SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

SEMINARSKI RAD IZ PREDMETA:
SUSTAVI ZA PRAĆENJE I VOĐENJE PROCESA

KONTROLA UDALJENOG PRISTUPA (RADIUS, KERBEROS, TACACS)

Ivan Pudar
JMBAG: 0036388498
INE

Zagreb, 2.6.2006.

Sadržaj

1. Sažetak

U ovom radu opisana su tri protokola: RADIUS, Kerberos i TACACS. U radu se ne ulazi duboko u samu srž protokola već samo u njihov osnovni način rada da bi čitatelji mogli shvatiti kako funkcioniraju. Opisane su razlike među protokolima te njihove prednosti i nedostaci**.**

2. RADIUS

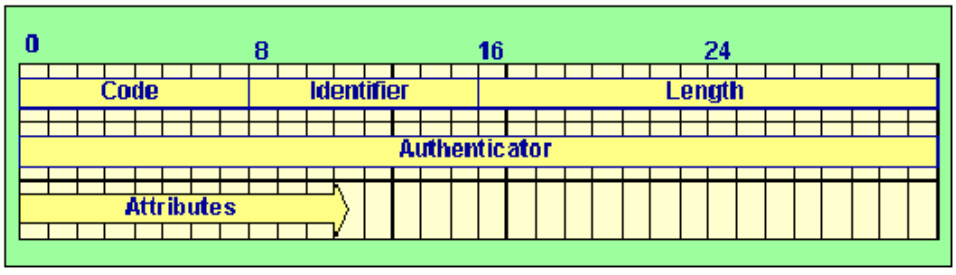
2.1. Uvod

RADIUS (Remote authentication dial-in user service) je AAA (authentication, authorization and accounting) protokol koji je danas vrlo često korišten. RADIUS je prvotno razvijen od Livingston Enterprises za njihove potrebe, ali kasnije (1997.) je objavljen kao RFC 2058 i RFC 2059 (trenutne verzije su RFC 2865 [2] i RFC 2866). Danas postoje razni komercijalni i open-source RADIUS serveri.
U primjeni je najrašireniji kod ugrađenih mrežnih uređaja poput usmjerivača, preklopnika, modema i sl. Protokol se bazira na klijent-poslužitelj modelu (eng. client-server), koji kao transportno sredstvo koristi UDP mrežni protokol. Na strani klijenta koristi se Network Access Server (NAS) programski paket, koji obavlja zadaće vezane za prosljeđivanja korisničkih parametara RADIUS poslužitelju, te obradu primljenih odgovora. S druge strane, RADIUS poslužitelji zaduženi su za prihvaćanje upita, provjeru primljenih korisničkih parametara, te vraćanja potrebnih konfiguracijskih parametara, koji će klijentu omogućiti pružanje adekvatne usluge korisniku.

RADIUS protokol se koristi iz više razloga: mrežni uređaji u osnovi ne posjeduju mogućnost pohranjivanja velikog broja autentikacijskih parametara različitih korisnika, s obzirom na ograničene resurse s kojima raspolažu, olakšava i centralizira administraciju korisnika, pruža određeni nivo zaštite protiv aktivnih napada neovlaštenih korisnika, velika podrška različitih proizvođača mrežne opreme. RADIUS se, iz navedenih razloga, danas smatra standardom za udaljenu autentikaciju korisnika.

2.2. Protokol

Podaci se između klijenta i poslužitelja razmjenjuju putem RADIUS podatkovnih paketa (*Slika 1*), inkapsuliranih unutar UDP paketa protokola niže razine.



Slika 1: Format RADIUS podatkovnog paketa

Značenje pojedinih polja:

-Code- jedan bajt veliko polje, koje definira tip RADIUS podatkovnog paketa. Moguće vrijednosti su dane u *Tablici 1.*

Vrijednost	Opis
00000001	Access-Request
00000010	Access-Accept
00000011	Access-Reject
00000100	Accounting-Request
00000101	Accounting-Response
00001011	Access-Challenge
00001100	Status-Server (experimental)
00001101	Status-Client (experimental)
11111111	Reserved

Tablica 1.

-Identifier (ID)- jedan bajt veliko polje, koje klijentu omogućuje jednoznačnu identifikaciju parova upit-odgovor.

-Lenght-veličina paketa (2 bajta).

-Authenticator-vrijednost koja se koristi za provjeru ispravnosti odgovora od strane RADIUS poslužitelja.

-Attributes-obavezni User-Name i User-Password atributi, a ostali su proizvoljni.

2.2.1. Postupak autentikacije

Korisnik započinje sesiju kreiranjem RADIUS upita sa postavljenim Access-Request kodom koji mora minimalno sadržavati User-Name i User-Password atribute. ID polje korisnik odabire proizvoljno. Unutar Authenticator polja paket sadrži RA (Request Authenticator) vrijednost – 16 bajtni znakovni niz. Pomoću RA i tajnog ključa kriptira se korisnička zaporka (User-Password) što ima presudan značaj za sigurnost protokola.

Nakon primljenog RADIUS Access-Request upita, poslužitelj provjerava da li za tog korisnika postoji tajni ključ koji oni dijele. Ukoliko postoji, slijedi postupak provjere ispravnosti dobivene zaporke koji će, ukoliko je regularan, vratiti Access-Accept paket korisniku. Ako je zaporka neispravna, vraća se Access-Reject paket. Vrijednost ID bajta je identična korisnikovom upitu – na taj način korisnik identificira regularnost odgovora. Ukoliko je korisniku vraćen Access-Accept paket sa valjanim sadržajem, korisničko ime i zaporka smatraju se regularnim i uspješno je izvršena autentikacija korisnika.

2.3. Problemi

RADIUS protokol sadrži sigurnosne propuste koji su posljedica propusta u implementaciji samog protokola ili neispravne i nepotpune implementacije programske podrške.

Neki od propusta su:

-Tajni ključ- Promatranjem algoritma koji se koristi za generiranje odgovora klijentu zaključuje se da postoji samo jedna nepoznanica u cijelom nizu

jednadžbi: upravo tajni ključ! Korištenjem snažnih algoritama za razbijanje, postoje velike šanse za probijanje vrijednosti tajnog ključa.

-Enkripcija User-Password atributa-algoritam koji se koristi za enkripciju korisničke zaporke je MD5 hash funkcija (stream chiper algoritam) koja se u slučaju RADIUS protokola smatra neprikladno odabranim alatom.

-Kao posljedica korištenja gore navedenog algoritma za kriptiranje korisničke zaporke, napadaču se i u ovom slučaju pruža **mogućnost odgonetanja tajnog ključa** kao i uspješno **nagađanje korisničke zaporke.**

-Kompletna sigurnost RADIUS protokola u osnovi ovisi o kvaliteti algoritma za generiranje Request Authenticator atributa. U svrhu uspješnog i sigurnog funkcioniranja protokola spomenuti atribut mora biti jedinstven i nepredvidljiv. Kvaliteta i sigurnost gotovo kod svih algoritama za kriptiranje ponajviše ovisi o mehanizmu za generiranje slučajnih brojeva. Određeni broj implementacija koristi loše i površne mehanizme za generiranje slučajnih brojeva koji se koristi u svrhu generiranja vrijednosti Request Authenticator atributa. Što više taj mehanizam posjeduje deterministička svojstva, to je algoritam enkripcije lakše provaliti. Stoga, pasivnim promatranjem mrežnog prometa između RADIUS klijenta i poslužitelja, napadač može kreirati neku vrstu RADIUS rječnika sa Request Authenticator atributima i njima odgovarajućim User-Password atributima. Neovlašteni korisnik također može praćenjem i analizom prometa kreirati rječnik sa odgovarajućim ID vrijednostima.

2.4. Zaključak

Iz iznesenog razmatranja može se zaključiti da korištenje RADIUS protokola povlači nekoliko sigurnosnih pitanja, koja su posljedica samog dizajna i implementacije protokola. Protokol se može poboljšati odabirom novog dobro poznatog simetričnog *block chiper* algoritma za enkripciju korisničke zaporke, uvođenjem novog User-Password atributa sa alternativnim algoritmom za enkripciju (npr. TDES), pažljivim odabirom ključa za enkripciju korisničke zaporke, nezavisno od ključa. Opcija je da se kao ključ za enkripciju koristi kombinacija tajnog ključa i Request Authenticator atributa.

Sama specifikacija RADIUS protokola trebala bi zahtijevati korištenje snažnijih i moćnijih generatora slučajnih brojeva, u svrhu generiranja Request Authenticator atributa (npr. ANSI X9.17 specifikacija generatora slučajnih brojeva). Za svakog korisnika preporučuje se korištenje drugačijeg tajno ključa, u obliku slučajnog znakovnog niza minimalne duljine 16 znakova, kojeg bi također generirao kvalitetan generator slučajnih brojeva.

Uvijek postoji mogućnost unošenja manjih promjena i poboljšanja, koja bi unaprijedila ovaj protokol, a ujedno bi se zadržala kompatibilnost sa starijim inačicama.

Trenutno ne postoji takav protokol koji bi zamijenio RADIUS, a ujedno bio savršen i koji bi riješio sve ranije spomenute probleme. Eventualni kandidat koji bi u skorije vrijeme mogao predstavljati rješenje je Diameter protokol. Najveći dio poboljšanja usmjeren je ka uklanjanju propusta u specifikaciji samog protokola, te funkcionalnih rješenja koje on nameće. No, malo je toga napravljeno u smjeru poboljšanja sigurnosti same komunikacije između korisnika i poslužitelja. To se kod

Diameter protokola ostavlja drugim mrežnim sigurnosnim protokolima. Upravo zbog takvih ograničenja, RADIUS će još neko vrijeme biti najšire korišten protokol.

3. Kerberos

3.1. Uvod

Kerberos je autentikacijski protokol koji osigurava individualnu komunikaciju preko neosigurane veze u cilju dokazivanja svog identiteta. Osigurava integritet podataka i sprječava osluškivanje i presretanje podataka. Prvenstveno je zamišljen kao client-server model i osigurava uzajamnu autentikaciju- i korisnik i poslužitelj provjeravaju jedan drugom identitet.

Kerberos je razvijen na Massachusetts Institute of Technology (MIT) za zaštitu mrežnih servisa na projektu Athena. Prve tri verzije koriste se interno na MIT-u, a tek četvrtu objavljuju Steve Miller i Clifford Neuman u kasnim 80-tim. 1993. godine objavljuju verziju 5 kao RFC 1510. Verzija iz 2005. je RFC 4120[6].

Windows 2000, Windows XP i Windows Server 2003 koriste varijantu Kerberosa kao autentikacijsku metodu.

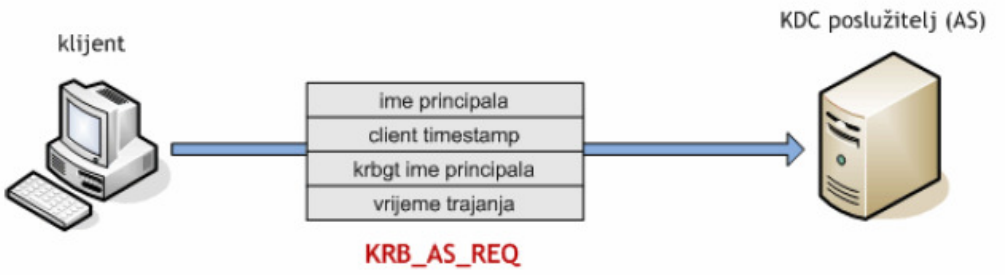
Kerberos se bazira na Needham-Schroeder protokolu. Zahtjeva pouzdanu treću stranu, zvanu Key Distribution Center (KDC) koja se sastoji od dva logički rastavljena dijela: Authentication Server (AS) i Ticket Granting Server (TGS). Kerberos radi na osnovi «karata» koje dokazuju identitet korisnika. Protokol uređuje bazu tajnih ključeva. Svatko na mreži - bilo korisnik ili poslužitelj - ima svoj tajni ključ.

3.2. Protokol

Osnovna načela Kerberos protokola, razložena prema pojedinim fazama komunikacije su:

- **Faza 1**: KRB_AS_REQ zahtjev
  Zahtjev korisnika za autentikacijom prvi je korak u postupku Kerberos autentikacije. Postupak autentikacije korisnik inicira slanjem KRB_AS_REQ zahtjeva KDC (AS) poslužitelju. Ova poruka šalje se u čistom tekstualnom obliku i sadrži sljedeće elemente:
  - ime principala Kerberos klijenta koji inicira zahtjev,
  - vremensku oznaku (eng. timestamp) – lokalno vrijeme na strani klijenta,
  - ime principala TGS poslužitelja,
  - traženo vrijeme trajanja karte.

  Izgled KRB_AS_REQ zahtjeva klijenta prikazan je na *slici 2*.

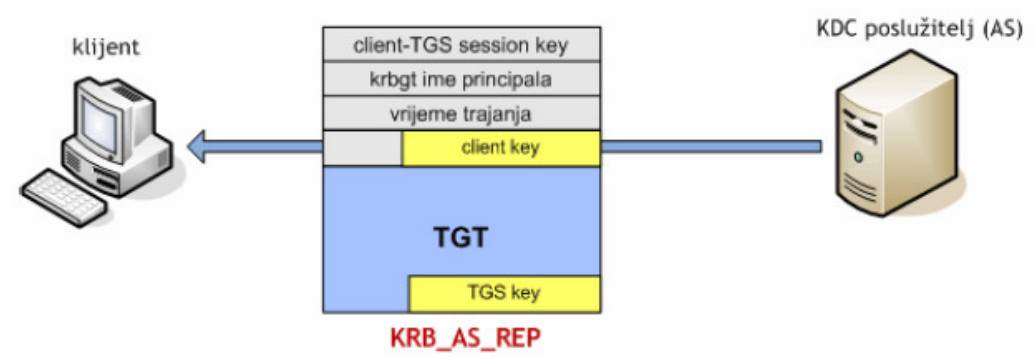


Slika 2: KRB_AS_REQ zahtjev

- **Faza 2:** KRB_AS_REP odgovor
  Pri primanju zahtjeva klijenta, AS poslužitelj u lokalnoj bazi provjerava postojanje navedenog klijentskog principala i ukoliko isti postoji vraća mu odgovor koji je kriptiran tajnim ključem koji KDC poslužitelj dijeli s istim korisnikom. Na ovaj način dobiveni odgovor može dekriptirati samo korisnik koji posjeduje odgovarajući tajni ključ, čime se poruka štiti od sniffing napada. KRB_AS_REP odgovor sastoji se od dva dijela. Prvi dio kriptiran je tajnim ključem korisnika (client key) i sadrži sljedeće elemente:
  - sjednički ključ koji će klijent u nastavku komunikacije koristiti za razmjenu poruka s TGS poslužiteljem (client-TGS session key),
  - ime principala TGS poslužitelja,
  - vrijeme trajanja karte.

  Dekriptiranjem prvog dijela poruke klijent dolazi do sjedničkog ključa kojeg će koristiti za enkripciju budućih poruka koje razmjenjuje s TGS poslužiteljem.
  Drugi dio poruke sadrži TGT kartu koja je kriptirana tajnim ključem koji KDC poslužitelj dijeli s TGS poslužiteljem (TGS key). To znači da ovaj dio poruke klijent nije u mogućnosti dekriptirati. Kriptiranu TGT kartu klijent će pohraniti u svoju lokalnu cache memoriju i iskoristit je prilikom sljedećih zahtjeva za pristupom ostalim mrežnim resursima u Kerberos sustavu. Cijelo vrijeme dok je TGT karta valjana, klijent ne mora unositi korisničku zaporku za pristup ostalim mrežnim resursima unutar Kerberos sustava.
  Sadržaj kriptirane TGT karte je sljedeći:
  - sjednički ključ koji će klijent koristiti za razmjenu poruka s TGS poslužiteljem (client-TGS session key),
  - ime principala Kerberos klijenta,
  - vrijeme trajanja karte,
  - vremensku oznaku KDC poslužitelja,
  - IP adresa klijenta (dobivena iz inicijalnog AS_REQ zahtjeva).

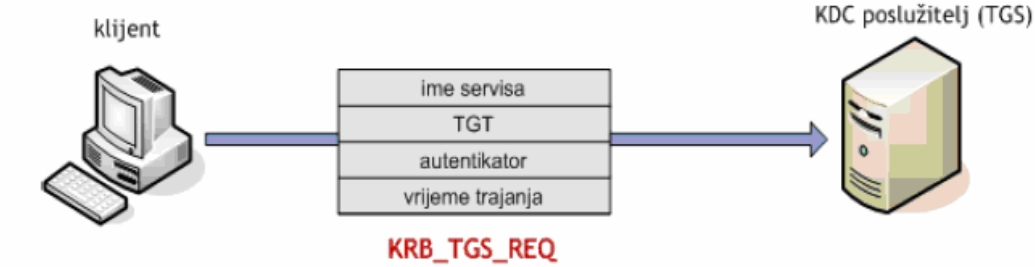
  Struktura opisanog KRB_AS_REP paketa prikazana je na *slici 3.*



Slika 3:KRB_AS_REP odgovor

- **Faza 3**: KRB_TGS_REQ zahtjev
  Nakon primanja KRB_AS_REP poruke, klijent svojim tajnim ključem (zaporkom koju je korisnik unio) pokušava dekriptirati prvi dio poruke koji sadrži sjednički ključ za komunikaciju s TGS poslužiteljem. Ukoliko je dekripcija uspješna, klijent u cache memoriju pohranjuje sjednički ključ i dobivenu TGT kartu. Treba napomenuti da u ovom trenutku klijent još uvijek nema pristup niti jednom mrežnom resursu unutar Kerberos sustava. On samo posjeduje TGT kartu i odgovarajući sjednički ključ koji će mu omogućiti da od TGS poslužitelja zatraži pristup željenom resursu. Upravo je to zadatak KRB_TGS_REQ upita.
  Zahtjev za pristup resursu sastoji se od tri dijela (*Slika 4*):
  - ime principala resursa kojem klijent želi pristupiti (npr. SSH servis na udaljenom poslužitelju),
  - traženo vrijeme trajanja karte,
  - TGT karte pohranjene u prethodnom koraku,
  - autentikatora.

  Autentikator osigurava da je svaki zahtjev za pristup resursu jedinstven i potvrđuje da korisnik posjeduje tajni sjednički ključ dogovoren u ranijim fazama komunikacije.



Slika 4:KRB_TGS_REQ zahtjev

- **Faza 4**: KRB_TGS_REP odgovor
  Slično kao i u drugoj fazi, pri primanju zahtjeva klijenta KDC poslužitelj formira odgovor koji će sadržavati novi sjednički ključ (client-service session key) koji će klijent koristiti za razmjenu poruka sa poslužiteljem. Format ovog odgovora identičan je onome u fazi 2, samo što su vrijednosti unutar poruke različite. Poruka se također sastoji od dva dijela. Prvi dio kriptiran je sjedničkim ključem dogovorenim u fazama 1 i 2 između klijenta i KDC (AS) poslužitelja, i sastoji se od sljedećih elemenata:

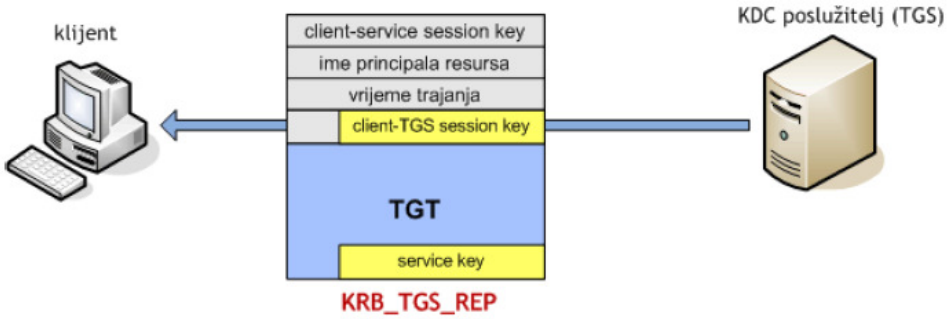
- ime principala resursa kojem klijent želi pristupiti (npr. SSH servis na udaljenom poslužitelju),
- vrijeme trajanja karte,
- sjednički ključ za razmjenu poruka sa resursom kojem se zahtjeva pristup.

Ovaj dio poruke mogu dekriptirati samo KDC (AS) poslužitelj i klijent, budući da su oni jedini koji poznaju ključ dogovoren u fazama 1 i 2.

Drugi dio poruke je TGS karta za pristup zatraženom resursu. Slično kao i TGT karta, ova ja karta kriptirana tajnim ključem koji dijele KDC poslužitelj i resurs (poslužitelj) (eng. service key) kojem je zatražen pristup. TGS karta sadrži sljedeće elemente:

- sjednički ključ za razmjenu poruka sa resursom kojem se zahtjeva pristup,
- ime principala klijenta,
- vrijeme trajanja karte,
- vremenska oznaka KDC poslužitelja,
- IP adresa klijenta.

Struktura opisanog KRB_TGS_REP zahtjeva prikazana je na *Slici 5.*



Slika 5:KRB_TGS_REP odgovor

3.3. Problemi

Najveći problem je u tome što glavni poslužitelj mora stalno biti dostupan, kada je Kerberos server nedostupan nitko se ne može logirati. Karte imaju period raspoloživosti. Promjena zaporke nije standardizirana i razlikuje se od implementacije do implementacije.

3.4. Zaključak

Kerberos protokol naziva se sigurnim zato jer zaporke računalnom mrežom nikad ne šalje u čistom tekstualnom obliku, već u tu svrhu koristi specijalne kriptirane poruke ograničenog perioda valjanosti– karte (tickets). Budući da se autentikacijski parametri mrežom šalju kriptirani, protokol je zaštićen od napada praćenjem i analizom mrežnog prometa (eng. sniffing). Iako je Kerberos prvenstveno autentikacijski protokol, njegovom implementacijom znatno se olakšavaju i ostala dva procesa koja zajedno čine poznati tzv. "AAA" (Authentication, Authorization, Auditing) koncept.

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

4. TACACS

4.1. Uvod

TACACS (Terminal access controller access control system) je udaljeni autentikacijski protokol korišten za komunikaciju sa autentikacijskim serverom uobičajeno korištenim u Unix mrežama. TACACS dozvoljava serveru udaljeni pristup autentikacijskom serveru u cilju određivanja korisničkog pristupa mreži. Protokol je definiran u RFC 1492 dokumentu [8]. Koristi TCP ili UDP vrata 49.

TACACS dozvoljava klijentu korisničko ime i zaporku te da pošalje upit autentikacijskom serveru (zvanom TACACS deamon ili TACACSD). TACACSD odlučuje da li prihvatiti ili odbiti zahtjev i šalje odgovor.

Sljedeća verzija TACACS-a predstavljena je 1990. godine i nazvana XTACACS (extended TACACS). Obje verzije su zamijenjene TACACS+ i RADIUS protokolima. TACACS+ je potpuno novi protokol i kao takav nije kompatibilan sa TACACS i XTACACS protokolima.

4.2. TACACS+

TACACS+ protokol osigurava kontrolu pristupa routerima, mrežni pristup posluživačima i ostalim mrežnim uređajima preko jednog ili više centraliziranih posluživača. Osigurava odvojenu autentikaciju i autorizaciju.

Format zaglavlja dan je na *Slici 6.*

4	8	16	24	32 bit
Major	Minor	Packet type	Sequence no.	Flags
Session ID				
Length				

Slika 6: TACACS+ zaglavlje

Značenja pojedinih polja:

- Major – major broj TACACS+ verzije;
- Minor – minor broj TACACS+ verzije;
- Packet type – moguće vrijednosti su:
  TAC_PLUS_AUTHEN:= 0x01 (Authentication),
  TAC_PLUS_AUTHOR:= 0x02 (Authorization),
  TAC_PLUS_ACCT:= 0x03 (Accounting);
- Sequence number – broj trenutnog paketa u trenutnoj sesiji;
- Flags – ovo polje sadrži različite zastavice koje označavaju da li je paket kriptiran;
- Session ID – identifikacijski broj sesije;
- Length – ukupna duljina tijela TACACS+ paketa (bez zaglavlja).

4.3. Usporedba TACACS+ i RADIUS protokola

- TCP i UDP:

  RADIUS koristi UDP, dok TACACS+ koristi TCP. TCP nudi više prednosti pred UDP-om: connection-oriented transport, potvrda prijema, itd. ali UDP nudi bolju efikasnost.

- Enkripcija paketa:

  RADIUS enkriptira samo zaporku u access-request paketu. Druge informacije, kao korisničko ime i autorizacijski servisi, nisu kriptirani i time su dostupni trećoj strani. TACACS+ enkriptira cijelo tijelo paketa, ali ostavlja TACACS+ zaglavlje u kojem piše da li je paket kriptiran. Stoga TACACS+ nudi bolju razinu sigurnosti.

- Autentikacija i autorizacija:

  RADIUS udružuje autentikaciju i autorizaciju što otežava njihovo moguće razdvajanje. TACACS+ odvaja autentikaciju i autorizaciju što omogućava odvojena rješenja, npr. Moguće je koristiti Kerberos za autentikaciju, a TACACS+ za autorizaciju.

- Multiprotocol Support:

  RADIUS ne podržava neke protokole kao što su AppleTalk Remote Access (ARA) protocol, Net BIOS Frame Protocol Control protocol, Novell Asynchronous Services Interface (NASI), X.25 PAD connection dok TACACS+ nudi njihovu podršku.

- Router Management:

  RADIUS ne daje korisniku kontrolu routera, tj. koje će naredbe biti izvršene, a koje neće. TACACS+ nudi dva načina upravljanja: specificira koje su naredbe dozvoljene i dodjeljuje autorizacijske razine.

5. Reference

[1] http://en.wikipedia.org/wiki/RADIUS
[2] RFC 2865 Remote Authentication Dial In User Service (RADIUS)
[3] www.cert.hr
[4] http://web.mit.edu/kerberos/
[5] http://en.wikipedia.org/wiki/Kerberos_protocol
[6] RFC 4120 Kerberos Protocol
[7] http://en.wikipedia.org/wiki/TACACS
[8] RFC 1492 - An Access Control Protocol, sometimes called TACACS