VI - Sigurnosni protokoli

S A D R Ž A J

1. Secure Shell Protokol (SSH)
2. Kerberos protokol
3. Radius protokol

6.1 – SSH (Secure Shell protocol)

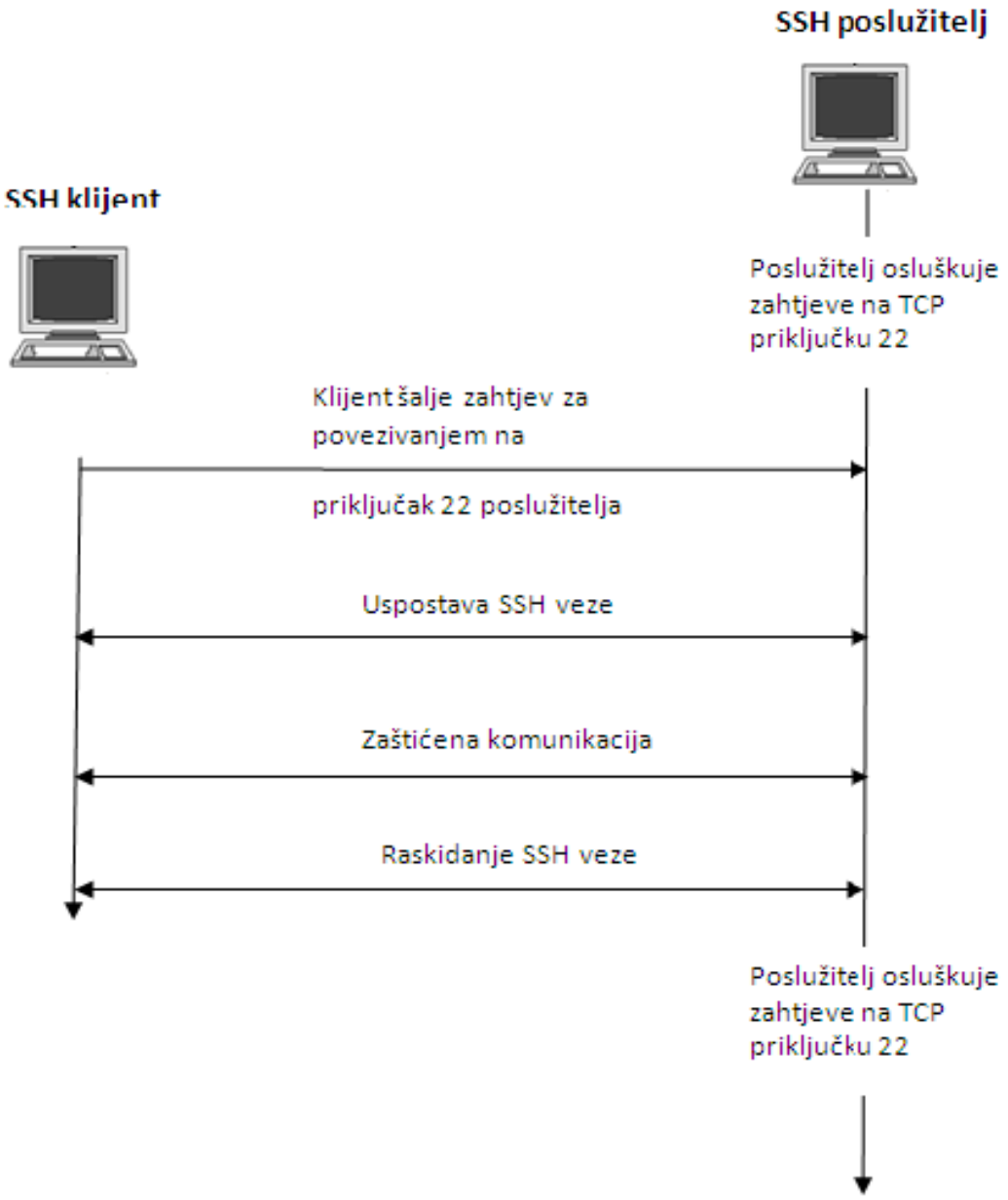
- Komunikacioni protokoli poput TELNET i FTP, rsh (*remote shell*), rcp (*remote copy*) i ostali, kojima se razmenjuju podaci između udaljenih računara, podatke šalju u **nešifrovanom obliku**
- Postoji realna opasnost za bezbednost podataka koje ti "nesigurni" protokoli prenose jer su oni izloženi potencijalnim napadačima.
- SSH protokol razvijen je kao zamena za postojeće nesigurne protokole tako što vrši šifrovanje podataka koje on prenosi.
- Protokol je razvijen na pricipu klijent/server tehnologije.
- Krajnje tačke komunikacije smatraju se **sigurnima**, dok se mreža koja ih povezuje smatra **nesigurnom**.
- **Tajnost**, **autentičnost** i **integritet** podataka osiguravaju se primenom snažnih kriptografskih metoda.
- ostupno više programskih verzija ovog protokola, daleko se najčešće koristi, posebno na operativnim sistemima UNIX/Linux, paket OpenSSH.
- Reč je o besplatnoj verziji SSH klijenta i servera koja omogućuje korišćenje većeg broja dodatnih mogućnosti protokola kao što su SSH tunelovanje i uspostavljanje VPN veze preko SSH kanala

6.1 – SSH (Secure Shell protocol)

- Komunikacioni protokoli poput TELNET i FTP, rsh (*remote shell*), rcp (*remote copy*) i ostali, kojima se razmenjuju podaci između udaljenih računara, podatke šalju u **nešifrovanom obliku**
- Postoji realna opasnost za bezbednost podataka koje ti "nesigurni" protokoli prenose jer su oni izloženi potencijalnim napadačima.
- SSH protokol razvijen je kao zamena za postojeće nesigurne protokole tako što vrši šifrovanje podataka koje on prenosi.
- Protokol je razvijen na pricipu klijent/server tehnologije.
- Krajnje tačke komunikacije smatraju se **sigurnima**, dok se mreža koja ih povezuje smatra **nesigurnom**.
- **Tajnost**, **autentičnost** i **integritet** podataka osiguravaju se primenom snažnih kriptografskih metoda.
- Dostupno je više programskih verzija ovog protokola, ali se najčešće koristi, posebno na UNIX/Linux sistemima, paket **OpenSSH**.
- **OpenSSH** je besplatna verzija SSH klijenta i servera koja omogućuje korišćenje većeg broja dodatnih mogućnosti protokola kao što su SSH tunelovanje i uspostavljanje VPN veze preko SSH kanala.

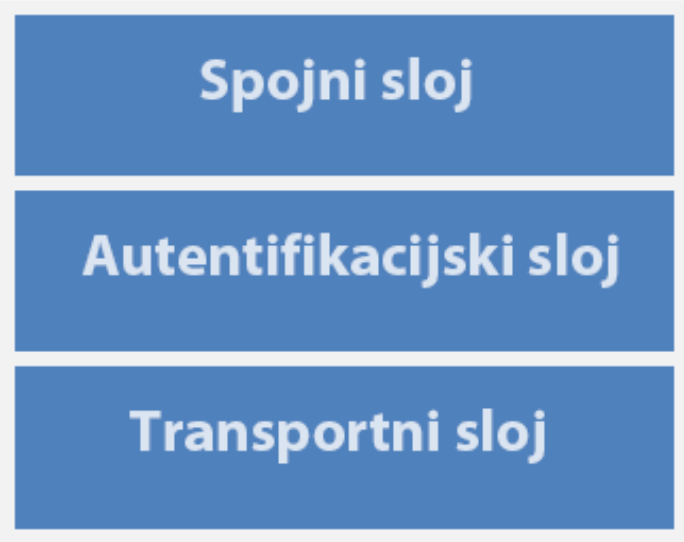
6.1 – SSH (Secure Shell protocol)

- SSH se temelji na modelu klijent/server
- To znači da se komunikacija odvija između dva entitea: klijenta i servera.
- Server sa jedne strane osluškuje zahteve na unapred određenom mrežnom portu, a klijent ih po potrebi šalje serveru.
- SSH server osluškuje zahteve klijenata na TCP portu 22.



6.1 – SSH (Secure Shell protocol)

- Uspostava komunikacije i sama komunikacija u SSH protokolu može se opisati troslojnom arhitekturom:
  1. **Transportni sloj** (*Transport Layer Protocol – RFC4253*),
  2. **Autentifikacijski sloj** (*Authentication Protocol – RFC4252*) i
  3. **Povezujući sloj** (*Connection Protocol – RFC4254*).
- Arhitektura protokola SSH i svaki sloj zasebno detaljno su opisani u odgovarajućim **RFC dokumentima**.
- Slojevi se nadograđuju jedan na drugi kao što je prikazano na sledećoj slici, a najniži, transportni sloj najčešće se nadograđuje na TCP/IP mrežu ali to nije obavezno.



6.1 – SSH (Secure Shell protocol)

1. Transportni sloj

- Može se koristiti i neka druga arhitektura koja garantuje pouzdan prenos podataka na nižim slojevima mrežne arhitekture.
- Ovaj sloj osigurava snažno šifrovanje i zaštitu integriteta podataka kao i autentifikaciju servera, a omogućeno je i sažimanje podataka
- Na ovom sloju klijent i server određuju i metode razmene ključeva, simetrične i asimetrične algoritme koji će se koristiti, te funkcije sažimanja, i algoritme utvrđivanja autentičnosti poruka.
- U okviru transportnog sloja koristi se i **Diffie-Hellman** metoda razmene simetričnog ključa.
- Za razmenu podataka nakon uspostavljene SSH veze na transportnom sloju se koristi binarni paketni protokol (*Binary Package Protocol*).
- Reč je o binarnom protokolu kojim se komunikacija odvija pomoću posebno organizovanih nizova bitova koji predstavljaju pakete.

6.1 – SSH (Secure Shell protocol)

2. Autentifikacijski sloj

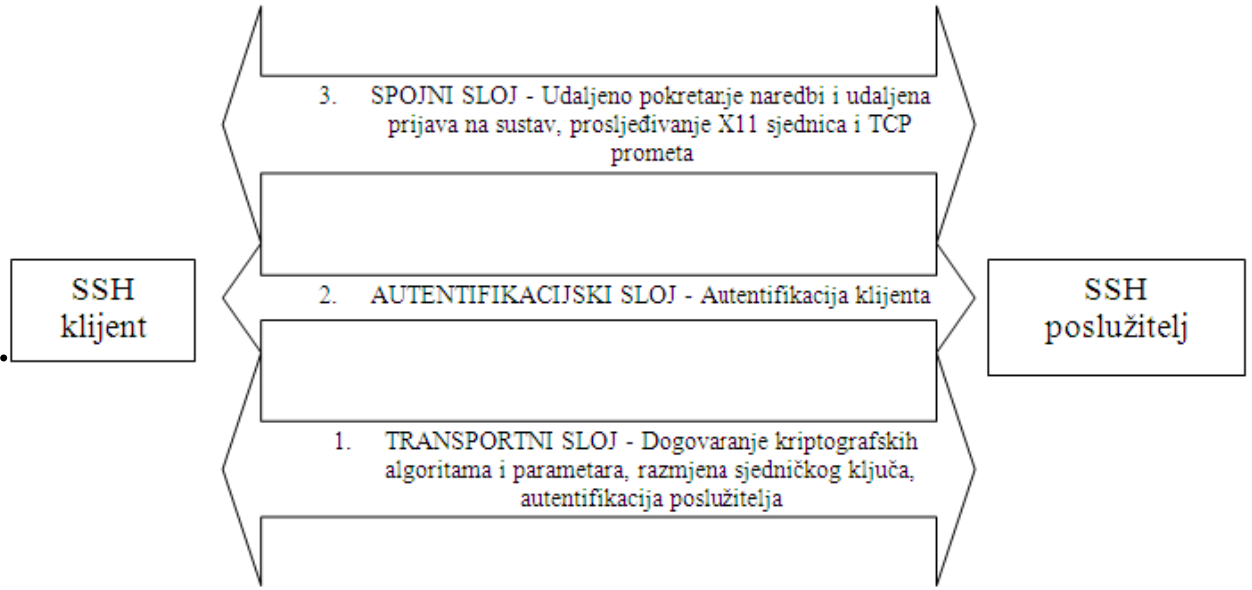
- Autentifikacijski sloj omogućava proveru identiteta klijenta na serveru, a komunikacija na tom sloju uspostavlja se tek nakon što je komunikacija na transportnom sloju već uspostavljena.
- Autentifikacija klijenta može se obaviti na više načina (koje server predlaže, a klijent bira) a neke od tih metoda su:
  1. **putem lozinke** (koja se šifrovana šalje SSH kanalom),
  2. **PKI**(*Public Key Infrastructure*)metode autentifikacije koja se zasniva na digitalnim potpisima i asimetričnim kriptografskim algoritmima (RSA, DSA), uključujući i proveru putem X.509 sertifikata
  3. autentifikacija zasnovana na proveri klijenta u bazi računara kojima je dopuštena autentifikacija, a koja se nalazi na serveru. Server, nakon što primi zahtev za autentifikacijom proverava klijentov FQDN (*Fully Qualified Domain Name*) i digitalni potpis i tako utvrđuje ispravnost klijenta koji se prijavio
- Ovaj sloj ostvaruje jedinstveni SSH komunikacijski kanal preko koga se može provesti sledeći sloj arhitekture, tzv. „povezujući sloj“.

6.1 – SSL (Secure Socket Layer)

3. Povezujući sloj

- Ostvaruju se udaljene prijave korisnika, udaljeno izvođenje naredbi, prosleđivanje TCP/IP i X11 veza
- Reč je o najvišem sloju SSH arhitekture na kome se sva komunikacija odvija putem jednog šifrovanog kanala.
- Ovde se virtuelno raspolaže proizvoljnim brojem kanala koje se međusobno razlikuju pomoću identifikatora.
- Sa ovim slojem korisnik je u direktnom dodiru.

- Naredbe koje se zadaju SSH programima (preko GUI interfejsa ili komandne linije) prvo obrađuje ovaj sloj.



6.2 – Kerberus protokol

- Predstavlja jedan je od najpoznatijih protokola za autentifikaciju
- Protokol se odlikuje brojnim funkcionalnostima i prednostima, a jedna od najznačajnijih je svakako ***Single Sign On*** (SSO) funkcionalnost
- Na taj način, korišćenjem Kerberos protokola, uklanja se potreba za upravljanjem velikim brojem korisničkih naloga i lozinki, a takođe se smanjuje vreme potrebno za pristup pojedinim servisima.
- Dodatna prednost sa sigurnosne strane je ta što Kerberos protokol korisničke lozinke nikad ne šalje mrežom u čistom tekstualnom obliku
- Budući da novije verzije Windows OS Kerberos protokol koriste kao primarni protokol za autentikaciju korisnika, Kerberos se često pogrešno smatra Microsoftovim proizvodom.
- Kerberos protokol razvijen je još davne 1980. godine na *Massachusetts Institute for Technology* (**MIT**) institutu u sklopu poznatog Athena istraživačkog projekta.
- Najveću popularnost protokol stekao implementacijom u Windows OS
- Postoje implementacije Kerberos protokola i za druge OS.

6.2 – Kerberus protokol

- Kerberos se definiše kao siguran, *single-sign-on* autentifikacioni protokol baziran na centralnom autentifikacijskom entitetu kome svi entiteti u informacionom sistemu u potpunosti veruju (*trusted entity*)
- Centralni autentifikacioni entitet u Kerberos sistemu naziva se **KDC** server (***Key Distribution Center***), i predstavlja centralno mesto u kome se čuvaju autentifikacioni parametri svih entiteta u Kerberos-u
- Ulogu KDC servera može obavljati i više servera
- Kerberos se naziva sigurnim jer lozinke računar.mrežom nikad ne šalje u čistom tekstualnom obliku, već u tu svrhu koristi specijalne šifrovane poruke ograničenog perioda trajanja-***tickets*** (trajanje tipično od **8-24h**)
- Ove poruke generiše KDC server na zahtev korisnika koji želi pristupiti određenom resursu u Kerberos sistemu.
- Nakon inicijalne prijave u sistem, pristup svim mrežnim resursima za korisnika je u potpunosti transparentan, što znatno olakšava rad u distribuiranim mrežnim okruženjima.
- Znatno se olakšavaju i ostala dva procesa koja zajedno čine poznati tzv. **AAA** (**Authentication**, **Authorization**, **Auditing**) koncept.

6.2 – Kerberus pojmovi

Kerberos realm i principali

- Svaki entitet Kerberos sistema, bez obzira da li se radi o korisniku, računaru, mrežnom servisu, serveru ili nečem trećem, opisan je sa odgovarajućim imenom u bazi KDC servera, koji se naziva **principal**.
- Svaki **principal** jedinstveno opisuje entitet u Kerberos sistemu i ima odgovarajuću strukturu definisanu specifikacijom protokola.
- Svaki principal u Kerberos sistemu poseduje i odgovarajući tajni ključ koji je poznat samo KDC serveru i entitetu o čijem se ključu radi.
- Tajni ključ koristi se za šifrovanje poruka u postupku autentikacije.
- **Opšta struktura** principala je sledeća: **identity/instance@realm**
  - **identity** - polje koje opisuje ime Kerberos entiteta (korisničko ime, mrežni servis, računar i sl.) - obavezno za svaki principal objekat.
  - **instance** - polje instance bliže opisuje Kerberos entitet i može se shvatiti kao opis grupe kojoj entitet pripada - nije obavezno.
  - **realm** – svaka posebna instalacija Kerberos sistema definiše jedinstveni *realm* koji opisuje sistem i koji se razlikuje od bilo kog drugog Kerberos okruženja - odgovara DNS imenu domena ali se piše velikim slovima

6.2 – Kerberus pojmovi

Key Distribution Center (KDC)

- KDC server predstavlja jezgro Kerberos sistema i njegova dostupnost neophodna je za funkcionisanje celog sistema.
- Iako se KDC server sastoji od tri različite komponente, sve su one najčešće integrisane u jedan program koji je pokrenut na odgovarajućem mrežnom serveru.
- **Tri komponente** koje čine KDC server su:
  1. **Baza sa svim principalima** unutar definisanog Kerberos realma s pripadajućim tajnim ključevima. Način na koji je implementirana baza sa ovim podacima zavisi od implementacije sistema. Kod Microsoft-a ovi se podaci čuvaju unutar **Active Directory** imenika, dok se kod Linux implementacija u tu svrhu koriste specijalizirane **LDAP** (*Lightweight Directory Access Protocol*) baze.
  2. **Authentication Server** (AS).
  3. **Ticket Granting Server** (TGS).
- Budući da KDC server sadrži tajne ključeve svih korisnika sistema, posebnu je pažnju potrebno posvetiti njegovoj zaštiti.

6.2 – Kerberus pojmovi

Authentication Server

- Uloga autentifikacijskog servera je da klijentima koji se prijavljuju u Kerberos sistem izda odgovarajuću **TGT**(*Ticket-Granting Ticket*) kartu
- TGT karta generiše se prilikom inicijalne prijave u sistem, nakon čega je klijenti lokalno snimaju i dalje koriste za pristup svim ostalim mrežnim resursima bez potrebe za ponovnim unosom lozinke.
- Postupak izdavanja karte je sledeći:

1. Prilikom inicijalne prijave korisnika u sistem, AS server generiše odgovarajuću TGT kartu
2. Ta karta se šifrira tajnim ključem (lozinkom), koji je poznat samo **KDC serveru** i **krajnjem korisniku** kome se ona izdaje.
3. Ukoliko proces autentikacije uspešno dešifruje dobijenu kartu lozinkom koju je korisnik uneo, proces autentifikacije je uspešan
4. Dobijena karta se čuva lokalno kako bi se kasnije mogla iskoristiti za pristup ostalim mrežnim resursima.

6.2 – Kerberus pojmovi

Ticket Granting Server

- Za razliku od AS servera, koji klijentima generiše inicijalnu TGT kartu prilikom prijave u sistem, TGS server zadužen je za izdavanje dodatnih karata za pristup ostalim mrežnim resursima.
- Za dobijanje odgovarajuće karte za pristup traženom resursu, klijent TGS serveru prosleđuje TGT kartu dobijenu od AS servera te ime resursa kojem želi pristupiti.
- Nakon što TGS proveri da li je dobijena TGT karta ispravna, klijentu se prosleđuje TGS karta kojom je moguće ostvariti pristup traženom mrežnom resursu.

Kerberos karte

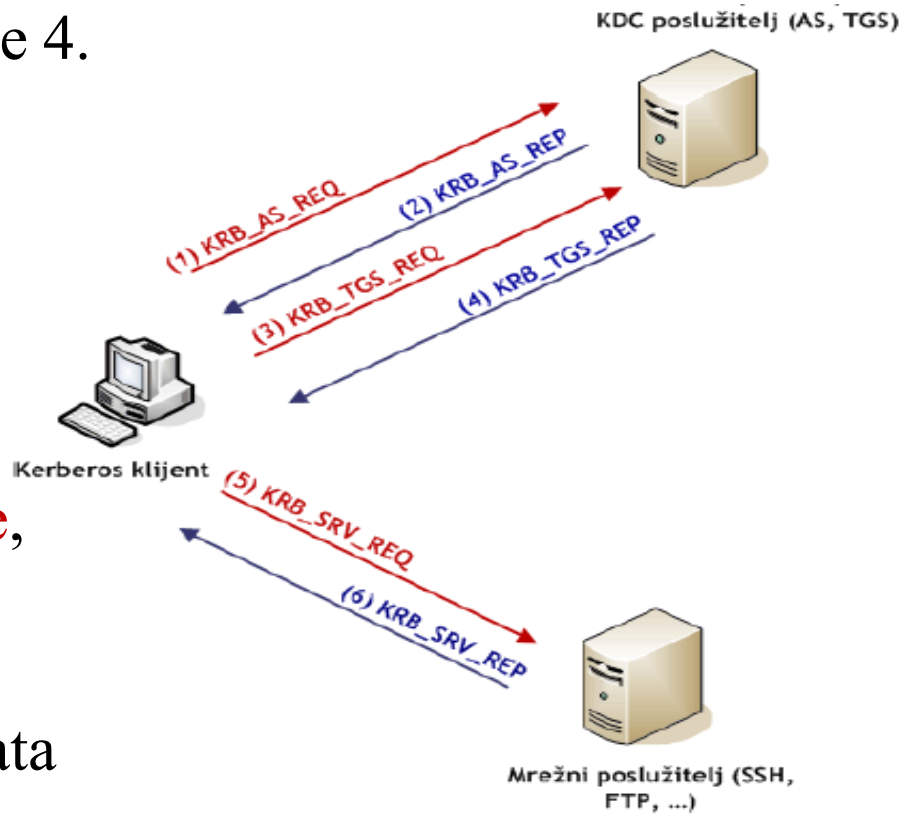
- Koncept karata Kerberos sistema jedinstven je za računarske sisteme.
- Ideja karata vrlo je jednostavna i slična konceptu koji se često primjenjuje u svakodnevnom životu.
- Kerberos karta može se uporediti sa vozačkom dozvolom.
- Centralni autoritet (MUP) izdaje odgovarajuću kartu (vozačku dozvolu) koja sadrži osnovne podatke o korisniku i samoj dozvoli

6.2 – Kerberus pojmovi

- Podaci koje sadrži svaka Kerberos karta su:
  - ✓ime principala koji zahteva pristup,
  - ✓ime principala kojem se zahteva pristup,
  - ✓vremenska oznaka (*timestamp*),
  - ✓vreme trajanja karte (*lifetime*),
  - ✓lista IP adresa s kojih je moguća upotreba karte,
  - ✓tajni ključ sesije za komunikaciju sa traženim resursom.
- Kerberos karte imaju dve osnovne funkcije: da se potvrdi identitet entiteta koji zahteva pristup resursu i da se uspostavi tajni ključ sesije
- Upotrebom vremenske oznake i vremenom trajanja sistem se štiti od tzv. **replay napada** u kojem neovlašćeni korisnik reprodukuje ranije zabeleženi mrežni saobraćaj sa ciljem neovlašćenog pristupa sistemu.
- Svaki zahtev klijenta sadrži vremensku oznaku koju generiše klijentski računar prilikom formiranja zahteva.
- KDC server upoređuje lokalno vreme sa vremenskom oznakom u zahtevu i proverava da li je vremenska razlika u skladu sa max. dozvoljenim odstupanjem(inicijalno 5 min.) - bitna vrem.sinhronizacija

6.2 - Kerberus komunikacija

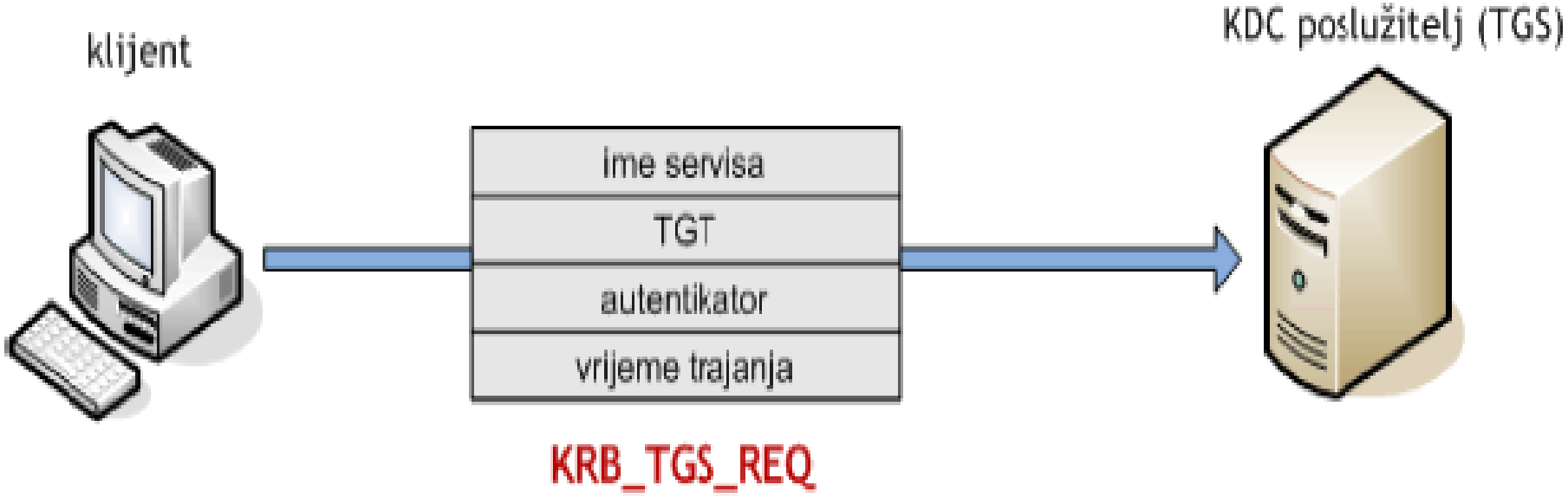
- Kerberos protokol najvećim se delom bazira na Needham-Schreder autentifikacijskom protokolu koji je objavljen još davne 1978. godine.
- Iako su osnovni koncepti vrlo slični, kod Kerberos 4, a nakon toga i Kerberos 5 verzije, dodate su brojne napredne funkcionalnosti koje uklanjaju nedostatke spomenutog Needham-Schreder protokola.
- Opis Kerberos komunikacije vezan je uz Kerberos 5, iako su svi opisani koraci identični i kod verzije 4.
- Razlike između verzija 4 i 5, uglavnom su vezane uz proširenje funkcionalnosti koje dodatno olakšavaju i proširuju mogućnosti primene Kerberos protokola: Forwardable, Proxiable, Renewable, Postdated tickets, korišćenje ASN.1 tehnologije za opis protokola, modifikaciju formata Kerberos karata



6.2 – Kerberus protokol

(1) KRB_AS_REQ zahtev

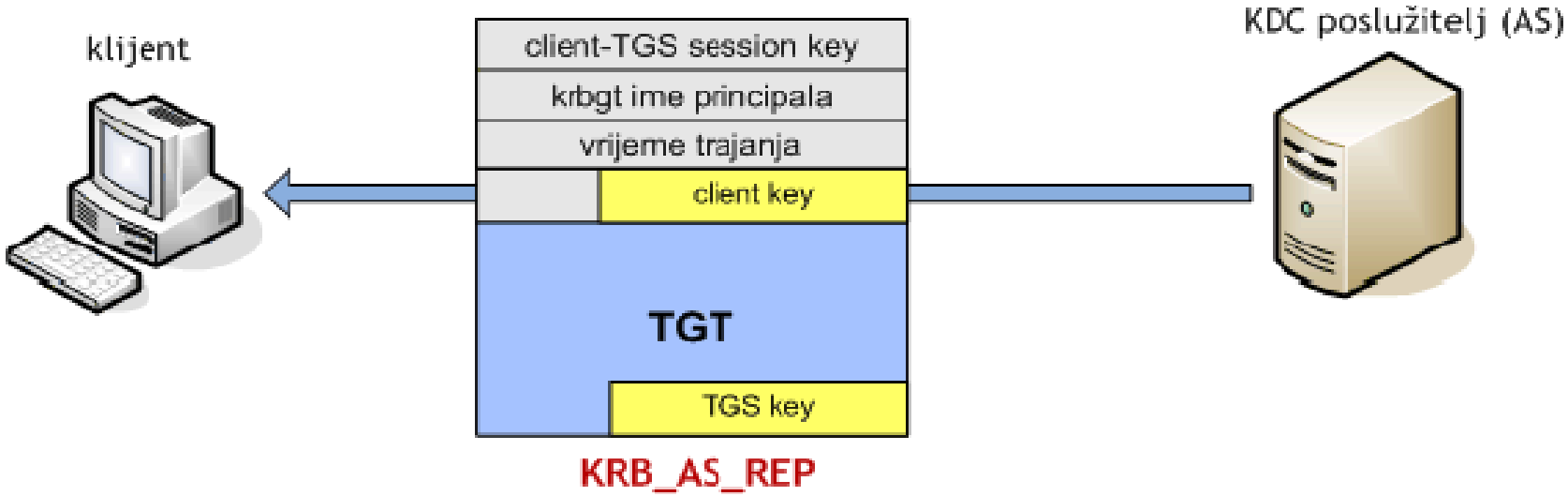
- Postupak autentifikacije korisnik inicira slanjem **KRB_AS_REQ** zahteva KDC (AS) serveru. Ova poruka šalje se u čistom tekstualnom obliku (*plain text*) i sadrži sledeće elemente:
  - ✓ ime principala Kerberos klijenta koji inicira zahtev,
  - ✓ vremensku oznaku – lokalno vreme na strani klijenta,
  - ✓ ime principala TGS servera (krbtgt),
  - ✓ traženo vreme trajanja karte.



6.2 – Kerberus protokol

(2) KRB_AS_REP odgovor

- Nakon primanja zahteva, AS server u lokal.bazi proverava postojanje klijentskog principala i ukoliko isti postoji vraća mu odgovor koji je šifrovan tajnim ključem koji KDC server deli sa istim korisnikom
- Dobijeni odgovor može se dešifrovati samo korisnik koji poseduje odgovarajući tajni ključ, čime se poruka štiti od ***sniffing*** napada
- Osim postojanja klijentskog principala, KDC server takođe proverava i vreme navedeno u dobijenom zahtevu i upoređuje ga sa lokalnim vremenom kako bi se sistem zaštitio od mogućnosti ***replay*** napada.



6.2 – Kerberus protokol

- Odgovor se sastoji se od dva dela:

1. **Prvi deo** šifrovan je tajnim ključem korisnika(*client key*) i sastoji se:
   - ✓ ključ sesije koji će klijent u nastavku komunikacije koristiti za razmenu poruka s TGS serverem (*client-TGS session key*),
   - ✓ ime principala TGS servera (krbtgt),
   - ✓ vreme trajanja karte.

- Dešifrovanjem prvog dela poruke klijent dolazi do ključa sesije koji će koristiti za šifrovanje budućih poruka koje razmenjuje sa TGS serverom (generisanje zahteva za pristup nekom mrežnom resursu)

2. **Drugi deo** poruke sadrži TGT kartu koja je šifrovana tajnim ključem koji KDC server deli s TGS serverem (*TGS key*).

- ✓ To znači da ovaj deo poruke klijent nije u mogućnosti dešifrovati.
- ✓ Šifrovanu TGT kartu klijent će sačuvati u svojoj lokalnoj keš memoriji i iskoristiti je prilikom sledećih zahteva za pristupom ostalim mrežnim resursima u Kerberos sistemu.
- ✓ TGT karta generiše se prilikom inicijalne prijave korisnika u sistem i uz pomoću nje moguće je zatražiti pristup bilo kom mrežnom resursu.

6.2 – Kerberus protokol

- Sve vreme dok je TGT karta aktivna, klijent ne mora unositi korisničku lozinku za pristup ostalim mrežnim resursima unutar Kerberos sistema.
- Nakon što TGT karta istekne, klijent ponovo od AS servera mora zatražiti novu TGT kartu generisanjem novog **KRB_AS_REQ** zahteva.

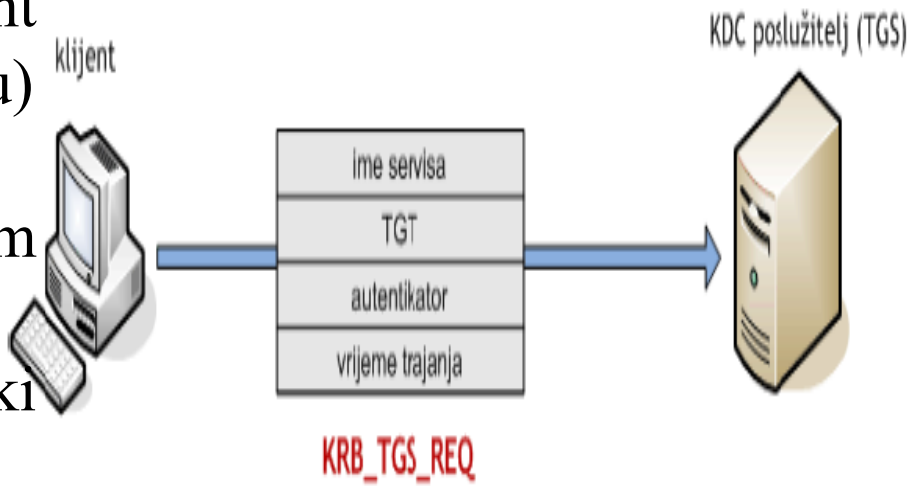
  Sadržaj šifrovane TGT karte je sledeći:
  - ključ sesije koji će klijent koristiti za razmenu poruka s TGS serverem (client-TGS session key),
  - ime principala Kerberos klijenta,
  - vreme trajanja karte,
  - vremensku oznaku KDC servera,
  - IP adresa klijenta (dobijena iz inicijalnog **AS_REQ** zahteva).

6.2 – Kerberus protokol

(3) KRB_TGS_REQ

- Nakon primanja **KRB_AS_REP** poruke, klijent svojim tajnim ključem (lozinkom koju je korisnik uneo) pokušava dešifrovati prvi deo poruke koji sadrži ključ sesije za komunikaciju s TGS serverom.
- Ukoliko je dešifrovanje uspešno, klijent u keš memoriji smešta ključ sesije i dobijenu TGT kartu ali klijent još uvek nema pristup niti jednom mrežnom resursu unutar Kerberos sistema.
- On samo poseduje TGT kartu i odgovarajuću ključ sesije koji će mu omogućiti da od TGS servera zatraži pristup željenom resursu.
- Upravo je to zadatak **KRB_TGS_REQ** koji se sastoji iz **četri dela**:

1. ime principala resursa kojem klijent želi pristupiti (SSH servis na serveru)
2. traženo vreme trajanja karte,
3. TGT karte snimljene u prethodnom koraku,
4. Autentifikatora-osigurava da je svaki zahtev za pristup resursu jedinstven



6.2 – Kerberus protokol

(4) KRB_TGS_REP

- Kao i kod drugog koraka, pri primanju zahteva klijenta KDC server formira odgovor koji će sadržati novi ključ sesije (*client-service session key*), koji će klijent koristiti za razmenu poruka sa resursom (serverom) kome se zahteva pristup.
- Format ovog odgovora identičan je onome u koraku 2, samo što su vrednosti unutar poruke različite.
- Dok je poruka 2 sadržavala TGT kartu i ključ koji klijent koristi za razmenu poruka sa KDC serverem, sada poruka sadrži ključ sesije za razmenu poruka sa zahtevanim resursom (serverom) i TGS kartu za pristup istom resursu i sastoji se od dva dela:

1. Prvi deo šifrovan je ključem sesije dogovorenim u koracima 1 i 2 između klijenta i KDC (AS) servera, i sastoji se od:
   - ✓ime principala resursa kojem klijent želi pristupiti
   - ✓vreme trajanja karte,
   - ✓ključ sesije za razmenu poruka sa resursom kome se zahteva pristup (*client – service session key*).

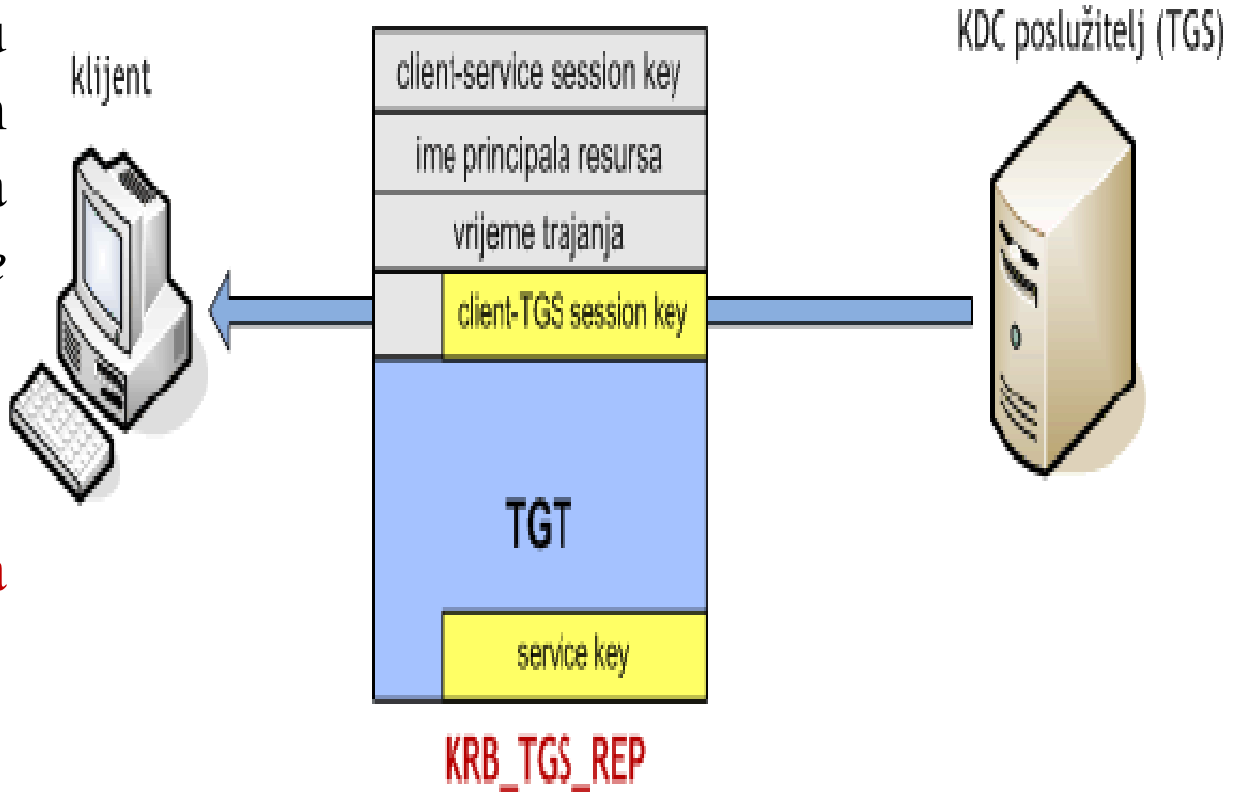
6.2 – Kerberus protokol

- Ovaj deo poruke mogu dešifrovati samo KDC (AS) server i klijent, budući da su oni jedini koji poznaju ključ dogovoren u koracima 1 i 2.

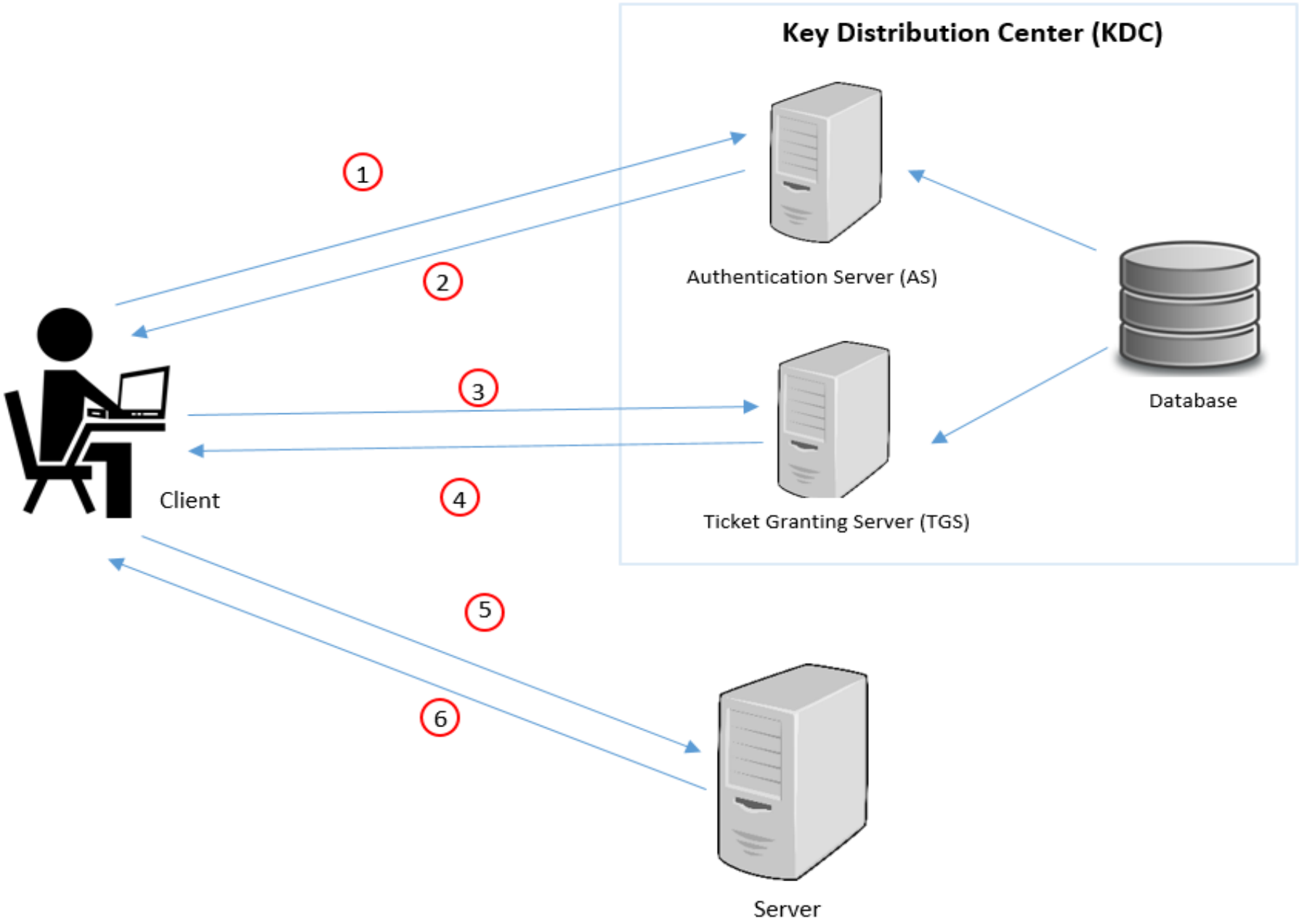
2. **Drugi deo** poruke je TGS karta za pristup traženom resursu.

- Slično kao i TGT karta, ova je karta šifrovana tajnim ključem koji dele KDC server i resurs (server) (*service key*) kojem je zatražen pristup.
- TGS karta sadrži sledeće elemente:

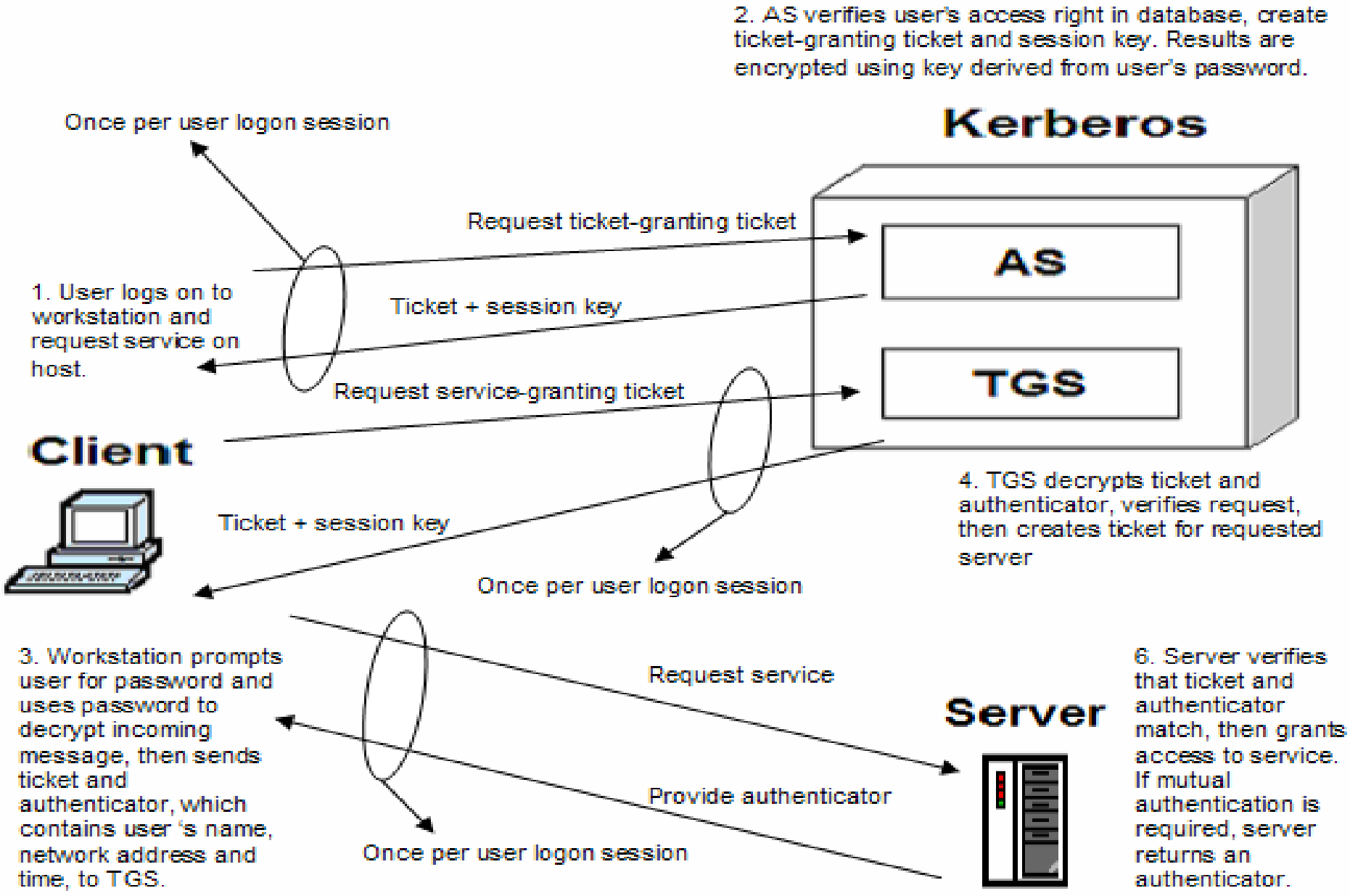
✓ ključ sesije za razmenu poruka sa resursom kojem se zahteva presto (*client–service session key*),
✓ ime principala klijenta,
✓ vreme trajanja karte,
✓ vremenska oznaka KDC servera,
✓ IP adresa klijenta.



6.2 - Kerberus protokol



6.2 - Kerberus protokol



6.3 – Radius protokol

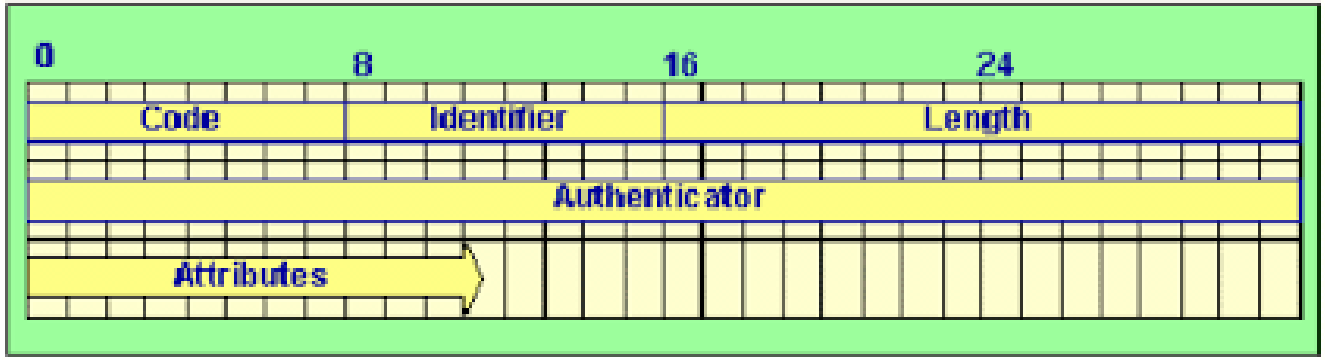
- Da bi se omogućila nesmetana i pouzdana komunikacija putem komunikacionih kanala mora se obezbediti sigurnost mrežnog pristupa
- Organizacije moraju da dodele i obezbede različite nivoe mrežnog pristupa na osnovu toga ko je korisnik, koje podatke za identifikaciju koriste, kako se konektuju, koji nivo enkripcije (šifrovanja) koriste u konekciji, vreme konektovanja, raspoložive resurse i itd.
- Sve ovo zahteva centralizovanu autentifikaciju mrežnog pristupa i sistem upravljanja polisama (Policy Management System) koji može da podrži ove složene zahteve jedne velike mreže kao što je Internet.
- ***Remote Authentication Dial-in User Servis*** (**RAIDUS**) je postao standard za odrađivanje autentifikacije i upravljanja polisama za mrežni prustp (***Network Access Policy***).
- To je rasprostranjen protokol koji se zasniva na Client-Server modelu.
- Takav model omogućava centralnu autentifikaciju i autorizaciju.
- U početku RADIUS protokol je razvijen da reši **Dial-in konekcije**
- Danas se razvio i postao standard za upravljanje mrežnim pristupom kod **VPN, Dial-up i bežičnih mreža**.

6.3 – Radius protokol

- Danas postoje razni komercijalni i open-source RADIUS serveri.
- Najčešću primenu nalazi kod mrežnih uređaja rutera, svičeva, modema
- Protokol na transportnom sloju koristi UDP protokol.
- Na strani klijenta koristi se NAS(Network Access Server), koji obavlja zadatke vezane za prosleđivanje korisničkih parametara RADIUS
- Sa druge strane se nalaze RADIUS serveri (najčešće pozadinski, ***daemon*** program) koji su zaduženi za primanje upita, proveru korisničkih parametara i zatim vraćanja potrebnih konfiguracijskih parametara koji će omogućiti pružanje adekvatne usluge klijentu.
- Protokol RADIUS se koristi iz više razloga:
  - ✓ mrežni uređaji u osnovi ne poseduju mogućnost čuvanja velikog broja autentifikacijskih parametara različitih korisnika s obzirom na ograničene resurse kojima raspolažu,
  - ✓ olakšava i centralizuje tarifiranje korisnika,
  - ✓ pruža određeni nivo zaštite protiv aktivnih napada neovlašćenih korisnika,
  - ✓ ima veliku podršku različitih proizvođača mrežne opreme.

6.3 – Radius format paketa

- **Code**: ovo polje je dužine jednog okteta i opisuje tip RADIUS paketa. Kada se paket primi sa pogrešnim Code poljem, odbacuje se.
- **Identifikator**: dužine jednog okteta i pomaže u slaganju upita i odziva.
- **Dužina**: zauzima 2 okteta i označava dužinu paketa, uključujući Code, Identifier, Length, Authenticator i Attributes. Minimalna dužina je 20, a maksimalna 4096 okteta (paketi izvan ovih granica se odbacuju)
- **Autentifikator**: polje je dužine 16 okteta. Najvažniji oktet se prenosi prvi. Ova vrednost se koristi da utvrdi verodostojnost poruka između klijenta i RADIUS tarifnog servera.
- **Atributi**: konkretni podaci koji se prenose; mogu imati više slučajeva i tada red atributa istog tipa treba biti sačuvan. Ne zahteva se da red atributa različitog tipa bude sačuvan.



6.3 - Radius proces konekcije

1. Korisnik se konektuje na **Network Access Server – NAS** (računar koji je konfigurisan *Routing and Remote Access servisom*) koristeći VPN, Dial-up ili Wireless konekciju.
2. **NAS** prosleđuje zahteve za autentifikacijom ka RADIUS(IAS)serveru. NAS se ponaša kao RADIUS klijent. Ako je provera digitalnog potpisa uspešna, IAS server će proslediti upit kontroleru domena (AD).
3. RADIUS (IAS) server pristupa informacijama iz korisničkog naloga koji se nalazi u **Active Directory** i proverava podatke koje je udaljeni klijent dao prilikom zahteva konekcije za udaljeni pristup.
4. Ako su podaci koji dokazuju njegov identitet autentifikovani, IAS server procenjuje pokušaj konekcije tako što upoređuje polise za udaljeni pristup sa Dial-in Properties informacijama u korisničkom nalogu da bi mogao da donese odluku da li treba da autorizuje zahtev.

6.3 - Radius proces konekcije

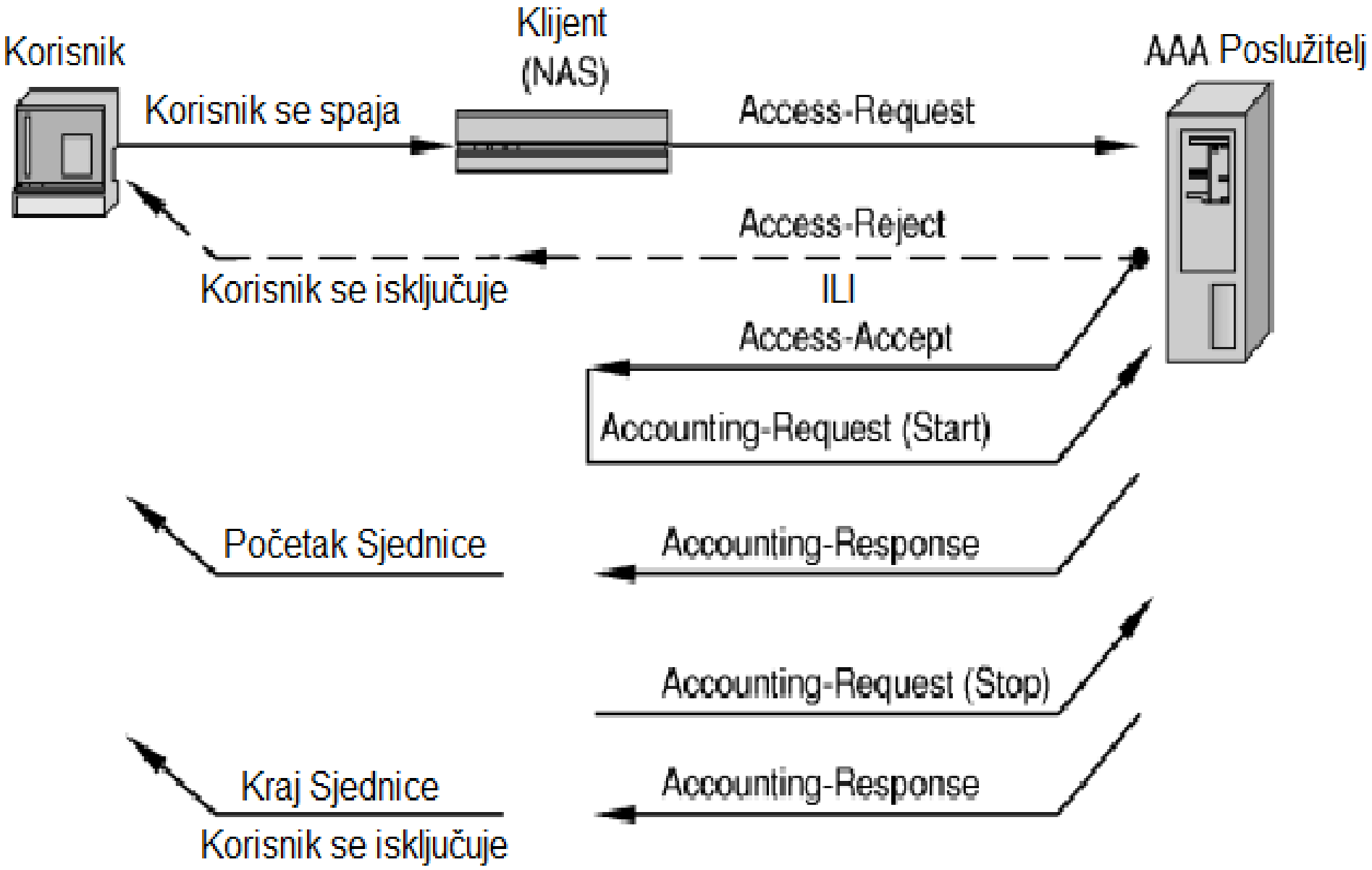
5. Ako pokušaj konekcije ispunjava uslove barem jedne polise za udaljeni pristup i u Dial-in Properties-u korisničkog naloga, IAS šalje **RADIUS Access-Accept** poruku nazad ka NAS koji je poslao zahtev
6. Ako pokušaj konekcije nije prošao proces autorizacije i autentifikacije, IAS šalje **RADIUS Access-Reject** poruku nazad ka NAS i pokušaj konekcije će biti odbijen.

6.3 – Radius komunikacija

- Prilikom prijave u mrežu, korisnik šalje svoje podatke RADIUS klijentu koji potom razmenjuje RADIUS poruke specifičnog formata sa RADIUS serverom. Svrha tih poruka je ostvarivanje AAA funkcija.
- **Kompletna komunikacija** izgleda ovako:

1. Korisnik šalje svoje identifikacione podatke RADIUS klijentu sa željom da mu se odobri pristup određenim mrežnim resursima,
2. Klijent izvršava proces utvrđivanja verodostojnosti i autorizacije razmenom poruka sa RADIUS serverom:
   a. klijent šalje **Access-Request** upit,
   b. server odgovara **Access-Reject** odzivom (u ovom slučaju se korisnikov upit za pristup odbacuje) ili **Access-Accept** odzivom,
3. Klijent izvršava proces tarifiranja korisnika:
   a. klijent šalje serveru poruku **Accounting-Request** (Start),
   b. server odgovara sa **Accounting-Response**, čime počinje sesija,
   c. kad korisnik želi završiti sesiju, klijent šalje serveru **Accounting-Request** (Stop),
   d. server odgovara sa **Accounting-Response**, čime se završava sesija i korisnik se isključuje iz mreže.

6.3 - Radius komunikacija



Hvala na pažnji !!!

Pitanja

? ? ?