|  | **Univerzitet u Nišu, Elektronski fakultet**  **Katedra za računarstvo** |  |
| --- | --- | --- |

**Sigurnost baza podataka (*SQL Server*)**

**Seminarski rad**

Profesor: Student:

Prof. Aleksandar Stanimirović Anđela Kričak, 1498/22

**Niš, 2024. godina**

**Sadržaj**

[Uvod 3](#_Toc176000713)

[Sigurnost baza podataka 4](#_Toc176000714)

[Značaj sigurnosti podataka? 4](#_Toc176000715)

[Najčešći faktori narušavanja sigurnosti 5](#_Toc176000716)

[Mehanizmi zaštite 6](#_Toc176000717)

[Najbolje prakse 7](#_Toc176000718)

[Sigurnost kod *SQL Server* baza podataka 9](#_Toc176000719)

[Ojačavanje *SQL Servera* 9](#_Toc176000720)

[*SQL Server Firewall* 10](#_Toc176000721)

[Strategija za *backup* i oporavak od katastrofa za *SQL Server* 11](#_Toc176000722)

[Sigurnosni model *SQL Servera* 12](#_Toc176000723)

[Autentifikacija 12](#_Toc176000724)

[Autorizacija 12](#_Toc176000725)

[Šifrovanje 13](#_Toc176000726)

[Audit i praćenje aktivnosti 14](#_Toc176000727)

[Upravljanje bezbednosnim pretnjama 16](#_Toc176000728)

[Zaključak 17](#_Toc176000729)

[Literatura 19](#_Toc176000730)

# Uvod

U savremenom poslovnom okruženju, podaci predstavljaju jedno od najvrednijih sredstava organizacije. Baze podataka, kao osnovna tehnologija za skladištenje, organizaciju i pristup podacima, igraju ključnu ulogu u upravljanju ovim resursom. Sa porastom količine i složenosti podataka, kao i potrebom za njihovom stalnom dostupnošću, bezbednost baza podataka postaje sve važniji aspekt u zaštiti integriteta, poverljivosti i dostupnosti informacija.

Sigurnost baza podataka obuhvata sveobuhvatan skup strategija, tehnologija i praksi koje su osmišljene kako bi se zaštitili podaci od različitih pretnji, uključujući neovlašćen pristup, krađu podataka, manipulaciju podacima, kao i napade zlonamernih aktera. Nedovoljno obezbeđene baze podataka mogu postati meta napadača koji mogu iskoristiti slabe tačke u sistemu za pristup poverljivim informacijama, što može dovesti do ozbiljnih finansijskih gubitaka i narušavanja ugleda organizacije.

Osim spoljašnjih pretnji, baze podataka mogu biti ugrožene i od strane zaposlenih unutar organizacije, bilo kroz slučajne greške ili namerne radnje. Interni napadi su često teže uočljivi, jer uključuju osobe koje već imaju određeni nivo pristupa sistemu. Stoga, implementacija strogih kontrola pristupa i redovno praćenje aktivnosti unutar baza podataka predstavlja ključni korak u zaštiti podataka.

Pored tehničkih mera zaštite, organizacije su obavezne da se pridržavaju različitih pravnih i regulatornih okvira koji postavljaju standarde za sigurnost podataka. Ovi standardi definišu obaveze organizacija u pogledu zaštite privatnosti korisnika, očuvanja integriteta podataka i omogućavanja njihove dostupnosti. U skladu s tim, sigurnost baza podataka postaje kritičan faktor za usklađenost sa zakonskim zahtevima i zaštitu od potencijalnih pravnih posledica.

U daljem tekstu će biti razmatrani osnovni koncepti sigurnosti baza podataka, uključujući ključne elemente kao što su autentifikacija, autorizacija, šifrovanje, kontrola pristupa, kao i pretnje sa kojima se baze podataka suočavaju.

# Sigurnost baza podataka

Sigurnost baze podataka odnosi se na niz alata, kontrola i mera dizajniranih za uspostavljanje i očuvanje poverljivosti, integriteta i dostupnosti baze podataka. Od ključnog je značaja za sigurnost čitave aplikacije. U najvećem broju povreda podataka poverljivost je element koji je ugrožen.

Pod bezbednošću baze podataka podrazumeva se zaštita podataka u bazi podataka, sistema za upravljanje bazom podataka (DBMS), svih povezanih aplikacija, fizičkog servera baze podataka ili servera virtuelne baze podataka sa hardverom koji je u njenoj osnovi, računarske ili mrežne infrastrukture koja se koristi za pristup bazi podataka.

Sigurnost baze podataka je složen i izazovan poduhvat koji uključuje sve aspekte tehnologija i praksi informacione bezbednosti. Pored toga, sigurnost je prirodno u suprotnosti sa upotrebljivošću baze podataka. Što je baza podataka pristupačnija i upotrebljivija, to je ranjivija na bezbednosne pretnje; što je baza podataka neranjivija na pretnje, teže joj je pristupiti i koristiti je. Ovaj paradoks se naziva Andersonovo pravilo.

## Značaj sigurnosti podataka?

Baza podataka je ključna komponenta svakog softverskog sistema i njena uloga je u čuvanju podataka koje aplikacija koristi. Stoga povreda ovih podataka ima značajan negativan efekat na sistem. Prema definiciji, povreda podataka je neuspeh da se održi poverljivost podataka u bazi podataka. U zavisnosti od same povrede podataka šteta može biti manja ili veća ali je sigurno da predstavlja značajan problem.

Povreda podataka može prouzrokovati:

* **Kompromitovanje intelektualne svojine**

Intelektualna svojina – poslovne tajne, izumi i slično – mogu biti od kritičnog značaja za održavanje konkurentske prednosti na tržištu. Ako je ta intelektualna svojina ukradena ili razotkrivena, prednost na tržištu može biti teško ili nemoguće povratiti.

* **Narušavanje reputacije brenda**

Jednom kada je reputacija brenda ili kompanije narušena jako teško ju je povratiti. Ukoliko je poverenje klijenata iznevereno i osećaju da vam ne mogu verovati da ćete zaštititi svoje podatke moguće je da više neće biti voljni da kupe vaše proizvode ili usluge ili posluju sa vašom kompanijom.

* **Kazne**

Važno je naglasiti da kompanija koja pretrpi povredu podataka može dobiti i novčanu kaznu zbog nepoštovanja globalnih propisa.

* **Prekid rada**

Povreda podataka može dovesti do prekida rada ukoliko sistem nije u stanju da nastavi sa radom. To dalje može dovesti do velikih gubitaka.

* **Troškove otklanjanja povreda i obaveštavanja klijenata**

Povreda podataka može izazvati velike troškove. Pored samih troškova popravke nastale povrede podataka i štete, popravku povređenih sistema, potrebno je i obaveštavanje klijenata o nastalom stanju, može biti potrebno i platiti forenzičke i istražne aktivnosti, upravljanje kriznim situacijama i još mnogo toga. Generalno povreda podataka može prouzrokovati jako velike troškove.

## Najčešći faktori narušavanja sigurnosti

Mnogo je razloga zbog kojih dolazi do narušavanja sigurnosti baze podataka. Najčešće pretnje i izazovi u očuvanju sigurnosti baze podataka su sledeći:

* **Unutrašnje pretnje**

Unutrašnja pretnja je bezbednosna pretnja iz bilo kog od tri izvora sa privilegovanim pristupom bazi podataka:

1. Zlonamerni insajder koji namerava da učini štetu.
2. Nemaran insajder koji pravi greške koje čine bazu podataka podložnom napadima.
3. Infiltrator, autsajder koji na neki način dobija kredencijale putem šeme, kao što je *fishing* ili dobijanjem pristupa samoj bazi podataka kredencijala.

Unutrašnje pretnje su među najčešćim uzrocima narušavanja bezbednosti baze podataka i često su rezultat dozvoljavanja prevelikom broju zaposlenih da imaju privilegovane kredencijale za korisnički pristup.

* **Ljudska greška**

Nesrećni slučajevi, slabe lozinke, deljenje lozinki i druga nemarna i slična ponašanja korisnika su uzrok skoro polovine (49%) svih prijavljenih povreda podataka.

* **Eksploatacija ranjivosti baze podataka**

Hakeri žive od toga što ciljaju i pronalaze ranjivosti u svim vrstama softvera, uključujući softver za upravljanje bazama podataka. Svi glavni dobavljači softvera za komercijalne baze podataka i platforme za upravljanje bazama podataka otvorenog koda izdaju redovna bezbednosna rešenja da bi se pozabavile ovim ranjivostima, ali ako se ne primene na vreme to povećava izloženost napadima.

* ***SQL* ili *NoSQL injection* napadi**

Ovo je pretnja koja je specifična za bazu podataka. Ona uključuje umetanje proizvoljnih *SQL* ili *NoSQL* stringova u upite baze podataka koje opslužuju *web* aplikacije ili *HTTP* zaglavlja. Organizacije koje ne prate bezbedne prakse kodiranja i ne vrše redovno testiranje ranjivosti otvorene su za ove napade.

* **Eksploatacija prekoračenja bafera**

Prekoračenje bafera se dešava kada proces pokuša da upiše više podataka nego što je dozvoljeno u blok memorije fiksne dužine. Napadači mogu koristiti višak podataka, koji se čuvaju na susednim memorijskim adresama, kao osnovu za početak napada.

* **Zlonamerni programi**

Zlonamerni program (*Malware*) je softver koji je posebno napisan s namerom da iskoristi ranjivosti ili na drugi način ošteti bazu podataka. Malver može da stigne preko bilo kog krajnjeg uređaja koji se povezuje na mrežu baze podataka.

* **Napadi na rezervne kopije**

U slučajevima kada rezervne kopije nisu zaštićene u onoj meri u kojoj je to obezbeđeno primarnoj bazi podataka, rezervne kopije postaju podložnije napadima. Ove pretnje pogoršavaju sledeći faktori: rastući obim podataka, širenje infrastrukture, sve stroži regulatorni zahtevi i nedostatak veština u sajber bezednosti.

* ***DoS* i *DDoS* napadi**

U napadima uskraćivanja usluga(Denial-of-Service – DoS), napadač zasipa ciljni server, u ovom slučaju server baze podataka, sa toliko zahteva da server više ne može da ispunjava legitimne zahteve stvarnih korisnika i često server postaje nestabilan ili se ruši.

U distribuiranom napadu uskraćivanja usluga(Distributed Denial-of-Service – DDoS), plavljenje dolazi sa više servera, što otežava zaustavljanje napada.

## Mehanizmi zaštite

Baza podataka mora biti zaštićena od neovlašćenih ili zlonamernih radnji. Kao što je već naglašeno, ona je deo celokupne zaštite. Stoga je potrebno voditi računa i o sigurnosti operativnog sistema, mreže itd. kako bi se obezbedila adekvatna zaštita čitavog softvera.

Za očuvanje sigurnosti baze podataka postoje ugrađeni mehanizmi u sistemima za upravljanje bazom podataka.

Za korisnike koji imaju fizički pristup neophodna je identifikacija korisnika. Odnosno autentifikacija korisnika nekom metodom, kako bi se osiguralo da je u pitanju odgovarajući korisnik.

Kontrola pristupa mora postojati kako bi svojstva koja definišu sigurnost baze podataka bila ispunjena. Stoga korisnici prostupaju podacima u bazi podataka u zavisnosti od ovlašćenja i prava pristupa koja imaju.

Politika pristupa specifira ko je autorizovan da vrši određene akcije. Mehanizam sigurnosti dozvoljava sprovođenje izabrane pollitike sigurnosti. Osnovni mehanizmi zaštite na nivou DBMS-a su: diskreciona kontrola pristupa(Discretionary access control – DAC), obavezna kontrola pristupa (Mandatory access control – MAC) i kontrola pristupa zasnovana na ulogama (Role-Based access control - RBAC).

Diskreciona kontrola pristupa se zasniva na konceptu prava pristupa ili privilegija za objekte (tabele i poglede) i mehanizmima dodele/oduzimanja prava korisnicima. Onaj korisnik koji je kreator objekta automatski dobija sve privilegije nad kreiranim objektom. DBMS čuva informacije o tome kad dobija i gubi privilegije i obezbeđuje da se prihvate zahtevi samo od korisnika koji ima odgovarajuće privilegije za posmatrani objekat (u trenutku slanja zahteva).

Obavezna kontrola pristupa zasniva se na politikama sigurnosti na nivou celog sistema koji ne može da promeni neki od korisnika. Svaki objekat baze podataka ima dodeljenu tzv. klasu sigurnosti (*security class*). Svaki subjekat (korisnik ili korisnička aplikacija) ima dodeljenu dozvolu (*clearance*) za klasu sigurnosti. Pravila koja se zasnivaju na klasama sigurnosti i dozvolama upravljaju ko može da izvrši operacije čitanja/pisanja za objekte.

Kontrola pristupa zasnovana na ulogama daje privilegije pristupa na osnovu posla koji obavljaju pojedinačni korisnici. Ograničava pristup tako da korisnik može da pristupi samo onim resursima koji su mu potrebni za obavljanje posla. Implementacija ovog mehanizma zahteva definisanje različitih uloga unutar organizacije i određivanje da li i u kom stepenu te uloge treba da imaju pristup svakom resursu.

## Najbolje prakse

S obzirom da su baze podataka dostupne preko mreže, svaka bezbednosna pretnja unutar mreže ili dela mrežne infrastrukture je takođe pretnja bazi podataka, a svaki napad koji utiče na uređaj ili radnu stanicu korisnika može da ugrozi bazu podataka. Dakle, bezbednost baze podataka se proteže daleko izvan granica same baze podataka.

U cilju očuvanja bezbednosti baze podataka važno je razmotriti sledeće oblasti:

* **Fizička sigurnost**

Bez obzira da li se server baze podataka nalazi lokalno ili je u oblaku, on mora biti lociran u bezbednom okruženju sa kontrolisanim klimatskim uslovima.

* **Administrativne i mrežne kontrole pristupa**

Pristup bazi podataka treba da ima minimalni broj korisnika, a njihove dozvole treba da budu ograničene na minimalne nivoe neophodne da bi obavljali svoj posao. Takođe, pristup mreži treba da bude ograničen na minimalni nivo neophodnih dozvola.

* **Bezbednost korisničkih naloga i uređaja**

Potrebno je biti svestan ko pristupa bazi podataka i kada i kako se podaci koriste. Rešenja za praćenje podataka mogu vas upozoriti ako su aktivnosti sa podacima neobične ili izgledaju rizične. Svi korisnički uređaji koji se povezuju na mrežu u kojoj se nalazi baza podataka treba da budu fizički bezbedni podvrgnuti bezbednosnim kontrolama u svakom trenutku.

* **Enkripcija**

Svi podaci, uključujući podatke koji su u bazi podataka i podatke o kredencijalima korisnika treba da budu zaštićeni enkripcijom – i dok su u mirovanju i dok su u tranzitu.

* **Sigurnost softvera baze podataka**

Poželjno je koristiti najnoviju verziju softvera za upravljanje bazom podataka i ažurirati sve bezbednosne popravke čim postanu dostupne.

* **Sigurnost aplikacija i *web* servera**

Svaka aplikacija ili veb server koji stupa u interakciju sa bazom podataka može biti kanal za napad i trebalo bi da bude podvrgnut stalnom testiranju.

* **Sigurnost rezervnih kopija**

Sve rezervne kopije, kopije ili slike baze podataka moraju biti podvrgnute istim ili jednako strogim bezbednosnim kontrolama kao i sama baza podataka.

* **Revizija**

Zabeležiti sve prijave na server baze podataka i operativni sistem i evidentirajte sve operacije koje se izvode na osetljivim podacima. Takođe, potrebno je obavljati revizije standarda bezbednosti baze podataka.

# Sigurnost kod *SQL Server* baza podataka

*SQL Server* je jedan od najpopularnijih sistema za upravljanje relacijskim bazama podataka, razvijen od strane *Microsofta*, koji nudi širok spektar funkcionalnosti za zaštitu podataka. Sigurnost u *SQL Serveru* je implementirana kroz nekoliko slojeva koji zajedno obezbeđuju sveobuhvatnu zaštitu podataka, od ojačanja *SQL Servera*, *SQL Server Firewall*-a, strategije za *backup* i oporavak od katastrofa, *auditing* i nadzor, sigurnosni model, enkripciju u *SQL Serveru*.

## Ojačavanje *SQL Servera*

Ojačavanje servera je proces osiguravanja *SQL Servera* smanjenjem površine za napad i minimiziranjem potencijalnih ranjivosti. Evo ključnih koraka koje možete preduzeti da ojačate svoj SQL Server:

* **Primena sigurnosnih ažuriranja:** Redovno ažurirajte *SQL Server* sa najnovijim sigurnosnim zakrpama i ažuriranjima. Redovno proveravajte dostupnost novih ažuriranja i primenjujte ih što pre kako biste se zaštitili od poznatih ranjivosti.
* **Onemogućavanje nepotrebnih servisa i protokola:** Onemogućite sve nepotrebne servise i protokole koji bi mogli biti iskorišćeni za neovlašćen pristup *SQL Serveru*. Ovo može uključivati protokole kao što su *SMB*, *FTP* i *Telnet*, kao i servise kao što su S*QL Server Browser* i *SQL Server Agent*.

-- Onemogućavanje servisa SQL Server Browser ako nije potreban

EXEC xp\_cmdshell 'net stop "SQL Server Browser"';

-- Onemogućavanje opcije xp\_cmdshell zbog sigurnosnih razloga

EXEC sp\_configure 'show advanced options', 1;

RECONFIGURE;

EXEC sp\_configure 'xp\_cmdshell', 0;

RECONFIGURE;

* **Onemogućavanje nepotrebnih funkcija:** Onemogućite sve funkcije i konfiguracije koje ne koristi aplikacija ili baza podataka. Na primer, onemogućite ***xp\_cmdshell*** jer omogućava pristup operativnom sistemu preko *SQL Servera*.

-- Onemogućavanje funkcije xp\_cmdshell

EXEC sp\_configure 'show advanced options', 1;

RECONFIGURE;

EXEC sp\_configure 'xp\_cmdshell', 0;

RECONFIGURE;

* **Ograničavanje dozvola za servise:** Ograničite dozvole za *SQL Server* servise samo na one koje su neophodne za rad SQL Servera. Koristite namenski nalog za pokretanje SQL Server servisa i primenjujte nalog servisa preko SQL Server Configuration Manager-a, jer će to automatski dodeliti potrebne dozvole SQL Serveru.
* **Korišćenje enkripcije:** Koristite enkripciju za zaštitu osetljivih podataka, kako u stanju mirovanja, tako i u tranzitu. Ovo može uključivati korišćenje ***Transparent Data Encryption (TDE)*** za enkripciju cele baze podataka ili ***Transport Layer Security (TLS)*** za enkripciju podataka u tranzitu.

## *SQL Server Firewall*

*Firewall* je važna komponenta sigurnosti *SQL Servera*, jer može pomoći u zaštiti baze podataka od neovlašćenog pristupa i potencijalnih napada. Evo nekoliko najboljih praksi za konfiguraciju *firewalla* za *SQL Server*:

* **Omogućavanje Windows Firewalla:** *Windows Firewall* je ugrađeni *firewall* koji može pomoći u zaštiti sistema od neovlašćenog pristupa. Potrebno je uveriti se da je *Windows Firewall* omogućen i konfigurisan da blokira sve dolazne veze po *defaultu*.

*Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled True*

* **Konfiguracija *SQL Server* porta:** Podrazumevano, *SQL Server* sluša na portu 1433 za dolazne veze. Promena broja porta na neku vrednost koja nije podrazumevana je poželjna, kako bi napadačima bilo teže da pronađu i ciljaju *SQL Server*.

*Set-ItemProperty -Path 'HKLM:\SOFTWARE\Microsoft\Microsoft SQL Server\MSSQL12.MSSQLSERVER\MSSQLServer\SuperSocketNetLib\Tcp\IPAll' -Name 'TcpPort' -Value '1500'*

* **Ograničavanje dolaznih veza:** Konfigurišite *firewall* da dozvoljava samo dolazne veze sa pouzdanih IP adresa ili mreža. Ovo može pomoći u zaštiti *SQL Servera* od napada koji potiču sa interneta.

*New-NetFirewallRule -DisplayName "Allow SQL Server" -Direction Inbound -Protocol TCP -LocalPort 1500 -RemoteAddress 192.168.1.0/24 -Action Allow*

* **Korišćenje grupe sigurnosti mreže:** Ako je *SQL Server* hostovan u oblaku, korišćenje grupne sigurnosti mreže (*NSG*) za kontrolu dolaznog i odlaznog saobraćaja je dobra opcija. *NSG* može pomoći u ograničavanju pristupa *SQL Serveru* na osnovu opsega IP adresa i protokola.

*az network nsg rule create --resource-group MyResourceGroup --nsg-name MyNSG --name AllowSQL --priority 1000 --source-address-prefixes "192.168.1.0/24" --destination-port-ranges 1500 --protocol Tcp --access Allow*

* **Praćenje logova *firewalla*:** Pratiti logove *firewalla* za sumnjive aktivnosti i pokušaje neovlašćenog pristupa. Ovo može pomoći u otkrivanju i odgovaranju na potencijalne sigurnosne povrede.

*Set-NetFirewallProfile -LogAllowed True -LogBlocked True*

## Strategija za *backup* i oporavak od katastrofa za *SQL Server*

Imati solidnu strategiju za backup i oporavak od katastrofa je ključno za sigurnost *SQL Servera*. Evo nekoliko ključnih koraka koje je moguće preduzeti kako bi osigurali sigurnost *SQL* *Server backup*-a i planova za oporavak od katastrofa:

* **Redovni *backup*:** Postaviti redovan raspored za *backup SQL Server* baze podataka. Ovo bi trebalo da uključuje potpune *backup*-ove, diferencijalne *backup*-ove i *backup*-ove transakcionih logova. Potrebno je skladištiti *backup*-ove na sigurnom mestu, kao što je zaseban server ili *cloud-based* rešenje za skladištenje.

-- Kreiranje potpune rezervne kopije baze podataka

BACKUP DATABASE [VašaBazaPodataka] TO DISK = N'D:\Backups\VašaBazaPodataka.bak' WITH NOFORMAT, NOINIT, NAME = N'Full Backup', SKIP, NOREWIND, NOUNLOAD, STATS = 10;

-- Kreiranje diferencijalne kopije

BACKUP DATABASE [VašaBazaPodataka] TO DISK = N'D:\Backups\VašaBazaPodataka\_diff.bak' WITH DIFFERENTIAL;

* **Verifikacija *backup*-a:** Redovno proveravati *backup*-ove, kako bi osigurali da se mogu uspešno povratiti. Testirati proces povratka i uveriti se da *backup*-ovi sadrže sve potrebne podatke.

-- Provera validnosti rezervne kopije

RESTORE VERIFYONLY FROM DISK = N'D:\Backups\VašaBazaPodataka.bak';

* **Sigurnost *backup*-a:** Šifrovati *backup*-ove kako bi bili zaštićeni od neovlašćenog pristupa. Koristiti snažne lozinke i čuvajti ih na sigurnom mestu. Ograničite pristup *backup*-ovima samo na ovlašćeno osoblje.

-- Šifrovanje rezervne kopije

BACKUP DATABASE [VašaBazaPodataka] TO DISK = N'D:\Backups\VašaBazaPodataka.bak' WITH ENCRYPTION(ALGORITHM = AES\_256, SERVER CERTIFICATE = MyServerCert);

* **Plan za oporavak od katastrofa:** Razviti plan za oporavak od katastrofa koji opisuje korake koje treba preduzeti u slučaju katastrofe, kao što je otkazivanje servera ili oštećenje podataka. Plan bi trebalo da uključuje vremenski okvir za vraćanje podataka i ponovnu uspostavu sistema.
* **Testiranje i ažuriranje:** Redovno testirati plan za oporavak od katastrofa kako bi osigurali da i dalje funkcioniše kako je predviđeno. Plan ažurirati sa najnovijim sigurnosnim zakrpama i ažuriranjima.

*Test-RestoreDatabase -DatabaseName 'VasaBazaPodataka' -BackupFilePath 'D:\Backups\VasaBazaPodataka.bak'.*

# Sigurnosni model *SQL Servera*

## ****Autentifikacija****

*SQL Server* podržava dva osnovna modela autentifikacije: *Windows* autentifikaciju i *SQL Server* autentifikaciju. *Windows* autentifikacija koristi *Windows* naloge i grupe, omogućavajući centralizovano upravljanje identitetima i pristupom. *SQL Server* autentifikacija, s druge strane, koristi *SQL* naloge koji su specifični za *SQL Server*, omogućavajući veću fleksibilnost u scenarijima gde *Windows* autentifikacija nije moguća.

* ***Windows* autentifikacija**: Koristi *Active Directory* za upravljanje nalozima, što omogućava lakše upravljanje korisnicima i grupama. Prednost ovog modela je u tome što se koristi postojeća infrastruktura za kontrolu pristupa.
* **SQL Server autentifikacija**: Ovaj model omogućava kreiranje lokalnih *SQL* naloga, koji su korisni u okruženjima gde nije moguće koristiti *Windows* autentifikaciju. *SQL Server* autentifikacija zahteva snažne lozinke i druge mere zaštite kako bi se minimizovao rizik od neovlašćenog pristupa.

CREATE LOGIN Korisnik1 WITH PASSWORD = 'JakaLozinka123!';

CREATE USER Korisnik1 FOR LOGIN Korisnik1;

Ovim kodom kreiramo korisnika Korisnik1 sa *SQL Server* autentifikacijom, gde je lozinka *JakaLozinka123!*. Nakon toga, korisnik se povezuje sa bazom podataka.

## ****Autorizacija****

Nakon uspešne autentifikacije, *SQL Server* koristi sistem dozvola i rola kako bi kontrolisao pristup korisnika različitim objektima unutar baze podataka. Ovaj sistem omogućava preciznu kontrolu nad tim ko može videti, menjati ili brisati podatke.

* **Dozvole**: *SQL Server* koristi granularan sistem dozvola koji omogućava dodeljivanje specifičnih prava korisnicima na nivoima objekata kao što su tabele, kolone ili procedure.

GRANT SELECT ON dbo.Tabela1 TO Korisnik1;

Ovim kodom dodeljujemo dozvolu za čitanje (*SELECT*) podataka iz tabele *Tabela1* korisniku *Korisnik1*.

* **Role**: Role predstavljaju skup dozvola koje se mogu dodeliti jednom ili više korisnika. *SQL Server* nudi unapred definisane role, kao što su db\_owner, db\_datareader i db\_datawriter, koje pokrivaju najčešće potrebe, ali takođe omogućava kreiranje prilagođenih rola.

CREATE ROLE CustomRole;

GRANT SELECT, INSERT, UPDATE ON dbo.Tabela1 TO CustomRole;

EXEC sp\_addrolemember 'CustomRole', 'Korisnik1';

## ****Šifrovanje****

*SQL Server* nudi nekoliko metoda šifrovanja za zaštitu podataka u mirovanju i tokom prenosa. Ovo obuhvata šifrovanje celokupne baze podataka, specifičnih kolona, kao i komunikacije između klijenta i servera.

* ***Transparent Data Encryption (TDE)***: *TDE* omogućava šifrovanje celokupne baze podataka, uključujući sve podatke, log fajlove i datoteke za rezervne kopije, čime se osigurava da su podaci zaštićeni čak i u slučaju fizičkog pristupa fajlovima.

-- Kreiranje ključa za šifrovanje baze podataka

CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'TDELozinka123!';

CREATE CERTIFICATE CertifikatTDE WITH SUBJECT = 'Certifikat za TDE';

CREATE DATABASE ENCRYPTION KEY WITH ALGORITHM = AES\_256 ENCRYPTION BY SERVER CERTIFICATE CertifikatTDE;

-- Omogućavanje šifrovanja baze podataka

ALTER DATABASE [ImeBaze] SET ENCRYPTION ON;

Ovim kodom omogućava se *Transparent Data Encryption (TDE)* za bazu podataka *ImeBaze*, koristeći *AES-256* algoritam za šifrovanje.

* ***Always Encrypted***: *Always Encrypted* omogućava šifrovanje specifičnih kolona unutar baze podataka, tako da su podaci šifrovani čak i za administratore baze podataka. Ključ za šifrovanje je smešten na klijentskoj strani, čime se dodatno povećava sigurnost.

-- Kreiranje kolone sa Always Encrypted

CREATE COLUMN MASTER KEY CMK\_Primer

WITH

KEY\_STORE\_PROVIDER\_NAME = 'MSSQL\_CERTIFICATE\_STORE',

KEY\_PATH = 'CurrentUser/My/KljučaoniyCertifikat';

CREATE COLUMN ENCRYPTION KEY CEK\_Primer

WITH VALUES

(

COLUMN\_MASTER\_KEY = CMK\_Primer,

ALGORITHM = 'RSA\_OAEP'

);

CREATE TABLE Osobe

(

ID int PRIMARY KEY,

Ime nvarchar(50),

Prezime nvarchar(50),

JMBG nvarchar(13) COLLATE Latin1\_General\_BIN2 ENCRYPTED WITH

(

ENCRYPTION\_TYPE = Randomized,

ALGORITHM = 'AEAD\_AES\_256\_CBC\_HMAC\_SHA\_256',

COLUMN\_ENCRYPTION\_KEY = CEK\_Primer));

Ovim kodom kreiramo tabelu *Osobe* u kojoj je kolona *JMBG* šifrovana koristeći *Always Encrypted* tehnologiju, sa nasumičnim šifrovanjem (*Randomized Encryption*).

* ***SSL/TLS* enkripcija**: *SQL Server* podržava šifrovanje komunikacije između klijenta i servera korišćenjem *SSL/TLS* protokola, čime se osigurava da podaci ne mogu biti presretnuti ili modifikovani tokom prenosa.

**Prvi korak u omogućavanju *SSL/TLS* enkripsije bi predstavljao instalaciju sertifikata (sertifikat treba da bude instaliran u „*Personal*“ (Lični) deo *Windows Certificate Store*-a za računar), zatim drugi korak podrazumeva** povezivanje *SQL Servera* sa *SSL* sertifikatom (otvoriti *SQL Server Configuration Manager* > *SQL Server Network Configuration* > *Protocols for SQL Server Instance Name* i omogućiti *SSL* enkripciju klikom na *Protocols* *for [SQL Server Instance Name]* izabrati *Properties* i postavljanjem opcije *Forc Encrtption* na *Yes*, u *Flags* tabu, na kraju povezati sertifikata sa *SQL Serverom* izborom željenog serifikata u tabu *Certificate*). Treći korak je podešavanje klijentskih aplikacija za *SSL/TLS* enkripciju:

AServer=myServerAddress;Database=myDataBase;User Id=myUsername;

Password=myPassword;Encrypt=true;TrustServerCertificate=true;

Poslednji korak podrazumeva proveru *SSL/TLS* enkripcije. Provera na klijentskoj strani podrazumeva - povezati se na *SQL Server* koristeći klijentski alat *SQL Server Management Studio*, u prozoru *Connection Properties*, proveriti da li je opcija *Encrypt connection* označena. Provera na serveru podrazumeva proveru u *SQL Server* logovima, da li je veza enkriptovana. Očekivana poruka bi trebala da sadrži podatke o uspešno uspostavljenoj *SSL/TLS* konekciji.

## ****Audit i praćenje aktivnosti****

*SQL Server* omogućava detaljno praćenje i auditiranje svih aktivnosti unutar baze podataka, što je ključno za otkrivanje sumnjivih aktivnosti i usklađivanje sa pravnim zahtevima.

* ***SQL Server Audit***: *SQL Server Audit* omogućava praćenje specifičnih događaja i aktivnosti, kao što su pokušaji pristupa određenim objektima ili promene u strukturi baze podataka. Ovi podaci se mogu koristiti za detekciju neovlašćenih aktivnosti i za analizu sigurnosnih incidenata.

-- Kreiranje audit specifikacije

CREATE SERVER AUDIT Audit\_Primer

TO FILE (FILEPATH = 'C:\SQLAudit\' );

ALTER SERVER AUDIT Audit\_Primer WITH (STATE = ON);

-- Kreiranje audit specifikacije na nivou baze podataka

CREATE DATABASE AUDIT SPECIFICATION DB\_Audit\_Primer

FOR SERVER AUDIT Audit\_Primer

ADD (SELECT ON OBJECT::dbo.Tabela1 BY [Korisnik1]);

ALTER DATABASE AUDIT SPECIFICATION DB\_Audit\_Primer WITH (STATE = ON);

Ovim kodom konfiguriše se *SQL Server Audit* za praćenje *SELECT* upita na tabeli *Tabela1* od strane korisnika *Korisnik1*. Logovi se čuvaju u datoteci na C:\SQLAudit\.

* **Praćenje promena (*Change Data Capture*)**: omogućava praćenje promena u podacima u specifičnim tabelama, beležeći umetanja (*INSERT*), izmene (*UPDATE*) i brisanja (*DELETE*). *CDC* zapisuje promene u posebne tabele, koje možete koristiti za praćenje istorije promena podataka.

USE [ImeBaze];

EXEC sys.sp\_cdc\_enable\_db;

Ovim kodom omogućava se *CDC* funkcionalnost na nivou baze podataka *ImeBaze*. Zatim se omogućava *CDC* za tabelu *Tabela1* u dbo šemi. Ako *@role\_name* nije definisano, svi korisnici će moći da pristupe podacima u *CDC* tabelama:

USE [ImeBaze];

EXEC sys.sp\_cdc\_enable\_table

@source\_schema = N'dbo',

@source\_name = N'Tabela1',

@role\_name = NULL;

Nakon što je *CDC* omogućen, *SQL Server* automatski kreira nekoliko sistemskih tabela i funkcija za praćenje promena. Promene u tabeli Tabela1 će se zabeležiti u *CDC* tabeli koja obično ima naziv poput *cdc.dbo\_Tabela1\_CT*.

Promene je moguće podgledati koristeći sledeći upit:

SELECT \*

FROM cdc.dbo\_Tabela1\_CT

ORDER BY \_\_$start\_lsn;

Ovaj upit vraća sve promene na tabeli *Tabela1*, uključujući informacije o tipu promene (*INSERT, UPDATE, DELETE*), vremenu promene, i samim podacima pre i posle promene.

Kako bi se onemogućio*CDC* na tabeli ili celoj bazi podataka, moguć je koristiti sledeće upite:

USE [ImeBaze];

EXEC sys.sp\_cdc\_disable\_table

@source\_schema = N'dbo',

@source\_name = N'Tabela1',

@capture\_instance = N'dbo\_Tabela1';

Ovim se onemogućava *CDC* na tabeli *Tabela1*.

USE [ImeBaze];

EXEC sys.sp\_cdc\_disable\_db;

Ovim se onemogućava *CDC* na celoj bazi podataka.

* ***SQL Server Management Studio (SSMS) Activity Monitor*:** Ovo je grafički alat koji vam omogućava da pratite aktivnosti *SQL Server* instance, uključujući aktivne procese, zaključavanja i sesije. Ovaj alat možete koristiti za identifikaciju uskih grla u performansama i potencijalnih sigurnosnih pretnji.
* **Alati trećih strana za nadzor:** Postoji nekoliko alata trećih strana koji pružaju naprednije mogućnosti za sigurnosni nadzor, kao što su praćenje u realnom vremenu, otkrivanje pretnji i analitika ponašanja korisnika. *WISdom*, novo rešenje za nadzor kompanije *Fortified*, poseduje mnoge od ovih funkcija zajedno sa sigurnosnim nalazima u realnom vremenu, koji pružaju radnje za *DBA* da bolje zaštiti svoje *SQL Server* okruženje.

## ****Upravljanje bezbednosnim pretnjama****

*SQL Server* takođe nudi alate za prepoznavanje i ublažavanje bezbednosnih pretnji, kao što su *SQL* injekcije i napadi na servis.

* ***Dynamic Data Masking (DDM)***: *Dynamic Data Masking* omogućava sakrivanje osetljivih podataka prilikom prikazivanja, čime se smanjuje rizik od slučajnog otkrivanja poverljivih informacija.

ALTER TABLE Osobe ALTER COLUMN JMBG ADD MASKED WITH (FUNCTION = 'partial(0,"XXXXXX",4)');

Ovim kodom primenjuje se *Dynamic Data Masking (DDM)* na kolonu *JMBG* u tabeli *Osobe*, tako da se prikazuju samo poslednje četiri cifre, dok su ostale maskirane.

* ***SQL Injection* zaštita***: SQL Server* podržava parametarske upite i strogu kontrolu unosa podataka, čime se smanjuje rizik od *SQL* injekcija.

DECLARE @KorisnikID int = 1;

DECLARE @SQLQuery nvarchar(max);

SET @SQLQuery = N'SELECT \* FROM Korisnici WHERE ID = @KorisnikID';

EXEC sp\_executesql @SQLQuery, N'@KorisnikID int', @KorisnikID;

Ovim kodom se koristi parametarski upit za izvršavanje *SELECT* upita na tabeli *Korisnici*, čime se smanjuje rizik od *SQL* injekcija.

# Zaključak

Sigurnost baza podataka u *SQL Serveru* predstavlja ključni aspekt u zaštiti podataka, koji su od suštinskog značaja za poslovanje. Kroz sveobuhvatne mere zaštite, *SQL Server* nudi robustan sistem sigurnosti, koji omogućava kontrolu pristupa, zaštitu podataka i praćenje svih aktivnosti unutar baze podataka. Kao jedan od najvažnijih alata za upravljanje podacima, SQL Server implementira više nivoa zaštite kako bi se osigurali integritet, poverljivost i dostupnost podataka.

*SQL Server* pruža napredne funkcionalnosti za autentifikaciju i autorizaciju, omogućavajući administratorima da definišu ko može pristupiti bazi podataka i koje akcije sme da preduzima. Korisnici mogu biti autentifikovani kroz *Windows* ili *SQL Server* autentifikaciju, dok se njihove dozvole precizno podešavaju na nivou pojedinačnih objekata ili celih šema. Ova granularnost omogućava preciznu kontrolu pristupa i minimizaciju rizika od neovlašćenog pristupa podacima.

Šifrovanje je još jedan kritičan aspekt sigurnosti u *SQL Serveru*. Korišćenjem tehnologija kao što su *Transparent Data Encryption (TDE)* i *Always Encrypted*, *SQL Server* omogućava šifrovanje podataka u miru i u pokretu, čime se osigurava da podaci ostanu zaštićeni čak i ako padnu u pogrešne ruke. Ove metode šifrovanja pružaju dodatni sloj zaštite, čuvajući poverljive informacije kao što su lični podaci korisnika ili finansijski zapisi.

*Auditing* i praćenje aktivnosti su neophodni za održavanje sigurnosti i usklađenosti sa regulatornim zahtevima. *SQL Server* omogućava detaljno praćenje svih akcija unutar baze podataka, beležeći ko je pristupio kojim podacima i kada. Ove informacije su dragocene ne samo za otkrivanje potencijalnih pretnji, već i za analizu i forenzičko ispitivanje u slučaju sigurnosnih incidenata.

Upravljanje bezbednosnim pretnjama takođe igra značajnu ulogu u zaštiti *SQL Server* baza podataka. Implementacijom tehnologija kao što su *Dynamic Data Masking (DDM)* i zaštita od *SQL* injekcija kroz parametarske upite, *SQL Server* se efikasno bori protiv potencijalnih napada. Ove mere omogućavaju administratorima da proaktivno štite podatke i minimiziraju rizike od kompromitovanja sistema.

Konačno, *SSL/TLS* enkripcija omogućava sigurno prenošenje podataka između klijenata i servera, čime se sprečava presretanje ili modifikacija podataka tokom prenosa. Ova tehnologija osigurava da se svi podaci razmenjuju preko zaštićenog kanala, što je od ključne važnosti za održavanje poverljivosti podataka.

Sigurnost baza podataka u *SQL Serveru* je kompleksna tema koja zahteva multidisciplinarni pristup. Kroz kombinaciju različitih bezbednosnih mehanizama, *SQL Server* pruža sveobuhvatnu zaštitu podataka, osiguravajući da su podaci zaštićeni od neovlašćenog pristupa, manipulacije i gubitka. Uvođenjem i pravilnom primenom ovih mera, organizacije mogu značajno smanjiti rizik od sigurnosnih incidenata i osigurati pouzdanost i integritet svojih podataka.

U današnjem poslovnom okruženju, gde su podaci postali najvrednija imovina, osiguranje baze podataka nije samo tehnička potreba, već i strateška imperativ. *SQL Server*, sa svojim naprednim sigurnosnim funkcijama, predstavlja jedan od najpouzdanijih alata za postizanje tog cilja.

# Literatura

<https://www.ibm.com/topics/database-security>

[*https://learn.microsoft.com/en-us/sql/relational-databases/security/securing-sql-server?view=sql-server-ver16*](https://learn.microsoft.com/en-us/sql/relational-databases/security/securing-sql-server?view=sql-server-ver16)

[*https://satoricyber.com/sql-server-security/sql-server-security/*](https://satoricyber.com/sql-server-security/sql-server-security/)

[*https://www.upguard.com/blog/11-steps-to-secure-sql*](https://www.upguard.com/blog/11-steps-to-secure-sql)

[*https://www.fortifieddata.com/enhance-microsoft-sql-server-security/*](https://www.fortifieddata.com/enhance-microsoft-sql-server-security/)