# Содержание

# Пример настройки FreeRadius 3

В этом разделе приведены минимальные изменения в конфигурацию FreeRadius 3.

 Данные изменения следует рассматривать лишь как один из примеров конфигурирования FreeRadius. Здесь не рассматривается интеграция FreeRadius с биллинговой системой или базой данных.

Допустим, IP-адрес Radius-сервера — 192.168.1.200, порт 1812.

#### Словарь VAS Experts

Сначала следует добавить словарь `vendor-specific` атрибутов `dictionary.vasexperts` в словарь Radius-сервера. Для этого:

- Скопировать словарь `/usr/share/dpi/dictionary.vasexperts` из дистрибутива fastPCRF в каталог `$freeRadius/share/freeradius`
- Добавить в главный словарь `$freeRadius/share/freeradius/dictionary` строку:

```
$INCLUDE dictionary.vasexperts
```

#### Создание клиента

В файле конфигурации fastpcrf.conf инстанса, который является Radius-клиентом, должно быть прописано соединение с Radius-сервером:

```
radius_server=secret123@192.168.1.200%eth0:1812;msg_auth_attr=1
```

Здесь `eth0` — это имя локального для клиента устройства (сетевой карты), с которой будет устанавливаться соединение с сервером 192.168.1.200.

**Настройки Radius-сервера и клиента должны совпадать!**

Для каждого инстанса fastPCRF первым делом следует создать клиента в FreeRadius. Допустим, название клиента fastdpi1. Все клиенты (инстансы fastPCRF) будут ссылаться на один и тот же виртуальный сервер fastdpi-vs.

Добавить в raddb/clients.conf Radius-сервера следующие строки:

```
client fastdpi1 {
    ipaddr        = 192.168.1.32
    secret        = secret123
    require_message_authenticator = yes
# add_cui = yes
    virtual_server    = fastdpi-vs
}
```

Здесь:

- `ipaddr` — задает IP-адрес инстанса fastPCRF, у нас это 192.168.1.32;
- `secret` — уникальный секрет, известный Radius-серверу и клиенту (то есть инстансу fastPCRF). Значение строки секрета нужно выбирать самостоятельно. Тот же самый секрет прописан в настройках fastpcrf.conf: `radius_server=secret123@192.168.1.200%eth0:1812`;
- `require_message_authenticator` — флаг, устанавливающий обязательность присутствия в Radius-запросе атрибута `Message-Authenticator`. RFC 2869 настоятельно рекомендует использовать данный атрибут. Эта настройка должна быть согласована с параметром `msg_auth_attr` в fastpcrf.conf: `radius_server=…;msg_auth_attr=1`;
- `add_cui` — **не выставляйте этот параметр в yes!** В целях безопасности Radius-сервер передает атрибут CUI (`Chargeable-User-Identity`) как зашифрованное хеш-значение логина пользователя, что неприемлемо для fastDPI, — нужен истинный логин пользователя. Поэтому `add_cui` здесь закомментировано.
- `virtual_server` — задает имя виртуального сервера (его конфигурирование см. далее).

**Создание виртуального сервера**

Для создания конфигурации виртуального сервера скопировать файл `raddb/sites-available/default`, входящий в поставку FreeRadius, в `raddb/sites-enabled/fastdpi-vs` и затем отредактировать`fastdpi-vs`:

- задать имя виртуального сервера — изменить в начале файла строку `server default` на `server fastdpi-vs`
- в секции listen для auth-запросов (type = auth) прописать, на каком IP-адресе и каком порту слушать входящие запросы (заметим, что это локальный адрес Radius-сервера):

```
ipaddr = 192.168.1.200
port = 1812
interface = eth0
```

- остальные секции listen удалить либо закомментировать — они не нужны
- всю основную работу по составлению ответа на Access-Request прописать в секции post-auth. Здесь дать какие-то рекомендации невозможно — все зависит от конкретного провайдера, от окружения Radius-сервера — откуда брать данные. Список необходимых атрибутов см. “RADIUS ACCESS-ACCEPT”. В качестве примера приводится статическое заполнение атрибутов ответа Access-Accept (Внимание, наличие в запросе Access-Request атрибута CUI (`Chargeable-User-Identity`), содержащего единственный нулевой байт, означает, что fastPCRF не знает логин пользователя и запрашивает его у Radius-сервера; в данном примере CUI формируется из `Framed-IP-Address` только в качестве иллюстрации):

```
post-auth {
…
    #
    # Add VasExperts attributes
    #
    if ( Chargeable-User-Identity == 0x00 ) {
      update reply {
        Chargeable-User-Identity := "u-%{Framed-IP-Address}"
```

```
      }
    }
    else {
       update reply {
         Chargeable-User-Identity := "%{Chargeable-User-Identity}"
        }
    }
    update reply {
        Framed-IP-Address := "%{Framed-IP-Address}"
        VasExperts-Policing-Profile := "test1"
        VasExperts-Service-Profile  += "1:test1"
        Session-Timeout := 300
    }
…
}
```

- **Параметр CUI секции post-auth оставить закомментированным!** FreeRadius вместо логина пользователя посылает в CUI хеш-значение логина, что нам не нужно, поэтому атрибут CUI в ответе нужно сформировать самостоятельно, см. пример выше.
- Ниже в секцию Post-Auth-Type REJECT (формирование Access-Reject) добавить:
  - Формирование атрибута CUI, если fastPCRF его запрашивает и пользователь известен;
  - Атрибут `VasExperts-Policing-Profile`, задающий профиль полисинга для неавторизованных пользователей (в примере ниже имя профиля — plc_unauth, у вас имя будет другое);
  - Атрибут `VasExperts-Service-Profile`, задающий профиль услуги 5 ("Белый список"). Обычно это профиль, разрешающий неавторизованным пользователям доступ только к Captive Portal. В примере ниже имя профиля — cp_unauth, у вас имя будет другое.

**Пример:**

```
if (Chargeable-User-Identity == "\0" ) {
   update reply {
    Chargeable-User-Identity := "login"
   }
}
update reply {
    VasExperts-Policing-Profile := "plc_unauth"
    VasExperts-Service-Profile  += "5:cp_unauth"
}
```

#### Редактирование users

В файл `raddb/users` следует добавить две записи для fastPCRF:

```
VasExperts.FastDPI.unknownUser Cleartext-Password := "VasExperts.FastDPI"
DEFAULT Cleartext-Password := "VasExperts.FastDPI"
```

Первая запись задает имя пользователя, которое шлет fastPCRF если логин ему не известен,

подробнее см. описание conf-параметра `radius_unknown_user`. Это имя настраивается в fastPCRF, так же как и пароль, см. conf-параметр `radius_unknown_user_psw`. Вторая запись задает пароль, с которым fastPCRF шлет запросы для известных логинов. Этот пароль настраивается в fastPCRF, см. conf-параметр `radius_user_password`.

# Трюки FreeRadius

## Балансировка нагрузки на DNS

```
# Два  DNS-сервера: 8.8.8.8 и 8.8.8.9.
# Хотим балансировать нагрузку на них.
# Пример: в секции  "post-auth",  добавить:

    if ( "%{rand:2}" == "0" ) {
        update reply {
            # удаляем все атрибуты  DNS
            VasExperts-DHCP-DNS !* ANY

            VasExperts-DHCP-DNS = "8.8.8.9"
            VasExperts-DHCP-DNS += "8.8.8.8"
        }
    }
    else {
        update reply {
            # удаляем все атрибуты  DNS
            VasExperts-DHCP-DNS !* ANY

            VasExperts-DHCP-DNS = "8.8.8.8"
            VasExperts-DHCP-DNS += "8.8.8.9"
        }
    }
```