Univerza v Ljubljani
Fakulteta za računalništvo in informatiko

Podiplomski magistrski program Informacijski sistemi in odločanje
Predmet: Kriptografija in računalniška varnost
Predavatelj: dr. Aleksandar Jurišić

Varovanje računalniških sistemov s poudarkom na zaščiti podatkovnih baz

Avtor: Alenka Rožanec

Ključne besede: računalniška varnost, zaupnost, integriteta, dostopnost, zaščita, grožnja, ranljivost, goljufija, zloraba, napad, vdor, vsiljivec, zasebnost, varnostna politika, overovitev, avtorizacija, nadzor dostopa, lista pristopnih pravic, relacijske podatkovne baze, SQL, dokumentne podatkovne baze, transakcija, nadzor nad sočasno uporabo podatkovne baze

V Ljubljani, 4.12.2000

Kazalo

Uvod

Računalniška varnost je zelo širok pojem, ki obsega številna področja: od fizičnega varovanja opreme, varnosti samih operacijskih sistemov, zaščite podatkov v podatkovnih bazah, zaupnosti sporočil, varnosti v lokalnih omrežjih pa do vse bolj popularnega interneta. Predvsem zaradi vse večjega vključevanja računalnikov v internet postajajo mehanizmi zaščite še bolj pomembni, saj z vključitvijo v globalno računalniško omrežje lahko postanemo tarča najrazličnejših napadov, od nepooblaščenih vpogledov v zaupne informacije do okužbe z vse bolj grozečimi črvi, ki nam lahko popolnoma ohromijo celoten sistem.

Prav nevedni uporabniki so pogosto tarča napadov, saj preko odpiranja okuženih datotek v elektronski pošti, lahko povzročijo okužbo celotnega omrežnega sistema in ne samo lokalnega računalnika. Pri varovanju računalniškega sistema morajo zato sodelovati vsi njegovi uporabniki v skladu s svojimi vlogami v organizaciji. Potrebno je natančno definirati varnostno politiko, pri kateri so na podlagi zadolžitev in pooblastil, posameznikom dodeljene določene pravice do uporabe sistemskih resursov. Natančno je potrebno določiti tudi pravila njihove uporabe in varovanja ter odgovornost posameznikov za njihovo neizvajanje. Sistemski administratorji morajo poskrbeti za seznanjanje vodstva z vsemi nevarnostmi in pripravo varnostne politike. Prav tako morajo poskrbeti, da bodo politiko in svojo vlogo v njej razumeli prav vsi uporabniki ter v primeru nevarnosti čimprej obvestiti vse uporabnike. Varnostna politika se mora ves čas dopolnjevati in spreminjati v skladu z nenehno pretečimi nevarnostmi.

V prvem delu svoje seminarske naloge predstavljam pojme zaupnosti, integritete in dostopnosti računalniških sistemov, številne nevarnosti, ki jim pretijo ter načine njihovega varovanja. Številni primeri iz preteklosti zelo nazorno prikazujejo načine vdorov zunanjih napadalcev. Še vedno pa povzročijo največ škode sami zaposleni z napakami, goljufijami ali zlorabo pooblastil.

Zato je zelo pomembna določitev primerne varnostne politike temelječe na ciljih organizacije, zakonih in standardih s tega področja, upoštevajoč vplive računalniške tehnologije in vsakodnevne groženje iz okolja. Na koncu poglavja je predstavljen Matrični varnostni model na katerem temeljijo tako mehanizmi za kontrolo dostopa do datotečnega sistema večuporabniških operacijskih sistemov kot tudi zaščita podatkovnih baz. Omenjena poglavja sem povzela po knjigi Secure Computing [1].

Drugi del naloge obravnava zaščito podatkovnih baz. Ker podatkovna baza predstavlja osrednji element sodobnega poslovnega sistema, mora biti tudi ustrezno varovana, tako da lahko v vsakem trenutku vsem pooblaščenim uporabnikom nudi konsistentne, točne, relevantne in ažurne podatke. V tem delu se naslanjam na literaturo o podatkovnih bazah [2], konkretnih SUPB Oracle [4] in Notes/Domino [5] ter svoje izkušnje pri delu z omenjenima sistemoma.

V prvih dveh poglavjih so podane nekatere definicije teorije podatkovnih baz in opisana naloga sistema za upravljanje s podatkovno bazo. V tretjem poglavju so podani integritetni principi ter pripadajoči mehanizmi za njihovo zaščito. Četrto poglavje obravnava zaščitne mehanizme na primeru relacijske podatkovne baze Oracle. Predstavljeni so: prijava v sistem, varnost sistema, varnost objektov podatkovne baze, definiranje vlog, uporaba pogledov za omejitev dostopa in revizija baze.

V zadnjem poglavju so predstavljeni zaščitni mehanizmi dokumentnih zbirk v okolju Lotus Notes, ki je okolje za skupinsko delo in ima po mojih izkušnjah najbolj široko razdelane zaščitne mehanizme. Predstavljen je kompleksen mehanizem za overovitev uporabnika ter načini dodeljevanja pravic za dostop do strežnika, zbirk in njihovih sestavnih elementov (obrazcev, pregledov, polj). Na koncu je opisana še uporaba enkripcije zbirke, dokumenta, polja, sporočil elektronske pošte ter komunikacijskih vrat. V okolje so vključeni algoritmi simetrične in javne kriptografije.

I. DEL – Splošno o varnosti računalniških sistemov

1. Varen računalniški sistem

Varen računalniški sistem zagotavlja **zaupnost**, **integriteto** in **dostopnost** podatkov. V preteklosti se je zdaleč največ pozornosti posvečalo zaupnosti podatkov in s tem povezani pravici do zasebnosti, medtem ko so modeli zaščite integritete manj razviti, modeli dostopnosti pa še v povojih.

Zaupnost (confidentiality) pomeni prikaz podatkov le z varnostno politiko pooblaščenim osebam. Najpomembnejša je v vojaških in sistemih državne varnosti, kjer je že tradicionalno definirana več nivojska klasifikacija dokumentov (strogo tajno, tajno, zaupno, običajno). S širjenjem zbiranja podatkov (kupovanje preko interneta in katalogov, zdravstveni podatki, beleženje telefonskih pogovorov in druge evidence) pa so tudi naši osebni podatki ob nespoštovanju predpisov lahko hitro izpostavljeni številnim zlorabam (npr. pošiljanje reklamnih katalogov, kraja številk kreditnih kartic). S širitvijo poslovanja na internet pa je v zadnjem času zelo pomembno tudi dobro varovanje poslovnih informcij.

Integriteta (integrity) pomeni, da so vse spremembe podatkov avtorizirane, da so podatki konsistentni navznoter in z zunanjim svetom ter da sistem pravilno deluje. Širše lahko pri integriteti govorimo tudi o kvaliteti podatkov (data quality) vsebovanih v sistemu. Kvalitetni podatki so pravočasni, popolni iz izvirajo iz zanesljivih virov. Za zagotavljanje integritete je pomembno tudi, da so v aplikacije vgrajeni mehanizmi za preprečevanje napak pri delu. Pri omrežjih integriteta pomeni, da je prejeto sporočilo identično poslanemu.

Dostopnost (availability) pomeni, da so sistem in njegovi servisi na voljo vedno, ko jih potrebujemo.

Računalniška varnost je zaščita računalniških sistemov proti kršitvam zaupnosti, integritete in dostopnosti. Z računalniškimi sistemi so mišljeni računalniki, omrežja ter informacije, ki jih vsebujejo.

Varovanje računalniškega sistema posega na tri medsebojno povezana področja: socialno, upravljavsko in tehnično. Osnovni varnostni okvir postavljajo vladne organizacije, mednarodna telesa, izobraževalne institucije ter mediji s sprejemanjem ustreznih zakonov in standardov, ki varujejo pred zlorabami, računalniškem kriminalu in kršenju zasebnosti. V organizacijah mora vodstvo poskrbeti za primerno varovanje sistema pred naravnimi nesrečami ter za motivacijo in izobraževanje zaposlenih v zvezi z varnostjo.

Tehnična varovala predstavljajo zelo široko in pomembno področje, saj morajo biti vgrajena v:

- strojno opremo,
- operacijske sisteme,
- podatkovne baze in
- omrežno arhitekturo.

Osnovna načela večine varnostnih politik so:

- **avtorizacija**: kdo lahko uporablja določene resurse in na kakšen način,
- **minimalna pooblastila**: zaposleni smejo biti avtorizirani samo za uporabo resursov, ki jih potrebujejo za opravljanje svojega dela. S tem se bistveno zmanjša možnost namernih ali nenamernih napak in zlorab,
- **delitev dolžnosti:** pri zahtevnejših opravilih je potrebno več ljudi za njegovo izvedbo
- **redundantnost**: procedur, informacij in računalniških resursov (npr. varnostne kopije podatkov).

2. Standardi in računalniška varnost

Standardi so v računalniški industriji izredno pomembni, saj morajo zagotavljati medsebojno povezljivost in združljivost strojnih, programskih in komunikacijskih komponent najrazličnejših proizvajalcev. Standardi zato predpisujejo ustrezne odprte vmesnike, servise in protokole. Druga pomembna vloga standardov pa je zagotavljanje točno predpisanega nivoja določene lastnosti (npr. kakovosti, varnosti). Standardi se ne nanašajo samo na izdelke pač pa tudi na ljudi in delovne postopke. Pogosto nek v praksi pogosto uporabljan standard sam postane "de facto" ter ga šele pozneje formalno standardizira tudi pristojna vladna ali nevladna organizacija za standardizacijo. Na zagotavljanje ustrezne računalniške varnosti vplivajo standardi iz številnih področij, ki jih prikazuje spodnja slika.

Slika 1: Standardi, ki vplivajo na računalniško varnost

Prvi računalniški standardi so bili razviti pri nacionalnem varnostnem centru zaradi vojaških potreb ZDA po varovanju skrivnosti[1]. TCSEC[2] kriteriji so bili prvič objavljeni leta 1983 in so vplivali na postavljanje poznejših standardov tudi v drugih državah. TCSEC poudarjajo zaupnost, manj pa se posvečajo integriteti in dostopnosti.

[1] National Computer Security Center je del Nacional Security Agency .

Sodelovanje Francije, Nemčije, Nizozemske in Anglije je vodilo v razvoj ITSEC[3] kriterijev, ki so postali de facto standard v Evropi.

Vedno večja pomembnost poslovnih sistemov je vodila v razvoj standardov za kakovost (ISO[4]), elektronsko izmenjavo podatkov (EDI in EDIFACT) in enkripcijo sporočil (DES, RSA). Razvit je bil tudi zmožnostno zrelostni model (Capability Maturity Model), ki razvijalce programske opreme (glede na kakovost procesa) razvršča v pet razredov[5]. Uvrstitev v višji razred posledično pomeni tudi kvalitetnejše izdelke. Leta 1992 je OECD[6] razvila množico smernic za varnost informacijskih sistemov, ki predstavlja podlago za sprejem zakonov, politik, tehničnih in administrativnih meril ter izobraževanje.

Ob vedno bolj množičnem povezovanjem računalnikov v globalno omrežje postajajo ključnega pomena varnostni mehanizmi na vseh nivojih omrežne arhitekture. Dve vodilni odprti arhitekturi sta sedem nivojski ISO OSI[7] model in štiri nivojski internetni protokol TCP/IP.

Na področju kriptografskih standardov imata glavno besedo NIST[8] in NSA[9] s standardi kot so: DES, RSA, DSS[10] itd.

3. Katere nevarnosti nam grozijo in kako se jih ubraniti?

Številne grožnje, ki pretijo našim sistemom, se delijo na:

- **premišljene** (goljufije, zlorabe, hackerski vdori, virusi, črvi) in
- **naključne** (človeške napake, izpadi energije, naravne nesreče).

Zelo pomembno je, da vse vrste groženj čim bolje spoznamo in spremljamo njihovo širjenje ter z različnimi varnostnimi mehanizmi skušamo sisteme zavarovati ter s tem preprečiti ali vsaj omiliti škodo, ki jo lahko povzročijo kršitve zaupnosti, integritete ali dostopnosti. Katero od lastnosti moramo najstrožje varovati in kakšno stopnjo varnosti potrebujemo je odvisno od vrste sistema in s tem informacij hranjenih v njem. V vojaških računalniških sistemih sta seveda najstrožje varovani zaupnost (številni nivoji zaščite tudi znotraj sistema) ter integriteta podatkov.

---

[2] TCSEC - Trusted Computer System Evaluation Criteria
[3] ITSEC – Information Technology Security Evaluation Criteria
[4] ISO – International Organization for Standardization (ISO 9000-3 za proces razvoja programske opreme, ISO 9126 za kakovost produkta)
[5] Razredi Zrelostno zmožnostnega modela: začetni, ponovljiv, definiran (tega zahteva ISO 9000-3), voden, optimizacijski
[6] OECD – Organization for Economic Cooperation and Development
[7] OSI – Open Systems Interconnection
[8] NIST – National Institute of Standards and Technnology
[9] NSA – National Security Agency
[10] DSS – Digital Signature Standard (1994)

3.1 Varovanje zasebnosti v tujini in pri nas

Uporaba računalniške tehnologije omogoča enostavno zbiranje, shranjevanje, uporabo in povezovanje najrazličnejših podatkov. Med njimi je tudi vedno več osebnih podatkov. Viri teh podatkov so elektronsko poslovanje, telefonski pogovori, obvezna razmerja do vladnih inštitucij (npr. vojaške in policijske evidence, centralni register prebivalstva) itd.

Za zaščito zasebnosti je bilo zato potrebno sprejeti ustrezne varnostne mehanizme, ki bi preprečevali zlorabo osebnih podatkov. Vsaka organizacija, ki rokuje z osebnimi podatki, mora poskrbeti za ustrezno varnostno politiko, ki pa je deloma odvisna od države in njenih zakonov. Kljub vsemu so osebni podatki pogosto napačni, njihova zasebnost pa neprimerno varovana. Zaradi težjega povezovanja podatkov iz različnih evidenc naj bi bila prepovedana uporaba enotnega identifikatorja, a je v ZDA številka zdravstvenega zavarovanja postala ravno to, saj se uporablja kot identifikator pri vladnih, zdravstvenih, bančnih in drugih inštitucijah, kar je z vidika zasebnosti izredno nevarno.

Večina držav danes ima svoje zakone o zasebnosti, spoštujejo pa tudi različne mednarodne sporazume. Na večino zakonov je vplival kodeks (Code of fair information practices), ki ga je leta 1973 objavil vladni oddelek ZDA in vsebuje pet poglavitnih načel poštene uporabe informacij:

1. tajne evidence osebnih podatkov ne smejo obstajati,
2. posameznikom mora biti omogočeno vedeti, kakšni osebni podatki se o njih hranijo in čemu so namenjeni,
3. preprečiti je potrebno uporabo podatkov zbranih za določen namen v druge namene brez privolitve posameznika,
4. omogočeno mora biti dopolniti ali popraviti nepopolne in netočne osebne podatke,
5. organizacija, ki zajema, shranjuje in uporablja osebne podatke, mora zagotoviti ustrezne mehanizme za preprečevanje njihovih zlorab.

Na osnovi tega kodeksa je bil v ZDA leta 1974 sprejet tudi zakon o zasebnosti (Privacy Act), ki pa se nanaša samo na zvezne vladne podatkovne zbirke. Pozneje so bili sprejeti še drugi zakoni, ki pokrivajo posamezna področja: zasebnost bančnih podatkov[11], elektronskih komunikacij[12] itd.

Tovrstna zakonodaja je širša in bolj splošna v Evropi kot v ZDA. Svet Evrope je leta 1995 sprejel direktivo o varstvu osebnih podatkov za Evropsko unijo, ki vsem članicam narekuje sprejem ustreznih nacionalnih zakonov v skladu s priporočili OECD iz leta 1980. V Sloveniji je bil prvi Zakon o varstvu osebnih podatkov sprejet že marca 1990, lani pa je bil sprejet nov popravljen zakon z istim imenom (veljaven zakon najdeš v [12]).

---

[11] Financial Privacy Act ščiti bančne podatke pred vladnimi vpogledi.

[12] The Electronic Communications Privacy Act (1986) prepoveduje prisluškovanje in prestrezanje sporočil. The Communications Assistance for Law Enforcement Act (1994) pa zahteva, da se komunikacije vršijo na način, da jim vladne inštuticije lahko prisluškujejo, če je to potrebno.

3.2 Največ škode povzročijo zaposleni in bivši zaposleni

Uporabniki sistema, kamor poleg samih zaposlenih štejemo tudi njihove družinske člane, poslovne partnerje in kupce, so največji potencialni grešniki, saj sistem dobro poznajo in imajo tudi določena pooblastila, zato predstavljajo najresnejšo grožnjo zaupnosti, integriteti in dostopnosti podatkov. Pogosto veliko škode povzročijo že z nenamernimi napakami, če programska oprema ne nudi dovolj enostavnega vmesnika in vgrajenih mehanizmov preverjanja napak, pozabijo gesla, s katerimi so kriptirali dokumente in podobno. Njihove namerne zlorabe pa so bodisi goljufije, kraje ali sabotaže, ki jih s primernimi pooblastili pogosto ni težko izvesti. Za varovanje pred namernimi zlorabami zaposlenih je še kako pomembno spoštovanje že omenjenih načel delitve dolžnosti in minimalnih pooblastil.

Najmočnejša motivacija za zlorabe je reševanje osebnih problemov, pritiski, izzivi in šele na zadnjem mestu finančna korist. Raziskave v poznih 80-iz so pokazale, da 65 % povzročene škode pade na nenamerne napake sicer lojalnih zaposlencev, 13 % na nelojalne, 6 % na razočarane in samo 3 % na vdore od zunaj. Ostalo je škoda, ki nastane zaradi izpadov napajanja in naravnih nesreč. Napake zaposlencev (npr. nepravilno postavljeni varnostni mehanizmi, slaba skrb za gesla) pa ne povzročajo le neposredne škode, ampak pomenijo priložnost za enostavnejše vdore in zlorabe drugih zaposlencev in zunanjih napadalcev.

Še posebej nevarni pa so sistemski administratorji s svojim privilegiranim statusom, saj imajo vse pravice nad vsemi resursi, torej ob slabi motivaciji za delo idealne pogoje za vse vrste zlorab, od prebiranja tuje elektronske pošte, pregledovanja in izdaje zaupnih podatkov, izvedbe sabotaž itd. Tudi le malomarno ravnanje s "superuser" uporabniškim imenom in geslom lahko postane najprimernejša tarča zunanjih napadalcev. Zato je za organizacijo lahko ključnega pomena kakšnega človeka zaposli na tako delovno mesto ter kako ga motivira in nagrajuje, da bo deloval v njeno korist in ne proti njej.

3.3 Napadi od zunaj (intrusions)

3.3.1 Vdori hackerjev

Izraz hacker je originalno označeval programerja, ki rad preizkuša najrazličnejše spretnosti na operacijskih sistemih in drugih programskih izdelkih. Zaradi uporabe tako pridobljenih znanj o ranljivostih in pomanjkljivostih posameznih sistemov na tujih sistemih je izraz začel označevati tiste, ki vdirajo v tuje računalniške sisteme.

Verjetno najbolj znan hacker starejše generacije je Kevin Mitnick, ki je bil večkrat obsojen na zaporno kazen, čeprav je bila edina finančna korist nekaj brezplačne uporabe telefona. Motivacija večine hackerjev je le izziv, da znajo in zmorejo prodreti v še tako dobro varovane sisteme (npr. Department od Defence). Pri tem ponavadi ukradejo kakšne podatke ali programe, sesujejo sistem ali prisluškujejo zaupnim pogovorom, bistvo pa ni finančna korist, ampak dokaz, da jim je uspelo. Hackerji pri svojih vdorih izkoriščajo zanje o napakah in luknjah v programski opremi, malomarno sistemsko administracijo, lahkovernost privilegiranih uporabnikov itd.

3.3.2 Zlonamerni programi - Trojanski konji, virusi in črvi

Z vedno lažjo izmenjavo informacij (pošiljanje programčkov po pošti in njihovo snemanje z nepreverjenih spletnih mest) je grožnja zlonamerne kode vedno večja. Zato je potrebno uporabiti zaščitne mehanizme kot so: antivirusni programi, striktne omejitve nad spremembami sistemskih knjižnic, zaganjanje programov le certificiranih virov in brisanje elektronske pošte sumljive vsebine ali izvora. Predvsem pa spremljati in hitro obveščati uporabnike v zvezi nevarnostmi, ki se zaradi popularizacije interneta širijo vedno hitreje.

Trojanski konj je najstarejša oblika zlonamerne kode. Pogosto so vgrajeni v igrice in grafične programčke, ki hitro pritegnejo uporabnike. Program vsebuje skrite funkcije, ki se izvajajo v ozadju (npr. brisanje datotek). Pri snemanju najrazličnejših poskusnih programčkov z interneta, katerih avtor ni znan, obstaja velika nevarnost tovrstne okužbe. Trojanski konj pridobi privilegije uporabnika, ki ga je zagnal, zato je še posebej nevarno, če ga zažene administrator. S trojanskim konjem mnogo lažje kršimo integriteto kot zaupnost. To izhaja iz privilegijev, saj lahko program spreminja in uniči vse objekte do katerih ima uporabnik dostop, težje pa je kopiranje podatkov na mesto, kjer ima uporabnik pisalni dostop, napadalec pa bralnega. Kršitev integritete (npr. dodelitev bralnega dostopa za napadalca na vse uporabnikove datoteke) omogoča kasnejšo kršitev zaupnosti.

Virus je program, ki se pripne h gostiteljskemu programu (izvršljivi datoteki) ter se izvaja in širi skupaj z njim. Pogosto se tudi spreminja, take viruse imenujemo polimorfni. Ker so virusi pogosto časovno tempirani, lahko preteče veliko časa preden ga odkrijemo in med tem se je že lahko prenesel na vse programske datoteke (tudi varnostnih kopij) ter še na številne druge računalnike. Množično širjenje virusov se je začelo z uporabo osebnih računalnikov, ki niso imeli ustrezne zaščite, za to obdobje pa je bila tudi značilna množična nelegalna izmenjava programske opreme. Še danes se največ virusov nahaja na domačih in šolskih računalnikih, v podjetjih pa večinoma že obstaja ustrezna protivirusna zaščita (npr. Sophosov Intercheck [11]). Odkrivanje novih virusov je težavno, saj ne obstaja algoritem, ki bi splošno lahko odkril ali neka koda predstavlja virus. Tako se je potrebno omejiti na odkrivanje znanih virusov in čim pogosteje nalagati nove različice protivirusnih programov. Ponudniki protivirusnih programov na svojih straneh objavljajo tudi opise, delovanje in razširjanje najnovejših virusov, s čimer lahko, seveda ob primerni uporabniški osveščenoti, močno omejijo njihovo širjenje.

V zadnjem času pa je na pohodu že nova oblika zlonamerne kode, imenovane **črvi** (npr. ILOVEYOU – glej [9] in [10]), ki izkorišča množično uporabo internetnih komunikacij za svoje hitro širjenje. Razlika med virusi in črvi je v načinu razširjanja, mestu hranjenja in načinu izvajanja.

Virus se širi s prenosom okužene datoteke iz računalnika v računalnik, medtem ko se črv širi preko komunikacijskih poti (npr. preko elektronske pošte) kot samostojen programček pripet k sporočilu, ki se aktivira, ko kliknemo priponko. Napisan je v enem od jezikov (npr. VBScript), ki se lahko izvajajo že z inštalacijo samega operacijskega sistema ali v spletnem brskalniku. S tem je zagotovljeno, da se bo črv izvedel v večini

računalnikov, kar omogoča njegovo izredno hitro širjenje, saj je napisan tako, da se pošlje naprej vsem naslovnikom iz uporabnikovega imenika. Pri zagonu črva se lahko izvedejo npr. spremembe in brisanje datotek v lokalnem računalniku in vseh omrežnih pogonih, do katerih ima uporabnik pravice. Prav zaradi hitrejšega širjenja izgleda še bolj uničujoč od svojega predhodnika virusa.

3.4 Priprava, aktiviranje in izvršitev napada

Napad na računalniški sistem je sestavljen iz treh faz:

- pripravljalne,
- aktivacijske in
- izvršitvene.

3.4.1 Pripravljalna faza

Pripravljalna faza je namenjena zbiranju najrazličnejših informacij. Notranji uporabniki lahko do teh pridejo dokaj enostavno, bolj prebrisani morajo biti zunanji napadalci. Le ti uporabljajo različne trike, da te informacije dobijo od uporabnikov ali celo administratorja z različnimi metodami.

1. Uporaba **Trojanskega konja** je eden od načinov za pridobivanje gesel uporabnikov in drugih podatkov do katerih ima dotični dostop ter spreminjanje dostopnih pravic.
2. **Stranska vrata** (trap door ali back door) so skriven mehanizem, ki ga razvijalec vgradi v programsko opremo in ga pozna le sam. Ko je to potrebno, lahko npr. vstopi v sistem brez ustreznega gesla, blokira delovanje sistema in podobno. V 70-ih se je govorilo, da ima DES taka stranska vrata.
3. Metoda **"masquerading and spoofing"** pomeni pretvarjanje, da si nekdo drug z namenom pridobiti njegove dostopne pravice, gesla, PIN kode itd.

3.4.2 Aktivacijska faza

Zakasnitev te faze glede na pripravljalno pogosto pomeni težje odkrivanje napadalca, napad je lahko bolj uničujoč, posledice pa težje odpravljive. Za sprožitev se uporabljajo logične in časovne bombe, ki sprožijo pripravljen napad (npr. nameščen virus se sproži šele na točno določen dan, že prej pa okuži številne druge računalnike).

3.4.3 Izvršitvena faza

Izvršitev pripravljenega napada lahko pomeni pasivno ali aktivno zlorabo. Pri aktivni zlorabi gre za kršenje integritete in dostopnosti sistema:

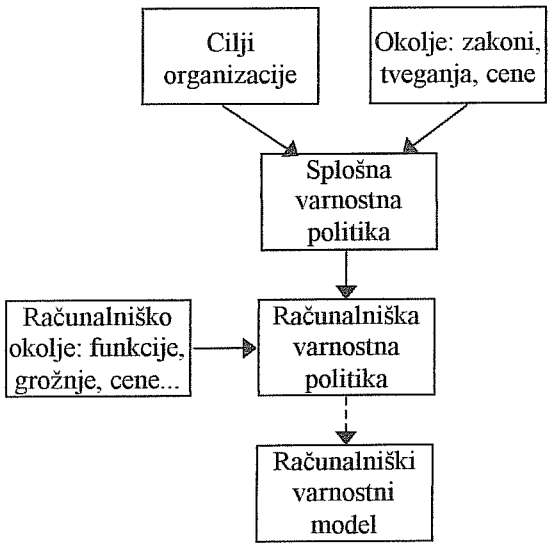
- uničenje ali spremembo datotek,
- spremembo, brisanje, vstavljanje ali preusmerjanje sporočil,
- sesutje celotnega sistema in
- uničenje evidenc o vdoru.

Ko je kršena le zaupnost, sam sistem pa ni kakorkoli spremenjen, govorimo o pasivni zlorabi. Tehnike so: prisluškovanje, prestrezanje sporočil, škiljenje pri vpisovanju gesel itd.

4. Varnostna politika

V vsaki organizaciji mora veljati neka varnostna politika, ki zagotavlja ustrezen nivo varnosti. Njena varnostna politika je veliki meri odvisna od okolja v katerem deluje in dejavnosti s katero se ukvarja. Tako je splošna varnostna politika pogojena z množico zakonov, pravil in praks, ki regulirajo upravljanje, zaščito in distribucijo virov za zagotavljanje ustrezne varnosti.

Računalniška varnostna politika mora tako interpretirati splošno politiko za vse računalniške vire: datoteka, podatkovne baze, transakcije in komunikacije. Skrbno pa mora pretehtati tudi vse nevarnosti, ki z uporabo informacijske tehnologije, dodatno ogrožajo organizacijo (npr. virusi) in določiti ustrezna varovala.



Slika 2:Vpliv različnih dejavnikov na računalniški varnostni model

Tako je računalniška varnostna politika sestavljena iz dve delov:

- interpretacije in delne avtomatizacije splošne politike (kdo lahko odobri bančno transakcijo, podpiše ček, ima vpogled v določene podatke) in
- politike glede splošnih računalniških varnostnih groženj (sprememba gesel vsak mesec, mesečno ažuriranje protivirusne zaščite).

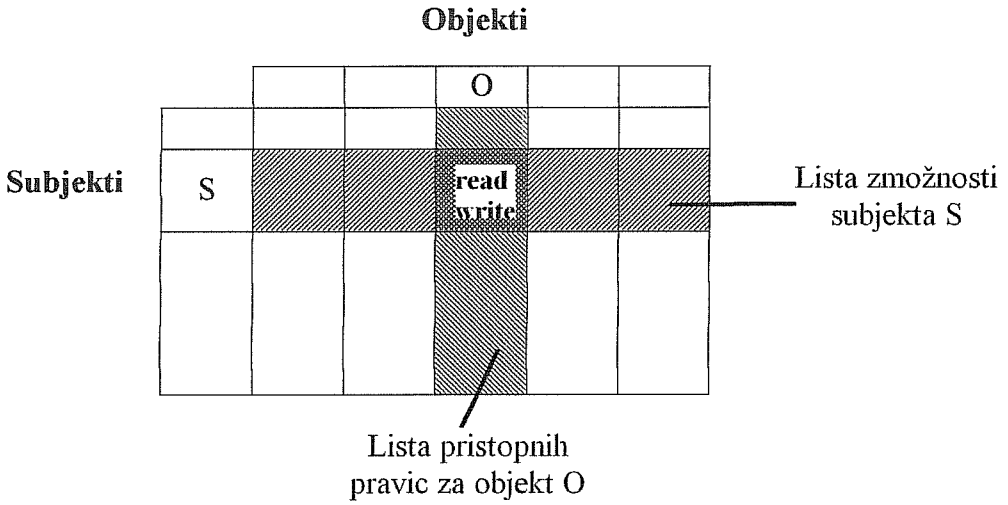
Računalniška politika definira tako sistemske lastnosti kot tudi odgovornost uporabnikov pri njenem striktnem uresničevanju. Prav razumevanje in izvrševanje predpisanih pravil s strani uporabnikov je ključnega pomena za njeno učinkovitost.

Kompleksno varnostno politiko pisano v naravnem jeziku je potrebno zaradi lažje implementacije in rokovanja zapisati kar se da formalno, matematično, zato so bili

razviti različni varnostni modeli. Model definira kateri servisi bodo vključeni in pogosto tudi s katerimi mehanizmi bodo realizirani. Od treh varnostnih lastnosti so je bilo največ modelov posvečenih zaupnosti, manj razviti so integritetni modeli, najslabše razviti pa modeli v zvezi z varovanjem dostopnosti. Prav integriteta podatkov pa je z uvajanjem računalniške tehnologije v poslovanje in zdravstvo postala celo najpomembnejša.

Čeprav je varnostna politika pogosto prilagojena posameznim aplikacijam pa obstajajo elementi, ki so skupni večini politik. Ključni element je vsekakor vloga (role). Že sama splošna politika temelji na principu vlog, iz njih pa izhajajo določene pravice do uporabe resursov, ki jih uporabniki potrebujejo za opravljanje svojega dela. Ko prenehajo opravljati delo, izgubijo pravice potrebne za to delo. Pri preučevanju vojaških aplikacij so bili odkriti številni skupni varnostni elementi: zaščita informacij pred nepooblaščenim razkritjem in spremembami, na vlogah temelječe dostopne pravice, izključevanje vlog, poverjanje pooblastil, dva in več osebni nadzor, identifikacija, overovitev in revizija. Ti elementi se pojavljajo v večini varnostnih politik ne glede na področje, ki ga pokriva aplikacija.

4.1 Matrični varnostni model



Slika 3: Matrični model dostopov – Access Matrix of Butler Lampson

Nadzor dostopov je proces, ki zagotavlja, da so vsi dostopi do resursov strogo avtorizirani. S tem podpira tako zaupnost kot integriteto podatkov.

Enostaven in intuitiven matrični model dostopov (slika 3) je bil leta 1971 razvit za varovanje večuporabniških operacijskih sistemov. V matriki imamo subjekte, objekte in pravice. Objekti so v konkretnem primeru lahko datoteke ali deli pomnilnika. Subjekti so aktivne entitete tega modela in so konkretno lahko uporabniki ali izvajajoči se procesi. Matrika dostopov vsebuje eno vrstico za vsak subjekt in stolpec za vsak objekt. Vsaka celica matrike AM[S,O] specificira pravice subjekta S nad objektom O (read, write na sliki). Stolpec v matriki imenujemo **lista pristopnih pravic** za objekt O (access control list), ki je seznam pravic vseh subjektov nad objektom O. Vrstico

matrike pa imenujemo **lista zmožnosti subjekta** S (capability list), ki je seznam vseh pravic subjekta.

Matrika dostopov ni statična, saj objekti in subjekti nenehno nastajajo in izginjajo. To pomeni, da mora model vsebovati mehanizme za spreminjanje matrike, za ustvarjanje in brisanje objektov in subjektov ter dodeljevanje pravic.

II. DEL - Zaščita podatkovnih baz

Podatkovna baza v številnih poslovnih sistemih predstavlja srce celotnega sistema, saj je v njej zapisana celotna organizacijska struktura pa tudi vsi podatki o poslovnih procesih in dogodkih povezanih z njimi. Podatkovna baza kot osrednji povezovalni element med posameznimi poslovnimi procesi mora v vsakem trenutku zagotavljati, da so podatki v njej v vsakem trenutku na voljo vsem uporabnikom, ki jih potrebujejo, ter da so ti podatki vedno konsistentni, točni, ažurni in podobno. Na osnovi podatkov iz preteklosti in določenim dodatnim poznavanjem gibanj na trgu (trendov) omogoča vodstvenemu kadru sprejemanje informacijsko podprtih odločitev, ki so že ali naj bi čimprej popolnoma nadomestile zgolj intuitivno odločanje.

Takšna vloga podatkovne baze v sodobnem poslovnem sistemu pomeni, da je zaščita podatkovne baze lahko eden od kritičnih dejavnikov njegove uspešnosti. Hitro ukrepanje, odzivanje na spremembe, nove tržne priložnosti (npr. internet), vse to je povezano s pravočasno, točno in relevantno informacijo, ki jo v določenem trenutku imamo ali pa nimamo na voljo. Ker so prave informacije bogastvo, jih moramo tudi primerno zaščititi pred nepooblaščeno uporabo. Varovanje različnih poslovnih skrivnosti, npr. podatkov o poslovnih parterjih je nujno potrebno, saj je v poslovnem svetu vedno več primerov tako imenovane "špijonaže", kjer skušajo podjetja s krajo informacij o svojih konkurentih pridobiti določeno konkurenčno prednost.

Za varovanje državnih podatkovnih baz (vojaške, policijske in druge evidence) so že dalj časa uporabljajo ustaljeni varnostni mehanizmi, ki zagotavljajo zaupnost, integriteto in dostopnost baz v vsakem trenutku. Na teh bazah so postavljeni mehanizmi za spremljanje vsake uporabnikove dejavnosti (tudi vpogledov), ki v primeru izdaje določenih podatkov, omogočajo najti vse osebe, ki so dostopale do podatkov. Nepooblaščeni dostopi v takih primerih pomenijo tudi kazensko odgovornost zaposlenega.

Stopnje in načini varovanja podatkovnih baz podjetij so le deloma urejeni z zakoni in predpisi, zato je to področje večinoma odvisno od primerne osveščenosti vodstva in usposobljenosti strokovnega kadra, ki morajo skupaj določiti ustrezno varnostno politiko in se dogovoriti o višini sredstev, ki jih bodo namenjali zaščiti.

Na trgu obstajajo številni produkti, ki že imajo vključene tudi ustrezne varnostne mehanizme. V svoji seminarski nalogi sem se odločila predstaviti dva vodilna produkta na področjih:

- relacijskih podatkovnih baz – Oracle in
- dokumentnih podatkovnih baz - Notes/Domino.

Produkta pokrivata vsak svoje področje in se zato medsebojno dopoljnjujeta. Oracle kot baza za vse transkacijske podatke in Notesove zbirke za upravljanje z dokumenti, znanjem in sodelovalne aktivnosti.

Ker se pogosto dogaja, da v organizacijah uporabljajo omenjena produkta, ne poznajo pa številnih že vgrajenih varnostnih mehanizmov, sem skušala predstaviti prav te. Zaščitni mehanizmi v Oraclovu bazi so povzeti po [3] in [4], medtem ko se pri opisu zaščite v okolju Lotus Notes naslanjam na [5], [6], [7], [13], [14] in lastne izkušnje pri administraciji.

V prvih dveh poglavjih so predstavljene tudi osnovne definicije pojma podatkovna baza in naloge sistema za upravljanje s podatkovno bazo za tiste, ki jim področje ni domače.

1. Kaj je podatkovna baza

Podatki omogočajo posamezniku ali organizaciji shraniti na pomnilniški medij tisti del svojega znanja, ki se nanaša na pomnjenje dejstev. Iz takega razumevanja podatkov sledi definicija podatkovne baze (definicije in pripadajoče razlage vzete iz [2]):

1. **Podatkovna baza je model okolja, ki služi kot osnova za izvajanje akcij in sprejemanje odločitev.**

Iz definicije tudi sledi, da so podatki osnova, na kateri temelji celotno delovanje organizacije, zato je potrebno z njimi, enako kot z ostalimi sredstvi, smiselno upravljati.

Upravljanje podatkovne baze zajema:

- zagotavljanje razpoložljivosti podatkov (availability) in
- nadzor nad uporabo podatkov, v okviru česar skrbimo za:
  - celovitost (integriteto) podatkov,
  - uporabo v skladu z njihovim namenom,
  - uporabnost podatkov tudi v prihodnje.

Bolj tehnično gledanje na pojem podatkovne baze prinaša še dve definiciji:

2. **Podatkovna baza je zbirka med seboj pomensko povezanih podatkov, ki so shranjeni v računalniškem sistemu, dostop do njih je centraliziran in omogočen s pomočjo sistema za upravljanje podatkovnih baz (SUPB).**

3. **Podatkovna baza je mehanizirana, večuporabniška, formalno definirana in centralno nadzorovana zbirka podatkov.**

Iz njiju sledi, da se v podatkovni bazi shranjujejo podatki, ki se nanašajo na določen del sveta, v katerem so vse stvari in dogajanja medsebojno povezana; da je baza računalniško podprta; da obstaja mehanizem – SUPB za nadzor in dostop do podatkov; da obstajajo predpisi, kakšni in kateri podatki se smejo nahajati v bazi ter kakšen je njihov pomen. To je širši pomen pojma podatkovna baza. V ožjem smislu pa pogosto pod tem pojmom mislimo le na podatke, ki so pod upravljanjem SUPB shranjeni na nekem pomnilnem mediju. Podatkovna baza je lahko z vidika upravljanja organizirana v centralizirani ali porazdeljeni obliki. SUPB je lahko izveden kot programska oprema lahko pa tudi kot samostojen podatkovni računalnik.

Podatkovni del baze je sestavljen iz dveh delov:

- fizične podatkovne baze in
- meta podatkovne baze.

V meta podatkovni bazi so shranjeni opisi fizičnih podatkov, kaj pomenijo in kako so dostopni uporabnikom. Obe bazi sta shranjeni v obliki datotek na zunanjem pomnilnem mediju, do katerih SUPB ne dostopa direktno, ampak preko operacijskega sistema, s čimer je dosežena neodvisnost od tipa centralne enote in tipa pomnilnih medijev. Tukaj pa pride do izraza, da je še kako pomembna varnost samega operacijskega sistema, saj so ob neprimernih zaščitnih mehanizmih na nivoju operacijskega sistema lahko tudi vse zaščite na višjih nivojih popolnoma brezpredmetne.

V skladu s trinivojsko arhitekturo podatkovne baze vsebuje meta podatkovna baza tri vrste opisov fizičnih podatkov, imenovanih:

- notranja shema,
- konceptualna shema,
- zunanja shema.

Namen trinivojske arhitekture je ločiti načine fizičnega shranjevanja podatkov (notranja shema) od konceptualnega modeliranja ter le-tega od uporabe podatkov (zunanja shema). V meta podatkovni bazi se nahaja toliko zunanjih shem, kolikor različnih uporabniških pogledov obstaja. Zunanja shema prilagodi in omeji uporabnikov pogled na zanj relevantne podatke.

Na konceptualnem in zunanjem nivoju je podatkovna baza opisana s pomočjo podatkovnih modelov. Trenutno najbolj razširjen je gotovo **relacijski podatkovni model**, vedno več govora je tudi o objektnih pristopih pri modeliranju (npr. UML[13]), predvsem v raziskovalnih krogih. V preteklosti sta se več uporabljala tudi mrežni in hierarhični podatkovni model.

[13] UML – Unified Modeling Language

2. Sistem za upravljanje s podatkovno bazo (SUPB)

Delo s podatkovno bazo poteka preko SUPB, ki vsebuje:

- Data Description Language za definiranje novih elementov podatkovne baze, za kar skrbi njen administrator,
- Data Manipulation Language, ki je namenjen poizvedovanju in spreminjanje njene vsebine in
- Data Control language za izvajanje kontrolnih funkcij in nadzor dostopov do podatkovne baze.

SUPB ščiti bazo pred aplikacijski napakami in nepooblaščenim dostopom (overovitev in mehanizmi za nadzor dostopa). Za celovitost (integriteto) podatkov skrbijo:

- integritetne omejitve (integrity constrains),
- mehanizmi za nadzor nad sočasno uporabo (concurrency control) in
- mehanizmi za obnavljanje podatkovne baze (recovery).

Omogoča tudi, da različni uporabniki ali aplikacije prikažejo iste podatke na različne načine.

3. Varnostni mehanizmi

Za vsak podatkovni sistem je potrebno definirati varnostno politiko in priskrbeti servise in mehanizme, ki implementirajo z njo definirane zahteve, ki se pri podatkovnih bazah v mnogočem razlikujejo od tistih pri samih operacijskih sistemih, saj so objekti, ki jih varujemo mnogo kompleksnejši, pogosto sestavljeni tudi iz drugih objektov. Poleg samih podatkov morajo biti posebej dobro varovani tudi metapodatki. Tudi nadzor nad sočasno uporabo in obnavljanje baze sta zahtevnejša kot na nivoju datotek, saj je baza sestavljena iz številnih relacij. Ker SUPB tesno podpira različne aplikacije, je na vlogah temelječ nadzor dostopov še pomembnejši kot pri varnosti na nivoju operacijskih sistemov.

V poglavju so predstavljeni mehanizmi za izvajanje transkacij, integritetne omejitve, nadzor nad sočasno uporabo, overovitev uporabnika in kontrolo dostopa. Predvsem overovitev uporabnika in različni nivoji pristopnih pravic pa sta tisti področji, ki sem ju želela še posebej poudariti, zato ju podrobneje predstavljam na dveh konkretnih primerih v poglavjih 4. Varnostni mehanizmi SUPB Oracle in 5. Varnost v okolju Lotus Notes. Predvsem dodeljevanje pravic je tisto področje, ki mu administratorji posvečajo premalo pozornosti. S spremembami v organizacijskem sistemu (novi zaposleni, napredovanja, odpuščanja) se dinamično spreminjajo tudi pravice oziroma bi se morale (na vlogah temelječe pravice). Pogosto je ravno tovrstna neažurnost vzrok za kršitve zaupnosti in integritete podatkov.

3.1 Integriteta podatkovne baze

V tabeli 1 predstavljam seznam devetih integritetnih principov, ki so podprti z različnimi mehanizmi SUPB, kot sta jih definirala Sandhu in Jajodia (1995 IEEE) in ki so večinoma razloženi tudi v različnih poglavjih moje seminarske naloge. Še podrobneje so opisani v knjigi Secure Computing [1].

Večina integritetnih mehanizmov pripada SUPB, saj operacijski sistem ne more zagotavljati ustrezne integritete podatkovnih baz, ki je pogosto pogojena tudi s samimi aplikacijskimi zahtevami. Prepustiti varovanje integritete samim aplikacijam pa bi zopet bilo napačno, saj razvijalci posamezne aplikacije pogosto ne poznajo celotne baze. Tako tukaj nastopi pomembna funkcija SUPB pri zagotavljanju njene integritete.

Integritetni varnostni mehanizmi SUPB se tako delijo v tri skupine:

- **integritetne omejitve**: varovanje pred napakami in zlorabami uporabnikov in aplikacij, imenujemo tudi semantična integriteta,
- **nadzor nad sočasno uporabo**: varovanje pred napakami, ki bi nastale zaradi sočasne rabe uporabnikov in aplikacij in
- **obnavljanje podatkovne baze**: ki je potrebno v primeru sistemskih napak.

Integritetni principi	SUPB mehanizem
Dobro definirane transkacije *(Well-formed transactions)*	Enkapsulirana ažuriranja *(Encapsulated updates)*
	Atomične transkacije *(Atomic Transactions)*
	Konsistentnostne omejitve *(Cosistency constraints)*
Neprekinjeno delovanje *(Continuity of operation)*	Redundantnost *(Redundancy)*
	Obnavljanje *(Recovery)*
Overovitev uporabnikov *(Autenticated users)*	Overovitev *(Autentication)*
Minimalna pooblastila *(Least privilege)*	Dobro razčlenjena dostopna pravila *(Fine grained access control)*
	Pogledi *(Views)*
Delitev dolžnosti *(Separation of duty)*	Transkacijski nadzor *(Transaction controls)*
	Slojena ažuriranja *(Layered updates)*
Rekonstrukcija dogodkov *(Reconstructions of events)*	Revizija *(Audit trail)*
Pooblaščanje *(Delegation of authority)*	Dinamična avtorizacija *(Dynamic autorization)*
	Omejitve nad širjenjem pooblastil *(Propagation constraints)*
	Vloge *(Roles)*
Preveritev dejanskega stanja *(Reality checks)*	Posnetki dejanskega stanja *(Consistent snapshots)*
Enostavnost varne uporabe *(Ease of safe use)*	Privzete vrednosti *(Fail-safe defaults)*
	Človeški faktor *(Human factors)*

Tabela 1: Integritetni principi (Sandhu in Jajodia, 1995 IEEE)

3.1.1 Transakcijski mehanizem

Transkacija grupira akcije nad podatkovno bazo v skupino, ki omogoča njen prehod iz enega konsistentnega stanja v drugo. Skupina akcij ima naslednje lastnosti:

- **atomarnost**: izvede se celotna skupina akcij (commit) ali pa nobena od njih (abort),

- **konsistentnost**: transakcija ohranja konsistentnost podatkovne baze, saj ne krši integritetnih pravil in omejitev,
- **izoliranost**: transakcija je izolirana od ostalih transakcij. Tudi če berejo in pišejo nad istimi podatki, je učinek enak kot če bi se izvedle zaporedno in
- **trpežnost** : če se transakcija izvede, bodo njene spremembe preživele poznejše napake.

3.1.2 Integritetne omejitve

Kriteriji, ki zagotavljajo konsistentnost podatkovne baze, so postavljeni v obliki integritetnih omejitev. Te se lahko nanašajo tako na stanje (state constraints) kot tudi na prehode (transition constraints) med stanji. Splošna oblika omejitve je (t,c,a), kjer predstvljajo:

- **t**: specificira tipe ažuriranj, ki lahko prekršijo omejitev (npr. brisanje),
- **c**: je predikat, ki mora biti resničen za stanje ali prehod in
- **a**: odzivna akcija v primeru kršitve (ponavadi abort transkacije, lahko brisanje določenih zapisov).

Preverjanje integritetnih omejitev šele na koncu transakcije lahko pomeni veliko izgubo časa, zato so v SUPB vgrajena tudi vmesna preverjanja (checkpoints). V primeru kršenja omejitev tako mehanizmi za povrnitev v prvotno stanje (rollback) to izvedejo samo za korak nazaj do vmesne točke, kjer omejitve še niso bile kršene. S tem se izboljšajo performanse sistema.

Pri opisovanju SQL standarda sta Chris Date in Hugh Darwen (1993) integritetne omejitve kategorizirala v naslednje tri skupine:

- **omejitve nad domeno**: te potem veljajo nad vsemi stolpci, ki so definirani nad to domeno (npr. domeno rojstni datum omejimo na neničelne vrednosti na intervalu od 1900 do 2100, tudi podatkovni tip je vnaprej ograjena omejitev takšne vrste),
- **splošne omejitve**: nad poljubno tabelo lahko definiramo neke vsebinske omejitve, npr. minimalni znesek naročila in
- **omejitve nad osnovnimi tabelami**: sem sta vključila omejitve primarnih in sekundarnih ključev (referenčna integriteta relacijskega modela) ter preveritvene omejitve (iz druge tabele), ki morajo držati za vsak zapis v tabeli.

3.1.3 Nadzor nad sočasno uporabo

Zaradi izmeničnega izvajanja transakcijskih programov v večuporabniških sistemih tudi SUPB izmenično izvaja ukaze, ki mu jih programi posredujejo. Takšno izvajanje pa povzroča dve nevarnosti: da podatkovna baza zaide v nekonsistentno stanje ali pa so rezultati povpraševanj napačni. Ena od takšnih nekonsistenc se imenuje "izgubljeno ažuriranje". Nastopi takrat, ko dve transkaciji na osnovi prebranih podatkov ažurirata isti podatek. Takrat ostane v bazi samo zadnje ažuriranje, prvo pa se izgubi. Druga možnost je "branje neobstoječega podatka" v primeru, če transakcija prebere podatek, ki ga je ažurirala druga transakcija, pri kateri pa je tik za tem prišlo do napake in je bila razveljavljena.

Za nadzor nad sočasno uporabo podatkovne baze skrbi komponenta SUPB imenovana Scheduler. Dobljeno zahtevo lahko dovoli, zakasni ali onemogoči. Njegova naloga je ohraniti konsistentnost podatkovne baze v večuporabniškem okolju, hkrati pa zagotoviti čim večjo sočasnost in dostopnost sistema. V ta namen se uporabljata dve tehniki: zaseganje podatkov (locking) in časovno označevanje (timestamping).

3.2 Overovitev

SUPB je navadno implementiran nad operacijskim sistemom, tako lahko operacijski sistem skrbi tudi za identifikacijo in overovitev uporabnikov pri uporabi podatkovne baze. V tem primeru je dobro da, ko pride do uporabnikovega dostopa do podatkovne baze, operacijski sistem izvede ponovno overovitev, da zagotovi, da isti uporabnik še vedno drži sejo.Če pa se SUPB nahaja na ločenem aplikacijskem strežniku, potem ta skrbi za uporabniška imena in overovitev.

SUPB Oracle nudi obe možnosti kot je opisano v poglavju 4.2 Prijava na sistem. V Lotus Notes-u potrebujemo novo uporabniško ime in geslo, ki je neodvisno od sistemskega. Overovitev se vrši z dovršenimi lastnimi mehanizmi, ki jih opisujem v poglavju 5.1 Overovitev uporabnika.

3.3 Kontrola dostopa (Access Control)

Kontrola dostopa je zaščitni mehanizem, ki dovoljuje le avtorizirane dostope do objektov podatkovne baze. Predstavljen je že z matričnim modelom v I. delu v poglavju 4.1. Avtorizacijski model je sestavljen iz pravil oblike (S,O,A,P), kjer ima subjekt S dostopne pravice tipa A do tistih primerkov objekta O, za katere je predikat P resničen. Pri večini podatkovnih baz prevladuje lastniški model (ownership model), kar pomeni, da ima tisti, ki objekt (tabelo, pogled) kreira, nad njim vse pravice in jih lahko podeli tudi drugim ali pozneje odvzame. Načini dodeljevanja in odvzemanja pravic (grant and revoke) so pri različnih sistemih definirani različno. Standard SQL-92 na primer onemogoča poverjanje sistemskih pravic (kreiranje tabel in uporabnikov, zagon in ustavitev SUPB) drugim, nad njimi smejo imeti nadzor le administrator in razvijalci aplikacij.

Mehanizmi kontrole dostopa SUPB Oracle so podrobneje predstavljeni od poglavja 4.3 Varnost objektov podatkovne baze do poglavja 4.7 Revizija podatkovne baze. Notesova kontrola dostopa je večnivojska: omogoča nadzor nad dostopom do strežnika, dokumentne zbirke, dokumenta, sekcije, obrazca, pogleda in samega polja. Predstavljena je v poglavjih 5.2 Zaščita strežnika, 5.3 Zaščita dostopa do dokumentnih zbirk in 5.4 Sistemske zbirke.

4. Varnostni mehanizmi SUPB Oracle

Podjetje Oracle je danes nedvomno največji ponudnik programske opreme na področju srednjih in velikih sistemov za upravljanje relacijskih podatkovnih baz. Rezultat dolgoletnih izkušenj na tem segmentu informacijske tehnologije je robustna podatkovna baza z obsežnim varnostnim sistemom, ki preprečuje nepooblaščen dostop do

podatkov. Še tako zanesljiv, tehnološko dorečen varnostni sistem pa ne more odpraviti pomanjkljivosti, ki so rezultat neustrezne varnostne politike organizacije. Človeški faktor je pri vzpostavitvi varnega sistema prav tako, če ne celo bolj pomemben od tehnološkega. V nadaljevanju so predstavljeni najpomembnejši varnostni mehanizmi sistema za upravljanje podatkovnih baz Oracle (povzeti po [3] in [4]), ki vključujejo upravljanje z objektnimi in sistemskimi pravicami ter izvajanje revizije podatkovne baze.

4.1 Relacijske podatkovne baze

Relacijski podatkovni model je bil razvit pri IBM-u, njegov avtor E. F. Cood je svoje delo objavil v članku "A relational model of data for large shared data banks" leta 1970. Vsi vodilni sistemi za upravljanje s podatkovno bazo (Oracle, Microsoft SQL Server, Ingres) danes temeljijo na relacijskem modelu, prav tako je bilo ravno na teh sistemih narejenega največ v zvezi z zaščito podatkov.

Osnovni element relacijskega podatkovnega modela je relacija, ki je predstavljena v obliki tabele. Atributi relacije povedo, kateri podatki se nahajajo v posameznih stolpcih. Atributi so zgrajeni nad določeno domeno (zalogo vrednosti). V vrsticah imamo n-terice podatkov, ki opisujejo posamezne primerke relacije. Vsaka relacija ima primarni ključ, ki je atribut ali množica atributov, ki enolično določa vsako n-terico relacije.

Relacijski model definira pet osnovnih operacij: selekcijo, projekcijo, unijo, razliko in kartezični produkt ter še druge izpeljane operacije kot je npr. združevanje tabel (join). S pomočjo teh operacij lahko gradimo nove relacije, imenovane pogledi (views). Pogledi so izrednega pomena pri varovanju podatkov, saj v njih lahko skrijemo določene atribute (glede na osnovno relacijo) ali implementiramo dostopne omejitve na osnovi podatkov.

Definicijski in poizvedovalni jezik relacijskih baz se imenuje Standard Query Language ali krajše **SQL**. Razvit je bil pri IBM-u kot del System-a R, ki se je pozneje razvil v veliko bolj znani SUPB DB2. Jezik omogoča definiranje in brisanje relacij (tabel in pogledov), ter manipulativne ukaze (insert, select, update, delete) za delo s samimi podatki. Za kontrolo dostopov med drugim vsebuje ukaza grant in revoke (dodelitev in odvzem določenih pravic). Standard SQL-92 je bil potrjen kot ANSI in ISO standard.

4.2 Prijava na sistem

Oracle podpira dva osnovna tipa prijave na podatkovno bazo:

- **Oraclova interna gesla**: interna gesla predstavljajo tradicionalni način prijave in so neodvisna od prijave v operacijski sistem. Uporabniku dodeli račun in začetno geslo administrator podatkovne baze, pri čemer ponavadi isto geslo zagotavlja dostop do različnih podatkovnih baz. To lahko predstavlja veliko nevarnost v primeru če se z geslom ne ravna pravilno oziroma je že geslo samo preveč predvidljivo. Prav zavoljo tega morajo gesla slediti naslednjim standardnim priporočilom:
  - dolga naj bodo vsaj šest znakov,
  - vsebujejo naj kombinacijo alfanumeričnih in numeričnih znakov,

- ne smejo vsebovati znanih nizov znakov (imen, rojstnih datumov, besed iz slovarjev itd.),
- menjave naj se opravljajo čim pogosteje.

Uporaba priporočil zmanjšuje možnost vdora v sistem, pri čemer pa je ne more povsem odpraviti. Iz navedenega sledi, da je potrebno veliko pozornost posvečati izobrazbi uporabnikov, ki se morajo zavedati svoje odgovornosti pri ravnanju z gesli.

- **Prijava preko gesla operacijskega sistema**: ta metoda predpostavlja obstoj gesla že na nivoju operacijskega sistema. Oraclova podatkovna baza ob zagonu vsake aplikacije za dostop do podatkov primerja račun uporabnika na operacijskem sistemu z računom v podatkovni bazi. Na ta način že sam operacijski sistem garantira za prijavljenega uporabnika. Če je pravilno uporabljen, omenjen način prijave zagotavlja razmeroma dobro varnost, pri čemer pa se je potrebno zavedati dejstva, da lahko npr. 'root' uporabnik v UNIX sistemih nemoteno preklaplja med uporabniki, kar mu posredno omogoča dostop do vseh objektov podatkovne baze. Metoda prijave na podatkovno bazo z geslom operacijskega sistema zahteva torej pristajanje na določene kompromise, kar pa zaradi varnostnih razlogov pogosto ni primerno.

Celoten koncept varnostnih mehanizmov podatkovne baze Oracle temeljni na tem, da uporabniki z višjimi pravicami (ponavadi administratorji) dodeljujejo posamezne pravice ali skupine pravic uporabnikom z nižjimi pravicami. Oracle pozna dva tipa pravic (objektni in sistemski), ki varujejo podatkovno bazo z dveh različnih vidikov: dostopa do objektov in dostopa do sistemskih virov.

4.3 Varnost objektov podatkovne baze

Oraclova podatkovna baza vsebuje obširen nabor objektov, do katerih je potrebno omejiti dostop zgolj na pooblaščene uporabnike. Najpomembnejši objekti so tabele, pogledi (poizvedbe nad podatkovno bazo), shranjeni postopki, prožilci, sekvence, paketi itd. V osnovi ima vsak uporabnik zgolj pravice nad objekti podatkovne baze, ki jih kreira sam v okviru svoje sheme (uporabniškega imena). Uporabnik lahko nadalje pravice nad svojimi objekti dodeljuje ostalim uporabnikom z uporabo ukaza GRANT:

GRANT pravica ON objekt TO uporabnik;

Oracle omogoča dodeljevanje naslednjih pravic nad objekti:

- ALTER: omogoča spreminjanje lastnosti objektov vključno z nastavitvami načina in oblike shranjevanja,
- DELETE: brisanje z uporabo SQL ukaza delete,
- EXECUTE: izvajanje shranjenih postopkov in paketov,
- INDEX: kreiranje in spreminjanje indeksov nad tabelami,
- INSERT: kreiranje novih zapisov v tabelah podatkovne baze,
- REFERENCES: kreiranje povezav (preko zunanjih ključev) med tabelami,
- SELECT: poizvedovanje po podatkih objektov,
- UPDATE: spreminjanje obstoječih podatkov in
- ALL: vključuje celotni nabor pravic.

Pri dodeljevanju pravic se lahko uporabi tudi opcija WITH GRANT OPTION, ki omogoča da ima uporabnik, kateremu pravice smo dodelili, možnost nadaljnjega dodeljevanja pravic, s čimer administratorja podatkovne baze v veliki meri razbremenimo. Po drugi strani, pa s tem izgubimo centraliziran nadzor nad varnostjo sistema, kar ponavadi ni najbolj priporočljivo.

4.4 Varnost sistema

Če se varnost objektov podatkovne baze ukvarja s tem, kaj lahko uporabniki delajo z obstoječimi objekti podatkovne baze, varnost sistema skrbi za varno izvajanje akcij uporabnika nad sistemom za upravljanje podatkovnih baz kot celoto. To vključuje že opisano prijavo na podatkovno bazo, kreiranje objektov, sistemskega prostora, uporabnikov itd. Oracle vsebuje več ko osemdeset različnih sistemskih pravic (CREATE TABLE, CREATE SESSION, ALTER DATABASE, DROP USER itd.) , ki jih uporabniku dodelimo z ukazom:

GRANT sistemska pravica TO uporabnik;

z dodatno možnostjo WITH ADMIN OPTION, ki omogoča uporabniku nadaljnje dodeljevanje pridobljenih sistemskih pravic.

Posebna vrsta sistemskih pravic so tako imenovane ANY pravice, ki omogočajo uporabnikom izvajanje dodeljenih akcij brez kakršnih koli omejitev. Tako npr. UPDATE ANY TABLE zagotavlja možnost spreminjanja zapisov vseh tabel v podatkovni bazi, tudi če uporabnik nima dodeljenih ustreznih pravic nad posameznimi tabelami. Pri dodeljevanju ANY pravic je torej potrebna velika pazljivost, saj lahko njihova neprimerna uporaba ustvari velike varnostne luknje v sistemu.

4.5 Definiranje vlog

Sistem za upravljanje podatkovnih baz Oracle vse do svoje sedme različice ni omogočal izdelave vlog (skupin pravic). To tedaj niti ni predstavljalo posebnega problema, saj so bile obsežne podatkovne baze z več sto objekti prej izjema kot pravilo. S časom pa se je njihov obseg znatno povečal in administratorji so problem sočasnega dodeljevanja pravic reševali predvsem z uporabo SQL skript, vse dokler podjetje Oracle ni v svojo podatkovno bazo vgradilo sistema vlog.

Preprosto povedano, vloga predstavlja skupino objektnih ali sistemskih pravic in se dodeli uporabnikom podatkovne baze z ukazom:

GRANT vloga TO uporabnik/vloga.

Vloge lahko poleg dodeljevanja uporabnikom dodeljujemo tudi drugi vlogam, pri čemer pa moramo paziti na skupno omejitev števila aktivnih vlog (zapisana v parametrski datoteki INIT.ORA kot parameter MAX_ENABLED_ROLES).

Ob namestitvi podatkovne baze Oracle se samodejno kreira nabor sistemskih vlog med katerimi so najpomembnejše:

- CONNECT: omogoča prijavo na podatkovno bazo,
- RESOURCE: omogoča kreiranje objektov,
- DBA: uporabnikom dodeli vse sistemske pravice nad podatkovno bazo (za administratorje).

4.6 Uporaba pogledov za omejevanje dostopa

Pogled je poseben objekt podatkovne baze, zgrajen na osnovi preproste ali sestavljene SQL poizvedbe (stavek SELECT), ki omogoča prikaz izbrane podmožice podatkov. Pogled lahko definiramo nad zapisi ene oziroma več medsebojno povezanih tabel, kar zagotavlja široko možnost njegove uporabe.

Problem pri dodeljevanju poizvedbenih pravic uporabnikom nad tabelo je v tem, da imajo uporabniki vpogled v vse zapise dotične tabele, brez kakršnih koli omejitev. Ne moremo namreč dodeliti pravice, ki bi zagotavljala, da se v tabeli vidijo le posamezni zapisi (npr. omejitev višine plač na <200.000 SIT). Rešitev se nahaja v kreiranju posebnega pogleda, ki omeji prikaz na podlagi podanih pogojev in določitve poizvedbene pravice nad pogledom namesto nad samo tabelo. Iz navedenega sledi, da pogledi v kombinaciji z ostalimi varnostnimi mehanizmi, predstavljajo močno orodje v rokah administratorjev podatkovne baze in razvijalcev programskih rešitev za preprečevanje nedovoljenih dostopov do shranjenih podatkov.

4.7 Revizija podatkovne baze (auditing)

Revizija podatkovne baze je posebna funkcija, ki zagotavlja administratorjem spremljanje vseh akcij nad podatkovno bazo vključno s podatki o tem, kdo jih je izvedel in kdaj so bile izvedene. Vsekakor gre za močan varnostni mehanizem, ki pa podaja informacije zgolj o aktivnostih, ki so se že zgodile in v tej luči neposredno ne vpliva na varnost podatkov. So pa tovrstne informacije ključnega pomena za ugotavljanje morebitnih kršitev, ki bi v nasprotnem primeru ostale prikrite.

Sprotno izvajanje revizije nad podatkovno bazo pa ima tudi svojo ceno, ki se kaže v večji porabi sistemskih in pomnilniških sredstev. V primeru izvajanja celovite revizije vsaka operacija povzroči vpis novega zapisa v podatkovno bazo, kar lahko bistveno zmanjša njene performanse. Potrebno je torej primerno uravnotežiti naše želje po količini revidiranih podatkov z razpoložljivimi sredstvi in zahtevanim odzivnim časom. Tako se administratorji pogosto odločajo za izvajanje sprotnega beleženja prijav uporabnikov na sistem, medtem ko le redko izvajajo revizijo uporabe SQL stavkov.

Oracle pozna dve vrsti revizije:

- **Revizija na nivoju SQL stavka:** beleži vsak posamezen SQL stavek (SELECT, CREATE, DELETE itd.), pri čemer je možno uvesti omejitev na vrsto stavka in uporabnika. Odvisno od podanega obsega omejitev je lahko količina generiranih informacij dokaj obsežna.

- **Revizija na nivoju objekta:** beleži aktivnosti na specifičnem objektu podatkovne baze, pri čemer se dodatno specificira nabor opazovanih operacij in uporabnikov. Možno je izvajati revizijo na tabelah, pogledih, sekvencah, programskih enotah in prožilcih.

Rezultati revizije se kopičijo v sistemskih tabelah (SYS.AUD$) podatkovne baze ali v tekstovnih datotekah operacijskega sistema. V prvem primeru dostopamo do njih z uporabo ustreznih SQL poizvedb, v drugem pa z uporabo preprostih urejevalnikov besedil. Vsekakor je primernejši prvi način, saj SQL zagotavlja hitro in preprosto filtriranje podatkov, kar je pogoj za uspešen nadzor.

5. Varnost v okolju Lotus Notes

Lotus Notes je integrirano okolje za podporo skupinskemu delu, ki ga je podjetje Lotus razvilo za lastno podporo sodelovanju in komuniciranju številnih prostorsko razpršenih pisarn in predstavništev. Prva leta so ga uporabljali le za lastne potrebe, vendar se je kmalu izkazalo, da je tudi tržno zelo zanimiv izdelek, ki se je pozneje zelo hitro širil, hkrati pa z vedno novimi različicami izboljševal in nadgrajeval. Tudi na našem tržišču je prisoten že dobrih šest let. Nekatere različice klientov so bile prevedene tudi v slovenščino. Zadnja različica je R5 (Release 5), ki zopet prinaša številne novosti.

Okolje Lotus Notes temelji na arhitekturi odjemalec/strežnik. Program *Notes Server* (Notesov strežnik, znan pod imenom **Domino**) teče na računalnikih z najrazličnejšimi operacijskimi sistemi (Windows 95, Windows NT, OS/2, AIX, NetWare, Solaris, HP-UNIX). Njegove funkcije so: je nosilec deljenih dokumentnih zbirk, nosilec poštnih zbirk uporabnikov, izvaja replikacije in izmenjuje pošto z drugimi strežniki, streže uporabnikom, zagotavlja številne načine varovanja sistema in izvaja še dodatna opravila (gateway[14]).

Uporabniki pri svojem delu uporabljajo Notesove delovne postaje. To so osebni računalniki, povezani v omrežje ali samostojni, na katerih teče program *Notes Client*. Ta uporabniku omogoča delo z deljenimi dokumentnimi zbirkami na enem ali več strežnikih, lokalnem disku ter replikacijo s strežnikom. Prav tako kot za strežniški program tudi tukaj obstajajo različice za številne operacijske sisteme. Platformi strežnika in klienta sta neodvisni, kar pomeni, da se lahko na Notes strežnik, ki teče na NetWare operacijskem sistemu, enostavno priključijo delovne postaje s klienti za Windows 95, Windows NT itd. R5 ima tri ločene kliente: Lotus Notes, Lotus Domino Designer in Lotus Domino Administrator.

Iz omenjenih značilnosti izhaja, da je okolje idealno za sodelovanje prostorsko razpršenih uporabnikov z najrazličnejšo infrastrukturo, ki želijo delati na nekih skupnih podatkih. Okolje vsebuje tudi elektronsko pošto, module za usklajevanje urnikov in koledarjev. V zadnjem času so v okolje integrirana tudi orodja za izdelavo in vzdrževanje spletnih mest (web sites) ter spletni brskalnik, kar naj bi vsem kvalitetam Notesa dodalo še prednosti, ki jih prinaša internet. Uporabnikom lahko omogočimo tudi

[14] gateway - prehodnik

dostopanje do Notesovih podatkovnih zbirk preko interneta (Državni zbor ponuja sedem zbirk iz svoje aplikacije širši javnosti).

Varnost je danes eden najpomembnejših faktorjev pri načrtovanju informacijskih sistemov. Kadar želimo svoje delo širiti izven lokalnega omrežja v omrežje internet se tega bolj zavedamo, manj pa kadar delujemo le znotraj LAN[15]-a. Vendar pa moramo pri sodelovanju številnih uporabnikov poskrbeti za ustrezno zaščito tudi znotraj omrežja. Pri načrtovanju okolja Lotus Notes so se tega še kako zavedali in zato razvili številne mehanizme varovanja, od varovanja dostopa do samega strežnika, varovanja dostopa do posameznih zbirk, dokumentov in polj do šifriranja in elektronskega podpisa. Prednost omenjenih mehanizmov je enostavnost, saj so vsi že integrirani v samo okolje, uporabnik pa za svojo varnost potrebuje le njihovo minimalno poznavanje. Seveda pa jih mora zelo dobro poznati sistemski administrator.

Skrb za varnost si delita Notes administrator in razvijalec aplikacije (po [6]).

Administrator mora poskrbeti za:

- varnost samega omrežja, v katerem se nahaja Notes strežnik,
- fizično zaščito strežnika,
- zaščito dostopa uporabnika do strežnika, ki poteka preko uporabe certifikatov, ki so pripeti v identifikacijski datoteki (user.id) vsakega registriranega uporabnika,
- zaščito dostopa do zbirke (ACL[16] postavita skupaj administrator in razvijalec, saj je potrebno poznati tako potrebe ljudi kot zgradbo aplikacije).

Za nižje nivoje dostopa kot so:

- pravice do uporabe kazal in obrazcev,
- pravice za dostop do dokumentov ter
- šifrirana polja

pa poskrbi razvijalec aplikacije.

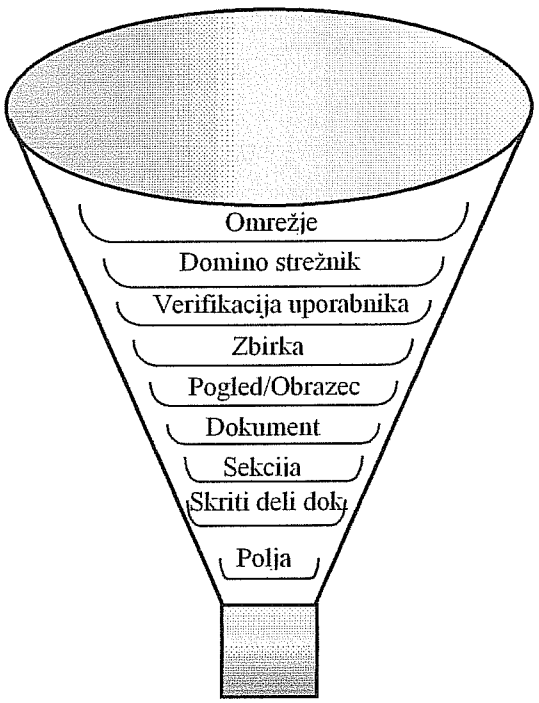
5.1 Overovitev uporabnika

Overovitev strežnik izvaja tako, da primerja svojo server.id datoteko z identifikacijsko datoteko uporabnika (user.id) ali drugega strežnika (server.id). S tem ugotovi ali sta bili certificirani z istim certifikatom ali enem od njegovih potomcev, kar je pogoj za vzpostavitev komunikacije.

[15] LAN - Local Area Network
[16] ACL - Access Control List - lista dostopnih pravic

Identifikacijska datoteka, ki predstavlja "ključ" za vstop v sistem, vsebuje (po [6]):

- **ime uporabnika ali strežnika,**
- **geslo**: je alfanumerični niz dolžine do 63 znakov (priporočeno 8), pri vnosu loči velike in male črke, geslo je možno spremeniti le ob poznavanju starega (velja tudi za administratorja!), vgrajen je tudi mehanizem časovnega zavlačevanja v primeru napačnega vnosa gesla ter protiprevarantska zaščita (pri vnosu se na ekran izpisuje različno število X-ov, da ni možno uganiti dolžine gesla).
- **licenco**: Lotus Notes Mail (samo pošta), Lotus Notes Desktop ali Lotus Notes (potrebna za razvijalce in administratorje),
- **šifrirne ključe**: Lotus Notes pozna sistem enojnega (simetričnega) šifriranja, ki se uporablja za šifriranje na nivoju polja in pri prenosu po omrežju ter sistem dvojnih ključev (javni in privatni). Javni ključ je na voljo vsem in se nahaja v PAB[17], privatni pa je del user.id datoteke. Ter
- **certifikate**: Certifikat pove Notesovemu strežniku, da je uporabnikovo identifikacijo (user.id) resnično oblikoval avtoriziran skrbnik. Za definiranje novega uporabnika je namreč potrebno imeti certifikat (datoteka cert.id), ki se zgradi ob postavitvi strežnika.

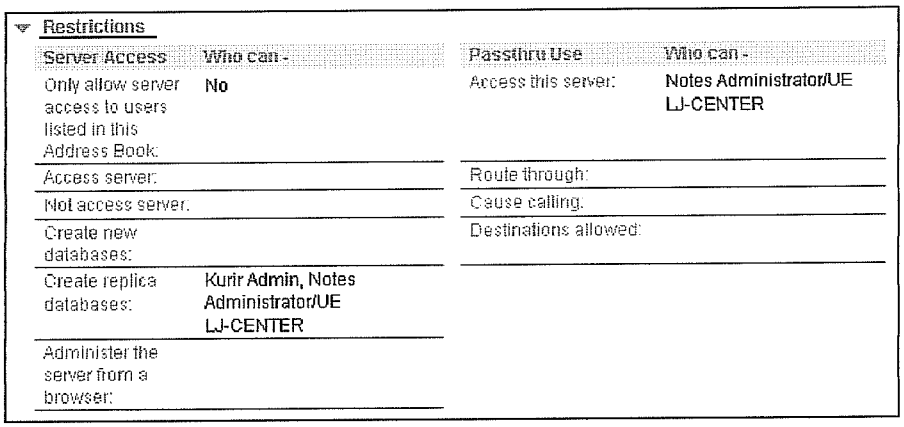


Slika 4: Nivoji varnosti v okolju Lotus Notes, z globino v lijaku narašča omejenost dostopa (po [7])

[17] PAB - Public Address Book

5.2 Zaščita strežnika

Dostop do Domino strežnika je možno ustrezno omejiti z uporabo razdelka Restrictions Server dokumenta (zbirka PAB). Preko ustreznih polj lahko eksplicitno navedemo uporabnike, prepovemo dostop določenim uporabnikom ali skupinam, prepovemo uporabo našega strežnika kot premostitvenega strežnika, omejimo dostop le na uporabnike navedene v PAB-u , dovolimo dostop le iz določenih lokacij itd.



Slika 5: Restrikcije pri dostopu do strežnika (Server dokument)

5.3 Zaščita dostopa do dokumentnih zbirk

Zaščita dostopa do dokumentnih zbirk poteka preko ACL-a (liste dostopnih pravic), ki se definira za vsako zbirko. V listi se navedejo uporabniki, skupine in strežniki, ki imajo določen nivo dostopnih pravic do zbirke in dokumentov v njej.

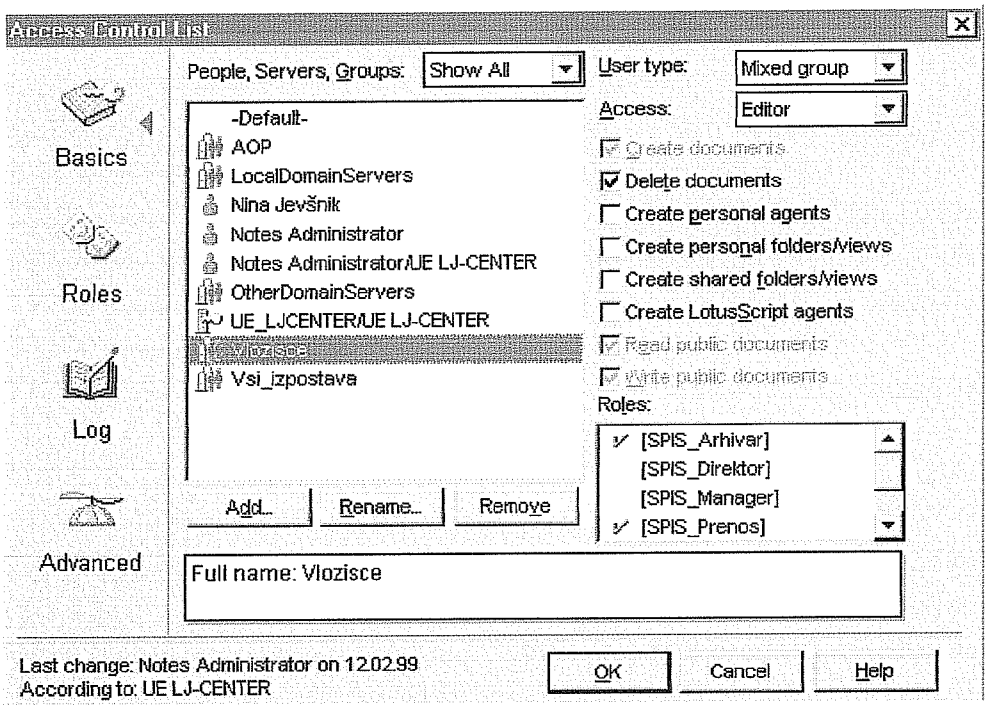
V okolju Lotus Notes se za zaščito uporablja sedem nivojev pravic [5]:

- **Ni dostopa** (No Access): uporabnik ne more odpreti zbirke,
- **Vlagatelj** (Depositor): uporabnik lahko le vnaša nove dokumente, ne more pa prebirati zbirke,
- **Bralec** (Reader): lahko išče in prebira dokumente v zbirki,
- **Avtor** (Author): lahko bere obstoječe, dodaja nove in popravlja svoje dokumente,
- **Urednik** (Editor): lahko bere obstoječe, dodaja nove in popravlja vse dokumente,
- **Načrtovalec zbirke** (Designer): poleg branja, dodajanja in popravljanja vseh dokumentov, lahko popravlja tudi obliko in delovanje zbirke,
- **Upravitelj** (Manager): lahko izvaja vse operacije nad zbirko, edini jo lahko tudi zbriše in nastavlja pravice drugim uporabnikom.

Z omenjenimi nivoji dostopa lahko glede na organizacijsko vlogo uporabnika precizno določimo, katere zbirke lahko pregleduje, v katerih lahko kreira nove dokumente ter določimo uporabnike, ki lahko popravljajo napake (editorji).

Nadalje poteka zaščita še na osnovi **vlog** (Roles), ki jih določi načrtovalec zbirke. Vloga je skupina ljudi, ki potrebujejo enak dostop do nekega obrazca ali pogleda

znotraj zbirke. V dokumentih imamo lahko tudi odseke z omejenim dostopom, pravice se nastavljajo z ustrezno formulo. Varnost na nivoju polja je zagotovljena s njihovim šifriranjem.



Slika 6: Prikaz ACL-a dokumentne zbirke (skupina Vložišče ima editorske pravice)

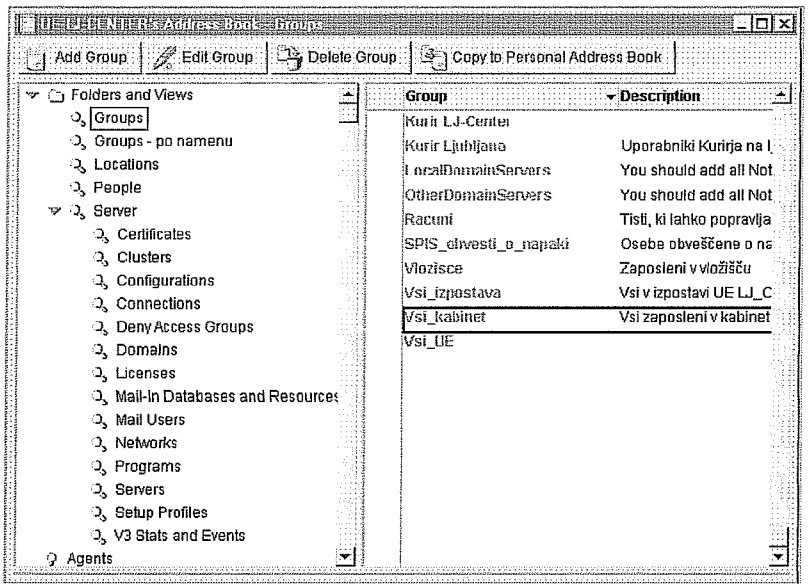
Preden končam z opisom zaščitnih mehanizmov dokumentnih zbirk, naj omenim še mehanizem avtomatskega beleženja kreiranja dokumenta in vseh njegovih sprememb. Na koncu dokumenta so zapisani avtor ter vsi, ki so dokument popravljali ter datum in čas narejene spremembe (za to mora poskrbeti razvijalec aplikacije). Tako so preprečeni nepooblaščeni posegi v dokument, saj ni možno v dokumentu nekaj spremeniti, ne da bi bilo tudi jasno razvidno kdo in kdaj je to storil. Tako je enostavno najti krivce za namerne ali nenamerne napake v dokumentu.

5.4 Sistemske zbirke

So najpomembnejše zbirke v okolju, saj hranijo informacije, ki so potrebne za njegovo pravilno delovanje. Zbirki *Javni imenik* in *Privatni imenik* hranita informacije o vseh uporabnikih v domeni oziroma privatne informacije konkretnega uporabnika. Poleg tega pa Javni imenik vsebuje številne sistemske dokumente od definiranih skupin in njihovih članov, povezovalnih dokumentov, do že omenjenih nastavitev za zaščito strežnika. Aktivnosti vsakega strežnika se zapisujejo v zbirko *Notes log*, ki je neprecenljiva pri razreševanju težav pri njegovem delovanju. *Database Catalog* je zbirka informacij o vseh zbirkah v sistemu, vključno z njihovimi lokacijami in pravili uporabe.

5.4.1 Javni imenik

Javni imenik je najpomembnejša zbirka v Notesovi domeni in nastane ob namestitvi prvega Notesovega strežnika. Njeno ime je names.nsf in nastane na osnovi predloge pubnames.ntf. Vsak strežnik v domeni mora imeti svojo kopijo istega javnega imenika in vse kopije v domeni morajo biti sinhronizirane preko replikacije. Imenik vsebuje informacije, ki omogočajo in zagotavljajo pravilno delovanje okolja. Razumevanje njegove vsebine je ključnega pomena za razumevanje delovanja Notesa, saj je to nekakšen živčni center celotne domene.



Slika 7: Javni imenik strežnika UE LJ-CENTER
(prikaz skupin uporabnikov in strežnikov)

Temeljne informacije, ki se v njem nahajajo, so:

- vsi registrirani uporabniki domene (so zapisani v Person Document-ih),
- definicija skupin uporabnikov za naslavljanje pošte in opredelitev dostopa do skupnih zbirk (v Group Document-ih),
- nastavitve strežnika v omrežju,
- način vzpostavitve komunikacije med strežniki,
- pogostost komuniciranja zaradi pošiljanja pošte ali repliciranja med strežniki,
- izvajanje drugih opravil na strežniku,
- katere druge organizacije lahko oziroma ne smejo dostopati do strežnika,
- način dostavljanja pošte uporabnikom itd.

5.5 Enkripcija v Notes okolju (podatki veljajo za različico 4.x)

V okolje Lotus Notes je vgrajen produkt BSAFE, proizvajalca kritografskih metod in algoritmov RSA Data Security, Inc., ki vsebuje algoritme RC2[18], RC4[19] in RSA[20]. (podtki v poglavju so povzeti po [8], [13] in [14]).

Zaradi izvoznih omejitev Združenih držav v zvezi z enkripcijskimi produkti (maksimalno 40 bitov pri simetričnih ključih) so se pri Lotusu odločili izdelati tri različice okolja in sicer:

- North American: v tej različici se uporabljajo 64-bitni enkripcijski ključi,
- Internetional: pri generiranju 64-bitnega ključa se zgornjih 24 bitov kriptira z javnim ključem U.S. Goverment-a in se shrani v tako imenovani Workfactor Reduction Field. Tako je zadoščeno izvoznim omejitvam, uporabniki pa imajo še vedno 64-bitno enkripcijo,
- French: 40-bitni enkripcijski ključ (povzeto po [8]).

Pri uporabi enkripcije med različicami se uporabi šibkejša enkripcijska shema. V okolju je mogoče kriptirati najrazličnejše elemente: dokumentno zbirko, dokument, določena polja, elektronska sporočila ter sam prenos po lokalnem omrežju (komunikacijska vrata).

5.5.1 Kriptiranje dokumentne zbirke

Enkripcija zbirke poteka z uporabo algoritma javne kriptografije. Največji element, ki ga lahko zaščitimo s kriptiranjem, je celotna dokumenta zbirka. Ta možnost je idealna za potujoče uporabnike, ki na svojih prenosnikih hranijo zaupne podatke. Ker pri lokalnih zbirkah ACL ne velja, je to edini način, da podatke zaščitimo pred nepooblaščenimi dostopi (npr. krajo računalnika, kopiranjem). V tem primeru je dobro tudi, da user.id datoteko, ki vsebuje privati ključ, hranimo ločeno npr. na disketi, drugače je naloga vohuna, da le ugane naše uporabniško geslo in lahko dostopi do podatkov.

Čas enkripcije in dekripcije je odvisen od njene moči (dolžine ključa), zato lahko izbiramo med tremi možnostmi: močna, srednja in šibka. Močnejša kot je enkripcija, daljši čas je potreben za odpiranje kriptirane dokumentne zbirke. Potrebno je najti primerno kombinacijo med hitrostjo in stopnjo varnosti, ki je odvisna od stopnje zaupnosti podatkov in hitrosti procesorja, ki ga uporabljamo.

5.5.2 Kriptiranje dokumenta

Kriptiranje dokumenta je možno le, če je na formi dokumenta enkripcija omogočena na vseh poljih. Kriptirati je potem mogoče:

---

[18] RC2 – Bločna enkripcijska metoda (kriptiramo bloke znakov hkrati), npr. za kriptiranje elektronske pošte. Počasnejši od RC4.

[19] RC4 – Tekoča enkripcijska metoda (kriptira se vsak znak posebej), pogosto se uporablja za enkripcijo komunikacijskih vrat.

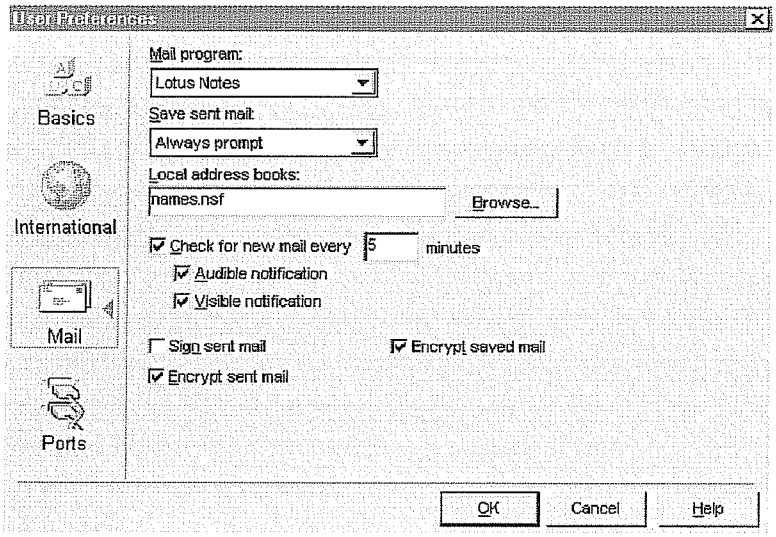
[20] RSA – Bločna enkripcijska metoda, ki uporablja par privatni-javni ključ. Je občutnejše počasnejša od RC2 in RC4 algoritma.

- Dokument (Document Properties): za dokument z zaupno vsebino navedeš uporabnike, ki ga lahko prebirajo (ne pozabi navesti sebe!), tako da jih izbereš iz PAB-a,
- Obrazec (Form Atribute): na nivoju obrazca določiš tajni ključ in z njim so potem kriptirani vsi dokumenti kreirani s pomočjo tega obrazca. Prebirajo jih lahko le tisti, ki imajo tajni ključ.

5.5.3 Kriptiranje polja

V dokumentih se pogosto nahajajo določena zaupna polja npr. plača, starost, EMŠO, številka kreditne kartice, ki jih ne želimo prikazati vsem, ki imajo pravico prebiranja dokumenta. Taka polja zato kriptiramo podobno kot to naredimo za celotni dokument.

5.5.4 Kriptiranje elektronskih sporočil



Slika 8: Kriptiranje izhodne in shranjene pošte

Pri osebni komunikaciji pogosto želimo ohraniti nek pogovor zaupen, prav tako si želimo, da naših sporočil ne bi prebirale nepooblaščene osebe. Zato nam Notes omogoča kriptiranje:

- **izhodne pošte**: Pred pošiljanjem sporočilo kriptiramo z javnim ključem naslovnika ročno ali pa si v osebnih nastavitvah (prikaz na sliki spodaj) izberemo, da bomo vsa sporočila pošiljali kriptirana. Notes kriptira samo telo sporočila ne pa tudi naslovnika in zadeve (Subject), zato moramo paziti, da tam ne navajamo zaupnih podatkov,
- **vhodne pošte**: Tudi sporočila, ki jih dobimo od zunaj in se potem shranijo v naši poštni bazi želimo skriti pred radovednimi očmi, zato je dobro, da si v Person Document-u v PAB-u nastavimo opcijo "Encrypt incoming mail" na Yes.
- **shranjenih sporočil**: Gre za zaščito sporočil, ki jih pošiljamo bi pa jih radi tudi shranili v naši poštni bazi. Kriptiranje shranjenih izhodnih sporočil dosežemo z opcijo "Encrypt saved mail" (slika 8).

5.5.5 Kriptiranje komunikacijskih vrat

Z vklopitvijo opcije "Encrypt network data" v osebnih nastavitvah uporabnika zagotovimo da komunikacija med odjemalcem in strežnikom poteka kriptirana s tajnim ključem (po varnem kanalu). Ob začetku seje strežnik generira tajni simetrični ključ (ki ga po zaključku seje zbriše) in ga kriptiranega z javnim ključem odjemalca pošlje po omrežju. Odjemalec s svojim tajnim ključem odkriptira generirani ključ, s pomočjo katerega poteka nadaljnja komunikacija v okviru seje. Nova seja uporablja nov naključno generiran ključ. Uporaba tovrstnega kriptiranja povprečno upočasni promet v omrežju za 10%, še bolj pa pri uporabi modema, ki že uporablja kompresijo.

Zaključek

S čim širšim pregledom področja računalniške varnosti sem skušala zaobjeti tako zunanje kot tudi notranje dejavnike, ki lahko ogrozijo naš sistem ter podati vsaj nekaj smernic, kako se pred njimi zaščititi. Pri načrtovanju računalniškega sistema moramo vedno upoštevati samo okolje (zakonodajo, pravila, prakse, dejavnost), v katerega bo umeščen ter na osnovi tega postaviti nek formaliziran varnostni model, ki bo služil kot podlaga načrtovalcem tehničnih in organizacijskih aktivnosti, ki bodo zagotovile kar se da varen sistem.

Pri izbiri konkretnih platform, ki jih bomo uporabili, moramo dobro preučiti njihove varnostne mehanizme ter njihovo medsebojno ujemanje. Zavedati se moramo, da moramo z varnimi tehnologijami pričeti pri dnu, na nivoju operacijskega sistema ali že sami strojni opremi ter to nadgrajevati s sistemi za upravljanje s podatkovnimi bazami, sporočilnimi sistemi, internetnimi tehnologijami itd. Nič nam ne pomaga imeti še tako dober sistem za upravljanje s podatkovno bazo z vsemi sodobnimi mehanizmi, če baza na nivoju datotečnega sistema ni ustrezno zaščitena in jo je možno na primer nekam kopirati ali celo zbrisati.

Potem ko uspemo izbrati in medsebojno združiti kopico varnih tehnologij pa je potrebno postaviti tudi pravila obnašanja v sistemu. Pomembno je definirati pravice in pooblastila posameznikov na podlagi opravil, ki jih opravljajo. Pri tem je treba upoštevati princip delitve dolžnosti ter skrbeti, da se s spreminjanjem poslovnega sistema vedno skladno spreminja tudi varnostni model, kar se ob vedno hitrejših spremembah pogosto izkaže za kritično. Prav tako moramo skrbeti za spremljanje vseh nevarnosti, ki nam vsakodnevno grozijo iz okolja, z njimi seznanjati vodstvo ter same uporabnike ter skrbeti za nenehno nadgrajevanje zaščitnih mehanizmov v skladu z novimi tehnološkimi pridobitvami ali odkritimi grožnjami (npr. nov virus). Le tako bomo zagotovili ustrezno stopnjo varnosti v hitro spreminjajočem se svetu tudi v prihodnosti.

Literatura

1. Rita C. Summers: Secure Computing - Threats and Saveguards, McGraw-Hill, 1997
2. T. Mohorič: Podatkovne baze, FER, 1991
3. Becker Rachel in ostali: Oracle Unleashed, Indianapolis, Sams Publishing, 1996
4. E. Aronoff : Oracle8 advanced tuning & administration, Berkeley: McGraw-Hill, 1998
5. B. Swedeen: Lotus Notes 4.5 administrator's guide, San Francisco: Sybex, cop. 1997
6. B. Žitko: Lotus Notes sistemska administracija 4.X, Ljubljana, maj 1997
7. I. Login, A. Bedalič: Varen sistem?, Sistemi za upravljanje z dokumenti in upravljanje z znanjem, DOK_SIS 99, Media.doc, 1999, str. I-51 – I-59
8. http://www.lotus.com/home.nsf/welcome/securityzone
9. http://www.gov.si/cvi/slo/virus/lovelet.htm
10. http://www.gov.si/cvi/slo/virus/novaljubezen.htm
11. http://www.sophos.com
12. Zakon o varstvu osebnih podatkov, Uradni list RS, št. 57-2684/1999
13. http://support.lotus.com/sims2.nsf/eb5fbc0ab175cf0885256560005206cf/c5c59e6c5bd5839a8525661800 64fa5a?OpenDocument (Notes Encrytion)
14. http://support.lotus.com/sims2.nsf/eb5fbc0ab175cf0885256560005206cf/737cf21da1319c15852561cc00727d28?OpenDocument (Using Encryption Between North American and International Notes Releases)