Implementacija AMRES VPN servisa

Dokument najbolje prakse
(smernice i preporuke)

Izrađen u okviru AMRES tematske grupe za oblast sigurnost
(AMRES BPD 112)

Autor: Jovana Palibrk
Saradnici: Ivan Ivanović, Dušan Pajin

Mart, 2013

Dokument broj: GN3-NA3-T4-AMRES-BDP-112
Verzija / datum: Mart 2013.
Izvorni jezik : Srpski
Originalni naslov: “Implementacija AMRES VPN servisa”
Originalna verzija / datum: Revizija 1 (dokumenta iz februara 2013.)/ 18. mart .2013.
Kontakt: jovana.palibrk@amres.ac.rs, ivan.ivanovic@rcub.bg.ac.rs

AMRES/RCUB snosi odgovornost za sadržaj ovog dokumenta. U izradi dokumenta učestvovala je tematska grupa za oblast sigurnost organizovana u AMRESu radi sprovođenja zajedničkih aktivnosti na razvoju i širenju dokumenata sa tehničkim smernicama i preporukama za mrežne servise u višokoškolskim obrazovnim i istraživačkim ustanovama u Srbiji.

Dokument je nastao kao rezultat istraživanja koja su finansirana sredstvima Sedmog okvirnog programa Evropske zajednice (FP7/2007-2013) po ugovoru br. 238875, koji se odnosi na projekat „Multi-Gigabit European Research and Education Network and Associated Services (GN3)”.

Sadržaj

Executive Summary

VPN (*Virtual Private Network*) technology enables a safe connection between a user at a remote location and his/her home institution through a shared or public network infrastructure (e.g., Internet). The basic requirements to be met by such a solution are:

- data confidentiality – protection from unauthorised access to information;
- data integrity – protection from unauthorised data modification;
- authentication – confirmation of the identity of devices at the ends of the VPN tunnel.

This document describes the deployment of the AMRES VPN service. This solution involves the implementation of the SSL/TSL protocol using OpenVPN technology. The main advantages of an OpenVPN solution are the implementation of advanced data encryption algorithms, the simplicity of installation and maintenance, and the fact that it is supported by almost all of the client and server platforms that are popular today. For user authentication, the AMRES VPN service relies on the RADIUS infrastructure, which was developed for AMRES' *eduroam®*[1] service. The document also provides a detailed configuration of the relevant RADIUS servers on the FreeRADIUS platform.

[1] eduroam is a registered trademark of TERENA, the Trans-European Research and Education Networking Association

Rezime

VPN (*Virtual Private Network*) tehnologija omogućava sigurno povezivanje korisnika na udaljenoj lokaciji sa mrežom njegove matične institucije korišćenjem deljene ili javne mrežne infrastrukture (npr. Internet). Osnovni zahtevi koji se postavljaju pred ovakavim rešenjem su:

- tajnost podataka – zaštita od neovlašćenog pristupa informacijama,
- integritet podataκa – zaštita od neovlašćene izmene podataka,
- autentifikacija korisnika – dokazivanje identiteta uređaja na krajevima VPN tunela.

U dokumentu je opisana realizacija AMRES VPN servisa. Ovo rešenje podrazumeva implementaciju SSL/TLS protokola korišćenjem OpenVPN tehnologije čije su osnovne prednosti korišćenje naprednih algoritama za kriptovanje podataka, jednostavnost prilikom instalacije i održavanja, kao i činjenica da je podržan na gotovo svim danas popularnim platformama. S druge strane, AMRES VPN servis se za realizaciju autentifikacije korisnika oslanja na RADIUS infrastrukturu koja je u AMRES mreži razvijena za potrebe eduroam servisa. U dokumentu je, takođe, data detaljna konfiguracija relevantnih RADIUS servera na FreeRADIUS platformi.

Uvod

U ovom dokumentu je predstavljeno tehničko rešenje i način realizacije infrastrukture za pružanje VPN (*Virtual Private Network*) servisa pojedinačnim korisnicima Akademske mreže Srbije (AMRES), pristup posebno klasifikovanim servisima sa eksternih mreža, pod identičnim uslovima koje imaju kada im pristupaju posredstvom lokalne računarske mreže. Servis je namenjen AMRES korisnicima koji vezu sa Internetom ostvaruju na različite načine - korišćenjem tehnologije ADSL-a, kablovskog pristupa i sl. ili su na putu (drugi Univerzitet, hotel, aerodrom itd.).

U prvom delu dokumenta su navedeni servisi i zahtevi koje servisi postavljaju VPN infrastrukturi. Predstavljeno je VPN rešenje implementirano u AMRES mreži. U drugom delu rada data je detaljna konfiguracija osnovnih komponenti koje čine AMRES VPN servis, VPN i RADIUS servera.

1 Planiranje

1.1 Zahtevi

Prilikom planiranja implementacije VPN servisa prvi korak predstavlja definisanje resursa koji se posredstvom ovog servisa obezbeđuju korisnicima i definisanje zahteva koje implementirana tehnologija treba da ispuni. VPN tehnologija je implementirana u Akademskoj mreži Republike Srbije sa ciljem da se njenim udaljenim korisnicima obezbede određeni AMRES servisi pod identičnim uslovima koje imaju kada pristupaju posredstvom lokalne računarske mreže. U cilju definisanja zahteva koje VPN rešenje treba da zadovolji određen je skup AMRES servisa kojima korisnik može pristupiti upotrebom VPN infrastrukture. U AMRES mreži VPN infrastruktura je realizovana tako da eksplicitno ne isključuje mogućnost korišćenja i dodatnih AMRES servisa, ali je njihova realizacija veoma zavisna od logičke arhitekture računarske mreže organizacije članice AMRES-a. Ovim tehničkim rešenjem se ostavlja prostor AMRES članici da ga prilagodi za pristup udaljenih internih korisnika takvim specifičnim servisima.

AMRES servisi koje korisnici AMRES VPN servisa mogu da koriste sa udaljenih lokacija su:

- KOBSON
- Proxy servis
- eLearning server
- E-mail
- Remote desktop na korisnički računar
- pristup specijalizovanim informacionim sistemima i servisima (npr. univerzitetski IS, fakultetski servisi itd.)
- drugi servisi kojima je pristup omogućen samo unutar AMRES mreže.

Posebno zanimljiv servis je KOBSON koji omogućava internom korisniku pristup bazama naučnih časopisa i publikacija u elektronskom obliku. KOBSON servis karakteriše to što se pravo na njegovo korišćenje ostvaruje implicitno, samo sa akademske mreže korišćenjem jednog od registrovanih *proxy* servera u KOBSON sistemu. Zahvaljujući AMRES VPN servisu korisnici AMRES mreže mogu KOBSON koristiti i kada se nalaze van akademske mreže preko istih registrovanih *proxy* servera.

Za korišćenje AMRES servisa pojedinačnom AMRES korisniku koji pristupa sa eksterne mreže potrebno je obezbediti IP adresu iz opsega adresa koje obezbeđuju autorizaciju za upotrebu tih servisa.

Opšti zahtevi koje AMRES VPN rešenje mora da ispuni su:

- autentifikacija korisnika na osnovu kombinacije korisničkog imena i lozinke,
- autorizacija koja implementira dodelu IP adrese korisniku
- mehanizmi praćenja aktivnosti korisnika (*accounting*).

Danas se uglavnom koriste tehnologije širokopojasnog pristupa (ADSL, kablovski pristup, wireless pristup i sl.). Pristup pomoću ovih tehnologija se realizuje posredstvom provajdera i to, najčešće, znači da bi udaljeni korisnik potencijalnim AMRES servisima trebalo da pristupa preko Interneta, koji predstavlja komunikacionu infrastrukturu koja ne garantuje sigurnost. Zato, sa stanovišta sigurnosti, implementirana VPN tehnologija podrazumeva:

- mehanizme za zaštitu poverljivosti i integriteta informacija potrebnih za autentifikaciju korisnika,
- mehanizme za zaštitu poverljivosti i integriteta podataka koje pojedinačni korisnici razmenjuju sa posebno klasifikovanim servisima.

Tehnologija kojom se realizuje pristup pojedinačnih korisnika sa eksternih mreža ne sme biti zavisna od tehnologije kojom je realizovan pristup korisnika Internetu.

1.2 Arhitektura i izbor tehnologije rešenja

Osnovne komponente VPN servisa su VPN klijent i VPN koncentrator (server). Oni moraju da implementiraju istu VPN tehnologiju. Postoji više tehnologija kojima je moguće realizovati funkcije VPN klijenta i VPN koncentratora. Izabrana tehnologija mora da ispuni minimalne uslove vezane za autentifikaciju, autorizaciju, *accounting* i sigurnost. Osim toga od značaja je i podrška koju data tehnologija ima u operativnom sistemu VPN koncentratora i VPN klijenta. Odabrano rešenje treba da ima podršku za gotovo sve danas popularne operativne sisteme (Windows, Linux, Mac OS, Solaris, FreeBSD, NetBSD i OpenBSD).

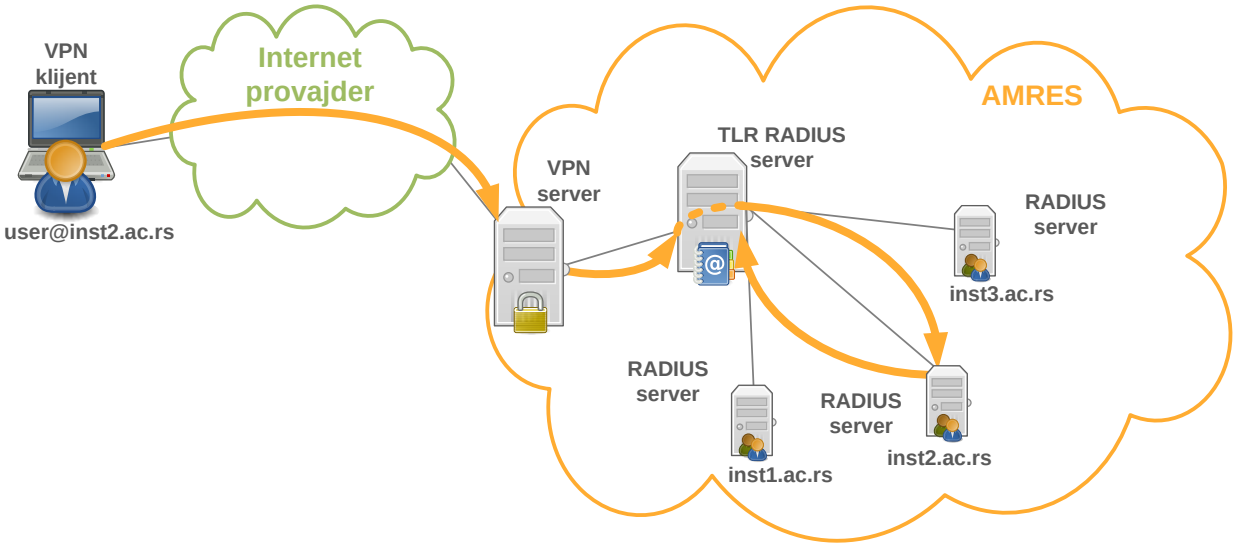
Uzimajući u obzir navedene zahteve, za realizaciju AMRES VPN servisa odabran je OpenVPN programski paket koji između ostalog koristi SSL/TLS protokol. Osnovni principi funkcionisanja i prednosti OpenVPN programa dati su u narednom poglavlju.

Funkcije autentifikacije, autorizacije i čuvanja podataka o aktivnosti korisnika (*accounting*) realizovane su korišćenjem već postojeće hijerarhijske RADIUS infrastrukture koja se koristi u okviru eduroam servisa. U skladu sa tim, RADIUS infrastruktura se sastoji iz dva nivoa:

- Prvi nivo predstavlja VPN TLR (*Top-Level RADIUS*) server koji ima ulogu *proxy* RADIUS servera. On sadrži listu domena AMRES institucija korisnica VPN servisa.
- Drugi nivo predstavljaju RADIUS serveri krajnjih institucija koji su odgovorni za autentifikaciju svojih korisnika. Matične institucije su odgovorne za održavanje podataka i kredencijala svojih korisnika. Oni se obično čuvaju u bazi podataka koju RADIUS server koristi u procesu autentifikacije. S obzirom da se radi o istoj infrastrukturi na koju se oslanja i eduroam servis, na ovaj način se postiže da korisnici mogu

da koriste iste autentifikacione parametre (istu kombinaciju korisničkog imena i lozinke) za oba servisa, i AMRES VPN i eduroam.

Logička arhitektura AMRES VPN servisa je prikazana na slici 1.

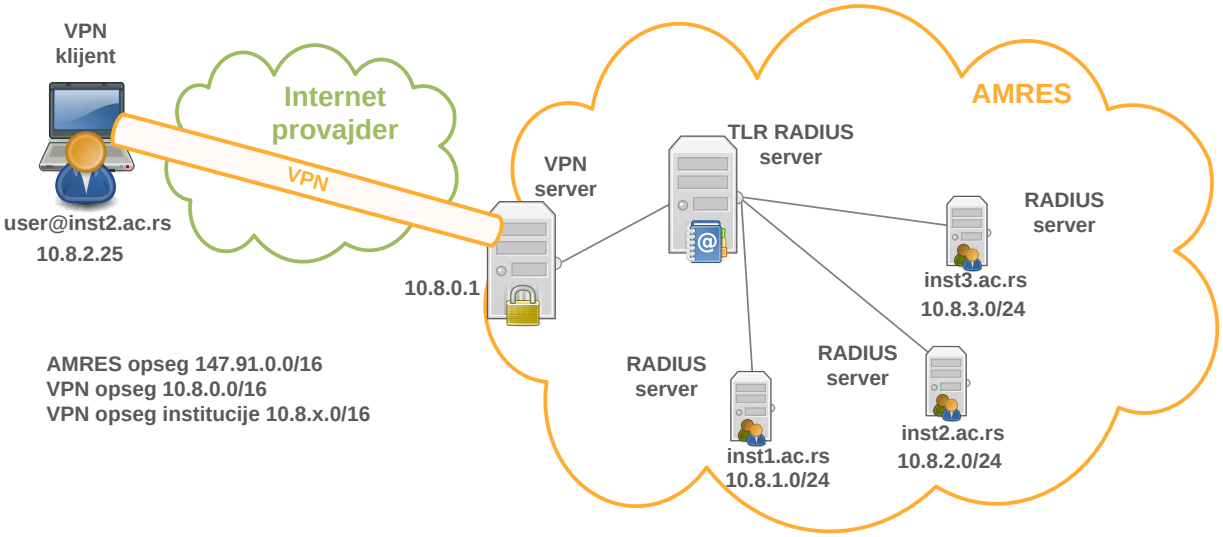


Slika 1: Logička arhitektura AMRES VPN servisa

Implementiran je jedan VPN koncentrator koji se nalazi na centralnoj lokaciji akademske mreže. VPN koncentrator je povezan sa VPN TLR (*Top-Level* RADIUS) serverom. U toku prijave korisnika na AMRES VPN servis i ostvarivanja sigurne konekcije između korisničkog uređaja i VPN koncentratora, vrši se autentifikacija korisnika. Korisnički kredencijali se šalju preko mreže Internet servis provajdera do VPN servera kriptovani. VPN server šalje zahtev za autentifikaciju korisnika TLR serveru. Svaki korisnik prilikom prijavljivanja na VPN servis mora da upotrebi svoje korisničko ime u formi *korisničko-ime@domen-institucije*, gde *domen-institucije* predstavlja DNS (*Domain Name Server*) ime institucije. TLR server koristi informaciju o domenu institucije kako bi zahtev prosledio RADIUS serveru institucije kojoj korisnik pripada. RADIUS server institucije, zatim, obrađuje pristigli zahtev. U toku procesiranja zahteva RADIUS server kontaktira korisničku bazu podataka u kojoj se čuvaju korisnički nalozi.

Ukoliko je korisnik uspešno autentifikovan, TLR server korisničkom uređaju dodeljuje IP adresu iz VPN opsega njegove institucije. Za AMRES VPN opseg odabran je 10.8.0.0/16, dok svaka institucija učesnica AMRES VPN servisa dobija jednu C klasu IP adresa iz opsega 10.8.0.0/16. Dodeljeni opseg može dalje da se podeli na podopsege u okviru jedne institucije, na primer za studente i zaposlene, na osnovu čega institucija može da vrši filtriranje saobraćaja u skladu sa svojom politikom pristupa.

Na slici 2 predstavljena je logička arhitektura AMRES VPN servisa sa detaljima koji se tiču adresiranja. Za potrebe dokumenta i radi njegove jednostavnosti, pretpostavljeno je da AMRES poseduje 147.91.0.0/16 opseg IP adresa.

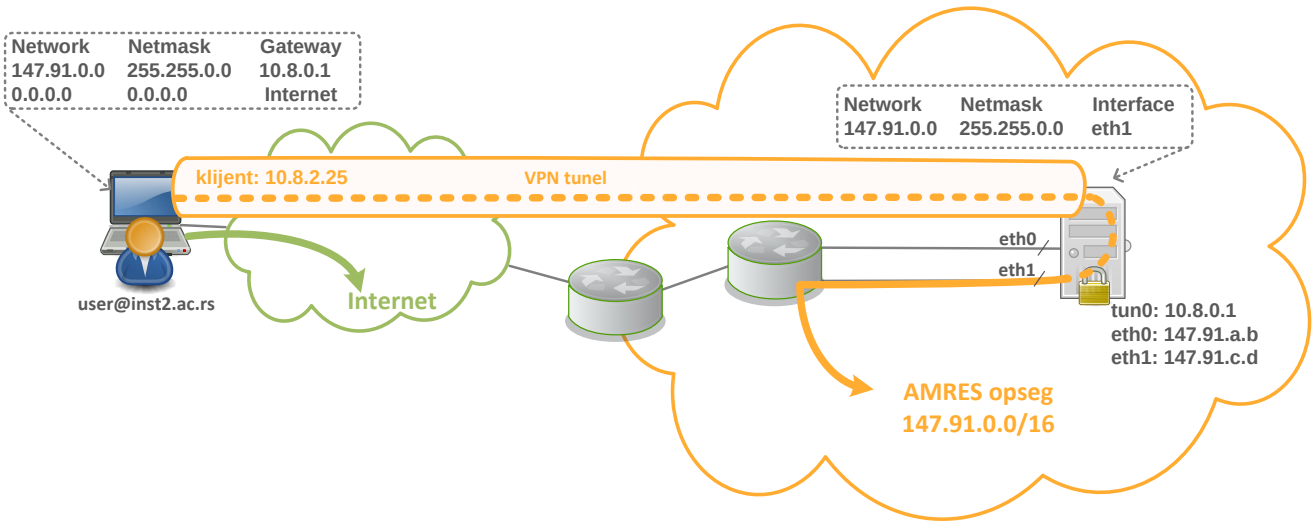


Slika 2: Raspodela IP adresa u okviru AMRES VPN servisa

Aktivnost korisnika VPN servisa (*accounting*) se prati takođe korišćenjem RADIUS servera. VPN server je konfigurisan da informacije o korisničkoj sesiji šalje TLR serveru. TLR server te podatke čuva u odgovarajućoj SQL bazi podataka.

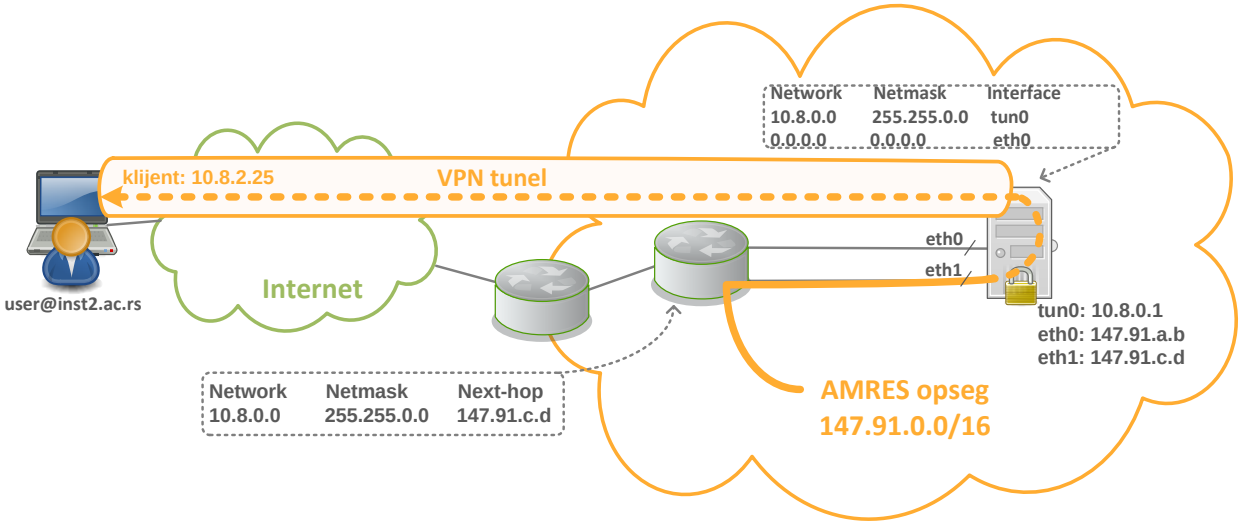
VPN server se nalazi na centralnoj lokaciji AMRES mreže. Na *firewall* sistemu otvoren je UDP port 1194 za IP adresu servera jer je OpenVPN servis konfigurisan da konekcije VPN klijenata prihvata po ovom portu. Kriptografska zaštita saobraćaja je obezbeđena samo između klijentskog uređaja i VPN servera. Kada VPN server primi saobraćaj od korisnika, saobraćaj se dekriptuje i prosleđuje u originalnoj formi do destinacije kroz akademsku mrežu. Ako je saobraćaj klijenta pre nego što je kriptovan poslat u *plain-text* formatu, od VPN servera do destinacije u AMRES mreži će takođe biti prosleđen u *plain-text* formatu.

Na slikama 3 i 4 je ilustrovano kako je rešeno rutiranje u okviru VPN servisa. Prilikom povezivanja korisnika na AMRES mrežu preko VPN servisa između VPN servera i korisničkog uređaja se formira *point-to-point* virtuelni tunel interfejs. Svakom kraju tunela dodeljuje se virtuelna IP adresa. Virtuelna IP adresa servera je 10.8.0.1, a klijent dobija adresu iz opsega koji pripada njegovoj instituciji (jedna C klasa iz opsega 10.8.0.0/16), kao što je prikazano na slici 2. Na slici 3 prikazano je kako saobraćaj od VPN klijenta putuje ka nekom hostu u AMRES mreži. AMRES VPN server ima dva mrežna interfejsa kojima su dodeljene javne IP adrese. Jedan mrežni interfejs služi za povezivanje VPN klijenata sa VPN serverom i formiranje VPN tunela, spoljašnji interfejs (na slici 3 označen kao eth0, IP adresa 147.91.a.b). Na tom mrežnom interfejsu se prima saobraćaj koji šalju VPN klijenti. VPN uređaj zatim, nakon dekripcije i enkapsulacije, saobraćaj ka AMRES mreži rutira preko drugog, unutrašnjeg interfejsa (na slici 3 označen kao eth1, IP adresa 147.91.c.d).



Slika 3: Rutiranje paketa od VPN klijenta ka AMRES mreži

Na slici 4 prikazana je situacija kada se paketi iz AMRES mreže vraćaju VPN klijentu. IP adresa klijenta se koristi kao izvorišna adresa IP paketa koje klijent šalje kroz VPN tunel. Ista adresa se kao izvorišna adresa nalazi i u paketu kada izađe iz SSL tunela i stigne do svoje destinacije. Zbog toga uređaji u AMRES mreži moraju da imaju konfigurisanu rutu do VPN opsega IP adresa, 10.8.0.0/16, kome adrese klijenata pripadaju. Ta ruta pakete za VPN opseg usmerava ka adresi unutrašnjeg interfejsa VPN servera, eth1.



Slika 4: Rutiranje paketa od AMRES mreže ka VPN klijentu

AMRES VPN servis podržava *split tunneling*. *Split tunneling* je proces koji dozvoljava da udaljeni VPN korisnik može istovremeno da pristupi javnoj mreži, najčešće Internetu, kroz svoju lokalnu mrežnu konekciju i resursima svoje organizacije kroz formirani VPN tunel. AMRES VPN servis je realizovan tako da se saobraćaj ka svim AMRES adresnim opsezima od uređaja VPN korisnika šalje kroz VPN tunel, a da sav ostali saobraćaj korisnički uređaj šalje kroz lokalnu mrežnu konekciju, kao što je ilustrovano na slici 3. Na ovaj način se postiže veća efikasnost i očuvanje propusnog opsega, jer je značajno smanjena količina saobraćaja koja se šalje kroz VPN tunel.

Za realizaciju ovakvog rešenja odabrani su *opensource* programski paketi. Za VPN server korišćen je OpenVPN paket. TLR server je realizovan korišćenjem FreeRADIUS programskog paketa. U narednom poglavlju izneti su detalji konfiguracije VPN i TLR servera, kao i primer konfiguracije potrebne za realizaciju RADIUS server institucije.

2 Implementacija

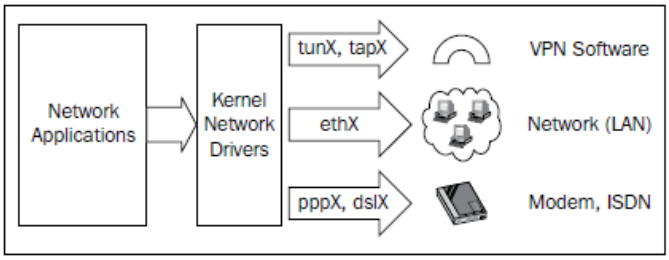
2.1 VPN server

2.1.1 OpenVPN

OpenVPN je *open sorce* programski paket za implementaciju virtuelnih privatnih mreža. U okviru OpenVPN softvera koristi se siguran i stabilan SSL/TLS mehanizam za uspostavu sigurne komunikacije između VPN tačaka. OpenVPN paket je potrebno instalirati i na strani VPN servera i na strani VPN klijenta, za razliku od drugih SSL VPN rešenja kod kojih na strani klijenta može da se koristi Internet pretraživač.

OpenVPN program se pokreće u korisničkom prostoru (*user-space*), a ne u jezgru operativnog sistema (*kernel-space*) što dovodi do povećanja sigurnosti i pojednostavljivanja postupka instalacije i održavanja OpenVPN softvera na različitim platformama.

Na mrežnom sloju OpenVPN koristi TUN/TAP virtuelne mrežne interfejse. Ovi mrežni interfejsi se ponašaju kao standardni mrežni interfejsi, a služe za tunelovanje IP saobraćaja. Na slici 5 je prikazano kako OpenVPN koristi standardne mrežne interfejse.



Slika 5:

TUN interfejs je virtuelni mrežni adapter kojeg operativni sistem „vidi“ kao virtuelni PPP (*Point-to-Point*) mrežni interfejs, kao na primer T1 linija. TUN interfejs simulira uređaj mrežnog sloja OSI referentnog modela (*layer* 3) i preko njega se šalju *layer* 3 paketi kao što su IP paketi. Ovaj mod rada se naziva *routed* mod jer se TUN interfejsi koriste za rutiranje saobraćaja. S druge strane, TAP interfejs se može koristiti kao virtuelni *ethernet* adapter. Njime se simulira uređaj *data link* sloja OSI referentnog modela (*layer* 2) koji šalje i prima *layer* 2 pakete, kao što su *ethernet* frejmovi. Ovaj mod rada se naziva *bridging* mod jer se TAP interfejs koristi kao mrežni *bridge* uređaj. Prilikom slanja podataka preko ovih interfejsa umesto na fizički mrežni adapter, podaci se šalju na TUN/TAP interfejs odakle ih preuzima program u korisničkom prostoru koji je povezan na taj interfejs, u ovom slučaju OpenVPN programu. VPN proces pakete sa TUN ili TAP virtuelnih adaptera kriptuje korišćenjem

SSL/TLS kriptografski biblioteka. Zatim tako kriptovani podatke enkapsulira i šalje na drugi kraj tunela. Za enkapsulaciju paketa može da se koristi UDP (standardno) ili TCP protokol (opciono). Prilikom izbora transportnog protokola u koji će se vršiti enkapsulacija kriptovanih podataka sa TUN/TAP interfejsa treba imati u vidu da se na taj način vrši enkapsulacija paketa koji već sadrži zaglavlje nekog od transportnih protokola (TCP i UDP) u jedan od ovih transportnih protokola. TCP protokol je dizajniran da radi u nepouzdanom mrežnom okruženju, stoga prilikom enkapsulacije TCP u TCP jedan pouzdan protokol se enkapsulira u drugi pouzdan protokol što može da dovede do smanjenje efikasnosti servisa. Zbog toga je bolji izbor enkapsulacija u UDP pakete. Takođe, na ovaj način se postiže da IP paket bude enkapsuliran u protokol čiji stepen nepouzdanosti približniji protokolima *data-link* sloja za koje je IP protokol dizajniran. Kada uređaj na drugom kraju SSL tunela primi kriptovane i enkapsulirane pakete, OpenVPN proces će prvo da izvrši deenkapsulaciju i dekripciju, a zatim da pakete pošalje na TUN/TAP interfejs. Pravila rutiranja i sigurnosna pravila se mogu primenjivati na ove interfejse kao i na bilo koje hardverske mrežne adaptere. U toku OpenVPN konfiguracije moguće je odabrati sve ove parametre prema potrebama (korišćenje TUN ili TAP interfejsa, enkapsulaciju u UDP ili TCP, port preko kog će konekcija biti uspostavljena), ali treba voditi računa da su isti parametri konfigurisani na obe strane tunela.

Sa ovakvim karakteristikama OpenVPN paket nudi sledeće mogućnosti:

- Jednostavna instalacija na bilo kom operativnom sistemu. OpenVPN paket je moguće koristiti na gotovo svim danas popularnim operativnim sistemima (Windows, Linux, Mac OS, Solaris, FreeBSD, NetBSD i OpenBSD).

- Visok nivo fleksibilnosti – programske skripte se mogu menjati kako bi se zadovoljile posebne potrebe (*failover*, *load balancing*).

- Transparentnost – nema potrebe za konfigurisanjem isključivo statičkih IP adresa na obe strane tunela, već se IP adrese mogu dodeljivati i dinamički, a sami korisnici ne moraju biti svesni promene IP adrese prilikom uspostave VPN tunela.

- Zaštita udaljenih korisnika pomoću internog zaštitnog (*firewall)* sistema - udaljeni korisnik koji je konektovan sa centralnom lokacijom svoje mreže VPN tunelom može da menja mrežna podešavanja na svom uređaju tako da se sav saobraćaj šalje kroz tunel. Centralni *firewall* u mreži sa kojom je korisnik povezan VPN tunelom može da zaštiti korisnički uređaj, čak iako se uređaj fizički ne nalazi u mreži koju *firewall* štiti.

- Samo jedan port treba da bude otvoren na *firewall* sistemu mreže kako bi se dolazne konekcije VPN klijenata bile dozvoljene.

- Virtuelni interfejsi dozvoljavaju specifična mrežna i *firewall* podešavanja i pravila – sva pravila, restrikcije, mehanizmi prosleđivanja, koncepti kao što je NAT (*Network Address Translation*) mogu se primenjivati na OpenVPN tunelima.

OpenVPN omogućava autentifikaciju korisnika korišćenjem tajnog ključa, sertifikata ili proverom korisničkog imena i lozinke.

2.1.2 Konfiguracija VPN koncentratora

OpenVPN server se podešava modifikovanjem konfiguracionih fajlova na strani servera i na strani klijenta. Pre podešavanja ovih fajlova potrebno je podesiti elemente PKI (*Public Key Infrastructure*) infrastrukture. Ovi koraci biće detaljno objašnjeni u nastavku. Lokacija konfiguracionog fajla na strani servera zavisi od načina na koji je OpenVPN paket instaliran. U slučaju AMRES VPN servera instaliran je standardni OpenVPN paket uključen u distribuciju operativnog sistema, pa je konfiguracioni direktorijum `/etc/openvpn`, a naziv konfiguracionog fajla o kome će biti reči u nastavku je `server.conf`.

2.1.2.1 *Kreiranje privatnih i javnih ključeva*

Prvi korak pri podešavanju OpenVPN servera jeste konfiguracija neophodnih parametara koji će se koristiti za autentifikaciju, enkripciju i razmenu ključeva. VPN autentifikacija treba da bude uzajamna, odnosno klijenti treba da autentifikuju server i server treba da izvrši autentifikaciju klijenata kako bi se ostvarilo uzajamno poverenje i na taj način obezbedila sigurna konekcija. Kao što je već rečeno, autentifikacija VPN klijenata se ostvaruje kroz RADIUS infrastrukturu. S druge strane, da bi VPN klijent autentifikovao VPN server koriste se elementi PKI infrastrukture. PKI infrastruktura se sastoji od privatnog ključa i sertifikata (sa javnim ključem) na strani servera i CA (*Certificate Authority*) sertifikata i ključa koji se koristi za potpisivanje serverskog sertifikata. Prilikom autentifikacije servera, klijent će prvo da proveri da li je serverski sertifikat potpisan od strane CA tela, a zatim će da proveri verodostojnost informacija kao što je vrednost CN (*Common Name*) polja iz serverskog sertifikata u koje se upisuje DNS ime servera.

Za podešavanje PKI elemenata koristi se skripte koje su dobijene u sklopu OpenVPN programa. Ova skripta se nalaze u `easy-rsa` direktorijumu. Lokacija `easy-rsa` direktorijuma se može pronaći pomoću sledeće komande:

```
$ find / -name easy-rsa
```

U radu je pretpostavljeno da se nalazi u `/usr/share/doc/openvpn` direktorijumu.

Korišćenjem skripti iz `easy-rsa` direktorijuma prvo se kreira CA privatni ključ i sertifikat (koraci 1, 2 i 3). Zatim se kreira privatni ključ i sertifikat za VPN server, a sertifikat servera se potpisuje CA privatnim ključem (korak 4). Dakle, vrši se samopotpisivanje sertifikata VPN servera. VPN klijent treba da poseduje CA sertifikat kako bi mogao da autentifikuje VPN server. Naravno, moguće je za VPN server koristiti i sertifikat potpisan od strane nekog drugog CA tela. Na primer, u ovu svrhu se mogu koristiti TERENA sertifikati koji su potpisani od strane TERENA CA sertifikacionog tela, a dostupni su preko TCS (TERENA Certificate Service) servisa. U tom slučaju koraci 1, 2, 3 i 4 bi se preskočili, a u odgovarajuće lokacije bi se smestili TERENA CA i dobijeni serverski sertifikat.

Koraci neophodni prilikom podešavanja elemenata PKI infrastrukture:

1. `easy-rsa` direktorijum se kopira na `/etc/openvpn` lokaciju

```
$ mkdir -m 700 -p /etc/openvpn/easy-rsa
$ cd /etc/openvpn/easy-rsa
$ cp -drp /usr/share/doc/openvpn/easy-rsa/2.0/* .
```

2. Zati je potrebno podesiti `KEY_COUNTRY`, `KEY_PROVINCE`, `KEY_CITY`, `KEY_ORG`, i `KEY_EMAIL` parametre u `/etc/openvpn/easy-rsa/vars` fajlu. Primer popunjenog `vars` fajla je dat u nastavku.

```
$ vi /etc/openvpn/easy-rsa/vars
```

```
export EASY_RSA="`pwd`"
export OPENSSL="openssl"
export KEY_CONFIG=`$EASY_RSA/whichopensslcnf $EASY_RSA`
export KEY_DIR="$EASY_RSA/keys"
export PKCS11_MODULE_PATH="dummy"
export PKCS11_PIN="dummy"
export KEY_SIZE=2048
export CA_EXPIRE=3650
export KEY_EXPIRE=3650
export KEY_COUNTRY="RS"
export KEY_PROVINCE="RS"
export KEY_CITY="Belgrade"
export KEY_ORG="AMRES"
export KEY_EMAIL=openvpn@example.com
```

Zbog veće sigurnosti, usvojeno je da minimalna dužina para asimetričnih ključeva, koji se generišu pri kreiranju zahteva za sertifikatom, mora biti 2048 bita što je definisano parametrom `KEY_SIZE`.

3. U ovom koraku se kreira CA sertifikat koji će kasnije biti korišćen za potpisivanje sertifikata za VPN server. Komandom `cd` potrebno je locirati se u direktorijumu u kom se nalazi `vars` fajl i setom komandi koje su navedene u nastavku kreirati CA privatni ključ i sertifikat. Pre pokretanja ovih komandi potrebno je proveriti da li postoji `openssl.cnf` fajl u `/etc/openvpn/easy-rsa` direktorijumu. Ukoliko ne postoji kopirati `openssl-x.x.x.cnf` (u primeru `openssl-1.0.0.cnf`) u `openssl.cnf`. Za CA sertifikat u okviru parametra `--pass` potrebno je odabrati snažnu lozinku.

```
$ cd /etc/openvpn/easy-rsa
$ cp openssl-1.0.0.cnf openssl.cnf
$ . ./vars
$ ./clean-all
$ ./build-ca --pass
```

Komanda `./build-ca` kreira CA setifikat i ključ pomoću openssl komande. Svi parametri osim CN imena biće preuzeti iz `vars` fajla. CN ime je potrebno naknadno uneti. Rezultat pokretanja ove komande je prikazan u nastavku.

Napomena: rezultat pokretanja komande se može razlikovati od prikazanog primera u zavisnosti od verzije OpenVPN programskog paketa.

```
root@openvpn:/etc/openvpn/easy-rsa# ./build-ca --pass
Generating a 2048 bit RSA private key
................++++++
..........................++++++
writing new private key to 'ca.key'
Enter PEM pass phrase:
```

```
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [RS]:
State or Province Name (full name) [RS]:
Locality Name (eg, city) [Beograd]:
Organization Name (eg, company) [AMRES]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) [AMRES CA]:
Name [changeme]: openvpn
Email Address [openvpn@example.com]:
```

4. Sledeći korak predstavlja generisanje serverskog sertifikata i privatnog ključa. Serverski sertifikat će biti potpisan CA privatnim ključem koji je generisan u prethodnom koraku. Za to se koristi skripta `build-key-server`. Prilikom pokretanja ove skripte potrebno je navesti ime servera (*hostname*) koji će biti upisan u CN polje generisanog sertifikata. U nastavku je dat rezultat pokretanja ove skripte. Kada se u okviru skripte zatraži `ca.key` lozinka potrebno je uneti lozinku za CA sertifikat koja korišćena u prethodnom koraku.

   Napomena: rezultat pokretanja komande se može razlikovati od prikazanog primera u zavisnosti od verzije OpenVPN programskog paketa.

```
root@openvpn:/etc/openvpn/easy-rsa# ./build-key-server openvpn
Generating a 2048 bit RSA private key
......++++++
......++++++
writing new private key to 'openvpn.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [RS]:
State or Province Name (full name) [RS]:
Locality Name (eg, city) [Beograd]:
Organization Name (eg, company) [AMRES]:
Organizational Unit Name (eg, section) []:
Common Name (eg, your name or your server's hostname) [openvpn]:
Name []:
Email Address [openvpn@example.com]:
```

```
Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
Using configuration from /etc/openvpn/easy-rsa/2.0/openssl.cnf
Enter pass phrase for /etc/openvpn/easy-rsa/2.0/keys/ca.key:
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'RS'
stateOrProvinceName   :PRINTABLE:'RS'
localityName          :PRINTABLE:'Beograd'
organizationName      :PRINTABLE:'AMRES'
commonName            :PRINTABLE:'openvpn'
emailAddress          :IA5STRING:'openvpn@example.com'
Certificate is to be certified until Feb 13 17:03:04 2023 GMT (3650 days)
Sign the certificate? [y/n]:y

1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

5. Da bi se uvela dodatna zaštita potrebno je generisati još jedan deljeni tajni ključ, `tls-auth` ključ, na sledeći način:

   ```
   $ cd /etc/openvpn/easy-rsa/keys
   $ openvpn --genkey --secret ta.key
   ```

   Komanda generiše tajni deljeni ključ i upisuje u `ta.key` fajl. Ovaj ključ treba preneti na siguran način na VPN server i VPN klijent.

   Dodatna zaštita koja se na ovaj način postiže podrazumeva dodavanje HMAC (*Hash-based Message Authentication Code*) potpisa svim paketima koji se razmenjuju u toku SSL/TLS *handshake* procesa radi provere integriteta tih paketa. Bilo koji UDP paket koji nema ispravan HMAC potpis biće odbačen bez dalje obrade. Na ovaj način server je dodatno zaštićen od:

   - DoS napada,
   - skeniranja portova kako bi se odredilo koji UDP port servera je u stanju slušanja,
   - započinjanje SSL/TLS handshake od strane neautorizovanog uređaja (u ovom slučaju pokušaj bi propao kada dođe do autentifikacije, ali sa tls-auth opcijom ovakvi pokušaji se sprečavaju dosta ranije).

6. Takođe je neophodno kreirati ključ koji će se koristiti za enkripciju podataka unutar VPN tunela. OpenVPN koristi DH (*Diffie-Hellman*) algoritam za razmenu i generisanje ključeva. DH prokol obezbeđuje sigurnu razmenu tajnog ključa između VPN klijenta i VPN servera. Skripta `build-dh` u `easy-rsa` direktorijumu se koristi za generisanje DH parametara. Skripta zapravo koristi `openssl dhparam` komandu, a ulazni parametri su definisani u `vars` fajlu. U pitanju je DH grupa 2 (standardno

podešavanje) što se može promeniti modifikovanjem odgovarajućih parametara (pogledati `man dhparam`). Skripta `build-dh` se pokreće komandom

```
root@openvpn:/etc/openvpn/easy-rsa# ./build-dh
Generating DH parameters, 2048 bit long safe prime, generator 2
This is going to take a long time
```

Ovim je završeno generisanje svih potrebnih ključeva, nakon čega se prelazi na podešavanje konfiguracionih fajlova.

2.1.2.2 *Podešavanje konfiguracionih fajlova*

OpenVPN program može da se konfiguriše preko komandne linije ili konfiguracionih fajlova. Iako su ova dva načina identična u dokumentu će biti pokazana konfiguracija preko konfiguracionih fajlova jer se taj način pokazao kao jednostavniji u poređenju sa korišćenjem dugačkih komandi. Jedna od značajnih prednosti je i to što je format konfiguracionih fajlova isti na svim operativnim sistemima, pa se može jednostavno kopirati sa jednog operativnog sistema na drugi ukoliko postoji potreba za tim.

Na strani servera potrebno je kreirati konfiguracioni fajl. Primeri konfiguracionih fajlova se nalaze u direktorijumu `sample-config-files`. Ovaj direktorijum se može pronaći pomoću sledeće komande:

```
$  find / -name sample-config-files
```

U nastavku je pretpostavljeno da je direktorijum sa primerima konfiguracionih fajlova pronađen u `/usr/share/doc/openvpn-2.0` direktorijumu.

U nastavku je dat izgled konfiguracionog fajla AMRES VPN servera.

```
$ cp /usr/share/doc/openvpn-2.0/sample-config-files/server.conf.gz /etc/openvpn
$ cd /etc/openvpn
$ gunzip server.conf.gz
$ vi /etc/openvpn/server.conf
```

```
local 147.91.a.b
port 1194
proto udp
dev tun

ca /etc/openvpn/easy-rsa/2.0/keys/ca.crt
cert /etc/openvpn/easy-rsa/2.0/keys/server.crt
key /etc/openvpn/easy-rsa/keys/server.key
dh /etc/openvpn/easy-rsa/keys/dh2048.pem
tls-auth /etc/openvpn/easy-rsa/keys/ta.key 0

server 10.8.0.0 255.255.0.0

plugin /etc/openvpn/radiusplugin/radiusplugin.so /etc/openvpn/radiusplugin/radi
usplugin.cnf
username-as-common-name
```

```
client-cert-not-required
topology subnet

fragment 1200
mssfix 1200

push "route 147.91.0.0 255.255.0.0"
push "route 147.91.a.b 255.255.255.255 net_gateway 0.0.0.0"
client-config-dir ccd
keepalive 10 120
comp-lzo
user nobody
group nogroup
daemon
persist-key
persist-tun
status /var/log/openvpn-status.log
log-append /var/log/openvpn.log
verb 5
```

U daljem tekstu objašnjenje su sve konfiguracione linije.

- Komandom `local` definiše se na kojoj IP adresi će VPN server da prihvata konekcije VPN klijenata. Podešeno je da to bude adresa interfejsa koji se koristi za spoljašnji saobraćaj.

  ```
  local 147.91.a.b
  ```

- Server konfiguriše prvi dostupan tun interfejs sa IP adresom 10.8.0.1 jer `server` direktivom definisan opseg 10.8.0.0/16. Nakon toga server sluša po UDP portu 1194 dolazeće konekcije. Klijenti će se preko ovog porta povezati sa VPN serverom.

  ```
  port 1194
  proto udp
  dev tun
  server 10.8.0.0 255.255.0.0
  ```

- Da bi TLS *handshake* bio moguć neophodno je u serverskom konfiguracionom fajlu navesti putanju do CA sertifikata, javnog i privatnog ključa servera i *Deffie-Helman* parametara. Takođe, potrebno je navesti putanju do direktorijuma u kome se nalazi `ta.key` fajl. Na serverskoj strani pored lokacije fajla dodaje se 0, a na strani klijenta 1.

  U primeru ovi parametri se nalaze u `/etc/openvpn/easy-rsa/keys/` direktorijumu.

  ```
  ca /etc/openvpn/easy-rsa/keys/ca.crt
  cert /etc/openvpn/easy-rsa/keys/openvpn.crt
  key /etc/openvpn/easy-rsa/keys/openvpn.key
  dh /etc/openvpn/easy-rsa/keys/dh2048.pem
  tls-auth /etc/openvpn/easy-rsa/keys/ta.key 0
  ```

- Kako je rešeno da se autentifikacija klijenata vrši na osnovu korisničkog imena i lozinke, a ne na osnovu sertifikata u konfiguracioni fajl servera dodate su direktive.

```
username-as-common-name
client-cert-not-required
```

Prva linija definiše da se korisničko ime, *korisničko-ime@domen.institucije,* koristi kao Common Name parametar ukoliko je to potrebno (koristi se, između ostalog, prilikom korišćenja konfiguracionih direktorijuma za VPN klijente, tzv. CCD (*Client Config Directory*) direktorijuma koji su objašnjeni kasnije). Drugom komandom se konfiguriše da se klijenti neće autentifikovati korišćenjem sertifikata.

- Takođe, pošto se autentifikacija klijenata vrši preko RADIUS protokola neophodno je instalirati i podesiti dodatak za povezivanje sa RADIUS serverom, *radiusplugin*. Ovaj dodatak može da se preuzme sa lokacije, http://www.nongnu.org/radiusplugin/. Nakon instalacije *radiusplugin* dodatka prema uputstvu iz `README` fajla, potrebno direktorijum u kom se nalaze `radiusplugin.cnf` i `radiusplugin.so` fajlovi kopirati u konfiguracioni direktorijum OpenVPN servera (naziv tog direktorijuma je `radiusplugin`). Nakon toga potrebno je podesiti fajl `radiusplugin.cnf` prema odgovarajućim parametrima. Primer tog fajla dat je u nastavku:

```
$ vi /etc/openvpn/radiusplugin/radiusplugin.cnf
```

```
NAS-Identifier=OpenVpn
Service-Type=5
Framed-Protocol=1
NAS-Port-Type=5
NAS-IP-Address=147.91.c.d
OpenVPNConfig=/etc/openvpn/server.conf
subnet=255.255.0.0
overwriteccfiles=true
server
{
        acctport=1813
        authport=1812
        name=147.91.p.q
        retry=1
        wait=1
        sharedsecret=pass
}
```

Promenljiva `NAS-IP-Address` definiše IP adresu kartice OpenVPN servera sa koje će se autentifikacioni zahtevi slati RADIUS serveru. Kako je maska za opseg adresa koji će se koristiti prilikom dodeljivanja IP adresa klijentima 255.255.0.0, ta vrednost je dodeljena promenljivoj `subnet`. Pod sekcijom `server` potrebno je podesiti IP adresu RADIUS servera kome će VPN server slati autentifikacione zahteve za VPN klijente (promenljiva `name`), što je u ovom slučaju TLR server, portove na koje će se slati zahtevi za autentifikaciju i *accounting* (promenljive `acctport` i `authport`). Port 1812 je standardni port za autentifikaciju, a 1813 za *accounting*. Takođe je važno ovde konfigurisati deljeni tajni ključ koji se koristi kako bi RADIUS server prihvatio autentifikacione zahteve koje šalje VPN server (promenljiva `sharedsecret`). Isti ključ mora da bude definisan i na strani RADIUS servera.

U konfiguracionom fajlu servera neophodno je navesti lokaciju gde se *radiusplugin* dodatak i konfiguracioni fajl nalaze.

```
plugin /etc/openvpn/radiusplugin/radiusplugin.so /etc/openvpn/radiusplugin/r
adiusplugin.cnf
```

- U nekim slučajevima može da se javi potreba da se opcije za određene klijente promene u odnosu na generalna podešavanja. Tada je moguće koristiti `client-config-dir` opciju. Ova opcija dozvoljava administratorima da, na primer, dodele određenu IP adresu klijentu ili da klijentu pošalju odgovarajuće opcije kao što je tip kompresije, IP adresa DNS server, ili da privremeno klijentu onemogući VPN povezivanje. Kada se klijent poveže na server sa svojim kredencijalima, OpenVPN server proverava da li postoji odgovarajući konfiguracioni fajl za tog klijenta (koristi njegovo CN ime kako bi pronašao fajl), tzv. CCD fajl u direktorijumu koji je definisan komandom `client-config-dir`. Ako postoji, server na osnovu informacija iz tog fajla konfiguriše parametre za datog klijenta.

  Dodatak *radiusplugin* koristi upravo konfiguracione fajlove klijenata definisane u ovom direktorijumu prilikom dodeljivanja IP adresa klijentima. Naime, kada RADIUS server dodeli IP adresu autentifikovanom klijentu, *radiusplugin* dodatak će tu informaciju da zabeleži u fajlu čiji je naziv zapravo korisničko ime datog korisnika. Zatim će OpenVPN server da tu informaciju primeni kao što je gore objašnjeno.

  Iz ovog razloga je neophodno kreirati `client-config-dir` direktorijum.

  ```
  $ cd /etc/openvpn/
  $ mkdir ccd
  ```

  A zatim sledeću konfiguracionu liniju dodati u `server.conf` fajl.

  ```
  client-config-dir ccd
  ```

- Komanda `keepalive` podešava odgovarajuće tajmere i na klijentskoj i na serverskoj strani. Prema podešavanju AMRES VPN servera svakih 10 sekundi šalje se paket od servera ka klijentu i obrnuto, kako bi se proverilo da li je VPN tunel aktivan i radi. Ako se ne dobije odgovor u roku od 120s na klijentskoj strani, VPN konekcija se automatski restartuje. Na serverskoj strani *timeout* period se duplira, tako da server restartuje VPN konekciju ukoliko se odgovor ne primi posle 240s.

  ```
  keepalive 10 120
  ```

- Pre enkripcije podataka podaci se opciono mogu kompresovati korišćenjem LZO (*Lempel-Ziv-Oberhumer*) biblioteka.

  ```
  comp-lzo
  ```

- Komande `mssfix` i `fragment` se koriste za podešavanje dužine UDP paketa koji se šalju kroz VPN tunel. Direktiva `fragment` definiše maksimalnu dužinu UDP datagrama. Preciznije, navedena vrednost se odnosi na dužinu paketa nakon enkapsulacije, ali bez UDP zaglavlja. Direktiva `mssfix` se koristi da bi se u slučaju TCP sesije koja se ostvaruje preko VPN tunela ograničila veličina paketa tako da nakon što OpenVPN enkapsulira pakete, rezultujući UDP datagram ne bude duži od veličine definisane ovom komandom. Komanda `mssfix` sprečava fragmentaciju paketa, ali ako se ipak desi da paket veće dužine treba da se pošalje kroz tunel (npr. nije u pitanju TCP protokol) zbog `fragment` komande OpenVPN server će taj paket da fragmentira. Potreba za ovakvom konfiguracijom se javila kada je određeni broj korisnika

konstantno imao problem sa VPN konekcijom. Naime, VPN konekcija je mogla uspešno da se ostvari, ali korisnici nisu mogli da je aktivno koristite. Utvrđeno je da su svi korisnici koji su imali ovaj problem lokalnu vezu ka Internetu ostvarivali preko istog Internet servis provajdera koji je smanjio MTU (*Maximum Transmission Unit*) vrednost na svojim uređajima ispod standardne. Na ovaj način problem je uspešno rešen.

```
fragment 1200
mssfix 1200
```

- AMRES VPN server je konfigurisan da rute za mrežne opsege koji su dostupni preko VPN tunela šalje svojim klijentima. To je omogućeno `push` komandom. VPN server šalje rute ka AMRES adresnim opsezima. Osim toga server šalje i rutu ka IP adresi svog eksternog interfejsa, 147.91.a.b, koja pakete ka toj adresi usmerava ka lokalnom *gateway* uređaja VPN klijenta, a ne šalje ih kroz VPN tunel. Razlog ovakve konfiguracije je što adresa eksternog interfejsa VPN servera pripada AMRES adresnom opsegu, a preko nje se VPN veza ostvaruje.

```
push "route 147.91.0.0 255.255.0.0"
push "route 147.91.a.b 255.255.255.255 net_gateway 0.0.0.0"
```

- OpenVPN se pokreće kao proces (`daemon`), a iz sigurnosnih razloga, OpenVPN proces se dodeljuje korisniku `nobody` i grupi `nogroup`, komandama `user` i `group` čime se smanjuju privilegije. Na taj način, čak iako udaljeni napadač uspe da kompromituje OpenVPN proces, biće mu dodeljne privilegije korisnika `nobody`, a ne korisnika `root`. Međutim, prilikom ponovnog pokretanja procesa korisnik `nobody` neće imati dovoljno privilegija da pristupi određenim resursima, zato je neophodno dodati direktive `persist-tun` i `persist-key`, koje sprečavaju da se tun intefejs resetuje prilikom ponovnog pokretanja procesa i da se određeni ključevi za čitanje zaštićenih fajlova ne moraju ponovo učitavati.

```
user nobody
group nogroup
daemon
persist-key
persist-tun
```

- Zbog direktive `log-append` OpenVPN proces će da upisuje *debug* informacije i poruke u navedeni fajl. Nove log informacije neće prebrisati stare, već će se dodativati (*append*). Direktiva `status` definiše upisivanje informacija o statusu trenutne VPN konekcije u navedeni fajl. Prilikom nadgledanje OpenVPN procesa i rešavanja eventualnih problema ove informacije mogu biti od velikog značaja.

```
status /var/log/openvpn-status.log
log-append /var/log/openvpn.log
```

- Direktivom `verb` se definiše koliko će biti detaljne informacije koje se upisuju u log fajlove. Empirijskom metodom zaključeno je da nivo 5 daje dovoljno informacija kako bi se rešavali problemi uspostave konekcije koji se i najčešće dešavaju.

```
verb 5
```

Ostale konfiguracione linije mogu da se zakomentarišu (dodavanjem karaktera `#` na početku linije).

2.1.3 Konfiguracija VPN klijenta

Nakon konfiguracije VPN servera neophodno je konfigurisati i VPN klijenta. Radi autentifikacije servera, neophodno je kopirati `ca` sertifikat na klijentu. Klijent takođe treba da sadrži isti `ta.key` ključ kao i server. Kako se klijenti autentifikuju korišćenjem korisničkog imena i lozinke, neophodno je navesti direktivu `auth-user-pass`. Ostala podešavanja koja su neophodne da bi VPN komunkacija se serverom koji konfigursan kao što je objašnjeno u prethodnom poglavlju uspela su: specificiranje korišćenja TUN interfejsa, enkapsulaciju paketa u UDP protokol, fragmentacija i tip kompresije. Sadržaj konfiguracionog fajla VPN klijenta dat je u nastavku.

```
client
fragment 1200
mssfix 1200
dev tun
proto udp
remote vpn-example.amres.ac.rs 1194
resolv-retry infinite
nobind
persist-key
persist-tun
ca ca.crt
ns-cert-type server
tls-auth ta.key 1
comp-lzo
verb 1
auth-user-pass
pull
explicit-exit-notify
```

Ime `vpn-example.amres.ac.rs` je DNS ime za eksternu mrežnu karticu VPN servera, 147.91.a.b.

2.2 RADIUS infrastruktura

AMRES VPN servis se oslanja na RADIUS infrastrukturu preko koje se vrši autentifikacija, autorizacija i praćenje aktivnosti korisnika. AMRES rešenje je realizovano korišćenjem FreeRADIUS platforme, stoga se ovo poglavlje odnosi na podešavanje osnovnih modula FreeRADIUS servera za potrebe konfiguracije TLR servera. Dat je i primer konfiguracije RADIUS servera institucije, takođe korišćenjem FreeRADIUS platforme.

U radu je pretpostavljeno da je FreeRADIUS program instaliran i ne postoje dodatne promene osim navedenih.

FreeRADIUS server se podešava modifikovanjem određenih konfiguracionih fajlova. Lokacija ovih fajlova zavisi od načina na koji je FreeRADIUS paket instaliran:

- Ako je instaliran standardni FreeRADIUS paket uključen u distribuciju operativnog sistema, konfiguracioni fajlovi će se nalaziti u `/etc/raddb` direktorijumu za CentOS ili u `/etc/freeradius` direktorijumu za Ubuntu operativni sistem.

- Ako je FreeRADIUS program kompajliran i instaliran korišćenjem `configure`, `make`, `make install` komandi konfiguracioni fajlovi će biti smešteni u `/usr/local/etc/raddb` direktorijumu.

U nastavku dokumenta je pretpostavljeno da je `/usr/local/etc/raddb` konfiguracioni direktorijum za FreeRADIUS.

Konfiguracija FreeRADIUS servera je logički podeljena na različite konfiguracione fajlove. Ovi fajlovi se modifikuju kako bi se konfigurisale određene funkcije, komponente i moduli za FreeRADIUS. Glavni konfiguracioni fajl je `radiusd.conf` i u sklopu ovog fajla sadržaj drugih konfiguracionih fajlova se uključuje pomoću komande `$INCLUDE`. Konfiguracioni fajlovi od značaja za podešavanje servera za potrebe predstavljenog rešenja su:

- `proxy.conf` – koristi se za definisanje domena za koje će se zahtevi prosleđivati drugim RADIUS serverima.

- `clients.conf` – fajl u okviru kog se definišu klijenti RADIUS servera. Bilo koji pristupni uređaj koji treba da šalje zahteve FreeRADIUS serveru mora da bude definisan u ovom fajlu. To može da bude neki mrežni uređaj, VPN server, drugi RADIUS server itd. Za svakog klijenta u ovom fajlu definiše se deljena lozinka (engl. *shared secret*) koja mora da bude definisana i na datom uređaju i koja omogućava sigurnu komunikaciju uređaja sa FreeRADIUS serverom.

- `sql.conf` – sadrži konfiguraciju `sql` modula. U ovom fajlu se nalaze sva potrebna podešavanja za povezivanje i korišćenje sql baze podataka.

- virtuelni server - Virtuelni serveri omogućavaju konfiguraciju većeg broja nezavisnih RADIUS servisa na FreeRADIUS platformi. Virtuelni server se kreira u `/sites-available` direktorijumu, a aktivira kreiranjem linka u `/sites-enabled` direktorijumu. Nakon instalacije FreeRADIUS servera aktivirana su dva virtuelna servera `default` i `inner-tunnel`. Virtuelni server `default` se koristi za tipične zahteve, a `inner-tunnel` za konfiguraciju EAP metoda. Ovde se koristi samo `deafult` virtuelni server.

- `users` – nalazi se u FreeRADIUS konfiguracionom direktorijumu. Sadržaj ovog fajla može da se koristi u svrhu autentifikacije i autorizacije. U ovom fajlu se unose podešavanja vezana za korisnike.

2.2.1 Konfiguracija Top Level RADIUS servera

U ovom poglavlju dato je objašnjenje konfiguracionih fajlova neophodnih za podešavanje TLR servera. Treba imati u vidu da je TLR server *proxy* RADIUS server, kao i da se preko njega vrši dodeljivanje IP adresa autentifikovanim VPN korisnicima i obrađuju podaci o aktivnostima VPN korisnika.

2.2.1.1 *vpn virtuelni server*

Unutar konfiguracionog fajla virtuelnog servera nalaze se sledeće sekcije: `listen`, `client`, `authorize`, `authenticate`, `post-auth`, `pre-proxy`, `post-proxy`, `preacct`, `accounting` i `session`. Osnovna procedura obrade pristiglog autentifikacionog zahteva se sastoji iz sledećih koraka:

- Kada FreeRADIUS server primi zahtev za autentifikaciju korisnika, *Access-Request*, zahtev se prvo obrađuje u `authorize` sekciji virtuelnog servera. Moduli definisani u `authorize` sekciji određuju

mehanizme koji se koriste za autentifikaciju korisika. U ovoj sekciji zahtev može dodatno da se obrađuje.

- Nakon što se odredi tip autentifikacije, zahtev se šalje u `authenticate` sekciju. U okviru ove sekcije odgovarajuća podsekcija će preuzeti zahtev i izvršiti autentifikaciju.
- Nakon uspešne autentifikacije prelazi se na `post-auth` sekciju.
- Za obradu pristiglih *accounting* poruka zadužena je `accounting` sekcija.

TLR server treba da primljene autentifikacione zahteve klijenta samo prosledi RADIUS serveru matične institucije klijenta. Potrebno je konfigurisati jedan virtuelni server koji će obrađivati autentifikacione zahteve na taj način. U tu svrhu u direktorijumu `/sites-available` kopiran je sadžaj `default` konfiguracionog fajla u novi `vpn` fajl.

Kreirani konfiguracioni fajl je izmenjen prema konfuguraciji koja je data u nastavku:

```
$ cd /usr/local/etc/raddb/sites-available
$ cp default vpn
$ vi vpn
```

```
server vpn {
authorize {
        preprocess
        auth_log
        chap
        mschap
        digest
        suffix
        eap
        files
        expiration
        logintime
        pap
}
authenticate {
        Auth-Type PAP {
                pap
        }
        Auth-Type CHAP {
                chap
        }
        Auth-Type MS-CHAP {
                mschap
        }
        digest
        unix
        eap
}
preacct {
        preprocess
        acct_unique
        suffix
        files
}
accounting {
        detail
        radutmp
        sqlippool
        sql
        exec
        attr_filter.accounting_response
```

```
}
session {
        radutmp
}
post-auth {
        sqlippool
        reply_log
        exec
        Post-Auth-Type REJECT {
                attr_filter.access_reject
        }
}
pre-proxy {
        pre_proxy_log
}
post-proxy {
        post_proxy_log
        eap
}
}
```

Od značaja su sledeće promene u odnosu na `default` konfiguracioni fajl:

1. Na samom početku fajla (iznad `authorize` sekcije) dodata je linija `server vpn {` , a na kraju fajla `}`
2. U `authorize` sekciji naveden je modul `files`. Isti modul se nalazi i u `default` virtuelnom serveru. Modul files se koristi za čitanje `users` konfiguracionog fajla za autentifikaciju i autorizaciju klijenata. U slučaju TLR servera `users` fajl se koristi kako bi se pristiglim autentifikacionim zahtevima pridružila informacija o imenu opsega IP adresa iz kog će se autentifikovanim klijentima dodeliti IP adresa. Konfiguracija opsega IP adresa je objašnjena u sledećoj tački. IP adresa se dodeljuje na osnovu domena iz korisničkog imena navedenog u autentifikacionom zahtevu. U tu svrhu je na kraju `users` fajla dodata sledeća komanda za svaku AMRES instituciju korisnicu VPN servisa

   ```
   $ vi /usr/local/etc/raddb/users
   . . . .
   DEFAULT Suffix == "inst1.ac.rs", Pool-Name := pool-inst1
   . . . .
   ```

   `DEFAULT` direktiva znači da se konfiguracija koja sledi odnosi na bilo koje korisničko ime, `Suffix` je atribut koji označava domen iz korisničkog imena, a `Pool-Name` atribut koji označava opseg IP adresa iz kog se dodeljuju IP adrese klijentima. Dakle, bilo koji korisnik čiji je domen `inst1.bg.ac.rs` će dobiti IP adresu iz opsega pod imenom `pool-inst1`.

3. U `post-auth` sekciji se nalaze konfiguracione linije koje se odnose na dodeljivanje IP adresa autentifikovanim VPN klijentima.

   Napomena: Za dodeljivanje IP adresa VPN korisnicima prvo je korišćen `rlm_ippool` modul. Međutim, pojavio se problem dodeljivanja iste IP adrese različitim VPN koristinicima istovremeno. Zato je dodeljivanje IP adresa VPN klijentima realizovano korišćenjem `rlm_sqlippool` modula. U tu svrhu potrebno je u `radiusd.conf` fajlu u sekciji `modules` dodati sledeće konfiguracione linije:

   ```
   $ vi /usr/local/etc/raddb/radiusd.conf
   ```

   ```
   . . . .
   modules {
   . . . .
   $INCLUDE sql.conf
   ```

```
    $INCLUDE sqlippool.conf
    . . . .
```

Fajl `sql.conf` se nalazi u konfiguracionom direktorijumu FreeRADIUS servera (`/usr/local/etc/raddb/sql.conf`), predstavlja konfiguracioni fajl `sql` modula i sadrži sve konfiguracione parametre neophodne za povezivanje sa odgovarajućom sql bazom u bilo koju svrhu. U slučaju TLR servera koristi se za dodeljivanje IP adresa VPN klijentima i za *accounting*, odnosno u istoj sql bazi nalazi se tabela opsega IP adresa koje se alociraju korisnicima i tabele koje se popunjavaju prilikom praćenja aktivnosti korisnika.

Fajl `sqlippool.conf` se takođe u konfiguracionom direktorijumu (`/usr/local/etc/raddb/sqlippool.conf`) ali on se odnosi samo na dodeljivanje IP adresa klijentima. Ovaj fajl predstavlja konfiguracioni fajl modula `sqlippool` koji se poziva u `post-auth` sekciji `vpn` virtuelnog servera. U nastavku je data konfiguracija ovog fajla

```
$ vi /usr/local/etc/raddb/sqlippool.conf
```

```
sqlippool {
 sql-instance-name = "sql"
 ippool_table = "radippool"
 lease-duration = 86400
 pool-key = "%{User-Name}"
$INCLUDE sql/mysql/ippool.conf
 sqlippool_log_exists = "Existing IP: %{reply:Framed-IP-Address} \
   (did %{Called-Station-Id} cli %{Calling-Station-Id} port %{NAS-Port}
user %{User-Name})"
 sqlippool_log_success = "Allocated IP: %{reply:Framed-IP-Address}
from %{control:Pool-Name} \
   (did %{Called-Station-Id} cli %{Calling-Station-Id} port %{NAS-Port}
user %{User-Name})"
 sqlippool_log_clear = "Released IP %{Framed-IP-Address}\
  (did %{Called-Station-Id} cli %{Calling-Station-Id} user %{User-Name})"
 sqlippool_log_failed = "IP Allocation FAILED from %{control:Pool-Name} \
   (did %{Called-Station-Id} cli %{Calling-Station-Id} port %{NAS-Port}
user %{User-Name})"
 sqlippool_log_nopool = "No Pool-Name defined \
   (did %{Called-Station-Id} cli %{Calling-Station-Id} port %{NAS-Port}
user %{User-Name})"
}
```

Od značaja su osenčene linije. U okviru ovog konfiguracionog fajla poziva `sql` modul koji je definisan u `sql.conf` fajlu. Za tabelu u kojoj će biti unet opseg IP adresa za dodeljivanje VPN klijentima (`ippool_table`) definisana je tabela pod nazivom `radippool` koja je definisana u odgovarajućoj MySQL bazi što će u nastavku biti objašnjeno. Takođe je definisano i vreme dodeljivanja adresa u okviru parametra `lease-duration` (u primeru je konfigurisano da to bude 24 sata). Parametar na osnovu kog se dodeljuje jedinstvena IP adresa (`pool-key`) je korisničko ime (`User-Name`).

Konfiguracioni fajl `ippool.conf` koji se poziva komandom `$INCLUDE` definiše različite upite u MySQL bazu koji služe za dodeljivanje IP adresa klijentima i za oslobađanje IP adresa u različitim slučajevima. Posebnu pažnju treba obratiti na konfiguraciju ovog fajla jer se može desiti da sintaksa koja je korišćena ne odgovara verziji MySQL baze, pa određene funkcije neće biti uspešno izvršene (npr. oslobađanje IP adresa klijenata koji su prekinuli VPN konekciju). U tom slučaju potrebno je modifikovati upite tako da sintaksa odgovara verziji MySQL baze koja se koristi.

Napomena: za kreiranje `radippool` tabele koristi se `/usr/local/etc/raddb/sql/mysql/ippool.sql` šema.

```
$ mysql -u root -p radius < /usr/local/etc/raddb/sql/mysql/ippool.sql
```

U nastavku je data konfiguracija `sql.conf` fajla.

```
$ vi /usr/local/etc/raddb/sql.conf
```

```
sql {
        database = "mysql"
        driver = "rlm_sql_${database}"
        server = "147.91.x.y"
        login = "radiusUSER"
        password = "radiusPASS"
        radius_db = "radius"
        acct_table1 = "radacct" # tabela koja se koristi za accounting
        acct_table2 = "radacct" # tabela koja se koristi za accounting
        postauth_table = "radpostauth"
        authcheck_table = "radcheck"
        authreply_table = "radreply"
        groupcheck_table = "radgroupcheck"
        groupreply_table = "radgroupreply"
        usergroup_table = "radusergroup"
        deletestalesessions = yes
        sqltrace = no
        sqltracefile = ${logdir}/sqltrace.sql
        num_sql_socks = 5
        connect_failure_retry_delay = 60
        lifetime = 0
        max_queries = 0
        nas_table = "nas"
        $INCLUDE sql/${database}/dialup.conf # popunjavanje radacct
#tabele
}
```

U `sql` sekciji `sql.conf` fajla su definisani parametri MySQL baze podataka u čijoj `radipool` tabeli `radius` baze se definišu opsezi adresa iz kojih se autentifikovanim korisnicima dodeljuju IP adrese na osnovu domena iz korisničkog imena. Svaka institucija ima svoj opseg adresa, i svaki od tih opsega ima pridruženo ime. Imena opsega IP adresa su mapirana sa domenima institucija u okviru `users` fajla (tačka 2). MySQL baza je instalirana na hostu sa IP adresom 147.91.x.y (direktiva `server = "147.91.x.y"`). Korisničko ime i lozinka za pristup bazi definisani su promenljivim `login` i `password` i kreirani su samo za ove potrebe. Promenljiva `radius_db` nosi ime baze u okviru koje je definisana `radipool` tabela (u primeru baza `radius`). Naravno, za svaku instituciju neophodno je u tabeli `radippool` definisati opseg IP adresa koje će se dodeljivati VPN korisnicima te institucije.

Primer izgleda `radippool` tabele dat je na slici

```
mysql> use radius
Database changed
mysql> select * from radippool;
+----+-------------+----------------+-------------+-----------------+-
----------------+--------------------+-------------------+-----------
--------+
| id | pool_name    | framedipaddress | nasipaddress | calledstationid |
callingstationid | expiry_time         | username           | pool_key
|
+----+-------------+----------------+-------------+-----------------+-
----------------+--------------------+-------------------+-----------
--------+
|  1 | pool-inst1  | 10.8.1.1    |              |                 |
| NULL                |                     |                    |
```

```
|  2 | pool-inst1  | 10.8.1.2    |              |                  |
| NULL                |                     |                      |
+----+-------------+----------------+-------------+----------------+-
----------------+--------------------+-------------------+-----------
--------+
```

4. I u sekciji `accounting` virtuelnog servera `vpn` potrebno je dodati direktivu `sqlippool`. Na ovaj način se omogućava da kada klijent prekine VPN konekciju, RADIUS server njegovu IP adresu oslobodi kako bi je mogao ponovo dodeliti.

- U sekciji se, takođe, navode konfiguracione linije koje definišu način na koji RADIUS server obrađuje *accounting* poruke o aktivnosti korisnika. VPN server će informacije o aktivnostima korisnika slati TLR serveru koji će ih, zatim, na osnovu informacija iz ove sekcije upisivati u odgovarajuću sql bazu. Zato je od značajna konfiguraciona linija `sql` u `accounting` sekciji. Ona zapravo predstavlja sekciju unutar konfiguracionog fajla `sql` modula, `sql.conf` koji je detaljno objašnjen u tački 3 (poziva se isti `sql` modul).
- Dakle, u pitanju je ista MySQL baza u kojoj je konfigurisana pomenuta `radius` baza. U ovoj bazi je takođe kreirana tabela `radacct` u koju se upisuju aktivnosti korisnika.
  Napomena: za kreiranje tabele za potrebe *accounting* mehanizma koristi se `/usr/local/etc/raddb/sql/mysql/schema.sql` šema.

  ```
  $ mysql -u root -p radius < /usr/local/etc/raddb/sql/mysql/schema.sql
  ```

  Konfiguracioni fajl koji se poziva na kraju ove sekcije `sql/${database}/dialup.conf` definiše kojim podacima će se popunjavati `radacct` tabela. Ovaj fajl već postoji na naznačenoj lokaciji i standardno se u okviru sql.conf konfiguracionog fajla poziva, a moguće ga je modifikovati prema potrebama.

  Svi ostali konfiguracioni parametri su standardni.

5. Da bi se vpn virtuelni server aktivirao potrebno je napraviti link ka `vpn` konfiguracionom fajlu u `/sites-enabled` direktorijumu:

   ```
   $ cd /usr/local/etc/raddb/sites-enabled
   $ ln –s /usr/local/etc/raddb/sites-available/vpn
   ```

2.2.1.2 *clients.conf*

TLR server prima autentifikacione zahteve samo od VPN servera zato se u njegovom `clients.conf` fajlu nalazi samo konfiguracija koja se odnosi na VPN server i data je u nastavku:

```
$ vi /usr/local/etc/raddb/clients.conf
```

```
. . . .
## AMRES VPN server
client vpn-server {
   ipaddr           = 147.91.c.d # IP adresa internog interfejsa VPN servera
   secret           = pass  # lozinka definisana i na strani VPN servera
   shortname        = OpenVPN
   nastype          = other
   virtual_server   = vpn
}
```

Veoma je bitno za paremetar `virtual_server` navesti `vpn` virtuelni server jer će se u tom slučaju zahtevi koji stižu od ovog klijenta obrađivati prema konfiguraciji `vpn` virtulenog servera. Deljena lozinka koja je ovde definisana je ista lozinka definisana u `radiusplugin.cnf` konfiguracionom fajlu prilikom konfiguracije VPN servera (poglavlje 2.1.2.2).

2.2.1.3 *proxy.conf*

Konfiguracioni fajl `proxy.conf` predstavlja modul na osnovu kog RADIUS server odlučuje da li će pristigli autentifikacioni zahtev biti obrađen lokalno ili će biti prosleđen nekom drugom serveru. U tu svrhu definišu se `home_server_pool`, `home_server` i `realm` parametri. Parametar `home_server` se koristi za definisanje pojedinačnih RADIUS servera kojima će se autentifikacioni zahtevi prosleđivati. Server definisan ovde grupiše se u `home_server_pool` skupu. Pod sekcijom `realm` se definišu domeni za koje se očekuje da će se naći u pristiglim autentifikacionim zahtevima. Za domen definisan u okviru `realm` sekcije se navodi odgovarajući `home_server_pool` skup koji sadrži RADIUS servere kojima će se zahtevi korisnika sa navedenim domenom prosleđivati.

S obzirom da je ovde reč o konfiguraciji TLR servera, svi pristigli zahtevi će biti prosleđivani RADIUS serverima institucija. Zato je za domen svake institucije potrebno definisati `realm`, `home_server` i `home_server_pool` sekcije prema primeru koji je dat u nastavku.

```
$ vi /usr/local/etc/raddb/proxy.conf
```

```
. . . .
realm inst1.ac.rs {
        auth_pool = inst1-server-pool
        nostrip
}
home_server_pool inst1-server-pool {
        type = fail-over
        home_server = inst1-home-server
}
home_server inst1-home-server {
        type = auth
        ipaddr =  # IP adresa servera institucije
        port = 1812
        secret = passINST1 # lozinka definisana na RADIUS server institucij
        response_window = 20
        zombie_period = 40
        revive_interval = 120
        status_check = none
        check_interval = 30
        num_answers_to_alive = 3
}
```

Ovim je završena konfiguracija vašeg TLR servera. Potrebno je pokrenuti server u *debug* modu:

- zaustavljanje FreeRADIUS servisa
    - `killall radiusd` za CentOS
    - `killall freeradius` za Ubuntu

- pokretanje servisa u debug modu:
    - `radiusd -X` za CentOS
    - `freeradius -X` za Ubuntu

U ovom modu moguće je proveriti da li FreeRADIUS server učitao konfiguraciju, i da li su definisani portovi otvoreni (1812, 1813).

2.2.2 Primer konfiguracije RADIUS servera institucije

U ovom poglavlju dat je primer konfiguracije FreeRADIUS servera institucije učesnice AMRES VPN servisa. RADIUS server institucije prihvata autentifikacione zahteve svojih korisnika od TLR servera i obrađuje ih. Svaka institucija podatke o korisnicima čuva u nekoj bazi podataka. Povezivanje FreeRADIUS servera sa konkretnim bazama podataka je van okvira ovog dokumenta. Ovde je dat primer korišćenja `users` fajla za autentifikaciju korisnika. Kao i u slučaju TLR servera poglavlje obuhvata konfiguraciju svih relevantnih FreeRADIUS konfiguracionih fajlova.

2.2.2.1 *vpn virtuelni server*

Potrebno je konfigurisati virtuelni server koji će obrađivati autentifikacione zahteve pristigle od TLR servera. U tu svrhu u direktorijumu `/sites-available` kopira se sadžaj `default` konfiguracionog fajla u novi `vpn` fajl.

Kreirani konfiguracioni fajl je izmenjen prema konfuguraciji koja je data u nastavku:

```
$ cd /usr/local/etc/raddb/sites-available
$ cp default vpn
$ vi vpn
```

```
server vpn {
authorize {
        auth_log
        suffix
        eap
        files
        pap
}
authenticate {
        Auth-Type PAP {
        pap
}
Auth-Type CHAP {
        chap
}
Auth-Type MS-CHAP {
        mschap
}
        unix
        eap
```

```
}
session {
        radutmp
}
post-auth {
        Post-Auth-Type REJECT {
        attr_filter.access_reject
        }
}
pre-proxy {
}
post-proxy {
        eap
}
}
```

Od značaja su sledeće promene u odnosu na `deafult` konfiguracioni fajl:

1. Na samom početku fajla dodata je linija `server vpn {` , a na kraju fajla `}`

2. U `authorize` sekciji naveden je modul `files`. Isti modul se nalazi i u `default` virtuelnom serveru. Modul `files` se ovde koristi za čitanje `users` konfiguracionog fajla za autentifikaciju klijenata. Korisničko ime i lozinka mogu da budu navedeni u ovom konfiguracionom fajlu. Primer konfiguracione linije koja može da se koriste u tu svrhu je:

   ```
   $ vi /usr/local/etc/raddb/users
   . . . .
   "alice" Cleartext-Password := "passme"
   . . . .
   ```

   Dakle, definisan je korisnički nalog sa korisničkim imenom *alice* i lozinkom *passme*.

3. Da bi se `vpn` virtuelni server aktivirao potrebno je napraviti link ka `vpn` konfiguracionom fajlu u `/sites-enabled` direktorijumu:

   ```
   $ cd /usr/local/etc/raddb/sites-enabled
   $ ln –s /usr/local/etc/raddb/sites-available/vpn
   ```

2.2.2.2 *clients.conf*

RADIUS server institucije prima autentifikacione zahteve samo od TLR servera zato je potrebno da i njegovom `clients.conf` fajlu bude konfigurisan TLR server, na sledeći način:

```
$ vi /usr/local/etc/raddb/clients.conf
```

```
. . . .

##AMRES VPN
client amres.vpn.radius {
ipaddr = tlr.amres.ac.rs
secret = passINST1
```

```
shortname = TLR
nastype = other
virtual_server = vpn
}
```

Veoma je bitno za parematar `virtual_server` navesti `vpn` virtuelni server jer će se u tom slučaju zahtevi koji stižu od ovog klijenta obrađivati prema konfiguraciji `vpn` virtulenog servera. Ime `tlr.amres.ac.rs` je DNS ime TLR servera, a deljena lozinka je ista lozinka koja je za RADIUS server ove institucije definisana u `proxy.conf` fajlu TLR servera.

2.2.2.3 *proxy.conf*

Obzirom da je reč o konfiguraciji RADIUS servera institucije, svi pristigli zahtevi će biti obrađivani lokalno, tako da je potrebno kreirati samo lokalni domen (*realm*) i u njemu označiti da se autentifikacija vrši lokalno. Primer konfiguracije `proxy.conf` fajla je dat u nastavku.

```
$ vi /usr/local/etc/raddb/proxy.conf
```

```
proxy server {
        default_fallback = no
}
home_server localhost {
        type = auth+acct
        ipaddr = 127.0.0.1
        port = 1812
        secret = testing123
        response_window = 20
        zombie_period = 40
        revive_interval = 120
        status_check = status-server
        check_interval = 30
        num_answers_to_alive = 3
}
realm inst1.ac.rs {
        authhost        = LOCAL
        accthost        = LOCAL
        User-Name       = "%{Stripped-User-Name}"
}
realm LOCAL {
}
realm NULL {
}
```

Gde je `inst1.ac.rs` domen institucije čiji je RADIUS server konfigurisan.

3 Zaključak

Ovim je kompletirana konfiguracija osnovnih komponenti AMRES VPN servisa. Ovaj servis je razvijen kako bi svim korisnicima AMRES institucija pružila mogućnost pristupa akademskoj mreži i korišćenje servisa koje nudi i kada se ne nalaze na svojim institucijama. Krajnji korisnici, naravno, moraju da poštuju uslove korišćenja VPN servisa. Kada ostvari VPN vezu uređaj VPN korisnika postaje deo AMRES mreže, zato krorisnici moraju da se pridržavaju pravila definisanih u Pravilniku o korišćenju Akademske mreže Srbije – AUP, koja se odnose kako na adekvatnu zaštitu njihovih računara, tako i na sve mrežne aktivnosti koje obavljaju na AMRES mreži.

VPN servis se oslanja na već razvijenu RADIUS infrastrukturu koja se koristi u okviru eduroam servisa. Takođe, da bi institucija svojim korisnicima omogućila da pristupe VPN servisu, neophodno je da im obezbedi korisničko ime i lozinku. Kvalitet podataka i kvalitet procedura dodeljivanja, održavanja i ukidanja identiteta moraju da budu na zadovoljavajućem nivou. Osim korisničkog imena i lozinke, u digitalnom identitetu osobe mora se nalaziti dovoljno podataka da se može identifikovati prava osoba koja koristi određeno korisničko ime. Identiteti se mogu dodeliti samo osobama koje su u datom trenutku u vezi sa matičnom institucijom i moraju biti isključivo lični. Ove uslove već zadovoljavaju sve AMRES institucije koje učestvuju u eduroam servisu. Iz tog razloga u AMRES VPN servisu može da učestvuje svaka institucija koja je AMRES članica i koja učestvuje u AMRES eduroam servisu. Na ovaj način ulaganje AMRES institucija koje žele da postanu korisnice VPN servisa je svedeno na minimum. S druge strane, AMRES pruža punu podršku svim svojim institucijama koje žele implementiraju svoje VPN rešenje.

4 Rečnik

VPN	Virtual Private Network
RADIUS	Remote Authentication Dial In User Service
TLR	Top Level RADIUS
EAP	Extensible Authentication Protocol
SSL	Secure Sockets Layer
TLS	Transport Layer Security
PAP	Password Authentication Protocol
MSCHAPv2	Microsoft Challenge Handshake Auth Protocol version 2
LZO	Lempel-Ziv-Oberhumer
DH	Diffie-Hellman
CA	Certificate Authority
SQL	Structured Query Language
HMAC	Hash-based Message Authentication Code
AUP	Acceptable Use Policy