Оглавление

Версия	Автор	Описание
0.1 [2016.10.19]	Хижинский М.	Первая редакция
0.2 [2016.10.21]	Хижинский М.	Добавлено описание параметра `coa_client`
0.3 [2016.10.25]	Хижинский М.	Добавлены описания Radius-пакетов `Access-Accept`, `Access-Reject`, `CoA_Request`, `Disconnect-Request`
0.4 [2016.10.28]	Хижинский М.	В `Access-Accept` и `CoA-Request` добавлен атрибут `Session-Timeout`.
0.5 [2016.10.31]	Хижинский М.	Добавлено описание команд ручной установки статуса авторизации утилиты `fdpi_ctrl`
0.6 [2016.11.02]	Хижинский М.	Добавлена возможность записи pcap-файлов (опции pcap_dump_path, pcap_dump_fsize, radius_dump_pcap)

0.7 [2016.11.08]	Хижинский М.	Добавлены conf-параметры: `default_reject_policing`, `default_reject_whitelist` Уточнены требования к Access-Reject, Disconnect-Request
0.8 [2016.11.11]	Хижинский М.	Добавлена глава “Конфигурирование FreeRadius”
0.9 [2016.11.24]	Хижинский М.	Удалены опции radius_emul и default_policing
0.10 [2017.01.25]	Хижинский М.	Добавлены рекомендации по созданию aslocal.bin и asnum.dscp

Назначение fdpi_pcrf

Сервер fdpi_pcrf фактически выполняет роль сервера авторизации (L3 BRAS) для fastdpi: для исходящего клиентского трафика fdpi_pcrf запрашивает у Radius-сервера авторизацию клиента, его профиль полисинга (аналог тарифного плана) и профили по услугам.

Сервер fdpi_pcrf тесно интегрирован с fastdpi и состоит из трех частей:

- Модуль авторизации в fastdpi, анализирующий исходящий трафик локальных клиентов; если клиент не авторизован, то модуль шлет TCP-запрос на сервер fdpi_pcrf;
- Сервер fdpi_pcrf, принимающий запросы на авторизацию от fastdpi по внутреннему протоколу и соответствующим образом их обрабатывающий;
- Управляющий модуль fastdpi, принимающий результаты работы fdpi_pcrf по протоколу fdpi_ctrl и запоминающий их в UDR (базе данных клиентов).

Настройки fastdpi

Прежде всего надо дать понять fastdpi, какие IP-адреса являются локальными. Только для локальных адресов будет проводится авторизация через Radius посредством fdpi_pcrf. Для этого:

1. Создаем `aslocal.bin` – см. http://vasexperts.ru/wiki/doku.php?id=statistics_asn. В файл `aslocal` заносим те диапазоны серых IP-адресов, которые используются в локальной сети провайдера. В качестве номера автономной системы для них указываем любой из диапазона 64512 – 65534.
2. Создаем файл asnum.dscp (или корректируем этот файл, если он уже есть) – см. http://vasexperts.ru/wiki/doku.php?id=priority_config_as. В этом файле нужно указать номера локальных автономных систем – именно для них будет производится авторизация. Как правило, это автономные системы для серых IP-адресов, указанные в `aslocal.bin`, плюс белые IP, выделенные провайдеру, если эти белые IP-адреса используются в локальной сети. Для всех IP-адресов автономных систем, помеченных как `local` в `asnum.dscp`, будет производится авторизация.

Конфигурационные параметры файла `fastdpi.conf`, относящиеся к fdpi_pcrf:

- `enable_auth` - булевый параметр, включает авторизацию через fdpi_pcrf. По умолчанию авторизация отключена.
- `auth_servers` – список серверов fdpi_pcrf. Формат задания в файле конфигурации: `ip%dev:port {,ip%dev:port }*`. Здесь `ip` – IP-адрес сервера fdpi_pcrf, `dev` – имя интерфейса, на котором создавать соединение, `port` - порт. Может быть задано до 16 адресов. Если авторизация включена (`enable_auth=yes`), должен быть задан адрес хотя бы одного сервера.
- `auth_trace` — булевый параметр, включает трассировку авторизации. Применять только при отладке, так как существенно нагружает fastdpi, как и всякая трассировка.
- `auth_resend_timeout` — тайм-аут перепосылки запроса на авторизацию серверу fdpi_pcrf, в секундах. Значение по умолчанию — 60 секунд.
- `auth_expired_timeout` — время действия авторизации, в минутах. Значение 0 — авторизация бессрочна. Значение по умолчанию — 0 (бессрочно). Если значение больше 0, то fastdpi по истечении тайм-аута `auth_expired_timeout` будет перезапрашивать авторизацию у fdpi_pcrf, тем самым предотвращая гипотетически возможные ситуации «вечной аренды» в случае сбоя одного из компонент (radius-сервера, CoA-клиента и пр.)

Для активации fdpi_pcrf в конфигурационном файле fastdpi как минимум должны быть заданы следующие настройки:

- `enable_auth = yes`
- `auth_servers = список серверов fdpi_pcrf`

Все сервера fdpi_pcrf считаются равноправными. Соединение устанавливается с первым доступным из списка `auth_servers`.

Установка статуса авторизации из командной строки

Утилита `fdpi_ctrl` поддерживает установку статуса авторизации пользователя. Пользователь должен быть задан либо IP-адресом (аргумент `--ip`), либо логином (аргумент `--login`).

Установка статуса авторизации:

```
fdpi_ctrl load --auth=1 [--timeout=nn] [--ip=xx.xx.xx.xx | --login=user_login]
```

Параметр `--auth` может принимать значения:

- `0` – пользователь не авторизован
- `1` – пользователь авторизован

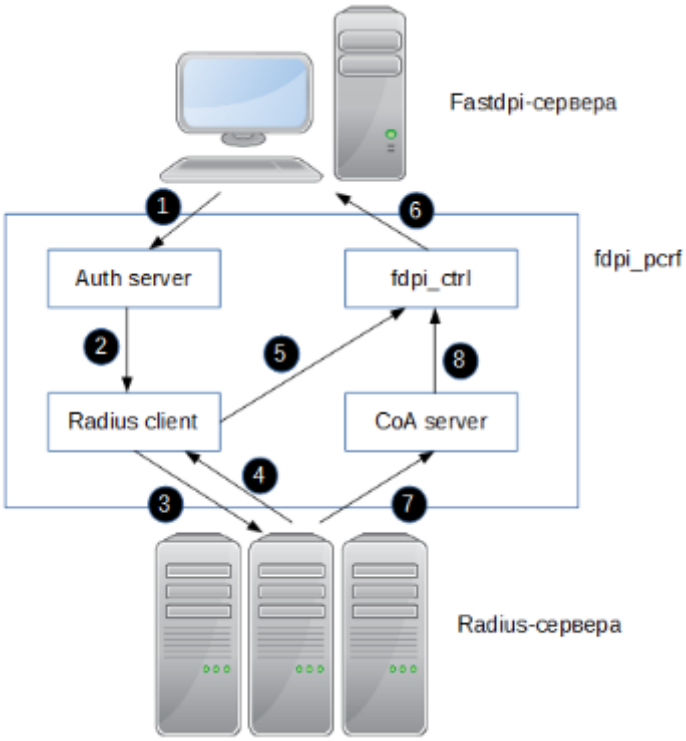
Опциональный аргумент `--timeout` задает время в секундах, в течение которого действует указанный статус авторизации. По окончании этого времени статус переводится в значение “неизвестный”, что запускает обычную авторизацию через Radius `Access-Request` при появлении пакета от пользователя.

Команда

```
fdpi_ctrl del --auth [--ip=xx.xx.xx.xx | --login=user_login]
```

позволяет вручную сбросить статус авторизации в значение “неизвестный”, что приведет к генерации Radius-запроса `Access-Request` при появлении пакета от пользователя.

Внутреннее устройство fdpi_pcrf



Fdpi_pcrf принимает запросы на авторизацию от Fastdpi-серверов (1) и передает их внутреннему Radius-клиенту (2). Radius-клиент формирует Access-Request PDU и передает его активному Radius-серверу (3). Получив ответ (4), Radius-клиент разбирает его и направляет (5) внутреннему клиенту fdpi_ctl, который, в свою очередь, запоминает его во внутреннем файловом FIFO-буфере и рассылает (6) всем fastdpi-серверам по протоколу fdpi_ctrl.

Также fdpi_pcrf реализует спецификацию RFC 5176 – Change of Authorization (CoA): получая (7) от Radius-серверов нотификации об изменении статуса авторизации (Disconnect-Request и CoA-Request), fdpi_pcrf через внутренний fdpi_ctl (8) рассылает всем fastdpi-серверам измененный статус клиента (6).

Персистентный FIFO-буфер внутри компонента fdpi_ctl гарантирует доставку параметров авторизации всем fastdpi-серверам даже если они находятся в offline: при старте fastdpi-сервера fdpi_pcrf устанавливает с ним соединение и посылает все находящиеся в буфере данные.

Внутреннее устройство fdpi_pcrf выполнено по модели апартаментов (apartments model). Физически апартамент находится в потоке (thread) и взаимодействует с другими апартаментами только асинхронно, через события. К потоку (thread) может быть привязано один или более апартаментов, а в каждый апартамент заселяется один или более компонентов; тем самым достигается большая гибкость решения: мы можем привязать все апартаменты к одному потоку или же для каждого апартамента создать свой поток.

Каждый внутренний компонент fdpi_pcrf (auth server, radius client, CoA server, fdpi_ctl) работает в своем апартаменте, то есть связь между компонентами – строго асинхронная, через сообщения. Внутри каждый компонент по сути является однопоточным (single-threaded) без какой-либо синхронизации, что существенно упрощает внутреннее строение

компонента и позволяет сосредоточиться на реализации внутренней логики компонента, а не на связи компонента с “внешним миром”.

Настройка fdpi_pcrf

Опции командной строки:

- `-c <conf_file_name>` - задает имя файла конфигурации.
- `-d` – запускать fdpi_pcrf как демон
- `-h` – вызов стравки (help)
- `-v` – печать версии

Конфигурационный файл

Настроечные параметры берутся из файла `fdpi_pcrf.conf`, который по умолчанию находится там же, что и `fastdpi.conf` – в каталоге `/etc/dpi`.

Общие параметры конфигурационного файла:

- `daemon` - булевый параметр, задающий режим запуска: если `daemon=1` – запускать fdpi_pcrf в режиме демона, иначе – как обычную программу. Значение по умолчанию – `0` (запускать как обычную программу). Данный параметр может быть переопределен опцией командной строки `-d`.
- `verbose` – булевый параметр, задает подробный уровень логирования: если `verbose=1` – программа будет подробно логировать свои действия, входящие запросы и исходящие данные; степень подробности задается параметром `trace` (см. далее). Значение по умолчанию – `0`.
- `trace` – битовая маска трассировки; задает, какие компоненты требуют подробного логирования. Значение по умолчанию – `0`.
- `rlimit_fsize` – максимальный размер файла при записи, байт. Значение по умолчанию – 1G (1073741824 байт).
- `print_stat_period` – период вывода внутренней статистики в файл `fdpi_pcrf_stat.log`. Задается в секундах, значение по умолчанию – `300` (статистика выводится раз в 5 минут).
- `work_thread_count` – количество рабочЗапись .pcap-файлових потоков, значение по умолчанию – `4`. Задавать значение больше 4 не имеет смысла, так как количество рабочих потоков не может быть больше, чем количество внутренних компонент fdpi_pcrf.
- `async_queue_size` – размер внутренней очереди передачи асинхронных сообщений; значение по умолчанию – `524288` (512K). Этот параметр лучше не трогать и тем более **не уменьшать**, так как при задании слишком маленького размера возможна потеря внутренних вызовов между компонентами, что равносильно потере работоспособности и/или утечке памяти.
- `pcap_dump_path` – задает путь, куда пишутся pcap-файлы, значение по умолчанию - `/var/log/dpi` (см. “Запись .pcap-файлов”)
- `pcap_dump_fsize` – максимальный размер pcap-файла в байтах, по умолчанию – 1Гигабайт. По достижении этого размера pcap-файл закрывается и создается новый (см. “Запись .pcap-файлов”).

Настройки компонента auth server:

- `auth_server_dev` - имя интерфейса (например, `eth0`), на котором слушаем входящие соединения. Значения по умолчанию нет. Если не задано – слушаем на всех интерфейсах.
- `auth_server_port` – номер порта, на котором слушаем входящие соединения. Значение по умолчанию – `29002`.
- `auth_server_max_connection` – максимальное число входящих соединений, значение по умолчанию – 16, максимальное значение – 16. Фактически, это число fastdpi-серверов, которые обслуживаются данным fdpi_pcrf сервером.

Настройки компонента fdpi_ctl

- `fdpi_server_list` – список fastdpi-серверов. Формат задания: `ip_address%dev:port{,ip_address%dev:port}*`. Необязательный суффикс `%dev` задает имя интерфейса, с которого следует связаться с указанным ip-адресом. Пример задания: `fdpi_server_list = 92.168.10.12%eth1:29000,10.10.25.99%eth0:29000`. В этом примере будет установлено соединение с 192.168.10.12, порт 29000 на интерфейсе eth1, с 10.10.25.99, порт 29000 – на интерфейсе eth0. Максимальное количество серверов в списке – 16.
- `fifo_fsize` – размер одного FIFO-файла, в байтах. Значение по умолчанию – 1048576 (1M). Этот параметр не должен быть больше `rlimit_fsize`. В случае, если текущий FIFO-файл достиг данного размера, будет создан следующий FIFO-файл. Количество создаваемых файлов в принципе не ограничено. Для каждого fastdpi-сервера из списка `fdpi_server_list` создается своя FIFO-очередь (то есть своя последовательность файлов), поэтому наличие в списке `fdpi_server_list` “мертвых душ” (неактуальных записей) приведет к неконтролируемому росту FIFO-очереди к этому серверу и как следствие – к переполнению диска. FIFO-файл автоматически удаляется, когда все данные из него будут отправлены fastdpi-серверу. При рестарте fdpi_pcrf FIFO-очередь от предыдущего запуска не используется – все данные теряются и FIFO-очередь создается заново.
- `fifo_dir` – каталог, в котором располагаются файлы FIFO-очереди. Значение по умолчанию – `/tmp`.
- `fifo_file_prefix` – префикс имени файла FIFO-очереди, значение по умолчанию – `fdpi_pcrf_fifo_`.
- `fifo_leave_bad_file` – булевый параметр, значение 1 запрещает fdpi_pcrf удалять FIFO-файл, на котором произошла какая-либо ошибка. Полезно для “разбора полетов” – что с файлом не так. Значение по умолчанию – `0`.

Настройки компонент radius client и CoA server

- `default_reject_policing` – имя профиля полисинга по умолчанию для неавторизованных пользователей. Подробнее см. “Radius Access-Reject”
- `default_reject_whitelist` – имя профиля услуги 5 (Белый список) по умолчанию для неавторизованных пользователей. Подробнее см. “Radius Access-Reject”
- `radius_revive_period` – периодичность (в секундах) задачи оживления подключения к главному Radius-серверу. Значение по умолчанию – 120 секунд. Radius-сервера в списке `radius_server` неравнозначны: первый считается главным radius-сервером, остальные – резервными. Если fdpi_pcrf обнаруживает, что главный radius-сервер слишком долго не отвечает, соединение с ним сбрасывается и fdpi_pcrf подключается к следующему radius-серверу из списка.

При этом производятся периодические попытки подключения к главному radius-серверу до тех пор, пока главный radius-сервер не станет доступным.

- `radius_max_pending_requests` – максимальное число ожидающих выполнения запросов от fastdpi-серверов. Значение по умолчанию – `1 000 000`. При превышении этого порога входящие запросы от fastdpi-серверов молча отбрасываются.
- `coa_max_pending_requests` – максимальное число ожидающих выполнения CoA-запросов от radius-серверов. Значение по умолчанию – `100 000`. Это значение не должно быть выше, чем значение параметра `async_queue_size`, рекомендуемое значение – не больше `async_queue_size / 2`.
- `radius_dump_pcap` – задает, какие Radius-пакеты будут записаны в pcap-файл. Значения опции:
  - 0 – ничего не писать в pcap (это значение по умолчанию)
  - 1 – сохранять в pcap-файле только ошибочные пакеты (обычно это ошибочные ответы от Radius-серверов)
  - 2 – сохранять в pcap-файле все Radius-запросы и ответы

  Параметр `radius_dump_pcap` можно менять “на лету”. Pcap-файлы создаются в каталоге, заданном параметром `pcap_dump_path`. Имя pcap-файла строится по маске `radius_mmddhhmmss_xxx.pcap`, где `mmddhhmmss` – текущая дата (месяц и день) и время. Подробнее см. “Запись .pcap-файлов”
- `coa_dump_pcap` - задает, какие CoA-пакеты будут записаны в pcap-файл. Значения опции:
  - 0 – ничего не писать в pcap (это значение по умолчанию)
  - 1 – сохранять в pcap-файле только ошибочные пакеты (обычно это ошибочные CoA-нотификации)
  - 2 – сохранять в pcap-файле все CoA-нотификации и ответы

  Параметр `coa_dump_pcap` можно менять “на лету”. Pcap-файлы создаются в каталоге, заданном параметром `pcap_dump_path`. Имя pcap-файла строится по маске `coa_mmddhhmmss_xxx.pcap`, где `mmddhhmmss` – текущая дата (месяц и день) и время. Подробнее см. “Запись .pcap-файлов”

- `radius_server` – задает адрес Radius-сервера и его конфигурационные параметры. Каждый radius-сервер в конфигурационном файле описывается отдельным параметром `radius_server`. Обычно задается как минимум 2 radius-сервера – основной и резервный, соответственно, в conf-файле должно быть как минимум 2 строки с параметром `radius_server` – для основного и резервного серверов. Максимальное число radius-серверов – 16. Radius-сервера не равнозначны: главным считается тот radius-сервер, который описан в conf-файле первым, остальные считаются резервными. Резервные сервера используются при недоступности главного и именно в том порядке, как задано в conf-файле. В каждый момент времени активным является только один radius-сервер.
  Формат задания `radius_server` следующий:
  `radius_server=secret@ip%dev:port{;param=value}*`
  - `secret` – секрет radius-сервера;
  - `ip` – ip-адрес radius-сервера
  - `dev` (необязательный) – имя интерфейса, на котором создавать соединение; если не задан – интерфейс выбирается операционной системой;
  - `port` – порт
  - `param=value` – перечень (через точку с запятой) конфигурационных параметров для данного radius-сервера.

  Конфигурационные параметры radius-сервера могут быть заданы тремя способами:
  - значения, одинаковые для всех radius-серверов, задаются как обычные параметры в conf-файле (все такие параметры перечислены ниже).

Основное условие – они должны быть заданы перед параметрами `radius_server`, - только в этом случае они применяются ко всем radius-серверам.

- Для каждого radius-сервера может быть создан свой конфигурационный файл, имя которого задается параметром `conf` в строке `radius_server`, например:
  `radius_server=secret@10.10.3.5:1812;conf=radius-main.conf`
  значения из `radius-main.conf` перекрывают default-значения параметров.
- Параметры, уникальные для конкретного radius-сервера, могут быть заданы прямо в строке `radius_server`, например:
  `radius_server=secret@10.10.3.5:1812;conf=radius-main.conf;msg_auth_attr=1`
  Здесь параметр `msg_auth_attr` задан для конкретного сервера `10.10.3.5` и перекрывает задание соответствующего параметра в файле конфигурации `radius-main.conf`. Следует учитывать, что порядок перечисления в `radius_server` важен: параметры применяются именно в том порядке, как они указаны в `radius_server`. Если в примере выше поменять местами `conf` и `msg_auth_param` и в conf-файле `radius-main.conf` задано `msg_auth_param=0`, то будет применен `msg_auth_param=0` из `radius-main.conf`.

Далее перечислены параметры radius-серверов, которые могут быть заданы индивидуально для каждого radius-сервера. Приводятся имена параметров в основном conf-файле `fdpi_pcrf.conf`, в скобках – так, как они задаются в параметре `radius_server` и в отдельном conf-файле radius-сервера:

- `radius_dead_timeout` (`dead_timeout`) – тайм-аут “безмолвствия” radius-сервера, в секундах, значение по умолчанию – 60. Если в течение этого периода времени от radius-сервера не пришло ни одного ответа, а запросы есть, то сервер считается умершим и fdpi_pcrf переключается на другой radius-сервер из списка (не забываем, что транспортным протоколом для RADIUS является UDP, в котором обрыв “соединения” определить точно невозможно). При этом если умер главный radius-сервер (тот, который задан первым в `fdpi_pcrf.conf`), то запускается задача оживления подключения к главному radius-серверу (см. `radius_revive_period` выше).
- `radius_max_connect_count` (`max_connect_count`) – максимальное число коннектов к одному radius-серверу, значение по умолчанию – 16. Согласно основной спецификации RADIUS (RFC 2865), под идентификатор, позволяющий сопоставить запрос с ответом, отводится поле размером 1 байт, то есть одно соединение может одновременно обслуживать не более 256 запросов. Для преодоления этого ограничения спецификация предлагает создавать несколько подключений к одному radius-серверу. Фактически этот параметр задает число одновременных запросов к одному radius-серверу: `radius_max_connect_count * 256`.
- `radius_response_timeout` (`response_timeout`) – тайм-аут ожидания ответа на запрос `Access-Request` к radius-серверу, секунд, значение по умолчанию – 30. Если в течение этого времени ответ на запрос не пришел, запрос считается отброшенным radius-сервером (например, по причине “слишком много запросов”) и fdpi_pcrf пытается послать запрос заново.
- `radius_resend_count` (`resend_count`) – максимальное количество попыток повторной отправки запроса, значение по умолчанию – 0 (без повторной отправки). Если число попыток повторной отправки запросов исчерпано и ответ от radius-сервера не получен, fdpi_pcrf ничего не сообщает fastdpi-серверу. Fastdpi в случае отсутствия ответа на авторизацию в течение определенного тайм-аута (параметр

auth_resend_timeout файла fastdpi.conf) пошлет повторный запрос на авторизацию.

- radius_status_server (status_server) – булевый параметр, задает, поддерживает ли radius-сервер запрос Status-Server (RFC 5997); значение по умолчанию – 1 (запрос поддерживается). Данный тип запроса используется fdpi_pcrf для пинга radius-сервера, особенно в случае временной недоступности основного radius-сервера. Без поддержки Status-Server понять, что основной radius-сервер восстановился, весьма затруднительно.
- radius_user_name_ip (user_name_ip) – булевый параметр, задающий, что должен содержать атрибут User-Name radius PDU Access-Request:
  - 1 (по умолчанию) – атрибут User-Name содержит IP-адрес пользователя (как в Cisco ISG)
  - 0 – атрибут User-Name содержит логин пользователя.

  Следует отметить, что fastdpi не всегда знает логин пользователя, тогда как его IP-адрес известен всегда. Если radius_user_name_ip=0 и логин неизвестен, то используется “логин” по умолчанию, задаваемый параметром radius_unknown_user. Также IP-адрес пользователя всегда передается в атрибуте Framed-IP-Address.
- radius_user_password (user_password) – строка, значение атрибута User-Password запроса Access-Request. Значение по умолчанию: ‘VasExperts.FastDPI’.
- radius_unknown_user (unknown_user) – строка, логин пользователя, если настоящий логин неизвестен fastdpi. Значение по умолчанию: ‘VasExperts.FastDPI.unknownUser’. Это значение атрибута User-Name запроса Access-Request, если radius_user_name_ip=0 и логин пользователя неизвестен. Предполагается, что radius-сервер в ответе Access-Accept сообщит истинный логин пользователя, определенный по его IP-адресу, взятому из атрибута Framed-IP-Address.
- radius_unknown_user_psw (unknown_user_pws) – строка, значение атрибута User-Password для неизвестного логина пользователя. Применяется только если radius_user_name_ip=0. Значение по умолчанию: ‘VasExperts.FastDPI’.
- radius_msg_auth_attr (msg_auth_attr) – булевый параметр, задает, поддерживает ли radius-сервер атрибут Message-Authenticator (RFC 2869). Значение по умолчанию – 1 (атрибут поддерживается). Если атрибут поддерживается, fdpi_pcrf будет вычислять и включать Message-Authenticator в каждый запрос Access-Request и Status-Server, а также анализировать этот атрибут в ответах; если в ответе проверка атрибута Message-Authenticator заканчивается ошибкой, то такой ответ отбрасывается.
- radius_attr_nas_port_type (attr_nas_port_type) – число, значение атрибута NAS-Port-Type (RFC 2865) запроса Access-Request; значение по умолчанию – 5 (Virtual).
- radius_attr_nas_port (attr_nas_port) – число, значение атрибута NAS-Port (RFC 2865) запроса Access-Request; значение по умолчанию – 0.
- radius_attr_nas_ip_address (attr_nas_ip_address) – IPv4-адрес, значение атрибута NAS-IP-Address (RFC 2865) запроса Access-Request. Задается в конфигурационном файле как строка, значение по умолчанию отсутствует. Если не задан – атрибут NAS-IP-Address не включается в запрос.
- radius_attr_nas_id (attr_nas_id) – строка, значение атрибута NAS-Identifier запроса Access-Request. Значение по умолчанию отсутствует. Согласно RFC 2865, либо NAS-IP-Address, либо NAS-Identifier должен быть задан в Access-Request.
- radius_attr_service_type (attr_service_type) – число, значение атрибута Service-Type (RFC 2865) запроса Access-Request. Значение по умолчанию – 2 (Framed).

- radius_attr_cui (attr_cui) – булевый параметр, задает, поддерживает ли radius-сервер атрибут Chargeable-User-Identity (CUI, RFC 4372). Значение по умолчанию – 1 (CUI поддерживается). Если этот атрибут поддерживается, то fdpi_pcrf в запросе Access-Request помещает в этот атрибут логин пользователя; если логин неизвестен, то в атрибут помещается нулевой байт, что означает, согласно RFC 4372, запрос логина у radius-сервера. В ответе Access-Accept fdpi_pcrf ожидает прихода в этом атрибуте истинного логина пользователя, который radius-сервер может определить по его IP-адресу (атрибут Framed-IP-Address запроса).
- radius_coa_port (coa_port) – UDP-порт, на который поступают Change-of-Authorization (CoA) оповещения Disconnect-Request, CoA-Request (RFC 5176).Значение по умолчанию: 3799 (определен в RFC 5176); если radius-сервер не поддерживает CoA, следует задать этому параметру значение 0.
- radius_coa_resend_timeout (coa_resend_timeout) – тайм-аут перепосылки CoA-ответов (Disconnect-ACK, Disconnect-NAK, CoA-ACK, CoA-NAK) в случае проблем с сокетом (обычно переполнение очереди сокета), секунд. Значение по умолчанию – 1 секунда. Количество повторных попыток задается параметром radius_resend_count.

CoA-клиенты

В некоторых конфигурациях CoA-клиент, посылающий CoA-запросы Disconnect-Request и CoA-Request, может быть отдельной сущностью, не являющейся radius-сервером. Например, это может быть некая утилита, умеющая формировать CoA-запросы и применяющаяся в скриптах. Fdpi_pcrf поддерживает такие “обособленные” CoA-клиенты. В конфигурационном файле fdpi_pcrf.conf каждый такой CoA-клиент задается отдельным параметром coa_client, имеющим формат, аналогичный параметру radius_server:

coa_client=secret@ip%dev:port{;param=value}*

- secret – секрет Radius;
- ip – ip-адрес CoA-клиента;
- dev (необязательный) – имя интерфейса, на котором слушать входящие запросы; если не задан – интерфейс выбирается операционной системой;
- port – слушаемый локальный порт;
- param=value – перечень (через точку с запятой) конфигурационных параметров для данного CoA-клиента. Поддерживаются параметры: max_resend_count, msg_auth_attr, coa_resend_timeout, см. описания выше для radius-серверов.

Каждый CoA-клиент описывается в conf-файле отдельным параметром coa_client. Всего может быть до 16 обособленных CoA-клиентов.
Fdpi_pcrf принимает CoA-запросы только от зарегистрированных (описанных в conf-файле) radius-серверов и CoA-клиентов. Если radius-сервер поддерживает CoA, нет необходимости описывать его ещё и параметром coa_client, - достаточно для этого radius-сервера указать опцию coa_port в параметре radius_server.

Минимальная конфигурация

Для обеспечения полноценной работы fdpi_pcrf необходимо задать в конфигурационном файле fdpi_pcrf.conf следующие параметры:

- fdpi_server_list – список серверов fastdpi
- radius_server – задает один radius-сервер. Каждый radius-сервер описывается собственным параметром radius_server. Как правило, должно быть задано как

минимум два radius-сервера – один основной и один резервный, поэтому в conf-файле должно присутствовать как минимум две записи `radius_server`. Порядок записей важен – основным radius-сервером считается сервер, описанный первым параметром.

В конфигурационных файлах всех fastdpi-серверов необходимо задать параметр `auth_servers` – список fdpi_pcrf серверов для авторизации.

Формат Access-Request RADIUS-запроса

Протокол RADIUS весьма обширен и неоднозначен. fdpi_pcrf шлет `Access-Request`-запросы со следующими атрибутами:

```
User-Name = "94.158.56.38"
User-Password = "VasExperts.FastDPI"
Framed-IP-Address = 94.158.56.38
NAS-Port-Type = Virtual
NAS-Port = 0
NAS-Port-Id = "708"
NAS-IP-Address = 192.168.0.40
Service-Type = Framed-User
Chargeable-User-Identity="some-login"
Message-Authenticator = 0x655ad71144647dd842afd3b65b08d421
```

Значения атрибутов:

- `User-Name` – IP-адрес пользователя в виде строки. Если conf-параметр `radius_user_name_ip=0`, то в качестве `User-Name` передается логин пользователя (см. ниже)
- `User-Password` – пароль, задается conf-параметром `radius_user_password`. Для всех пользователей пароль один и тот же.
- `Framed-IP-Address` – IPv4-адрес пользователя.
- `NAS-Port-Type` – задается значением conf-параметра `radius_attr_nas_port_type`. Перечень допустимых значений определен RFC 2865. В данном примере значение равно 5 (Virtual).
- `NAS-Port` – задается значением conf-параметра `radius_attr_nas_port`.
- `NAS-Port-Id` – задает VLAN, если есть. Вычитывается из L2 заголовка пакета. Если сеть поддерживает Q-in-Q, то есть содержит два заголовка VLAN, значением этого поля будет “VLAN1/VLAN2”, например, “`708/209`”. Если заголовок пакета не содержит VLAN, данный атрибут не включается в `Access-Request`.
- `NAS-IP-Address` – IP-адрес, заданный conf-параметром `radius_attr_nas_ip_address`. Обычно в качестве IP-адреса NAS задают IP-адрес сервера fdpi_pcrf. Либо можно не задавать `radius_attr_nas_ip_address`, но задать conf-параметр `radius_attr_nas_id` – текстовый идентификатор NAS. В этом случае Access-Request будет содержать атрибут `NAS-Identifier` вместо `NAS-IP-Address`.
- `Service-Type` – задается значением conf-параметра `radius_attr_service_type`. Полный список значений данного атрибута приведен в RFC 2865. В данном примере `Service-Type=2` (Framed-User).
- `Chargeable-User-Identity` (CUI) – задает логин пользователя, если он известен fastdpi. Если логин не известен, атрибут CUI содержит ровно один нулевой байт (nul CUI), что означает, согласно RFC 4372, что NAS запрашивает у radius-сервера логин пользователя. В ответе fdpi_pcrf ожидает увидеть в атрибуте

CUI правильный логин пользователя. Атрибут CUI включается в `Access-Request` только если conf-параметр `radius_attr_cui=1` (рекомендуемое значение).

- `Message-Authenticator` – вычисляемый атрибут, см. RFC 2869. Этот атрибут включается в `Access-Request`, если conf-параметр `radius_msg_auth_attr=1` (рекомендованное значение)

Если conf-параметр `radius_user_name_ip=0` (не рекомендуется), то fdpi_pcrf будет слать логин пользователя в атрибуте User-Name вместо IP-адреса:

```
User-Name = "some-login"
User-Password = "VasExperts.FastDPI"
Framed-IP-Address = 94.158.56.38
NAS-Port-Type = Virtual
NAS-Port = 0
NAS-Port-Id = "708"
NAS-IP-Address = 192.168.0.40
Service-Type = Framed-User
Chargeable-User-Identity="some-login"
Message-Authenticator = 0x655ad71144647dd842afd3b65b08d421
```

При этом может возникнуть следующая проблема: fastdpi может не знать логин пользователя (например, произошло рассогласование базы данных UDR сервера fastdpi). В этом случае в качестве логина используется значение conf-параметра `radius_unknown_user`, в качестве пароля – значение conf-параметра `radius_unknown_user_psw`, а атрибут `Chargeable-User-Identity` (CUI) будет состоять ровно из одного нулевого байта (nul CUI). FastDPI ожидает получить от Radius-сервера правильный логин пользователя в ответе, поэтому Radius-сервер должен различать эти две ситуации – логин известен или не известен, - и предпринимать соответствующие действия.

Формат ответов на Radius-запросы

Основными данными, помимо собственно факта авторизован или нет пользователь, которые необходимы fdpi_pcrf в ответ на `Access-Request`, являются:

- Профиль полисинга пользователя
- Профили пользователя по услугам fastdpi
- Какие услуги подключены пользователю
- Тип пользователя: сколько IP-адресов связано с пользователем, один или много.

Эти данные передаются в `Vendor-Specific` атрибуте [26] ответов.

```
VENDOR          VasExperts                    43823
BEGIN-VENDOR    VasExperts
ATTRIBUTE       VasExperts-Policing-Profile   1   string
ATTRIBUTE       VasExperts-Service-Profile    2   string
ATTRIBUTE       VasExperts-Enable-Service     3   string
ATTRIBUTE       VasExperts-Multi-IP-User      4   integer
END-VENDOR      VasExperts
```

`VendorID` для VasExperts = `43823`

Атрибуты:

- [1] `VasExperts-Policing-Profile` – строковый атрибут, задающий имя профиля полисинга для пользователя. В ответе на `Access-Request` или в CoA-нотификации (`Disconnect-Request`, `CoA-Request`) должно быть не более одного данного атрибута.
- [2] `VasExperts-Service-Profile` – строковый параметр, задающий имя профиля для конкретной услуги fastDPI. Формат строки:

  ```
  service_id:profile_name
  ```

  Где:

  - `service_id` – число, идентификатор услуги fastDPI
  - `profile_name` – строка, имя профиля по услуге

  PDU может содержать ноль или более (но не более пяти) атрибутов `VasExperts-Service-Profile` – по одному атрибуту для каждой услуги. Если услуге сопоставлен профиль, услуга считается подключенной.

- [3] `VasExperts-Enable-Service` – строковый параметр, задающий включение/отключение конкретной услуги. Формат строки:

  ```
  service_id:flag
  ```

  где:

  - `service_id` – число, идентификатор услуги fastDPI
  - `flag` – признак включения/отключения услуги. Допустимые значения:
    - `1`, `on`, `enabled` – услуга включена
    - `0`, `off`, `disabled` – услуга отключена

  Пример подключенной услуги: “`5:on`”
  Пример отключенной услуги: “`5:off`”
  Важное замечание: по умолчанию, для услуг действует правило “что не подключено, то отключено”, то есть если услуга явно не помечена как enabled (или не задан профиль услуги), то услуга считается отключенной. Но для услуги 4 (black list, фильтрация запрещенного трафика) действует более строгое правило: эту услугу нужно явно отключать для пользователя, если требуется, то есть для отключения услуги 4 “black-list” в ответе Radius-сервера должен явно присутствовать атрибут `VasExperts-Enable-Service` со значением ‘`4:off`’.
  По умолчанию подключение услуги 4 “black-list” регулируется глобальными настройками fastDPI, см. http://vasexperts.ru/wiki/doku.php?id=filtration_ctrl. Услуга 4 обычно глобально включена, чтобы не нарушать федеральное законодательство.
- [4] `VasExperts-Multi-IP-User` – признак, связано ли с данным пользователем множество IP-адресов или только один. Данный атрибут может быть либо байтом, либо 32-битовым числом. Значение `1` говорит о том, что данному пользователю может быть сопоставлено несколько IP-адресов, значение `0` – только один IP-адрес.

Также, если логин пользователя неизвестен fastDPI, в ответе на `Access-Request` должен присутствовать атрибут `Chargeable-User-Identity` (`CUI`, RFC 4372), содержащий логин пользователя. Ответ также может содержать атрибут `User-Name`, который также должен содержать логин пользователя. Если ответ содержит оба атрибута – `User-Name` и `CUI`, - `CUI` имеет более высокий приоритет.

Fdpi_pcrf анализирует оба возможных ответа на `Access-Request` – как `Access-Accept`, так и `Access-Reject` (`Access-Challenge` трактуется как `Access-Reject`).

Заметим, что ответы на Access-Request и CoA-оповещения хоть и похожи по своему внутреннему строению (множеству атрибутов), но обрабатываются по-разному. Это связано с тем, что положительный ответ на Access-Request *задает полные параметры* пользователя, тогда как CoA-нотификации оповещают об *изменениях* этих параметров.

Radius Access-Accept

Положительный ответ на Access-Request, - Access-Accept, – должен содержать следующие атрибуты:

- Framed-IP-Address –IP-адрес пользователя (тот же, что и в запросе). Это обязательный атрибут.
- User-Name – имя (логин) пользователя. Заметим, что в запросе Access-Request атрибут User-Name обычно содержит IP-адрес пользователя (это зависит от conf-параметрf radius_user_name_ip, см. “Формат Access-Request RADIUS-запроса”). В ответе Access-Accept этот атрибут должен содержать истинный логин пользователя, если не задан атрибут Chargeable-User-Identity.
- Chargeable-User-Identity (CUI) – имя (логин) пользователя. Если ответ Access-Accept содержит оба атрибута – User-Name и CUI - то CUI имеет более высокий приоритет (то есть User-Name игнорируется). Access-Accept должен обязательно содержать хотя бы один из атрибутов CUI или User-Name (CUI предпочтительнее)
- VasExperts-Policing-Profile – имя профиля полисинга для пользователя, обязательный атрибут. В Access-Accept допустимо не более одного атрибута VasExperts-Policing-Profile.
- VasExperts-Enable-Service – задает статус услуги: подключена или отключена. В Access-Accept можно указывать только подключенные услуги, все остальные услуги считаются отключенными. Каждая подключенная услуга должна задаваться отдельным атрибутом VasExperts-Enable-Service, то есть Access-Accept может содержать ноль или более атрибутов VasExperts-Enable-Service.
- VasExperts-Service-Profile – задает имя профиля услуги (то есть набор параметров услуги). Заметим, что не все услуги требуют (или допускают) наличие профиля. Если услуга имеет профиль, она считается подключенной, даже если атрибут VasExperts-Enable-Service говорит, что услуга отключена (то есть VasExperts-Service-Profile имеет более высокий приоритет, чем VasExperts-Enable-Service). Каждый профиль услуги задается отдельным атрибутом VasExperts-Service-Profile, то есть Access-Accept может содержать ноль или более атрибутов VasExperts-Service-Profile.
- VasExperts-Multi-IP-User – задает признак, один или много IP-адресов связано с данным пользователем. По умолчанию, если этот атрибут не задан, предполагается, что с пользователем связан только один IP-адрес. Следует учитывать, что этот атрибут задает важное свойство пользователя, весьма существенно влияющее на поведение fastDPI.
- Session-Timeout – опциональный атрибут, задает время действия авторизации в секундах. Если пакет не содержит данный атрибут, считается, что авторизация действует бессрочно (до тех пор, пока не придет Disconnect-Request или же CoA-Request с атрибутом Session-Timeout). По истечении этого времени статус авторизации пользователя устанавливается в “неизвестен”, что приводит к отправке запроса на авторизацию Access-Request.

Следует учитывать, что Access-Accept трактуется как исчерпывающий и корректный набор параметров пользователя: fastDPI корректирует свою внутреннюю БД в полном соответствии с атрибутами Access-Accept. Это значит, что при обнаружении расхождений между атрибутами Access-Accept и внутренней БД fastDPI предпочтение будет отдано атрибутам Access-Accept; более того, внутренняя БД fastDPI будет скорректирована соответствующим образом, чтобы параметры пользователя не отличались от присланных в Access-Accept.

Radius Access-Reject

В Access-Reject должны быть переданы особые профили для неавторизованного пользователя:

- профиль полисинга;
- имя профиля услуги 5 (белый список) – задает список сайтов, которые пользователь имеет право посещать.

Access-Challenge трактуется как Access-Reject.

Access-Reject должен содержит следующие атрибуты:

- Framed-IP-Address –IP-адрес пользователя (тот же, что и в запросе). Это обязательный атрибут.
- User-Name – имя (логин) пользователя. Заметим, что в запросе Access-Request атрибут User-Name обычно содержит IP-адрес пользователя (это зависит от conf-параметрf radius_user_name_ip, см. “Формат Access-Request RADIUS-запроса”). В ответе Access-Reject этот атрибут должен содержать истинный логин пользователя, если не задан атрибут Chargeable-User-Identity.
- Chargeable-User-Identity (CUI) – имя (логин) пользователя. Если ответ Access-Reject содержит оба атрибута – User-Name и CUI - то CUI имеет более высокий приоритет (то есть User-Name игнорируется). Access-Reject должен обязательно содержать хотя бы один из атрибутов CUI или User-Name (CUI предпочтительнее)
- VasExperts-Policing-Profile – имя профиля полисинга для пользователя; если этот атрибут отсутствует в Access-Reject, то пользователю сопоставляется профиль по умолчанию, заданный conf-параметром default_reject_policing. В Access-Reject допустимо не более одного атрибута VasExperts-Policing-Profile.
- VasExperts-Service-Profile – имя профиля услуги 5 (белый список). Если этот атрибут отсутствует в Access-Reject, то пользователю сопоставляется профиль по услуге 5, заданный conf-параметром default_reject_whitelist. Задание профилей для других услуг игнорируется.

Radius CoA

CoA (Change of Authorization, RFC 5176) нотификации оповещают об изменении параметров пользователя (CoA-Request) или о потере авторизации (Disconnect-Request), например, вследствие исчерпания средств на счете.

CoA-Request

CoA-Request нотификация говорит о том, что пользователь авторизован и, опционально, у него изменились некоторые параметры. Таким образом, CoA-Request может приходить в следующих случаях:

- пользователь перешел из состояния “не авторизован” в состояние "авторизован” (например, пополнил счет) – см. далее “CoA-Request для неавторизованного пользователя”
- У авторизованного пользователя изменились параметры (подключение/отключение услуг, изменение профилей услуг).

Если пользователь не авторизован и изменяются его параметры, `CoA-Request` не должен генерироваться. Если все же нотификация `CoA-Request` для неавторизованного пользователя генерируется, она будет обработана fastDPI специальным образом, см. далее.

Для авторизованного пользователя нотификация `CoA-Request` **содержит только изменения параметров пользователя**; поддерживаются следующие атрибуты:

- `User-Name` – имя (логин) пользователя. `CoA-Request` должен обязательно содержать хотя бы один из атрибутов `CUI` или `User-Name` (`CUI` предпочтительнее).
- `Chargeable-User-Identity` (`CUI`) – имя (логин) пользователя. Если `CoA-Request` содержит оба атрибута – `User-Name` и `CUI` - то `CUI` имеет более высокий приоритет (то есть `User-Name` игнорируется). `CoA-Request` должен обязательно содержать хотя бы один из атрибутов `CUI` или `User-Name` (`CUI` предпочтительнее).
- `VasExperts-Multi-IP-User` – задает *изменение* признака, один или много IP-адресов связано с данным пользователем. Если пользователь становится многоадресным (то есть с одним пользователем может быть связано много IP-адресов), данный атрибут должен быть установлен в `1`. Если пользователь становится одноадресным – данный атрибут должен быть установлен в `0`. Если признак “multi-IP” пользователя не изменяется, CoA-Request не должен содержать атрибут `VasExperts-Multi-IP-User`. Допустимо не более одного данного атрибута в `CoA-Request`.
- `VasExperts-Policing-Profile` – имя профиля полисинга для пользователя. Данный атрибут должен включаться только если изменился профиль полисинга пользователя. В `CoA-Request` допустимо не более одного атрибута `VasExperts-Policing-Profile`.
- `VasExperts-Enable-Service` – задает изменение статуса услуги: подключена (`on`) или отключена (`off`). В `CoA-Request` указываются все услуги, статус подключения которых изменился. Если некоторая услуга не содержится в `CoA-Request`, то это значит, что статус её “подключенности” не изменился для пользователя. Каждая сменившая статус услуга должна задаваться отдельным атрибутом `VasExperts-Enable-Service`, то есть `CoA-Request` может содержать ноль или более атрибутов `VasExperts-Enable-Service`.
- `VasExperts-Service-Profile` – задает имя нового профиля услуги (то есть набор параметров услуги). Если данная услуга отключена, она включается (то есть `VasExperts-Service-Profile` имеет более высокий приоритет, чем `VasExperts-Enable-Service`). Для того, чтобы отключить услугу с профилем, следует задать для неё атрибут `VasExperts-Enable-Service` со значением “`off`” (например, для услуги 5: `VasExperts-Enable-Service="5:off"`). Каждое изменение имени профиля услуги задается отдельным атрибутом `VasExperts-Service-Profile`, то есть `CoA-Request` может содержать ноль или более атрибутов `VasExperts-Service-Profile`.
- `Session-Timeout` – опциональный атрибут, задает время действия авторизации в секундах. Значение 0 игнорируется. По истечении этого времени статус авторизации

пользователя устанавливается в “неизвестен”, что приводит к отправке запроса на авторизацию Access-Request.

CoA-Request для неавторизованного пользователя

Существует неоднозначность нотификации CoA-Request: ситуация, когда пользователь не авторизован, но у него изменились некоторые параметры. Например, у пользователя кончились средства на счете, мы получили нотификацию Disconnect-Request, а затем пользователь отключает некоторую услугу. Отключение услуги – это изменение параметров пользователя, значит, нужно послать оповещение CoA-Request, но в данный момент пользователь не авторизован, а CoA-Request применяется только для авторизованных пользователей.

RFC 5176 п. 3.2 предлагает такое решение данной коллизии: если пользователь не авторизован, любые изменения его параметров не приводят к генерации оповещения CoA-Request. Когда статус пользователя изменяется на “авторизован” (например, пользователь пополнил свой счет), Radius-сервер должен послать нотификацию CoA-Request, сформированную особым образом: нотификация должна включать атрибут Service-Type со значением 8 (Authorize Only); CoA-сервер (в нашем случае это fdpi_pcrf), получив такой запрос, должен запросить авторизацию, то есть сформировать полноценный Access-Request. fdpi_pcrf поддерживает такой алгоритм работы, но не имеет достаточно данных для формирования полноценного Access-Request. Поэтому fdpi_pcrf, в полном соответствии с RFC 5176, при получении CoA-Request с атрибутом Service-Type=8 отвечает CoA-NAK с атрибутом Error-Cause=405 (Unsupported service), но при этом формирует особый внутренний запрос к fastDPI “сбрось данные авторизации”. FastDPI при получении такого запроса устанавливает статус авторизации в “неизвестно”, тем самым fastDPI как бы ставит сам себе метку “следует запросить статус авторизации для данного IP-адреса, когда в потоке появится пакет с этого IP”. При появлении пакета с локального IP-адреса с неизвестным статусом авторизации fastDPI сформирует через fdpi_pcrf Radius-запрос Access_Request, что и требуется.

Таким образом, если статус пользователя меняется с “неавторизован” на “авторизован”, fdpi_pcrf ожидает нотификацию CoA-Request со следующими обязательными атрибутами:

- Service-Type=8 (Authorize-Only)
- User-Name – имя (логин) пользователя. CoA-Request должен обязательно содержать хотя бы один из атрибутов CUI или User-Name (CUI предпочтительнее).
- Chargeable-User-Identity (CUI) – имя (логин) пользователя. Если CoA-Request содержит оба атрибута – User-Name и CUI - то CUI имеет более высокий приоритет (то есть User-Name игнорируется). CoA-Request должен обязательно содержать хотя бы один из атрибутов CUI или User-Name (CUI предпочтительнее).

Все прочие атрибуты игнорируются.

Этот трюк с Service-Type=8 можно применить и тогда, когда нет возможности послать в CoA только те параметры, которые изменились. Получение CoA-Request с Service-Type=8 для fdpi_pcrf является сигналом реавторизации пользователя: будет послан обычный запрос Access_Request, на который ожидается получение полного профиля пользователя.

Disconnect-Request

Нотификация `Disconnect-Request` сигнализирует о том, что пользователь стал неавторизованным (например, закончились средства на счете). Нотификация `Disconnect-Request` содержит следующие атрибуты:

- `User-Name` – имя (логин) пользователя. `Disconnect-Request` должен обязательно содержать хотя бы один из атрибутов `CUI` или `User-Name` (`CUI` предпочтительнее).
- `Chargeable-User-Identity` (`CUI`) – имя (логин) пользователя. Если `Disconnect-Request` содержит оба атрибута – `User-Name` и `CUI` - то `CUI` имеет более высокий приоритет (то есть `User-Name` игнорируется). `Disconnect-Request` должен обязательно содержать хотя бы один из атрибутов `CUI` или `User-Name` (`CUI` предпочтительнее).
- `VasExperts-Policing-Profile` – имя профиля полисинга для пользователя; если этот атрибут отсутствует в `Disconnect-Request`, то пользователю сопоставляется профиль по умолчанию, заданный conf-параметром `default_reject_policing`. В `Disconnect-Request` допустимо не более одного атрибута `VasExperts-Policing-Profile`.
- `VasExperts-Service-Profile` – имя профиля услуги 5 (белый список). Если этот атрибут отсутствует в `Disconnect-Request`, то пользователю сопоставляется профиль по услуге 5, заданный conf-параметром `default_reject_whitelist`. Задание профилей для других услуг игнорируется.
-

Запись .pcap-файлов

Fdpi_pcrf позволяет записывать Radius и CoA запросы и ответы в pcap-файлы для последующего анализа проблемных случаев взаимодействия с внешними системами. По умолчанию запись pcap-файлов отключена. Включается запись следующими параметрами .conf-файла:

- `radius_dump_pcap` – задает режим записи Radius-пакетов: 0 – не записывать, 1 – записывать только ошибочные пакеты (обычно это ответы), 2 – записывать все запросы и ответы. Имя pcap-файла строится по маске: `radius_mmddhhmmss_xxx.pcap`
- `coa_dump_pcap` - задает режим записи Radius CoA-пакетов: 0 – не записывать, 1 – записывать только ошибочные пакеты (обычно это входящие оповещения), 2 – записывать все оповещения и ответы. Имя pcap-файла строится по маске: `coa_mmddhhmmss_xxx.pcap`

Эти опции являются горячими, то есть поддерживают изменение без перезапуска fdpi_pcrf.

pcap-файлы размещаются в каталоге, заданном в conf-параметре `pcap_dump_path` (по умолчанию - `/var/log/dpi`). Размер одного файла не может превышать значения, задаваемого conf-параметром `pcap_dump_fsize` (по умолчанию – 1 Гигабайт). По достижении этого размера текущий pcap-файл закрывается и создается новый.

Следует учитывать, что ***fdpi_pcrf никогда не удаляет записанные pcap-файлы***, поэтому рекомендуется включать опции `radius_dump_pcap`/`coa_dump_pcap` временно, на короткий промежуток.

При записи заголовки уровней L1 (Ethernet frame) и L2 (IP header) эмулируются.

Конфигурирование FreeRadius

В этом разделе приведены минимальные изменения в конфигурацию FreeRadius. Предположим, IP-адрес Radius-сервера есть `192.168.1.200`, порт 1812.

Словарь VasExperts

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

- копируем `dictionary.vasexperts` из дистрибутива fastdpi в каталог `$freeRadius/share/freeradius`
- Добавляем в главный словарь `$freeRadius/share/freeradius/dictionary` строку:

```
$INCLUDE dictionary.vasexperts
```

Создание клиента

В файле конфигурации `fdpi_pcrf.conf` нашего 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`.

Имеем в виду, что настройки Radius-сервера и клиента должны совпадать!

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

Добавляем 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 fdpi_pcrf, у нас это `192.168.1.32`
- `secret` – уникальный секрет, известный Radius-серверу и клиенту (то есть инстансу fdpi_pcrf). Значение строки секрета вы выбираете сами. Заметим, что тот же самый секрет прописан в настройках `fdpi_pcrf.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 `fdpi_pcrf.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, - нам нужен истинный логин пользователя. Поэтому `add_cui` здесь закомментировано.

- `virtual_server` – задает имя виртуального сервера, который мы сконфигурируем далее.

Создание виртуального сервера

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

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

```
ipaddr = 192.168.1.200
port = 1814
interface = eth0
```

- остальные секции `listen` удаляем (или закомментируем – они нам не нужны)
- всю основную работу по составлению ответа на `Access-Request` прописываем 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` (не забываем, наличие в запросе `Access-Request` атрибута `CUI` (`Chargeable-User-Identity`), содержащего единственный нулевой байт, означает, что fdpi_pcrf 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` оставляем закомментированным! FreeRadius вместо логина пользователя посылает в CUI хеш-значение логина, что нам не нужно, поэтому атрибут CUI в ответе будем формировать сами, см. пример выше.
- Ниже в секцию `Post-Auth-Type REJECT` (формирование `Access-Reject`) добавляем

- Формирование атрибута CUI, если fdpi_pcrf 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.

См. “RADIUS ACCESS-REJECT”. 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 fdpi_pcrf:

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

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

Первая запись задает имя пользователя, которое шлет fdpi_pcrf если логин ему не известен, подробнее см. описание conf-параметра `radius_unknown_user`. Это имя настраивается в fdpi_pcrf, так же как и пароль, см. conf-параметр `radius_unknown_user_psw`.

Вторая запись задает пароль, с которым fdpi_pcrf шлет запросы для известных логинов. Этот пароль настраивается в fdpi_pcrf, см. conf-параметр `radius_user_password`.