

Setting up external Radius server for RUTOS authentication

[Main Page](#) > [General Information](#) > [Configuration Examples](#) > [Router control and monitoring](#) > **Setting up external Radius server for RUTOS authentication**



Contents

- [1 Summary](#)
- [2 Topology used in this example](#)
- [3 Prerequisites](#)
- [4 Preparing Ubuntu machine](#)
 - [4.1 Installing the FreeRadius server](#)
 - [4.2 Defining a client](#)
 - [4.3 Defining user login credentials](#)
- [5 Preparing router](#)
 - [5.1 Setting a static IP for the FreeRadius server](#)
 - [5.1.1 First method](#)
 - [5.1.2 Second method](#)
 - [5.2 Creating a new RUTOS user](#)
 - [5.3 PAM package installation](#)
 - [5.4 Radius server configuration](#)
 - [5.4.1 For SSH authentication](#)
 - [5.4.2 For WebUI authentication](#)
- [6 Testing configuration](#)

Summary

In this example, we will set up a Teltonika Networks router to use a Radius server for SSH and/or WebUI authentication. We will use the *freeradius* package to set up a local Radius server on an Ubuntu virtual machine. Then we will create a new user. Lastly, we will test the configuration.

This is the idea of how a Radius server is used for RUTOS authentication:



Topology used in this example



Prerequisites

- **Router** with the ability to install the PAM package and running firmware version 7.6 or later
- **Ubuntu machine** with the ability to host a local FreeRadius server

Note: in this example Ubuntu version 22.04.3 LTS was used

Preparing Ubuntu machine

Installing the FreeRadius server

Firstly, update the package source lists and upgrade the packages to their latest version:

```
sudo apt update
sudo apt upgrade
```

Next, install the FreeRadius package:

```
sudo apt install freeradius
```

Defining a client

Client - a router that will use FreeRadius to authenticate WebUI and/or SSH users. In order to add/edit clients, we need to access the **clients.conf** file. Use your favorite text editor to edit it:

```
sudo nano /etc/freeradius/3.0/clients.conf
```

For this example, we will add the following lines in order to accept any IP address as a client:

```
client 0.0.0.0/0 {
    secret = demoscrt
    shortname = 0.0.0.0/0
}
```

Note: a specific public IP of the client can be used instead of 0.0.0.0/0

Defining user login credentials

Before we create the user's login credentials, let's create an MD5 hash and use it instead of a clear text password. We will generate a hash value of **demo123** using the following command:

```
echo -n demo123 | md5sum | awk '{print $1}'
```

We will now define credentials for user **demo**. Use your favorite text editor to edit the file **users**:

```
sudo nano /etc/freeradius/3.0/users
```

Add the name of the user, MD5 hash value of its password, and a reply message:

```
demo      MD5-Password:= "62cc2d8b4bf2d8728120d052163a77df"
          Reply-Message := "Hello, %{User-Name}"
```

Once these changes are made, start the FreeRadius service:


```
sudo /etc/init.d/freeradius start
```

Preparing router

Setting a static IP for the FreeRadius server

Firstly, we will set a static IP for the Ubuntu machine running FreeRadius server. To do that you can use two methods.

First method

- Connect to the WebUI
- Navigate to **Status → Network → LAN**
- In the **DHCP Leases section** you should see Ubuntu machine's IP address
- Press  near the instance to create a static IP lease

Second method

- Connect to the WebUI
- Navigate to **Network → DHCP → Static Leases**
- Add the Ubuntu machine's MAC, IP, and provide a description



- Press 

Creating a new RUTOS user

Now we will need to create a new user for SSH and/or WebUI access. To do that follow these steps:

- Go to **System → Administration → User Settings → System Users** section
- In the Add new user section fill in the user's login credentials.

You can specify your own custom role or choose one from the default roles. In this example, the admin role was chosen.



Remember: use the **same username as in** FreeRadius **users** file. The password can be different, compared to the one in FreeRadius **users** file.

PAM package installation

Now we will need to install a PAM package, to do that follow these steps:

- Go to **System → Package Manager → Packages**
- **Install** the **PAM** package



Radius server configuration

Now we will set the FreeRadius server's information on the router


For SSH authentication

Firstly, we will need to enable SSH access for the created user. To do that, follow these steps:

- Go to **System → Administration → User Settings → System Users** section

- Press  near the newly created user
- **Enable** the **SSH access**
- Press 

To enable PAM authentication for SSH, follow these steps:


- Go to **System** → **Administration** → **Access Control** → **PAM** section
- Press  near the SSH instance
- **Enable** the **instance**
- Set **module** to **RADIUS**
- Set **type** to **Required**
- Set **server** to **Ubuntu machine's IP**
- Set **secret** to **the one defined in** the FreeRadius **clients.conf** file
- Leave **Port** and **Timeout** to their **default** values



- Press 

For WebUI authentication

To enable PAM authentication for WebUI, follow these steps:

- Go to **System** → **Administration** → **Access Control** → **PAM** section
- Press  near the WebUI instance
- **Enable** the **instance**
- Set **module** to **RADIUS**
- Set **type** to **Required**
- In the **Select users** add the newly created **user** or **enable** PAM authentication **for all users**
- Set **server** to **Ubuntu machine's IP**
- Set **secret** to **the one defined in** the FreeRadius **clients.conf** file
- Leave **Port** and **Timeout** to their **default** values



- Press 

Testing configuration

Now that we have the setup configured, we can test if the server properly authenticates the user. To see authentication requests on the FreeRadius server side, follow these steps:

- Stop the FreeRadius service using this command:

```
sudo /etc/init.d/freeradius stop
```

- Start the FreeRadius server in debug mode using this command:

```
sudo freeradius -X
```

- Try connecting to the router's WebUI and/or SSH service

If the authentication is successful the server logs will contain these lines:

```
Auth-Type PAP {
  pap: Login attempt with password
  pap: Comparing with "known-good" MD5-Password
  pap: User authenticated successfully
    [pap] = ok
  } # Auth-Type PAP = ok
```

If the authentication is unsuccessful the server logs will contain these lines:

```
Auth-Type PAP {
  pap: Login attempt with password
  pap: Comparing with "known-good" MD5-Password
  pap: ERROR: MD5 digest does not match "known good" digest
  pap: Passwords don't match
    [pap] = reject
  } # Auth-Type PAP = reject
```