IdP eduroam FreeRadius v3 - Habilitando RadSec TCP/TLS

Antes de começar

Informe ao administrador do FLR (federação) que deseja utilizar TLS em sua comunicação.
Ele irá gerar seus certificados e enviá-los a você!

Será necessário informar seu endereço IP ou FQDN a ele (que será correspondente ao CN de seu certificado)

1. Configuração do FreeRadius v3 com suporte à comunicação TLS **>>**

Variáveis utilizadas no roteiro:

<DOMINIO>: seu realm

<FQDN>: seu nome de certificado

<CA>: nome do arquivo da CA (e.g. rnp-ca)

<SENHA_CERTIFICADO>: a senha do certificado

<SENHA_LOCALHOST>: senha para 127.0.0.1 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

<USUARIO_EDUROAM>: usuário cadastrado na base na instituição

<SENHA_USUARIO_EDUROAM>: senha desse usuário

1.1 Passos e configuração dos arquivos

1.1.1 Toda a configuração é feita, basicamente, em um único arquivo do FreeRadius v3. Esse arquivo existirá na pasta */etc/freeradius/sites-enabled*. A pasta que contém os arquivos que chamamos de túnel (internos ao FreeRadius v3).

Sendo assim, entre na pasta do FreeRadius3, e em *sites-available* crie o arquivo que será responsável pela comunicação TLS, o *radsec*:

```
# entre na pasta do freeradius
cd /etc/freeradius/

# entre na pasta de configuração dos sites
cd sites-available

# crie um arquivo chamado radsec
touch radsec

# edite o arquivo
vim radsec
```

1.1.2 O arquivo radsec deverá ter o conteúdo **(lembrando de alterar a nome de domínio, certificados e senhas, quando necessário):**

/etc/freeradius/sites-available/radsec (Clique abaixo para expandir o código da configuração):

```
listen {
    ipaddr = *
    port = 2083
    type = auth

    proto = tcp
        virtual_server = default

    clients = radsec
        limit {
              max_connections = 0
              lifetime = 0
              idle_timeout = 3600
              }
    tls {
        certdir = ${confdir}/certs/radsec
        cadir = ${confdir}/certs/radsec

        private_key_password = <SENHA_CERTIFICADO>
        private_key_file = ${certdir}/<FQDN>.key

        certificate_file = ${certdir}/<FQDN>.crt
        ca_file = ${cadir}/<CA>.crt

        dh_file = ${certdir}/dh
        random_file = ${certdir}/random

        fragment_size = 8192
        include_length = yes
        cipher_list = "DEFAULT"
        cache {
              enable = yes
              lifetime = 24 # hours
              max_entries = 255
        }

        require_client_cert = yes

        verify {
        }
    }
}


listen {
    ipv6addr = ::
    port = 2083
    type = auth
    proto = tcp
    clients = radsec

         limit {
              max_connections = 0
              lifetime = 0
              idle_timeout = 600
        }
    tls {
        certdir = ${confdir}/certs/radsec
        cadir = ${confdir}/certs/radsec
        private_key_password = <SENHA_CERTIFICADO>
        private_key_file = ${certdir}/<FQDN>.key
        certificate_file = ${certdir}/<FQDN>.crt
        ca_file = ${cadir}/<CA>.crt
```

```
        dh_file = ${certdir}/dh
        random_file = ${certdir}/random
        fragment_size = 8192
        include_length = yes

                cipher_list = "DEFAULT"
        cache {
              enable = yes
              max_entries = 255
        }
        require_client_cert = yes
        verify {
        }
    }
}

clients radsec {
         limit {
              max_connections = 0
              lifetime = 0
              idle_timeout = 3600
        }
    client 127.0.0.1 {
        ipaddr = 127.0.0.1
        proto = tls
        secret = <SENHA_LOCALHOST>
    }
    client rps01 {
        ipaddr = rps01.eduroam.org.br
        proto = tls
        secret = <SENHA_CLIENTE>
        limit {
              max_connections = 0
              lifetime = 0
              idle_timeout = 3600
        }
    }
    client rps02 {
        ipaddr = rps02.eduroam.org.br
        proto = tls
        secret = <SENHA_CLIENTE>
        limit {
              max_connections = 0
              lifetime = 0
              idle_timeout = 3600
        }
    }
}


# local test listener for debug (present by default)
listen {
       ipaddr = 127.0.0.1
       port = 4000
       type = auth
}

# As linhas abaixo fazem a autenticação na federacao Eduroam
home_server rps01SRV {
    ipaddr = rps01.eduroam.org.br
    port = 2083
    type = auth
    secret = <SENHA_FEDERACAO>
    proto = tcp
    status_check = none

    tls {
        certdir = ${confdir}/certs/radsec
        cadir = ${confdir}/certs/radsec

        private_key_password = <SENHA_CERTIFICADO>
```

```
        private_key_file = ${certdir}/<FQDN>.key

        certificate_file = ${certdir}/<FQDN>.crt

        ca_file = ${cadir}/<CA>.crt

        dh_file = ${certdir}/dh
        random_file = ${certdir}/random

        #fragment_size = 1024
        fragment_size = 1500

        include_length = yes
       CA_path = ${cadir}
        cipher_list = "DEFAULT"
    }
}

# No piloto não existe rps02, apenas 1 bloco para homolog-flr.rnp.br
home_server rps02SRV {
    ipaddr = rps02.eduroam.org.br
    port = 2083
    type = auth
    secret = <SENHA_FEDERACAO>
    proto = tcp
    status_check = none

    tls {
        certdir = ${confdir}/certs/radsec
        cadir = ${confdir}/certs/radsec

        private_key_password = <SENHA_CERTIFICADO>
        private_key_file = ${certdir}/<FQDN>.key

        certificate_file = ${certdir}/<FQDN>.crt

        ca_file = ${cadir}/<CA>.crt

        dh_file = ${certdir}/dh
        random_file = ${certdir}/random

        #fragment_size = 1024
        fragment_size = 1500

        include_length = yes
       CA_path = ${cadir}
        cipher_list = "DEFAULT"
    }
}
home_server_pool BR {
         type = fail-over
         home_server = rps01SRV
                 home_server = rps02SRV
}


realm "~.+$" {
       auth_pool = BR
       nostrip
}
```

1.1.2.1 O arquivo pode ser baixado em

```
wget https://svn.rnp.br/repos/eduroam/roteiros/atividade4/etc_freeradius_sites_enabled_radsec -O /etc/freeradius
/sites-available/radsec
```

1.1.3 Feita a edição do arquivo, salve-o. Agora vamos criar o link simbólico desse arquivo no *sites-enabled*. Só assim ele realmente será utilizado pelo FreeRadius3.

```
cd /etc/freeradius/sites-enabled/
ln -s ../sites-available/radsec radsec
```

1.1.4 Caso você esteja utilizando o arquivo padrão exibido nesse texto, com mesma senha para o certificado de sua instituição, **é substituir as ocorrência de <FQDN> em seu arquivo pelo nome da sua instituição**. Isso pode ser feito pelo editor VIM digitando \[esc] dois pontos etc. Como segue no exemplo:

```
Alterar todos os <FQDN> para seu FQDN - leve em consideração o nome dos arquivos enviados pelo administrador
eduroam.
:%s/<FQDN>/seufqdn/g
```

1.2 Configurando os certificados na pasta radsec

> **Atenção!**
>
> Você irá receber do responsável pela federação 2 arquivos: **<FQDN>.crt** e **<FQDN>.key** além do **ca.crt** (o certificado da CA) e deverá seguir os passos da listagem de comandos abaixo para que a pasta de certificados funcione perfeitamente.

Criando pasta, arquivos e copiando certificados para RadSec

```
# Entre na pasta
cd /etc/freeradius/certs

# Crie a pasta radsec
mkdir /etc/freeradius/certs/radsec

# Entre no diretório radsec
cd /etc/freeradius/certs/radsec

# Criar os arquivos random e DH na pasta
dd if=/dev/urandom of=./random count=10 && openssl dhparam -out dh 2048

# Agora copie os certificados e chave. Supondo que os arquivos estejam armazenados no /tmp
cp /tmp/rnp-ca.crt /tmp/<FQDN>.crt /tmp/<FQDN>.key /etc/freeradius/certs/radsec/
```

Finalizados os passos de configuração, o modo debug de execução deve ser executado pela linha de comando (uma vez que agora seu FreeRadius utiliza TCP e TLS):

```
service freeradius stop
freeradius -fxx -l stdout
```

1.3 Testes

Se tudo estiver ok, prossiga com o teste de autenticação para um usuário em roaming

Primeiro, saia do modo debug de execução do FreeRadius e inicie o serviço

```
service freeradius start
```

Ou mantenha o modo debug em execução e abra um novo terminal para o teste de roaming

1.3.1 Teste de roaming com TLS

```
# teste para usuário de instituição participante do eduroam
radtest <USUARIO_EDUROAM>@<DOMINIO_VISITANTE>.br <SENHA_USUARIO_EDUROAM> localhost 0 <SENHA_LOCALHOST>
```

1.3.2 A saída esperada é:

```
Sent Access-Request Id 186 from 0.0.0.0:59114 to 127.0.0.1:1812 length 90
        User-Name = "<USUARIO_EDUROAM>@<DOMINIO_VISITANTE>.br"
        User-Password = "<SENHA_USUARIO_EDUROAM>"
        NAS-IP-Address = 127.0.1.1
        NAS-Port = 0
        Message-Authenticator = 0x00
        Cleartext-Password = "<SENHA_USUARIO_EDUROAM>"
Received Access-Accept Id 186 from 127.0.0.1:1812 to 0.0.0.0:0 length 20
```