FreeRADIUS-konfigurointi

Tämän sivun sisältö:

Johdanto

FreeRADIUS on RADIUS-palvelin, joka perustuu avoimeen lähdekoodiin ja toimii mm. useammilla Linux-alustoilla. Sen avulla RADIUS-autentikointiviestit voidaan välittää eteenpäin RADIUS-hierarkiassa ja oman organisaation käyttäjät voidaan autentikoida paikallisesti tai liitettyä tietokantaa käyttäen. Näissä ohjeissa selitetään yksityiskohtaisesti miten FreeRADIUS-palvelin konfiguroidaan lähettämään RADIUS-viestejä eteenpäin verkkovierailuhierarkiassa palveluntarjoajana (service provider) sekä miten omat käyttäjät voidaan autentikoida paikallisesti (identity provider). Käyttäjiä erotellaan aina domainin, realmin perusteella.

FreeRADIUS-palvelimen asentaminen

FreeRADIUS-asennukseen löytyy ohjeita selityksineen netistä: http://wiki.freeradius.org/building/Build. Tähän on pyritty keräämään yhteenveto oheista ilman tarkempia selityksiä. Tässä esimerkissä alustana on ollut *RedHat Enterprise Linux Server 5 64 bit*.

FreeRADIUS on saatavilla netistä: http://koji.fedoraproject.org/koji/packageinfo?packageID=298. Täältä ladataan SRPM-tiedosto, joka tämän esimerkin tapauksessa oli freeradius-2.1.3-1.fc9.src.rpm. Tiedosto siirretään /usr/src/redhat/:in alle esim. tiedostoon /usr/src/redhat/FreeRadSrmp. SRPM-tiedosto puretaan ajamalla *rpm -ihv* -komentoa hakemistossa, johon freeradius-2.1.3-1.fc9.src.rpm on tallennettu: *rpm -ihv freeradius-2.1.3-1.fc9.src.rpm* .

Seuraavassa vaiheessa tarvitaan yum-builddep-työkalu, joka löytyy yum-utils-pakkauksesta. Se asennetaan komennolla *yum install yum-utils*. Seuraavaksi ajetaan *yum-builddep freeradius-2.1.3-1.fc9.src.rpm* FreeRadSrmp-hakemistossa. Jos komennon tuloksessa lukee esim *Error: No Package found for perl-devel* se tarkoittaa, että muutamat riippuvuudet joudutaan asentamaan käsin. Tarvittavat riippuvuudet saadaan selville ajamalla *rpmbuild -ba /usr/src/redhat /SPECS/freeradius.spec* FreeRadSrmp-hakemistossa. Esimerkiksi *error: Failed build dependencies: gdbm-devel is needed by freeradius-2.1.3-1.x86_64* korjataan ajamalla komentoa *yum install gdbm-devel*. Kun tarvittavat riippuvuudet on asennettu käsin, luodaan rpm-tiedosto komennolla *rpmbuild -ba /usr /src/redhat/SPECS/freeradius.spec* FreeRadSrmp-hakemistossa. /usr/src/redhat/RPMS/x86_64/-hakemistossa asennetaan sitten tarvittavat rpm:t. Peruspakkauksen lisäksi tarvitaan ainakin libs-pakkaus, eli ajetaan komento *sudo rpm -Uhv freeradius-2.1.3-1.x86_64.rpm freeradius-libs-2.1.3-1.x86_64.rpm*. Luettelon asennetuista freeradius-paketeista saa komennolla *rpm -qa freeradius**.

Konfigurointi palveluntarjoajaksi (Service Provider)

FreeRADIUS:en tomiminen palveluntarjoajana tarkoittaa sitä, että se välittää tulevat RADIUS-viestit eteenpäin RADIUS-hierarkiassa ja saapuvat paketit WLAN-tukiasemille, kontrollerille tai kytkimille. Se ei itse autentikoi käyttäjiä tietokantaa tai salasanatiedostoa käyttäen.

Tiedostot, jotka täytyy konfiguroida löytyvät /etc/raddb/:n alta ja ovat nimeltään **clients.conf**, **proxy.conf** ja **radiusd.conf**. Lisäksi luodaan uusi tiedosto sites-enabled-hakemistoon ja tehdään symbolinen linkki tähän sites-available-hakemistosta. Esimerkkitiedostot löytyvät liitteestä FreeRADIUS_SP, mutta seuraavassa käydään läpi miten jokainen tiedosto kannattaa konfiguroida.

clients.conf

Tähän tiedostoon määritellään laitteet, jotka saavat lähettää RADIUS-viestejä FreeRADIUS-palvelimelle. Laite lisätään konfiguraatioon seuraavalla tavalla:

```
client my_client{

   ipaddr = xxx.yyy.zzz.www
   netmask = aa
   secret = v8493nfnkwenGYEj     # Tämä salasana on myös oltava klientilla tiedossa
   require_message_authenticator = no
   shortname = my_client
   nastype = other     #tai cisco jos käytetään cisco:n laitteita.
   virtual_server =  eduroam
}
```

Lisäksi on hyvä jättää testausta varten localhost-lohko:

```
client localhost {
    ipaddr = 127.0.0.1
    netmask = 32
    secret        = testing123
    require_message_authenticator = no
    shortname    = loopback
    nastype     = other
    virtual_server = eduroam
}
```

proxy.conf

Tässä tiedostossa määritellään miten klientiltä tulevat RADIUS-viestit välitetään eteenpäin RADIUS-hierarkiassa. Alla on esitetty esimerkki tapauksesta, jossa organisaation oma RADIUS-palvelin liitetään suoraan Suomen juuripalvelimiin. Jos kampuksella on sisäinen RADIUS-hierarkia, määritellään tässä seuraavalla tasolla oleva RADIUS-palvelin/palvelimet.

```
proxy server {
        default_fallback        = yes
}

home_server ftlr_funet_fi {
        type                    = auth+acct
        ipaddr                  = 193.166.5.150
        port                    = 1812
        secret                  = MfhurewrbDm886PR # Tämä salasana on oltava yhteinen molemmilla palvelimilla.
        # Ota tähän liittyen yhteyttä noc@funet.fi:hin.
        response_window         = 20
        zombie_period           = 40
        revive_interval         = 60
        status_check            = status-server
        check_interval          = 30
        num_answers_to_alive    = 3
}

home_server ftlr2_funet_fi {
        type                    = auth+acct
        ipaddr                  = 193.166.4.105
        port                    = 1812
        secret                  = MfhurewrbDm886PR # Tämä salasana on oltava yhteinen molemmilla palvelimilla.
        # Ota tähän liittyen yhteyttä noc@funet.fi:hin.
        response_window         = 20
        zombie_period           = 40
        revive_interval         = 60
        status_check            = status-server
        check_interval          = 30
        num_answers_to_alive    = 3
}

home_server_pool EDUROAM-FTLR {
        type                    = fail-over
        home_server             = ftlr_funet_fi
        home_server             = ftlr2_funet_fi

}

realm LOCAL {
      nostrip
}

realm NULL {
      nostrip
}

realm DEFAULT {
        pool                    = EDUROAM-FTLR
        nostrip
}
```

/sites-available/eduroam

sites-available-hakemistoon luodaan virtuaalipalvelin, joka määriteltiin käytettäväksi clients.conf-tiedostossa (virtual_server = eduroam). Tiedoston nimi on sama kuin virtuaalipalvelimen nimi, eli tämän esimerkin tapauksessa *eduroam*. Palvelimen sisältö esitetään liitetiedostossa.

/sites-enabled/eduroam

Sites-enabled-hakemistoon on luotava symbolinen linkki virtuaalipalvelimeen (eduroam). Se tehdään ajaamalla *ln -s ../sites-available/eduroam eduroam* -komentoa sites-enabled-hakemistossa.

radiusd.conf

Tarkistetaan ensimmäiseksi, että

```
user = radiusd
group = radiusd
```

eivät ole kommenteissa, eli FreeRADIUS:ta ei ole pakko ajaa roottina.

Sitten muutetaan listen-lohkot seuraavasti:

```
listen {
    type = auth
    ipaddr = *  # Määrittele IP-osoitteet
    port  = 1812
}
listen {
    type = acct
    ipaddr = * # Määrittele IP-osoitteet
    port = 1813
}
```

On myös mahdollista lisätä IPv6-tuki.

Varmista seuraavaksi, että security-lohkossa on seuraavat määritelmät:

```
security {
        max_attributes = 200
        reject_delay = 0
        status_server = yes
}
```

Tiedoston loppuun lisätään seuraavat lohkot:

```
detail auth_log {
        detailfile = ${radacctdir}/%Y%m%d/eduroam/auth-detail
        detailperm = 0600
}
realm suffix {
        format = suffix
        delimiter = "@"
}
attr_filter attr_filter.pre-proxy {
        attrsfile = ${confdir}/attrs.pre-proxy
}
```

Testaus ja virheselvitys

Palvelin käynnistetään /etc/raddb/-hakemistosta komennolla *radiusd -X* (debuggaus-moodi). Komento löytyy tarvittaessa /usr/sbin/-hakemistosta.

Palvelimen konfigurointia voidaan ensimmäisessä vaiheessa testata paikallisesti käyttäen klienttiä localhost. Hakemistosta /usr/src/redhat/BUILD /freeradius-server-2.1.3/src/main/ ajetaan komento */usr/src/redhat/BUILD/freeradius-server-2.1.3/src/main/radtest kayttajatunnus@myorganisation.fi SaLaSaNa localhost 1 testing123*

Ongelmien ilmetessä kannattaa tarkistaa, että palomuurit (iptables) on konfiguroitu niin, että FreeRADIUS-palvelin pystyy vastaanottamaan RADIUS-viestejä määritetyiltä klienteiltä, ja että next_server_in_hierarchy pystyy vastaanottamaan palvelimen lähettämät RADIUS-viestit.

Konfigurointi identiteettitarjoajaksi (Identity provider)

Seuraava vaihe on muuttaa konfiguraatiota niin, että palvelin toimii sekä palveluntarjoajana että identiteettitarjoajana. Saapuvien RADIUS-viestien domain-tietoja tarkkaillaan ja omat käyttäjät autentikoidaan ja vieraiden viestit välitetään eteenpäin hierarkiassa.

Muutoksia pitää tehdä seuraavissa tiedostoissa: **clients.conf**, **proxy.conf**, **/sites-available/eduroam**, **eap.conf** ja **users**. Lisäksi pitää luoda /sites-available/eduroam-inner-tunnel-tiedosto ja tähän tiedostoon symbolinen linkki sites-enabled-hakemistoon. Esimerkkitiedostoja löytyy liitteestä FreeRADIUS_IdP, mutta seuraavassa käydään läpi miten jokainen tiedosto kannattaa konfiguroida.

clients.conf

Jotta Suomen juuripalvelimet voisivat ottaa yhteyttä palvelimeen omien käyttäjien autentikointia varten, heidän vierailessaan muilla kampuksilla, tähän tiedostoon on määriteltävä juuripalvelimet:

```
client ftlr_funet_fi{
   ipaddr = 193.166.5.150
   netmask = 32
   secret = MfhurewrbDm886PR     # Tämä salasana on oltava yhteinen molemmilla palvelimilla.
   # Ota tähän liittyen yhteyttä noc@funet.fi:hin.
   require_message_authenticator = no
   nastype = other
   virtual_server =  eduroam
}

client ftlr2_funet_fi {
   ipaddr = 193.166.4.105
   netmask = 32
   secret = MfhurewrbDm886PR    # Tämä salasana on oltava yhteinen molemmilla palvelimilla.
   # Ota tähän liittyen yhteyttä noc@funet.fi:hin.
   require_message_authenticator = no
   nastype = other
   virtual_server =  eduroam
}
```

proxy.conf

home_server_pool EDUROAM-FTLR-lohkon jälkeen lisätään lohko, joka määrittelee, että omat käyttäjät (mydomain.fi) muodostavat erikoistapauksen:

```
realm mydomain.fi {
        nostrip
}
```

/sites-available/eduroam

Tämän tiedoston (virtuaalipalvelimen) authorize- ja authenticate-haarojen loppuun on määriteltävä EAP:ia käytettäväksi:

```
authorize {
          auth_log
          suffix
          eap
}
authenticate {
          eap
}
```

eap.conf

Käytettävät EAP-metodit määritellään eap.conf-tiedostossa ja suositellaan käytettäväksi PEAP ja TTLS. Jotta nämä toimisi, täytyy määritellä myös TLS:n asetukset! Konfiguroinnin yksityiskohdat näkyvät liitetiedostosta.

users

Tässä tiedostossa on käyttäjätunnukset ja salasanat niille käyttäjille, jotka autentikoidaan paikallisesti, eli mydomain.fi:n käyttäjät. Alhaalla olevassa esimerkissä määritellään yksi käyttäjä, jolla on salasana cleartext-muodossa ja yksi käyttäjä, jonka salasana on NT-hash-muodossa:

```
mina@mydomain.fi Cleartext-Password := "hello"

#sina@mydomain.fi NT-Password := "goodbye"
sina@mydomain.fi NT-Password := "CAC331BC07EC8830CA1563716472A22C"
```

Käyttäjätili, jonka salasana on clear-text-muodossa, soveltuu hyvin TTLS-PAP-menetelmän testaukseen ja NT-hash-salasanalla varustetulla käyttäjätilillä voidaan testata TTLS-MSCHAPv2- ja PEAP-MSCHAPv2 -menetelmät.

/sites-available/eduroam-inner-tunnel ja /sites-enabled/eduroam-inner-tunnel

eap.conf-tiedoston PEAP- ja TTLS-haaroissa määriteltiin käytettäväksi virtuaalipalvelin eduroam-inner-tunnel siinä tapauksessa, että käyttäjä autentikoidaan paikallisesti ja EAP-menetelmä on jompikumpi näistä. Virtuaalipalvelin luodaan samaan hakemistoon kuin eduroam-virtuaalipalvelinkin eli sites-available-hakemistoon. Konfiguroinnin yksityiskohdat selviävät liitetiedostosta, mutta tärkeimmät lohkot ovat autorize- ja authenticate-lohkot, joissa määritellään käytettävät autentikointitavat ja -menetelmät:

```
authorize {

    auth_log
    files
    mschap
    pap
    eap {

        ok = return

    }
}
authenticate {
    Auth-Type PAP{
        pap
    }
    Auth-Type MS-CHAP{
        mschap
    }
   #  Allow EAP authentication.
   eap
}
```

Näihin lohkoihin määritellään myös vastaavalla tavalla LDAP, jos liitetään palvelin LDAP-tietokantaan. LDAP:in tapauksessa täytyy myös muokata radius.conf-tiedostoa.

Lisäksi on luotava symbolinen linkki eduroam-inner-tunnel-virtuaalipalvelimeen *ln -s ../sites-available/eduroam-inner-tunnel eduroam-inner-tunnel* -komennolla sites-enabled-hakemistosta.

Testaus

RADIUS-viestien välittämistä eteenpäin voidaan edelleen testata paikallisesti käyttäen klienttiä localhost. Paikallista autentikointia on parasta testata liittämällä palvelimeen tukiasema ja testaamalla langattoman verkon yli käyttäen esimerkiksi Linuxin wpa_supplikanttia. Näin voidaan varmistaa, että palvelin toimii oikeissa olosuhteissa.

Yhteenveto

Tällä tavalla konfiguroituna palvelin välittää onnistuneesti autentikointipyyntöjä eteenpäin ja autentikoi paikallisesti mydomain.fi:n käyttäjät joko TTLS-PAP:illa, TTLS-MSCHAPv2:lla tai PEAP-MSCHAPv2:lla.

Varmenteen käyttöönottaminen ja salakirjoituksen säätäminen

FreeRADIUS sisältää testausvarmenteita konfiguroinnin ja ensimmäisen testauksen helpottamiseksi, mutta näitä varmenteita ei käytetä tuotannossa. Tuotantoa varten voidaan joko luoda itseallekirjoitettu varmenne tai tilata palvelinvarmenne Funetin varmennepalvelun kautta. Itseallekirjoitetun varmenteen luomiseen löytyy ohjeita /etc/raddb/certs-hakemiston README-tiedostosta. Paras tietoturvan taso on saavutettavissa omalla, pelkästään eduroam-käyttöön perustetulla CA:lla.

Varmenne otetaan käyttöön muokkaamalla eap.conf-tiedoston TLS-lohkoa seuraavalla tavalla (katso vaihtoehtoisesti liitetiedostoa eap.conf).

```
tls {
    private_key_password = SaLaSanaSi
    private_key_file = /pathToCert/my_server.pem
    certificate_file = /pathToCert/my_server.crt
    CA_file = /pathToCert/cert_chain.pem

    # make_cert_command = "${certdir}/bootstrap" Tämän rivin avulla luodaan testausvarmenteet ja se on tässä
vaiheessa kommentoitava pois.

}
```

cert_chain.pem-tiedostossa on varmenneketju, joka on määritettävä, jos juurivarmenteen ja palvelinvarmenteen välillä esiintyy välivarmenteita, kuten tämänhetkisessä Funet-varmennepalvelun myöntämissä varmenteissa. cert_chain.pem-tiedosto löytyy liitetiedostoista. cert_chain.pem-tiedosto on luotu väl ivarmenteista seuraavalla komennolla:

```
cat AddTrust_External_CA_Root.pem UTN-USERFirst-Hardware.pem TERENA_SSL_CA.pem > cert_chain.pem
```

Kun palvelin on konfiguroitu tällä tavalla, autentikointi onnistuu jos käyttäjän suplikantissa on valittu joku välivarmenteista luotetuksi varmenteeksi ja palvelimen nimi on määritetty. Autentikointi ei onnistu, jos palvelimen nimi on väärin tai mikään välivarmenteista ei ole määritetty luotetuksi varmenteeksi. Koska suplikantissa voidaan määritellä suhteellisen paljon autentikoinnin tietoturvallisuudesta, autentikointi onnistuu vaikka varmenne ja palvelimen nimi jätetään määrittelemättä. IT-tuen tulisi kuitenkin mahdollisuuksien mukaan varmistaa, että käytetty varmenne ja palvelimen nimi on määritetty käyttäjien suplikanteissa.

Tässä vaiheessa on myös hyvä säätää salakirjoitus kuntoon. Oletuksena hyväksytään myös vanhempia heikompia menetelmiä, kuten MD5, ja tästä syystä eap.conf-tiedoston cipher_list tulisi muuttaa "DEFAULT"-arvosta esim. seuraavaksi:

```
cipher_list = "HIGH:RC4-SHA:!ADH:!MD5"
```

Yllä olevalla määrittelyllä hyväksytään avaimet, jotka ovat 128 bittiä tai pidempiä (HIGH) sekä RC4-SHA-menetelmät. MD5 ja anonyymit Diffie-Hellmanit kielletään.

Virtuaaliverkkojen huomioonottaminen

Jos on käytössä virtuaaliverkkoja (Virtual LAN, VLAN), käyttäjiä voidaan asettaa eri VLAN-verkkoihin, jos esimerkiksi halutaan antaa oman organisaation käyttäjille enemmän oikeuksia kun vierailijoille. Virtuaaliverkot on luonnollisesti määriteltävä myös WLAN-kontrolleriin sekä lähiverkon muihin elementteihin, kuten kytkimiin. Voi olla järkevää määritellä vierailijoille tarkoitettu VLAN oletukseksi ja laittaa omasta realmista tulevat käyttäjät (user@myorganisation.fi) heille tarkoitetulle VLAN:eihin autentikoinnin onnistuttua. Tässä yhteydessä on myös otettava huomioon, että omasta realmista tulevat käyttäjät laitetaan heille tarkoitettuun VLANiin vain jos he ovat oman organisaationsa verkossa. Heidän vieraillessaan vieraileva organisaatio päättää, mihin VLANiin heidät laitetaan eikä VLAN-määrityksiä lähetetä oman organisaation ulkopuolelle.

Omat käyttäjät laitetaan heille tarkoitettuun VLANiin lisäämällä seuraavat rivit **/sites-available/eduroam-inner-tunnel**-tiedoston post-auth-lohkoon:

```
post-auth {
.
.
.
   if ("%{User-name}" =~ /@myorganisation.fi$/ && "%{NAS-IP-Address}" =~ /^xyz.zyx.xzy./){
   # Tässä tapauksessa organisaation tukiasemat käyttävät xyz.zyx.xzy.0/24-avaruuden osoitteita.
      update reply {
         Tunnel-Type := 13
         Tunnel-Medium-Type := 6
         Tunnel-Private-Group-ID := haluttu_VLAN_id_nr
         }
     }
}
```

Apua

Netistä löytyy hyvin ohjeita vianselvitykseen. Myös *freeradius-users*-sähköpostilista voi olla hyödyllinen. Wenche Backman-Kamila (CSC/Funet) auttaa myös tarvittaessa.

Eteenpäin

Ohjeita siitä, miten FreeRADIUS liitetään ulkoiseen tietokantaan, löytyy FreeRADIUSLiittaminenTietokantaan-sivulta

Kommentteja

Ole hyvä ja lisää kommentteja tai omaa tekstiä.

Tarvittavat varmenteet saa helpoiten käyttäjien päätelaitteisiin tarjoamalla heille käyttöjärjestelmäkohtaiset eduroam-asennuspaketit. Asennuspaketit voi luoda eduroam CAT -palvelussa (https://cat.eduroam.org), johon korkeakoulun verkkoylläpito voi pyytää käyttäjätunnukset Funetilta. Asennusohjelmat luovat päätelaitteeseen eduroam-verkon kaikkine tarvittavine asetuksineen ja varmenteineen, mikä helpottaa eduroamin käyttöönottoa loppukäyttäjän näkökulmasta merkittävästi. Erityisesti Windows-koneet on helppo konfiguroida väärin. – Tomi Salmi (CSC/Funet)