopije cijele baze podataka, jer sustav vreća samo one podatke koji su bili promijenjeni.

Sustav za upravljanje bazom podataka mora osigurati i garantirati seribijalnost, a to se postiže posebnom kontrolom kod istodobnog obavljanja transakcija. Ako se u višekorisničkoj bazi podataka izvodi nekoliko transakcija paralelno tako da se pojedini dijelovi tih transakcija izvode vremenski izmješano, i ako je konačni učinak njihovog izvođenja isti kao da su one izvršene serijski ili sekvencijalno, to znači da se radi o seribijalnom ili serijalizabilnom izvršavanju transakcija.
Postoji više načina kontrole, a to su:

- Lokoti i dvofazni protokol zaključavanja
- Vremenski žigovi

Lokotima se sprječava istovremeni pristup podacima koji zaključavaju pojedine dijelove baze podataka, ok jedna određena transakcija ne završi posao.(Varga i suradnici, 2012: 84)
Veličinu dijela baze podataka koja je zaključana određuje zrnatost. Što je veći dio baze podataka zaključan, to je manji stupanj paralelnosti, a kontrola je jednostavnija. Baze podataka su podijeljene na više dijelova tako da svakom dijelu odgovara jedan lokot. Transakcija koja želi pristupiti nekom podatku iz baze najprije mora uzeti odgovarajući lokot namijenjen određenom dijelu baze. Uzimanjem lokota zaključava se točno određeni dio baze podataka. Nakon što transakcija obavi operaciju, treba se vratiti preuzeti lokot i time otključati

podatke iz dijela baze. Kad transakcija naiđe na podatke koji su već zaključani, ona mora čekati dok ih prethodna transakcija ne otključa. Time se izbjegava istovremeni pristup istom podatku. Upotreba lokota krije i određene opasnosti, a najveća od njih je moguća međusobna blokada dviju ili više transakcija. Softver koji koristi lokote mora računati na mogućnost blokade transakcija, te mora osigurati da se ta blokada spriječi ili prekine. Rješenje koje se često koristi je da se povremeno kontrolira ima li blokiranih transakcija, odnosno traži se ciklus u usjerenom grafu koji prikazuje koja transakcija čeka koju. Ako takve blokirane transakcije postoje, tada se jedna od njih prekida, neutralizira se njen dosadašnji učinak te se ona ponovno starta u nekom kasnijem trenutku.
Ako u svakoj od transakcija sva zaključavanja slijede prije prvog otključavanja, tada proizvoljno istovremeno izvršavanje tih transakcija mora biti serijalizabilno. Takvo pravilo nazivamo dvofazni protokol zaključavanja.

7.1. KRIPTOGRAFSKA ZAŠTITA BAZE PODATAKA

Kriptografija je znanost tajnog pisanja, tj. znanost pohrane informacija u onoj formi koja će biti čitljiva samo onima kojima je informacija namijenjena dok će za ostale biti neupotrebljiva.

Ako želimo da korisnici baze podataka međusobno komuniciraju, razmjenjuju ideje i šalju pozive za razna druženja, ako se svim korisnicima omogući pristup bazi podataka preko sustava, postoji opasnost da neki podaci dođu u ruke društvu s lošim namjerama. Kriptografija može problem riješiti kada je riječ o sigurnosti baze podataka. Baza podataka može se šifrirati tako da će biti jednostavno doći do adrese jedne osobe, a teško do liste svih članova. (Varga i suradnici, 2012: 84)
Potebno je izabrati jednosmjernu *hash funkciju* [2]i *simetrični algoritam za šifriranje*[3]. Pomoću hash funkcija mogu se dobiti sažeci poruka. Za takve funkcije se kaže da su jednosmjerne jer je jednostavno izračunati sažetak poruke, ali je teško rekonstruirati poruku na temelju sažetka. Kod simetričnog algoritma za šifriranje isti se ključ koristi za šifriranje i dešifriranje. Svaki zapis iz baze podataka ima dva polja. Indeksirano polje je prezime člana koji je podvrgnut jednosmjernoj hash funkciji. Polje podataka je puno ime i adresa člana, šifrirano korištenje prezimena kao ključa. Polje podataka se ne može šifrirati ako se ne zna prezime. Traženje određenog prezimena je jednostavno.
Prvo se napravi hash prezimena i traže se hash vrijednosti u indeksiranom polju baze podataka. Ako se pronađe traženi podatak, tada je to prezime u bazi. Ako pronađete nekoliko traženih vrijednosti, tada u bazi postoji više podataka o više osoba s istim prezimenom. Za svaki odgovarajući zapis dešifriramo puno ime i adresu koristeći prezime kao ključ. Osnovni problem u ovom sustavu je traženje osoba za koje ne znamo kako napisati njihovo prezime. Mogu se isprobati razne varijante dok se ne pronađe prava, ali nije pogodno pregledati sva

[2] Hash funkcija se koristi za transformiranje ključa u indeks (hash) to jest mjesto u nizu elemenata gde treba tražiti odgovarajuću vrijednost.

[3] Postupak kojime se podaci pomoću ključa promjene te više ne mogu čitati (osim ako imate ključ).

imena čija prezimena počinu s npr. Sch. Samo vrlo uporna osoba može rekonstruirati bazu podataka o članovima organizacije tako da će primijeniti napad silom, isprobavajući svako novo prezime. Ako on ima telefonski imenik u bazi podataka, može koristiti listu svih mogućih prezimena. (Varga i suradnici, 2012: 88).

7.2. OŠTEĆENJE I OBNOVA BAZE ELEKTRONIČKIH PODATAKA U FINANCIJSKOM PODSUSTAVU

Kod financijskog podsustava prisutni su brojni procesi, a najčešći su: nabava, prodaja, računovodstvo, putni nalozi, blagajničko poslovanje. Svi ti procesi imaju svoje potprocese i aktivnosti na nižoj razini. Organizacijska struktura određene tvrtke prilagođena je spomenutim funkcijskim područjima, kao i poslovima na temelju kojih ostvaruje dobit.

Oštećenje baze podataka može nastati zbog kvara hardverskog dijela računala (disk i drugi mediji za pohranu) ili zbog pogreške u sustavnoj programskoj podrški, također može biti oštećena i radom zlonamjernih osoba (tzv. hakera), slučajno ili nesretno.

Bez obzira na razloge oštećenja, baza podataka se mora vratiti u stanje očuvanog fizičkog integriteta. Tu podrazumijevamo ispravnost informacija, tj. podataka sadržanih u bazi. Problemi s integritetom baze podataka obuhvaćaju sve mjere zaštite kojima je cilj sprječavanje unosa neispravnih podataka u bazu elektroničkih podataka. Za rezultate neispravnosti u bazi krive su pogreške koje se dogode prilikom unosa ili ažuriranja, a programske i sklopovske pogreške rezultat su namjernog unosa krivih podataka da se ošteti baza podataka. Baza podataka se štiti ograničenjima. (Varga i suradnici, 2012: 89). Pravila integriteta predstavljaju ograničenja sadržaja baze podataka na dopuštena stanja koja osiguravaju međusobnu usklađenost podataka baze podataka prilikom unošenja, ažuriranja i brisanja podataka. Financijske organizacije rade s podacima koji su im dostupni i u skladu s njima donose vrlo važne odluke. Posljedice su velike ako podaci s kojima se radi u financijskom odjelu nisu ispravni ili ih je promijenila zlonamjerna osoba ili napadač. Npr. ako bi neka škola izgubila sve podatke u računalnom sustavu o zaposlenima, njihovim plaćama, radnom stažu, radnik u računovodstvu bi u sustav ponovno morao unijeti sve podatke o zaposlenima, a to bi bio preveliki posao.

Da bi se baza podataka mogla obnoviti, treba prethodno sigurnosno pohranjivanje podataka iz baze na neki medij koji se nalazi izvan baze podataka. Zbog toga se periodički uzima sigurnosna kopija čitave baze podataka na poseban medij, a sve promjene podataka u bazi podataka evidentiraju se u dnevniku izmjena. Sigurnosno pohranjivanje podataka svakih pet dana je možda prerijetko. Postavlja se pitanje što bi se dogodilo ako bi određena tvrtka ostala pet dana bez važnih podataka u bazi. Stoga je sigurnije svaki dan stvarati sigurnosnu kopiju podataka. Preporuka tvrtkama bi bila da zaposle vlastitog informatičara koji bi brinuo o sigurnosnim kopijama te bi bio odgovoran za podatke u bazi. Sustav za upravljanje bazom podataka u određenoj tvrtki mora biti dostupan svaku minutu, čime se omogućava izrada sigurnosnih kopija 24 sata. (Varga i suradnici, 2012: 90). Sigurnosne kopije se mogu stvarati

tijekom rada. U određenim tvrtkama nikad nije došlo do zlonamjernog upada u informacijski sustav financija, a razlog je taj što se lozinke često mijenjaju, te za svaki dio aplikacije postoji posebna lozinka. Samo određeni zaposlenici imaju pristup određenim programskim modulima. U takvim tvrtkama lozinke su postavljene s kombinacijom velikih i malih slova i brojeva. Radi zaštite svih elektroničkih podataka u bazi, tvrtke na osobnim računalima i serverima u financijskim odjelima koriste antivirusnu zaštitu. Nadogradnja antivirusnog softvera je ažurna. Svaki put prilikom korištenja računala korisnik mora napraviti update antivirusnog programa.

7.3. SQL POSLUŽITELJ I ZAŠTITA PODATAKA

SQL je upitni jezik za rad s relacijskom bazom podataka. Relacijska baza podataka pruža najveću fleksibilnost u dekompoziciji podataka, što osigurava veliku prednost u planiranju distribucije podataka po pojedinim čvorovima sustava.(Varga i suradnici, 2012: 90) Relacije se mogu dijeliti vertikalno i horizontalno, te mogu biti ponovno spojene operacijama spajanja. SQL jezik razvio je IBM. Standardni strukturni upitni jezik je najpopularniji, najpoznatiji i najčešće korišten jezik. Služi za manipulaciju i upravljanje podacima koji se nalaze u bazi.
SQL poslužitelj je proizvod i blisko je povezan s drugim poslužiteljima i slojevima u tipičnom IT okruženju koje je zasnovano na temeljima tvrtki o čijoj se bazi radi. Poslužitelj se ne koristi samo za pohranjivanje podataka o korisnicima, on je veoma važan jer sam rad pomoću njega postaje mnogo ekonomičniji i jednostavniji. Jedan od načina zaštite u SQL okruženju je osiguranje dodatnog sloja sigurnosti za osjetljive podatke. Drugi način je pridržavanje pravila, standarda i zakona vezanih uz zaštitu podataka. Treći način se odnosi na zaštitu objekata poslužitelja na kojima se nalaze povjerljive informacije o korisnicima i lozinke za pristupanje povezanim poslužiteljima. Četvrti način je osiguravanje kontroliranog načina za definiranje privilegija za one module koji sadrže kod.

Kada je riječ o procesima za upravljanje integritetom podataka, oni bi trebali osigurati preciznost, ispravnost, potpunost ili cjelovitost podataka koji su smješteni u bazi podataka. Integritet podataka na SQL poslužitelju postiže se pomoću određenih ograničenja i trigera. Ograničenja su pravila koja se primjenjuju na relacije, njima se definira koje vrijednosti je dopušteno unijeti u odgovarajuću relaciju i u kojem rasponu. U mnogim aplikacijama su definirana pravila za očuvanje integriteta podataka. Nije dovoljno samo definirati pravila za očuvanje integriteta podataka u aplikaciji, nego treba definirati odgovarajuća ograničenja na razini baze podataka tako da se ne unose neispravni podatci.

Vrste integriteta podataka su entitetski, domenski i referencijalni.
Entitetski integritet osigurava da svaki tip u relaciji može biti jedinstveno identificiran pomoću stupca. On se osigurava u SQL-u korištenjem indeksa koji služi za provjeru postojanja dupliciranih vrijednosti. Primarni ključ i ograničenja koja trebaju radi jedinstvenosti primjenjuju se prilikom osiguravanja entitetskog integriteta, i osigurava da se ne dogodi dupli zapis u relaciji.

Domenski integritet se odnosi na utvrđivanje dopuštenih vrijednosti za svaki stupac tablice. Tip podataka koji se primjenjuje na stupac u relaciji predstavlja jedan od načina za osiguranje domenskog integriteta. Kod domenskog integriteta CHECK ograničenje omogućava da se ograniče vrijednosti dopuštene u koloni na temelju logičkog izraza. Ograničenjem pomoću vanjskog ključa definiraju se vrijednosti dopuštene u koloni, ali na temelju sadržaja primarnog ključa.

Referencijalni integritet osigurava i realizira strani ili vanjski ključ te CHECK. CHECK ograničenja mogu referencirati veći broj kolona u istoj relaciji korištenjem odgovarajućeg logičkog izraza, on može biti dio create table naredbe ili se može dodati u neku od već postojećih tablica. Nakon što se doda CHECK ograničenje u postojeću tablicu s podacima, provjerava se u odnosu na uvjet definiran CHECK ograničenjem. Ako u određeom stupcu imamo mnogo podataka, dodavanje CHECK ograničenja za tablicu može potrajati neko vrijeme.
U SQL-u postoji način preskakanja provjere podataka prilikom kreiranja novog CHECK ograničenja u postojeću tablicu. Preskakanje se postiže WITH NOCHECK opcijom. Ograničenje s vanjskim ključem referencira primarni ključ, obično u drugoj relaciji kako bi se očuvala relacija (slabog entiteta). Referencijalni integritet se može osigurati i proceduralno korištenjem trigera.
Ograničen vanjski ključ podrazumijeva pravila za svaki zapis u tablici Kupac mora postojati odgovarajući zapis u tablici; za svaki zapis u tablicama Prodavatelj i Oznaka dokumenta mora postojati odgovarajući zapis u tablici Ostali Dio Dokumenta. Unos zapisa o kupcu s neodgovarajućim tipom identifikatora može narušiti ograničenje vezano za primarni ključ, a rezultat toga je obično greška. Zapisi u tablici Ostali Dio Dokumenta, mogu biti uklonjeni ako ih referenciraju zapisi u tablicama Kupac, Prodavatelj i Oznaka Dokumenta. Primarni ključ zapisa za ostali dio dokumenta ne može se ažurirati ako postoje zapisi vezani uz kupca, prodavatelja i oznaku dokumenta koji ga referenciraju. (Varga i suradnici, 2012: 92)

Ograničenje s vanjskim ključem može omogućiti opercije ažuriranja i uklanjanja kaskadno u međusobno povezanim relacijama. Operaciju kaskadnog uklanjanja zapisa treba pažljivo koristiti. Uklanjanje jednog zapisa može dovesti do uklanjanja velikog broja ovisnih blokova na osnovu niza tablica i relacija koje su međusobno povezane stranim ili vanjskim ključevima. U SQL-u postoje četiri mogućnosti podešavanja koja su vezana za izvršavanje kaskadnog ažuriranja i brisanja vanjskog ključa: SET NULL podešavanje, SET DEFAULT podešavanje, NO ACTION podešavanje i CASCADE podešavanje. DML trigeri su također sredstva za očuvanje integriteta podataka. Trigeri dopuštaju definiranje mnogo složenijih pravila za očuvanje integriteta podataka u odnosu na metode deklaracijskog definiranja integriteta te zahtijevaju dodatna izračunavanja. Trigeri se primjenjuju kod sprječavanja neispravnih naredbi kao što su: insert, update, i delete, kod prikazivanja korisničkih poruka o pojavama greški te prilikom kaskadnih izmjena u poveznim tablicama u bazi podataka koje se ne mogu ostvariti pomoću vanjskog ključa.

7.3.1. SQL INJECTION

Medu najosjetljivijim točkama u pogledu očuvanja sigurnosti baza podataka je provjera podataka koje korisnik šalje bazi. Ako je posjetiteljima neke Web stranice dozvoljen unos podataka u bazu, potrebno je provjeriti da li podaci koje je unesao korisnik sadrže neke SQL naredbe. Na primjer, nakon posljednjeg unesenog podatka, korisnik može unijeti zapovijed DELETE FROM IME_TABLE;COMMIT. Ako ne postoji provjera unosa, ta naredba će obrisati neku tabelu iz baze. Napad SQL injection direktna je posljedica loše projektirane aplikacije koja stvara dinamičke SQL upite na osnovu interakcije s korisnikom.
Dinamički SQL formira se prilikom izvršenja programa (na primjer, na osnovu podataka koje korisnik unosi u obrazac). Primjer takvog upita je izdvajanje iz skladišta robe čiji je proizvođač X, gdje je X neki parametar koji korisnik unosi u obrazac.
To omogućava napadaču da bazi podataka proslijedi SQL upit po svojoj volji. Ukoliko to zanemarimo i ostavimo prostor za mogući napad, svako daljnje osiguravanje SUBP – a postaje beskorisno.

Iako su SQL injection napadi po svojoj prirodi jednostavni (napadač aplikaciji prosljeđuje unos koji sadrži SQL upit), poželjno je da napadač koji želi da izmjeni podatke u bazi poznaje strukturu baze (koje tabele postoje, od kojih se kolona koja tabela sastoji itd.). izvođenje napada na bazu čija je struktura nepoznata, znatno je kompliciranije.

SQL Injection napadi mogu se podjeliti u četiri kategorije:

* *Modifikacija SQL upita*. Napadač modificira SQL upit pomoću operacija nad skupovima (najčešće UNION) ili mijenja odredbu WHERE da bi se dobio drugačiji rezultat. Najpoznatiji napad ove vrste je modifikacija odredbe WHERE upita za provjeru identiteta korisnika tako da odredba uvijek daje rezultat TRUE.
* *Umetanje koda*. Napadač unosi novi SQL upit ili novu zapovijed u postojeći SQL upit. Ova vrsta napada funkcionira isključivo u SUBP – ovima koji podržavaju veći broj SQL upita po jednom zahtjevu bazi podataka (na primjer, naredba EXECUTE u MS SQL Serveru). Ovakav napad u Oracle SUBP teško se ostvaruje.
* *Umetanje funkcijskih poziva*. Napadač umeće Oracleove ugrađene funcije ili neke korisničke funkcije u ranjiv SQL upit. Ovi funkcijski pozivi se zatim mogu iskoristiti za izvršavanje funkcijskih poziva operativnog sustava ili za izmjenu podataka u bazi.
* *Napad prekoračenjem bafera* zasniva se na prethodno uspješnom izvedenom napadu umetanja poziva funkcije. Skoro svaki SUBP sadrži neku uskladištenu proceduru ili ugrađenu funkciju čijom se zlouporabom može izazvati prekoračenje bafera. Prepisivanje podataka u baferu može onogućiti napadaču da pokrene proizvoljan programski kod u memorijskom prostoru rezerviranom za proces koji je pokrenuo SUBP. Takav kod može biti iskorišten u različite svrhe, na primjer, jednostavno se može isključiti SUBP server, ili se može napraviti novi proces radi preuzimanja kontrole nad serverom.

7.4. ZAŠTITA ELEKTRONIČKIH PODATAKA U SUSTAVU BANKE

Kada su posrijedi napadi hakera na bankarski sustav, oni nisu toliko česti u Republici Hrvatskoj kao u drugim zemljama. Banke su danas korak ispred potencijalnih opasnosti, čime je siguran novac građana i tvrtka. Banke su usvojile novi pristup IT segmentu. Implementacijom tzv. „early warning" signala možemo unaprijed predvidjeti problematična područja i na vrijeme reagirati. Čestim revizijama i testiranjima probojnosti sustava osigurava se kontinuirano podizanje razine sigurnosti te provođenje mjera za smanjenje IT rizika. Na razini bankarskog sustava postoje autoriteti koji prikupljaju podatke o najčešćim napadima i incidentima te na adekvatan način informiraju sve sudionike i zahtijevaju unaprjeđenja sustava. To se zahtijeva kada je riječ o zaštiti podataka. Nastoji se povećati sigurnost sustava za upravljanje bazom podataka banke kako njoj ne bi naštetile zlonamjerne osobe. Da bi banke preventivno djelovale, one neprekidno kontroliraju sustav zaštite podataka preko internih i vanjskih revizora te neovisnih procjenjivača sigurnosti informacijskog sustava. Radi učinkovitijih sigurnosnih mjera u kartičnom poslovanju i povećanja sigurnosne razine transakcija, banke moraju uvesti PCI DSS (Payment Card Industry – Data Security Standard), koji je razvio konzorcij vodećih kartičnih kuća u svrhu učinkovitije zaštite važnih kartičnih podataka. Cilj mu je smanjiti broj prijevara i povećati sigurnosni standard u tvrtkama koje u svom poslu procesiraju ili pohranjuju podatke s kreditnih kartica, a to su najčešće banke. Ostvarenjem zahtijeva koje postavlja PCI DSS učinkovito se upravlja rizicima informacijskih sustava.(Varga i suradnici, 2012: 95)

Do danas nije zabilježen ni jedan slučaj probijanja informacijskog sustava banke i njegove vanjske zaštite. U većini slučajeva napadaju se klijenti. Kada se ošteti informacijski sustav banke, izvor napada je najčešće unutar samog sustava. DDoS napadom (Denial of Service) može se onemogućiti bilo kakva usluga bilo koje institucije u svijetu, uključujući i Pentagon. To nije nikakvo hakiranje i nije nikakva vještina probijanja u informacijski sustav. Hakeri zavladaju drugim poslužiteljima i onda s previše zahtjeva opterete poslužitelja kojeg žele napasti. Stvaraju gužvu zbog čega se ostali klijenti informatički ne mogu probiti do poslužitelja. Sustav se od toga ne može zaštititi jer nije došlo do proboja u taj sustav, nego u druge. Danas su u Republici Hrvatskoj i u zapadnom svijetu bankarski sustavi sigurni. Kada je riječ o bankarskim transakcijama, banke koriste zaštitu jednokratnim lozinkama. One su oblik jake autentifikacije korisnika koja pruža veću razinu zaštite transakcija preko interneta, mrežnim sustavima u tvrtkama i drugim sustavima koji sadrže osjetljive i povjerljive informacije.

Da bi što bolje i efikasnije spriječili upad u računalni sustav i prostore gdje je on smješten, nužno je osigurati fizičku zaštitu računala i prostorije gdje se računalni sustav nalazi. Ako su računala s važnim podacima u nekoj prostoriji, te prostorije treba dodatno zaštititi video – nadzorom, pametnim karticama, karakteristikama biometrije ili na neki drugi način.
Elektronički podaci se mogu zaštititi uz pomoć hardverske zaštite, softverske zaštite, fizičke, komunikacijske, revizijsko – administrativne i organizacijske zaštite. Računalni kriminal se

nikad neće do kraja suzbiti, ali se svakakao preporuča preventivno djelovati. Treba se paziti kome se šalju i gdje se objavljuju podatci. Za sprječavanje gubitka elektroničkih podataka preporuča se spremiti dokument na standardni način, tj. sa standardnom ekstenzijom datoteke, često snimati promjene, često izrađivati sigurnosnu kopiju, koristiti antivirusni program s licencom i sigurnosnu stijenku, izbjegavati stavljanje osjetljivih podataka na web, često brisati povijest pregledavanja, raditi enkripciju podataka.

8. ZAKLJUČAK

Baze podataka sa gledišta sigurnosti uvijek je aktualna i neiscrpna tema. Svaki veći posao, kao na primjer zdravstvena skrb ili državne ustanove oslanjaju se na baze podataka u kojima spremaju ključne podatke. Upravo zbog toga vrlo je važno baze podataka zaštititi i njima upravljati na pravi način. Kao glavne preporuke za čuvanje sigurnosti baze podataka su stalna nadogradnja programskih paketa, odvajanje baze na sigurne segmente mreže, korištenje enkripcije pri transferu i skladištenju osjetljivih podataka, korištenje autorizacije autentifikacije i uloga. Postoje različiti načini napada na baze podataka,koji su već dobro poznati, a zbog nedostataka u drugim sustavima mogu pogoditi i SUBP i bazu podataka kojom se upravlja. Takvi napadi su SQL umetanje. No, osim nedostataka drugih sustava, postoje ranjivost koje uključuju ljudske faktore kao što su dodjeljivanje akreditacije zlonamjernim korisnicima, nemarnost administratora pri nadgledanju baza podatka i sl.

Baze podataka ranjive su na vanjske i unutarnje prijetnje (loša konfiguracija SUBP – a, napadi unutar tvrtke, pogreške unutar tvrtke), ali moguće je smanjiti broj ranjivosti na prihvatljivu razinu rizika. To se postiže tako da se koriste napredni sigurnosni mehanizmi, u kojima se konstantno nadograđuju programski paketi uz operacijski sustav i SUBP, koriste sigurnosni resursi te sigurnosni proizvodi kao što su vatrozidi i antivirusni alati.

U svakom informacijskom sustavu potrebno je obratiti pažnu na realne sigurnosne rizike posebno u domeni baza podataka i to posebno u dijelu podataka koji se klasificiraju kao povjerljivi podaci. Potrebno je izraditi sigurnosno pouzdan sustav koji će voditi računa na razna ponašanja u sustavu koji mogu ukazivati na ozbiljne probleme usmjerene ka bazama podataka i ugrožavanju sigurnosti koja proistječu iz upotrebe takvih podataka. Zaštita podataka predstavlja skup metoda i tehnika kojima se ograničava pristup podacima od strane programa koji se izvršavaju, to podrazumijeva skup pravnih normi kojima se ograničava pristup podacima od strane programa i ljudi.

Putem zaštite štiti se fizički integritet cijelokupnog informacijskog sustava, bilo da je distrubuiran ili ne, odnosno centraliziran ili decentraliziran. Informacijska sigurnost je proces stalnog održavanja sigurnosti korisnika i informacijskih sustava.

9. POPIS SLIKA:

SADRŽAJ

LITERATURA:

1. Administriranje baza podataka priručnik; Damir Korbar, dipl.ing; Algebra d.o.o., Zagreb, 2010.

2. Cybernetica; Marina Čičin – Šain; Bilten Društva Kibernetičara - Baze podataka; Rijeka, 2007.

3. Sigurnost računarskih sistema i mreža – Dragan Pleskonjić, Nemanja Maček, Borislav Đorđević, Marko Carić; Mikro knjiga; Beograd, 2007.

4. Zaštita elektroničkih informacija; Matija Varga mag.inf.univ.spec.oec, Vladimir Šimović prof.dr.sc.v.š, dr.h.c., Marin Milković doc.dr.sc.M.M., prof.v.š.; Velučilište u Varaždinu, Varaždin, 2012.

5. http://e-student.fpz.hr/Predmeti/B/Baze_podataka/Materijali/Auditorne_vjezbe_2.pdf

6. http://www.unizd.hr/portals/1/primjena_rac/brodostrojarstvo/predavanje_4.pdf

7. http://www.cis.hr/files/dokumenti/CIS-DOC-2012-08-059.pdf