



**AMPEC**

Академска мрежа Србије



## Конфигурација FreeRADIUS-а за LDAP

*Историја верзија документа*

Верзија	Датум	Иницијали аутора	Опис промене
1.0			Прва верзија овог документа

## Садржај

<b>1</b>	<b>УВОД</b>	<b>4</b>
1.1	CLIENTS.CONF	4
1.2	EDUROAM ВИРТУЕЛНИ ТУНЕЛ	5
1.3	EDUROAM-INNER-TUNNEL ВИТУЕЛНИ ТУНЕЛ	8
1.4	EAP МОДУЛ	9
1.5	PROXY.CONF	11
1.6	RADIUSD.CONF	12
1.7	ПОВЕЗИВАЊЕ RADIUS СЕРВЕРА И LDAP СЕРВЕРА	13
1.7.1	Конфигурација ldap модула	13

## 1 Увод

Ово упутство се односи на подешавање основних модула FreeRADIUS сервера за даваоца идентитета. Приказани су они модули који омогућавају читање података из Active Directory-ја. Сви модули су приказани без коментара ради јаснијег и прегледнијег приказа.

### 1.1 clients.conf

clients.conf се налази у raddb директоријуму и представља модул у коме се дефинишу RADIUS клијенти - уређаји у мрежи од којих се прихватају захтеви. Ови уређаји могу бити други RADIUS сервери или NAS (Network Access Server) тачке. Локација raddb директоријума зависи од начина на који је FreeRADIUS инсталиран. Најчешће су то /etc или /usr/local/etc директоријуми.

У оквиру clients.conf фајла је потребно дефинисати два AMRES FTLR (Federation Top Level Radius Server) сервера и још један сервер који се користи за надгледање оперативности RADIUS сервера институције. AMRES FTLR сервери представљају националне сервере који се налазе на врху AMRES eduroam хијерархије. Конфигурација која је дата у наставку може се ископирати и додати директно у clients.conf фајл. Параметри који омогућавају комуникацију са FTLR серверима се зову secret. Овај параметар се добија лично или телефонским путем од AMRES-а и њега је потребно заменити у конфигурацији.

```
## 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
    nas_type        = 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
    nas_type        = 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
    nas_type        = other
}
```

```
virtual_server    = eduroam
}
```

## 1.2 eduroam виртуелни тунел

Виртуелни тунели омогућавају конфигурацију већег броја независних сервиса на FreeRADIUS платформи. За потребе eduroam сервиса, креира се нови виртуелни тунел који ће обрађивати аутентификационе захтеве:

- ❖ прелази се у `/raddb/sites-available/` поддиректоријум
- ❖ `default` виртуелни тунел се копира у нови фајл, чије име ће бити `eduroam`:

```
cp default eduroam
```

- ❖ сада је потребно изменити креирани фајл. На самом почетку, пре `authorize` секције, уместо `default`, ставља се `eduroam`.
- ❖ у овом кораку, потребно је изменити конфигурациони фајл, тако да изгледа као у наставку (коментари су избачени преко `grep` команде):

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

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

listen {
    type = auth
    ipv6addr = ::
    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
    unix
    eap
```

```
}
preacct {
    preprocess
    acct_unique
    suffix
    files
}
accounting {
    detail
    unix
    radutmp
    exec
    attr_filter.accounting_response
}
session {
    radutmp
}
post-auth {
    exec
    reply_log
    Post-Auth-Type REJECT {
        attr_filter.access_reject
    }
}
pre-proxy {
}
post-proxy {
    eap
}
}
```

- ✦ у последњем кораку, потребно је прећи у `/raddb/sites-enabled` поддиректоријум и направити *soft* линк ка `eduroam` виртуелном тунелу из `/raddb/sites-available` поддиректоријума:

```
ln -s ../sites-available/eduroam
```

### 1.3 eduroam-inner-tunnel виртуелни тунел

Sada je potrebno formirati eduroam-inner-tunnel virtuelni server, koji u svojoj konfiguraciji poziva određene module koji su odgovorni za komunikaciju sa korisničkom bazom (u ovom slučaju AD):

- ❖ прелази се у `/raddb/sites-available/` поддиректоријум
- ❖ `inner-tunnel` виртуелни тунел се копира у нови фајл, чије име ће бити `eduroam-inner-tunnel`:

```
cp inner-tunnel eduroam-inner-tunnel
```

- ❖ у овом кораку, потребно је изменити конфигурациони фајл, тако да изгледа као у наставку (коментари су избачени преко `grep` команде):

```
server eduroam-inner-tunnel {
authorize {
    auth_log
    suffix
    update control {
        Proxy-To-Realm := LOCAL
    }
    eap
    ldap
    pap
}
authenticate {
    Auth-Type PAP {
        pap
    }
    Auth-Type CHAP {
        chap
    }
    Auth-Type MS-CHAP {
        mschap
    }
    eap
}
session {
    radutmp
}
post-auth {
    reply_log
    Post-Auth-Type REJECT {
```



```
        attr_filter.access_reject
    }
}
pre-proxy {
}
post-proxy {
    eap
}
```

- » у последњем кораку, потребно је прећи у `/raddb/sites-enabled` поддиректоријум и направити `soft` линк ка `eduroam` виртуелном тунелу из `/raddb/sites-available` поддиректоријума:

```
ln -s ../sites-available/eduroam-inner-tunnel
```

## 1.4 еар модул

У FreeRADIUS 3.0.x верзијама, сви модули су премештени у `/raddb/mods-available` поддиректоријум. Модули који могу бити од интереса су `ldap`, `ntlm_auth`, `cui`, `eap`, `sql` итд. Активирање жељених модула се постиже тако што се прави `soft` линк за тај модул у оквиру `/raddb/mods-enabled` директоријума.

`eap` модул је потребно изменити да буде исти као у наставку. Жељени метод аутентификације се подешава на самом почетку `eap` модула (`default_eap_type`). У овом упутству је дат пример за EAP-TTLS аутентификацију. Након што се у првој линији дефинише метод аутентификације, потребно је у одговарајућој секцији (`ttls` или `peap`) променити параметар `virtual_server` тако да његова вредност буде једнака `eduroam-inner-tunnel`. Уколико желите да као секундарни метод аутентификације омогућите и PEAP, тада је потребно у `peap` секцији `eap` модула такође променити вредност параметра `virtual_server` на `eduroam-inner-tunnel`.

```
eap {
    default_eap_type = ttls    # ili peap
    timer_expire     = 60
    ignore_unknown_eap_types = no
    cisco_accounting_username_bug = no
    max_sessions     = 4096
    md5 {
    }
    leap {
    }
    gtc {
        challenge = "Password: "
        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"
ecdh_curve = "prime256v1"
verify {
}

}

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 {
}

}
```

Институција је слободна да одлучи код ког СА (Certification Authority) тела ће набавити сертификат који ће се користити за EAP-TTLS или PEAP (потребно је направити измене у оквиру `tls-config` `tls-common` секције за осенчене параметре). Начин на који се серверски сертификат учитава у конфигурацију описан је детаљно на страници:

<https://www.amres.ac.rs/institucije/digitalni-sertifikati>

Када је завршена конфигурација `eap` модула, потребно је прећи у `/raddb/mods-enabled` поддиректоријум и направити `soft` линк ка `eap` модулу из `/raddb/mods-available` поддиректоријума:

```
ln -s ../mods-available/eap
```

## 1.5 proxy.conf

`proxy.conf` се налази у `raddb` директоријуму. Овај конфигурациони фајл служи како би сервер знао на који начин ће обрађивати пристигле RADIUS захтеве. Захтеви се могу обрађивати локално или се могу прослеђивати (проксирати) неком другом серверу. Како је ово упутство намењено за даваоца идентитета, сматра се да се захтеви обрађују локално. Ово значи да је потребно дефинисати само локални домен (`realm`) и у њему означити да се корисници аутентификују локално. Линеје које су осенчене се додају у фајл.

```
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 inst.ac.rs {
    authhost = LOCAL
    accthost = LOCAL
    User-Name = "%{Stripped-User-Name}"
}
realm LOCAL {
}
realm NULL {
```

```
}
```

Уместо домена `inst.ac.rs` потребно је ставити домен ваше институције који користите у оквиру AMRES-а (нпр. `rgf.bg.ac.rs`).

Овим су завршена основна подешавања неопходна за функционисање RADIUS сервера и eduroam-а. За проверу конфигурације препорука је да се RADIUS процес покрене прво у *debug* моду, командом `radiusd -X`. Уколико је конфигурација без грешака, на екрану се исписује следеће:

```
.  
. .  
Listening on authentication address * port 1812  
Listening on accounting address * port 1813  
Listening on proxy address * port 1814  
Ready to process requests.
```

Препорука је да се *debug* мод користи само у случају када се праве измене у конфигурацији или у случају када неко од корисника има проблем са повезивањем на eduroam, јер се у *debug* моду виде лозинке у *clear-text* формату.

RADIUS процес се покреће у стандардном моду командом `radiusd`, а зауставља се командом `killall radiusd`.

## 1.6 radiusd.conf

Иако давалац идентитета нема обавезу да чува лог фајлове, препоруке је да се омогући бележење аутентификационих захтева у `radius.log` фајл. Овај фајл се налази у `/usr/local/var/log/radius` директоријуму. За сваког корисника (уколико се користи EAP-TTLS) се уписују две линије:

- ✦ корисничко име из спољашњег (тј. eduroam) тунела, најчешће у форми **anonymous@inst.ac.rs**, и
- ✦ право корисничко име из унутрашњег (тј. eduroam-inner-tunnel) тунела, нпр. **pera.peric@inst.ac.rs**.

Бележење аутентификационих захтева може бити веома корисно у случају када неко од корисника даваоца идентитета на нерегуларан начин користи eduroam (нпр. дељење корисничког имена и лозинке другим корисницима, кршење ауторских права коришћењем *torrent*-а и сл).

Да би FreeRADIUS бележио аутентификационе захтеве, потребно је да се у `log` секцији `radiusd.conf` фајла (налази се у `raddb` директоријуму) направе измене тако да изгледа као у примеру који је дат у наставку:

```
.  
. .  
log {  
    destination = files  
  
    file = ${logdir}/radius.log
```

```
syslog_facility = daemon

stripped_names = no

auth = yes

auth_badpass = no
auth_goodpass = no

}
.
.
.
```

Након измена, потребно је проверити да ли се конфигурација учитала на прави начин тако што се RADIUS сервер покрене у debug моду. Уколико се конфигурација учитала без проблема, RADIUS процес се покреће у стандардном моду командом `radiusd`.

## 1.7 Повезивање RADIUS сервера и LDAP сервера

### 1.7.1 Конфигурација ldap модула

Последњи корак подразумева конфигурисање FreeRADIUS-а да аутентификује кориснике преко ldap модула. Сви модули се налазе у `/raddb/mods-available` директоријуму. Осенчене параметре је потребно променити тако да одговарају параметрима институције.

**НАПОМЕНА:** Параметар `server` се користи за дефинисање сервера на ком се налази LDAP директоријум. Параметри `identity` и `password` могу бити исти као у `slapd.conf` конфигурационом фајлу. Препорука је да се дефинише налог који нема root привилегије, а који ће се користити за повезивање са LDAP сервером. Последњи освечени параметер у почетном делу ldap модула, `base_dn`, се користи за дефинисање гране од које ће почињати претрага корисника, а у овом случају је то корен стабла. Свуда где је наведено `dc=inst,dc=ac,dc=rs`, потребно је ову вредност заменити тако да буде иста као параметар `suffix` из `slapd.conf` фајла.

```
ldap {
    server = 'localhost'
    identity = 'cn=Manager,dc=inst,dc=ac,dc=rs'
    password = secret
    base_dn = 'dc=inst,dc=ac,dc=rs'
    sasl {
    }
    update {
```

```
control:Password-With-Header += 'userPassword'  
control:Stripped-User-Name := 'uid'  
  
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
    }
}
```

Када је завршена конфигурација `ldap` модула, потребно је прећи у `/raddb/mods-enabled` поддиректоријум и направити `soft` линк ка `ldap` модулу из `/raddb/mods-available` поддиректоријума:

```
ln -s ../mods-available/ldap
```

Након измена, потребно је проверити да ли се конфигурација учитала на прави начин тако што се RADIUS сервер покрене у *debug* моду. Уколико се конфигурација учитала без проблема, RADIUS сервер се покреће у стандардном моду командом `radiusd`.