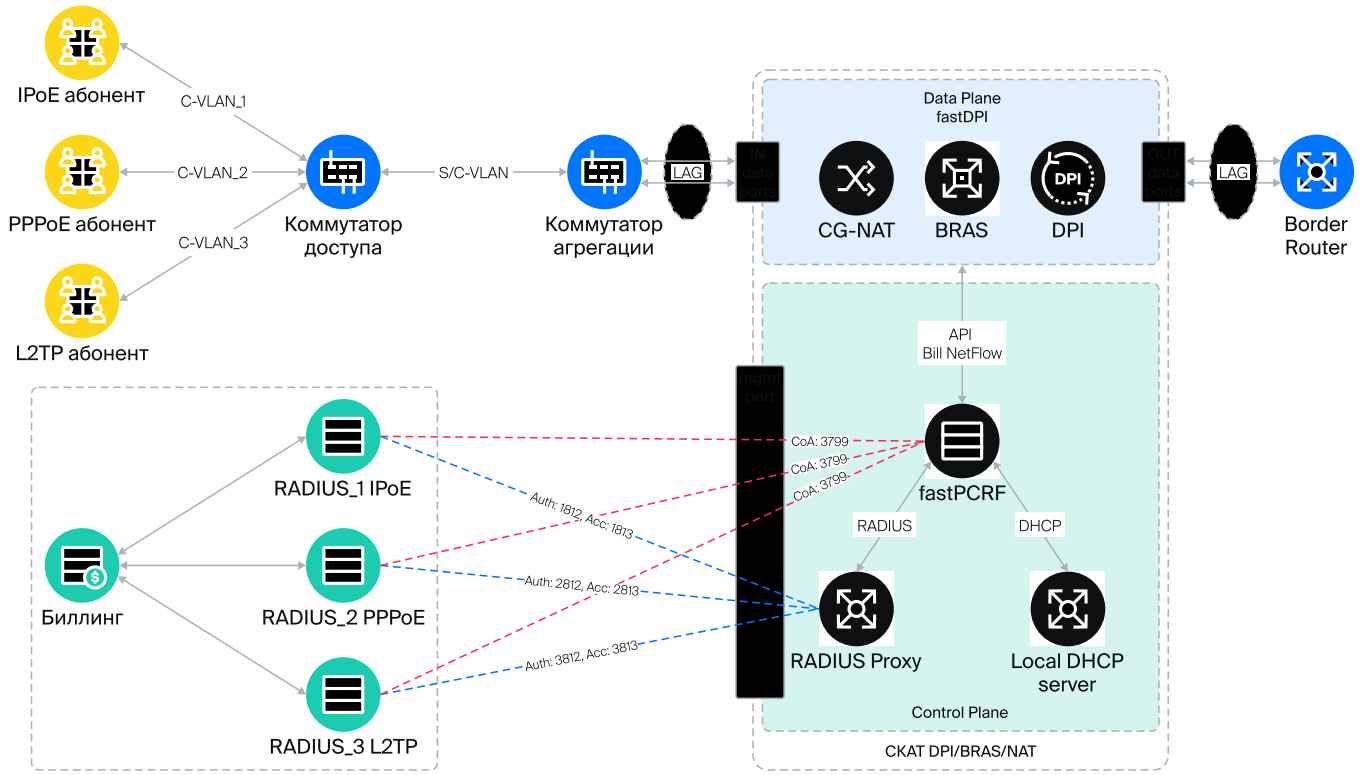
Содержание

Настройка балансировки и распределения по группам RADIUS серверов 3
Конфигурация FreeRADIUS как Proxy 3
Конфигурация виртуального сервера FreeRADIUS 6

Настройка балансировки и распределения по группам RADIUS серверов

Используется для сценария, когда необходимо распределить RADIUS трафик абонентов с разными типами доступа IPoE, PPPoE, L2TP между разными группами RADIUS серверов.



В данном примере рассмотрим распределение авторизации и аккаунтинга в режиме load-balance для разных типов доступа между группами RADIUS-серверов:

1. IPoE на RADIUS_1 (1812, 1813), RADIUS_1.1 (1822, 1823)
2. PPPoE на RADIUS_2 (2812, 2813), RADIUS_2.1 (2822, 2823)
3. L2TP на RADIUS_3 (3812, 3813), RADIUS_3.1 (3822, 3823)

Разделение Auth/Acct различных типов доступа происходит на основе атрибута `VasExperts-Service-Type` с использованием операторов ветвления (if-else).

RADIUS CoA сообщения могут отправляться от RADIUS серверов в BRAS (fastPCRF) следующим способом:

1. Напрямую от RADIUS (billing) в fastPCRF. Необходимо добавить отдельные CoA-клиенты в fastPCRF. RADIUS CoA. Указано на схеме красным пунктиром.
2. Через Proxy: RADIUS (billing) → RADIUS Proxy → fastPCRF. Описан пример ниже.

Конфигурация FreeRADIUS как Proxy

Конфигурация proxy расположена в `/etc/raddb/proxy.conf`. В ней определены основные

разделы:

```
proxy server{
  retry_delay = 5
  retry_count = 3
  default_fallback = no
  #dead_time = 120
  wake_all_if_all_dead = yes
}
```

- retry_delay — интервал ожидания (в секундах) после неудачной попытки установить соединение с сервером.
- retry_count — максимальное число попыток отправки запроса к серверу, после которого он считается "мёртвым".
- default_fallback — параметр, определяющий отправку reject-ответа клиенту, если все серверы находятся в состоянии "мёртвые".
- wake_all_if_all_dead — параметр, определяющий периодическую проверку доступности серверов, помеченных как "мёртвые".

Раздел, определяющий параметры "домашних" серверов (на которых хранятся данные об абонентах).

```
home_server rad_1 {
        type = auth+acct
        ipaddr = 10.166.220.232
        port = 1812
        secret = secret
        # proto = udp
# optional items
        src_ipaddr = 10.16.20.117
        response_window = 6
        zombie_period = 40
        status_check = status-server
        check_interval = 6
        check_timeout = 4
        num_answers_to_alive = 2
        max_outstanding = 65536
        coa {
                irt = 2
                mrt = 16
                mrc = 5
                mrd = 30
        }
        limit {
              max_connections = 16
              max_requests = 0
              lifetime = 0
              idle_timeout = 0
        }
}
```

- `type` — назначение сервера; чаще всего используется для авторизации (`auth`) либо для авторизации и аккаунтинга (`auth+acct`).
- `ipaddr` — IPv4-адрес RADIUS-сервера; при необходимости может использоваться `ipv6addr`.
- `port` — порт, на который проксируются запросы (обычно 1812). В режиме `auth+acct` задаётся только порт авторизации, а порт аккаунтинга определяется как `port+1`.
- `proto` — протокол передачи данных; по умолчанию udp.
- `secret` — общий секрет, используемый для подписи и защиты пакетов между RADIUS-сервером и прокси.
- `src_ipaddr` — IP-адрес источника, с которого прокси отправляет запросы.
- `response_window` — интервал ожидания ответа от сервера; по его истечении сервер помечается как "zombie" и получает минимальный приоритет при выборе.
- `zombie_period` — максимальный период ожидания ответа от сервера на любой пакет, по истечении которого сервер считается "мёртвым".
- `status_check` — способ проверки состояния сервера.
- `check_interval` — интервал между отправкой пакетов проверки состояния.
- `check_timeout` — время ожидания ответа на пакет проверки состояния.
- `num_answers_to_alive` — количество успешных проверок подряд, после которых сервер считается "живым".
- `max_outstanding` — максимальное число неподтверждённых пакетов (разница между отправленными и полученными), при превышении которого отправка новых пакетов приостанавливается для предотвращения перегрузки RADIUS-сервера.
- `coa` — секция, описывающая интервалы и количество повторных передач для Change of Authorization.
- `limit` — секция, применимая только при использовании TCP. Включает параметры:
  - `max_connections` — максимальное количество соединений;
  - `max_requests` — максимальное число запросов в рамках одного соединения;
  - `lifetime` — время жизни соединения в секундах;
  - `idle_timeout` — максимальный период бездействия в рамках соединения, после которого оно закрывается.

    Значение `0` для всех параметров означает отсутствие ограничений.

Для proxy определить необходимое количество серверов. Серверы, отвечающие за один и тот же тип авторизации, сгруппировать в `pool`. Раздел `home_server_pool` использовать для балансировки нагрузки и переключения между серверами.
В статье приведён пример минимальной конфигурации, полный вариант конфигурации доступен в архиве.

```
home_server_pool pool_rad_servers {
    type = load-balance
    home_server = rad_1
}
```

Важно, чтобы `home_server` были одного типа, то есть все — `auth` или `auth+acct`.

В разделе `realm` указать, какой пул серверов следует использовать для этой области.

```
realm rlm_prod_servers {
    pool = pool_rad_servers
    nostrip
```

```
}
```

При использовании пула, содержащего только серверы авторизации, применять `auth_pool`; при пуле, включающем только аккаунтинг, использовать `acct_pool`. Рекомендуется использовать универсальный `pool`, объединяющий оба варианта.
Проксирование CoA-пакетов выполнять на основе атрибута `Operator-Name` через `coa_pool`.
Для настройки CoA использовать файл `/etc/raddb/sites-available/coa`, в котором задаются параметры проксирования CoA-запросов. В конфигурации указать условия, при которых запрос направляется в соответствующий `realm`, а в каждом `home_server` определить параметры, используемые для CoA (как правило, они заданы по умолчанию).
Рекомендуется для работы с CoA использовать прямое взаимодействие PCRF – RADIUS, так как все параметры настраиваются в `fastpcrf.conf`, а обмен данными осуществляется напрямую.
Описание настройки данных функций приведено в статье статья.

Параметр `nostrip` использовать для отключения разделения значения `User-Name`.

Конфигурация виртуального сервера FreeRADIUS

Файл конфигурации расположен в `/etc/raddb/sites-available/default`.
Все параметры указывать для `default`. В первую очередь настроить раздел `listen`, в котором задать подсети и порты для прослушивания, а также типы принимаемых сообщений.

```
server default {
listen {
        type = auth
        ipaddr = *
        port = 0
        limit {
              max_connections = 16
              lifetime = 0
              idle_timeout = 30
        }
}
listen {
        ipaddr = *
        port = 0
        type = acct
        limit {
        }
}
```

Настроить раздел для IPv6:

```
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 {
        }
}
```

Далее перейти к разделу авторизации, в котором перечислены поддерживаемые протоколы авторизации.
В данном разделе задать правило маршрутизации запросов: все запросы авторизации с атрибутом `VasExperts-Service-Type` со значениями `0` и `1` направлять в область DHCP, со значениями 2, 3 и 4 — в область PPPoE, остальные запросы отклонять с Reject. Значения атрибута для других типов авторизации с расшифровкой приведены в словаре vasexperts и в статье.

```
authorize {
        preprocess
        chap
        mschap
        digest
        suffix
        files
        -sql
        -ldap
        expiration
        logintime
        if (Tunnel-Type) {
            update control {
                Proxy-To-Realm := "rlm_prod_servers_3"
           }
        }

        else {
              if (VasExperts-Service-Type == 0  || VasExperts-Service-Type
== 1) {
          update control {
                Proxy-To-Realm := "rlm_prod_servers_1"
          }
        }
  else ( VasExperts-Service-Type == 2 || VasExperts-Service-Type == 3 ||
VasExperts-Service-Type == 4 ) {
                update control {
                      Proxy-To-Realm := "rlm_prod_servers_2"
                }
              }
```

```
                else {
                    reject
                }
        }
}
```

Раздел определяет методы аутентификации, которые будет использовать FreeRADIUS. Так как они пустые — будут использоваться модули по умолчанию.

```
authenticate {
        Auth-Type PAP {
        }
        Auth-Type CHAP {
        }
        Auth-Type MS-CHAP {
        }
}

preacct {
        preprocess
        acct_unique
        suffix
        files
}
```

Раздел тарификации и аккаунтинга, в котором заданы правила обработки пакетов с различными значениями `VasExperts-Service-Type`, аналогично модулю авторизации. Существенным отличием является первое правило для `Acct-Status-Type`, на основе которого прокси направляет на RADIUS общие пакеты начала тарификации, а не отбрасывает их.

```
accounting {
        -sql
        if (Acct-Status-Type == Accounting-On || Acct-Status-Type ==
Accounting-Off) {
            update control {
                Proxy-To-Realm := "rlm_acct_servers"
            }
          }

        else {
           if (Tunnel-Type) {
            update control {
                Proxy-To-Realm := "rlm_prod_servers_3"
            }
        }
        else {
              if (VasExperts-Service-Type == 0 || VasExperts-Service-Type ==
1) {
                update control {
                   Proxy-To-Realm := "rlm_prod_servers_1"
                }
```

```
                }
                else   (VasExperts-Service-Type == 2 || VasExperts-Service-
Type == 3 || VasExperts-Service-Type == 4 ) {
                    update control {
                      Proxy-To-Realm := "rlm_prod_servers_2"
                }
                }
            }
      }
}
session {
}
```

Завершающие разделы. Задать параметр таймаута сессии и действие системы при Reject.

```
post-auth {
        update reply {
                Session-Timeout := 4294967295
        }
        -sql
        Post-Auth-Type REJECT {
        }
        Post-Auth-Type Challenge {
        }
}
pre-proxy {
}
post-proxy {
}
}
```