AMRES
Akademska mreža Srbije

# RADIUS i LDAP konfiguracija

*Istorija verzija dokumenta*

| Verzija | Datum | Inicijali autora | Opis promene |
|---|---|---|---|
| 1.0 | 27.04.2016. | ME | Prva verzija dokumenta |

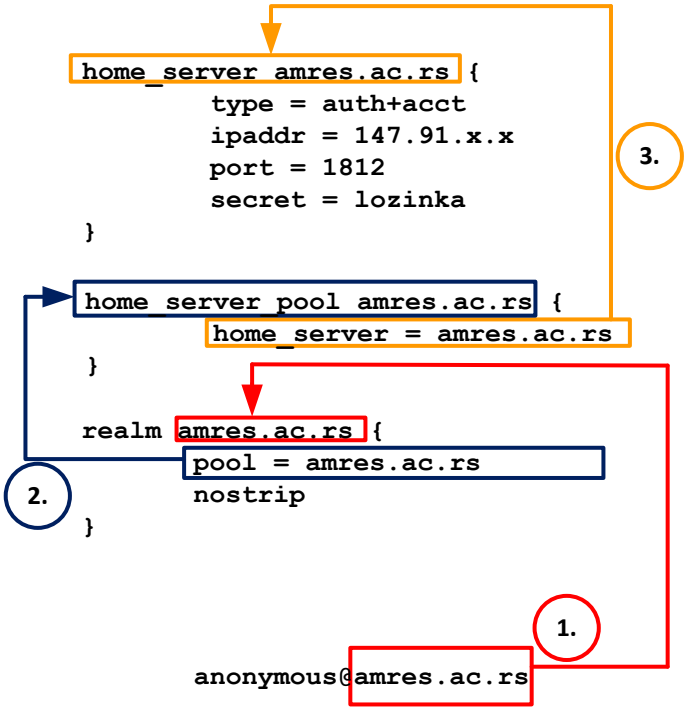
# Sadržaj

# 1 FreeRADIUS konfiguracija

FreeRADIUS je instaliran iz source-a. To znači da se, po default-u, svi konfiguracioni fajlovi nalaze u /usr/local/etc/raddb direktorijumu. U novijim verzijama, počev od verzije 3 i narednih, struktura ovog direktorijuma približno je ista kao struktura apache2 (httpd) direktorijuma. Najvažniji konfiguracioni fajlovi su:

## 1.1 proxy.conf

proxy.conf - namena ovog konfiguracionog fajla je da definiše za koje će sve domene (realm) ovaj RADIUS server biti zadužen. To znači da jedan RADIUS server može da se konfiguriše tako da obrađuje zahteve za jedan ili više domena, ili da samo služi kao proxy server. U slučaju da je konfiguraisan kao proxy, prosleđivaće zahteve do drugih RADIUS servera. Struktura proxy.conf fajla je sledeća:




Kada do vašeg servera stigne zahtev sa nekim domenom, prvo se vrši provera da li je taj domen definisan. U slučaju da jeste, sledeći korak podrazumeva pronalaženje odgovarajućeg pool-a (skupa) servera. Jedan pool mogu sačinjavati grupe servera ili samo jedan server. Potrebno je napraviti izmene u ovom konfiguracionom fajlu tako da imate jedan domen (**realm**) koji odgovara zvaničnom domenu vaše institucije, jedan **home_server_pool** i jedan **home_server**. Obzirom da je pretpostavka da je baza sa korisničim nalozima na istom serveru, nije potrebno definisati ništa dodatno, već iskoristiti **home_server localhost**. Konvencija koja omogućava da se najlakše povezuje domen sa skupom servera je da se za ime **home_server_pool** koristi isto ime kao i za domen. Potrebno je naći liniju „**realm inst.ac.rs**" i tu umesto „inst.ac.rs" uneti domen (npr. za AMRES je to amres.ac.rs). Isto je potrebno uraditi i za „**home_server_pool inst.ac.rs**". Nakon toga, potrebno je snimiti izmene i izaći iz ovog konfiguracionog fajla. Za proveru prethodnih podešavanja, u komandnoj liniji unesite sledeću komandu:

```
cat proxy.conf | grep -v "#" | less
```

Na ovaj način se iz konfiguracionog fajla izbacuju sve linije koje predstavljaju komentar, tj. sve linije koje počinju znakom „#". Izgled konfiguracionog fajla pre izmena i sa izbačenim komentarima je dat u nastavku.

```
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
        check_timeout = 4
        num_answers_to_alive = 3
        max_outstanding = 65536
}
realm LOCAL {
}
realm inst.ac.rs {
        pool = inst.ac.rs
}
home_server_pool inst.ac.rs {
        home_server = localhost
}
```

## 1.2 clients.conf

clients.conf - RADIUS protokol je klijent-server protokol. To znači da je potrebno definisati sa kojim će sve klijentima vaš RADIUS server komunicirati. Komunikacija je moguća jedino u slučaju kada je sa obe strane konfigurisan isti **shared_secret** parametar. Pored ovoga je potrebno podesiti i firewall na samom serveru kao i mrežne uređaje kako bi mogla da se ostvari komunikacija na mrežnom nivou. Prilikom konfiguracije servera za eduroam servis, uvek se dodaju minimalno tri klijenta:

- FTLR1 server,
- FTLR2 server,
- NetIIS server.

FTLR1 i FTLR2 serveri se koriste kako bi korisnicima iz AMRES omogućili „roming", tj. kako bi korisnici mogli nesmetano da se autenfikuju kada se ne nalaze u svojoj matičnoj instituciji. U konfiguracioni fajl je potrebno dodati tri klijenta, tako da ima izgled kao u nastavku:

```
## eduroam Federation Top Level Radius serveri:
##eduroam ftlr1
client ftlr1.ac.rs {
 ipaddr = 147.91.4.204
 secret = pass # - lozinka se dobija od AMRES-a
 shortname = ftlr1
 nastype = other
 virtual_server = eduroam
}
##eduroam ftlr2
client ftlr2.ac.rs {
    ipaddr = 147.91.1.101
    secret = pass # - lozinka se dobija od AMRES-a
    shortname = ftlr2
    nastype = other
    virtual_server = eduroam
}
##Monitoring eduroam servisa
client netiis.monitor {
    ipaddr = 147.91.3.12
    secret = pass # - lozinka se dobija od AMRES-a
    shortname = netiis
    nastype = other
    virtual_server = eduroam
}
```

Nakon što unesete izmene, potrebno ih je snimiti i izaći iz konfiguracionog fajla. Kako biste proverili da li je sve konfigurisano u skladu sa primerom iz ovog priručnika, ponovo pokrenite komandu za ispisivanje fajla bez prikazanih linija sa komentarima.

```
cat proxy.conf | grep -v "#" | less
```

Pored konfiguracionih fajlova, u okviru raddb direktorijuma se nalaze i poddirektorijumi. Izmena u ovim poddirektorijumima utiče na ponašanje servera, tako da je potrebno voditi računa prilikom promena. Najvažniji poddirektorijumi su:

- mods-available
- mods-config
- sites-available i sites-enabled

## 1.3 virtuelni server

virtuelni server - FreeRADIUS je poznat po ovom konceptu. Zahvaljujući njemu, jedan RADIUS server se može konfigurisati tako da se koristi za više različitih servisa (npr. eduroam, VPN, iAMRES). U isto vreme,

virtuelni serveri predstavljaju spregu između konfiguracionih fajlova i modula. Kada se konfiguriše EAP (Extensible Authentication Protocol), uvek moraju postojati dva virtuelna servera - jedan koji je namenjen za uspostavu komunikacije između korisničkog uređaja i RADIUS servera (eduroam) i drugi koji se koristi za proveru kredencijala (korisničkog imena i lozinke) nakon uspostave tunela (eduroam-inner-tunnel). Veza između baze (direktorijuma) u kojoj se nalaze korisnički nalozi se definiše u ovom „unutrašnjem" tunelu. Svi predefinisani virtuelni serveri se nalaze u okviru **sites-available** poddirektorijumu. Kada se kreira novi virtuelni server, on neće biti funkcionalan sve dok se ne „aktivira" u okviru **sites-enabled** poddirektorijuma.

Početna verzija eduroam i eduroam-inner-tunnel virtuelnih servera se dobija kopiranjem **default** i **inner-tunnel** virtuelnih servera, respektivno. Nakon kopiranja, potrebno je napraviti odgovarajuće izmene u konfiguraciji kako bi RADIUS server obrađivao autentifikacione zahteve u skladu sa potrebama. Kreiranje novih virtuelnih servera je neophodno kako bi se napravila razlika u odnosu na već postojeće i dobra je praksa da se za svaki servis defniše virtuelni server sa imenom koje odgovara željenom servisu. U nastavku su dati primeri konfiguracije **eduroam** i **eduroam-inner-tunnel** virtuelnih servera.

```
server eduroam {
listen {
        type = auth
        ipaddr = *
        port = 0
        limit {
              max_connections = 16
              lifetime = 0
              idle_timeout = 30
        }
}

listen {
        ipaddr = *
        port = 0
        type = acct
        limit {
        }
}

listen {
        type = auth
        port = 0
        limit {
              max_connections = 16
              lifetime = 0
              idle_timeout = 30
        }
```

```
}

listen {
        ipv6addr = ::
        port = 0
        type = acct
        limit {
        }
}

authorize {
        filter_username
        preprocess
        auth_log
        suffix
        eap {
                ok = return
        }
        expiration
        logintime
}



authenticate {
        Auth-Type PAP {
                pap
        }

        Auth-Type CHAP {
                chap
        }

        Auth-Type MS-CHAP {
                mschap
        }

        digest
```

```
        eap
}


preacct {
        preprocess
        acct_unique
        suffix
        files
}

accounting {
        detail
        exec
        attr_filter.accounting_response
}

session {
}

post-auth {
        update {
                &reply: += &session-state:
        }
        reply_log
        exec
        remove_reply_message_if_eap
        Post-Auth-Type REJECT {
                attr_filter.access_reject
                eap
                remove_reply_message_if_eap
        }
}

pre-proxy {
}

post-proxy {
```

```
        eap
}
}
```

```
server eduroam-inner-tunnel {
authorize {
        filter_username
        chap
        mschap
        suffix
        update control {
                &Proxy-To-Realm := LOCAL
        }
        files
        -ldap
        pap
}


authenticate {
        Auth-Type PAP {
                pap
        }

        Auth-Type CHAP {
                chap
        }

        Auth-Type MS-CHAP {
                mschap
        }

        eap
}

session {
}
```

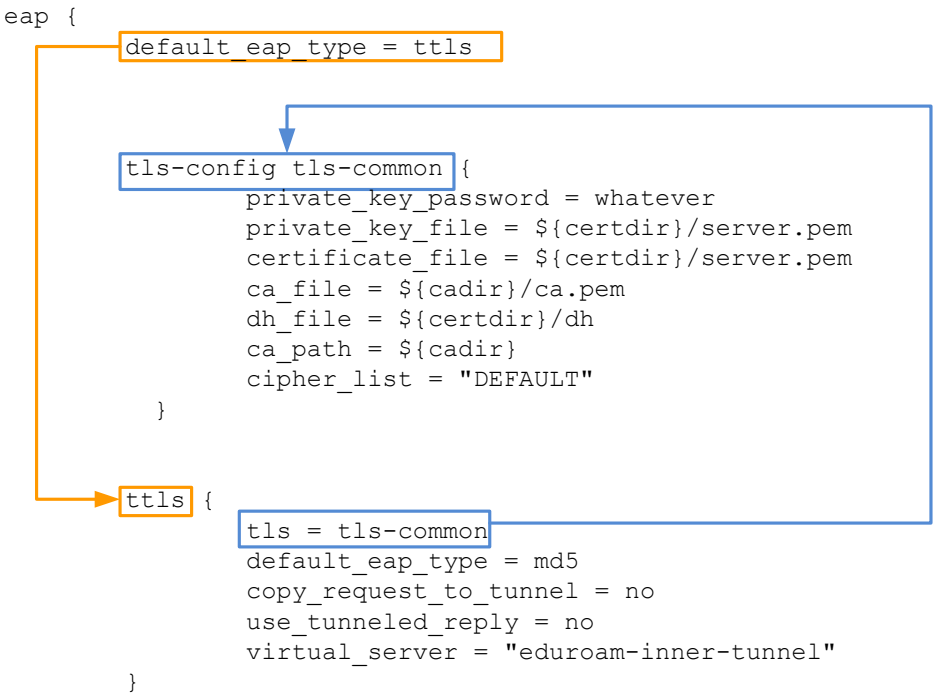
```
post-auth {
        reply_log
        Post-Auth-Type REJECT {
           attr_filter.access_reject
            update outer.session-state {
              &Module-Failure-Message := &request:Module-Failure-Message
            }
          }
}

pre-proxy {
}

post-proxy {
        eap
}
}
```

Da bi se novi virtuelni server koristio, potrebno je napraviti soft link u okviru sites-enabled poddirektorijuma:

```
[root@radius sites-enabled]# ln -s ../sites-available/eduroam
[root@radius sites-enabled]# ln -s ../sites-available/eduroam-inner-tunnel
```

## 1.4 moduli - eap, ldap

eap - eduroam se zasniva na primeni EAP metoda autentifikacije. Da bi bilo koji EAP metod mogao da se primenjuje, na RADIUS serveru se mora postaviti odgovarajući sertifikat. Takođe, ovaj modul se koristi za definisanje metoda autentifikacije će se koristiti. Jedan server može biti konfigurisan da podržava više različitih tipova EAP-a (npr. EAP-TTLS, PEAP-MSCHAP). Najbitniji delovi eap modula su **default_eap_type** i **tls-config**. Prvi parametar se koristi za definisanje podrazumevanog tipa autentifikacije, npr. **default_eap_type = ttls**. Ovo znači da će se ostali parametri potrebni za uspostavu TTLS tunela čitati iz **ttls** sekcije. U ttls sekciji se nalazi parametar **tls**. Ovaj parametar se koristi kako bi se definisala **tls-config** sekcija iz koje se čitaju konfiguracioni parametri neophodni za uspostavljanje EAP tunela, bez obzira da li se koristi TTLS ili PEAP. Zbog organizacije eap modula, uvek se prvo definiše **tls-config** sekcija.




U okviru ove sekcije se podešavaju svi parametri koji se odnose na serverski sertifikat: sam sertifikat sa celim lancem poverenja i privatni ključ. Ovaj sertifikat server predstavlja korisničkom uređaju prilikom pokušaja autentifikacije. Ukoliko na korisničkom uređaju ne postoji ovaj sertifikat, povezivanje na eduroam nije moguće, tj. neće se uspostaviti siguran tunel između korisničkog uređaja i RADIUS servera. Ubacivanje sertifikata u listu root sertifikacionih tela kojima se veruje je komplikovanije u slučaju da se koriste samopotpisani sertifikati, pa je stoga preporuka da se koristi eduroam CAT alat. Primer konfigurisanog eap modula, bez linija koje počinju znakom „#", dat je u nastavku.

```
eap {
        default_eap_type = ttls
        timer_expire     = 60
        ignore_unknown_eap_types = no
        cisco_accounting_username_bug = no
        max_sessions = ${max_requests}
        md5 {
        }
        leap {
        }
        gtc {
                auth_type = PAP
        }

        tls-config tls-common {
                private_key_password = whatever
                private_key_file = ${certdir}/server.pem
```

```
                certificate_file = ${certdir}/server.pem
                ca_file = ${cadir}/ca.pem
                dh_file = ${certdir}/dh
                ca_path = ${cadir}
                cipher_list = "DEFAULT"
        }
        tls {
                tls = tls-common
        }
        ttls {
                tls = tls-common
                default_eap_type = md5
                copy_request_to_tunnel = no
                use_tunneled_reply = no
                virtual_server = "eduroam-inner-tunnel"
        }
        peap {
                tls = tls-common
                default_eap_type = mschapv2
                copy_request_to_tunnel = no
                use_tunneled_reply = no
                virtual_server = "inner-tunnel"
        }
        mschapv2 {
        }
}
```

U okviru ttls sekcije, može se videti da se za EAP-TTLS koristi prethodno kreiran **eduroam-inner-tunnel** virtuelni server.

## 2 LDAP konfiguracija

LDAP konfiguracija - svi konfiguracioni fajlovi i LDAP šeme se nalaze u okviru /usr/local/etc/openldap direktorijuma. Konfiguracioni fajl koji se koristi za definisanje šema koje će se koristiti, LDAP ACL, domena i rootdn (superadmin) naloga je **slapd.conf**. **Rootdn nalog** je veoma značajan i bez obzira na definisane ACL, **uvek ima read/write privilegije nad celim LDAP stablom**. Nakon instalacije, LDAP direktorijum je prazan, tj. nema definisano stablo i korisničke naloge. U okviru /usr/local/etc/openldap direktorijuma definisan je ldif fajl sa imenom **inst.ac.rs.ldif**. U ovom fajlu je potrebno napraviti izmene tako da ime fajla i njegov sadržaj odgovaraju zvaničnom domenu institucije. Kada se naprave izmene, iste izmene se moraju napraviti i u slapd.conf fajlu (osenčene linije) čiji je izgled dat u nastavku:

```
    include         /usr/local/etc/openldap/schema/core.schema
    include         /usr/local/etc/openldap/schema/cosine.schema
```

```
include         /usr/local/etc/openldap/schema/inetorgperson.schema
include         /usr/local/etc/openldap/schema/nis.schema
pidfile         /usr/local/var/run/slapd.pid
argsfile        /usr/local/var/run/slapd.args
database        mdb
maxsize         1073741824
suffix          "dc=inst,dc=ac,dc=rs"
rootdn          "cn=Manager,dc=inst,dc=ac,dc=rs"
rootpw          secret
directory       /usr/local/var/openldap-data
index   objectClass     eq
```

Fajl je prikazan bez linija koje počinju znakom „#". Ovo takođe znači da nisu definisane ACL za pristup LDAP direktorijumu. U slučaju loše definisane politike pristupa samom RADIUS serveru ili loše definisanih ACL na mrežnim uređajima, ovo može predstavljati problem. Zbog toga je preporuka da se definišu i LDAP ACL za svaki servis koji će koristiti isti LDAP za autentifikaciju. Na ovaj način se omogućava ne samo da se zaštiti LDAP prema svakom servisu ponaosob, već se može ograničiti i pristup pojedinim granama u LDAP stablu koje su od većeg značaja. Primer LDAP ACL je dat u nastavku.

```
access to dn.subtree="ou=People,dc=inst,dc=ac,dc=rs"
                    by dn.exact="cn=radius,ou=SystemAccounts,dc=inst,
dc=ac,dc=rs" read
                    by
dn.exact="cn=sync,ou=SystemAccounts,dc=inst,dc=ac,dc=rs" write
                    by self write
                    by users none
                    by * auth
```

Kada su svi parametri u slapd.conf fajlu ispravno podešeni, može se preći na ubacivanje inicijalnog inst.ac.rs.ldif fajla u LDAP, komandom:

```
ldapadd -f inst.ac.rs.ldif -D rootdn -w password
```

Osenčeni su parametri koje je potrebno promeniti. Inicijalni fajl koji je prethodno kreiran i koji se ubacuje u LDAP mora imati ***.ldif*** ekstenziju. Parametar **rootdn** odgovara superadmin nalogu iz slapd.conf fajla, a parametar **password** je lozinka za ovaj nalog. Na ovaj način će biti kreirano LDAP stablo sa korenom čije ime će odgovarati domenu institucije, sa jednom granom (People) i jednim korisničkim nalogom (test). U koren stabla se naknadno mogu dodati i nove grane, bilo iz komandne linije, bilo uz korišćenje Apache Directory Studio softvera.

Poslednji korak u osposobljavanju RADIUS servera da bude potpuno funkcionalan je uspostavljanje komunikacije između FreeRADIUS ldap modula i samog LDAP direktorijuma. Kao i u slučaju eap modula, ldap modul se takođe nalazi u mods-available poddirektorijmu. U nastavku je dat izgled početnog fajla, bez izmena. Parametri koje je potrebno promeniti su osenčeni.

```
ldap {
        server = 'localhost'
        identity = 'cn=admin,dc=example,dc=org'
```

```
        password = mypass
        base_dn = 'dc=example,dc=org'
        sasl {
        }
        update {
                control:Password-With-Header    += 'userPassword'
                control: += 'radiusControlAttribute'
                request: += 'radiusRequestAttribute'
                reply: += 'radiusReplyAttribute'
        }
        user {
                base_dn = "${..base_dn}"
                filter = "(uid=%{%{Stripped-User-Name}:-%{User-Name}})"
                sasl {
                }
        }
        group {
                base_dn = "${..base_dn}"
                filter = '(objectClass=posixGroup)'
                membership_attribute = 'memberOf'
        }
        profile {
        }
        client {
                base_dn = "${..base_dn}"
                filter = '(objectClass=radiusClient)'
                template {
                }
                attribute {
                        ipaddr = 'radiusClientIdentifier'
                        secret = 'radiusClientSecret'
                }
        }
        accounting {
                reference = "%{tolower:type.%{Acct-Status-Type}}"
                type {
                        start {
                                update {
```

```
                                        description := "Online at %S"
                                }
                        }
                        interim-update {
                                update {
                                        description := "Last seen at %S"
                                }
                        }
                        stop {
                                update {
                                        description := "Offline at %S"
                                }
                        }
                }
        }

        post-auth {
                update {
                        description := "Authenticated at %S"
                }
        }

        options {
                chase_referrals = yes
                rebind = yes
                res_timeout = 10
                srv_timelimit = 3
                net_timeout = 1
                idle = 60
                probes = 3
                interval = 3
                ldap_debug = 0x0028
        }
        tls {
        }
        pool {
                start = ${thread[pool].start_servers}
                min = ${thread[pool].min_spare_servers}
```

```
                max = ${thread[pool].max_servers}
                spare = ${thread[pool].max_spare_servers}
                uses = 0
                retry_delay = 30
                lifetime = 0
                idle_timeout = 60
        }
}
```

Dakle, **server** parametar se koristi kako bi se definisala lokacija LDAP direktorijuma iz koga će se čitati korisnička imena i lozinke. Ovaj parametar može biti ili IP adresa ili DNS ime servera. U ovom slučaju, potrebno je ostaviti vrednost localhost, jer je LDAP proces pokrenut na istom serveru kao i FreeRADIUS. Parametri **identity**, **password** i **base_dn** odgovaraju parametrima iz LDAP direktorijuma koji je konfigurisan u prethodnom koraku: nalog sa kojim se FreeRADIUS bind-uje na LDAP, lozinka za taj nalog i grana u kojoj će FreeRADIUS početi da traži korisničko ime prilikom autentifikacije, respektivno. Nije preporučljivo da se za bind-ovanje koristi rootdn nalog, pa je bolje imati već konfigurisan nalog u npr. SystemAccounts grani. Vrednost base_dn parametra se takođe razlikuje u zavisnosti od toga kako je konfigurisan LDAP, pa njegova vrednost može biti npr. **base_dn = 'ou=People,dc=inst,dc=ac,dc=rs'**. Sekcija **update** se koristi iz dva razloga:

- radi mapiranja para kontrolnih atributa iz LDAP-a koji odgovaraju paru korisničko ime/lozinka u odgovarajuće RADIUS atribute User-Name i User-Password,
- radi definisanja skupa atributa koji će nakon uspešne autentifikacije biti vraćeni u okviru RADIUS paketa nekom servisu (npr. Filesender).

U **update** sekciju je stoga potrebno dodati parametar koji će odgovarati korisničkom imenu, sa prefiksom **control**.

Ukoliko se koriste tzv. Vendor Specific Attributes (VSA), tada se mora napraviti poseban **dictionary** fajl u kome će biti definisani svi atributi koji su potrebni da bi korisnici mogli da pristupaju nekom servisu. Ovo je neophodno kako bi FreeRADIUS mogao da „razume" VSA atribute. Svi VSA atributi se smeštaju u jedan **dictionary** fajl, čije ime bi trebalo da odgovara vendoru, pa je tako npr. za AMRES potrebno napraviti fajl **dictionary.amres**, a zatim ovaj fajl uključiti dodavanjem sledeće linije na kraj glavnog **dictionary** fajla:

```
$INCLUDE dictionary.amres
```

Ovaj postupak je potrebno ponoviti za svaki VSA dictionary koji se dodaje. Primer AMRES VSA dictionary fajla je dat u nastavku.

```
VENDOR          AMRES           11067

BEGIN-VENDOR AMRES

ATTRIBUTE       AMRES-Attribute-sn          1       string
ATTRIBUTE       AMRES-Attribute-gn          2       string
ATTRIBUTE       AMRES-Attribute-uid         3       string
ATTRIBUTE       AMRES-Attribute-cn          4       string
ATTRIBUTE       AMRES-Attribute-mail        5       string
ATTRIBUTE       AMRES-Attribute-o           6       string
```

```
ATTRIBUTE       AMRES-Attribute-entitlement  7       string
ATTRIBUTE       AMRES-Attribute-displayName  8       string
ATTRIBUTE       AMRES-Attribute-Affiliation  9       string

END-VENDOR AMRES
```

Ukoliko je potrebno da se servisu pošalje ime organizacije iz koje korisnik dolazi, tada takav atribut mora imati prefiks reply, npr: **reply:AMRES-Attribute-o := 'o'**. To znači da RADIUS VSA atributu AMRES-Attribute-o odgovara LDAP atribut „o".

Poslednja sekcija u ldap modulu kojoj je potrebno posvetiti pažnju je **user** sekcija. Parametar **base_dn** je isti kao i na početku ldap modula. Vrednost ovog parametra se može razlikovati ukoliko su korisnički nalozi smešteni u drugu granu. Još jedan parametar kome je potrebno posvetiti pažnju je **filter**. Ovaj parametar je namenjen kako bi omogućio bržu pretragu LDAP direktorijuma, odnosno odgovarajuće grane u LDAP-u u kojoj se nalaze korisnički nalozi. U većini slučajeva, korisničko ime se u LDAP-u nalazi smešteno u atributu **uid**. Korisničko ime može biti bez domena, pa se zbog toga koristi vrednost **Stripped-User-Name**, ili sa domenom čija vrednost je u tom slučaju **User-Name**. Stoga parametar **filter = "(uid=%{%{Stripped-User-Name}:-%{User-Name}})"** znači da će FreeRADIUS prvo pokušati da kao kontrolni RADIUS atribut koristi korisničko ime bez domena (npr. marko), a ako u tome ne uspe, pokušaće sa proverom da li postoji odogovarajuće korisničko ime sa domenom (npr. marko@inst.ac.rs).

Kada završite sa izmenom fajla, potrebno je izaći iz njega i snimiti sve promene. Na ovaj način je završena konfiguracija RADIUS servera i LDAP direktorijuma. Bilo koja promena u konfiguraciji podrazumeva da se RADIUS proces mora zaustaviti i ponovo pokrenuti. Najbolji nači da se utvrdi da li se konfiguracija učitala bez problema i da li možda postoje neka upozorenja je da se RADIUS server prvo pokrene u ***debug*** modu. U komandnoj liniji je potrebno uneti:

```
radiusd -X
```

Ukoliko se konfiguracija učitala bez problema, poslednja linija koja se ispisuje je:

```
Ready to process requests
```

Nakon toga, prilikom inicijalne konfiguracije, preporučljivo je da se testira autentifikacija. Ovo je moguće uraditi na dva načina:

- primenom **eapol_test** alata i
- povezivanjem preko AP-a koji se koristi u okviru eduroam servisa.

Za oba testa je potrebno prethodno napraviti test korisnika u LDAP direktorijmu koji je kreiran. Alat koji omogućava testiranje iz komandne linije i bez povezivanja na AP-ove je **eapol_test**. Ovaj alat simulira vezu AP/server, tj. predstavlja se serveru kao klijent koji je u stvari AP. Komanda koja omogućava testiranje je:

```
eapol_test -c ttls-pap.conf -s testing123
```

pri čemu ***-c*** definiše odakle će se čitati parametri korisničkog naloga, a ***-s*** predstavlja **shared_secret** lozinku za klijenta. Ovaj alat je već instaliran na serveru, pa je potrebno samo izmeniti parametre u **ttls-pap.conf** fajlu (/opt/ttls-pap.conf). Izgleda ovog fajla je dat u nastavku, a osenčene linije predstavljaju onaj deo koji je potrebno izmeniti tako da odgovaraju test nalogu iz LDAP-a. Vrednosti parametra **anonymous_identity** treba samo dodati domen vaše institucije.

```
network={
        ssid="example"
        key_mgmt=WPA-EAP
```

```
        eap=TTLS
        identity="bob"
        anonymous_identity="anonymous"
        password="hello"
        phase2="auth=PAP"
}
```

Pre pokretanja eapol_test komande, otvorite drugi Putty prozor u kom ćete pokrenuti FreeRADIUS u debug modu. Da li je pokušaj autentifikacije bio uspešan, može se utvrditi posmatranjem ispisa u oba prozora. U prozoru u kom je pokrenut eapol_test, ispisuju se sledeće linije na kraju uspešne autentifikacije:

```
MPPE keys OK: 1  mismatch: 0
SUCCESS
```

U isto vreme, u drugom otvorenom prozoru gde je FreeRADIUS pokrenut u debug modu, ispisuju se, između ostalog, sledeće linije:

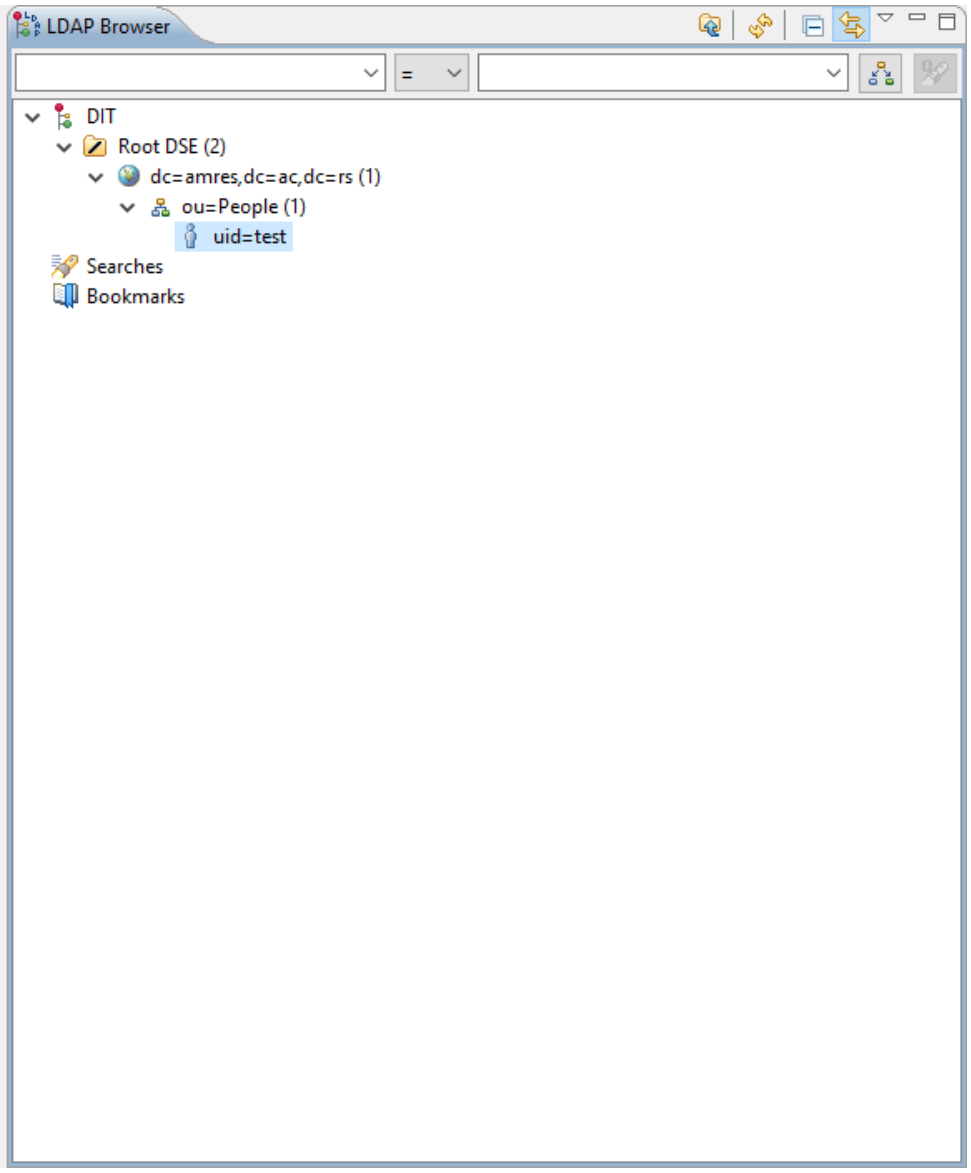
```
(6) eap_ttls: Got tunneled Access-Accept
(6) eap_ttls: No information to cache: session caching will be disabled for
session f9a223792ea99a6b2f34eb2b33ef1cc8e16f418e6c0fb05334c19bcb2af66a4a
(6) eap: Sending EAP Success (code 3) ID 6 length 4
.
.
.
(6) Sent Access-Accept Id 6 from 127.0.0.1:1812 to 127.0.0.1:47362 length 0
(6)   MS-MPPE-Recv-Key =
0x54a9906be2c1f5417f0dba474f8c2eaaf58dbd34a3c5db8e55644caa68833114
(6)   MS-MPPE-Send-Key =
0x454348a4289ec6161f2a702b66f86fd424c5c032134ce7fafb902ebb8b25c69b
(6)   EAP-Message = 0x03060004
(6)   Message-Authenticator = 0x00000000000000000000000000000000
(6)   User-Name = "anonymous@amres.ac.rs"
(6) Finished request
Waking up in 4.9 seconds.
(0) Cleaning up request packet ID 0 with timestamp +2
(1) Cleaning up request packet ID 1 with timestamp +2
(2) Cleaning up request packet ID 2 with timestamp +2
(3) Cleaning up request packet ID 3 with timestamp +2
(4) Cleaning up request packet ID 4 with timestamp +2
(5) Cleaning up request packet ID 5 with timestamp +2
(6) Cleaning up request packet ID 6 with timestamp +2
Ready to process requests
```

Krajnji ispis za uspešnu autentifikaciju je isti bez obzira da li se testiranje radi sa korisnicima definisanim u okviru tekstualnog fajla ili se koristi LDAP direktorijum. U okviru **users** fajla na početak je potrebno dodati novog test korisnika sa sledećim parametrima:

```
test Password-With-Header := "{SHA}cojt0Pw//L6ToM8G41aOKFIWh7w="
```

Razlika u ispisu se javlja u zavisnosti od toga da li se koristi tekstualni fajl ili se koristi neki eksterni izvor autentifikacije (SQL-like baza, AD, LDAP). U okviru eduroam-inner-tunnel fajla, u authorize sekciji, potrebno je zakomentarisati liniju **files**, a otkomentarisati liniju **ldap**. Ovo znači da će sada RADIUS server upoređivati korisničko ime i lozinku sa podacima koji se nalaze u LDAP direktorijumu. Zbog toga se i razlikuje ispis koji se generiše prilikom autentifikacije kada je FreeRADIUS pokrenut u **debug** modu.

Dodavanje novog korisnika u LDAP može biti nezgrapno ukoliko se za to koristi komandna linija. Zbog toga je najbolje koristiti npr. Apache Directory Studio softver. Izgled početnog LDAP stabla sa People granom i test korisnikom se može videti na slici.




Novi korisnik se dodaje u nekoliko koraka:

1. Desnim klikom na **uid=test** se otvara novi meni iz koga je potrebno odabrati opciju „**Copy Entry/DN**"

2. Desni klik na People granu daje, između ostalih, i opciju „**Paste Entry**"
3. Odabrati opciju „**Rename entry and continue**", a u delu **RDN: uid = test**, potrebno je dodeliti novu vrednost, npr. **uid = test2** i kliknuti na **dugme „OK"**

Nakon koraka 3 se u stablu može videti novi korisnik, odnosno uid koji je prethodno kreiran.