HRVATSKA AKADEMSKA I ISTRAŽIVAČKA MREŽA
CROATIAN ACADEMIC AND RESEARCH NETWORK

Sigurnost sustava za upravljanje bazama podataka

CCERT-PUBDOC-2006-10-171

CARNet CERT u suradnji s LS&S

Sigurnosni problemi u računalnim programima i operativnim sustavima područje je na kojem CARNet CERT kontinuirano radi.

Rezultat toga rada ovaj je dokument, koji je nastao suradnjom CARNet CERT-a i LS&S-a, a za koji se nadamo se da će Vam koristiti u poboljšanju sigurnosti Vašeg sustava.

CARNet CERT, www.cert.hr - nacionalno središte za **sigurnost računalnih mreža** i sustava.

LS&S, www.lss.hr - laboratorij za sustave i signale pri Zavodu za elektroničke sustave i obradbu informacija Fakulteta elektrotehnike i računarstva Sveučilišta u Zagrebu.

Sadržaj

1. Uvod

Baze podataka su skupovi neredundantno pohranjenih i organiziranih podataka koje održavaju, distribuiraju i nadziru programi nazvani SUBP - sustavi za upravljanje bazama podataka (eng. DBMS – *Database Management System*).

U bazama podataka pohranjuju se brojne informacije iz svih mogućih područja. Različiti programi (računovodstveni, organizacijski, istraživački itd.) zahtijevaju različite informacije, a one se u današnje doba pohranjuju u bazama podataka. Stoga se danas organizacije pouzdaju u SUBP-ove kada se radi o spremanju i osiguravanju svih, pa tako i onih najtajnijih i najvrjednijih, podataka. Zbog toga za tim sustavima raste zanimanje kriminalne zajednice, a samim time i potreba da ih se učini sigurnijima.

Osim bogatstva informacija koje čuvaju, postoji još nekoliko čimbenika koji pridonose ranjivosti baza podataka. Uz današnje trendove sveprisutnosti Interneta, SUBP-ovi koji su tradicionalno bili smješteni u zatvorene mreže i iza vatrozida, postaju sve otvoreniji prema udaljenim korisnicima, a time i sve izloženiji napadima. Također je postalo vrlo lako pribaviti programske pakete popularnih SUBP-ova, što zlonamjernim korisnicima omogućuje istraživanje i pronalaženje sigurnosnih propusta.

U mnogo čemu je osiguranje baza podataka slično osiguranju računalnih mreža. U oba slučaja nastoji se korisniku dati samo neophodne ovlasti, smanjiti ranjivu „površinu“ onemogućavanjem nepotrebnih funkcionalnosti, strogo autorizirati izmjene i nadzirati pristup, odvojiti funkcionalne blokove, inzistirati na enkripciji, itd. Jedina stvarna razlika je u tome što kod baza podataka svi ovi mehanizmi djeluju unutar samog SUBP-a.

U nastavku dokumenta opisani su razlozi zbog kojih se unutar baza podataka javljaju ranjivosti. Zatim su navedeni elementi osiguranja baza podataka, kao i modeli zaštite koji se koriste kako bi se uklonio ili barem umanjio učinak tih ranjivosti. Sva ova općenito obrađena sigurnosna pitanja su na kraju dokumenta sagledana u kontekstu Oracle, MsSQL, IBM DB2, Sybase i MySQL SUBP-ova. Navedenih pet sustava izabrano je zbog njihove raširene rasprostranjenosti.

2. Ranjivosti sustava za upravljanje bazama podataka

Ranjivosti baza podataka mogu proizaći iz neispravne konfiguracije SUBP-a, programskih propusta ili sigurnosnih nedostataka unutar aplikacija povezanih s njima.

2.1. Konfiguracijski propusti

Iako SUBP-ovi često ne podržavaju sigurnosne mogućnosti tradicionalno prisutne kod drugih sustava, ispravno postavljanje postojećih mogućnosti može mnogo podići sigurnosnu razinu zaštićenosti podataka te ukloniti veliki broj ranjivosti.

2.1.1. Slaba zaštita korisničkih računa

SUBP-ovi većinom nemaju mogućnosti zaštite korisničkih računa koje su prisutne kod primjerice operacijskih sustava. Tu se prvenstveno misli na (ne)mogućnost kontrole zaporki provjerama u rječniku i na (ne)mogućnost određivanja roka valjanosti korisničkog računa. Često se tijekom postavljanja SUBP-a izvorno postavljeni i opće poznati korisnički računi i korisničke zaporke ostavljaju aktivnima bez promjene.

2.1.2. Neprikladna podjela odgovornosti

Na području upravljanja bazama podataka nije priznata uloga administratora za sigurnost. Zbog toga administratori baza podataka moraju voditi računa o korisničkim računima i zaporkama, u isto vrijeme osiguravajući ispravan rad i zadovoljavajuće performanse administrirane baze podataka. Takva situacija, uz to što otežava posao administratorima, onemogućava efikasno upravljanje ljudskim resursima.

2.1.3. Neprikladne metode nadzora

Nadzoru SUBP-a često su pretpostavljeni zahtjevi visokih performansi i štednje diskovnog prostora. Zbog toga je umanjena učinkovitost forenzičke analize i otežano utvrđivanje odgovornosti. Ispravne metode nadzora su ključne u razumijevanju napada na SUBP-ove jer bilježe aktivnosti izravno vezane uz pohranjene podatke.

2.1.4. Neiskorištene mogućnosti zaštite baza podataka

Uobičajeno je ugrađivati sigurnosne elemente u pojedine aplikacije, zanemarujući pri tome sigurnost SUBP-a. Nedostatak takvog pristupa je u tome što spomenuti sigurnosni elementi djeluju samo u slučaju kada korisnik za pristup bazi koristi klijentske aplikacije. Postoje mnogi alati (npr. Microsoft Access) koji omogućuju pristup bazi podataka pomoću ODBC-a (eng. *Open Database Connectivity*) ili nekog drugog protokola koji u potpunosti zaobilazi sigurnosne provjere ugrađene u aplikacije. Jedina pouzdana sigurnosna ograničenja su ona ugrađena izravno u SUBP.

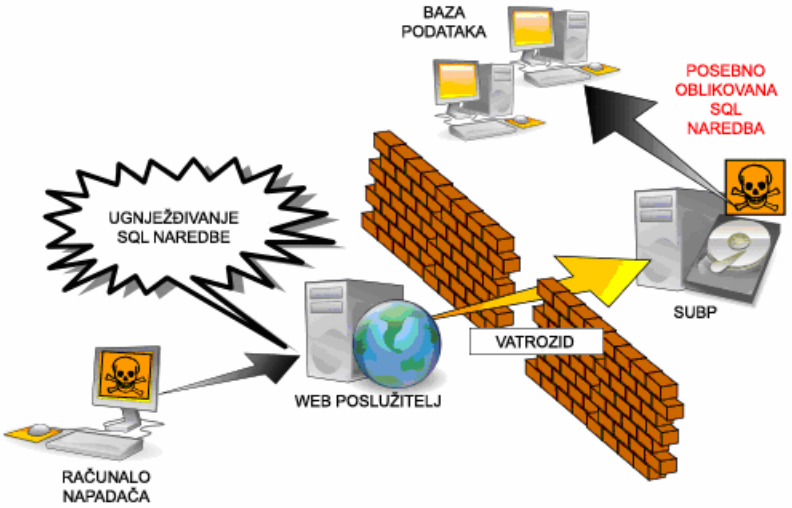
2.2. Programski propusti unutar SUBP-a

Programski propusti uključuju razne pogreške prepisivanja spremnika koje mogu udaljenim zlonamjernim korisnicima omogućiti izvođenje napada zasnovanih na uskraćivanju resursa (eng. DoS - *Denial of Service*) napada ili izvršavanje proizvoljnog programskog koda s različitim posljedicama.

2.3. Propusti u aplikacijama povezanim s bazama podataka

2.3.1. Ugnježđivanje SQL naredbi

Činjenica da se SUBP nalazi iza vatrozida ne čini ga apsolutno sigurnim od napada. Postoji nekoliko vrsta napada koje je moguće izvesti kroz vatrozid, a ugnježđivanje SQL naredbi (eng. *SQL injection*) je najčešći. To nije napad izravno na SUBP već predstavlja pokušaj promjene parametara koji se šalju aplikaciji (najčešće web aplikacija) s namjerom mijenjanja SQL naredbe poslane bazi podataka.



Slika 1: Napad ugnježđivanjem SQL naredbi

Napad ugnježđivanjem SQL naredbi najbolje se može ilustrirati primjerom autorizacije na web stranici. Korisnik unosi svoje korisničko ime i zaporku pomoću kojih se stvara SQL upit za pretraživanje tablice s korisničkim imenima i zaporkama. Ako se u tablici pronađu uneseno ime i zaporka, korisnik postaje autoriziran. Problem kod ovakvog pristupa je što se SQL upit stvara ulančavanjem bez izuzimanja jednostrukih navodnika.
Ako je uneseno korisničko ime „*Ivan*“ i zaporka „*zaporka2*“, ulančani SQL upit će izgledati kao što je prikazano u nastavku:

```
SELECT *
FROM WebKorisnici
WHERE KorisnickoIme='Ivan' AND Zaporka='zaporka2'
```

Napadač može umjesto zaporke upisati niz slova i završiti znakovni niz jednostrukim navodnikom te dodati logički izraz, koji je uvijek istinit, te tako kao odgovor dobiti sve retke tablice. Na primjer, ako napadač umjesto zaporke upiše sljedeći niz znakova:

```
Aa' OR 'A'='A
```

SQL upit postaje:

```
SELECT *
FROM WebKorisnici
WHERE KorIme='Ivan' AND Zaporka=' Aa' OR 'A'='A'
```

Ovaj upit uvijek vraća sve retke tablice te tako uvjerava web stranicu da je napadač unio ispravno korisničko ime i zaporku. Zbog toga što je uobičajeno da se u zapisima koji sadrže podatke svih korisnika na prvom mjestu nalazi administrator, postoji velika mogućnost da se napadač autorizira kao administrator aplikacije.
Sprečavanje ugnježđivanja SQL naredbi može biti jednostavno ako se poznaje mehanizam napada. Dva su moguća pristupa: provjera korisničkih unosa i korištenje parametriziranih upita. Prvi pristup se odnosi na prihvaćanje samo onih unosa koji se sastoje od dozvoljenih znakova, a to su najčešće samo alfanumerički znakovi. Korištenje parametriziranih upita znači povezivanje varijabli umjesto ulančavanja znakovnih nizova.

3. Elementi zaštite sustava za upravljanje bazama podataka

Ugrađivanje sigurnosnih elemenata izravno u SUBP-ove i njihova ispravna primjena jedini su pravi način za uklanjanje ranjivosti. Ti elementi obuhvaćaju dodjeljivanje primjerenih ovlasti i dozvola pristupa, primjenu efektivnih korisničkih računa i zaporki, primjerene metode nadzora i logiranja, korištenje enkripcije i nadzor nad pristupom tablicama.

3.1. Dodjeljivanje primjerenih ovlasti i dozvola pristupa

Korisnicima se dodjeljuju minimalne potrebne ovlasti prema tzv. '*least privilege*' načelu. Ovo načelo temelji se na dozvoli pristupa samo onim podacima baze i funkcionalnostima SUBP-a koji su korisnicima neophodno potrebni, obzirom na njihov status i opis posla. Pri tome treba voditi računa o ugrađivanju opisanih ograničenja izravno u SUBP, a ne u klijentsku aplikaciju koja pristupa nekoj od pohranjenih baza podataka.

U svrhu podizanja računalne sigurnosti, ne preporuča se izravno dodjeljivanje ovlasti pojedinim korisničkim računima. Puno je bolji način da se oblikuju tzv. "uloge" (eng. *roles*) i da se njima dodijele pojedine ovlasti. Nakon toga se svakom korisniku dodaju "uloge" koje mu pripadaju. Na taj način jedan korisnik može zauzeti više uloga, a olakšano je dodjeljivanje i oduzimanje ovlasti vezanih uz radne zadatke.

Administratorima se savjetuje dokumentiranje zahtjeva za stvaranje, kao i samo stvaranje korisničkih računa, te pridjeljivanje i oduzimanje pojedinih uloga korisnicima. Također, prilikom promjene radnog mjesta ili radnog zadatka potrebno je preispitati ovlasti korisnika. Korisničke račune bivših zaposlenika potrebno je odmah ukinuti i provesti odgovarajuće postupke nad objektima baze podataka koji su pripadali takvim korisnicima.

3.2. Efikasni korisnički računi i zaporke

Korisničke račune, nužne za pristup bazi podataka, potrebno je definirati u skladu s tradicionalnim metodama upravljanja korisničkim računima. To podrazumijeva promjenu izvorno postavljenih zaporki, onemogućenje korisničkog računa nakon određenog broja neuspjelih prijava, ograničenje pristupa podacima, onemogućenje neaktivnih korisničkih računa te upravljanje životnim ciklusom korisničkih računa.

3.3. Primjerene metode nadzora i evidencije

Jedan od ključnih elemenata zaštite SUBP-ova je nadzor koji treba biti usklađen s njihovom primjenom. Pogrešan je pristup nadzoru temeljen na načelu „sve ili ništa“. Pažljivo postavljen sustav nadzora omogućava uštede vremena i ne utječe značajno na performanse nadziranog SUBP-a.

Zbog toga je potrebno ograničiti veličinu dnevničkih zapisa kako bi do izražaja došli događaji kritični za sigurnost sustava. Ako korisnici bazi podataka trebaju pristupati samo sa svojih terminala, čija su imena ili IP adrese poznate, onda bilježenje svih pristupa bazi kod kojih korisnici nisu koristili svoje terminale može otkriti zloupotrebu korisničkog računa. Još jedan primjer je nadzor nad pristupima bazi izvan radnog vremena što može otkriti nedozvoljene radnje na koje se korisnici ne bi odvažili tijekom normalnog radnog vremena.

3.4. Korištenje enkripcije

Obzirom na podatke koji se kriptiraju i na razinu na kojoj se kriptiranje obavlja, postoje različite vrste enkripcija primjenjivih u zaštiti baza podataka.

Jedna od mogućnosti je korištenje enkripcije za zaštitu podataka tijekom prijenosa (eng. *data-in-motion*). U tu svrhu većina SUBP-ova podržava komunikaciju uporabom SSL (eng. *Secure Socets Layer*) zaštitnog protokola. Drugi je način primjena enkripcije na podatke u mirovanju (eng. *data-at-rest*), ali ni tada nije potrebno kriptirati sve podatke, već samo najosjetljivije.

Moguće je raditi i enkripciju datoteka (eng. *file-based*), ali takav pristup ne štiti od napada kroz SUBP. Enkripcija na razini programskog sučelja (eng. API - *Application Programming Interfaces*) podrazumijeva kriptiranje komunikacije među pojedinim podsustavima SUBP-a. Ona je složena i zahtjeva puno posla što povećava rizik od pogreške.

Najslabiju podršku baze podataka imaju za tzv. '*transparent*' enkripciju. To je enkripcija koja se automatski primjenjuje pri svakoj promjeni ili unosu potencijalno osjetljivih podataka. Automatska primjena enkripcije znači da nema potrebe za eksplicitnim pozivanjem enkripcijskih funkcija. Time se izbjegavaju programerske pogreške kao što su pozivanje pogrešnih enkripcijskih funkcija, pozivanje ispravnih funkcija, ali s krivim parametrima ili njihovo nepozivanje u slučajevima kada je to potrebno. Ovo se postiže premještanjem enkripcije s razine aplikacijskog sloja na sloj baze podataka.

3.5. Kontrola pristupa tablicama

Kontrola pristupa tablicama je vjerojatno najzanemarivaniji element zaštite baza podataka zbog toga što je njena implementacija složena i zahtjeva suradnju sistemskog administratora i razvojnog programera baze podataka. Primjeri ovakve kontrole su onemogućavanje čitanja tablice u istoj sjednici u kojoj je u nju obavljen upis ili dozvoljavanje čitanja samo određenog tipa tablica

4. Modeli zaštite baza podataka

Osim ugrađenih sigurnosnih elemenata, u onemogućavanju napada na baze podataka važnu ulogu imaju i modeli njihove zaštite. Ovi modeli obuhvaćaju delegiranje odgovornosti, smještanje poslužitelja u unutarnju mrežu, primjenu sustava dozvoljenih IP adresa, periodičke analize, postavljanje zamki te primjenu zakrpi. Kako bi povećali sigurnost SUBP-ova bitno je primjenjivati sve navedene modele koji sustav štite na različitim razinama.

4.1. Delegiranje odgovornosti

Administratore baze podataka potrebno je zadužiti samo za poslove upravljanja SUBP-ovima i osiguravanja zadovoljavajućih performansi, ali im je također potrebno omogućiti i delegiranje administracije sigurnosnih poslova. Time se postiže veća efikasnost u obavljanju radnih zadataka te olakšava prijenos odgovornosti prilikom prelaska zaposlenika na nova radna mjesta. Delegiranjem odgovornosti može se pojedinim administratorima omogućiti obavljanje radnih zadataka u okviru pojedinog odjela tvrtke, npr. marketinškog ili financijskog odjela.

4.2. Smještanje SUBP-a u unutarnju mrežu

Smještanjem SUBP-a u unutarnju mrežu ograničava se pristup samoj bazi podataka. Ako je baza nedostupna, onda je i sigurna od napada. Ovaj pristup je vrlo efikasan i uglavnom primjeren jer često ne postoji potreba za vanjskim pristupom bazi podataka. Kada je to ipak potrebno, kao npr. kada baza podataka služi kao izvor informacija dinamičkim web stranicama, tada web poslužitelj i baza podataka trebaju biti smješteni na odvojenim računalima, kako zbog sigurnosnih razloga tako i zbog poboljšanja performansi. U takvim situacijama poslužitelj s bazom podataka treba prihvaćati isključivo veze s pridijeljenim mu web poslužiteljem.

4.3. Sustav dozvoljenih IP adresa

SUBP treba posluživati isključivo sigurne IP adrese. Ako se radi o spomenutom slučaju s dinamičkim web stranicama onda to treba biti samo IP adresa web poslužitelja, a ako se baza podataka koristi u sprezi s nekom lokalno korištenom aplikacijom onda pristup treba biti dozvoljen isključivo s IP adresa koje pripadaju lokalnoj mreži. Lokalnim i izvana vidljivim bazama podataka treba pridijeliti zasebne poslužitelje.

4.4. Periodička analiza promjena i sumnjivih situacija

Korištenjem Unix naredbe „*grep*“ ili Windows naredbe „*find*“ moguće je pronaći zaporke zapisane u skriptama, tekstualnim datotekama, porukama elektroničke pošte te čak u log datotekama. Ovakve pretrage su dugotrajne pa ih je potrebno raditi izvan radnog vremena kako bi se umanjio utjecaj na korisnike i istovremeno sakrila činjenica da se takva pretraga uopće provodi. Periodički je također potrebno pregledati i korisničke račune ne bi li se pronašli korisnici s nepotrebno visokim ovlastima ili ulogama.

4.5. Postavljanje zamki

Neke od periodičkih analiza poželjno je automatizirati tako da rezultate dostavljaju elektroničkom poštom ili ih spremaju u posebnu datoteku ili tablicu. Primjer primjene ove strategije je zapisivanje svakog dodjeljivanja uloge administratora korisnicima kojima ta uloga inače ne pripada. Ovdje je, dodatno, potrebno voditi računa o tome da automatske pretrage neće pronaći ranjivosti za koje nisu unaprijed programirane, što može dovesti do stvaranja lažnog osjećaja sigurnosti.
U slučaju kada jedan od korisnika baze podataka treba dobiti otkaz, može se pokazati korisnim nadgledati njegov korisnički račun određeno vrijeme prije napuštanja posla. Tako se mogu otkriti pokušaji pospremanja u zadnji tren, krađe ili nanošenja štete bazi podataka.
Nadziranjem datoteka ili tablica u koje se spremaju podaci prikupljeni nadzorom baze podataka, smanjuje se mogućnost prikrivanja tragova djelovanja eventualnog uljeza.

4.6. Primjena zakrpi i testiranje

Iako sve zakrpe uklanjaju ranjivosti treba ih oprezno primjenjivati zbog mogućnosti unošenja novih pogreški u sustav. Jedino oružje protiv takvih pogrešaka je ispitivanje. Neke zakrpe postavljaju zahtjeve na sustav na kojem se primjenjuju. Te zahtjeve i opseg zakrpe je potrebno poznavati kako bi se utvrdilo postoji li uopće potreba za njezinom primjenom.

5. Sigurnost nekih izvedbi SUBP-a

U ovom poglavlju opisane su neki poznatiji i učestaliji sustavi za upravljanje bazama podataka s naglaskom na ranjivost i zaštitu tih istih sustava.

5.1. Oracle

Oracle je najviše raširen SUBP i pokriva najveći dio tržišta. Razlozi za to su duga tradicija i podržanost od strane većine operacijskih sustava.

5.1.1. Ranjivosti

Listener je poslužitelj preko kojega klijenti pristupaju bazi podataka. Sama činjenica da je ovaj poslužitelj smješten izvan baze podataka predstavlja problem zbog toga što mogućnosti njegova udaljenog administriranja i postavljanja zaporke nisu dovoljno dokumentirane te su često nepoznate. Unutar *Listener* poslužitelja ne postoje uobičajene mogućnosti upravljanja zaporkama kao što su onemogućavanje računa, odvojen nadzor ili istjecanje zaporke. Zbog toga je pomoću jednostavne skripte moguće probiti čak i vrlo jake zaporke.
Listener poslužitelj u nekim situacijama može neovlaštenim korisnicima dozvoliti pristup potencijalno osjetljivim informacijama. Ako se poslužitelju pošalje paket s neispravnim "SIZE OF PACKET" poljem on odgovara paketom koji sadrži dio prethodne naredbe. Izdana je zakrpa za ovaj propust, a napadi se mogu spriječiti korištenjem vatrozida.
Unutar *Listener* poslužitelja otkriveno je i nekoliko pogrešaka prepisivanja spremnika. Jedan od tih propusta može udaljenom zlonamjernom korisniku omogućiti izvođenje proizvoljnog programskog koda manipuliranjem SEH (eng. *Structured Exception Handling*) mehanizmom. I za ove sigurnosne nedostatke su objavljene odgovarajuće zakrpe.
Pored sigurnosnih problema vezanih uz *Listener* poslužitelj, unutar Oracle baze podataka postoji značajna ranjivost povezana sa "SYS.LINK$" tablicom. U nju se, u slučaju ostvarivanja veze s nekom drugom bazom podataka, zapisuju vrijeme stvaranja spomenute veze te korisničko ime i zaporka korišteni pri tome. Podaci se spremaju bez enkripcije pa im može pristupiti svaki korisnik sa SELECT ANY TABLE ovlastima.

5.1.2. Sigurnosni elementi

Korištenjem "PRODUCT USER PROFILES" alata moguće je onemogućiti korištenje određenih naredbi i funkcionalnosti od strane pojedinih korisnika. Tako se može globalno onemogućiti "HOST" mogućnost koja dozvoljava pristup operacijskom sustavu.

Oracle omogućuje enkripciju korisničkih zaporki tijekom mrežne komunikacije. Ako se ova mogućnost uključi na klijentskom i poslužiteljskom računalu, Oracle koristi prilagođeni DES (eng. *Data Encription Standard*) algoritam za enkripciju zaporki prije slanja. Za enkripciju cjelokupnog mrežnog prometa prema SSL protokolu potrebno je instalirati *Oracle Advanece Security* paket. Inačice namijenjene Windows operacijskim sustavima podržavaju enkripciju na razini datoteka korištenjem EFS (eng. *Encripting File System*) datotečnog sustava. Enkripcija na razini programskog sučelja je omogućena "DBMS_OBFUSCATION_TOOLKIT" alatom.
U svrhu podizanja računalne sigurnosti, korisnicima se savjetuje pronalaženje i promjena svih izvorno postavljenih korisničkih zaporki kao što su: "SYS", "SYSTEM" ili "APPS". Oracle omogućuje kontrolu složenosti korisničkih zaporki, njihovog roka trajanja i ponovnog korištenja.
Također, Oracle posjeduje nekoliko metoda autorizacije korisnika:

1. *Kerberos security* – implementira Kerberos protokol za sigurno uzajamno dokazivanje identiteta korisnika tijekom komunikacije koja se temelji na enkripciji simetričnim ključem i zahtjeva sigurnu treću stranu (eng. *Trusted Third Party*, TTP),
2. VPD (eng. *Virtual Private Databases*) – tehnologija koja omogućava ograničenje pristupa pojedinim zapisima u tablici,
3. *Role-based security* – omogućuje grupiranje ovlasti u uloge koje je potom moguće pridijeliti pojedinim korisnicima,
4. *Grant-execute security* – omogućuje ograničavanje mogućnosti procedura ovisno o ovlastima korisnika koji ih pokreće,
5. *Authentication servers* – poslužitelji za sigurnu identifikaciju vanjskih korisnika,
6. *Port access security* – *Listener* poslužitelj može se postaviti tako da ograniči pristup pojedinim portovima.

Nadzor nad Oracle bazom podataka obavlja se stvaranjem "AUDIT TRAIL VIEWS" zapisa pomoću "CATAUDIT.SQL" skripte. Podatke prikupljene nadzorom moguće je pohranjivati za svaku sjednicu ili za svaki uočen pokušaj pristupa. Za vremenski ograničen nadzor koristi se "DBMS_JOB" mogućnost, koja uz "TRIGGERS" mogućnost može poslužiti i za postavljanje zamki uljezima. Korisnicima se pokazalo korisnim nadziranje već spomenute "SYS.LINK$" tablice te tablice "SYS.AUD$" u koju se spremaju podaci prikupljeni nadzorom.

5.2. MsSQL

Microsoft SQL Server (MsSQL) je SUBP koji je, u usporedbi s Oracle i IBM DB2 SUBP-ovima, nov proizvod s brzo rastućom popularnošću.

5.2.1. Ranjivosti

Kada se MsSQL izvodi u načinu rada mješovite autentikacije (eng. *mixed-mode authentication*), pristupne zaporke se spremaju na raznim lokacijama. Neke od njih se štite snažnom enkripcijom i uključuju visok stupanj ograničenja. Preostale se (npr. one koje SQL poslužitelj koristi za uspostavljanje veza prema samom sebi i prema drugim poslužiteljima) štite slabom enkripcijom i uz nizak stupanj ograničenja. Pregledavanjem sistemskih tablica i pohranjenih procedura ili korištenjem SQL Profiler alata, napadači mogu otkriti gdje i kako se ove zaporke spremaju. Ovom ranjivošću su prvenstveno ugrožene zaporke "SQL Agent" paketa, DTS (eng. *Data Transformation Services*) alata te zaporke korištene prilikom replikacije.
Zlonamjerni prijavljeni korisnik može neovlašteno steći više korisničke ovlasti ubacivanjem trojanskog konja u SQL poslužitelj. Kako bi to učino, korisnik mora imati "db_ddladmin" ulogu te na određen način promijeniti neku od "dbo" (eng. *DataBase Owner*) pohranjenih procedura. Kada korisnik s "db_owner" (ili višim) ovlastima pokrene promijenjenu proceduru, napadač može dobiti i "db_owner" ulogu. U procedure ranjive na ovakve napade spadaju sve GTSP (eng. *Global Temporary Stored Procedures*) procedure.
Korisnik ne može pristupiti tablicama za koje nema odgovarajuće ovlasti, ali se takvim tablicama može pristupiti pomoću procedura ili pogleda koje je stvorio njihov vlasnik. Također, zbog toga što svi korisnici mogu stvarati privremene procedure i tablice, vrlo je jednostavno izvesti napad uskraćivanjem resursa (DoS) na MsSQL poslužitelj. Potrebno je samo stvoriti privremenu tablicu i

pokrenuti beskonačnu petlju koja ju puni. Privremene tablice se spremaju u "tempdb" sistemskoj bazi podataka koja se u slučaju ovakvog napada povećava do trenutka rušenja poslužitelja.
Najpoznatije ranjivosti unutar MsSQL baze podataka vezane su uz prepisivanja spremnika koja su 25. siječnja 2003. omogućila *Slammer* računalnom crvu izazivanje DoS uvjeta na desecima tisuća računala i značajno usporenje cjelokupnog internetskog prometa. Ovaj crv je tako malen da stane u jedan UDP paket te nema programskog koda kojega treba zapisati na disk, već ostaje u memoriji računala. *Slammer* generira nasumične IP adrese te se šalje na njih. Neki usmjerivači (eng. *router*) su se srušili pod opterećenjem, što je izazvalo niz poruka za osvježavanje tablica usmjeravanja. Šest mjeseci prije spomenutog datuma, Microsoft je izdao zakrpe koje su onemogućavale izvođenje ovog crva, ali na velikom broju instalacija ranjivih Microsoft SQL Server 2000 i Microsoft Desktop Engine (MSDE) 2000 programskih paketa te zakrpe nisu bile primijenjene.

5.2.2. Enkripcija

Microsoft SQL Server poslužitelj podržava SSL protokol za kriptiranu mrežnu komunikaciju. Enkripcija datoteka je također moguća korištenjem EFS datotečnog sustava dok je enkripcija na razini programskog sučelja omogućena Crypto API sučeljem koje koristi proširene pohranjene procedure.

5.2.3. Operacijski sustav

SQL Server poslužitelj može biti instaliran na više Windows datotečnih sustava (NTFS, FAT, FAT32). Preporuča se korištenje NTFS datotečnog sustava za SQL poslužitelj i za datoteke s podacima jer se time omogućava ograničenje pristupa pojedinim datotekama i direktorijima. NTFS datotečni sustav je nužan preduvjet i za korištenje već spomenutog EFS sustava za enkripciju datoteka.
Tijekom instalacije potrebno je odabrati Windows korisnički račun koji će biti pridijeljen MsSQL poslužitelju. Pri tome se treba voditi načelom minimalnih ovlasti jer se time ograničuju mogućnosti napadača koji je probio obranu SQL Server poslužitelja.

5.2.4. Postavke

Broj aktivnih mrežnih programskih biblioteka (eng. *netlib*) treba ograničiti na minimum potreban za funkcioniranje SUBP-a. Najpopularnija mrežna biblioteka TCP/IP u kombinaciji sa SSL protokolom omogućuje siguran pristup SQL Server poslužitelju. Budući da MsSQL poslužitelj ne podržava onemogućavanje korisničkih računa, vrlo je uputno nadzirati i bilježiti neuspješne pokušaje prijave na sustav.
Prema početnim postavkama, SQL Server poslužitelj omogućuje drugim poslužiteljima na mreži udaljeno pokretanje pohranjenih procedura. Ako nije potrebna, ovu mogućnost treba isključiti. Također, Mogućnost izravnog mijenjanja sistemskih tablica bi u normalnim radnim uvjetima, trebala biti onemogućena.
SQL Server Monitor je alat koji uobičajeno sluša na UDP portu 1434 i daje informacije o instancama prisutnim na poslužitelju i ne bi trebao biti dostupan klijentima. Zbog toga bi vatrozid trebao blokirati vanjski promet prema TCP portu 1433 i UDP portu 1434.
Nadalje, module *SQL Server Agent*, MSDTC (eng. *Microsoft Distributed Transaction Coordinator*) i *MSSearch* potrebno je isključiti u svakoj konfiguraciji u kojoj nisu apsolutno neophodni. Također treba ukloniti i sve nepotrebne i potencijalno opasne pohranjene procedure.

5.3. IBM DB2

DB2 SUBP ima na raspolaganju manji broj funkcionalnosti od, primjerice, Oracle ili MsSQL SUBP-ova, ali se ipak ne može smatrati sigurnijom od njih. Unatoč tome, poznat je po izdavanju vrlo kvalitetnih zakrpi u vrlo kratkom roku, što mu donosi određenu prednost.

5.3.1. Ranjivosti

Kao i kod drugih programskih paketa, tako i kod DB2 sustava postoje različiti sigurnosni propusti koji mogu uzrokovati velike sigurnosne propuste. Neki od njih koji su otkriveni i za koje su izdane zakrpe, su i sljedeći:

- Sigurnosni propust postoji unutar *Control Centar* grafičkog korisničkog sučelja za udaljeno upravljanje objektima baze podataka pod Windows operacijskim sustavima. Spomenuti programski paket izvršava se pod imenom "db2ccs.exe" i osluškuje mrežni promet na TCP portu 6790. Ako se na spomenuti port pošalje samo jedan oktet (eng. *byte*) aplikacija se sruši, a ako se pošalje posebno oblikovani paket postoji mogućnost izvođenja proizvoljnog programskog koda.
- Pogreška prepisivanja spremnika je otkrivena kod "db2ckpw" usluge za provjeru korisničkih imena i zaporki. Do prepisivanja spremnika dolazi prilikom provjere korisničkog imena duljeg od osam znakova, što udaljeni zlonamjerni korisnik može iskoristiti za stjecanje potpune kontrole nad potpornim operacijskim sustavom.
- DB2 SUBP sadrži podršku za SQL jezik u vidu *Query Compiler* prevodioca unutar kojega postoje dvije ranjivosti koje mogu rezultirati ostvarivanjem DoS uvjeta. Prvi propust se javlja tijekom obrade SELECT CASE naredbe, a drugi tijekom obrade posebno oblikovanog upita koji sadrži *datetime* i *varchar* tipove podataka.

5.3.2. Enkripcija

IBM DB2 baze podataka posjeduju vlastitu enkripciju podataka tijekom prijenosa. Enkripcija datoteka korištenjem EFS datotečnog sustava je podržana kod inačica namijenjenih Windows operacijskim sustavima dok je korisnicima omogućeno vlastito izvođenje enkripcije na razini programskog sučelja.

5.3.3. Načini autorizacije

IBM DB2 omogućuje odabir između više različitih načina autorizacije korisnika:

1. SERVER,
2. SERVER_ENCRYPT,
3. CLIENT,
4. DCE,
5. DCS,
6. DCS_ENCRYPTED,
7. KERBEROS i
8. KRB_SERVER.

Načini autorizacije definiraju kada i kako se obavlja autorizacija korisnika te se postavljaju na klijentskom i poslužiteljskom računalu. Na poslužitelju se to čini pomoću *database manager configuration* datoteke koja je povezana s instancom SUBP-a. Sve baze podataka kontrolirane istom instancom dijele konfiguracijsku datoteku pa se odabrani način autorizacije primjenjuje na sve njih, kao i na sve njihove korisnike.

Ako se odabere CLIENT način autorizacije, moguće je odabrati TRUST_ALLCLNTS ili TRUST_CLNTAUTH. Ovi načini se ne preporučuju jer nije moguće ocijeniti dobronamjernost svih klijenata, a dodatno postoji i mogućnost krađe identiteta ovlaštenog korisnika. Zbog toga se preporuča odabir jednog od sigurnih načina autorizacije: SERVER_ENCRYPT, DCE_SERVER_ENCRYPT ili KRB_SERVER_ENCRYPT. Oni se smatraju sigurnima jer implementiraju enkripciju korisničkih podataka prilikom slanja preko mreže.

5.3.4. Izvorno postavljeni korisnički računi

Odmah nakon instaliranja SUBP-a potrebno je promijeniti izvorno postavljena korisnička imena i zaporke jer, u protivnom, napadač može vrlo lako zaobići postavljena sigurnosna ograničenja. IBM DB2 SUBP namijenjen Windows operacijskim sustavima sadrži korisnički račun s imenom "db2admin" i korisničkom zaporkom "db2admin". Kod inačice namijenjene UNIX operacijskim sustavima prisutni su korisnički računi s imenima (zaporkama): "db2as" ("ibmdb2"), "db2fenc" ("ibmdb2") i "db2inst1" ("ibmdb2").

5.4. Sysbase

Uz Sybase SUBP-ove koriste se dva poslužitelja. *Sybase Adaptive Server Enterprise* poslužitelj je vrlo čest u poslovnom svijetu. Koriste ga banke, burze i osiguravajuća društva. Drugi poslužitelj korišten uz

Sybase SUBP-ove je *Adaptive Server Anywhere* koji se koristi za posluživanje manjih baza podataka u slučajevima ograničenih sklopovskih resursa kao npr. kod mobitela i drugih ugrađenih aplikacija.

5.4.1. Ranjivosti

Ranjivosti za koje su izdane zakrpe:

- *Sybase Adaptive Server* poslužitelj sadrži DBCC CHECKVERIFY funkciju koja se koristi za provjeru rezultata zadnjeg pokretanja *dbcc checkstorage* funkcije. Jedini parametar kojega DBCC CHECKVERIFY funkcija prima je ime baze podataka koju treba ispitati, ali pri tome ne obavlja provjeru duljine tog znakovnog niza. Predviđeno je samo ovlašteno pokretanje DBCC CHECKVERIFY funkciju, ali kada to učini neovlašteni korisnik postoji mogućnost za prepisivanje spremnika. Do njega dolazi prije sigurnosnih provjera, što napadaču omogućuje stjecanje potpune kontrole nad samim poslužiteljem. Jednaka ranjivost se javlja i kod DROP DATABASE funkcije za uklanjanje baze podataka.
- Pogreška preljeva spremnika javlja se i unutar *xp_freedll* proširene pohranjene procedure za otpuštanje DLL datoteka koje su bile učitane od strane drugih ESP procedura. Iako se ova procedura odnosi na DLL datoteke Windows operacijskog sustava ranjivost unutar nje pogađa i Sybase baze podataka namijenjene UNIX operacijskim sustavima. *Xp_freedll* kao jedini parametar prima ime DLL datoteke koju treba otpustiti, ali ne obavlja provjeru duljine znakovnog niza. Zatim pokušava taj niz upisati u relativno mali spremnik, pri čemu dolazi do prepisivanja pokazivača na stog i samog stoga. To napadaču omogućuje izvršavanje proizvoljnog programskog koda u sigurnosnom kontekstu proširene pohranjene procedure.

5.4.2. Enkripcija

Sybase podržava SSL protokol za enkripciju mrežnog prometa i EFS datotečni sustav za enkripciju datoteka na Windows operacijskim sustavima. Za enkripciju na razini programskog sučelja koriste se proširene pohranjene procedure.

5.4.3. Operacijski sustav

Kako bi se samo ovlaštenim korisnicima omogućilo povezivanje na Sybase bazu podataka preporuča se filtriranje mrežnih paketa na poslužiteljskom računalu. Ovime se operacijski sustav na kojemu se izvodi Sybase osigurava i od sigurnosnih problema neovisnih o bazi te se osigurava mreža u slučaju uspješnog napada na Sybase SUBP. Za ovo filtriranje je dovoljno koristiti IPTables alat Linux operacijskih sustava, odnosno IPSec funkcionalnost kod Windows operacijskih sustava.
Sybase SUBP treba pokretati sa što manjim ovlastima. On zahtjeva različite ovlasti ovisno o platformi na kojoj se izvodi i o načinu uporabe. Unatoč tome, potrebno je otkriti i ukinuti one koje su nepotrebne.
Ako platforma na kojoj se Sybase izvodi to dozvoljava, također je potrebno promijeniti početni direktorij samog poslužitelja (eng. *chroot jail*). Na taj se način uvelike ograničava broj datoteka kojima Sybase proces ima pristup, što se može pokazati kao izrazito uspješna sigurnosna mjera. Također, Sybase poslužitelj treba imati samo ograničen pristup datotečnom sustavu.
Kao dodatnu mjeru predostrožnosti, uputno je korisnicima ograničiti pristup datotečnoj strukturi Sybase poslužitelja. U suprotnom postoji mogućnost preuzimanja kontrole nad SUBP-om ili neovlaštenog pristupa podacima.
Sve preporuke vezane uz operacijski sustav kod uporabe Sysbase SUBP-a vrijede i za MySQL SUBP.

5.4.4. Konfiguracija

U početnim postavkama instalacije Sybase SUBP-a nisu uključeni programski paketi potrebni za nadzor i evidenciju. Zbog toga je potrebno utrošiti određeno vrijeme na njihovu konfiguraciju.
Sybase SUBP najlakše je napasti pomoću *xp_cmdshell* proširene pohranjene procedure i zbog toga ju treba ukloniti ukoliko se ne koristi. Ako ju se ipak koristi njezin kontekst treba postaviti na vrijednost 1 što će njezino pokretanje omogućiti samo ovlaštenim korisnicima i to samo u sigurnosnom kontekstu korisnika koji ju pokreću.

Savjetuje se i onemogućavanje svih drugih funkcionalnosti Sybase SUBP-a koje se ne koriste. Najčešći primjeri takvih funkcionalnosti su podrška za Java sustave i podrška za interakciju s datotečnim sustavom.
Sybase SUBP posjeduje mogućnost integracije s *Kerberos*, *Windows NT Lan Manager* i DCE sustavima za autentikaciju korisnika. Ovi sustavi podržavaju značajno kvalitetnije upravljanje korisničkim računima te su sigurniji od autorizacijskog sustava ugrađenog u sam Sybase SUBP.

5.5. MySQL

MySQL je najviše korišten SUBP iz skupine programa otvorenog programskog koda. Njegova popularnost temelji se na mogućnosti besplatnog korištenja, podržanosti velikog broja platformi, relativnoj jednostavnosti, lakom održavanju i zadovoljavajućim performansama.

5.5.1. Ranjivosti

MySQL određuje razinu ovlasti pojedinog korisnika ovisno o računalu s kojega se spaja na MySQL SUBP. Ako se radi o računalu s lokalne mreže pretpostavljaju se maksimalne ovlasti i zbog toga lokalni napadi mogu biti puno opasniji od udaljenih.
MySQL sadrži brojne skripte koje u radu koriste privremene datoteke. U nekim slučajevima te se privremene datoteke stvaraju na nesigurnim mjestima i s predvidljivim imenima pa mogu biti zamijenjene simboličkim vezama prema kritičnim sistemskim datotekama. MySQL skripta prilikom prepisivanja sistemske datoteke koristi ovlasti MySQL procesa koji ju je pokrenuo.
Značajna ranjivost postoji kod alata WinMySQLAdmin koji u datoteci *my.ini* u tekstualnom nekriptiranom (eng. *plaintext*) formatu sprema administratorsku '*root*' zaporku.
Za razliku od drugih SUBP-ova, kao što su npr. Oracle ili Sybase, MySQL izvorno sadrži prilično slabu mrežnu podršku. Zbog toga napadač, nakon proboja u MySQL poslužitelj, nema puno mogućnosti za proširivanje napada na ostatak računalne mreže.

5.5.2. Enkripcija

MySQL komunikacija izvorno nije kripirana pa zlonamjerni korisnik koji prisluškuje vezu između klijenta i poslužitelja može doznati korisničko ime i zaporku. Kako bi se to izbjeglo potrebno je postaviti REQUIRE SSL opciju u GRANT izjavi koja se koristi prilikom povezivanja korisnika. Time se osigurava enkripcija prometa, izbjegava djelovanje značajnog broja napadačkih programskih skripti i osigurava zaštićenost zaporki.

5.5.3. Korisnici

Kod MySQL SUBP-a poznato je postojanje korisničkog računa s imenom "*root*". Nekoliko dostupnih alata, skripti i tehnika napada temelje se upravo na postojanju takvog korisničkog računa. Sa stajališta funkcionalnosti korisničko ime uopće nije bitno i zbog toga se savjetuje preimenovanje „*root*“ korisničkog računa.
Stvaranje posebnog MySQL korisnika za svaku ulogu unutar web aplikacije ograničava napadača koji je uspio probiti zaštitu određenog dijela aplikacije na ovlasti tog dijela.
Nitko osim tzv. *root* korisnika ne bi smio imati pristup *mysql.user* tablici jer to napadaču omogućuje neovlašteno stjecanje povišenih korisničkih ovlasti.

5.5.4. Konfiguracija

Mogućnost *general query log* se u dokumentaciji MySQL-a smatra alatom za pronalaženje i uklanjanje pogrešaka, ali može poslužiti i kao dio rutinskih sigurnosnih provjera. Ova mogućnost bilježi sva uspješna povezivanja i sve upite. Iako ne bilježi rezultate tih upita niti vraćene podatke može dati dobar uvid u zbivanja unutar SUBP-a. *General query log* mogućnost inicijalno nije aktivna pa se preporuča njezino pokretanje u sklopu konfiguracije MySQL poslužitelja. Pri tome treba paziti tko ima pristup dnevničkoj datoteci jer ona može sadržavati osjetljive informacije.
Savjetuje se i onemogućavanje LOAD DATA LOCAL INFILE naredbe. Ova naredba klijentima omogućuje učitavanje podataka iz lokalnog datotečnog sustava izravno u MySQL tablicu. Pod određenim okolnostima napadač može pomoću te naredbe pročitati datoteke s klijentskog računala.

Korisnički definirane funkcije mogu zlonamjernom korisniku omogućiti proširenje ovlasti nad napadnutim poslužiteljem pa ih treba ukloniti ukoliko se ne koriste. Jednako tako treba onemogućiti *skip-networking* i *skip-symbolic-links* mogućnosti ukoliko nisu potrebne.

6. Zaključak

Svi SUBP-ovi sadrže ranjivosti i nije moguće odrediti niti najsigurnijeg niti najranjivijeg među njima. Jedino je sa sigurnošću moguće tvrditi kako je najsigurniji onaj sustav koga se najbolje poznaje. Dobro poznavanje arhitekture i funkcionalnosti sustava od strane administratora, omogućuje njegov siguran rad.

Broj funkcionalnosti koje SUBP posjeduje može također biti pokazatelj njegove sigurnosti, odnosno nesigurnosti. Veći broj funkcionalnosti znači i veće mogućnosti za pojavljivanje ranjivosti, odnosno veću „površinu" izloženu napadima.

Preporuke vezane uz sigurnost baza podataka se mogu sažeti u slijedeći popis:

- korisnicima je potrebno dodjeljivati samo neophodne ovlasti,
- posebnu pažnju potrebno je posvetiti upravljanju korisničkim računima i zaporkama,
- ispravno primijenjene metode nadzora, periodičke analize i korištenje zamki mogu uvelike pomoći prilikom otkrivanja napada, a time i olakšati pronalaženje ranjivosti i njihovo uklanjanje,
- korištenje enkripcije zlonamjernim korisnicima otežava pristup osjetljivim informacijama, kako korisničkim zaporkama, tako i svim ostalim podacima pohranjenim u bazi,
- postavljanje poslužitelja s bazom podataka u unutarnju mrežu čini ga daleko sigurnijim, a primjena sustava dozvoljenih IP adresa dodatno smanjuje vjerojatnost udaljenih napada.

Za sigurnost SUBP-a je vrlo važna stalna i redovita primjena zakrpi. Informacije o uočenim nedostacima i odgovarajućim ispravkama mogu se pronaći na *web* stranicama proizvođača, stranicama tvrtki i nezavisnih organizacija koje se bave računalnom sigurnošću, ali i na hakerskim forumima. Pri tome treba uvijek voditi računa o tome da je prvi korak kod uklanjanja ranjivosti spoznavanje njenog postojanja. Iz primjera *Slammer* crva može se izvući pouka o mogućim posljedicama neinformiranosti i neprimjenjivanja zakrpa.

7. Reference

[1] S. Brian Suddeth: Database – The Final Firewall, SANS Institute, 2002,

[2] Common Vulnerabilities in Database Security, Hurwitz Group, 2001,

[3] Application Security Inc.: Database Security, A Key Component of Application Security, http://hosteddocs.ittoolbox.com/Database_Security.pdf, listopad 2006.

[4] Aaron Newman: Protecting Database, Application Security, http://appsecinc.com/presentations/ProtectingDatabases.pdf, listopad 2006.

[5] David Litchfield, Chris Anley, John Heasman, Bill Grindlay: The Database Hacker's Handbook, John Wiley & Sons, 2005,

[6] Database Security (Common-sense Principles), http://www.governmentsecurity.org/articles/DatabaseSecurityCommon-sensePrinciples.php, listopad 2006,

[7] SQL slammer (computer worm), http://en.wikipedia.org/wiki/SQL_slammer_worm, listopad 2006.