Univerzitet u Nišu
Elektronski fakultet
Katedra za računarstvo

Sigurnost MongoDB baze podataka

Mentor:

Prof. dr Aleksandar Stanimirović

Student:

Draginja Anđelković, 1028

Niš, maj 2021.

Sadržaj:

1. Uvod

Poslednjih godina sve je zastupljeniji organizovani sajber kriminal i napadi hakera su sve češći. Stoga je povećana zabrinutost oko privatnosti podataka, a potreba za osiguravanjem pristupa podacima nikada nije bila veća.

Statistika je alarmantna. Sa povećanjem uticaja i troškova kršenja privatnosti podataka, organizacije moraju da naprave holističku strategiju bezbednosti koja obuhvata korisnike i aplikacije, baze podataka i njihovo fizičko okruženje, uz stalno praćenje i reviziju, kako bi se umanjile pretnje bezbednosti informacija.

Sigurnost podataka i privatnost je krucijalna briga u današnjem povezanom svetu. Prikupljeni i analizirani podaci sa društvenih mreža, mobilnih uređaja i senzorskih mreža postali su podjednako osetljivi kao i tradicionalni podaci generisani iz sistema internog poslovanja (engl. *back-office systems*). Iz tog razloga, tehnologije velikih količina podataka (engl. *big data technologies*) moraju da se razvijaju kako bi zadovoljile standarde usklađenosti propisa koje zahtevaju industrija i Vlada.

Zahvaljujući naprednim bezbedonosnim funkcijama dostupnim u *MongoDB Enterprise Advanced*, organizacije imaju široke mogućnosti za odbranu, otkrivanje i kontrolu pristupa vrednim velikim količinama podataka na mreži.

Prvo poglavlje ovog rada je uvod u obrađenu temu, Sigurnost MongoDB baze podataka.

U drugom poglavlju ovog rada biće reči o opštim bezbednosnim zahtevima, upravljanju korisničkim pravima, pre svega autentifikaciji i autorizaciji. U nastavku drugog poglavlja biće reči o reviziji i šifrovanju kod MongoDB baze podataka. Na kraju poglavlja opisano je okruženje u kojem radi baza podataka i kontroli procesa.

Treće poglavlje ovog rada posvećeno je sigurnosnim karakteristikama MongoDB baze podataka. Opisano je kako su autentifikacija, autorizacija, šifrovanje i revizija implementirani kod MongoDB baze podataka. Poslednji deo trećeg poglavlja govori o okruženju u kojem radi MongoDB baza podataka, kao i kako se sprovode nadgledanje MongoDB baze podataka, oporavak od katastrofe I održavanje MongoDB baze podataka.

U okviru četvrtog poglavlja sumirane su osnove karakteristike sigurnosti kod MongoDB baze podataka.

2. Zahtevi za obezbeđivanje velikih količina podataka na mreži

Troškovi narušavanja bezbednosti mogu biti značajni ako se mere u smislu izgubljenog prihoda od prekida poslovanja, sudskih sporova, novčanih kazni, narušavanja poverenja potrošača i reputacije brenda. Posečni troškovi kršenja podataka u SAD-u porasli su na 5,4 miliona dolara, uglavnom zbog porasta zlonamernih napada. U isto vreme, smanjili su se troškovi kršenja bezbednosti izazvani nemarom ili nedostacima IT sistema.

U svetlu sve većih pretnji tokom prošle decenije, zajedno sa povećanom zabrinutošću za privatnost pojedinaca, industrije i Vlade širom sveta započele su niz inicijativa za povećanje bezbednosti, smanjenje prevara i zaštitu podataka o ličnosti (engl. *personally identifiable information* - PII), uključujući:

- PCI DSS za upravljanje informacijama o vlasnicima kartica
- HIPAA standardi za upravljanje zdravstvenim informacijama
- NIST 800-53 katalozi za kontrolu sigurnosti za sve američke savezne informacione sisteme osim onih koji se odnose na nacionalnu bezbednost
- STIG za sigurnu instalaciju I održavanje računarskih sistema, dizajniran za američko Ministarstvo odbrane
- Direktiva Evropske unije o zaštiti podataka
- Standardizacija zaštite podataka Azija-Pacifik ekonomske saradnje (engl. *Asia Pacific Economic Cooperation* - APEC)

Pored ovih inicijativa, svake godine se razvijaju novi propisi koji će se nositi sa novonastalim pretnjama i novim zahtevima za pooštrenom kontrolom upravljanja podacima.

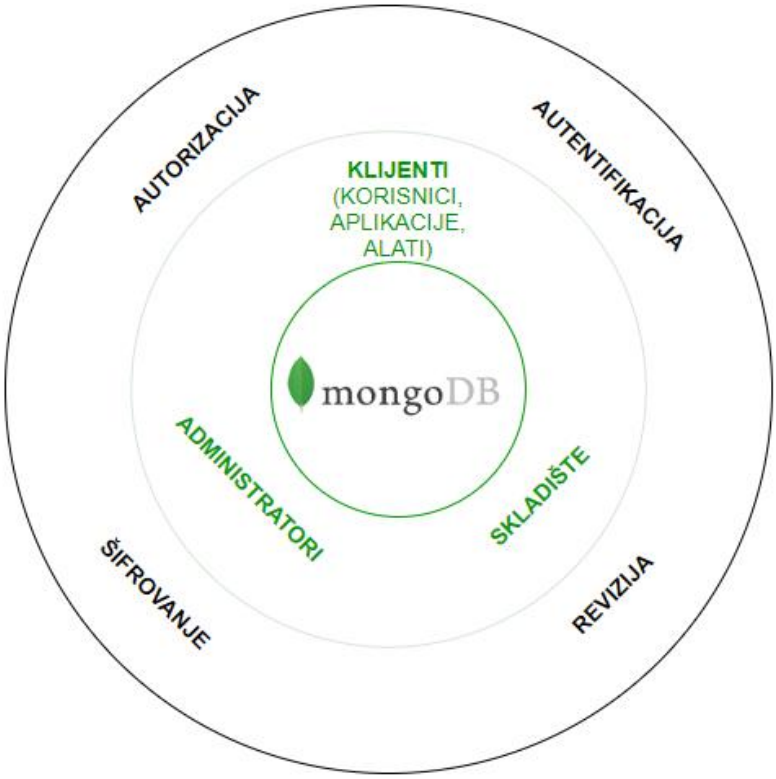
Svaki skup propisa određuje sigurnosne i revizijske zahteve koji su jedinstveni za određenu industriju ili aplikaciju, a usklađenost se procenjuje na osnovu projekta. Uprkos ovim razlikama, postoje zajednički bazični zahtevi svih direktiva, uključujući:

- Ograničavanje pristupa podacima, nametnuto putem unapred podešenih privilegija i nivoa bezbednosti
- Mere za zaštitu od slučajnog ili zlonamernog otkrivanja, gubitka, uništavanja ili oštećenja osetljivih podataka
- Razdvajanje dužnosti prilikom pokretanja aplikacija i pristupa podacima
- Snimanje aktivnosti korisnika, administrativnog osoblja i aplikacija za pristup i obradu podataka. [1]

Ovi zahtevi informišu bezbednu arhitekturu MongoDB, sa najboljim praksama za implementaciju sigurnog, usklađenog okruženja za upravljanje podacima. Bezbednosna arhitektura MongoDB baze podataka prikazana je na slici 1.

Holistička bezbednosna arhitektura mora da pokriva sledeće:

- Upravljanje korisničkim pravima za ograničavanje pristupa osetljivim podacima, implementirano pomoću autentifikacije i autorizacije
- Evidentiranje operacija u bazi podataka u revizijskom tragu
- Zaštita podataka šifrovanjem u procesu kretanja kroz mrežu i u trajnom skladištu
- Kontrola okruženja i procesa



Slika 1: MongoDB end-to-end bezbednosna arhitektura

Zahtevi za svaki od ovih elemenata razmatranu su u nastavku.

2.1 Upravljanje korisničkim pravima – Autentifikacija

Autentifikacija je dizajnirana da potvrdi identitet entiteta koji pristupaju bazi podataka. U ovom kontekstu, entiteti su definisani kao:

- Korisnici kojima je potreban pristup bazi podataka kao deo njihove svakodnevne poslovne funkcije
- Administratori (tj. sistemski administratori, administratori baze podataka (engl. *Database Administrator* – DBA), testeri garancije kvaliteta (engl. *Quality Assurance*)) i programeri
- Softverski sistemi uključujući servere aplikacija, alate za izveštavanje i pakete za upravljanje i izradu rezervnih kopija
- Fizički i logički čvorovi na kojima radi baza podataka. Baze podataka mogu se distribuirati kroz više čvorova, kako za operacije skaliranja, tako i za osiguravanje visoke dostupnosti u slučaju kvara ili održavanja sistema.

Najbolje prakse za autentifikaciju su sledeće.

- **Pravljenje sigurnih akreditiva[1].** Kreirajte akreditive (engl. *credentials*) za prijavu za svaki entitet kojem će pristup bazi podataka biti potreban i izbegavajte kreiranje jednostavnog "admin" korisničkog imena koje deli svaki korisnik.
  Stvaranjem akreditiva postaje lakše definisanje, upravljanje i praćenje pristupa sistemu za svakog korisnika. Ako se akreditivi korisnika ugroze, ovaj pristup olakšava opoziv korisnika bez ometanja ostalih korisnika kojima je potreban pristup bazi podataka.
  Programeri, administratori i administratori baze podataka (DBAs) bi trebalo da imaju jedinstvene akreditive za pristup bazi podataka. Kada se isti kredencijali dele između korisnika, nemoguće je identifikovati ko je pokušao različite operacije, i ne postoji mogućnost upravljanja dozvolama. Uz jedinstvene akreditive za prijavu, osoblju koje napusti projekat ili organizaciju se može opozvati pristup bez uticaja na druge korisničke naloge.
  Najbolja praksa je da kreirate odvojene akreditive za prijavljivanje za svaku aplikaciju koja će pristupati bazi podataka. Kako se uvode nove aplikacije, a povlače stare, ovaj pristup pomaže u upravljanju pristupom. Neki MongoDB korisnici kreiraju višestruke prijave za različite komponente jedne aplikacije, koje se evidentiraju i prate u revizijskim stazama (engl. *audit trails*) i evidencijama upita. Ovo je moguće zahvaljujući identifikaciji komponenata aplikacije koje pokušavaju da izvrše određene upite ili operacije, ili komponenata koje bi imale koristi od optimizacije performansi.
  Autentifikacija, tj. potvrda identiteta treba da se izvrši između čvorova. Ovo sprečava neovlašćene servere da se pridruže klasteru baze podataka, sprečavajući time nezakonito kopiranje ili premeštanje podataka na nesigurne čvorove.

[1] Skup informacija za identifikaciju korisnika koji se koriste za pristup lokalnim i mrežnim resursima. Primeri akreditiva su korisnička imena i lozinke, pametne kartice i sertifikati

- **Podrška u bazi podataka i centralizovano upravljanje korisnicima**. Baze podataka treba da pružaju mogućnost upravljanja autentifikacijom korisnika, bilo u samoj bazi podataka, bilo kroz integraciju sa sistemima za upravljanje identitetima širom organizacije. Integrisanje MongoDB baze podataka u postojeću infrastrukturu bezbednosti informacija nameće centralizovanu i standardizovanu kontrolu nad korisničkim pristupom. Ako, na primer, korisnički pristup mora biti opozvan, ažuriranje se može izvršiti u jednom skladištu i odmah primeniti na svim sistemima, uključujući MongoDB bazu podataka.
- **Poštovanje smernica za lozinku**. Lozinke treba da se pridržavaju minimalnih zahteva složenosti koje je uspostavila organizacija, i treba ih povremeno menjati. Lozinke sa niskom entropijom lako je razbiti, čak i ako su šifrovane (engl. *encripted*). Razbijanje lozinki sa visokom entropijom je moguće ako je dovoljno vremena na raspolaganju, ali to često nije isplativo. [1]

2.2 Upravljanje korisničkim pravima - Autorizacija

Jednom kada je identitet entiteta potvrđen (autetnifikovan), autorizacija reguliše šta taj entitet ima parvo da radi u bazi podataka. Privilegije, koje definišu određeni skup akcija koje se mogu izvršiti nad bazom podataka, se dodeljuju korisničkim ulogama.

Ovaj odeljak sadrži najbolje prakse za upravljanje privilegijama, kao jednog dela procesa autorizacije.

- **Dodeliti minimalni pristup entitetima.** Entitetima treba omogućiti minimalni pristup bazi podataka potreban za obavljanje njihove funkcije. Ako aplikacija zahteva pristup logičkoj bazi podataka, trebalo bi da bude ograničena na operacije samo nad tom bazom podataka. U tom slučaju, treba sprečiti pristup drugim logičkim bazama podataka od strane aplikacije. Ovo pomaže u zaštiti od zlonamernog i slučajnog pristupa ili neovlašćene izmene podataka.
- **Grupisanje zajedničkih privilegija pristupa i uloge.** Entiteti se često mogu grupisati u uloge (engl. *roles*) kao što su "Administrator baze podataka", "Sistemski administrator" i "Server aplikacija[2]". Dozvolama za ulogu može se centralizovano upravljati, a korisnici se po potrebi mogu dodavati ili uklanjati iz uloga. Korišćenje uloga pomaže pojednostavljenju upravljanja kontrolom pristupa definisanjem jednostavnog skupa pravila koja se primenjuju na određene klase entiteta, umesto da se pravila definišu pojedinačno za svakog korisnika.
- **Kontrolisanje koje akcije entitet može da izvrši.** Prilikom odobravanja pristupa bazi podataka, treba razmotriti koje specifične akcije ili naredbe svaki entitet treba da pokreće, i dodeljivati dozvole za pokretanje u skladu sa tim. Na primer, aplikaciji će možda biti potrebne dozvole za čitanje/pisanje (engl. *read-write*) u bazu podataka, dok će alat za

[2] Uloga koja se sastoji od usluga (engl. *services*) koje korisnicima omogućavaju pristup podacima i sadržaju.

izveštavanje biti ograničen na dozvolu samo za čitanje (engl. *read-only*). Nekim korisnicima se mogu dodeliti privilegije koje im omogućavaju ubacivanje novih podataka u bazu podataka, ali ne i ažuriranje ili brisanje postojećih podataka. Treba voditi računa da se obezbedi minimalni skup privilegija. Ako se akreditivi najprivilegovanijih naloga hakuju interno ili od strane spoljnog uljeza, cela baza podataka može biti ugrožena.

- **Kontrola pristupa osetljivim podacima.** Da bi se sprečila pojava silosa podataka[3] (engl. *data silos*), trebalo bi omogućiti definisanje dozvola za pojedinačne zapise na osnovu bezbednosnih privilegija. Na primer, neka polja zapisa mogu biti dostupna svim korisnicima baze podataka, dok druga koja sadrže osetljive podatke, kao što su podaci za identifikaciju ličnosti, treba da budu ograničene na korisnike sa određenom bezbednosnom dozvolom. [1]

2.3 Revizija

Stvaranjem revizorskih staza, promene u konfiguraciji baze podataka i podacima mogu se uhvatiti za svaki entitet koji pristupa bazi podataka, pružajući evidenciju za uslađenost i sigurnosnu analizu. Revizijom se takođe mogu otkriti pokušaji neovlašćenog pristupa podacima.

Okvir revizije (engl. *auditing framework*) je uveden u MongoDB 2.6. Proširen je tako da uključuje evidentiranje operacija čitanja i pisanja (DML) u bazu podataka. Pomoću ovih proširenja administratori sada mogu konstruisati i filtrirati revizijske staze za bilo koju operaciju nad MongoDB bazom podataka, bez potrebe da se oslanjaju na alate nezavisnih proizvođača. Revizija zasnovana na ulogama je dodata kako bi se poboljšala selektivnost revizije. Sada je moguće evidentirati i izveštavati o aktivnostima po ulogama. Revizija je dostupna kao deo *MongoDB Enterprise Advanced*-a.

- **Praćenje promena u konfiguraciji baze podataka.** Svaki put kada se konfiguracija baze podataka promeni, akcija treba da zabeleži u dnevnik revizije (engl. *audit log*) koji treba da sadrži akciju promene, identitet korisnika koji je izvršio promenu i vremensku oznaku.
- **Praćenje promene podataka.** Trebalo bi omogućiti konfigurisanje traga revizije da obuhvati svaki upit ili operaciju upisa u bazu podataka. Međutim, treba biti oprezan prilikom konfigurisanja ovog pravila za aplikacije. Na primer, ako aplikacija ubaci desetine hiljada zapisa u sekundi, svaka operacija upisivanja u dnevnik revizije može narušiti performance baze podataka. Odgovornost projektnog tipa je da utvrdi bilo kakve kompromise između performansi i sigurnosti. [1]

[3] Silos podataka je kolekcija podataka u organizaciji koje je izolovana i kojoj drugi delovi organizacije ne mogu pristupiti.

2.4 Šifrovanje

Šifrovanje je kodiranje kritičnih podataka kad god su u tranzitu ili u mirovanju, omogućavajući samo ovlašćenim entitetima da ih pročitaju. Podaci će biti zaštićeni u slučaju da prisluškivači ili hakeri dobiju pristup serveru, mreži ili bazi podataka.

- **Šifrovanje konekcije sa bazom podataka.** Pristup bazi podataka od strane svih korisnika treba da bude preko šifriranih kanala, uključujući i veze uspostavljene preko upravljačkih programa (engl. *drivers*), komandne linije ili šel skipti[4] (engl. *shell script*), kao i sesije udaljenog pristupa samim serverima baze podataka.
  Interna komunikacija između čvorova baze podataka takođe treba da bude šifrovana, tj. saobraćaj se replicira između čvorova klastera baze podataka.
- **Podsticanje jakog šifriranja** (engl. *strong encryption*). Baza podataka treba da podržava FIPS (engl. *Federal Information Processing Standard*) 140-2 kako bi se osigurala implementacija sigurnih algoritama šifrovanja.
- **Šifrovanje podataka u mirovanju.** Jedna od najčešćih pretnji sigurnosti dolazi od napada koji zaobilaze samu bazu podataka i ciljaju osnovni operativni sistem i fizičko skladište produkcionog servera ili uređaja sa rezervnim kopijama (engl. *backup devices*), kako bi pristupili sirovim podacima. Šifrovanje fajlova podataka baze podataka na disku ublažava ovu pretnju.
- **Potpisivanje i rotiranje ključeva za šifrovanje.** Ključeve za šifrovanje mreže i diska treba povremeno rotirati. SSL (engl. *Secure Socket Layer*)[5] kanali za šifrovanje treba da koriste potpisane sertifikate kako bi osigurali da klijenti mogu da potvrde akreditive koje dobijaju od komponenti servera. [1]

2.5 Okruženje i kontrola procesa

Okruženje u kojem radi baza podataka i osnovna infrastruktura treba da bude zaštićeno i fizičkim i logičkim kontrolama. Kontrole se primenjuju u osnovnom okruženju za razvoj (engl. *deployment*), umesto u samoj bazi podataka, i uključuju:

- Ugradnju zaštitnih zidova (engl. *firewalls*)
- Mrežne konfiguracije
- Definisanje dozvola za sistem datoteka (engl. *file system*)
- Stvaranje fizičkih kontrola pristupa IT okruženju

[4] Skripta koju izvršava interpretator naredbi operativnog sistema

[5] SSL je bezbednosni protokol koji se koristi za slanje poverljivih podataka preko interneta. SSL kreira bezbednu konekciju između pretraživača i servera. U tom procesu se koristi enkripcija, kako bi se obezbedila privatnost podataka.

Pored toga, postoji niz operativnih procesa koje treba usvojiti za dalje unapređenje i sprovođenje sigurnog rada, uključujući:

- Obuka administratora baze podataka i programera
- Nadgledanje baze podataka i izrada rezervnih kopija
- Održavanje baze podataka, tj. primena najnovijih zakrpa[6] (engl. *patches*). [1]

[6] Zakrpe su softverske ispravke koje su izdali dobavljači kako bi ispravili nedostatke softverskih proizvoda koji ih mogu učiniti nepouzdanima i dovesti do gubitka ili oštećenja podataka. Neki nedostaci proizvode čine ranjivim i osetljivim na napade, i u tom slučaju se izdaju sigurnosne zakrpe za ispravljanje problema.

3. MongoDB sigurnosne karakteristike

Sa grubozranstim (engl. *coarse grained*) kontrolama pristupa[7] i limitiranim ograničenjima sigurnosti podataka na mnogim NoSQL platformama i platformama za velike količine podataka, organizacije su često primorane da zaključaju podatke u odvojene silose, sprešavajući obradu i analizu u realnom vremenu preko kompletnih skupova podataka.

Zahvaljujući sveobuhvatnim kontrolama za upravljanje korisničkim pravima, revizijom i šifrovanjem, zajedno sa najboljim praksama u zaštiti okruženja, MongoDB može ispuniti prethodno opisane zahteve najbolje prakse, omogućavajući korisnicima da otkrivaju moć velikih količina podataka na mreži.

Sledeće poglavlje razmatra bezbednosnu arhitekturu MongoDB baze podataka pre predstavljanja kontrolne liste onih mogućnosti potrebnih za stvaranje bezbednog okruženja baze podataka.

3.1 MongoDB Enterprise Advanced baza podataka

MongoDB Enterprise Advanced je sertifikovano i podržano proizvodno izdanje MongoDB baze podataka, sa naprednim bezbednosnim funkcijama.

[7] Granularnost (engl. *granularity*) je mera u kojoj se sistem raščlanjuje na manje delove, bilo sam sistem ili njegov opis ili posmatranje. To je mera u kojoj je veći entitet podeljen na manje delove.
Grubozrnasti sistemi se sastoje od manjeg broja većih komponenata u poređenju sa sitnozrnastih sistema. Grubozrnasti opis sistema se odnosi na velike potkomponente, dok se sitnozrnasti opis odnosi na manje komponente od kojih su veće sastavljene.
Kod grubozranstih sistema nekoliko objekata sadrži puno povezanih podataka, zbog čega usluge imaju širi opseg funkcionalnosti. Na primer, jedan objekat „Račun“ sadrži ime kupca, adresu, stanje računa, datum otvaranja, datum poslednje promene itd.
Kod finozrnastih sistema više objekata sadrži manje podataka, pa usluge imaju uži opseg funkcionalnosti. Na primer, objekat „Račun“ čuva stanje, „Kupac“ ima ime i adresu, objekat „OtvaranjeRačuna“ sadrži datum otvaranja itd.
Dakle, kod krupnozranstih sistema je povećana složenost dizajna, a smanjen broj ćelija za različite operacije. Kod finozrnastih sistema je smanjena složenost dizajna, ali je veći broj ćelija za različite operativne usluge. [5]

Slika 2: Integrisanje MongoDB baze podataka sa centralizovanom kontrolom pristupa korisnika

MongoDB Enterprise Advanced nudi napredne bezbednosne funkcije kao što su Kerberos[8], LDAP (engl. *Lightweight Directory Access Protocol*)[9] i revizija, povrh sveobuhvatnog bezbednosnog okvira MongoDB baze podataka, koji uključuje kontrolu pristupa zasnovanu na ulogama, PKI (engl. *Public key infrastructure*)[10] sertifikate, uređivanje na nivou polja i SSL. Možete besplatno pristupati svim funkcionalnostima MongoDB Enterprise za razvojna okruženja.

Na slici 2 prikazano je integrisanje MongoDB baze podataka sa centralizovanom kontrolom pristupa korisnika.

O MongoDB Enterprise Advanced i srodnim proizvodima i uslugama možete više saznati u tehničkoj dokumentaciji.

3.2 MongoDB autentifikacija

Autentifikacijom entiteta koji pristupaju MongoDB bazi podataka može se upravljati iz same baze podataka ili putem integracije sa spoljnim mehanizmom (tj. LDAP, x.509 sertifikata ili Kerberos servisa). MongoDB Enterprise Advanced je potreban kada koristite LDAP ili Kerberos protokoli.

[8] Kerberos je protokol za potvrdu identiteta računarske mreže koji radi na osnovu tiketa kako bi omogućio čvorovima koji komuniciraju preko nesigurne mreže da na siguran način međusobno dokažu svoj identitet. Usmeren je na klijent-server model i pruža uzajamnu potvrdu identiteta (i klijent i server međusobno potvrđuju identitet). Kerberos se temelji na simetričnoj kriptografiji i zahteva pouzdanu treću stranu, a opciono može da koristi kriptografiju javnog ključa tokom određenih faza potvrde identiteta. Podrazumevano koristi UDP port 88.

[9] LDAP – lak protocol za pristup direktorijumu je otvoreni, neutralan prema dobavljaču (dobavljač ne može da kontroliše definiciju, reviziju ili distribuciju specifikacije), industrijski standardni aplikativni protocol za pristup i održavanje distribuiranih informacionih usluga direktorijuma preko mreže Internet protokola (IP). Uobičajena upotreba LDAP protokola je obezbeđivanje centralnog mesta za čuvanje korisničkih imena i loziniki.

[10] PKI – infrastruktura javnog ključa je opšti termin za sve što se koristi za uspostavljanje i upravljanje enkripcijom javnog ključa, jednim od najčešćih oblika enkripcije na internetu.

3.2.1 Autentifikacija u bazi podataka

MongoDB potvrđuje identitet entiteta na nivou baze podataka. Potvrda identiteta korisnika vrši se pomoću naredbe za autentifikaciju (potvrdu identiteta), dok čvorovi baze podataka mogu biti autentifikovani u MongoDB klasteru putem ključnih fajlova.

Pogledajte dokumentaciju o kontroli pristupa (engl. *Access Control documentation*) da biste saznali više. [2]

3.2.2 LDAP autentifikacija

Mnoge organizacije široko koriste LDAP za standardizaciju i pojednostavljivanje načina upravljanja velikim brojem korisnika u unutrašnjim sistemima i aplikacijama. U mnogim slučajevima, LDAP se takođe koristi kao centralizovano telo, centralizovani autoritet, za kontrolu korisničkog pristupa kako bi se osiguralo da su unutrašnje polise bezbednosti usklađene sa korporativnim i regulatornim smernicama.

Sa LDAP integracijom, MongoDB može da autentifikuje korisnike direktno na osnovu korporativne LDAP infrastrukture, eliminišući potrebu za dupliciranjem upravaljanja lozinkom između LDAP direktorijuma i unutrašnjih kontrola za autentifikaciju MongoDB baze podataka. Imajte na umu da MongoDB trenutno podržava LDAP autentifikaciju, a ne podržava autorizaciju. Više detalja o kontrolama autorizacije koje su dostupne u MongoDB bazi podataka nalaze se u sledećem odeljku tehničke dokumentacije.

Administratori mogu konfigurisati MongoDB bazu podataka za autentifikaciju korisnika putem Linux PAM modula (engl. *Pluggable Authentication Modules*)[11] ili prosleđivanjem zahteva za potvrdu identiteta određenom LDAP servisu. LDAP integracija je dostupna u MongoDB bazi podataka. [2]

3.2.3 Kerberos autentifikacija

Sa MongoDB Enterprise Advanced podržana je autentifikacija (potvrda identiteta) pomoću servisa Kerberos. Kerberos je industrijski standardni protokol za potvrdu identiteta za velike klijent/server sisteme, koji omogućava klijentu i serveru da međusovno verifikuju identitet. Uz

[11] Priključni moduli za potvrdu identiteta

podšku Kerberos protokola, MongoDB može iskoristiti posojeću infrastrukturu i procese za potvrdu identiteta, uključujući *Microsoft Windows Active Directory*.

Kao i kod LDAP protokola i x.509 sertifikata, pre nego što korisnici mogu da se autentifikuju u MongoDB pomoću Kerberos protokola, moraju prvo da se kreiraju i da im se dodele privilegije unutar MongoDB baze podataka. Proces za to, zajedno sa kompletnom listom za konfiguraciju, opisan je u vodiču za MongoDB I Kerberos. [2]

3.2.4 Autentifikacija x.509 sertifikatom

Uz podršku za x.509 sertifikate, MongoDB se može integrisati sa postojećom infrastrukturom informacione sigurnosti i ovlašćenjima za izdavanje sertifikata, podržavajući autentifikaciju korisnika i autentifikaciju među čvorovima.

Korisnici mogu biti autentifikovani za MongoDB pomoću klijentskih sertifikata umesto samoodržavanih i potencijalno ranjivih lozinki.

Potvrda identiteta među klasterima i komunikacija između MongoDB čvorova mogu se osigurati sa x.509 sertifikatima članova, a ne sa ključnim fajlovima, osiguravajući strože kontrole članstva sa manje administrativnih troškova, tj. ukljanjanjem zajedničke lozinke koju koriste ključni fajlovi. Čvorovi mogu da koriste x.509 sertifikat da bi verifikovali svoje članstvo u MongoDB skupovima replika i zaoštrenim klasterima. Jedno telo za izdavanje sertifikata (engl. *Certificate Authority* - CA) treba da izda sve x.509 sertifikate za članove zaoštrenog klastera ili skupa replika.

Uputstva za konfiguraciju opisana su u vodiču za MongoDB I x.509 sertifikate. [2]

3.2.5 MongoDB i upravljanje identitetom pomoću “Crvenog šešira”[12]

Red Hat Enterprise Linux (RHEL) je popularno okruženje za MongoDB razvoj. Pružajući jednostavnost upotrebe administratorima i profesionalcima iz oblasti bezbednosti koji rade u ovim okruženjima, sigurnosne funkcije MongoDB Enterprise baze podataka integrisane su sa RHEL karakteristikom upravljanja identitetom (engl *Identity Management* - IdM). Ova integracija

[12] Engl. *Red Hat Identity Management(IdM)* – pruža centralizovani i objedinjeni način upravljanja skladištima identiteta, autentifikacijom, smernicama i polisama autroizacije u domenu zasnovanom na Linux operativnom sistemu. IdM značajno smanjuje administrativne troškove individualnog upravljanja različitim uslugama i korišćenja različitih alata na različitim mašinama. [6]

omogućava centralno upravljanje pojedinačnim entitetima i njihovu autentifikaciju, autorizaciju i privilegije.

Pogledajte uputstva za upravljanje idenitetima, Red Hat Linux Identity Management tutorial sa uputstvima za konfiguraciju MongoDB baze podataka.

Rad Hat IdM integracija je dostupna sa MongoDB bazom podataka i zahteva konfiguraciju baze podataka za Kerberos autentifikaciju. [1]

3.2.6 MongoDB i Microsoft aktivni direktorijum

MongoDB Enterprise Advanced pruža podršku za potvrdu identiteta pomoću Microsoft aktivnog direktorijuma (engl. *Microsoft Active Directory*) i Kerberos protokola. Kontroler domena aktivnog direktorijuma potvrđuje identitet MongoDB korisnika i servera koji rade na Windows mreži.

3.3 MongoDB autorizacija

MongoDB omogućava administratorima da definišu dozvole koje aplikacija ili korisnik ima, i koje prodatke mogu videti prilikom postavljanja upita prema bazi podataka.

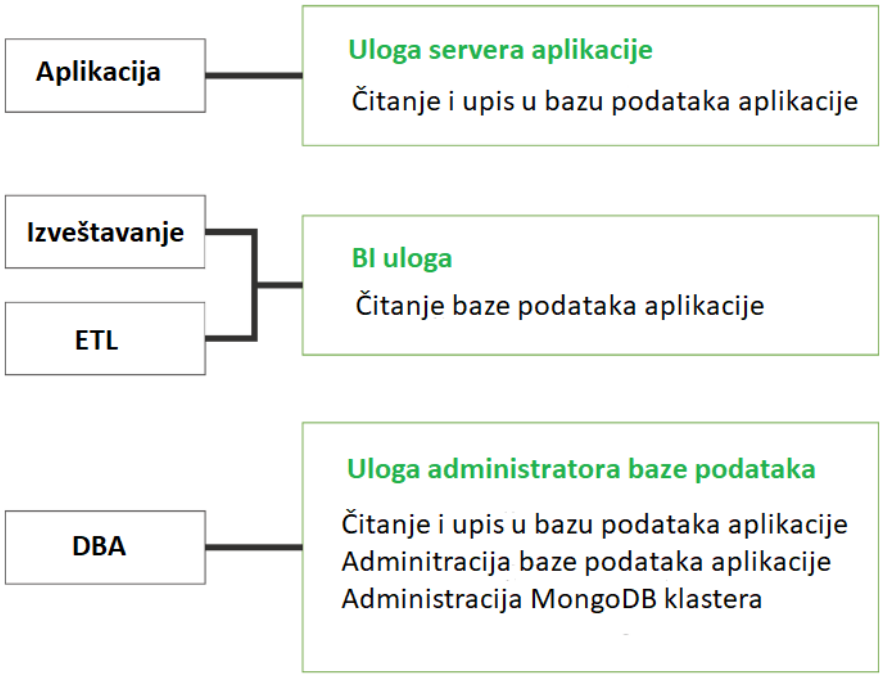
3.3.1 Korisnički definisane uloge

MongoDB 2.4 je predstavio mogućnost razlikovanja privilegija korisnika i administratora sa ugrađenim ulogama. MongoDB 2.6 je proširio svoje mogućnosti autorizacije pomoću korisnički definisanih uloga, omogućavajući administratorima da dodeljuju precizne privilegije korisnicima ili aplikacijama. Ove privilegije za autorizaciju mogu se zasnivati na sepcifičnoj funkcionalnosti koja je korisniku potrebna u njihovoj ulozi ili na njihovoj organizacionoj strukturi. MongoDB pruža mogućnost određivanja privilegija korisnika i nad bazom podataka, i na nivou konekcije.

Privilegije se dodeljuju ulogama, a uloge se dodeljuju korisnicima. Na primer:

- Klasama korisnika i aplikacijama se mogu dodeliti privilegije za unos podataka (insert), ali ne i za ažuriranje ili brisanje podataka iz baze podataka

- Administratorima baze podataka mogu biti dodeljene privilegije koje im omogućavaju da kreiraju kolekcije i indekse u bazi podataka, doku su programeri ograničeni na CRUD[13] operacije
- Određene administratorske uloge mogu imati privilegije širom klastera za izgradnju skupova replika i konfigurisanje oštrenja, dok su druge ograničene na stvaranje novih korisnika ili pregledanje evidencija
- Procesi nadgledanja MongoDB klastera mogu biti ograničeni na pokretanje samo onih naredbi koje preuzimaju status servera, bez potpunog administrativnog pristupa za obavljanje operacija nad bazom podataka
- Unutar okruženja sa više zakupaca, programerima i administratorima, „stanodavcima“ se mogu dodeliti dozovle za fizičke baze podataka, dok se programerima i administratorima „stanarima“ može dodeliti ograničeniji skup akcija u logičkim bazama podataka ili pojedinačnim kolekcijama. Ova funkcionalnost omogućava jasno razdvajanje dužnosti i kontrole, kako između organizacija, tako i unutar njih. [1]



Slika 3: MongoDB korisnički definisane uloge dozvoljavaju razdvajanje dužnosti

Slika 3 prikazuje korisnički definisane uloge I privilegije koje se tim ulogama dodeljuju u MongoDB bazi podataka.

[13] CRUD = Create, Read, Update, Delete

Da bi se osiguralo lako obezbeđivanje i održavanje naloga, uloge se mogu delegirati među timovima, osiguravajući sprovođenje konzistentnih polisa u okviru određenih funkcija u organizaciji.

Više informacija o ulogama u MongoDB bazi podataka nalaze se u odeljku Autorizacija u dokumentaciji.

U kombinaciji sa mogućnostima revizije dostupnim u MongoDB Enterprise Advanced, korisnici mogu da definišu određene administrativne radnje po ulozi, i da evidentiraju sve te radnje. Kao rezultat toga, organizacija je u stanju da sprovede operativnu kontrolu od kraja do kraja (engl. *end-to-end*) i da održava uvid u akcije radi usklađivanja i izveštavanja.

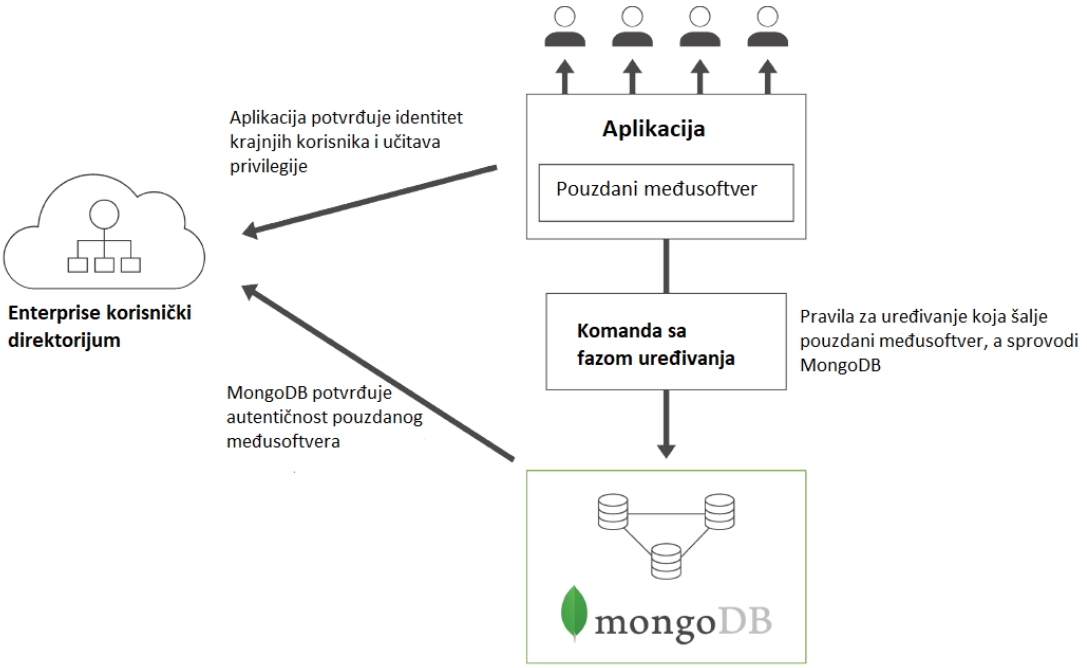
3.3.2 MongoDB uređivanje na nivou polja

Uređivanje na nivou polja kod MongoDB baze podataka omogućava izgradnju kontrole pristupa na nivou polja u pouzdanom međusoftveru[14] (engl. *middleware*). Implementirano kroz novu fazu uredništva na MongoDB cevovodu za agregaciju (engl. *Aggregation Pipeline*), programeri dobijaju metod za ograničavanje sadržaja vraćenog dokumenta na nivou polja. Dozvole se mogu zasnivati i na sadržaju dokumenta i na specifičnim korisničkim privilegijama, na osnovu bezbednosnih labela. Polise za kontrolu pristupa mogu se opisati korišćenjem MongoDB upitnog jezika, što programerima olakšava implementaciju potrebnih kontrola.

Budući da se podaci uređuju pre nego što se vrate aplikaciji, izloženost osetljivih informacija je smanjena. Uređivanje na nivou polja je primenljiva na širok spektar osetljivih podataka, uključujući podatke koji mogu da identifikuju osobu, kao što su imena, brojevi socijalnog osiguranja, datumi rođenja i brojevi bankovnih računa.

Slika 4 prikazuje uređivanje na nivou polja koje ograničava pristup osetljivim podacima kod MongoDB baze podataka.

[14] Softver koji se nalazi između dve ili više vrsta softvera i prevodi informacije između njih. Međusoftver može da pokrije širok spektar softvera i uglavnom se nalazi između aplikacije i operativnog sistema, mrežnog operativnog sistema ili sistema za upravljanje bazama podataka.



Slika 4: MongoDB uređivanje na nivou polja ograničava pristup osetljivim podacima

U ovoj početnoj implementaciji, aplikacija mora da preda logiku za uređivanje bazi podataka sa svakim zahtevom. Stoga se oslanja na pouzdani međuprogram pokrenut u aplikaciji da bi se osigurao da faza uređivanja doda bilo kojem upitu koji zahteva logiku uređivanja. [1]

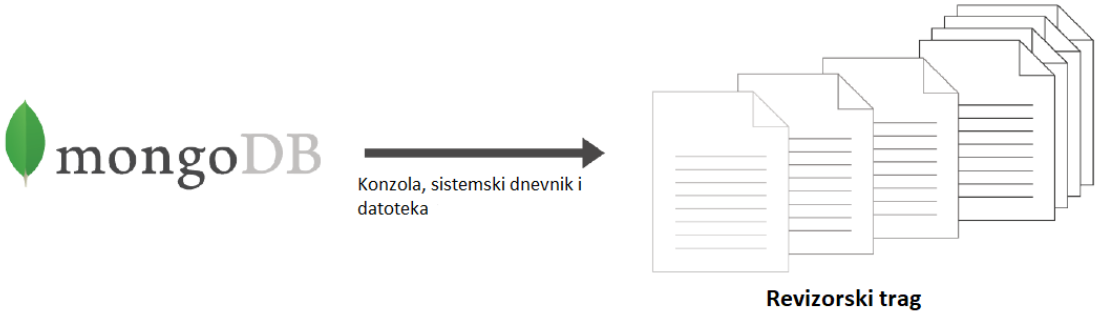
3.4 MongoDB revizija

MongoDB Enterprise Advanced revizorski okrir evidentira sve pristupe i akcije izvršene prema bazi podataka. Okvir revizije obuhvata administrativne radnje (DDL[15]) kao što su operacije šeme, kao i aktivnosti autentifikacije i autorizacije, zajedno sa operacijama čitanja i pisanja (DML[16]) u bazu podataka. Administratori mogu konstruisati i filtrirati revizijske staze za bilo koju operaciju nad MongoDB bazom podataka, bilo da je to DML, DCL[17] ili DDL, bez potrebe da se oslanjaju na alate nezavisnih porizvođača. Na primer, moguće je evidentirati i revidirati identitet korisnika koji su preuzeli određene dokumente, i sve promene napravljene u bazi podataka tokom njihove sesije.

[15] Engl. *Data definition language* – jezik za definisanje podataka
[16] Engl. *Data manipulation language* – jezik za manipulisanje podacima
[17] Engl. *Data control language* – jezik za kontrolu podataka [7]

Administratori mogu da konfigurišu MongoDB tako da evidentiraju sve akcije ili da primene sve filtere za hvatanje samo određenih događaja, korisnika ili uloga. Evidentirane revizije se mogu zapisati na više odredišta u različitim formatima, uključujući i konzolu i sistemski dnevnik (engl. *system log*) (u JSON[18] formatu), i u datoteku (JSON ili BSON[19]), koja se zatim može učitati u MongoDB i analizirati radi identifikovanja relevantnih događaja. Svaki MongoDB server upisuje događaje u svoju pridruženu memoriju. Administrator baze podataka tada može koristiti sopstvene alate za spajanje ovih događaja u jedan dnevnik revizije, omogućavajući klasterski prikaz operacija koje su uticale na više čvorova.

Održavanje revizorskog traga administrativnih radnji nad MongoDB bazom podataka prikazana je na slici 5.



Slika 5: MongoDB održava revizorski trag administrativnih radnji nad bazom podataka

MongoDB Enterprise Advanced podržava reviziju zasnovanu na ulogama. Moguće je evidentirati i prijaviti aktivnosti prema određenoj ulozi, kao što je userAdmin ili dbAdmin – zajedno sa bilo kojim nasleđenim ulogama koje ima svaki korisnik – umesto izdvajanja aktivnosti za svakog pojedinačnog administratora.

Revizija dodaje opšte troškove MongoDB sistemu. Iznos zavisi od nekoliko faktora, uključujući događaje koji se evidentiraju i gde se održava dnevnik revizije, kao što su spoljni uređaji za skladištenje i format dnevnika revizije. Korisnici bi trebalo da razmotre svoje posebne potrebe aplikacije za reviziju i svoje ciljeve performansi kako bi odredili svoju optimalnu konfiguraciju. [1]

Saznajte više iz revizijske dokumentacije MongoDB baze podataka.

[18] Engl. JavaScript Object Notation – tekstualno baziran otvoreni standard dizajniran za ljudima razumljivu razmenu podataka.

[19] Engl. Binary JavaScript Object Notation – binarni JSON je format računarske razmene podataka. Koristi se za predstavljanje jednostavnih ili složenih struktura podataka, uključujući asocijativne nizove (poznate i kao parovi imena i vrednosti), celobrojne indeksirane nizove i skup osnovnih skalarnih tipova.

3.5 MongoDB šifrovanje

Administratori mogu šifrovati MongoDB podatke tokom kretanja kroz mrežu i u mirovanju u trajnom skladištu.

3.5.1 Mrežno šifrovanje

Podrška za SSL omogućava klijentima da se povežu sa MongoDB preko šifrovanog kanala. Klijenti su definisani kao bilo koji entitet sposoban za povezivanje sa MongoDB serverom, uključujući:

- Korisnike i administratore
- Aplikacije
- MongoDB alate (npr. *mongodump*, *mongorestore*, *mongotop*)
- Čvorovi koji čine MongoDB klaster, kao što su članovi skupa replika, ruteri upita i konfiguracioni serveri.

Mnogi MongoDB upravljački programi podržavaju SSL konekcije, uključujući one za Java, Rubi, Python, node.js i C# aplikacije.

Moguće je mešati SSL sa konekcijama koje nisu SSL na istom portu, što može biti korisno prilikom primene finozrnastih kontrola šifrovanja za unutrašnji i spoljašnji saobraćaj, kao i izbegavanje zastoja prilikom nadogradnje MongoDB klastera za podršku SSL protokolu.

TLS protokol je takođe podržan sa x.509 sertifikatima.

MongoDB podržava FISP 140-2 šifrovanje, ako se izvodi u FIPS režimu sa FISP potvrđenim kriptografskim modulom. Proces *mongod* i *mongos* treba da budu konfigurisani „sslFIPSMode" podešavanjem. Pored toga, ove procese treba primeniti na sistemima sa *OpenSSL* bibliotekom konfigurisanom sa FISP 140-2 modulom.

MongoDB dokumentacija uključuje vodič za konfigurisanje SSL konekcija.

3.5.2 Šifrovanje diska

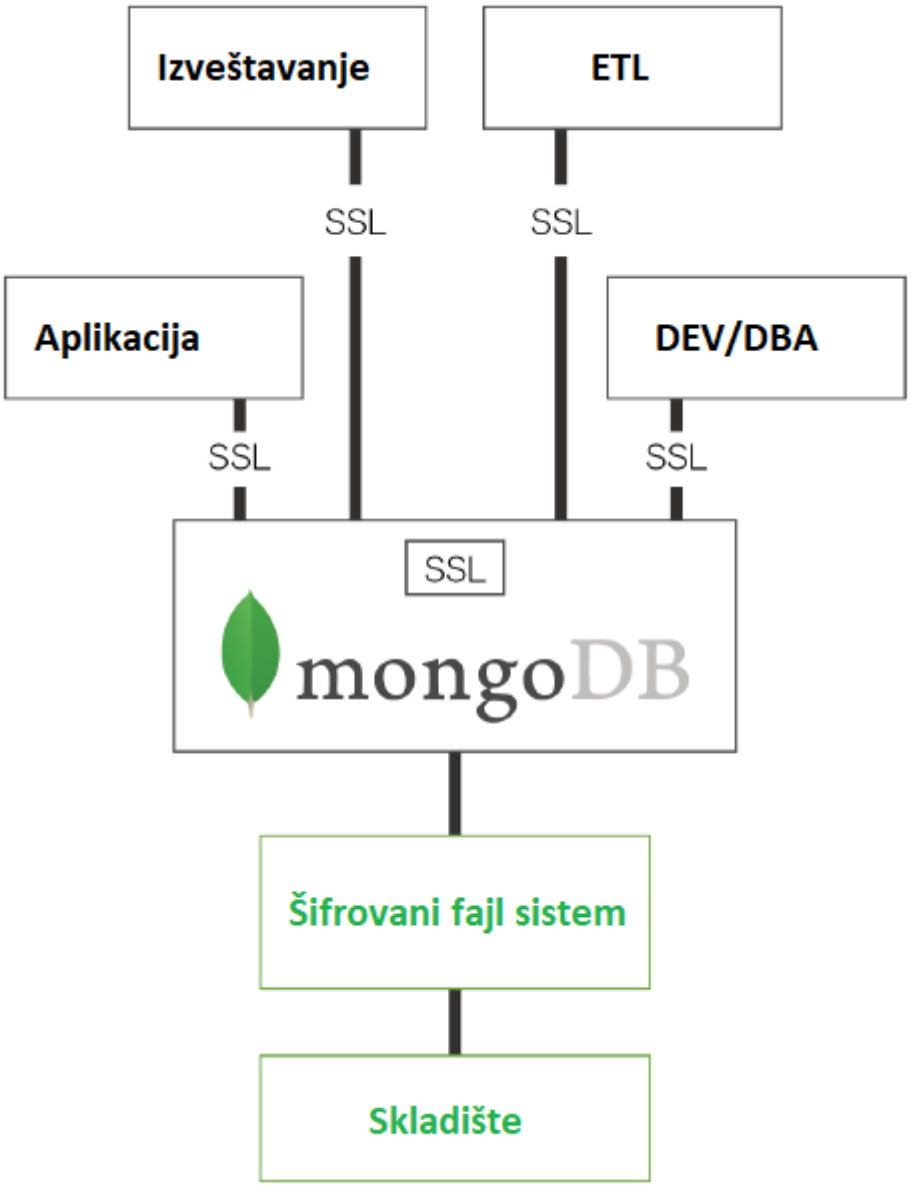
Postoji više načina za šifrovanje podataka u stanju mirovanja pomoću MongoDB baze podataka. Jedan od pristupa je šifrovanje podataka na nivou polja u aplikativnom sloju, korišćenjem potrebnog tipa šifrovanja koji odgovara aplikaciji. Rešenja poput vorometrijskog šifrovanja

aplikacije (engl. *Vormetric Application Encryption*) integrisani su sa MongoDB bazom podataka za podršku šifrovanja dokumenta i polja.

Druga opcija je upotreba nezavisnih biblioteka koje pružaju šifrovanje na nivou diska kao deo jezgra operativnog sistema, pružajući napredno upravljanje ključevima koje osigurava da samo ovlašćeni procesi mogu pristupiti tim podacima. Biblioteke šifrovanja diska mogu osigurati da kriptografski ključevi ostanu sigurni i osiguravaju usklađenost sa standardima kao što su HIPAA, PCI-DSS i FERPA. Primeri rešenja uključuju:

- *Linux UniSed Key Setup* (LUKS) je dostupan u više Linux distribucija. *Red Hat* dokumentacija dostupna je kao primer LUKS konfiguracije na Linux-u
- *IBM Guardium Data Encryption* pruža podršku za šifrovanje podataka iz MongoDB baze na nivou diska u Linux i Windows okruženjima
- *Vormetric Data Security Platform* pruža šifrovanje operativnog sistema, fajlova i aplikativnog nivoa.
- *BitLocker Drive Encryption* se takođe može koristiti za obezbeđivanje šifrovanja na Windows platformi.

Slika 6 vizuelno prikazuje šifrovanje MongoDB baze podataka. [1]



Slika 6: MongoDB end-to-end šifrovanje

3.6 Okruženje i procesi

Nadovezujući se na prethodno razmotrene bezbednosne kontrole baze podataka, da bismo dalje smanjili rizik od ekploatacije, pokrenuli MongoDB u pouzdanom okruženju, implementirali smo siguran razvojni životni ciklus i primenili najbolje operativne razvojne prakse.

Preporučuje se pristup „odbrane u dubini“ (engl. „*Defense in Depth*“) kao dopuna sigurnom MongoDB razvoju, obraćajući se brojnim različitim metodama za upravljanje rizikom i smanjenje izloženosti.

Namera pristupa „odbrane u dubini“ je da sloj vašeg okruženja osigura da nema eksploatabilnih pojedinačnih slabih tačaka, tačaka kvara (engl. *points of failure*) koje uljezu ili nepouzdanoj strani mogu omogućiti pristup podacima uskladištenim u MongoDB bazi podataka.

Sigurna okruženja koriste sledeće strategije za kontrolu pristupa, a više detalja je dostupno u odeljku Izloženost i sigurnost mreže (engl. *Network Exposure and Security*) u dokumentaciji.

- **Mrežni filter.** Korišćenjem filtera kao što su zaštitni zid (engl. *firewall*) i ACL[20] pravila rutera, konekcije sa MongoDB iz nepoznatih sistema mogu biti blokirane.
  Zaštitni zidovi bi trebalo da ograniče i dolazni i odlazni saobraćaj ka/sa određenog porta na pouzdane i nepouzdane sisteme. Za najbolje rezultate i da biste umanjili ukupnu izloženost, uverite se da samo saobraćaj iz pouzdanih izvora može doći do *mongod* i *mongos* instanci, i da se instance *mongod* i *mongos* mogu povezati samo na pouzdane izlaze. Pored toga, nepotrebne sistemske usluge treba deaktivirati.
- **Povezivanje IP adresa.** *bind_id* podešavanje za *mongod* i *mongos* instance ograničava mrežne interfejse na kojima će MongoDB programi osluškivati dolazne konekcije.
- **Pokretanje na VPN-u.** Ograničite MongoDB programe na lokalne mreže koje nisu javne i virtuelne privatne mreže. Virtuelne privatne mreže (engl. *Virtual Private Networks* - VPNs) omogućavaju povezivanje dve mreže preko šifrovane poverljive mreže sa ograničenim pristupom. Korsnici MongoDB obično konfigurišu SSL, a ne IPSEC protokole radi prednosti u performansama.
- **Namenski OS korisnički nalog.** Treba kreirati i koristiti korisnički nalog posvećen MongoDB za pokretanje MongoDB izvršnih datoteka. MongoDB ne bi trebalo da radi kao „korenski“ (engl. „*root*“) korisnik.
- **Dozvole za fajl sistem.** Serveri koji pokreću MongoDB treba da koriste dozvole sistema datoteka koji sprečavaju korisnike da pristupe datotetkama koje je kreirao MongoDB.

[20] Engl. *Access control list* – Lista za kontrolu pristupa. U računarskoj bezbednosti lista za kontrolu pristupa je lista dozvola povezanih sa sistemskim resursom (objektom). ACL određuje kojim korisnicima ili sistemskim procesima je odobren pristup objektima, kao i koje su radnje dozvoljene nad datim objektima. Svaki unos u ACL se sastoji od predmeta i operacije.

MongoDB konfiguracioni fajlovi i ključne datoteke klastera treba da budu zaštićene kako bi neovlašćenim korisnicima onemogućili pristup.

- **Ubacivanje upita.** Kako klijentski program sastavlja upit u MongoDB, on pravi BSON objekat, a ne string. Zahvaljujući tome tradicionalni napadi SQL ubačenih upita ne bi trebalo da predstavljaju rizik za sistem za upite predate kao BSON objekti.
  Međutim, nekoliko MongoDB operacija dozvoljava procenu proizvoljnih JavaScript izraza i treba voditi računa da se izbegnu zlonamerni izrazi. Srećom, većina upita može biti izražena u BSON formatu, a u slučajevima kada je potreban JavaScript, moguće je kombinovati JavaScript i BSON tako da se vrednosti koje oređuju korisnici evaluiraju kao vrednosti, a ne kao kod.
  MongoDB takođe omogućava administratoru da konfiguriše MongoDB server kako bi sprečio izvršavanje JavaScript skripti. Ovo će sprečiti pokretanje *MapReduce* poslova, ali okvir za agregaciju može se koristiti kao alternativa u mnogim slučajevima korišćenja.
- **Kontrole fizičkog pristupa.** Pored gore opisanih logičkih kontrola, kontrola fizičkog pristupa serverima, medijima za skladištenje i sigurnosnim kopijama pruža kritičnu zaštitu okruženja. [1]

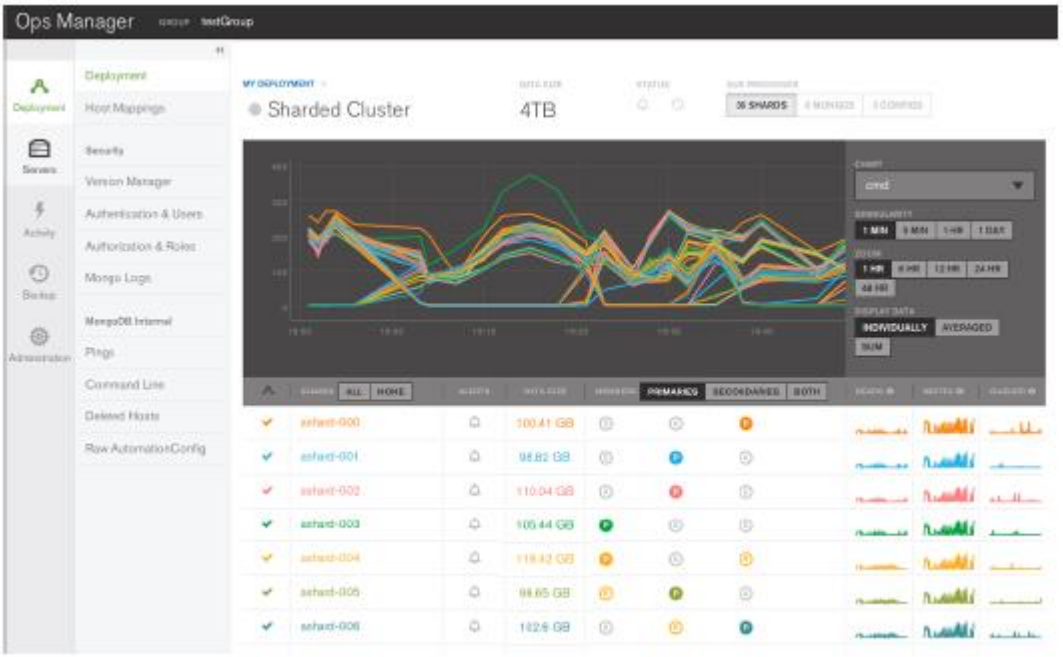
3.6.1 Nadgledanje baze podataka

Proaktivno praćenje svih komponenti u IT okruženju uvek je najbolja praksa. Performanse i dostupnost sistema zavise od pravovremenog otkivanja i rešavanja potencijalnih problema pre nego što ih korisnici uoče.

Iz perspektive bezbednosti baze podataka, nadgledanje je presudno za identifikovanje potencijalnih iskorišćavanja u realnom vremenu, čime se smanjuje uticaj bilo kog kršenja. Na primer, nagli vrhunac opterećenja centralnog procesora (engl. *Central processing unit* - CPU) i memorije sistema domaćina i visoke vrednosti brojača operacija u bazi podataka mogu ukazivati na napad uskraćivanja usluge (engl. *Denial of Service*). MongoDB se isporučuje sa raznim alatima, uključujući *mongostat* i *mongotop* koji se mogu koristiti za nadgledanje baze podataka.

Najopsežnije rešenje za nadgledanje pruža *MongoDB Ops Manager*, što je najjednostavniji način za pokretanje MongoDB baze podataka. Ops Manager olakšava operativnim timovima da nadgledaju, osiguraju, prave rezervne kopije i skaliraju MongoDB bazu podataka. Ops Manager je dostupan sa MongoDB Enterprise Advanced.

Grafikoni karakteristika, prilagođene kontrolne table i automatsko upozoravanje, Ops Manager prati 100+ ključnih metrika baze podataka i pokazatelja stanja sistema, uključujući operacije brojanja, iskorišćenja memorije i centralnog procesora, status replikacije i otvorene konekcije, redove i bilo koji status čvora. Izgled Ops Manager grafikona, kontrolnih tabli i automatskih upozorenja prikazan je na slici 7.



Slika 7: Ops Manager nudi grafikone, prilagođene kontrolne table i automatsko upozoravanje

Metrike se sigurno prijavljuju Ops Manager-u, gde se obrađuju, objedinjuju i vizualizuju u pretraživaču, omogućavajući administratorima da lako utvrde stanje MongoDB baze podataka u realnom vremenu. Pogledi se mogu zasnivati na eksplicitnim dozvolama, tako da vidljivost projektnog tima može biti ograničena na njihove sopstvene aplikacije, dok sistemski administratori mogu nadgledati sve primene MongoDB baze podataka u organizaciji.

Ops Manager omogućava administratorima da postave prilagođena upozorenja kada su ključne metrike van dometa. Upozorenja se mogu konfigurisati za niz parametara koji utiču na pojedinačne hostove, skupove replika, agente i rezervne kopije. Upozorenja se mogu slati putem SMS-a i elektronske pošte (email-a), ili integrisati u postojeće sisteme za upravljanje kao što su *PagerDuty* i *HipChat* kako bi proaktivno upozorili na potencijalne probleme pre nego što prerastu u skupe prekide rada. [1]

3.6.2 Oporavak od katastrofe: Rezervne kopije i oporavak u trenutku

Ops Manager rezervne kopije se održavaju kontinuirano, samo nekoliko sekundi iza operativnog sistema. Ako MongoDB doživi neuspeh, najnovija rezervna kopija zaostaje samo nekoliko trenutaka, minimizirajući izloženost gubitku podataka. Ops Manager je jedino MongoDB rešenje

za rezervne kopije koje nudi trenutni oporavak skupova replika i snimke širom klastera i izoštrenih klastera. Možete da vratite tačno trenutak koji vam je potreban, brzo i sigurno. [1]

3.6.3 Održavanje baze podataka

Uvek se trudite da koristite najnovije izdanje MongoDB baze podataka i upravljačkih programa koje su sertifikovane za proizvodnju i da primenite najnoviji skup zakrpa. Dok MongoDB Enterprise Advanced korisnici dobijaju pristup zakrpama za hitne slučajeve, popravke za sigurnosne propuste dostupne su svim MongoDB korisnicima čim budu objavljeni. [1]

4. Zaključak

Sigurnost podataka i privatnost je od ključne važnosti. Podaci prikupljeni sa društvenih medija, mobilnih uređaja i senzorskih mreža postali su podjednako osetljivi kao i tradicionalni podaci generisani iz pozadinskih sistema. Iz tog razloga, tehnologije velikih količina podataka moraju da se razvijaju kako bi zadovoljile standardne regulative koje zahtevaju industrije i Vlade.

Bezbednosna arhitektura baze podataka pre svega mora da pokriva upravljanje korisničkim pravima za ograničavanje pristupa osetljivim podacima, što se implementira pomoću autentifikacije i autorizacije. Takođe, radi obezbeđivanja sigurnosti baze podataka neophodno je evidentiranje operacija u bazi podataka u revizijskom tragu, zaštita podataka šifrovanjem i kontrola okruženja i procesa.

Zahvaljujući sveobuhvatnim kontrolama za upravljanje korisničkim pravima, revizijom i šifrovanjem, zajedno sa najboljim praksama u zaštiti okruženja, MongoDB može ispuniti prethodno opisane zahteve najbolje prakse, omogućavajući korisnicima da otkrivaju moć velikih količina podataka na mreži.

Kao što je prikazano u ovom radu, sa MongoDB Enterprise bazom podataka organizacije imaju koristi od širokih mogućnosti za odbranu, otkrivanje i kontrolu pristupa vrednim velikim količinama podataka na mreži.

Možete započeti sa pregledom MongoDB bezbednosne dokumentacije (engl. *MongoDB Security Documentation*) i preuzimanjem MongoDB Enterprise za evaluaciju danas.

5. Literatura

[1] MongoDB Security Architecture
https://www.scribd.com/document/362371231/MongoDB-Security-Architecture-WP

[2] MongoDB Security Guide Master
https://www.scribd.com/doc/299778357/MongoDB-Security-Guide-Master

[3] *MongoDB High Availability*, Afshin Mehrabani, ISBN: 9781783986736
https://www.scribd.com/book/272074313/MongoDB-High-Availability

[4] *MongoDB Cookbook - Second Edition*, Dasadia Cyrus and Nayak Amol, ISBN: 9781785286827
https://www.scribd.com/book/342442814/MongoDB-Cookbook-Second-Edition

[5] Coarse-grained vs fine-grained
https://stackoverflow.com/questions/3766845/coarse-grained-vs-fine-grained

[6] INTRODUCTION TO RED HAT IDENTITY MANAGEMENT
https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/linux_domain_identity_authentication_and_policy_guide/introduction

[7] SQL | DDL, DQL, DML, DCL and TCL Commands
https://www.geeksforgeeks.org/sql-ddl-dql-dml-dcl-tcl-commands/