

# Содержание

|   |   |
|---|---|
| <b>Description of RADIUS proxy based on FreeRADIUS and installation</b> ..... | 3 |
| <i><b>Purpose</b></i> .....   | 3 |
| <i><b>Load balancing modes</b></i> .....                                      | 3 |
| <i><b>Installation</b></i> .....  | 4 |
| Step 1. install FreeRADIUS .....  | 4 |
| Step 2. add the vasexperts dictionary .....                                   | 4 |
| <i><b>RADIUS proxy update</b></i> .....                                       | 5 |



# Description of RADIUS proxy based on FreeRADIUS and installation

## Purpose

In SSG BNG, the Control Plane is implemented via fastPCRF, which supports working with RADIUS only in active-standby (fail-over) mode with a group of RADIUS servers — operation with only one active RADIUS server from the group and switching to the next one if the current server becomes unavailable.

To implement load-sharing scenarios and distribution across a group of RADIUS servers, the RADIUS Proxy component is used — the FreeRADIUS package. At the same time, fastPCRF retains a single entry point — the FreeRADIUS server, which acts as a proxy server (it balances and distributes requests between the operator's RADIUS servers known to it). The proxy server remembers on which RADIUS server a subscriber was authorized and subsequently forwards accounting and re-authorization requests to the same server.

## Load balancing modes

1. **fail-over** — the request is sent to the first alive home server in the list. That is, if the first home server is marked as "dead", the second one is selected, and so on.
2. **load-balance** — the least loaded home server is selected, where "least load" is determined by counting the number of requests sent to the home server and subtracting the number of responses received from it.

If two or more servers have the same lowest load, one of them is selected at random. This configuration is most similar to the old round-robin, although it is not exactly the same.

Note that load balancing does not work very well with EAP, because EAP requires packets belonging to the same EAP exchange to be sent to the same home server. The load-balancing method does not keep state between packets, which means that EAP packets for the same conversation may be sent to different home servers. This will prevent EAP from working.

For authentication methods other than EAP, and for accounting packets, we recommend using load-balance. This helps ensure maximum network availability.

3. **client-balance** — the home server is selected by hashing the source IP address of the packet. If this home server is unavailable, the next one in the list is used, as in the fail-over case.

It is not possible to predict which source IP address will be mapped to which home server.

This configuration is most useful for simple load balancing of EAP sessions, since an EAP session will always be sent to the same home server.

4. **client-port-balance** — the home server is selected by hashing the source IP address and source port of the packet. If this home server is unavailable, the next one in the list is used, as in the fail-over case.

This method provides slightly better load balancing for EAP sessions than `client-balance`. However, it also means that authentication and accounting packets for the same session MAY be sent to different home servers.

5. **keyed-balance** — the home server is selected by hashing (FNV) the contents of the Load-Balance-Key attribute from the control items. The request is then sent to the user-selected home server:

```
server = (hash % num_servers_in_pool)
```

If the Load-Balance-Key attribute is not present in the control items, the load-balancing method is identical to `load-balance`.

For most non-EAP authentication methods, the User-Name attribute is a good key. The `unlang` policy can be used to copy User-Name into the Load-Balance-Key attribute. This method may not work for EAP sessions, since the outer identity outside the TLS tunnel is often static, for example, "anonymous@realm".

## Installation

### Step 1. install FreeRADIUS

Install the base FreeRADIUS packages:

```
sudo yum install freeradius freeradius-utils freeradius-mysql
```

### Step 2. add the vasexperts dictionary

Add the dictionary to the `/usr/share/freeradius/` directory and then include it in the dictionary file. If it is already present, it is recommended to overwrite it, since not all new attributes may be defined in the original version:

```
cp /usr/share/dpi/dictionary.vasexperts /usr/share/freeradius/
```



It is important to note that `freeradius` is the package name, while `radiusd` is the name of the service (daemon) that manages it. The directory containing the FreeRADIUS configuration prior to version 3 is called `/etc/raddb/`

## **RADIUS proxy update**

Update the version using the following command:

```
sudo yum upgrade freeradius freeradius-utils freeradius-mysql
```