1. FreeRADIUS Kurulumu

Bu belge, FreeRADIUS 2.1.4 sürümünün, 2009 Mayıs ayı güncellemeleri yapılmış Debian Testing ve Stable dağıtımları üzerinde kurulum adımlarını kapsamaktadır. Diğer Linux dağıtımları için benzer adımlar, dağıtıma özgü arayüzler ve paketler kullanılarak gerçekleştirilebilir.

Kuruluma başlamadan önce işletim sisteminin son güncellemelerini yaparak aşağıda sıralanmış olan paketlerin kurulu olduğundan emin olunuz.

gcc, libssl-dev, make, flex, bison, libtool, autoconf, gcc-multilib, tcpdump, libpcap-dev

Ayrıca sunucu kurulumuna başlamadan önce **OpenSSL** paketinin kurulu olduğundan emin olunuz. Daha sonra aşağıda verilen adımları izleyerek FreeRADIUS sunucu kurulumunu yapabilirsiniz.

```
~# mkdir depo
~# cd depo/
~/depo# wget ftp://ftp.freeradius.org/pub/freeradius/freeradius-server-2.1.4.tar.gz
~/depo# tar zxvf freeradius-server-2.1.4.tar.gz
~/depo# cd freeradius-server-2.1.4
~/depo/freeradius-server-2.1.4# ./configure –prefix=/usr/freeradius --with-openssl >
radcompile.log 2> radcompile.error.log
~/depo/freeradius-server-2.1.4# make -s
~/depo/freeradius-server-2.1.4# make -s install
```

Öncelikle depo adında bir klasör oluşturulur ve FreeRADIUS 2.1.4 sürümünün kaynak kodlarını içeren sıkıştırılmış arşiv dosyası (freeradius-server-2.1.4.tar.gz) bu klasör altına kopyalanarak açılır. Daha sonra ***"configure"*** ve ***"make"*** komutları kullanılarak program derlenir ve ***"make install"*** komutu ile kurulum gerçekleştirilir. Örnek kurulumda FreeRADIUS **/usr/freeradius** dizini altına kurulmuştur. Ayrıca kurulum ile ilgili hata mesajları daha sonra inceleyebilmeniz için **radcompile.error.log** dosyasına yazdırılmıştır. Bu dosyadaki hataları inceleyip, ihtiyacınız olan tüm modüllerin derlendiğini görmeden kurulumu tamamlamayınız.

2. Sertifika Oluşturulması

FreeRADIUS kurulduktan sonra, öncelikle kurulumu yaptığımız **/usr/freeradius/etc/raddb/** dizini altında bulunan **certs** dizinine giderek test amacıyla kurulum sırasında oluşturulan bütün test sertifikalar silinir.

.

```
~/depo/freeradius-server-2.1.4# cd /usr/freeradius/etc/raddb/certs
~/usr/freeradius/etc/raddb/certs# rm -f *.pem *.der *.csr *.crt *.key *.p12 serial*
index.txt* dh random
```

Daha sonra aynı dizinde bulunan ve kök sertifika oluşturulması için gerekli tanımları içeren **ca.cnf** dosyası üzerinde gerekli değişiklikler yapılır. Aşağıda bu dosyada değişiklik yapılması gereken bölümler verilmiştir.

Daha sonra sunucu sertifikası ve kullanıcı sertifikası için gerekli tanımları içeren **server.cnf** ve **client.cnf** dosyaları hazırlanır. Bu dosyalar ile ilgili değişiklikleri içeren bölümler aşağıda verilmiştir.

ca.cnf

```
[ req ]
prompt                  = no
distinguished name      = certificate authority
default_bits            = 2048
input_password          = demirgibikollarimhicaffetmemsollarim
output_password         = demirgibikollarimhicaffetmemsollarim
x509 extensions         = v3 ca

[certificate_authority]
countryName             = TR
stateOrProvinceName     = ANKARA
localityName            = BILKENT
organizationName        = TUBITAK ULAKBIM
emailAddress            = eduroam@ulakbim.gov.tr
commonName              = "TUBITAK ULAKBIM eduroam Certificate Authority"
```

server.cnf:

```
[ req ]
prompt                  = no
distinguished_name      = server
default bits            = 2048
input_password          = demirgibikollarimhicaffetmemsollarim
output_password         = demirgibikollarimhicaffetmemsollarim

[server]
countryName             = TR
stateOrProvinceName     = ANKARA
localityName            = BILKENT
organizationName        = TUBITAK ULAKBIM
emailAddress            = eduroam@ulakbim.gov.tr
commonName              = "TUBITAK ULAKBIM eduroam Radius Server Certificate"
```

client.cnf:

```
[ req ]
prompt                  = no
distinguished_name      = client
default_bits            = 2048
input_password          = demirgibikollarimhicaffetmemsollarim
output_password         = demirgibikollarimhicaffetmemsollarim

[client]
countryName             = TR
stateOrProvinceName     = ANKARA
localityName            = BILKENT
organizationName        = TUBITAK ULAKBIM
emailAddress            = eduroam@ulakbim.gov.tr
commonName              = "TUBITAK ULAKBIM eduroam Radius Client Certificate"
```

Dosyalarda gerekli değişiklikler yapıldıktan sonra, **bootstrap** komutu çalıştırılarak gereken sertifikalar oluşturulur. Bu komutu çalıştırdığınızda ekranda aşağıdaki mesajları görebilirsiniz.

```
~/usr/freeradius/etc/raddb/certs# ./bootstrap
openssl dhparam -out dh 1024
Generating DH parameters, 1024 bit long safe prime, generator 2
This is going to take a long time
.....................................................................+.....................
.+......+........+..........+..................+.....................................
..................++*++*++*
openssl req -new  -out server.csr -keyout server.key -config ./server.cnf
Generating a 2048 bit RSA private key
.....................................................................................
.....................................................+++
.........................................+++
writing new private key to 'server.key'
-----
openssl req -new -x509 -keyout ca.key -out ca.pem \
              -days `grep default days ca.cnf | sed 's/.*=//;s/^ *//'` -config
./ca.cnf
Generating a 2048 bit RSA private key
```

```
.....................................................+++
.........+++
writing new private key to 'ca.key'
-----
openssl ca -batch -keyfile ca.key -cert ca.pem -in server.csr  -key `grep
output_password ca.cnf | sed 's/.*=//;s/^ *//'` -out server.crt -extensions
xpserver ext -extfile xpextensions -config ./server.cnf
Using configuration from ./server.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
       Serial Number: 1 (0x1)
       Validity
           Not Before: Mar 18 09:11:23 2009 GMT
           Not After : Mar 17 09:11:23 2012 GMT
       Subject:
           countryName               = TR
           stateOrProvinceName       = ANKARA
           organizationName          = TUBITAK ULAKBIM
           commonName                = TUBITAK ULAKBIM eduroam Radius Server
Certificate
           emailAddress              = eduroam@ulakbim.gov.tr
       X509v3 extensions:
           X509v3 Extended Key Usage:
               TLS Web Server Authentication
Certificate is to be certified until Mar 17 09:11:23 2012 GMT (1095 days)

Write out database with 1 new entries
Data Base Updated
openssl pkcs12 -export -in server.crt -inkey server.key -out server.p12  -passin
pass:`grep output password server.cnf | sed 's/.*=//;s/^ *//'` -passout pass:`grep
output password server.cnf | sed 's/.*=//;s/^ *//'`
openssl pkcs12 -in server.p12 -out server.pem -passin pass:`grep output_password
server.cnf | sed 's/.*=//;s/^ *//'` -passout pass:`grep output_password server.cnf |
sed 's/.*=//;s/^ *//'`
MAC verified OK
openssl x509 -inform PEM -outform DER -in ca.pem -out ca.der
~/fr-2.1.4/etc/raddb/certs# cd ..
~/fr-2.1.4/etc/raddb#
```

3. FreeRADIUS Konfigurasyon Dosyalarının Yapılandırılması

3.1 “radiusd.conf” Dosyasının Yapılandırılması

Bu dosya radius sunucusu ile ilgili temel parametreleri içermektedir. Sunucunun hizmet vereceği IP adresi ve port numaraları burada tanımlanmaktadır. Bir değişiklik yapılmaması durumunda FreeRADIUS sunucusu, TCP ve UDP protokollerinde 1812, 1813 ve 1814 portlarını kullanmaktadır.

Bu dosyada ayrıca radius sunucusu tarafından tutulacak kayıtlar (log) ile ilgili parametreler tanımlanmaktadır. Daha ayrıntılı kayıt tutmak için log modülünün 430 ve 436 satırlarda aşağıdaki değişiklikler yapılır.

```
   #Log the full User-Name attribute, as it was found in the request.
            #
            # allowed values: {no, yes}
            #
            stripped_names = yes

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

3.2 “eap.conf” Dosyasının Yapılandırılması

Öncelikle bu dosyadaki ilk modül altında kullanılacak eap tipi tanımlanır. Ayrıca bu modül altında kullanmayacağınız diğer kimlik denetimi yöntemleri olan md5, leap ve gtc ile ilgili satırları # koyarak kapatabilirsiniz.

```
eap {
                #  Invoke the default supported EAP type when
                #  EAP-Identity response is received.
                #
                #  The incoming EAP messages DO NOT specify which EAP
                #  type they will be using, so it MUST be set here.
                #
                #  For now, only one default EAP type may be used at a time.
                #
                #  If the EAP-Type attribute is set by another module,
                #  then that EAP type takes precedence over the
                #  default type configured here.
                #
                default_eap_type = ttls
```

Daha sonra **tls** modülü altında (satır 154) sertifika oluştururken kullanılan parola yazılır.

```
 tls {
                #
                #  These is used to simplify later configurations.
                #
                certdir = ${confdir}/certs
                cadir = ${confdir}/certs

               private_key_password = demirgibikollarimhicaffetmemsollarim

                   private_key_file = ${certdir}/server.pem
```

Gerekli sertifikaları bir defaya mahsus oluşturmak için daha önce bootstap komutunu çalıştırdığımızdan dolayı, dosyanın 260 satırına # koyarak kapatmalısınız.

```
#  This configuration entry should be deleted
                #  once the server is running in a normal
                #  configuration.  It is here ONLY to make
                #  initial deployments easier.
                #
            #make_cert_command = "${certdir}/bootstrap"
```

Bir sonraki adımda ttls modülü ile ilgili tanımlamaları yapalım. Öncelikle tünel içinde, kimlik doğrulama için kullanılacak şifreleme yöntemi seçilmelidir. Bizim kurulumumuzda **md5** olarak seçilmiştir.

Tünel üzerinden gönderilen kimlik doğrulama isteklerinde, tünel içindeki ve tünel dışındaki bilgiler farklıdır. Eğer gönderilen istekte yeralan ve dolayısıyla tünel dışında kalan bazı önemli bilgilerin (attribute) tünel içindeki istek paketlerinde de bulunmasını istiyor iseniz ttls modülü altında aşağıda verilen değişiklikleri yapmalısınız (copy_request_to_tunnel = yes).

Kurumlarda güvenlik nedeniyle kullanıcıların iç ve dış kullanıcı adları farklı olarak tanımlaynabilir. Dış kimlik olarak anonymous@alan.adı veya benzer şekilde, alan adı kısmı sabit kalmak kaydıyla farklı bir kullanıcı ismi kullanılması, kurumun kullanıcı adlarının 3. kişilerce ele geçirilmesini

önler. Eğer sizde radius sunucunuzun, kimlik doğrulama işlemini tünel içindeki kullanıcı adına göre gerçekleştirmesini istiyor iseniz ttls modülünde değişiklik yapmanız gerekecektir (use_tunneled_reply = yes).

Son olarak bu dosyada tünel içindeki isteklerin gönderileceği sanal sunucu tanımlanır. Yine kullanmadığınız modüller var ise ilgili satırların başlarına # koyarak kapatabilirsiniz.

```
ttls {
   #  The tunneled EAP session needs a default
   #  EAP type which is separate from the one for
   #  the non-tunneled EAP module.  Inside of the
   #  TTLS tunnel, we recommend using EAP-MD5.
   #  If the request does not contain an EAP
   #  conversation, then this configuration entry
   #  is ignored.
   default_eap_type = md5

   #  The tunneled authentication request does
   #  not usually contain useful attributes
   #  like 'Calling-Station-Id', etc.  These
   #  attributes are outside of the tunnel,
   #  and normally unavailable to the tunneled
   #  authentication request.
   #
   #  By setting this configuration entry to
   #  'yes', any attribute which NOT in the
   #  tunneled authentication request, but
   #  which IS available outside of the tunnel,
   #  is copied to the tunneled request.
   #
   # allowed values: {no, yes}
   copy_request_to_tunnel = yes


   #  The reply attributes sent to the NAS are
   #  usually based on the name of the user
   #  'outside' of the tunnel (usually
   #  'anonymous').  If you want to send the
   #  reply attributes based on the user name
   #  inside of the tunnel, then set this
   #  configuration entry to 'yes', and the reply
   #  to the NAS will be taken from the reply to
   #  the tunneled request.
   #
   # allowed values: {no, yes}
  use_tunneled_reply = yes
        #
        #  The inner tunneled request can be sent
        #  through a virtual server constructed
        #  specifically for this purpose.
        #
        #  If this entry is commented out, the inner
        #  tunneled request will be sent through
        #  the virtual server that processed the
        #  outer requests.
        #
        virtual_server = "inner-tunnel"
}
```

3.3 “inner-tunnel” Sanal Sunucu Dosyasının Yapılandırılması

FreeRADIUS konfigürasyon dosyalarının bulunduğu **/usr/freeradius/etc/raddb** dizini altındaki **sites-available** dizini altında inner-tunnel konfigürasyon dosyası bulunmaktadır. Bu dosya içerisinde, kullanıcılarınızın kimlik doğrulama işlemlerini yaparken hangi kaynaktan yararlanacağınızı belirtirsiniz. Kullanıcı adı ve şifreler bir text dosyasında (users dosyası) veya bir veritabanında kayıtlı olabileceği gibi aynı sunucu üzerinde tanımlı kullanıcı tablosunun kullanılması

amacıyla doğrudan **/etc/passwd** dosyasında da bulunabilir. Aşağıda verilen örnek konfigürasyon dosyasında kullanıcı bilgileri **/etc/passwd** dosyasında tutulmaktadır. Bu nedenle yetkilendirme için kullanılacak modül ile ilgili tanımları içeren authorize bölümünde **unix** modülü suffix modülünden hemen sonra yer almaktadır. Eğer kullanıcı bilgilerini users dosyasında tutmak istiyor iseniz **files** modülünü, eğer bir veritabanı kullanmak istiyor iseniz **sql** veya **ldap** gibi modülleri tanımlamalısınız. Aşağıda örnek olarak verilen inner-tunnel dosyası, içerisindeki #'li satırlar ayıklanarak bu belge içerisine dahil edilmiştir, bu nedenle bütün tanım opsiyonlarını içermemektedir. Kimlik doğrulama ile ilgili tanımların yer aldığı authentication bölümünde ise Auth-Type pap olarak tanımlanmıştır. Detaylı kayıt bilgisine sahip olmak için **post-auth** bölümünde, **reply_log**, **pre_proxy_log** ve **post_proxy_log** modülleri ile ilgili tanımlar da yapılmalıdır.

```
~/usr/freeradius/etc/raddb/sites-available# cat inner-tunnel
server inner-tunnel {
authorize {
       suffix
       unix
       update control {
              Proxy-To-Realm := LOCAL
       }
       eap {
               ok = return
       }
       files
       pap
}
authenticate {
       Auth-Type PAP {
               pap
       }
       eap
}
session {
       radutmp
}
post-auth {
       reply_log
       Post-Auth-Type REJECT {
               attr filter.access reject
       }
}
pre-proxy {
       pre_proxy_log
}
post-proxy {
       post_proxy_log
}
```

Son olarak inner-tunnel dosyası için aşağıdaki sembolik link oluşturulmalıdır. Sanal sunucu dosyaları, **sites-available** altında bulunur, aktive edilecek olanların sembolik linki **sites-enabled** altına konulur.

```
~/usr/freeradius/etc/raddb/sites-available# cd ..
~/usr/freeradius/etc/raddb# ln -fs sites-available/inner-tunnel sites-enabled/inner-
tunnel
```

3.4 Sanal Sunucu (Virtual Server) Yapılandırılması

FreeRADIUS 2 sürümü ile birlikte sanal sunucu (virtual servers) desteği gelmiştir. Böylece bir adet FreeRADIUS sunucusu ile farklı IP ve portlardan gelen farklı profillere sahip isteklere, farklı sunucu hizmeti verilebilinir. Bu belgede sites-available dizini altında eduroam-local isimli bir sanal sunucunun tanımları verilmiştir. Öncelikle kullanıcı bilgilerini doğrulamak için kullanılacak araç tanımlanır. Aşağıdaki örnekte **/etc/passwd** dosyası öncelikli olarak tanımlanmıştır, daha sonra **users** dosyası da kullanılabilir. Ayrıca aşağıdaki örnek tanımları arasında 5 satırlık bir if döngüsü

yerleştirilerek, alan adı içermeyen **"kullanı_adı@"** şeklinde gelen istekler var ise bunların başka bir sunucuya gönderilmeden red cevabı gönderilmesi sağlanmıştır. FreeRADIUS sunucuları sadece kullanıcı_adı içeren (@ içermeyen) istekleri NULL istek olarak sınıflandırarak gerekli işlemleri yapabilir, ancak @ işareti bulunan istekleri gereksiz yere yönlendirebilir.
Detaylı kayıt bilgisine (log) sahip olmak için gerekli tanımlar da aşağıdaki örnek konfigurasyonda yeralmaktadır.

```
~/usr/freeradius/etc/raddb/sites-available# cat eduroam-local
server eduroam-lokal {
       authorize {
         auth log
         suffix
         eap
         unix
         files
         pap
         #Following 5 lines are for rejecting "user@" type domainless reqs
         if ((Realm == DEFAULT) && (User-Name =~ /.*@$/))  {
           update control {
                 Auth-Type := Reject
           }
         } # End of reject domainless reqs.
       }
       authenticate {
               Auth-Type PAP {
                       pap
               }
               eap
       }
       preacct {
         acct_unique
         suffix
         files
       }
       accounting {
       }
session {
       radutmp
}
       post-auth {
         reply log
         exec
         Post-Auth-Type REJECT {
                  reply_log
         }
       }
       pre-proxy {
               attr filter.pre-proxy
               pre_proxy_log
       }
       post-proxy {
               post proxy log
               attr filter.post-proxy
       }
}
~/fr-2.1.4/etc/raddb/sites-available#
```

Son olarak inner-tunnel dosyası gibi bu dosya için de aşağıdaki sembolik link oluşturulmalıdır.

```
~/usr/freeradius/etc/raddb/sites-available# cd ..
~/usr/freeradius/etc/raddb# ln -fs sites-available/eduroam-local sites-
enabled/eduroam-local
```

3.5 "proxy.conf" Dosyasının Yapılandırılması

Bu dosyada kendi lokal istekleriniz için kurumunuzun alan adı ile ilgili tanımları yapmanız yeterli olacaktır. RFC4282 dokümanında kullanıcı adı ile alan adını ayıran karakter @ olarak belirlenmiştir. Aşağıdaki örnekte sunucu kullanıcı adı ile alan adının ayırıp kimlik doğrulama işlemlerini sadece kullanıcı adına göre yapak üzere yapılandırılmıştır. Bu nedenle **nostrip** satırı # ile kapatılmıştır.

```
realm ulakbim.gov.tr {
    #nostrip
}
```

3.6 “users” Dosyasının Yapılandırılması

Kullanıcı bilgileri harici bir kaynakta değil de FreeRADIUS sunucusu üzerinde tutulacak ise **users** dosyasının içerisine aşağıdaki gibi kullanıcılar eklenir.

```
#
# Gokhan Eryol
eryol@ulakbim.gov.tr          Cleartext-Password := "wholovespassword."
            Reply-Message = "Hello, %u"
```

3.7 “clients.conf” Dosyasının Yapılandırılması

FreeRADIUS sunucunuza istek gönderebilecek cihazların tanımları **clients.conf** dosyasında bulunmaktadır. İstekte bulunacak bütün cihazların (anahtarlama cihazlarının, kablosuz erişim cihazlarının ve varsa diğer radius sunucularının) IP adresleri ve karşılıklı belirlenmiş parolaları burada tanımlanır. IP adresi tanımlarını yaparken IP adreslerini tek tek belirtebileceğiniz gibi IP adres aralığı da tanımlanabilir. Tanımları yaparken isteklerin hangi sanal sunucuya gönderilmesi gerektiği belirtilmelidir. Örnek konfigurasyonda bütün istekler daha önce tanımlanan eduroam-local sunucusuna gönderilmektedir.

```
# ULAKBIM AP Network
client 10.10.60.0/24 {
       secret = supersecretpass
       shortname = floor3-aps
       virtual server = eduroam-local
}
#
# Tens of AP's behind NAT (193.140.100.1)
client 193.140.100.1 {
       secret = supersecretpass2
       shortname = enstitution-aps
       virtual_server = eduroam-local
}
#
```

3.8 FreeRADIUS Sunucusunun Çalıştırılması

Sunucu ilk kez başlatılacak ise test amaçlı ve detaylı log verecek şekilde ***“radiusd –X”*** komutu ile debug kipinde çalıştırılmalıdır. Bir sorun yok ise sadece **“*radiusd*”** komutu ile çalıştırılabilir ve bir servis olarak işletim sistemine eklenebilir.