

# radsecproxy Configuration for the OpenRoaming

Rev. 20230520

## Introduction

This document explains the minimum configuration of radsecproxy for setting up a RadSec endpoint of the WBA OpenRoaming. This document aims to help operators obtain the first working environment but is not intended for providing official recommendations.

The following configuration examples assume to use v1.9.3 which is the latest as of this writing. The source package is available at <https://radsecproxy.github.io/>.

The Open Source Software (OSS) version of radsecproxy can be used to add RadSec transport and Dynamic Peer Discovery (DPD) features to existing RADIUS IdP/proxy software. In this document, we assume the network configuration shown in Fig.1. For simplicity, we assume that both radsecproxy and the RADIUS server are running on the same host.

Throughout the document, the locations of the configuration files are relative to the configuration directory of radsecproxy unless otherwise mentioned. It would be convenient to put all configuration files in /etc/radsecproxy or create a symbolic link to the ./etc directory at the radsecproxy installation path.

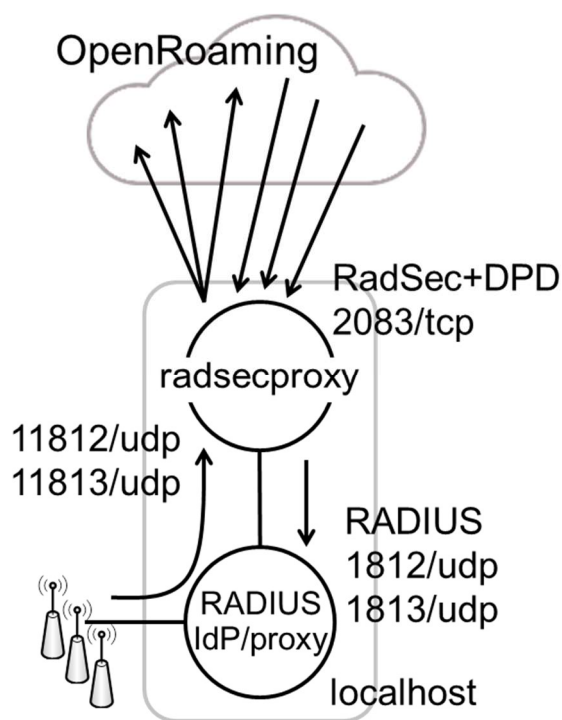


Fig.1 Using radsecproxy as a RadSec gateway for existing RADIUS IdP/proxy.

**See also:** [1] “PKI RadSec End Entity Deployment Guidelines” (WBA Members Only)

## Obtaining an OpenRoaming endpoint certificate and CA/I-CA certificates

Each operator directly connecting to the OpenRoaming network needs to receive an endpoint

certificate package from a certificate issuer, i.e. a WBA agent or a broker. The package contains the following certificates.

- (1) A server (IdP) certificate, a client (ANP) certificate, or a combined (IdP+ANP) certificate.
- (2) Root CA certificate, normally named as WBA\_OpenRoaming\_Root.pem .
- (3) All I-CA (Intermediate CA) certificates, normally the issuer's I-CA certificate and the policy certificate named as WBA\_Policy\_CA.pem .

These certificates should be in text format. In addition, the operator should retain the key file which was generated during the CSR creation [1] and its passphrase.

Suppose an operator "example.com" receives a certificate named as example.com.cer, which is a text file. The contents can be seen by a command line as follows.

```
$ openssl x509 -noout -text -in example.com.cer
```

A new certificate chain file can be created by simply concatenating the operator's certificate and the I-CA certificates.

```
$ cat example.com.cer WBA_Issuing_CA.pem WBA_Policy_CA.pem > cert-chain.pem
```

The content of the file looks like:

```
-----BEGIN CERTIFICATE-----
    <operator' s certificate>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
    <I-CA certificate WBA_Issuing_CA.pem>
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
    <I-CA certificate WBA_Policy_CA.pem>
-----END CERTIFICATE-----
```

We assume that all certificate files and the key file are stored in the radsecproxy configuration directory, e.g., /etc/radsecproxy .

Note, if the certificate and chain have been issued using WBA's Agent API, the responses may have been delivered to you using a JSON format. This means that the certificate and chain need to be reformatted back to a multi-line text format. Assuming your json certificate is in a file named "cert-json.pem", this can be achieved using the awk command:

```
$ awk ' {gsub(/\n/, "\n")} 1' cert-json.pem > cert-text.pem
```

## Configuring radsecproxy as OpenRoaming IdP (server) gateway

The only configuration file to be edited is radsecproxy.conf. By using the following configuration example, all requests from external OpenRoaming clients (ANPs) will be forwarded to the local RADIUS server which is listening on the standard ports 1812/udp (auth) and 1813/udp (acct).

```

ListenTLS *:2083
ListenUDP *:11812          # This UDP port is for local testing purposes only.
LogThreadId on
LogLevel 3

tls OR-certs {
    CACertificateFile /etc/radsecproxy/WBA_OpenRoaming_Root.pem
    CertificateFile /etc/radsecproxy/cert-chain.pem
    CertificateKeyFile /etc/radsecproxy/<keyfile.key>
    CertificateKeyPassword <passphrase>
    TLSVersion TLS1_2:      # Don't miss the last colon (:).
}

client 0.0.0.0/0 {
    type tls
    tls OR-certs
    secret radsec
    CertificateNameCheck off
}
client [::]/0 {
    type tls
    tls OR-certs
    secret radsec
    CertificateNameCheck off
}

# This client block is for local testing only and may be removed on a production
system.
client 127.0.0.1/32 {
    type udp
    secret testing123        # Adjust it to the local RADIUS server setting.
}

server localproxy {
    type udp
    host localhost
    secret testing123        # Adjust it to the local RADIUS server setting.
    statusServer on          # Turn it on if RADIUS server supports it.
}

server localproxy-acct {
    type udp
    host localhost
    port 1813
    secret testing123        # Adjust it to the local RADIUS server setting.
    statusServer on          # Turn it on if RADIUS server supports it.
}

realm /@.*\..example\..com$/ { # Fix the regex accordingly to catch all your realms
here.
    server localproxy
    accountingServer localproxy-acct
    accountingResponse on

```

```
}
```

CertificateKeyPassword line can be omitted if the key file was generated without passphrase.

To test the functionality locally, try the following on the same host, using a test ID/PW pair.

```
$ radtest testID@example.com testPW localhost:11812 1 testing123
```

In addition, check that the TLS is configured correctly by using the openssl s\_client command as follows from an external host.

```
$ openssl s_client -showcerts <ip address of FreeRADIUS server>:2083
```

Once the above tests have been successful, i.e., you have received Access-Accept, you may ask an ANP outside your own network to test the authentication over the RadSec transport. Alternatively, you may set up your own ANP on an external network for testing.

Note: Some RadSec products, e.g. Cisco Spaces Connector, require EAP and may block non-EAP communication. Please see also eaopl\_test command for tests using EAP.

## Configuring radsecproxy as OpenRoaming ANP (client) gateway

The only configuration files to be edited are radsecproxy.conf and a new file naptr-openroaming.sh. By using the following configuration example, all requests from the local RADIUS proxy will be forwarded to external OpenRoaming servers (IdPs). Since the local RADIUS server is using the standard ports 1812/udp (auth) and 1813/udp (acct), shifted ports 11812/udp and 11813/udp are used in this example. The local RADIUS server needs to be configured accordingly.

[radsecproxy.conf]

```
ListenUDP *:11812
ListenUDP *:11813
LogThreadId on
LogLevel 3

tls OR-certs {
    CACertificateFile /etc/radsecproxy/WBA_OpenRoaming_Root.pem
    CertificateFile /etc/radsecproxy/cert-chain.pem
    CertificateKeyFile /etc/radsecproxy/<keyfile.key>
    CertificateKeyPassword <passphrase>
    TLSVersion TLS1_2:      # Don't miss the last colon (:).
}

client 127.0.0.1/32 {
    type udp
    secret testing123      # Adjust it to the local RADIUS server setting.
}
```

```

server OR-dynamic {
    type tls
    tls OR-certs
    secret radsec
    certificateNameCheck off
    dynamicLookupCommand /etc/radsecproxy/naptr-openroaming.sh
}

realm /@.+\.+$/ { # This realm block must be placed at the end if others exist.
    server OR-dynamic
    accountingServer OR-dynamic
    accountingResponse on
}

```

CertificateKeyPassword line can be omitted if the key file was generated without passphrase.

New script file for the dynamic lookup, named `naptr-openroaming.sh`, can be created based on the example script for eduroam, `tools/naptr-eduroam.sh`, included in the `radsecproxy` source tree. The easiest way is to replace all `"x-eduroam:radius.tls"` with `"aaa+auth:radius.tls.tcp"`.

Please see also <https://github.com/wireless-broadband-alliance/openroaming-config> and "Pull requests".

## Consideration on separating the IdP and ANP radsecproxy

Some operators may adopt the OpenRoaming settled model or work as an ANP for non-OpenRoaming operators. If this is the case, care must be taken in the proxy configurations.

It is recommended to separate inbound and outbound proxies by running two `radsecproxy` instances. If the proxies were combined, inbound requests would be transferred to other operators, allowing another operator to gain false authorization without a bilateral roaming agreement, intentionally or by mistake.

The secondary `radsecproxy` can be run as shown below.

```
# radsecproxy -c /etc/radsecproxy2.conf -i /var/run/radsecproxy2.pid
```

## Firewall configuration (informational)

Because there are many RadSec clients doing the DPD on the OpenRoaming network, the source addresses of these clients cannot be determined beforehand.

The RadSec endpoint of the OpenRoaming ANP needs to have port 2083/tcp open to any hosts.

For your information, if `firewalld` is used on a Linux distribution, the configuration file `/etc/firewalld/zones/public.xml` would need an additional rule:

```

<rule family="ipv4">
  <source address="0.0.0.0/0"/>
  <port port="2083" protocol="tcp"/>
  <accept/>
</rule>

```

On Ubuntu use the command:

```
$ sudo ufw allow 2083
```

Rev. 20200817	Hideaki Goto, Cityroam/eduroam	
Rev. 20200910	Hideaki Goto, Cityroam/eduroam	
Rev. 20210108	Hideaki Goto, Cityroam/eduroam	RadSec secret updated.
Rev. 20210409	Hideaki Goto, Cityroam/eduroam	Removed ambiguities in client certificate handling.
Rev. 20210719	Hideaki Goto, Cityroam/eduroam	Fixed accounting configuration. Updated firewall rule example.
Rev. 20220107	Hideaki Goto, Cityroam/eduroam	Revised for clarity. Added an explicit TLS version setting. Described a risk in IdP/ANP combined deployment.
Rev. 20220220	Hideaki Goto, Cityroam/eduroam	Added some testing stuff.
Rev. 20230304	Hideaki Goto, Cityroam/eduroam	Fixed a typo.
Rev. 20230421	Hideaki Goto, Cityroam/eduroam	Fixed the issuer's name.
Rev. 20230517	Mark Grayson, Cisco	Added JSON to TEXT format command and openssl test command.
Rev. 20230520	Hideaki Goto, Cityroam/eduroam	(merge & clean-up)