Uputstvo za konfiguraciju FreeRADIUS-a za davaoce identiteta i LDAP bazu

Uvod

Ovo uputstvo se odnosi na podešavanje osnovnih modula FreeRADIUS servera za davaoce identiteta, pri čemu su prikazani moduli konfigurisani tako da pri procesu eduroam autentifikacije čitaju korisničke podatke iz LDAP baze. Konfiguracioni moduli su prikazani bez komentara, da bi prikaz bio pregledan.

1. clients.conf

clients.conf se nalazi u raddb folderu i predstavlja modul u kome se definišu klijenti RADIUS servera – uređaji od kojih se primaju RADIUS zahtevi (u opštem slučaju mogu biti drugi RADIUS serveri ili mrežni pristupni uređaji).

Potrebno je definisati dva AMRES FTLR (*Federation Top Level Radius Server*) servera i još jedan server koji se koristi za nadgledanje operativnosti eduroam servisa. AMRES FTLR serveri predstavljaju nacionalne eduroam servere najvišeg nivoa. Lozinke za servere se mogu dobiti telefonskim putem ili se mogu preuzeti lično. Konfiguracija clients.conf modula koja je ovde data se može dodati u postojeću konfiguraciju u clients.conf fajl (naravno, potrebno je promeniti lozinke).

```
## 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
}
```

2. eduroam virtuelni server

Virtuelni serveri omogućavaju konfiguraciju većeg broja nezavisnih servisa na FreeRADIUS platformi. Za potrebe eduroam servisa, potrebno je formirati eduroam virtuelni server koji će obrađivati autentifikacione zahteve:

a) odlazimo u direktorijum /raddb/sites-available/

b) kopiramo *default* konfiguracioni fajl u novi eduroam fajl:

```
cp default eduroam
```

c) sada je potrebno je izmeniti kreirani eduroam konfiguracioni fajl, na početku, pre *Authorize* sekcije je potrebno uneti „`server eduroam {`“ i na kraju dokumenta „`}`“ (osenčeni redovi u konfiguraciji pod d)).

d) zatim je potrebno promeniti parametre u eduroam konfiguracionom fajlu tako da krajnja konfiguracija bude kao u nastavku (komentari su izbačeni preko grep komande, da bi konfiguracija bila preglednija):

```
server eduroam {
authorize {
        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
}
}
```

e) na kraju je potrebno preći u folder /usr/local/etc/raddb/sites-enabled i napraviti *soft link* ka eduroam konfiguracionom fajlu pomoću sledeće komande:

```
ln -s /usr/local/etc/raddb/sites-available/eduroam
```

3. eduroam-inner-tunnel

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 LDAP):

a) u direktorijumu /raddb/sites-available prekopirati *inner-tunnel* u novi modul eduroam-inner-tunnel:

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

b) kreiranom fajlu u prvoj konfiguracionoj liniji promeniti ime u eduroam-inner-tunnel (prikazano osenčeno u konfiguraciji u nastavku) i konfigurisati ga tako da krajnja konfiguracija bude:

```
server eduroam-inner-tunnel {
authorize {
        auth_log
        suffix
        eap
        ldap
        pap
```

```
}
authenticate {
        Auth-Type PAP {
        pap
}
Auth-Type CHAP {
        chap
}
Auth-Type MS-CHAP {
        mschap
}
        unix
        eap
}
session {
        radutmp
}
post-auth {
        reply_log
        Post-Auth-Type REJECT {
        attr_filter.access_reject
        }
}
pre-proxy {
}
post-proxy {
        eap
}
}
```

c) na kraju je potrebno preći u folder /usr/local/etc/raddb/sites-enabled i napraviti *soft link* ka eduroam-inner-tunnel konfiguracionom fajlu pomoću sledeće komande:

```
ln -s  /usr/local/etc/raddb/sites-available/eduroam-inner-tunnel
```

4. Komunikacija sa ldap bazom

Naredni korak je definisanje komunikacije sa LDAP korisničkom bazom:

Mapiranje atributa između FreeRADIUS-a i ldap baze je definisano u ldap.attrmap fajlu (raddb folder). U ovom fajlu je potrebno definisati koje atribute će RADIUS čitati iz ldap baze. Primer ovog modula je dat u nastavku (osenčeni redovi su promenjeni u odnosu na default konfiguraciju):

```
checkItem       Auth-Type                   radiusAuthType
checkItem       Simultaneous-Use            radiusSimultaneousUse
checkItem       Called-Station-Id           radiusCalledStationId
checkItem       Calling-Station-Id          rsEduAccessPhoneNumber
checkItem       LM-Password                 lmPassword
checkItem       NT-Password                 ntPassword
checkItem       LM-Password                 sambaLmPassword
checkItem       NT-Password                 sambaNtPassword
```

```
checkItem       SMB-Account-CTRL-TEXT      acctFlags
checkItem       Expiration                 radiusExpiration
checkItem       NAS-IP-Address             radiusNASIpAddress
checkItem       Password-With-Header       userPassword
checkItem       User-Name                  uid
checkItem       Pool-Name                  ismemberof

replyItem       Service-Type               radiusServiceType
replyItem       Framed-Protocol            radiusFramedProtocol
replyItem       Framed-IP-Address          radiusFramedIPAddress
replyItem       Framed-IP-Netmask          radiusFramedIPNetmask
replyItem       Framed-Route +             radiusFramedRoute
replyItem       Framed-Routing             radiusFramedRouting
replyItem       Filter-Id                  radiusFilterId
replyItem       Framed-MTU                 radiusFramedMTU
replyItem       Framed-Compression         radiusFramedCompression
replyItem       Login-IP-Host              radiusLoginIPHost
replyItem       Login-Service              radiusLoginService
replyItem       Login-TCP-Port             radiusLoginTCPPort
replyItem       Callback-Number            radiusCallbackNumber
replyItem       Callback-Id                radiusCallbackId
replyItem       Framed-IPX-Network         radiusFramedIPXNetwork
replyItem       Class                      radiusClass
replyItem       Session-Timeout            radiusSessionTimeout
replyItem       Idle-Timeout               radiusIdleTimeout
replyItem       Termination-Action         radiusTerminationAction
replyItem       Login-LAT-Service          radiusLoginLATService
replyItem       Login-LAT-Node             radiusLoginLATNode
replyItem       Login-LAT-Group            radiusLoginLATGroup
replyItem       Framed-AppleTalk-Link      radiusFramedAppleTalkLink
replyItem       Framed-AppleTalk-Network   radiusFramedAppleTalkNetwork
replyItem       Framed-AppleTalk-Zone      radiusFramedAppleTalkZone
replyItem       Port-Limit                 radiusPortLimit
replyItem       Login-LAT-Port             radiusLoginLATPort
replyItem       Reply-Message              radiusReplyMessage
```

Modul ldap (`/raddb/modules`) je odgovoran za komunikaciju FreeRADIUS servera sa ldap bazom. U nastavku je prikazan ldap modul koji je potrebno konfigurisati prema vašim parametrima (osenčene linije su one koje je potrebno promeniti, ostale ne menjati):

- `server` - ako je ldap server na istoj mašini kao i FreeRADIUS onda je ovde potrebno ostaviti localhost, ako je ldap server na drugoj mašini onda je ove potrebno definisati IP adresu tog servera
- `identity` - korisnik koji postoji u ldap bazi i kome je dozvoljeno da čita iz nje
- `password` - lozinka tog korisnika
- `basedn` - čvor u ldap stablu gde se nalaze korisnici

```
ldap {
        server = "localhost"
        identity = "uid=radius,ou=SystemAccounts,dc=bg,dc=ac,dc=rs"
        password = pass
        basedn = "ou=People,dc=bg,dc=ac,dc=rs"
        filter = "(uid=%{Stripped-User-Name:-%{User-Name}})"
```

```
        ldap_connections_number = 5
        timeout = 4
        timelimit = 3
        net_timeout = 1
        tls {
                start_tls = no
        }
        dictionary_mapping = ${confdir}/ldap.attrmap
        password_attribute = userPassword
        auto_header = yes
        edir_account_policy_check = no
}
```

5. eap.conf

Sada je potrebno izmeniti eap.conf modul, čime se postiže aktiviranje željenog autentifikacionog metoda (u ovom primeru je to EAP-TTLS a može biti i EAP-PEAP). Prvo je potrebno aktivirati željenu autentifikaciju u prvoj osenčenoj liniji prikazanoj u narednoj konfiguraciji, zatim u odgovarajućoj sekciji (TTLS ili PEAP) u zavisnosti od metoda koji ste odabrali promeniti parametar virtual_server na eduroam-inner-tunnel (druga osenčena linija u narednoj konfiguraciji - u ovom primeru je prikazana konfiguracija za TTLS):

```
    eap {
                default_eap_type = ttls     #može biti i 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 {
                         certdir = ${confdir}/certs
                         cadir = ${confdir}/certs
                         private_key_password = whatever
                         private_key_file = ${certdir}/server.pem
                         certificate_file = ${certdir}/server.pem
                         CA_file = ${cadir}/ca.pem

                         dh_file = ${certdir}/dh
                         random_file = ${certdir}/random
                         CA_path = ${cadir}
                         cipher_list = "DEFAULT"
                         make_cert_command = "${certdir}/bootstrap"
                         cache {
                               enable = no
                               max_entries = 255
```

```
                        }
                        verify {
                        }
                }
                ttls {
                        default_eap_type = md5
                        copy_request_to_tunnel = no
                        use_tunneled_reply = no
                        virtual_server = "eduroam-inner-tunnel"
                }
                peap {
                        default_eap_type = mschapv2
                        copy_request_to_tunnel = no
                        use_tunneled_reply = no
                        virtual_server = "inner-tunnel"
                }
                mschapv2 {
                }
        }
```

Institucija je slobodna da odluči od kog RA (*Registration Authority*) će nabaviti digitalni sertifikat koji će postaviti na RADIUS server. AMRES nudi besplatne TCS serverske sertifikate, koji se mogu dobiti preko postupka opisanog na stranici:

http://www.amres.ac.rs/index.php?option=com_content&task=view&id=236&Itemid=259

postupak postavljanja digitalnog sertifikata na FreeRADIUS server je opisan na stranici:

http://www.amres.ac.rs/index.php?option=com_content&task=view&id=240&Itemid=263

6. proxy.conf

proxy.conf predstavlja modul koji odlučuje da li će pristigli autentifikacioni zahtev biti obrađen lokalno ili će biti prosleđen nekom drugom serveru (proksiran). Obzirom da je ovde reč o konfiguraciji davaoca identiteta, svi pristigli zahtevi će biti obrađivani lokalno, tako da je potrebno kreirati samo lokalni domen (*realm*) i u njemu označiti da je autentifikacija lokalna. Primer konfiguracije (konfiguracione linije koje su osenčene je potrebno dodati u 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 inst.ac.rs {
        authhost        = LOCAL
        accthost        = LOCAL
        User-Name       = "%{Stripped-User-Name}"
}
realm LOCAL {
}
realm NULL {
}
```

Umesto domena ***inst.ac.rs*** potrebno je da postavite domen vaše institucije (primer ***fon.bg.ac.rs***).

7. radiusd.conf

Iako Davalac Identiteta nema obavezu da čuva logove, preporuka je da se omogući upisivanje autentifikacionih zahteva u radius.log fajl. Ovaj fajl se nalazi u /usr/local/var/log/radius direktorijumu. Za svakog korisnika (ukoliko se koristi EAP TTLS) se upisuju dve linije:

- korisničko ime iz spoljašnjeg (eduroam) tunela, najčešće anonymous@inst.ac.rs, i
- pravo korisničko ime, iz unutrašenjeg (eduroam-inner-tunnel) tunela, pera.peric@inst.ac.rs.

Logovanje autentifikacionih zahteva može biti veoma korisno u slučaju kada neko od korisnika Davaoca Identiteta na neregularan način koristi eduroam servis (npr. deljenje korisničkog imena/lozinke).

Da bi FreeRADIUS logovao autentifikacione zahteve, potrebno je da se u log sekciji radiusd.conf fajla naprave izmene tako da izgleda kao u sledećem primeru:

```
.
.
.
log {
        destination = files

        file = ${logdir}/radius.log


        syslog_facility = daemon

        stripped_names = no

        auth = yes

        auth_badpass = no
        auth_goodpass = no

}
.
.
.
```

8. 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 jednistveni identifikator svakog eduroam korisnika, formira ga Davalac Identiteta i šalje Davaocu Resursa.
- **ON** (Operator-Name) je jednistveni 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 *instantiate* 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 Identiteta bitna samo sledeća:

- cui_postauth – funkcija kojom davalac identiteta izračunava CUI i vraća ga davaocu resursa

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).

```
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
   }
}
```

Da bi se slao CUI atribut, potrebno je u /raddb/sites-available/eduroam fajlu, pre sekcije server eduroam, dodati sledeće parametre:

```
      #  This defines the salt value for CUI. See also other CUI configration
      #  If *returning* the CUI, set cui_hash_key to some random string
      #  and uncomment the line below

           cui_hash_key = "eduroam"

      #  If *returning* the CUI and the Operator-Name attribute in request is
      #  required, uncomment the line below

           cui_require_operator_name = yes

      #  Authorization. First preprocess (hints and huntgroups files),
      #  then realms, and finally look in the "users" file.
      #
      #  The order of the realm modules will determine the order that
      #  we try to find a matching realm.
      #
      #  Make *sure* that 'preprocess' comes before any realm if you
      #  need to setup hints for the remote radius server

      server eduroam {
```

```
        .
        .
        .
}
```

Parametar cui_hash_key u kombinaciji sa User-Name atributom (unutar tunela, jer bi se u suprotnom koristila vrednost anonymous) i Operator-Name atributom se koristi za dobijanje vrednosti CUI na sledeći način: obavi se nadovezivanje ova tri stringa kako bi se dobio jedan veći string, a zatim se primeni MD5 hash algoritam nad tim stringom (ovo je definisano u policy.conf fajlu iz ovog uputstva). Ovako dobijena vrednost se vraća RP-u i upisuje se u MySQL tabelu, iz koje se u slučaju zloupotrebe eduroam servisa može obaviti mapiranje između MAC adrese i vrednosti CUI atributa i tako doći do krajnjeg korisnika koji je zloupotrebio servis.

U eduroam-inner-tunnel-u se može videti pravo korisničko ime, pa je potrebno napraviti samo izmenu u post-auth sekciji ovog fajla, tako da se umesto anonymous vraća pravo korisničko ime nakon uspešne autentifikacije:

```
post-auth {
        #  cui_postauth reacts to the Chargeable-User-Identity request
        #  by adding the md5 hash created from a configurable local
        #  salt (cui_hash_key) and the (inner) User-Name value
        #  uncomment the line below if *returning* the CUI
        #

        cui_postauth
        .
        .
        .
}
```

Na ovaj način je završeno podešavanje kojim se omogućava vraćanje vrednosti CUI atributa.

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).