UDK 004.738.5:341.456
DOI 10.7251/ZRSNG1708068C
COBISS.RS-ID 7298584

Šifrovanje baze podataka sa više korisnika

Encryption in a multi-user database

Aleksandar Sandro Cvetković, Saša Adamović, Univerzitet Sinergija, Bijeljina

***Sažetak* — Kerberos je protokol za autentifikaciju. Dizajniran je da omogući jaku autentifikaciju za klijet-server aplikacije koristeći kriptografiju tajnog ključa. Kerberos je nastao kao rešenje za probleme mrežne sigurnosti i razvijen je od strane MIT-a. Kerberos protokol koristi jaku kriptografiju kako bi klijent mogao da dokaže svoj identitet serveru i obrnuto preko nesigurne mreže. Nakon što su klijent i server dokazali svoje identitete preko Kerberosa, oni takođe mogu šifrovatni njihovu kompletnu komunikaciju kako bi privatnost i integritet podataka bio osiguran. U ovom radu bavili smo se simulacijom rada Kerberos protokola. Detaljno su opisana podešavanja i funkcionalnosti kao i koje sve vrste implementacija postoje za Kerberos protokol i kakvu vrstu bezbednost pruža. Takođe je opisano koje su prednosti i mane ovog protokola. Kerberos protokol je predstavljen kao novo rešenje za autentifikaciju i šifrovanje baze podataka sa više korisnika.**

***Ključne riječi* – Kerberos protokol; autentifikacija; šifrovanje baze podataka; Kerberos implementacija; KDC**

Abstract – Kerberos is an authentication protocol. It is designed to provide strong authentication for client-server applications by using secret key cryptography. Kerberos has emerged as a solution to network security issues and has been developed by MIT. The Kerberos protocol uses strong cryptography so that the client can prove his identity to the server and vice versa over the insecure network. Once the client and server have proven their identities through Kerberos, they can also encrypt their complete communications to ensure privacy and data integrity. In this paper we discussed the simulation of the Kerberos protocol operation. Detailed settings and functionality are described as well as all types of implementations that exist for the Kerberos protocol and what kind of security it provides. Furthermore we discuss the advantages and disadvantages of this protocol. The Kerberos protocol is presented as a new solution for authenticating and encrypting multiple-user database.

***Keywords* – Kerberos protocol; authentication; database encryption; Kerberos implementation; KDC**

I. UVOD

Šifrovanje baze podataka zna da bude jako kompleksan proces, zato što je potrebno poznavati sve najaktuelnije vrste napada, sagledati situaciju iz trenutne i buduće perspektive, poznavati kriptografske algoritme i pažljivo proceniti koje rešenje je najbolje za odgovarajući problem. Uzimajući u obzir kompleksnot problema za najbolje bezbednosno rešenje smatra se kombinacija kriptografskih mehanizama sa klasičnim bezbednosnim mahanizmima kao što su kontrola pristupa i privilegija. Osim kombinacije kvalitetnih mehanizama zaštite, potrebna je i kvalitetna implementacija kao i siguran nivo poverljivosti. Upravljanje kriptološkim ključevima je važan i specifičan parametar zaštite baza podataka. Jako je važno kako će upravljanje kriptološkim ključevima biti rešeno i uvek ide uz visok nivo poverljivosti. Uspeh kvalitetne zaštite takođe zavisi i od kvaliteta kriptološkog ključa, računar sam nije dovoljno dobar da pravi kvalitetne ključeve zato što lako može doći do predvidivih podataka. Zato se danas kvalitetni kirptološki ključevi prave dodavanjem nekih prirodnih elemenata kao npr. šum. Međutim ni to nije dovoljno ako je ključ kompromitovan. Zbog toga je potrebno da ona strana koja čuva i upravlja kriptološkim ključevima bude od poverenja. Šifrovanje baze podataka moguće je obaviti na nivou memorije, na nivou baze podataka preko SUBP-a (engl. DBMS – Database Managment System) ili na nivou aplikacije. Pored šifrovanja potrebno je osigurati i autentifikaciju korisnika.

Svi ljudi imaju urođene mogućnosti za autentifikaciju. Ljudi se međusobno veoma lako autentifikuju bilo da to rade po izgledu, jednostavnim pitanjima, tajnim informacijama, pokretima tela, govorom i naglaskom ili na skroz neki drugi način. U poređenju sa računarima, situacija je mnogo kompleksnija, zato što računari nemaju te ljudske mogućnosti. Razvoj računara i tehnologija znatno napreduje i teži se ka tome da računari stignu ljude, međutim još uvek se ne mogu toliko porediti. Računari koriste razne metode za autentifikaciju od kojih je najzastupljenija metoda upotrebom lozinke. Lozinka predstavlja tajnu informaciju pomoću koje se određuje da li je osoba za računarom ta osoba za koju se predstavlja. Obično lozinke se čuvaju u bazi podataka. Međutim ova vrsta autentifikacije nije toliko pouzdana zato što postoje određene slabosti, takođe danas postoje znatno sigurnije metode za autentifikaciju (kao npr. biometrija) ali još uvek nisu toliko zaživele da zamene metodu upotrebom lozinke. Svaku lozinku je moguće otkriti samo je pitanje vremena koje je potrebno za otkrivanje te lozinke. Pošto su vremenom računari postajali sve jači i jači, vreme za probijanje lozinke se smanjivalo, zbog toga se danas zahteva da svaka lozinka bude kompleksna. To uključuje korištenje kombinacije malih i velikih slova, brojeva, znakova, određenu dužinu karaktera, sve što lozinku čini kompleksnom, sigurnom.

Prvi problem nastaje u ljudskom faktoru zato što je ljudima teško pamtiti ovako kompleksne lozinke, još ako koriste više različitih lozinki za različita mesta. Zbog toga ljudi umesto da koriste jake lozinke, koriste svoja imena, prezimena, datume,

ascvetkovic@sinergija.edu.ba

brojeve telefona i uopšteno predvidive, jednostavne i slabe lozinke. U ovom slučaju napadačima je dosta olakšan posao. Kao drugi problem jeste taj da lozinka putuje preko mreže i računari je najčešće šalju u običnom tekstu što znači da napadači mogu presresti pakete i ukrasti lozinku. Zbog toga je važno da pored toga što lozinka mora biti jaka bude i šifrovana.

Rešenje za sve navedene probleme jeste Kerberos protokol za autentifikaciju. Kerberos protokol omogućava korisnicima da lozinku koriste samo jednom i čak ni tada se lozinka ne šalje preko mreže. Takođe Kerberos protokol pruža šifrovanje i integritet poruka, brine se za to da osetljivi podaci nikada ne budu poslati u običnom tekstu nego uvek kao šifrovani. Kerberos pruža siguran mehanizam za autentifikaciju koji je od velike koristi i korisnicima i administratoru.

II. PREGLED U OBLASTI ISTRAŽIVANJA

Zaštita osetljivih podataka, pogotovo podataka uskladištenih u bazi podataka (engl. data at rest) od zlonamernog korisnika kao što je uljez ili korumpirani zaposleni može se postići kriptološkim šifrovanjem baze podataka. Čak i ako zlonamerni korisnik može da zaobiđe pravila kontrole pristupa ili dobije pristup sistemu datoteka (engl. file system) on će i dalje trebati odgovarajuće ključeve za dešifrovanje podataka. Gore navedeni primer ističe dva glavna zahteva koji uključuju šifrovanje podataka. [*1*]

- Sigurna tehnologija šifrovanja u cilju zaštite osetljivih podataka.
- Pouzdana šema za generisanje i upravljanje ključem.

Postoje brojne odluke o dizajnu i implementaciji koje treba razmotriti pre implementacije šeme kriptološkog šifrovanja baze podataka [*2*]. Potrebno je odgovoriti na pitanja kao što su:

1. Gde se šifrovanje odvija, na nivou memorije, na nivou baze podataka ili na nivou aplikacije?
2. Kako da se minimizuje broj korisnika koji imaju pristup ključu za dešifrovanje?
3. Da li ključevi trebaju biti smešteni na odvojenim lokacijama od podataka?
4. Da li delimično šifrovanje baze podataka pruža odgovarajuću sigurnost?
5. Uticaj performansi različitih strategija kriptološkog šifrovanja baza podataka?

A. Šifrovanje unutar SUBP-a

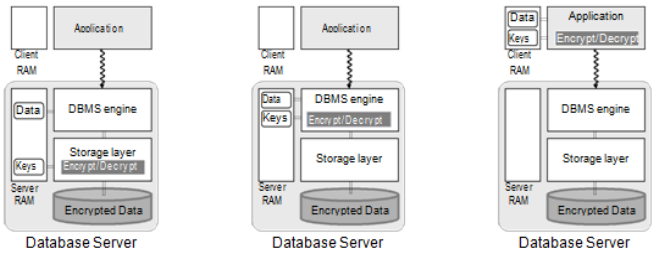
U ovoj strategiji šifrovanja podaci će biti šifrovani čim budu uskladišteni, što znači da će biti preneti preko mreže u običnom tekstualnom obliku. Podaci se dešifruju na serveru baze podataka, stoga ključevi za dešifrovanje se moraju preneti ili smestiti u bazu podataka što pruža nedovoljnu sigurnost protiv korumpiranog administratora baze podataka ili zlonamernog korisnika (engl. hacker) koji je uspeo da autentifikuje sebe kao administratora. Jedna od prednosti ove strategije šifrovanja jeste da je ona transparentna za aplikacije, nijedna promena nije potrebna na aplikaciji. Prema tome implementacija šifrovanja unutar SUBP-a je jednostavna, međutim postoje problemi kod performansi i bezbednosti koje treba razmotriti. Npr. neki SUBP-ovi nude ograničene mogućnosti šifrovanja, kao što su spori (npr. 3DES) ili nesigurni (npr. DES) algoritmi šifrovanja ili nedostatak mogućnosti selektivnog šifrovanja podataka. Pošto se podaci prenose u obliku običnog teksta, oni su u opasnosti prilikom prenosa. Sve operacije šifrovanja i dešifrovanja se odvijaju na serveru baze podataka čime se dodatno opterećuje server. Šifrovanje unutar SUBP-a može se izvesti na dva načina.

1) Šifrovanje na nivou memorije

Šifrovanje na nivou memorije, šifruje podatke u memoriji podsistema stoga je pogodno za šifrovanje celih fajlova i datoteka i za zaštitu podataka uskladištenih u bazi podataka. Međutim podsistem memorije ne poznaje šemu baze podataka niti njene korisnike pa zbog toga strategija šifrovanja ne može biti vezana za privilegije korisnika. Osim toga, izbor podataka koji se šifruju je ograničen na granularnost fajlova što može da dovede do povećanja zahtevnosti zbog nepotrebnog šifrovanja podataka. Potrebno je imati u vidu da postoji mogućnost da kopije osetljivih podataka koje nisu šifrovane mogu ostati u log ili privremenim (engl. temp) fajlovima.

2) Šifrovanje na nivou baze podataka

Šifrovanje na nivou baze podataka nudi veću fleksibilnost u pogledu na granularnost šifrovanja, zato što strategija šifrovanja može biti povezana sa šemom baze podataka. Podaci mogu biti šifrovani na nivou tabele, reda ili kolone. Npr. moguće je šifrovati samo određena polja tabele kao što je polje za lozinku (engl. password) ili broj socijalnog osiguranja (jmbg), ili neke redove koji su bazirani na logičnom stanju kao što je šifrovanje svih plata koje su iznad 2 hiljade KM mesečno. Šifrovanje na nivou baze podataka može degradirati performance jer komplikuje indeksiranje šifrovanih podataka. Potrebno je koristiti specijalne šifarske algoritme kao što je redosled šifrovanja kako bi se obavila pretraga šifrovanih indeksa.



Slika 1**.** Šifrovanje na nivou memorije, šifrovanje na nivou baze podataka i šifrovanje na nivou aplikacije

B. Šifrovanje izvan SUBP-a

Ako je šifrovanje podataka potrebno prilikom prenosa onda je prikladnije rešenje šifrovanje podataka izvan SUBP-a na nivou aplikacije. Na ovaj način podaci koji se prenose u tekstualnom obliku koji su uskladišteni i oni koji se pozivaju iz SUBP-a su u šifrovanoj formi. Takođe je povećana sigurnost tokom prenosa podataka i rešava se problem opterećenja servera baze podataka zbog operacija šifrovanja i

dešifrovanja, budući da se sve operacije šifrovanja i dešifrovanja obavljaju na nivou aplikacije. Međutim potrebno je modifikovati sve aplikacije kako bi mogle podržati sve mogućnosti šifrovanja i dešifrovanja. Kako bi ovo rešenje bilo što bolje iskorišteno preporučeno je da se koristi zajedno sa serverom za šifrovanje koji pruža usluge šifrovanja i dešifrovanja aplikacijama na solidan i dosledan način. Ovo rešenje odvaja ključeve za šifrovanje od šifrovanih podataka, budući da ključevi nikad ne napuštaju server za šifrovanje što sistem čini sigurnijim jer napadač mora dobiti pristup bazi podataka i serveru za šifrovanje. Kako bi ova strategija šifrovanja bila zaista sigurna i efektivna, važno je da između aplikacije i servera za šifrovanje postoji jaka strategija provere autentičnosti (engl. authentication) kojom se osigurava da samo ovlašteni korisnici mogu dešifrovati osetljive podatke. Server za šifrovanje bi trebao biti osiguran protiv napada pomoću pravilnog evidentiranja događaja (engl. event logging) i revizije. Zaključak, ovo rešenje nudi fleksibilnost u pogledu na algoritm šifrovanja koji može smanjiti troškove opterećenja i povećati sigurnost. Osim toga, ovo rešenje je vrlo skalabilno u odnosu na broj korisnika i šifrovanih baza podataka (tj. može se dodati više baza podataka bez modifikovanja servera za šifrovanje). [*3*]

C. Kerberos

Definicija za najsigurniji računar glasi: „Najsigurniji računar je računar koji nije povezan na mrežu ili je ugašen“. Međutim u današnjem svetu to nije realno niti prihvatljivo.

Kerberos je protokol za autentifikaciju na mreži. Dizajniran je kako bi pružao čvrstu autentifikaciju za klijent-server aplikacije koristeći simetrični ključ. Kerberos protokol je jedan od najpoznatijih protokola za autentifikaciju korisnika. Protokol je razvijen još davne 1980. godine na MIT (Massachusetts Institute for Technology) institutu u sklopu Athena istraživačkog projekta. Najveću popularnost protokol je stekao nakon implementacije u Windows operativne sisteme (Windows 2000 i Windows Server 2003), postoje implementacije Kerberos protokola i za druge operativne sisteme.

Kerberos se definiše kao siguran, SSO (Single-Sign On) protokol za autentifikaciju baziran na centralnom autentifikacionom entitetu kojem svi drugi entiteti u informacionom sistemu u potpunosti veruju (engl. trusted entity). Centralni autentifikacioni entitet u Kerberos sistemu naziva se KDC server (engl. Key Distribution Center), i predstavlja centralno skladište u kojem su uskladišteni svi autentifikacioni parametri svih entiteta u Kerberos sistemu. Nije određeno da samo jedan server može biti KDC, u slučaju da on postane nedostupan moguće je da ulogu KDC-a obavlja neki drugi server ili više njih. Kerberos protokol korisničke podatke nikada ne šalje mrežom u tekstualnom obliku (engl. plaintext), što ga čini otpornim na napade praćenjem i analizom mrežnog prometa (engl. sniffing). Sve poruke šalje u šifrovanom obliku sa ograničenim životnim vekom (engl. TGTs - Ticket Granting Tickets). Životni vek tiketa je od 8-12h. Ovakve poruke generiše KDC server na zahtev korisnika koji želi pristupiti određenom resursu u Kerberos sistemu. Ovakav način rada Kerberos protokol čini idealnim mehanizmom autentifikacije za računarske sisteme u kojima se ne može verovati svim korisnicima. Single-Sign On funkcionalnost podrazumeva proces u kojem se korisnik samo jednom prijavljuje na sistem, nakon čega mu je omogućen pristup svim mrežnim servisima koji podržavaju Kerberos protokol. Pored toga što je Kerberos protokol za autentifikaciju, njegovom implementacijom znatno se olakšavaju i ostala dva procesa koji zajedno čine AAA koncept (Authentication, Authorization, Auditing). Tri slova A čine sveto trojstvo svake mreže ako se posmatra iz ugla zaštite. Autentifikacija, autorizacija i kontrola su ključni delovi svake šeme za zaštitu mreže, ali često dolazi do zabune jer razlika između njih nije dovoljno jasna. Svaka od ovih komponenti ima različitu ulogu u zaštiti mreže. [*4*]

Autentifikacija predstvalja proces potvrđivanja identiteta određenog korisnika. Kako bi korisnik bio autentifikovan, od korisnika će biti zatražena informacija koja će dokazati njegov identitet. Za proveru ove informacije postoje tri kategorije: šta on zna, šta on ima i šta je on. Informacija nije vezana samo za jednu kategoriju, može pripadati jednoj ili više.

Kada je korisnik autentifikovan sledeći korak nakon autentifikacije je autorizacija. To znači da je korisniku odobren pristup u sistem ali kojim resursima korisnik unutar sistema može pristupati za to se brine autorizacija. Autorizacija na osnovu autentifikacije identiteta korisnika određuje kojim resursima korisnik može pristupiti a kojim ne. [*5*]

U koraku kontrole podaci se prikupljaju iz autentifikacije i autorizacije i smeštaju se u log fajl. U log fajlu su zapisani svi koraci preduzeti putem autentifikacije i autorizacije kako bi kasnije administrator sistema mogao da ih prekontroliše.

Za razliku od autentifikacije i autorizacije koje su proaktivne ova metoda je reaktivna. Omogućava administratoru pristup tragovima nakon što se incident dogodio.

Iz svih navedenih karakteristika može se zaključiti da Kerberos protokol pored svojih sigurnosnih svojstava donosi i druge pogodnosti, što je jedan od razloga njegove velike popularnosti.

Glavne karakteristike Kerberosa:

- Siguran je, nikada ne šalje lozinku ako nije šifrovana;
- Samo je jedno logovanje potrebno po sesiji. Podaci prikupljeni prilikom logovanja se kasnije koriste između resursa bez dodatnog logovanja;
- Koncept zavisi od treće strane od poverenja (TTP – Trusted Third Party). KDC predstavlja treću stranu od poverenja. KDC je svestan svih sistema unutar mreže i svi sistemi njemu veruju;
- Obavlja međusobnu autentifikaciju gde klijent dokazuje svoj identitet serveru i server dokazuje svoj identitet klijentu. [*6*]

Centar za distribuciju ključeva (KDC) je glavni deo Kerberos sistema. Sastoji se od tri komponente:

- baze podataka koja sadrži sve principale i ključeve koji su vezani za njih;

- Servera za autentifikaciju (AS);
- Servera za dodelu tiketa (TGS).

Iako su sve ove komponente odvojene tj. imaju različite uloge, prilikom implementacije dolaze u paketu. Međusobno komuniciraju i zajedno rade kao jedan proces. Pošto KDC uvek mora biti dostupan, moguće je imati više od jednog KDC-a. Ako jedan od njih nije dostupan postoji drugi koji će ga zameniti, pravilo je da unutar Kerberos okruženja (engl. realm) mora biti minimalno jedan KDC.

Kao ključni deo Kerberos sistema KDC je ujedino i najosetljiviji deo. Ukoliko je KDC kompromitovan celi sistem pada u vodu, zbog toga je neophodno dobro zaštititi KDC. Za razliku od PKI infrastrukture (engl. Public Key Infrastructure), nijedan javni ključ nije zahtevan. Kerberos KDC igra sličnu ulogu kao i vrhovni CA (engl. Certificate Authority) u PKI infrastrukturi [*7*]. Zahtevnost KDC-a je mala ali radi zaštite preporučuje se da svaki KDC radi na zasebnom računaru. Pošto se svi ključni podaci, uključujući i tajne podatke svakog principala unutar okruženja nalaze na svakom KDC-u unutar mreže, serveri na kojima rade moraju biti zaštićeni što je bolje moguće. Svaki KDC ima svoju bazu podataka. Windows 2000 i 2003 čuvaju bazu podataka unutar Aktivnog Direktorijuma (engl. Active Directory) koristeći LDAP. Svaka baza podataka mora biti sinhronizovana sa drugim bazama podataka u suprotnom sistem neće moći pravilno da funkcioniše.

Uloga servera za autentifikaciju (AS) je da izdaje TGT klijentima koji žele da se uloguju na Kerberos okruženje. Klijent ne mora da dokazuje identitet KDC-u zato što je poruka koju AS šalje nazad klijentu šifrovana lozinkom klijenta koju znaju samo KDC i klijent. Na osnovu toga KDC zna da samo pravi klijent može dešifrovati poruku i nastaviti komunikaciju. Ukoliko je klijent iskoristio pogrešnu lozinku tiket će biti dešifrovan u nešto beskorisno i klijent će ponovo morati da unese lozinku [*8*].

Za razliku od AS-a koji izdaje TGT kako bi klijent mogao komunicirati sa TGS-om, TGS izdaje ST kako bi klijent dalje komunicirati sa odgovarajućim serverom. TGT služi kako bi TGS mogao da verifikuje da li je klijent kontaktirao KDC tj. da li je klijent autentifikovan. TGS proverava da li je TGT šifrovan glavnim KDC ključem. Ako je uspeo da dešifruje TGT znači da jeste, u suprotom prekida proces. Uz TGT poruku TGS dobija i „ticket request“ koji služi kako bi TGS znao za koji server treba da izda ST. Nakon što je provera uspešna TGS izdaje ST, šifruje i šalje klijentu [*9*].

MIT je prva i glavna referenca za implementaciju Kerberosa v4 i v5. Mnoge velike institucije kao što su univerziteti koriste MIT KDC-ove kako bi rešili problem autentifikacije. Dostupan je na platformama kao što su: Apple Mac OS X, Sun Solaris i Redhat Linux, Android a posebno postoji verzija za Windows operativne sisteme „Kerberos for Windows“. MIT podržava različite tipove šifrovanja: DES, 3DES, RC4 i AES. Ovaj način implementacije podržava Kerberos API i GSS-API. [*10*].

Heimdal implementacija je mlađa nego MIT, ali kao i MIT implementacija Heimdal pruža punu podršku za Kerberos v4 i v5 kao i tiket prevodioc. MIT je planski bio namenjen samo za upotrebu unutar US-a, pošto je Heimdal nastao u Švedskoj i održava ga Stockholm Univerzitet Heimdal se smatrao kao implementacija Kerberosa van US-a.

Kerberos Microsoft implementacija koja je integrisana u Windows 2000 i druge je mlađa i od Heimdal implementacije. Windows implementacija podržava samo Kerberos verziju 5 i nema podršku za klijente verzije 4.

Pored ove tri osnovne Kerberos implementacije postoje i druge: GNU Shishi, Spring, Sun's Java, Oracle implementacija kao i mnoge druge. Važan faktor koji je prisutan prilikom svake implementacije jeste kompatibilnost.

Tabela 1. Kompatibilnost između Kerberos implementacija

	KDC			
Client	Active Directory	Heimdal	MIT Kerberos 5	Shishi
Active Directory	✓			
Heimdal	✓	✓	✓	✓
MIT Kerberos 5	✓	✓	✓	✓
Shishi	✓	✓	✓	✓
Service	Active Directory	Heimdal	MIT Kerberos 5	Shishi
Active Directory	✓			
Heimdal	✓	✓	✓	✓
MIT Kerberos 5	✓	✓	✓	✓
Shishi	✗	✗	✗	✗

Tabela 1. prikazuje rezultate kompatibilnosti između svake implementacije. Rezultati su pokazali da Active Directory principali (klijent i server) rade samo na Active Directory KDC-u. Jedino Shishi 1.0.2 verzija koja je bila korištena za testiranje u pogledu na server nema kompatibilnost ni sa jednom drugom implementacijom zbog nedostatka GSS-API-a, mnoge Kerberos implementacije koje se koriste u kombinaciji sa GSS-API-jem nisu u mogućnosti da komuniciraju sa Shishi serverom. MIT i Heimdal implementacije su potpuno kompatibilne sa svim ostalim implementacijama [*11*].

III. PREDLOŽENO REŠENJE

A. *AS_REQ (1)*

Postupak autentifikacije započinje korakom koji se naziva inicijalizovanje tako što korisnik unosi svoje korisničko ime i lozinku. Računar korisnika vrši transformaciju lozinke u simetrični ključ upotrebom neke heš funkcije. Taj ključ korisnik deli sa KDC-om. Nakon toga klijent (računar korisnika) šalje zahtev AS_REQ (komanda kinit) AS-u unutar KDC-a za TGT-om.

B. *AS_REP (2)*

Nakon što je AS primio zahtev od klijenta, AS proverava da li se klijent nalazi u bazi podataka tako što poredi heš vrednosti, heš koji je klijent poslao i heš koji se nalazi u bazi podataka. Pored klijenta AS proverava i vreme koje je dobio uz zahtev i upoređuje ga sa lokalnim vremenom. Dozvoljena vremenska razlika je najviše 5 minuta, ukoliko je razlika veća od 5 minuta AS klijentu vraća poruku o grešci. Na ovaj način sistem se štiti od napada zasnovanog na ponovnom slanju poruka (engl. replay attack).

Nakon što je klijent uspešno autentifikovan kao odgovor AS_REP na zahtev klijenta AS šalje TGT klijentu koji je šifrovan TGS tajnim ključem i šalje sesijski ključ prijave (engl. logon session key) šifrovan tajnim ključem klijenta.

Sesijski ključ prijave generiše KDC i uz pomoć njega korisnik neće morati stalno da unosi lozinku kako bi komunicirao sa serverom. Upravo ovaj proces se naziva SSO (Single-Sign On). Na ovaj način samo klijent koji ima odgovarajući ključ može dešifrovati poruku, time je sistem zaštićen od napada praćenjem (engl. sniffing attack).

AS_REP odgovor se sastoji iz dva dela, prvi deo je deo koji klijent može da dešifruje i to je sesijski ključ prijave za dalju autentifikaciju. Drugi deo je TGT koji je šifrovan sa tajnim ključem TGS i taj deo poruke klijent nije u mogućnosti da dešifruje ali važan je jer će se koristiti u daljoj komunikaciji. TGT će biti potreban za odobrenje pristupa mrežnim resursima. TGT takođe u sebi sadrži sesijski ključ prijave kako KDC ne bi morao da ga pamti.

C. TGS_REQ (3)

Nakon što je klijent dobio odgovor, klijent dešifruje šifrovani sesijski ključ prijave sa svojim tajnim ključem. Ukoliko je klijent uspešno dešifrovao AS_REP odgovor, klijent smešta sesijki ključ prijave i TGT u „ticket cache“ čime je okončana autentifikacija. Posle ovog koraka klijent je autentifikovan. Klijentu više nije potreban njegov tajni ključ jer će uvek moći da dešifruje poruke dobijene od KDC-a uz pomoć sesijkog ključa prijave. Uloga tajnog ključa klijenta je bila samo kako bi klijent dobio šifrovani sesijski ključ prijave generisan od strane KDC-a. Pošto klijent više neće koristiti svoj tajni ključ, klijent ga zaboravlja.

Klijent još uvek nije dobio pristup mrežnim resursima, dobio je samo potrebne parametre za komunikaciju sa TGS-om. Kada klijent želi da kontaktira server, klijent šalje TGS_REQ zahtev TGS-u unutar KDC-a. Unutar tog zahteva klijent šalje TGT i autentifikator generisan od strane klijenta šifrovan sesijskim ključem prijave.

D. TGS_REP (4)

Nakon što su parametri stigli do TGS-a unutar KDC-a. TGS dešifruje TGT svojim tajnim ključem sa kojim je TGT prvenstveno i bio šifrovan. Na ovaj način KDC opet dobija sesijski ključ prijave.

TGS onda generiše novi sesijski ključ servera (engl. server session key) i kopiju tog ključa šifruje sa glavnim serverskim ključem (engl. server master key). Pomoću ovog procesa nastaje ST (engl. Service Ticket). Glavni serverski ključ međusobno dele KDC i server, to znači da KDC i server imaju jedan primerak ovog ključa.

Još jedna kopija sesijskog ključa (engl. client session key) se šifruje sa sesijskim ključem prijave. Nakon toga KDC šalje klijentu šifrovani sesijski ključ klijenta i ST.

E. AP_REQ (5)

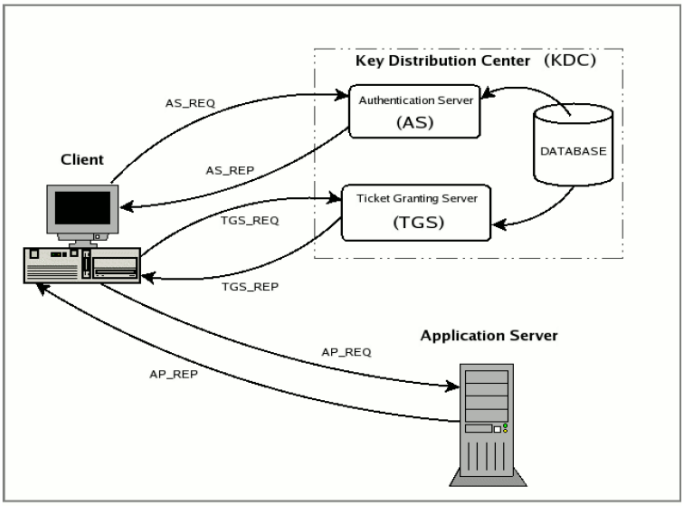
Kada je klijent primio odgovor od KDC-a. Klijent koristi sesijski ključ prijave da dešifruje sesijski ključ klijenta. Nakon toga sesijski ključ klijenta i ST se čuvaju u „ticket cache-u“ klijenta. Klijent nije u mogućnosti da dešifruje ST jer ne poseduje glavni serverski ključ ali mu šifrovan ST treba za dalju komunikaciju. Klijent sada ima odgovarajuće informacije kako bi komunicirao sa serverom.

Klijent šalje serveru zahtev AP_REQ koji sadrži ST i autentifikator koji je šifrovan sesijskim ključem klijenta.

F. AP_REP (6)

Nakon što je server prihvatio zahtev, server dešifruje ST sa glavnim serverskim ključem. Server tada dobija sesijski ključ servera sa kojim dešifruje autentifikator. Važno je napomenuti da sesijski ključ klijenta i sesijski ključ servera su ustvari isti ključevi, samo radi lakšeg razumevanja nose različite nazive.

Ako je server uspešno dešifrovao poruku, server zna da je klijent autentifikovan od strane KDC-a. Na taj način klijent je uspešno autentifikovan i od strane servera [*12*]. (Slika 2.)



Slika 2. Šematski prikaz Kerberos protokola

Kerberos autentifikacija pruža više prednosti nego ostale metode za autentifikaciju na mreži, unutar Kerberos okruženja svaki entitet veruje svakom entitetu tokom čitave komunikacije. Prednosti:

Obostrana autentifikacija - Kada dve strane kao što su klijent i server ili server i server komuniciraju, komunikacija se obavlja preko treće strane od poverenja TTP tj. KDC. Cela komunikacija se odvija tako što svaki klijent i svaki server veruju KDC-u i uz pomoć njega imaju međusobno poverenje.

Lozinke - Unutar Kerberos 5 okruženja lozinka se nikada ne šalje preko mreže u otvorenom tekstu. Lozinka korisnika služi kako bi korisnik šifrovao prvu poruku koju šalje KDC-u. Na ovaj način komunikacija je šifrovana i bezbedna od napada praćenjem (sniffing).

Integrisane sesije - Kada je klijent uspešno autentifikovan u Kerberos okruženju, klijent dobija tiket sa vremenskim pečatom odnosno vremenom trajanja. Dokle god je taj tiket važeći klijent je autentifikovan i može pristupati bilo kom resursu unutar Kerberos orkuženja bez potrebe da se ponovo autentifikuje. Ako je sesija klijenta još uvek aktivna ali je vreme tiketa isteklo, klijent može zatražiti drugi tiket.

Obnovljive sesije - Jednom kada se klijent i server međusobno autentifikuju, više nema potrebe da to ponovo rade. Klijent od servera dobija informacije koje će moći koristiti uvek za pristup tom serveru. Pomoću te informacije server prepoznaje klijenta i autentifikuje ga ponovo. Na ovaj

način nema više potrebe za KDC-om što znači da se konekcija između klijenta i servera još brže odvija nego prvi put. [*13*]

Kerberos 5 protokol je veoma siguran protokol ali kao i sve u ovom svetu i on ima svoje slabosti. Slabosti:

Dostupnost KDC-a - Jedna od slabosti jeste da Kerberos zahteva dostupnost centralnog servera (KDC-a). Ako bi taj server bio blokiran svima bi pristup bio onemogućen. Ovaj problem se može rešiti upotrebom više od jednog servera.

Vremensko podešavanje - Tehnologija je takođe osetljiva na vremenska podešavanja. Kompletan sistem neće funkcionisati ako satovi na svim hostovima nisu sinhronizovani. Dozvoljena razlika je 5 minuta.

Kompromitovanje KDC-a - S obzirom da su svi tajni ključevi korisnika smešteni u centralnom serveru (KDC-u), kompromitovanjem tog servera kompromitovani su i svi tajni ključevi korisnika.

Kompromitovanje lokalnog računara - Ukoliko se ne koriste pametne kartice, Kerberos je ranjiv ukoliko je lokalni računar kompromitovan ili neki virus (malware) pokupi lozinku. [*14*]

Kerberos protokol može biti nadograđen, ne u smislu menjanja već postojećeg nego u smisli dodavanja novih komponenti koje će dodatno pojačati sigurnost. Danas skoro sve aplikacije koriste internet konekciju za neku namenu, važno je da te aplikacije imaju sigurnu i pouzdanu konekciju. Upotrebom Kerberos protokola rešen je problem autentifikacije unuter mreže, sprečen je napad ponovnim slanjem poruke (replay attack) i napad praćenjem (sniffing) [*15*], kompletna komunikacija između klijenta i servera je šifrovana, osiguran je integritet i privatnost podataka i rešen je problem šifrovanja baze podataka sa više korisnika.

IV. ZAKLJUČAK

Na početku istraživanja, u ovom radu utvrđeno je da proces šifrovanja baze podataka nije jednostavan, nego sadrži niz kompleksnih segmenata. Istraživanjem smo došli do saznanja da pored glavnog dela, šifrovanja baze podataka, postoji još mnogo faktora koji upravo utiču na to šifrovanje i od kojih zavisi kvalitet samog šifrovanja. Nije dovoljno samo odabrati neki algoritam za šifrovanje i šifrovati kompletnu bazu podataka kako bi problem bio rešen. Potrebno je sagledati situaciju u kojoj se nalazi dati problem, na osnovu nje pažljivo proceniti da li je potrebno šifrovati sve podatke ili samo određene kao i koji algoritam za šifrovanje je najprikladniji za datu situaciju, kako po performansama tako i po sigurnosti.

Daljim radom, istražili smo da obično nije dovoljno koristiti jedan sloj zaštite, npr. samo šifrovanje podataka jer se takav vid zaštite u praksi pokazao kao slab. Za kvalitetnu zaštitu smatra se višeslojna zaštita. Nema garancije da nakon što se podaci šifruju neko neće uspeti da ih dešifruje i dođe do njih, zbog toga se za kvalitetnu zaštitu smatra ona zaštita koja napadaču povećava vreme koje je njemu potrebno za razbijanje svakog sloja zaštite i da dođe do podataka.

Istraživanje je pokazalo da postoje protokoli koji upravo nude višeslojnu zaštitu i visok nivo bezbednosti. Jedan takav protokol čini Kerberos.

Postavljeni su čvrsti ciljevi istraživanja da se obradi i simulira Kerberos protokol, prikažu njegove prednosti i slabosti kao i koje sve implementacije Kerberos protokola postoje.

Osnovna motivacija ovog istraživanja je da se razvije novo rešenje za šifrovanje baze podataka koje će biti pouzdano i stabilno, pružati sigurnu komunikaciju između klijenta i servera i omogućiti jaku autentifikaciju korisnika.

ZAHVALNICA

Zahvaljujem se Univerzitetu Sinergija zbog pružanja potrebnih uslova za kvalitetno akademsko obrazovanje, kao i svim profesorima koji su doprineli mom akademskom obrazovanju. Posebnu zahvalnost iskazujem mentoru prof. dr. Saši Adamoviću koji je uvek bio tu sa kvalitetnim savetima, usmeravao me na pravi put, doprineo mom akademskom obrazovanju i čovek koji je uzor svim studentima.

LITERATURA

[1] Mladen Veinović, Goran Šimić, Aleksandar Jevremović, and Igor Franc, *Baze podataka*. Beograd: Univerzitet Singidunum, 2013.

[2] Mladen Veinović and Saša Adamović, *Kriptologija 1*. Beograd: Univerzitet Singidunum, 2013.

[3] Zoe Paraskevopoulou and Nick Giannarakis, "Database Security & Cryptography," National Tehnical University of Athens, School of Electrical and Computer Engineering, 2013.

[4] CARNet CERT, "Implementacija Kerberos protokola u Linux okruženjima," 2010.

[5] S. P. Miller, B. C. Neuman, J. I. Schiller, and J.H. Saltzer, "PROJECT ATHENA TECHNICAL PLAN," *Section E.2.1 Kerberos Authentication and Authorization System*, 1988.

[6] Intel AMT Implementation and Reference Guide. Introduction to Kerberos Authentication. [Online]. https://software.intel.com/sites/manageability/AMT_Implementation_and_Reference_Guide/default.htm?turl=WordDocuments%2Fintroductiontokerberosauthentication.htm

[7] Milan Milosavljević and Saša Adamović, *Kriptologija II (Osnove za analizu i sintezu šifarskih sistema)*. Beograd, Srbija: Univerzitet Singidunum, 2017.

[8] Jason Garman, *Kerberos The Definitive Guide*. USA, 2003.

[9] William Stallings,.: Prentice Hall, 2005.

[10] MIT Kerberos Documentation. MIT Kerberos features. [Online]. https://web.mit.edu/kerberos/krb5-devel/doc/mitK5features.html

[11] Esan Wit and Mick Pouw, "Cross-realm Kerberos implementations," 2014.

[12] Fulvio Ricciardi. (2007) KERBEROS PROTOCOL TUTORIAL. [Online]. http://www.kerberos.org/software/tutorial.html#1.3.5

[13] Laura Gittins. What Are the Advantages of Kerberos Authentication? [Online]. https://itstillworks.com/advantages-kerberos-authentication-4863.html

[14] Bill Brenner. (2008) Kerberos: Authentication with some drowbacks. [Online]. http://searchsecurity.techtarget.com/news/1308058/Kerberos-Authentication-with-some-drawbacks

[15] XIAOHONG YUAN et al., "Visualization Tools for Teaching Computer Security," North Carolina A&T State University, 2010.