Roteiro de instalação do servidor IdP do eduroam com FreeRadius v3

Instalando e configurando os serviços (FreeRadius) **>>**

Atenção!

V2 incompatível com V3 para arquivos de configuração.

Executar os comandos:

apt-get purge libfreeradius2 freeradius-common; rm -rf /etc/freeradius

Instituição

execute os comandos com *root* ou *sudo*

1. Instalação de pacotes necessários

1.1 Instalação do FreeRadius

1.1.0 Crie um diretório de instalação e entre neste diretório

```
mkdir ~/freeradius
cd ~/freeradius
```

A arquitetura de referência do pacote de instalação é de **64 bits.**

1.1.1 Faça o download do FreeRadius armazenado no SVN da RNP:

```
wget https://svn.rnp.br/repos/eduroam/freeradius3.0.15.zip
```

`1.1.1.1 Faça o download do unzip`

```
apt-get install unzip
```

`1.1.1.2 Descompacte o arquivo baixado do SVN`

```
unzip freeradius3.0.15.zip
```

1.1.2 Instale as dependências listadas:

```
apt-get install libpcap0.8 libcurl3 libhiredis0.13 libiodbc2 libmemcached11 libmysqlclient20 libpq5 libpython2.
7 libpython2.7-minimal libpython2.7-stdlib libtalloc2 libwbclient0 libykclient3 libyubikey0 make mysql-common
libc6 libldap-2.4-2 libssl1.0.0 lsb-base libgdbm3 libpam0g libperl5.22 libreadline6 libsqlite3-0 libtalloc2
libwbclient0 ssl-cert adduser
```

1.1.3 Instale os pacotes na ordem:

```
dpkg -i freeradius-common_3.0.15+git_all.deb
```

```
dpkg -i freeradius-config_3.0.15+git_amd64.deb
```

```
dpkg -i libfreeradius3_3.0.15+git_amd64.deb
```

```
dpkg -i freeradius_3.0.15+git_amd64.deb
```

1.1.4 Reinicie o serviço

```
service freeradius restart
```

1.1.5 Instale todos os demais pacotes:

```
dpkg -i *.deb
```

2. Configurando o FreeRadius

Substitua as variáveis:

<SENHA_LOCALHOST>: senha para 127.0.0.1 (ou localhost) cadastrada no clients.conf

<SENHA_CLIENTE>: a senha do cliente (que é o servidor da federação) para se conectar ao FreeRadius local (IdP)

<SENHA_FEDERACAO>: a senha no servidor (da federação) para FreeRadius local (IdP) se conectar

2.1 Acessando a pasta do FreeRadius 3

```
cd /etc/freeradius/
```

2.2Alterando senha principal de acesso do cliente localhost ao FreeRadius 3 no "clients.conf"

```
vim clients.conf
# altere o <SENHA_LOCALHOST>
```

2.2.1 O conteúdo do arquivo deve ser

```
client localhost {
        ipaddr = 127.0.0.1
        proto = *
        secret = <SENHA_LOCALHOST>
        require_message_authenticator = no
        limit {
                max_connections = 16
                lifetime = 0
                idle_timeout = 30
        }
}
client localhost_ipv6 {
        ipv6addr        = ::1
        secret          = <SENHA_LOCALHOST>
}
```

2.2.2 O arquivo pode ser baixado em:

```
wget https://svn.rnp.br/repos/eduroam/roteiros/atividade2/etc_freeradius_clients.conf -O /etc/freeradius
/clients.conf
```

2.3 Habilitando o usuário "bob" para autenticação local por arquivo "files"

```
vim users
# adicionando o usuário "bob" com senha "hello".
# Esse usuário já existe! Descomente as linhas 87 e 88 do arquivo padrão
```

2.3.1 O conteúdo (excluindo todas linhas comentadas) do arquivo deve ser

```
bob     Cleartext-Password := "hello"
        Reply-Message := "Hello, %{User-Name}"
DEFAULT Framed-Protocol == PPP
        Framed-Protocol = PPP,
        Framed-Compression = Van-Jacobson-TCP-IP
DEFAULT Hint == "CSLIP"
        Framed-Protocol = SLIP,
        Framed-Compression = Van-Jacobson-TCP-IP
DEFAULT Hint == "SLIP"
        Framed-Protocol = SLIP
```

2.3.2 O arquivo pode ser baixado em

```
wget https://svn.rnp.br/repos/eduroam/roteiros/atividade2/etc_freeradius_users -O /etc/freeradius/users
```

2.4 Reinicie o serviço

```
service freeradius restart
```

3. Testes de autenticação

3.1 Teste de autenticação local com PAP

```
# teste de autenticação local
radtest bob hello localhost 0 <SENHA_LOCALHOST>
```

Retorno esperado do Teste 01

```
Sent Access-Request Id 35 from 0.0.0.0:36069 to 127.0.0.1:1812 length 73
        User-Name = "bob"
        User-Password = "hello"
        NAS-IP-Address = 200.130.15.24
        NAS-Port = 0
        Message-Authenticator = 0x00
        Cleartext-Password = "hello"
Received Access-Accept Id 35 from 127.0.0.1:1812 to 0.0.0.0:0 length 32
        Reply-Message = "Hello, bob"
```

3.2 Teste de autenticação local com MSCHAP

```
# teste de autenticação local
radtest -t mschap bob hello localhost 0 <SENHA_LOCALHOST>
```

3.2.1 Retorno esperado do Teste 01

```
Sent Access-Request Id 72 from 0.0.0.0:52444 to 127.0.0.1:1812 length 129
        User-Name = "bob"
        MS-CHAP-Password = "hello"
        NAS-IP-Address = 127.0.1.1
        NAS-Port = 0
        Message-Authenticator = 0x00
        Cleartext-Password = "hello"
        MS-CHAP-Challenge = 0x9cad4336729e48a8
        MS-CHAP-Response =
0x000100000000000000000000000000000000000000000000945b65afa00a7d7103b3379548b46d5662a78d1a721f80f0
Received Access-Accept Id 72 from 127.0.0.1:1812 to 0.0.0.0:0 length 96
        Reply-Message = "Hello, bob"
        MS-CHAP-MPPE-Keys = 0xfda95fbeca288d44ac0782e2de2337dee40e54ee732c1af5
        MS-MPPE-Encryption-Policy = Encryption-Allowed
        MS-MPPE-Encryption-Types = RC4-40or128-bit-Allowed
```

3.3 Configurando o "proxy.conf" para responder por "*<DOMINIO>*"

Instituição

A instituição no Brasil deve ser algo como: ufjf.br aqui representado por <DOMINIO>

Substitua as ocorrências de <DOMINIO> pelo domínio de sua instituição - sem o .br por enquanto. Exemplo: ufjf, uff, unipampa etc

3.3.1 Configurando a instituição para responder por <DOMINIO> e também para redirecionar domínios desconhecidos para o proxy da federação.

```
# a federação deve adicionar o seu IP (do seu IdP) como cliente no "clients.conf"
# e seu IdP também deve aceitar a federação como cliente. Sendo assim,
# adicione a federação como cliente no "clients.conf"
vim clients.conf
```

3.3.2 O conteúdo do arquivo deve ser, para o clients.conf

clients.conf

```
# Este bloco libera a consulta em localhost
client localhost {
        ipaddr = 127.0.0.1
        proto = *
        secret = <SENHA_LOCALHOST>
        require_message_authenticator = no
        shortname = localhost
        nas_type         = other        # localhost isn't usually a NAS...
        limit {
                max_connections = 16
                lifetime = 0
                idle_timeout = 30
        }
}

# Este bloco libera a consulta em localhost usando IPv6
client localhost_ipv6 {
        ipv6addr        = ::1
        secret          = <SENHA_LOCALHOST>
}

# Este bloco libera a consulta vinda de um cliente da federação
# RPS01
client rps01 {
        ipaddr          = rps01.eduroam.org.br
        secret          = <SENHA_FEDERACAO>
        shortname       = rps01_federacao
}

# RPS02
client rps02 {
        ipaddr          = rps02.eduroam.org.br
        secret          = <SENHA_FEDERACAO>
        shortname       = rps02_federacao
}
```

3.3.2.1 O arquivo pode ser baixado em

```
wget https://svn.rnp.br/repos/eduroam/roteiros/atividade2/etc_freeradius_clients2.conf -O /etc/freeradius
/clients.conf
```

3.3.4 Configurando o arquivo proxy.conf

```
# configurando o roaming e a instituição para responder por "<DOMINIO>"
vim proxy.conf
```

3.3.4.1 Conteúdo deve ser, para proxy.conf

proxy.conf

```
proxy server {
        default_fallback = no
}

home_server localhost {
        type = auth
        ipaddr = 127.0.0.1
        port = 1812
        secret = <SENHA_LOCALHOST>
        response_window = 20
        zombie_period = 40
```

```
        revive_interval = 120
        status_check = status-server
        check_interval = 30
        check_timeout = 4
        num_answers_to_alive = 3
        max_outstanding = 65536
        coa {
                irt = 2
                mrt = 16
                mrc = 5
                mrd = 30
        }
        limit {
              max_connections = 16
              max_requests = 0
              lifetime = 0
              idle_timeout = 0
        }
}
# FEDERACAO EDUROAM RPS01
home_server rps01 {
        type = auth
        ipaddr = rps01.eduroam.org.br
        port = 1812
        secret = <SENHA_FEDERACAO>
        response_window = 20
        zombie_period = 40
        revive_interval = 120
        status_check = status-server
        check_interval = 30
        check_timeout = 4
        num_answers_to_alive = 3
        max_outstanding = 65536
        coa {
                irt = 2
                mrt = 16
                mrc = 5
                mrd = 30
        }
        limit {
              max_connections = 16
              max_requests = 0
              lifetime = 0
              idle_timeout = 0
        }
}
# FEDERACAO EDUROAM RPS02
home_server rps02 {
        type = auth
        ipaddr = rps02.eduroam.org.br
        port = 1812
        secret = <SENHA_FEDERACAO>
        response_window = 20
        zombie_period = 40
        revive_interval = 120
        status_check = status-server
        check_interval = 30
        check_timeout = 4
        num_answers_to_alive = 3
        max_outstanding = 65536
        coa {
                irt = 2
                mrt = 16
                mrc = 5
                mrd = 30
        }
        limit {
              max_connections = 16
              max_requests = 0
              lifetime = 0
              idle_timeout = 0
```

```
        }
}

home_server_pool my_localhost {
        type            = fail-over
        home_server     = localhost
}

home_server_pool my_auth_failover {
        type = fail-over
        home_server = rps01
        home_server = rps02
}

realm DEFAULT {
        auth_pool = my_auth_failover
        nostrip
}
realm LOCAL {
}

realm NULL {
        secret          = eduroam
}
# SUBSTITUA <DOMINIO> PELA SUA INSTITUICAO, SEM O BR
# EXEMPLO: <DOMINIO> -> ufjf
realm "~(.*\.)*<DOMINIO>\.br$" {
        auth_pool       = my_localhost
        secret          = <SENHA_LOCALHOST>
}
```

3.3.4.2 O arquivo pode ser baixado em

```
wget https://svn.rnp.br/repos/eduroam/roteiros/atividade2/etc_freeradius_proxy.conf -O /etc/freeradius/proxy.
conf
```

3.3.5 Reinicie o serviço para que a nova configuração tenha efeito.

```
service freeradius restart
```

3.4 Teste de autenticação local com domínio utilizando PAP

Realize o teste de acesso local com domínio

```
# teste local com domínio "<DOMINIO>.br"
radtest bob@<DOMINIO>.br hello localhost 0 <SENHA_LOCALHOST>
```

3.5 Teste de autenticação local com domínio utilizando MSCHAP

Realize o teste de acesso local com domínio

```
# teste local com domínio "<DOMINIO>.br"
radtest -t mschap bob@<DOMINIO>.br hello localhost 0 <SENHA_LOCALHOST>
```

3.6 Teste de autenticação em roaming

Agora faça um teste de roaming (teste de usuário de outra instituição visitando a sua)

```
# teste de roaming
radtest bob@rnp.br hello localhost 0 <SENHA_LOCALHOST>


Retorno esperado:

Sent Access-Request Id 35 from 0.0.0.0:36069 to 127.0.0.1:1812 length 73
        User-Name = "bob"
        User-Password = "hello"
        NAS-IP-Address = 200.130.15.24
        NAS-Port = 0
        Message-Authenticator = 0x00
        Cleartext-Password = "hello"
Received Access-Accept Id 35 from 127.0.0.1:1812 to 0.0.0.0:0 length 32
        Reply-Message = "Hello, bob"
```

4. Configuração dos túneis virtuais

4.1 Configure os túneis FreeRadius com os seguintes conteúdos

4.1.1 Criando os links simbólicos para os arquivos na pasta

```
cp /etc/freeradius/sites-available/default /etc/freeradius/sites-available/default.original
cp /etc/freeradius/sites-available/inner-tunnel /etc/freeradius/sites-available/inner-tunnel.original

cd /etc/freeradius/sites-enabled/
rm default inner-tunnel
```

4.1.2 O arquivo */etc/freeradius/sites-enabled/inner-tunnel* deve ter a seguinte configuração

/etc/freeradius/sites-enabled/inner-tunnel

```
server inner-tunnel {
listen {
        type = auth
        ipaddr = *
        port = 18120
        limit {
              max_connections = 16
              lifetime = 0
              idle_timeout = 30
        }
}
authorize {
        suffix
        filter_username
        preprocess
        chap
        mschap
        digest

        # Seu realm - Exemplo: @ufjf.br deve ser "ufjf.br"
              if ( Realm == "<DOMINIO>.br" ) {
                 update control {
                    &Proxy-To-Realm := LOCAL
                 }
              }

        eap {
                ok = return
        }
        files
```

```
        -sql
        -ldap
        expiration
        logintime
        pap
}
authenticate {
        Auth-Type PAP {
                pap
        }
        Auth-Type CHAP {
                chap
        }
        Auth-Type MS-CHAP {
                mschap
        }
        mschap
        digest
        eap
}
preacct {
        preprocess
        acct_unique
        files
}
accounting {
        detail
        unix
        -sql
        exec
        attr_filter.accounting_response
}
session {
}
post-auth {
        update {
                &reply: += &session-state:
        }
        -sql
        -ldap
        exec
        remove_reply_message_if_eap
        Post-Auth-Type REJECT {
                -sql
                attr_filter.access_reject
                eap
                remove_reply_message_if_eap
        }
}
pre-proxy {
}
post-proxy {
        eap
}
}
```

4.1.2.1 O arquivo pode ser baixado em

```
wget https://svn.rnp.br/repos/eduroam/roteiros/atividade2/etc_freeradius_sites_enabled_inner_tunnel -O /etc
/freeradius/sites-available/inner-tunnel
```

4.1.3 O arquivo */etc/freeradius/sites-enabled/default* deve ter a seguinte configuração

/etc/freeradius/sites-enabled/default

```
server default {
listen {
        type = auth
        ipaddr = *
        port = 0
        limit {
              max_connections = 16
              lifetime = 0
              idle_timeout = 30
        }
}
listen {
        ipaddr = *
        port = 0
        type = acct
        limit {
        }
}
listen {
        type = auth
        ipv6addr = ::        # any.  ::1 == localhost
        port = 0
        limit {
              max_connections = 16
              lifetime = 0
              idle_timeout = 30
        }
}
listen {
        ipv6addr = ::
        port = 0
        type = acct
        limit {
        }
}
authorize {
        suffix
        filter_username
        preprocess
        chap
        mschap
        digest
        # Seu realm - exemplo: "ufjf.br"
        if ( Realm == "<DOMINIO>.br" ) {
                 update control {
                    &Proxy-To-Realm := LOCAL
                 }
        }
        eap {
                ok = return
        }
        files
        -sql
        -ldap
        expiration
        logintime
        pap
}
authenticate {
        Auth-Type PAP {
                pap
        }
        Auth-Type CHAP {
                chap
        }
        Auth-Type MS-CHAP {
                mschap
        }
        mschap
        digest
        eap
```

```
}
preacct {
        preprocess
        acct_unique
        files
}
accounting {
        detail
        unix
        -sql
        exec
        attr_filter.accounting_response
}
session {
}
post-auth {
        update {
                &reply: += &session-state:
        }
        -sql
        -ldap
        exec
        remove_reply_message_if_eap
        Post-Auth-Type REJECT {
                -sql
                attr_filter.access_reject
                eap
                remove_reply_message_if_eap
        }
}
pre-proxy {
}
post-proxy {
        eap
}
}
```

4.1.3.1 O arquivo pode ser baixado em

```
wget https://svn.rnp.br/repos/eduroam/roteiros/atividade2/etc_freeradius_sites_enabled_default -O /etc
/freeradius/sites-available/default
```

4.1.3.2 Criando os links simbólicos

```
cd /etc/freeradius/sites-enabled/
ln -s ../sites-available/default .
ln -s ../sites-available/inner-tunnel .
```

4.2 Reiniciando o serviço

```
service freeradius restart
```

4.3 Em caso de erro, rode no modo debug

```
radiusd -fxx -l stdout
ou
freeradius -fxx -l stdout
```

Refaça os testes 3.3, 3.4 e 3.5