



Konfiguracija FreeRADIUS servera za davanje resursa

Sadržaj

Uvod	3
1. Konfiguracija klijenata RADIUS servera - <i>clients.conf</i>	4
2. Konfiguracija prosleđivanja RADIUS zahteva - <i>proxy.conf</i>	6
3. Aktiviranje logovanja RADIUS zahteva	7
4. Podešavanje RADIUS Accounting-a	8
5. <i>policy.conf</i>	11

Uvod

Ovo uputstvo prikazuje dodatnu konfiguraciju relevantnih FreeRADIUS fajlova, koja je neophodna u slučaju kada je institucija davalac resursa. Konfiguracioni fajlovi su prikazani bez komentara, u cilju jasnijeg i preglednijeg prikaza.

1. Konfiguracija klijenata RADIUS servera - *clients.conf*

Davaoci resursa imaju svoje mrežne uređaje (NAS - *Network Access Server*) koje koriste kao pristupnu infrastrukturu za eduroam servis. NAS uređaji mogu biti *access point*-i, WLC kontroleri, ili svičevi koji podržavaju 802.1x standard. Oni se moraju podesiti tako da vrše kontrolu pristupa korisnika, što obuhvata aktiviranje 802.1x protokola, konfigurisanje autentifikacije korišćenjem Radius servera i sl. (nije predmet ovog uputstva).

Na FreeRADIUS-u je potrebno definisati NAS uređaje kao klijente RADIUS servera, što se konfigurira u *raddb/clients.conf* konfiguracionom fajlu. Pored osnovne konfiguracije (za davaoca identiteta) u kojoj se definišu AMRES FTLR serveri i NetIIS server za nadgledanje, potrebno je dodati klijenta za svaki NAS uređaj. Primer konfiguracije jednog klijenta dat je u nastavku:

```
client NAS-naziv {
    ipaddr          = 1.1.1.1      # IP adresa NAS-a
    secret          = lozinka      # lozinka podešena na NAS-u ka radius-u
    shortname       = AP-br        # skraćeno ime NAS-a
    nastype         = other
    virtual_server  = eduroam
}
```

Za svaki NAS uređaj je potrebno napraviti poseban *client* ulaz u *clients.conf* fajlu. Ukoliko se koristi WLC kontroler sa *lightweight access point*-ima, tada se konfigurira samo jedan klijent koji predstavlja kontroler.

Nakon dodavanja novih klijenata, *clients.conf* fajl bi trebao da izgleda ovako (sa nekoliko *access point*-a navedenih kao primer - osenčene linije):

```
## 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  
}  
client AP-kancelarija {  
    ipaddr          = 1.1.1.1  
    secret          = lozinka1  
    shortname       = AP1  
    nastype         = other  
    virtual_server  = eduroam  
}  
client AP-hodnik {  
    ipaddr          = 1.1.1.2  
    secret          = lozinka2  
    shortname       = AP2  
    nastype         = other  
    virtual_server  = eduroam  
}  
client AP-biblioteka {  
    ipaddr          = 1.1.1.3  
    secret          = lozinka3  
    shortname       = AP3  
    nastype         = other  
    virtual_server  = eduroam  
}
```

2. Konfiguracija prosleđivanja RADIUS zahteva - *proxy.conf*

U *raddb/proxy.conf* fajlu podešava se način prosleđivanja zahteva koji se ne obrađuju lokalno (zahtevi za autentifikaciju korisnika koji nisu sa te institucije). Potrebno je definisati AMRES FTLR servere i *default* domen što će omogućiti prosleđivanje zahteva koji nisu lokalni prema AMRES FTLR serverima. Takođe, potrebno je opciju *default_fallback* podesiti na *yes*. Primer *proxy.conf* fajla, gde su izmene koje je potrebno uneti osenčene, dat je u nastavku:

```
proxy server {
    default_fallback = yes
}
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 inst.ac.rs {
    authhost          = LOCAL
    accthost          = LOCAL
    User-Name         = "%{Stripped-User-Name}"
}
home_server amres.eduroam.ftlr1 {
    type = auth+acct
    ipaddr = 147.91.4.204
    port = 1812
    secret = pass #ista kao u clients.conf fajlu
    response_window = 20
    zombie_period = 40
    revive_interval = 120
    status_check = status-server
    check_interval = 30
    num_answers_to_alive = 3
}
home_server amres.eduroam.ftlr2 {
    type = auth+acct
    ipaddr = 147.91.1.101
    port = 1812
    secret = pass #ista kao u clients.conf fajlu
    response_window = 20
    zombie_period = 40
    revive_interval = 120
    status_check = status-server
    check_interval = 30
    num_answers_to_alive = 3
}
home_server pool amres.eduroam {
```

```
home_server = amres.eduroam.ftlr1
home_server = amres.eduroam.ftlr2
}
realm DEFAULT {
    pool = amres.eduroam
    nostrip
}
```

3. Aktiviranje logovanja RADIUS zahteva

FreeRADIUS platforma omogućava dve vrste logovanja:

- Sažeto logovanje u jedan fajl `/usr/local/var/log/radius/radius.log`. Beleži se prikaz rezultata autentifikacije koji uključuje: datum i vreme pristizanja zahteva, rezultat autentifikacije (*OK* ili *reject*), spoljni identitet korisnika, *shortname* klijenta od kog je stigao zahtev, kao i MAC adresu korisnika. Primer jednog log zapisa uspele autentifikacije:

```
Wed Jun 1 12:24:05 2011 : Auth: Login OK: [anonymous@rcub.bg.ac.rs] (from
client AP1 port 1 cli b4-07-f9-74-75-f0)
```

Ovaj način logovanja se aktivira u `radius.conf` konfiguracionom fajlu, tako što se vrednost parametra `auth` postavi na `yes` (osencena linija):

```
log {
.
.
# Log authentication requests to the log file.
#
# allowed values: {no, yes}
# default value = no
#
auth = yes
```

- Detaljno logovanje poruka u više fajlova. Beleže se sve RADIUS poruke tokom procesa autentifikacije korisnika i proksiranja zahteva. Ovi logovi se po osnovom podešavanju beleže u poseban folder za svaki klijent koji upućuje zahtev RADIUS serveru (klijent može biti NAS ili FTLR server), gde ime foldera odgovara IP adresi klijenta. Tako će npr., sve autentifikacione poruke između vašeg RADIUS-a i AMRES FTLR1 servera biti zabeležene u folder `/usr/local/var/log/radius/radacct/147.91.4.204`. U folderu za svaki NAS se nalaze sledeći log fajlovi:
 - ***auth-detail-datum*** - log pristiglih RADIUS autentifikacionih zahteva (aktiviran prilikom konfiguracije FreeRADIUS servera za potrebe davanja identiteta)
 - ***reply-detail-datum*** - log poslanih RADIUS autentifikacionih odgovora (aktiviran prilikom konfiguracije FreeRADIUS servera za potrebe davanja identiteta)
 - ***pre-proxy-detail-datum*** - log autentifikacionih zahteva ili odgovora pre nego što se proslede drugom RADIUS serveru

- **post-proxy-detail-datum** - log autentifikacionih zahteva ili odgovora nakon prosleđivanja drugom RADIUS serveru

Prva dva log fajla su aktivirana prilikom podešavanja za davaoca identiteta, dok je sada potrebno aktivirati preostala dva. U *raddb/sites-available/eduroam* fajlu potrebno je:

- u *pre-proxy* sekciji odkomentarisati *pre_proxy_log* liniju:

```
pre-proxy {  
  
    # If you want to have a log of packets proxied to a home  
    # server, un-comment the following line, and the  
    # 'detail pre_proxy_log' section, above.  
    pre_proxy_log
```

- u *post-proxy* sekciji odkomentarisati *post_proxy_log* liniju:

```
post-proxy {  
  
    # If you want to have a log of replies from a home server,  
    # un-comment the following line, and the 'detail post_proxy_log'  
    # section, above.  
    post_proxy_log
```

4. Podešavanje RADIUS Accounting-a

Davalac resursa može dodatno da prati aktivnosti korisnika eduroam servisa korišćenjem RADIUS *Accounting*-a. RADIUS *Accounting* se aktivira na NASu, tako da informacije o korisničkoj sesiji (putem RADIUS *Accounting* paketa) NAS šalje RADIUS serveru, koji zatim te podatke na adekvatan način beleži. Tako se npr. može beležiti IP i MAC adresa korisnika, vreme trajanja korisničke sesije, količina prenetih podataka itd.

Na RADIUS serveru potrebno je konfigurisati na koji način se informacije dobijene putem RADIUS *Accounting* paketa čuvaju. Uobičajena praksa koja omogućuje lak pregled informacija, je da se upis vrši u SQL bazu. U ovom uputstvu, dat je primer podešavanja RADIUS *Accounting*a u MySQL bazu koja se nalazi na istom serveru.

U nastavku je prikazana instalacija MySQL baze i startovanja MySQL servisa. Najoptimalnije je instalirati MySQL pre instaliranja FreeRADIUS platforme, kao što je prikazano u uputstvu za instalaciju FreeRADIUS-a. U slučaju da se MySQL baza naknadno instalira, potrebno je prekompajlirati instalaciju FreeRADIUS-a.

```
yum install mysql-server mysql php-mysql  
chkconfig --levels 235 mysqld on  
service mysqld start
```

Zatim je potrebno ulogovati se kao root na bazu i konfigurisati novu root lozinku (umesto "lozinka" uneti željenu lozinku):

```
mysql -u root  
SET PASSWORD FOR 'root'@'localhost' = PASSWORD('lozinka');  
SET PASSWORD FOR 'root'@'127.0.0.1' = PASSWORD('lozinka');
```


Nakon toga potrebno je kreirati novog korisnika koji će FreeRADIUS koristiti da upisuje podatke u MySQL bazu (ovde je prikazano kreiranje novog korisnika - „marko“ sa šifrom „b1gS3cRet“ - naravno, potrebno je promeniti ime korisnika i podesiti drugačiju lozinku).

```
create user 'marko'@'localhost' identified by 'b1gS3cRet';
GRANT ALL PRIVILEGES ON *.* TO 'marko'@'localhost';
GRANT ALL PRIVILEGES ON *.* TO 'marko'@'%';
GRANT ALL ON *.* TO 'marko'@'%' IDENTIFIED BY 'b1gS3cRet';
```

Sada je potrebno kreirati novu bazu podataka, u ovom primeru se ona naziva *radius*.

```
Create database radius;
```

Zatim je potrebno pristupiti bazi podataka i kreirati novu tabelu koja će se koristiti za RADIUS *Accounting*. U narednom primeru kreira se tabela sa nazivom *eduroam-acct*:

```
use radius;
DROP TABLE IF EXISTS `eduroam_acct`;
SET @saved_cs_client      = @@character_set_client;
SET character_set_client = utf8;
CREATE TABLE `eduroam_acct` (
  `User-Name` varchar(100) NOT NULL default '',
  `Calling-Station-Id` varchar(100) NOT NULL default '',
  `Client-IP-Address` varchar(100) NOT NULL default '',
  `Called-Station-Id` varchar(100) NOT NULL default '',
  `NAS-IP-Address` varchar(100) NOT NULL default '',
  `NAS-Port` int(10) unsigned NOT NULL default '0',
  `Timestamp Start` datetime NOT NULL default '1970-01-01 01:00:00',
  `Timestamp Dhcp` datetime NOT NULL default '1970-01-01 01:00:00',
  `Timestamp Stop` datetime NOT NULL default '1970-01-01 01:00:00',
  `Acct-Unique-Session-Id` varchar(100) NOT NULL default '',
  `Acct-Session-Time` int(10) unsigned NOT NULL default '0',
  `Acct-Input-Octets` int(10) unsigned NOT NULL default '0',
  `Acct-Output-Octets` int(10) unsigned NOT NULL default '0',
  `Acct-Input-Packets` int(10) unsigned NOT NULL default '0',
  `Acct-Output-Packets` int(10) unsigned NOT NULL default '0',
  `Acct-Terminate-Cause` varchar(100) NOT NULL default '',
  PRIMARY KEY (`Acct-Unique-Session-Id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
SET character set client = @saved_cs_client;
```

Ovim je završeno kreiranje MySQL baze i tabelle u koju će se upisivati RADIUS *Accounting* informacije.

Zatim je potrebno podesiti FreeRADIUS tako da koristi kreiranu MySQL tabelu za RADIUS *Accounting*. Sledeće linije je potrebno dodati u *raddb/sql.conf* fajl (osenčene linije je potrebno promeniti tako da odgovaraju parametrima koje ste prethodno konfigurisali; u poslednjoj liniji upisati *eduroam.conf*):

```
sql eduroam {
    database = "mysql"
    driver = "rlm_sql_${database}"
    server = "localhost"
```

```
login = "marko"  
password = "blgS3cRet"  
radius_db = "radius"  
acct_table1 = "eduroam_acct"  
acct_table2 = "eduroam_acct"  
postauth_table = "radpostauth"  
authcheck_table = "radcheck"  
authreply_table = "radreply"  
groupcheck_table = "radgroupcheck"  
groupreply_table = "radgroupreply"  
usergroup_table = "radusergroup"  
deletestalesessions = yes  
sqltrace = yes  
sqltracefile = ${logdir}/sqltrace.sql  
num_sql_socks = 5  
connect_failure_retry_delay = 60  
lifetime = 0  
max_queries = 0  
nas_table = "nas"  
$INCLUDE sql/${database}/eduroam.conf  
}
```

Zatim, u folderu `/raddb/sql/mysql/` kreirati novi fajl sa imenom `eduroam.conf` i u njega uneti sledeće linije (osenečene linije je potrebno promeniti tako da odgovaju nazivima tabela koje ste kreirali, ukoliko se razlikuju):

```
accounting_start_query = „INSERT INTO eduroam_acct\  
    (`User-Name`, `Calling-Station-Id`, `Called-Station-Id`,\  
    `NAS-IP-Address`, `NAS-Port`, `Timestamp Start`, \  
    `Acct-Unique-Session-Id`, `Client-IP-Address`)\  
VALUES\  
    ('${User-Name}', '${Calling-Station-Id}', '${Called-Station-Id}',\  
    '${NAS-IP-Address}', '${NAS-Port}', '%S',\  
    '${Acct-Unique-Session-Id}', '${Framed-IP-Address}')“  
  
accounting_update_query = „UPDATE eduroam_acct SET\  
    `Acct-Session-Time` = '${Acct-Session-Time}',\  
    `Acct-Input-Octets` = '${Acct-Input-Octets}',\  
    `Acct-Output-Octets` = '${Acct-Output-Octets}',\  
    `Acct-Input-Packets` = '${Acct-Input-Packets}',\  
    `Acct-Output-Packets` = '${Acct-Output-Packets}'\  
WHERE `Acct-Unique-Session-Id` = '${Acct-Unique-Session-Id}'\  
LIMIT 1  
“  
accounting_stop_query = „UPDATE eduroam_acct SET\  
    `Timestamp Stop` = '%S',\  
    `Acct-Session-Time` = '${Acct-Session-Time}',\  
    `Acct-Input-Octets` = '${Acct-Input-Octets}',\  
    `Acct-Output-Octets` = '${Acct-Output-Octets}',\  
    `Acct-Input-Packets` = '${Acct-Input-Packets}',\  
    `Acct-Output-Packets` = '${Acct-Output-Packets}',\  
    `Acct-Terminate-Cause` = '${Acct-Terminate-Cause}',\  
    `Client-IP-Address` = '${Framed-IP-Address}'\  
WHERE `Acct-Unique-Session-Id` = '${Acct-Unique-Session-Id}'\  
LIMIT 1  
“
```

Nakon toga, potrebno je u `/raddb/sites-available/eduroam` fajlu, u *accounting* sekciji uneti:

```
accounting {  
  
    .  
    .  
    .  
  
    # Log traffic to an SQL database.  
    #  
    # See "Accounting queries" in sql.conf  
    #      sql  
    eduroam
```

Takođe, u `/raddb/radiusd.conf` fajlu potrebno je odkomentarisati liniju `$INCLUDE sql.conf`:

```
# Include another file that has the SQL-related configuration  
# This is another file only because it tends to be big.  
#  
$INCLUDE sql.conf
```

Ovim je završeno podešavanje RADIUS *Accounting*-a.

5. policy.conf

Ako dođe do sigurnosnog incidenta u eduroam servisu, u najvećem broju slučajeva Davalac Resursa u svojim logovima može da pronađe samo anonimni identitet korisnika (`anonymous@idp.ac.rs`). U tom slučaju, dok se dotični korisnik ne identifikuje u bazi Davaoca Identiteta, Davalac Resursa jedino može da blokira ceo domen problematičnog korisnika.

Rešenje ovog problema se postiže korišćenjem Radius atributa:

- **CUI** (Chargeable User Identity) predstavlja jedinstveni identifikator svakog eduroam korisnika, formira ga Davalac Identiteta i šalje Davaocu Resursa.
- **ON** (Operator-Name) je jedinstveni identifikator Davaoca Resursa.

Ako je autentifikacija uspešna, Davalac Identiteta formira CUI atribut formiranjem MD5 hash-a koristeći UID korisnika, Operator-Name atribut iz zahteva i opciono ključa (u ovom primeru `cui_hash_key`). Ova vrednost se u Access-Accept poruci vraća Davaocu Resursa. Bitno je napomenuti da će vrednosti CUI atributa biti različita za istog korisnika koji koristi eduroam resurse kod različitih Resurs Provajdera.

`policy.conf` se nalazi u `raddb` folderu i predstavlja virtuelni modul, sličan onima definisanim u *stantiate* sekciji `radiusd.conf` fajla. Funkcija definisana u `policy.conf` fajlu se može pozivati na više mesta u konfiguraciji. Ova pravila su slična *subroutine*-ama u drugim programskim jezicima, samo se ovde ne mogu pozivati rekurzivno i moraju biti definisana u odgovarajućem redosledu.

U policy.conf-u su definisane četiri funkcije koje omogućavaju uključivanje CUI atributa u konfiguraciju FreeRADIUS servera, od kojih je za Davaoca Resurs bitna tri:

- cui_pre-proxy - omogućava ubacivanje praznog CUI atributa u autentifikacioni zahtev,
- cui_updatedb - (za davaoce resursa) upisivanje CUI atributa u bazu, i
- cui_accounting - (za davaoce resursa) upisivanje CUI atributa u Accounting bazu

NAPOMENA: od verzije 2.1.10 FreeRADIUS-a, CUI je podržan u policy.conf fajlu. Proverite verziju servera (komanda radiusd -v). Ako nemate CUI definisan u policy.conf fajlu, možete *upgrade*-ovati vaš server, ili prekopirati CUI funkcije u policy.conf.

U nastavku je dat samo deo fajla koji se odnosi na definisanje CUI atributa (ovaj deo je potrebno dodati u policy.conf ukoliko ne postoji).

```
cui_pre-proxy {
    update proxy-request {
        Chargeable-User-Identity='\\000'
        Operator-Name := "%{config:modules.sql[cui].sp_operator_name}"
    }
}

cui_postauth {
    if (FreeRadius-Proxied-To == "127.0.0.1") {
        if (outer.request:Chargeable-User-Identity && (outer.request:Operator-Name) ||
!( "%{config:cui_require_operator_name}" ) ) {
            update outer.reply {
                Chargeable-User-Identity:="%{md5:%{config:cui_hash_key}%{request:User-
Name}%{outer.request:Operator-Name}:-}"
            }
        }
    }
    else {
        if (!( "%{control:Proxy-To-Realm}" ) && (Chargeable-User-Identity) && !(reply:Chargeable-
User-Identity) && (Operator-Name) || !( "%{config:cui_require_operator_name}" ) ) ) {
            update reply {
                Chargeable-User-Identity:="%{md5:%{config:cui_hash_key}%{request:User-
Name}%{Operator-Name}:-}"
            }
        }
    }
}

cui_updatedb {
    if (reply:Chargeable-User-Identity) {
        cui
    }
}

cui_accounting {
    if (!Chargeable-User-Identity) {
        update control {
            Chargeable-User-Identity = "%{cui: SELECT cui FROM cui WHERE clientipaddress =
' %{Client-IP-Address}' AND callingstationid = '%{Calling-Station-Id}' AND username =
' %{User-Name}' }"
        }
    }
}

if (control:Chargeable-User-Identity && (control:Chargeable-User-Identity != "")) {
```

```
update request {  
    Chargeable-User-Identity := "%{control:Chargeable-User-Identity}"  
}  
cui  
}
```

U nastavku je prikazano podešavanje MySQL baze. Potrebno je kreirati novu tabelu, naziva cui (raddb/sql/mysql/cui.sql):

```
use radius;  
DROP TABLE IF EXISTS `cui`;  
CREATE TABLE `cui` (  
    `clientipaddress` varchar(15) NOT NULL default '',  
    `callingstationid` varchar(50) NOT NULL default '',  
    `username` varchar(64) NOT NULL default '',  
    `cui` varchar(32) NOT NULL default '',  
    `creationdate` timestamp NOT NULL default CURRENT_TIMESTAMP,  
    `lastaccounting` timestamp NOT NULL default '0000-00-00 00:00:00',  
    PRIMARY KEY (`username`,`clientipaddress`,`callingstationid`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

Zatim je potrebno podesiti FreeRADIUS tako da koristi kreiranu MySQL tabelu za CUI Accounting. Sledeću konfiguraciju je potrebno dodati u *raddb/sql.conf* fajl (osene linije je potrebno promeniti tako da odgovaraju parametrima mysql baze; u poslednjoj liniji upisati *cui.conf*). Takođe potrebno uneti odgovarajuće ime za *sp_operator_name* atribut (za ARMUNS: „1uns.ac.rs“, ARMUK: „1kg.ac.rs“, JUNIS: „1nis.ac.rs“):

```
sql cui {  
    database = "mysql"  
    driver = "rlm_sql_${database}"  
    server = "localhost"  
    login = "user"  
    password = "secret"  
    radius_db = "database"  
    sp_operator_name = "ime_operatora"  
#  
#    sqltrace = yes  
#    sqltracefile = ${logdir}/cuitrace.sql  
    num_sql_socks = 5  
    connect_failure_retry_delay = 60  
    cui_table = "cui"  
    sql_user_name = "%{User-Name}"  
    $INCLUDE sql/${database}/cui.conf  
}
```

Zatim, u folderu */raddb/sql/mysql/* kreirati novi fajl sa imenom *cui.conf* (ukoliko već ne postoji) i u njega uneti sledeće linije:

```
postauth_query = "INSERT IGNORE INTO ${cui_table} \  
    (clientipaddress, callingstationid, username, \  
    cui, lastaccounting) \  
VALUES \  
"
```

```
('{Client-IP-Address}', '{Calling-Station-Id}', \
'{User-Name}', '{reply:Chargeable-User-Identity}', NULL) \
ON DUPLICATE KEY UPDATE \
lastaccounting='0000-00-00 00:00:00', \
cui='{reply:Chargeable-User-Identity}';

accounting_start_query = "UPDATE ${cui_table} \
SET lastaccounting = CURRENT_TIMESTAMP \
WHERE clientipaddress = '{Client-IP-Address}' \
AND callingstationid = '{Calling-Station-Id}' \
AND username = '{User-Name}' \
AND cui = '{Chargeable-User-Identity}";

accounting_update_query = "UPDATE ${cui_table} \
SET lastaccounting = CURRENT_TIMESTAMP \
WHERE clientipaddress = '{Client-IP-Address}' \
AND callingstationid = '{Calling-Station-Id}' \
AND username = '{User-Name}' \
AND cui = '{Chargeable-User-Identity}";

accounting_stop_query = "UPDATE ${cui_table} \
SET lastaccounting = CURRENT_TIMESTAMP \
WHERE clientipaddress = '{Client-IP-Address}' \
AND callingstationid = '{Calling-Station-Id}' \
AND username = '{User-Name}' \
AND cui = '{Chargeable-User-Identity}";
```

Nakon ovih podešavanja, potrebno je preći u */raddb/sites-available/eduroam* fajl, i u *accounting* sekciji dodati:

```
accounting {

    # cui_accounting reads the record form the temporary database,
    # selects the corresponding CUI value, as set cui_updatedb
    # and adds the CUI attribute to the accounting request
    # uncomment the line below if *requesting* the CUI

    cui_accounting

    .
    .
    .
}
```

Takođe, potrebno je napraviti izmene i u *post-auth* sekciji, tako što se u postojećoj konfiguraciji doda osenčena linija:

```
post-auth {

    # cui_updatedb updates the temporary database adding
    # the record containing the received CUI value to be later
    # used in accounting
    # uncomment the line below if *requesting* the CUI

    cui_updatedb

    .
    .
    .
}
```

```
}
```

RP eduroam servisa mora zahtevati CUI atribut od IdP prilikom slanja Access-Request zahteva, pa se u pre-proxy sekciji dodaje:

```
pre-proxy {  
  
    # cui_pre-proxy adds the NULL CUI value to Access-Request  
    # thus making it a Chargeable-User-Identity request  
    # uncomment the line below if *requesting* the CUI  
  
    cui_pre-proxy  
  
    .  
    .  
    .  
  
}
```

Ovim je završena konfiguracija vašeg FreeRADIUS servera. Potrebno je startovati server u *debug* modu (stopiranje servera: `killall radiusd`, startovanje u debug modu: `radiusd -X`) i videti da li je server učitao konfiguraciju, i da li sluša na definisanim portovima (1812, 1813, 1814).