Sigurnost baza podataka

Čaić, Karlo

SVEUČILIŠTE U ZAGREBU
FILOZOFSKI FAKULTET
ODSJEK ZA INFORMACIJSKE I KOMUNIKACIJSKE ZNANOSTI
Ak.god 2016./2017.

Karlo Čaić

Sigurnost baza podataka

Završni rad

Mentor: doc. dr. sc. Vedran Juričić

Zagreb, 2017.

Sadržaj

1. Uvod 4
2. Što su baze podataka? 5
   2.1. Čuvanje integriteta 6
   2.2. Istovremeni pristup 6
3. Zaštita Podataka 7
   3.1. Zaštita podataka kroz povijest 7
4. Prijetnje sigurnosti baza podataka 8
   4.1. Prekomjerna prava korisnika 8
   4.2. SQL Umetanje 8
   4.3. DoS (Denial of Service) 11
      4.3.1. Sprječavanje DoS napada 12
      4.3.2. Razlika između SQL umetanja i DoS napada 13
   4.4. Pogreške u implementaciji sustava 14
   4.5. Aplikacijski napadi 14
5. Načini zaštite baza podataka 15
   5.1. Modeli kontrole pristupa 15
      5.1.1. Model temeljen na mogućnostima 15
      5.1.2. Lista kontrole pristupa 15
   5.2. Metode kontrole pristupa 16
      5.2.1. Diskrecijski model 16
      5.2.2. Obavezni model 16
      5.2.3. Model temeljen na ulogama 16
   5.3. Kriptiranje Podataka 17
   5.4. Tokenizacija 19
   5.5. Praćenje odgovornosti i revizija pristupa 20
   5.6. Fizička zaštita 20
   5.7. Sustavi za otkrivanje i sprječavanje provale 21
   5.8. Rudarenje zaključivanjem 24

5.9. Motrenje aktivnosti baza podataka 24
5.10. Vatrozidi 25
6. Povijest napada 26
7. Zaključak 28
8. Izvori 29

1. Uvod

Moderne baze podataka postoje od 60-ih godina prošlog stoljeća i od tada su promijenile način na koji se odnosimo prema podacima. Usporedno s njima, ljudi su koristili razne sustave kategoriziranja i pohranjivanja podataka. Tek s uvođenje računala se sustav drastično mijenja. Ogromne količine podataka se odjednom mogu spremati, dijeliti i organizirati. Mnogi aspekti modernog života su napredovali ili su nastali zbog postojanja i korištenja baza podataka. Banke lakše mogu spremati podatke o računima, rezervacije za avione je moguće organizirati, i mnoge druge. Kako su se baze podataka razvijale tako su i njihove namjene. Razne moderne tehnologije koriste baze podataka, od strujanja multimedijskih zapisa do pohrane na oblaku, od društvenih mreža do vremenske prognoze. Baze podataka nisu jedine zaslužne za navedene tehnologije ali bez njih su teško zamislive. U ovom radu pokušat ću pojasniti s kojim se sve prijetnjama susreću baze podataka i koji načini obrane postaje.

2. Što su baze podataka?

Baze podataka koriste se kako bi olakšale računalni pristup podacima, učinile ga pouzdanijim, lakšim za pretraživanje te kako bi čitav proces pohranjivanja i pretraživanja podataka u aplikacijama učinile produktivnijim. Baze podataka definiramo kao skup međusobno povezanih podataka koji su pohranjeni u vanjskoj memoriji računala. Ti podaci su istovremeno dostupni raznim korisnicima i aplikacijskim programima. Njih možemo putem zajedničkog softvera mijenjati, čitati, brisati. Pritom korisnici i aplikacije ne moraju poznavati detalje fizičkog prikaza podataka, već se referenciraju na logičku strukturu baze.

Za upravljanje bazom podataka koristimo DBMS sustav (Data Base Management System), odnosno sustavom upravljanja baze podataka (SUBP). SUBP nam omogućuje oblikovanje fizičkog izgleda baze u odnosu na njenu logičku strukturu. Neki od najpoznatijih sustava su: Oracle, DB2, MySQL, Informix, PostgreSQL i SQL server. SUBP-om se koristimo kako bismo obavljali sve operacije s podacima u pojedinoj bazi, on također brine za sigurnost navedenih podataka te automatizira administrativne poslove s podacima u bazi. Svi podaci u bazi su logički organizirani prema jednom od modela podataka. Model podataka je skup pravila koja određuju kako može izgledati logička struktura baze, a odabir modela čini osnovu za koncipiranje, projektiranje i implementiranje baze.

Podaci u bazi su logički organizirani na temelju jednog od postojećih modela podataka. Neki od modela kojima se SUBP može služiti su: relacijski model zasnovan na matematičkim pojmu relacije, prema kojem su podaci prikazani pravokutnim tablicama, mrežni model koji je predočen usmjerenim grafom, hijerarhijski model predočen stablom ili skupom stabala, s hijerarhijskim odnosom nadređeni-podređeni, te objektni model inspiriran objektno-orijentiranim programskim jezicima, u kojima je baza skup trajno pohranjenih objekata koji se sastoje od svojih internih podataka i operacija za rukovanje tim podacima, koji su uređeni prema klasama.[1]

[1] http://jadran.izor.hr/~dadic/EKO/baze-podataka.pdf

2.1. Čuvanje integriteta

Integritet baza podataka podrazumijeva čuvanje ispravnosti i konzistentnosti podataka. To očuvanje postiže se implementacijom metode provjere grešaka, validacijskih procedura i raznih ograničenja. Ograničenja koja postavljamo su ništa drugo nego pravila koja konzistentni podaci moraju zadovoljavati, a ako nisu zadovoljena, SUBP će nam poslati poruku o greški, te neće izvršiti traženu promjenu. Ograničenja koja SUBP može postaviti odnose se na čuvanje integriteta domene, integriteta unutar relacije, te referencijalnog integriteta.

2.2. Istovremeni pristup

S obzirom na to da su baze podataka uglavnom namijenjene većem broju korisnika u isto vrijeme, SUBP mora pažljivo koordinirati istovremeni rad s bazom, kako ne bi došlo do zloupotrebe i neovlaštenog pristupa podacima. Također, svaki korisnik treba imati osjećaj kako sam radi s bazom, iako istovremeno možda tri računala koriste jedan isti podatak.

Rad korisnika u bazi se temelji na pokretanju unaprijed definiranih transakcija. Iako se jedna transakcija čini kao jedna operacija, ona se sastoji od nekoliko njih. Prema tome, stanja između pojedinih operacija podataka dovode do nekonzistentnog stanja u samoj bazi. Kako bi stanje baze ostalo konzistentno i kako bi njen integritet bio očuvan, transakcija koja se provodi mora biti u cijelosti izvršena ili uopće ne smije biti izvršena. Tako svaka započeta, a nedovršena transakcija, mora biti poništena. [2]

[2] http://jadran.izor.hr/~dadic/EKO/baze-podataka.pdf

3. Zaštita Podataka

Kako bi razumjeli važnost pravilne zaštite podataka, potrebno je shvatiti moguće posljedice krađe podataka. U današnjoj sveopćoj povezanosti raznih sustava i sve većoj digitalizaciji sadržaja ogroman broj informacija nalazi se pohranjen u bazama podataka. Neki od važnijih primjera:

1. Zdravstvene podatke kao što su digitalizirani zdravstveni kartoni (e-kartoni) koji sadrže anamnezu pojedinih pacijenata.
2. Financijski podaci kao što su osobni računi u bankama, ali i interni podaci raznih tvrtki o njihovom poslovanju.
3. Strogo povjerljivi podaci poput državnih tajni, informacija o kretanjima i planovima vojske također su na meti potencijalnih napada.

Najraširenija skupina podataka su osobni podaci. Prema istraživanju[3] tvrtke International Data Corporation indirektne i direktne aktivnosti pojedinaca stvorile su otprilike 70% digitalnih podataka 2010. godine. Osobni podaci imaju razne primjene, uključujući marketing, poboljšanje online iskustva, ali se također može koristiti za razna kriminalna djela kao što je krađa identiteta, uhođenje i razni drugi zločini na razini pojedinca i njegove obitelji.

3.1. Zaštita podataka kroz povijest

Otkad podaci postoje, postoje i ljudi koji im žele pristupiti i zloupotrijebiti. Stoga, zaštita podataka, kao i načini kršenja te zaštite konstantno evoluiraju. U prošlosti podaci su se štitili običnim zaključavanjem arhiva koji su sadržavali važne podatke. Danas je manje važna fizička zaštita podataka, a više digitalna zaštita. Stoga, sve više ljudi pokušava dobiti pristup podacima na mreži, što znači da se tehnologija mijenja kako bi pokušala spriječiti razne upade u sustave.

[3] *http://www3.weforum.org/docs/WEF_ITTC_PersonalDataNewAsset_Report_2011.pdf*

4. Prijetnje sigurnosti baza podataka

Korisnici sve više pristupaju bazama podataka preko interneta, što povećava sigurnosne rizike i uvodi nove koji prije nisu postojali. Kako se razvijaju razni načini pristupa podacima i njihova zaštita, tako se istovremeno pronalaze novi načini neovlaštenog korištenja podataka i razbijanja, tj. zaobilaženja zaštite.

4.1. Prekomjerna prava korisnika

Najčešći način prijetnje sigurnosti baze podataka je prekomjerno korištenje prava. Istraživanja su pokazala da su bivši ili sadašnji zaposlenici odgovorni za čak 80% napada na bazu podataka od firme.[4] Neki od ovih slučajeva nastali su iz neznanja ili nemara, ali oni ukazuju na važnost pravovremenog oduzimanja i mijenjanja korisničkih prava. Ova vrsta "napada" se najčešće lako otkrije pogledom na revizijski trag i povećana predostrožnost prilikom pridodavanja prava korisnicima.

4.2. SQL Umetanje

SQL (eng. Structured Query Language, strukturirani jezik upita) koristi se za komunikaciju s bazom podataka. Prema ANSI i ISO institutima za standarde to je standardni jezik za relacijske sisteme baza podataka[5]. Koriste se za ažuriranje podataka u bazi ili za dohvaćanje podataka iz nje. Neki od uobičajenih sustava koji koriste SQL su: Oracle, Sybase, Microsoft SQL server, Access, Ingres, itd. Iako većina SUBP-a koristi standardni SQL, većina njih također koristi specifične nadogradnje koje se uporabljuju samo u tom sustavu.

Napad korištenjem SQL-a vrši se tako da neovlašteni korisnik umetne neautorizirani izraz u SQL kod određenog dijela baze podataka[6]. Tako se ciljaju ulazni parametri i mijenja se originalna namjena aplikacije ili funkcije baze. Glavni cilj SQL umetanja je pristup bazi podataka. Također je moguće samo prikupiti podatke, zaobići korisničku prijavu, izmjenjivati podatke unutar baze bez revizije pristupa, izmjenjivati razine prava korisnika i brisati podatke.

---

[4] https://pdfs.semanticscholar.org/9c9a/d8150e8ddf7427ea2a53482ef106bc324e72.pdf

[5] http://www.sqlcourse.com/intro.html

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

Postoje dva načina SQL umetanja. Prvi od njih je SQL umetanje koji je namijenjen na standardne SUBP. Uglavnom se obavlja umetanjem neautoriziranih tvrdnji u polje za unos.

Drugi način, koristeći NoSQL umetanje namijenjen je platformama koje barataju „velikim podacima[7]“. Bez obzira na metodu i dalje je glavni cilj pristup i potpuna kontrola nad bazom podataka.

Napadač SQL umetanjem će manipulirati standardni SQL upit kako bi eksploatirao nepotvrđene ulazne ranjivosti u bazi podataka. Najlakši način prikazivanja principa SQL umetanje je preko primjera:[8]

Uobičajen SQL upit za bazu podataka nekog elektronskog dućana može biti sljedeći:

```
SELECT ItemName, ItemDescription

FROM Item

WHERE ItemNumber = ItemNumber
```

Korisnik može manipulirati upitom tako da u adresno polje upiše vlastiti upit koji će aplikacija pretvoriti u SQL upit i poslati bazi podataka. Dodavanjem naredbe „itemid=999“ u adresu „http://www.estore.com/items/items.asp?itemid=999“ stvorit će sljedeći SQL upit prema bazi:

```
SELECT ItemName, ItemDescription

FROM Item
```

[7]Veliki podaci, tj. Big Data, su nakupine podataka koji su preveliki i prekompleksni da bi ih procesuirali tradicionalne relacijske baze podataka

https://www.merriam-webster.com/dictionary/big%20data

[8] https://www.incapsula.com/web-application-security/sql-injection.html

```
WHERE ItemNumber = 999
```

Promjenom upita napadač je pozvao iz baze podatke o stavci pod brojem 999. Stavka 999 može biti dostupna svima, ali ne znači da su i ostale stavke također. Napadač može pridodati upitu još neke parametre kako bi izvukao više podataka. Ubacivanjem izraza „1=1“ u upit dobijemo sljedeće:

```
SELECT ItemName, ItemDescription

FROM Items

WHERE ItemNumber = 999 OR 1=1
```

A kako je izraz 1=1 uvijek istinit, upit će vratiti sva imena i opise iz baze podataka, čak i one koje nisu inače dostupni. Koristeći pogrešno filtrirane znakove napadač može manipulirati SQL naredbama. Recimo dodajući točku-zarez kako bi odvojio dva polja upita: „http://www.estore.com/items/iteams.asp?itemid=999; DROP TABLE Users“, Što bi poslalo sljedeći SQL upit prema bazi:

```
SELECT ItemName, ItemDescription

FROM Items

WHERE ItemNumber = 999; DROP TABLE USERS
```

Rezultat toga bi bilo brisanje podataka svih korisnika baze. Koristeći izraz UNION SELECT mogu se kombinirati dva nepovezana SELECT upita kako bi se prikupili podaci iz dvije različite tablice. Primjerice, „http://www.estore.com/items/items.asp?itemid=999 UNION SELECT user-name, password FROM USERS“ bi stvorilo upit:

```
SELECT ItemName, ItemDescription

FROM Items

WHERE ItemID = '999' UNION SELECT Username, Password FROM Users;
```

Ovaj upit bi povukao podatke o stavci 999 ali i imena i lozinke svih korisnika u bazi.

4.3. DoS (Denial of Service)

Denial of Service, tj. napad uskraćivanja usluga je jedan od najčešćih vrsta napada na baze podataka. Cilj mu je učiniti bazu podataka nedostupnom korisnicima. Česti način postizanja toga cilja je zasićenje mete napada s vanjskim zahtjevima kako bi preopteretili sustav i onemogućili normalno korištenje. Zbog pokušaja sustava da odgovori na sve zahtjeve, dolazi do značajnog usporavanja rada sustava, a potencijalno čak i potpunog prekida rada

DoS napadi na baze podataka mogu se raspodijeliti u nekoliko vrsta[9]

1. Zlouporaba funkcija
2. Kompleksni upiti
3. Greške u kodu
4. Aplikacijski napad

Jedan od najčešćih napada jest zlouporaba funkcija, a radi tako da napadač natjera neku funkciju baze podataka da učini nešto što ne smije. Relacijske baze podataka sastoje se od mnogo različitih procesa i prestanak rada čak i jednog od njih može uzrokovati pad cijelog sistema. Problem kod obrane od ovakve vrste napada jest u tome što može iskoristiti nedostatak u raznim dijelovima SUBP-a, što otežava prijevremeni pronalazak potencijalnih problema. Primjer ovakvog

[9] https://securosis.com/assets/library/reports/Database_DoS.pdf

napada je slanje krivo oblikovanog RPC (daljinskog izvršenja naredbe), koji sintaksni analizator sustava nije u stanju pročitati, što ga natjera da stane. Na isti način mogu se slati XML, TDS i SNMP upiti. SQL umetanje je također vrsta zlorabljenja funkcije, ali nije vrsta DoS napada jer je cilj SQL umetanja da preuzme kontrolu nad bazom podataka, a ne da ju samo „sruši“.

1. Kompleksni upiti natjeraju bazu podataka da iskoristi previše radne snage na njihovo rješavanje, te kao rezultat toga prestane s radom. Takvi napadi često ciljaju potrošnju radne memorije, procesorske snage i slično.
2. Izračunati stupci i pogledi su virtualni rezultati upita, najčešće spremljeni u radnu memoriju. Ako neki upit potražuje veću količinu podataka iz base, rezultat pretrage zauzet će veći dio radne memorije, isto vrijedi i ako je upit kompleksan što bi uzrokovalo prekomjernu potrošnju procesorske snage.
3. Ugniježđeni upiti i rekurzije zatraže od SUBP-a da sam sebe prizove prilikom izvođenja nekog upita. Sustav će izvršavati upit kontinuirano dok ne prestane s radom.
4. Prilikom operator IN upita napadač zatraži od SUBP-a da pronađe određenu varijablu unutar nekog dijela baze. Sama po sebi ova operacija je vrlo spora, čak i ako pretražuje malen raspon podataka. Napadač može pretražiti poveću količinu podataka, tražeći varijablu koja ne postoji, a sustav će polako trošiti resurse uzaludnom pretragom.
5. Operator JOIN spaja redove iz dvije ili više tablica, a kartezijski produkt je suma svih redova svih tablica određenih FROM rečenicom. Ovakav upit može potencijalno kao rezultat dati jako velike količine podataka, čak i veće od same baze.

Korisnički upiti. Ako korisnik sam upiše svoj upit ima potencijal napraviti što god poželi, čak i upisati jednu od prije opisanih upita.

4.3.1. Sprječavanje DoS napada

Ne postoji jedan konkretan način kako spriječiti DoS napade ali postoje mnogi načini obrane[10]. Neki detektiraju potencijalne napade, a drugi ih sprječavaju. Jedan od načina je brisanje nepotrebnih funkcija baze podataka, tj. funkcije koje se ne koriste u namijenjenoj uporabi pojedine baze. Većina SUBP-a sadrži velik broj značajki koje nisu potrebne svakom korisniku. Smanjenjem

[10] https://securosis.com/assets/library/reports/Database_DoS.pdf

broja protokola, usluga i komponenata koji nisu potrebni ili se uopće ne koriste, povećava se sigurnost. Manji broj elemenata jednostavno znači da je onima koji žele napasti bazu teže pronaći "rupu" u sigurnosti.

Uobičajen postupak administratora baza podataka je uvođenje vremenskih ograničenja na razne upite, smanjivanje broja mogućih upita i stavljanja hardverskog limita na upite. Iako neće zasigurno zaustaviti napadača, uvelike će im otežati napad.

Često zanemarivan ili kasno provođen način zaštite je ažuriranje softvera u što bržem roku. Nažalost, ažuriranja često kasne ili ih je zbog načina uporabe baze podataka teško implementirati na vrijeme.

4.3.2. Razlika između SQL umetanja i DoS napada

SQL umetanje i DoS napadi su dva najčešća vanjska napada na baze podataka. Razlika je u tome što SQL umetanje dozvoljava pristup bazi podataka ali je zato puno teže izvršiti takvu vrstu napada, dok DoS napad može izvršiti osoba s puno manjim znanjem o sigurnosnim sustavima i eksploataciji istih. Iako je šteta DoS napada puno manja od SQL napada, koristi se često baš zbog svoje jednostavnosti i lakoće izvedbe. Mala je vjerojatnost gubitka podataka prilikom takvog napada, ali nedostatak usluge potencijalno je vrlo skupa situacija za tvrtke.

Važno je i napomenuti odnos između DoS i DDoS napada. tj. distribuiranih napada uskraćivanja usluge koji se razlikuju samo po broju napadača. Obični napadi imaju samo jednog, a distribuirani napadi više napadača. Zbog većeg broja napadača takvi napadi su učestaliji i efikasniji ali se baziraju na istim ili sličnim principima. Oba sustava se isto tako često koriste sakrivajući identitet (izvorno računalo) napadača koristeći jedan ili više upravljačkih računala koja kontroliraju veći broj drugih računala i tako zameću svoj IP trag. Jedna od naprednijih vrsta napada koristeći više napadača je DRDoS (distribuirani reflektirani napad uskraćivanja usluge) koji ne samo koristi takozvana računala robote, nego i drugi niz reflektivnih računala robova koja primaju naredbe od prvog niza robova direktno povezanih s napadačima i tek onda obavljaju svoj napad. Na ovaj način se količina upita eksponencijalno povećava, a i samim time efikasnost.

4.4. Pogreške u implementaciji sustava

Greške postoje u svakom kodu i napadači ih mogu iskoristiti kako bi ugasili baze podataka. Dovoljan je jedan propust i napadač može s udaljene lokacije pristupiti bazi podataka bez korisničke prijave i poslati jedan upit koji će potpuno ugasiti bazu podataka. Primjer bi bio pretek međuspremnika XML baze podataka na Oracle9 SUBP-u, kao što se dogodilo 2003. i opisano je u Oraclovom sigurnosnom izvješću 58[11]. Kako izlaze nove usluge i softverski napredci, tako napadači stalno traže nove propuste ili koriste starije tehnike na novim inačicama. Zbog kompleksnosti sustava teško je predvidjeti ovakve napade prilikom stvaranja SUBP-a. Nakon otkrivanja greške, vijest se vrlo brzo proširi po internetu i veliki broj ljudi je u stanju iskoristiti tu grešku puno prije nego proizvođač stigne distribuirati sigurnosnu zakrpu.

4.5. Aplikacijski napadi

Aplikacijski napadi[12] su najjednostavniji jer iskorištavaju samu aplikaciju koja koristi bazu podataka. Baze podataka moraju otkriti svoje funkcije aplikacijama koje im pristupaju kako bi iste mogle koristiti spremljene podatke. Ako je aplikacija nedovoljno zaštitila tu komunikaciju moguće je provesti napad. Moguće je da neka online trgovina nema postavljenu dovoljnu zaštitu na razini svoje usluge elektronske košarice i napadač može preopteretiti košaricu stavljajući u nju poprilično količinu proizvoda te ju osvježavati, što prisiljava aplikaciju da pošalje upit u bazu podataka za svaki proizvod koji se nalazi u košari. Nekoliko simultano otvorenih košara može slati tisuće odvojenih upita prema bazi svake sekunde.

11 https://www.oracle.com/technetwork/topics/security/2003alert58-128165.pdf

12 https://securosis.com/assets/library/reports/Database_DoS.pdf

5. Načini zaštite baza podataka

Svaki administrator zadužen je za njenu adekvatnu zaštitu. Implementacija zaštite baze treba biti ustanovljena od samog početka i integrirana u svaki dio baze. Jedna od najosnovnijih metoda osiguranja baza podataka je kontrola pristupa. Kontrola pristupa podrazumijeva ograničavanje slobode korisnika pri pristupu i uređivanju sustava i datoteka. Svrha kontrole pristupa podataka je umanjivanje štete koju potencijalno može napraviti neovlašteni korisnik. Naravno, kontrola pristupa se može zaobići ili prevariti.

5.1. Modeli kontrole pristupa

Razni dijelovi mehanizma kontrole pristupa nalaze se u nekoliko slojeva računalnih sustava. Mehanizmi viših slojeva su izraženiji ali i podložniji napadima. Neki od razloga su kompleksnost sustava i programerske pogreškse. Moderni sustavi dijele kontrolu pristupa u dvije grupe.[13]

5.1.1. Model temeljen na mogućnostima

Model kontrole pristupa u kojoj korisnik koristi značku (token) kako bi pristupio željenom dokumentu. Korisnik ne mora nužno imati potpuni pristup tom dokumentu, nego mu je pristup ograničen sukladno s ovlastima pridodanim njegovoj znački.

Mana ovog modela je što ovisi o međuljudskom povjerenju. Ako dođe do krađe značke i ako se na vrijeme ne poduzmu mjere poništavanja mogućnosti značke, može doći do neželjenog mijenjanja podataka ili korištenja tih podataka u razne svrhe.

5.1.2. Lista kontrole pristupa

(eng. Access Control List - ACL)

Lista kontrole pristupa je, kao što se može iščitati iz imena, lista dozvola, tj. ovlasti koji su pridodani svakom korisniku ili grupi korisnika, te se nalaze u opisu svakog digitalnog objekta. Objekti u računalu, kao što su pojedini podaci i datoteke imaju u svojim svojstvima zapisane

[13] http://www.cis.hr/www.edicija/LinkedDocuments/CCERT-PUBDOC-2008-02-218.pdf

ovlasti za različite liste kontrole pristupa. Korisnik pri prijavi dobiva predodređene ovlasti koje se provjeravaju prilikom pokušaja manipulacije podataka.

5.2. Metode kontrole pristupa

Osim prijašnje podjele, kontrola pristupa, dijeli se i prema metodi implementacije. Svaka metoda ima svoje prednosti i mane, a odabir metode ovisi o potrebama i o vrstama napada na koje je sustav najpodložniji. Diskrecijski i obavezni modeli mogu koristiti listu kontrole pristupa.

5.2.1. Diskrecijski model

(eng. Discretional Model – DAC)

U diskrecijskom modelu kontrole pristupa, vlasnik datoteke po vlastitom nahođenju dodjeljuje prava pristupa ostalim korisnicima sustava. Liste kontrole pristupa su najuobičajeniji primjer diskrecijskog modela. Većina operativnih sustava koristi ovu metodu (Windows, iOS, većina Linux sustava).

5.2.2. Obavezni model

(eng. Mandatory Access Control – MAC)

Korisnici sustava u obaveznom modelu nemaju mogućnost mijenjanja ovlasti datoteka. Operativni sustav unaprijed ima zadane pravila za pridodavanje ovlasti određenim dijelovima sustava i određenim korisnicima. Svaki korisnik i podatak dobiju sigurnosnu oznaku koja određuje razinu njihove ovlasti unutar sustava. Nitko od korisnika nije u mogućnosti promijeniti ta unaprijed zadana pravila, čak i ako su oni tvorci ili vlasnici datoteka. Pravila upravljanja datoteka postavlja administrator sustava, a operativni sustav podržava takav način zaštite. Ovakvi sustavi koriste se za zaštitu povjerljivih baza podataka zbog svoje rigidnosti i strogoće. Glavnu uporabu nalaze u vojnim sustavima, ali i najsigurnijim sustavima korporacija, vlada i banaka.

5.2.3. Model temeljen na ulogama

(eng. Role Based Access Control – RBAC)

Najnoviji model koji nastao spajanjem diskrecijskog i obaveznog modela naziva se model temeljen na ulogama. Niti jedan korisnik nema osobno dodijeljena prava, nego su prava dodijeljena po ulogama koje su pridodane svim korisnicima od strane administratora sustava. Jedan korisnik

može biti član više grupa, ali nema mogućnost promijene svoje grupe ili prava unutar nje. Na ovaj način je lakše raspodijeliti prava ako pristup nekom sustavu ima veći broj ljudi koji su hijerarhijski raspoređeni. Uz grupe, u RBAC modelu postoje i uloge koje se mogu dodijeliti pojedincima. Uloge mogu imati istu ili vrlo sličnu razinu pristupa kao grupe, ali se dodjeljuju na individualnoj bazi. Primjer toga bi bila osoba kojoj treba pridodati prava grupe koja je hijerarhijski iznad, ali samo na određeno vrijeme. Ulozi se mogu stoga mijenjati vrlo brzo razine prava i pristupa, te neće utjecati na ostale korisnike u grupama.

5.3. Kriptiranje Podataka

Enkripcija baza podataka je uporaba enkripcijskih tehnika u svrhu pretvaranja nešifrirane baze podataka u (djelomično) šifriranu bazu podataka, čineći ju nečitljivom svima osim onih koji posjeduju enkripcijski ključ. [14]

Kriptiranje baza podataka se vrši na podacima koji se trenutno ne koriste i podataka u tranzitu. Kada podaci miruju u bazi oni su kriptirani, a kada ih povučemo na čitanje ili izmjenjivanje sustav ih dekriptira. Korištenje podataka s baza podataka je problem jedino ako je računalo spojeno na internet zbog prijetnje napada nakon što sustav dekriptira podatke prilikom korištenja baze podataka. Kako bi se izbjegla krađa podataka u tranzitu koristi se SSL (Secure Socket Layer) protokol. SSL protokol kriptira podatke prilikom njihovog slanja preko mreže. Ova zaštita stvara probleme jer zahtjeva povećanu uporabu sistemskih resursa što usporava rad i snižava performanse.

Samo kriptiranje može se vršiti na cijelom SUBP, na cijelom sadržaju baze podataka ili na nekom dijelu sadržaja, kao što je stupac ili tablica. To se može postići koristeći funkcije samog SUBP-a ili neke druge aplikacije. Korištenje tokena se također koristi kao alternativa ili dodatak ovoj zaštiti.

Dvije osnovne vrste zaštite pomoću enkripcije su prozirna, tj. eksterna enkripcija i korisnička, tj. podatkovna enkripcija.[15]

[14] https://hal.archives-ouvertes.fr/file/index/docid/623915/filename/BOUGA_B6_ENC_CRYPT_2009.pdf

[15] https://securosis.com/assets/library/reports/Securosis_Understanding_DBEncryption.V_.1_.pdf

1. Prozirna/Eksterna naznačuje enkripciju cijele baze podataka koju omogućuju moduli same baze. Moguće je enkriptirati i pojedine dijelove na ovaj način, ali se uglavnom primjenjuje na cijeloj bazi. Naziva se prozirna enkripcija jer nije vidljiva korisnicima i aplikacijama i ne zahtijeva nikakvu prilagodbu kod aplikacija. Razlog uporabe je sprječavanje izlaganja podataka zbog gubitka fizičkog medija ili ugrožavanja podataka u bazi. Prozirna enkripcija štiti podatke od korisnika bez pravilnih vjerodajnica, ali ne od autoriziranih korisnika. Utjecaj korištenja prozirne enkripcije na performanse sustava je otprilike 3-5% u odnosu na sustav bez enkripcije[16].
2. Korisnička/Podatkovna enkripcija se odnosi na enkripciju stupaca, tablica ili čak pojedinih elemenata unutar istih. Cilj je spriječiti otkrivanje dijelova baze podataka autoriziranim korisnicima ili primorati odvajanje zaduženja korisnika unutar baze. Mana ovakve enkripcije je u tome što zahtjeva mijenjanje kôda baze i/ili aplikacije. Nastoji se samo enkriptirati najvažnije podatke kako bi se smanjio utjecaj na učinak sustava i količinu potrebnih promjena. Usporedno s prozirnom enkripcijom, jednostavna podatkovna enkripcija samo jednog stupca koristi oko 20% više procesorske snage.[17]

Prozirna enkripcija koristi različite metode i tehnologije.

1. Lokalna enkripcija objekata baze podataka - Sustavi upravljanja bazama podataka kao što su Oracle, Sybase, Microsoft SQL Server i IBM DB2 imaju ukomponirane mogućnosti enkriptiranja lokalnih objekata u bazi (tablica i slično) ili pohranjenih podataka (datoteka) koristeći lokalne funkcije. Pošto sami SUBP sadrži sve potrebno za ovu vrstu enkripcije, nikakva vrsta modifikacije nije potrebna. Ključevi se nalaze unutar baze podataka ali ih je moguće i spremiti na vanjske sustave za upravljanje ključevima ako odabrana platforma to podržava.
2. Vanjska enkripcija - Podaci unutar baze podataka su ovom metodom enkriptirani koristeći vanjski alat. Ova metoda također štiti podatke prilikom njihovog micanja, kopiranja ili izrađivanja sigurnosne kopije. Ključevi se uglavnom nalaze van samog poslužitelja i

[16]https://technet.microsoft.com/en-us/library/cc278098(v=sql.100).aspx#_Toc189384679

[17] (https://info.townsendsecurity.com/sql-server-tde-vs-cell-level-encryption-a-brief-comparison

pristup ne bi trebao biti omogućen korisnicima, stoga pokušaj napada na bazu podataka neće uspjeti dekriptirati same podatke. Neki enkripcijski alati mogu onemogućiti aplikacijama pristup podacima, što znači da samo lokalni procesi baze podataka imaju pristup.

3. Enkripcija medija - Ova metoda uključuje enkripciju cijelog diska ili skladišne mreže. Ova metoda se koristi u bazama podataka koje zahtijevaju visoke performanse i kod kojih je glavna opasnost fizička krađa diskova s pohranjenim podacima.

Korisnička tj. podatkovna enkripcija slična je nekim drugim mjerama zaštite, kao što je kontrola pristupa po tome što se bazira na davanju pristupa, tj. ključeva zasebno od korisnika do korisnika. Ova metoda u biti samo nadodaje sigurnost na postojeći sustav kontrole pristupa u slučaju ugroženosti korisničkog računa. Enkripcija uz to dopušta veću preciznost nego metode pristupa, osiguravajući podatke čak i ako se pomaknu iz dijela baza kojem je pristup ograničen u dio baze dostupan svima. Pomoću takve enkripcije moguće je i zaobići problem administratora baza podataka koji imaju zbog naravi pozicije imaju pristup svim podacima u bazi, ali radi sigurnosti ne bi trebali imati mogućnost njihovog čitanja.

5.4. Tokenizacija

Tokenizacija djeluje na principu zamjene nekog podatka iz baze sa surogatom podatkom. [18]Taj zamjenski podatak se zove token. On sam po sebi ne sadrži važne informacije, nego ih zamjenjuje koristeći jednu vrijednost za neku drugu. U svrhu olakšanja poslovanja tokeni često imaju istu veličinu ili duljinu kao podaci koje zamjenjuju. Koristi se kao efektivnija zamjena kriptiranju jer nije potrebno kriptirati sve podatke na svim bazama nego samo na jednoj, glavnoj bazi, a u ostalim bazama jednostavno postaviti tokene koji nemaju vrijednost bez glavne baze. Na ovaj način se glavna baza podataka može zaštiti na više načina, a ostale baze nemaju tu potrebu jer ne sadrže vitalne podatke. Pošto bez korištenja glavne baze u kojoj su zapisane poveznice

[18] https://townsendsecurity.com/sites/default/files/Encryption_vs_Tokenization.pdf

između tokena i njihovih podataka nije moguće dekriptirati token (jer ne predstavlja šifrirane podatke već neki nasumičan slijed), siguran je od takvih napada.

5.5. Praćenje odgovornosti i revizija pristupa

Revizija pristupa uključuje zapisivanje i praćenje promjena nastalih na bazama podataka prilikom korištenja.[19] Praćenje odgovornosti je proces održavanja revizijskog traga nad korisničkim radnjama u sustavu. Ove provjere i zapisivanja se provode kako bi se osigurao integritet podataka u sustavu. Svaki uspješni i neuspješni pristup sustavu je zapisan, te sve radnje korisnika imaju pridodano točno vrijeme i datum koji je zabilježen u revizijskom tragu. Sustavni administrator jedini ima pristup revizijskom tragu te može iskoristiti te informacije kako bi pronašao neovlašteno ili potencijalno štetnu manipulaciju podataka unutar sustava.

5.6. Fizička zaštita

Dio zaštite baza podataka koji se ne smije zanemariti je i sam fizički pristup.[20] Ovakav način napada zaobilazi većinu softverskih sigurnosnih postavki i rješenja. Potrebno je zaštiti bazu podataka od intrinzičnih rizika i krađe. Same baze nalaze se na računalima poslužiteljima i u pogonu su uglavnom cijeli dan svaki dan, stoga je potrebno održavati temperaturu prostorije što nižom kako visoka temperatura ne bi izazvala kvar. Isto tako, kao što je pristup samoj bazi kroz SUBP ograničen, tako je i pristup fizičkoj lokaciji ograničen samo ovlaštenim ljudima pružatelja usluge. Treba uzeti u obzir da je oko 9% slučajeva ukradenih podataka iz baza počinio netko iz same tvrtke, a vanjski napadači (hakeri) čine 7%. [21]

[19] https://pdfs.semanticscholar.org/9c9a/d8150e8ddf7427ea2a53482ef106bc324e72.pdf

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

[21] Di Justo, P. "Your secret is out: data breaches cost companies billions each year." // Wired USA, Veljača 2007. Str. 50.

5.7. Sustavi za otkrivanje i sprječavanje provale

(eng. IDS - Intrusion Detection System i IPS - Intrusion Prevention System)

Provala u sustav se dogodi kada neki napadač pokuša dobiti pristup ili poremeti normalan rad sustava, skoro pa uvijek s namjerom da učini štetu. IDS sustavi služe kao alarmi protiv lopova i aktiviraju uzbunu u slučaju otkrivanja nekog narušavanja sigurnosti sustava.[22] Manifestacija tog alarma ovisi o administratoru sustava i može biti sve od elektroničke do SMS poruke. Zajedno sa sustavom sprječavanja provale (sada IDPS - Intrusion Detection/Prevention System), sustav otkrivanja je među prvim linijama obrane i upozorenja prilikom napada. Neki napadi krenu s organiziranim i temeljitim ispitivanjem baze, mreže i ugrađenih obrana. Ta vrsta ispitivanja se zove "Zveckanje brave" (eng. doorknob rattling) i provodi se koristeći tehniku "otisaka stopala" (eng. footprinting), koja uključuje prikupljanje podataka vezanim uz ciljanu bazu podataka i njenim sustavima zaštite, te onda koristeći tehniku "otiska prsta" skeniranjem same baze i povezane mreže i SUBP-a. Sustav otkrivanja provale je napravljen kako bi otkrio te prvotne korake i prijavio ih administratoru, što dozvoljava da se poduzmu mjere predostrožnosti i umanje ili potpuno izbjegnu potencijalni gubitci od budućeg napada. Uz to nude i razne druge mogućnosti kao što su analize i statistike aktivnosti i korištenja sustava i dnevnik svih radnji.

Dijelovi IDPS sustava su:

1. Senzor - Nadgleda i analizira aktivnosti mreža.
2. Poslužitelj za upravljanje - Centraliziran uređaj koji prima podatke od senzora i upravlja njima. Može obavljati analizu na podacima koje primi od senzora i identificirati događaje koje su zabilježili. Korelacijom spaja odvojeno prikupljene podatke od više senzora (npr. Dva senzora su otkrila neki događaj od iste IP adrese).
3. Baza podataka - Sami IDPS sustavi spremaju podatke prikupljene senzorima i analizom poslužitelja za upravljanje na bazu podataka.
4. Konzola - To je program koji služi kao sučelje administratorima IDPS sustava. Neke konzole se koriste samo za konfiguraciju senzora i primjenjivanje ažuriranja, dok neke

[22] http://ecinetworks.com/wp-content/uploads/bsk-files-manager/86_SP800-94.pdf

druge samo za analizu i praćenje senzora. Naravno, neke konzole mogu raditi i administrativni i nadgledavalački dio posla.

Dijelovi IDPS sustava mogu biti povezani koristeći standardnu mrežu ili koristeći odvojenu mrežu koja služi izričito samo za upravljanje sigurnosnim softverom zvanu upravljačka mreža. Korištenjem upravljačke mreže svaki senzorski domaćin ima dodatno mrežno sučelje zvano upravljačko sučelje koje se spaja na upravljačku mrežu. Također, svaki senzor nije u mogućnosti slati bilo kakve podatke iz svog mrežnog sučelja prema bilo kojem drugom vanjskom mrežnom sučelju. Poslužitelj za upravljanje, baza podataka i konzole su spojene samo na upravljačku mrežu. Ovakva vrsta mrežnog ustroja izolira upravljačku mrežu od ostalih mreža. Prednost ovakvog sustava je prikrivanje postojanja i vrste IDPS-a od napadača, zaštita IDPS-a od napada i osiguravanje dovoljne količine mrežne propusnosti kako bi sustav radio u nepovoljnim uvjetima i tijekom napada. Nedostatci ovakvog umreženja su cijena dodatne mrežne opreme i potreba za korištenjem dodatnih računala za upravljanje IDPS mrežama.

IDPS je u mogućnosti zaustaviti napad i promijeniti sigurnosno okruženje. Zaustavljanje napada uključuje prekid veze ili sesije između klijenta i SUBP-a, blokiranje korisničkog računa povezanog uz napad, blokiranje IP adrese i konačno, potpuna blokada ciljane baze podataka. U slučaju mijenjanja sigurnosnog okruženja, IDPS je u mogućnosti utjecati na mrežne uređaje kao što su vatrozid, usmjerivač i preklopnik kako bi blokirao pristup napadaču. Kao i kod vatrozida, IDSP može imati pogrešno otkrivanje što usporava uobičajenu uporabu baze podataka.

Sami način detektiranja obavlja se koristeći tri različite metodologije. Neki sustavi koriste samo jednu, a neki više njih.

1. Otkrivanje pomoću potpisa - Koristeći uzorke zvane potpisi, IDPS pokušava pronaći prijetnju. Ova metoda utvrđivanja zlonamjerne efektivna je kod otprije poznatih prijetnji, ali nimalo efektivna protiv novih ili dovoljno izmijenjenih starih prijetnji. Ovo je najosnovnija metoda otkrivanja jer samo uspoređuje ulazne podatke i aktivnosti s popisom potpisa koristeći usporedbu nizova. Ova vrsta otkrivanja nije u stanju povezivati više upita ili pregledavati uzročno-posljedične veze između upita i dobivenog odgovora. Isto tako ne pamte prijašnje upite prilikom pregleda trenutačnog upita. Sve to ograničava ovu metodu

ukoliko neki napad koristi višestruke upite, a niti jedan od njih sam po sebi nema naznaku da je napad.

2. Otkrivanje pomoću anomalija - Ova metoda uspoređuje aktivnosti koje smatra uobičajenima s onima koje senzori detektiraju kako ne bi pronašla razlike koje ukazuju na odstupanje od upisane norme. Koriste se profili korisničkih aktivnosti, rada SUBP-a, djelovanja mreže, itd. koje sustav stvara promatrajući kontrolirane, tipične aktivnosti u nekom vremenskom roku. Nakon toga koristi statističke metode usporedbe kako bi dobio uvid u bilo kakvu diskrepanciju tijekom budućeg, nenadgledanog korištenja sustava. Npr. ako se u neko vrijeme odjednom koristi puno više mrežne propusnosti nego što je zapisano u profilu, sustav će obavijestiti administratora ili poduzeti mjere opreza koje su mu zadane u profilu za taj određeni slučaj. Zbog ovakvog načina osiguranja ova metoda je korisna za otkrivanje nepoznatih vrsta napada. Sami profili se stvaraju kroz više dana, ili čak tjedana u takozvanom trening periodu. Profili imaju mogućnost biti statični i dinamični. Statični su isti dok ih administrator ne promijeni, a dinamični neprestano uče i nadopunjuju svoje značenje uobičajenog rada sustava. Statične profile je stoga potrebno mijenjati nakon nekog vremena zbog promjena u načinu korištenja sustava ali su zato otporni na inkrementalne promjene koje bi mogle prevariti dinamični profil. Također postoji problem slučajnog obuhvaćanja nenormalnog ponašanja tijekom trening perioda koji onda postane dio profila i samim time ga sustav smatra normalnim.
3. Analiza protokola stanja (eng. Stateful Protocol Analysis[23]) također koristi profile prihvaćenog ponašanja i uspoređuje ih sa zabilježenim događajima kako bi pronašao odstupanja. Za razliku od potpisa koji otkrivaju anomalije, ova analiza koristi potpise koje je priložio proizvođač sigurnosnog sustava. Bazira se na razumijevanju i praćenju stanja mreže i povezanih protokola prateći koja su, prema profilu, dopuštena stanja. Važan dio razumijevanja stanja je uparivanje upita s njihovim odgovorima. Ako se neki korisnik odluči prijaviti u bazu podataka, IDPS će provjeriti tu radnju sa svojim profilom i tek onda dopustiti prijavu. Ako korisnik u aplikaciji, na razini prijave pokuša nešto što IDSP smatra da nije pravilno ponašanje, uključit će sigurnosne mjere. Ukoliko se korisnik samo prijavio,

[23] http://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-94.pdf

sustav to zapisuje i dopušta korištenje drugih naredbi i upita koje odgovaraju toj razini u profilu. Isto tako ako korisnik s novim dopuštenim mogućnostima pokuša koristiti upite u čudnim redoslijedima ili upite pretjerane veličine, sustav će reagirati i spriječiti komunikaciju između poslužitelja i korisnika. Mana ovakve metode je što zahtjeva poveću količinu resursa od sustava zbog kompleksnosti takve analize i što je u stanju blokirati legitimne upite jer profil koji se koristi možda nije upoznat sa svim dijelovima baze podataka koju pokušava štititi.

5.8. Rudarenje zaključivanjem

Zaključivanje u ovom smislu označuje napad koji koristi princip rudarenja podataka u svrhu dobivanja informacija o nepoznatim dijelovima ciljane baze podataka. Napadač u ovom napadu koristi dostupne podatke koji su sami po sebi naizgled dovoljno zaštićeni kako bi zaključio nešto o podacima kojima nema pristup[24]. Ovim načinom napadač može dobiti potrebne informacije o strukturi baze podataka, kako su pohranjeni podaci i slično, u svrhu dobivanja dovoljno znanja koje može iskoristiti u sljedećoj, direktnoj fazi napada ili jednostavno zaključiti dovoljno o podacima koji ga zanimaju.

5.9. Motrenje aktivnosti baza podataka

(DAM - Database Activity Monitoring) je popularan način zaštite sustava. Ovi programi napravljeni su da označe uzbunu ako prema bazi bude poslan upit koji krši neka predodređena pravila ili ako korisnik postupa van uobičajenih parametara. Ti parametri uključuju prijave s neuobičajenih IP adresa, vrijeme prijave koje ne odgovara povijesti korisnika, pokušaj pristupa nekim drugim dijelovima baze podataka van normalne aktivnosti tog korisnika. Sama po sebi, aplikacija DAM-a neće spriječiti napad nego upozoriti na potencijalno sumnjivo ponašanje, ali raznim preinakama moguće ih je konfigurirati da zaustave promet sa sumnjivih adresa ili prilikom

[24] http://www.cs.uah.edu/~delugach/Papers/Protecting-Databases-From-Inference-Attacks.pdf

sumnjivog ponašanja, te krivog korištenja upita, tj. Upita koji ne odgovaraju unaprijed zapisanim parametrima.

5.10. Vatrozidi

Vatrozidi mogu zaštiti od napada na razini same baze podataka ili na razini aplikacije. Malen broj naredbi je moguć unutar SQL upita, ali kombinirajući razne upite i varijable broj permutacija raste. Vatrozid baze podataka topografski se nalazi između same baze i sustava koji upravlja bazom. Bazira se na listi mogućih i legitimnih upita, te jednostavno blokira one koji ne odgovaraju njegovim pravilima. Tako se otprije spomenuti broj permutacija smanjuje na nekolicinu odabranih i provjerenih. Vatrozid koristi pravila nadzora koji uzimaju u obzir prijašnje incidente kako bi razotkrili zlonamjerne pokušaje na trenutačnim upitima. Ti prijašnji incidenti, tj. uzorci napada zovu se potpisi. SQL upiti, prije nego stignu do same baze provjeravaju se prema potpisima zapisanim u vatrozidu. Korisnici mogu sami pridodati potpise ili ih dobiti preko ažuriranja, koja najnovije otkrivene slabosti sustava automatski pridodaje. Kako mogu blokirati sve što ne odgovara listi sigurnih upita (bijeloj listi - Whitelist), vatrozidi mogu i blokirati upite koji odgovaraju suprotnoj listi (crnoj listi - Blacklist). Neki vatrozidi mogu i sami uočiti nepravilnosti i ranjivosti u bazama podataka te obavijestiti administratore istih o tom problemu. Uz blokiranje upita prema bazi podataka, njihova zadaća je i blokiranje potencijalnih “izljeva” podataka iz baze, koji su možda nastali koristeći upite s bijele liste, ali su zlorabili neku drugu funkciju. Vatrozid je po mnogočemu sličan DAM-ovima i obavlja sličnu zadaću uz dodatnu funkciju automatske blokade. No, kako vatrozid prilikom nailaženja potencijalnog sumnjivog događanja odmah blokira upite i/ili korisnika dolazi do mogućeg pogrešnog otkrivanja (eng. False positive) koje će blokirati legitimne korisnike. Vatrozid instaliran na razini aplikacije koja pristupa bazi djeluje na isti način, ali indirektno štiti bazu. Takav vatrozid nije vrlo djelotvoran, iako ima svoju funkciju i ne oduzima resurse, nego samo stvara još jednu razinu zaštite. [25]

[25] https://securosis.com/assets/library/reports/Database_DoS.pdf

6. Povijest napada

Kao i sve druge vrijednosti baze podataka su podložne napadima i krađi. Otkad postoje baze postoje i napadi na njih. Kako se količina podataka pohranjena u bazama eksponencijalno povećava[26] tako su i sami napadi sve destruktivniji i zapaženiji. U samim počecima komercijalne uporabe baza podataka izljevi podataka nisu bili uvijek otkriveni javnosti. U 1980-ima počinje veća javna osviještenost javnosti o takvim napadima i njihovim posljedicama.

Većina organizacija koja prati napade na baze podataka počinje zabilježavati napade koji su počinjeni u 2005. i kasnije. Tvrtka Privacy Rights Clearinghouse od tada je zabilježila 7717 napada na baze podataka i preko milijardu ukradenih zapisa. Oba broja su u stvarnosti veća jer su zapisani samo javno objavljeni napadi i javno objavljene brojke. Neki od napada nemaju zaračunate brojeve povrijeđenih zapisa nego su samo zabilježeni kao nepoznato.[27]

Više od 40 milijuna[28] zapisa o brojevima kreditnih kartica ukradeno je 2005. godine od tvrtke CardSystems Solutions Inc. koja obrađuje platne transfere između različitih banaka.

Lanac trgovina TJ Maxx 2006. godine pretrpio je napad na njihovu bazu podataka u Framinghamu u Massachusettsu. Ukradeno je oko 94 milijuna podataka[29] korištenih u transakcijama kreditnim karticama ali i veliki broj nepoznatih podataka, koje zbog tehnologije koju su koristili napadači nisu mogli utvrditi.

Jedan od većih napada dogodio se 2011. godine nad bazama podataka Sony Playstation Networka (PSN). Osobni podaci, lozinke i potencijalno podaci o kreditnim karticama od otprilike 70 milijuna ljudi je bilo ukradeno. Sama usluga PSN-a, koju koristi puno više ljudi diljem svijeta prestala je raditi na više tjedana. Do danas nema dokaza da su ukradeni podaci iskorišteni za bilo kakvu krađu, ali je sam napad koštao Sony 171 milijun dolara.[30]

---

[26] https://insidebigdata.com/2017/02/16/the-exponential-growth-of-data/

[27] https://www.privacyrights.org/data-breaches

[28] http://www.washingtonpost.com/wp-dyn/content/article/2005/06/17/AR2005061701031.html

[29] https://www.computerworld.com/article/2544306/security0/tjx-data-breach--at-45-6m-card-numbers--it-s-the-biggest-ever.html

[30] https://www.wired.com/2011/05/sony-psn-hack-losses/

Poznata tvrtka Yahoo pretrpjela je dva velika napada na svoje baze oko 2013. i 2014. godine. Sama prijava o napadu objavljena je tek 2016. Prva prijava iz 9. mjeseca 2016. pokazala je kako je ukradeno 500 milijuna korisničkih računa, a samo par mjeseci kasnije otkrili su još jednu krađu od preko milijardu računa. U napadu su ukradena imena, brojevi telefona, datumi rođenja, kriptirane lozinke i sigurnosna pitanja. Tvrtka je vrlo polako primjenjivala novije metode zaštite čak i nakon nekoliko prijašnjih napada i krađa.[31] Po broju ukradenih podataka ovo je najveći napad na baze podataka u povijesti.

Najnoviji veliki napad na baze podataka dogodio se ove godine oko srpnja. Kreditnoj tvrtki Equifax ukradeno je 143 milijuna zapisa. Sami zapisi su sadržavali veći broj osobnih podataka uključujući imena, brojeve socijalnog osiguranja (ekvivalent osobnoj iskaznici), datume rođenja, adrese i brojeve vozačkih dozvoli. Informacije s kojima je lagano lažirati identitet.[32]

Manji napadi događaju se svake godine i za mnoge veće napade treba proći više godina da budu prijavljeni ili čak otkriveni.

[31] https://www.nytimes.com/2016/12/14/technology/yahoo-hack.html?mcubz=0

[32] http://money.cnn.com/2017/09/07/technology/business/equifax-data-breach/index.html

7. Zaključak

Baze podataka kontinuirano se razvijaju iz dana u dan. Nove tehnologije i rješenja uvode se u postojeće sustave i novi sustavi se osmišljavaju ali i polako uvode. Svaki korak naprijed tehnologije znači da i sigurnost te tehnologije mora ići naprijed. Nažalost, zbog naravi problema sigurnosti tako kompleksnih sustava, sigurnost i zaštita zaostaju. Cilj je onda pokušati predvidjeti što veći broj poteškoća i na vrijeme popraviti što više nedostataka. Kompleksnost i različitost raznih sustava i samih baza podataka, te veliki broj namjena zahtjeva još veću razinu sigurnosti. Pažnja je potrebna na svim razinama implementacije baza podataka. Od fizičke sigurnosti i brige o poslužiteljima, do održavanja korisničkih računa i njihovih prava. Mnoga rješenja neće uvijek biti dobra za svaku upotrebu i potrebno je ekopsežno istraživanje kako bi svaka pojedina baza bila što sigurnija. Napadači imaju razne motivacije i namjere, od benignih bijelih napadača (eng. white hat) koji će ukazati na probleme, do kriminalnih crnih (eng. black hat) koji će uništiti sustave i krasti podatke ali najviše moguće stanje sigurnosti potrebno je održavati bez obzira na razlog napada. Naravno, sama implementacija sigurnosti nije dovoljna ako sustavi sami nisu dovoljno dorađeni. Kasno otkrivanje grešaka i kašnjenje puštanja ažuriranja u opticaj mogu imati vrlo negativne posljedice bez ikakve mogućnosti, od samih korisnika i administratora, da to spriječe. Svaki korisnik, administrator i tvorac baza podataka i sustava koji njima upravljaju odgovoran je za održavanje jednog od najvažnijih elemenata modernog poslovanja, te za pametno i promišljeno korištenje privatnih podataka.

8. Izvori

1. A Relational Database Overview, *The Java Tutorials*, https://docs.oracle.com/javase/tutorial/jdbc/overview/database.html, (26.9.2017.)
2. Associated Press, *Ex-AOL worker who stole e-mail list sentenced*, 17.8.2005. http://www.nbcnews.com/id/8985989/ns/technology_and_science-security/t/ex-aol-worker-who-stole-e-mail-list-sentenced/#.Wcur-WiCzcs, (27.9.2017.)
3. CARNet, *Modeli kontrole pristupa*, 21.2.2008., http://www.cis.hr/www.edicija/LinkedDocuments/CCERT-PUBDOC-2008-02-218.pdf, (26.9.2017.)
4. *Data Breaches*, 2017., https://www.privacyrights.org/data-breaches, (27.9.2017.)
5. Di Justo, P. *Your secret is out: data breaches cost companies billions each year.* // Wired USA. Veljača 2007. Str. 50.
6. Goel, V, Perlroth, N, Yahoo Says 1 Billion User Accounts Were Hacked, 14.12.2016., https://www.nytimes.com/2016/12/14/technology/yahoo-hack.html?mcubz=0, (27.9.2017.)
7. Hayden, E. *Data breach protection requires new barriers*, Svibanj 2013., http://searchsecurity.techtarget.com/feature/Data-breach-protection-requires-new-barriers, (27.9.2017.)
8. Hinke, Thomas H; Delugach, Harry S; Wolf, Randall P. *Protecting databases from inference attacks* // Computers & Security, Vol. 16, No. 8. Velika Britanija : Elsevier Science Limtied, 1997. 687-708.
9. Katayama, F, *Hacker hits up to 8M credit cards*, 27.2.2003., http://money.cnn.com/2003/02/18/technology/creditcards/, (27.9.2017.)
10. Krim, J, Barbaro, M, *40 Million Credit Card Numbers Hacked*, 18.6.2005., http://www.washingtonpost.com/wp-dyn/content/article/2005/06/17/AR2005061701031.html, (27.9.2017.)
11. Laboratorij za sustave i signale, *Zaštita baza podataka*, 5.8.2012., http://www.cis.hr/files/dokumenti/CIS-DOC-2012-08-059.pdf, (26.9.2017.)
12. Lord, N, *The History of Data Breaches*, 27.7.2017., https://digitalguardian.com/blog/history-data-breaches (27.9.2017.)
13. Luc Bouganim, Yanli Guo. Database encryption, 2009., S. Jajodia and H. van Tilborg. *Encyclopedia*
14. *of Cryptography and Security*, Springer, str.1-9, https://hal.archives-ouvertes.fr/file/index/docid/623915/filename/BOUGA_B6_ENC_CRYPT_2009.pdf, (26.9.2017.)
15. Manger, Robert. *Baze Podataka*, rujan 2003., http://jadran.izor.hr/~dadic/EKO/baze-podataka.pdf, (26.9.2017.)
16. McDowell, M, U*nderstanding Denial-of-Service Attacks*, 4.11.2009, https://www.us-cert.gov/ncas/tips/ST04-015, (26.9.2017.)
17. Mogull, R, Lane, A, *Understanding and Selecting a Database Encryption or Tokenization Solution*,

https://securosis.com/assets/library/reports/Securosis_Understanding_DBEncryption.V_.1_.pdf, (26.9.2017.)
18. O'Brien, Sarah A, *Giant Equifax data breach: 143 million people could be affected*, 8.9.2017., http://money.cnn.com/2017/09/07/technology/business/equifax-data-breach/index.html, (27.9.2017.)
19. Patel, T; Malik M. *Database Security : Attacks and Control Methods* // International Journal of Information Sciences and Techniques. Vol. 6, No. ½. Changa : Charotar University of Science & Technology, 2016. Str. 175-183.
20. Pepitone, J, 5 *of the biggest-ever credit card hacks*, 12.1.2014. http://money.cnn.com/gallery/technology/security/2013/12/19/biggest-credit-card-hacks/4.html, (27.9.2017.)
21. *Personal Data: The Emergence of a New Asset Class*, Siječanj 2011., http://www3.weforum.org/docs/WEF_ITTC_PersonalDataNewAsset_Report_2011.pdf, (26.9.2017.)
22. Scarfone, K; Mell, P. *Guide to Intrusion Detection and Prevention Systems (IDPS) : Recommendations of the National Institute of Standards and Technology*. NIST Special Publication 800-94. Gaithersburg : National Institute of Standards and Technology, 2007.
23. Schreier, J, *SONY ESTIMATES $171 MILLION LOSS FROM PSN HACK*, 23.5.2011., https://www.wired.com/2011/05/sony-psn-hack-losses/, (27.9.2017.)
24. Securosis, *Dealing with Database Denial of Service*, 22.8.2013., https://securosis.com/assets/library/reports/Database_DoS.pdf, (26.9.2017.)
25. *SQL Injection*, https://www.incapsula.com/web-application-security/sql-injection.html, (30.9.2017.)
26. *The Exponential Growth of Data*, 16.2.2017., https://insidebigdata.com/2017/02/16/the-exponential-growth-of-data/, (30.9.2017.)
27. Townsend Security, *Encryption and Tokenization*, 2010, https://townsendsecurity.com/sites/default/files/Encryption_vs_Tokenization.pdf, (26.9.2017.)
28. Vijayan, J, *TJX data breach: At 45.6M card numbers, it's the biggest ever*, 29.3.2007., https://www.computerworld.com/article/2544306/security0/tjx-data-breach--at-45-6m-card-numbers--it-s-the-biggest-ever.html, (27.9.2017.)
29. What is SQL?, *SQL Course*, http://www.sqlcourse.com/intro.html, (26.9.2017.)
30. Whitman, Michael E; Mattord, Herbert J., *Principles of Information Security*. Cengage Learning EMEA, 2009.