

Table of Contents

Быстрый старт: Тарифный план и Captive Portal (доступ к СЗР)	3
Введение	3
<i>Распределение трафика по классам для тарифного плана</i>	3
<i>Создание тарифного плана</i>	4
<i>Подготовка Captive Portal с доступом к платежным системам и Социально-значимым ресурсам (СЗР)</i>	5
<i>Интеграция с биллингом без Radius</i>	7

Быстрый старт: Тарифный план и Captive Portal (доступ к СЗР)

Введение

Для реализации [BRAS](#) в данном разделе приведен пример создания двух тарифных планов (policing) и услуги Captive Portal:

- **rate_10M** - базовый тарифный план, который используется после авторизации абонента.
- **blocked** - тарифный план, который используется для блокировки абонента и предоставления доступа только по определенным протоколам. Его имя необходимо передавать в Radius Access-Reject.
- При блокировке совместно с тарифным планом **blocked** абоненту назначается (Услуга 5) **my_white_list** - Белый список ресурсов, на которые у него есть доступ для оплаты услуги. Также осуществляется редирект для HTTP ресурсов на Captive Portal.



Имена данных профилей необходимо передавать в соответствующих атрибутах сообщений Radius Access-Accept, Access-Reject.

Распределение трафика по классам для тарифного плана

Для разметки приоритетов используем опцию [Назначение приоритетов в зависимости от протокола](#).

1. Создаем файл **protocols.txt** с описанием групп протоколов, которые мы хотим выделить из общего трафика, и назначенных им приоритетов (классов):

```
dns cs0
ICMP cs0
http cs0
https cs0
QUIC cs1
default cs2
bittorrent cs7
```

где

- cs0 соответствует приоритету 0, class0 соответственно
- cs1 - приоритету 1, class1
- cs7 - приоритету 7, низший класс



Выделенные таким образом классы можно использовать в описании тарифных



планов, вводя для них отдельные ограничения, кроме того в соответствии с ними будет производиться приоритезация протоколов в полосе.

2. Конвертируем его в формат dscp, который понимает fastDPI

```
cat protocols.txt|lst2dscp /etc/dpi/protocols.dscp
```

3. Применяем настройки

```
service fastdpi reload
```

Создание тарифного плана

Для организации абонентской полосы согласно тарифному плану используем опцию [Распределение канала доступа между абонентами](#).

1. Для каждого тарифного плана абонента в биллинге создаем файл конфигурации, с описанием его настроек для DPI.



Удобное соглашение: имена файлов конфигурации, с описанием настройки тарифного плана на DPI, сделать совпадающим с именем тарифного плана в биллинге.

Пример описания для тарифа 10mbit, название в биллинге "rate_10M"

Создаем файл rate_10M.cfg

```
htb_inbound_root=rate 10mbit
htb_inbound_class0=rate 4mbit ceil 10mbit
htb_inbound_class1=rate 3mbit ceil 10mbit
htb_inbound_class2=rate 8bit ceil 10mbit
htb_inbound_class3=rate 8bit ceil 10mbit
htb_inbound_class4=rate 8bit ceil 10mbit
htb_inbound_class5=rate 8bit ceil 10mbit
htb_inbound_class6=rate 8bit ceil 10mbit
htb_inbound_class7=rate 8bit ceil 10mbit
htb_root=rate 10mbit
htb_class0=rate 4mbit ceil 10mbit
htb_class1=rate 3mbit ceil 10mbit
htb_class2=rate 8bit ceil 10mbit
htb_class3=rate 8bit ceil 10mbit
htb_class4=rate 8bit ceil 10mbit
htb_class5=rate 8bit ceil 10mbit
htb_class6=rate 8bit ceil 10mbit
htb_class7=rate 8bit ceil 10mbit
```

Примечания:

- htb_class0-1 - имеют гарантированную скорость в 4Мбит/с и 3Мбит/с соответственно
- htb_class7 - минимальную полосу 8bit, что означает, что может зажиматься в 0 Мбит/с (0 - указывать нельзя зарезервировано)

2. Создаем тарифный план с именем **rate_10M**

```
fdpi_ctrl load profile --policing /path/to/rate_10M.cfg --profile.name rate_10M
```

3. Чтобы наши настройки для абонентов, которые мы сделаем в дальнейшем, не пропали при перезагрузке DPI подключаем [БД UDR](#)

```
udr=1
```

4. Применяем настройки через перезапуск fastDPI

```
service fastdpi restart
```

Подготовка Captive Portal с доступом к платежным системам и Социально-значимым ресурсам (СЗР)



Услуга 5 (Белые списки и Captive Portal) регулирует доступ только TCP-based протоколам. Для того, чтобы ограничить доступ к остальным ресурсам с использованием различных протоколов, необходимо использовать соответствующий профиль тарифного плана, который пропускает трафик только определенных классов.

1. Создаем описание тарифного плана для абонентов в блокировке blocked.cfg. Разрешаем только трафик [cs0](#), с протоколами [согласно списка в п.1](#).

```
htb_inbound_root=rate 10mbit
htb_inbound_class0=rate 1mbit ceil 10mbit
htb_inbound_class1=rate 8bit ceil 8bit
htb_inbound_class2=rate 8bit ceil 8bit
htb_inbound_class3=rate 8bit ceil 8bit
htb_inbound_class4=rate 8bit ceil 8bit
htb_inbound_class5=rate 8bit ceil 8bit
htb_inbound_class6=rate 8bit ceil 8bit
htb_inbound_class7=rate 8bit ceil 8bit
htb_root=rate 10mbit
htb_class0=rate 1mbit ceil 10mbit
htb_class1=rate 8bit ceil 8bit
htb_class2=rate 8bit ceil 8bit
htb_class3=rate 8bit ceil 8bit
htb_class4=rate 8bit ceil 8bit
```

```
htb_class5=rate 8bit ceil 8bit
htb_class6=rate 8bit ceil 8bit
htb_class7=rate 8bit ceil 8bit
```

2. Создаем тарифный план с именем **blocked** для заблокированного абонента

```
fdpi_ctrl load profile --policing /path/to/blocked.cfg --profile.name
blocked
```

3. Создаем список сайтов, доступных в режиме Captive Portal. Подробнее в описании опции [Белый список](#).

Создаем файл **my_white_url_list.txt** с url сайтов платежных систем. Каждая строка файла содержит один url (без префикса http://), рекомендуется включать также и субдомены, например:

```
online.sberbank.ru
*.online.sberbank.ru
```

Для добавления **Социально-значимых ресурсов** необходимо [скачать архив с VAS Cloud](#) распаковать и добавить содержимое файлов из архива к вашим спискам **до** конвертации. Содержание архива:

```
url_list.txt - URL для HTTP запросов
cn_list.txt - Common Name для HTTPS запросов
sni_list.txt - Server Name Indication для HTTPS запросов
ip_list.txt - IP адреса
```

Для формирования белого списка рекомендуем использовать готовый список.



1. [Перечень платежных систем на github](#)
2. [Список банков подготовленный нашими партнерами](#)
3. [Список социально-значимых ресурсов на VAS Cloud](#)

4. Конвертирование во внутренний формат:

```
cat my_white_url_list.txt|url2dic my_url_list.bin
cat my_white_cn_list.txt|url2dic my_cn_list.bin
cat my_white_sni_list.txt|url2dic my_sni_list.bin
cat my_white_ip_list.txt|ip2bin my_ip_list.bin
```

Любой из списков может отсутствовать. Подробнее в описании опции [Белый список](#).



Чтобы исключить блокировку для HTTPS сайтов нужно подготовить белый список для CN и SNI с **символом ***, сигнализирующего что CN и SNI может быть любой.

5. Создаем **именованный профиль** для белого списка

```
fdpi_ctrl load profile --service 5 --profile.name my_white_list --  
profile.json '{ "url_list" : "/path/to/my_url_list.bin" , "sni_list" :  
"/path/to/my_sni_list.bin", "cn_list" : "/path/to/my_cn_list.bin", "ip_list"  
: "/path/to/my_ip_list.bin", "redirect" : "mysite.ru/block" }'
```

где

- redirect - страница переадресации¹⁾²⁾
- url_list: белый список URL
- sni_list: белый список SNI
- cn_list: белый список Common Name³⁾
- ip_list: белый список IP адресов **включающий СЗР**

Интеграция с биллингом без Radius



Если у в сети все же используется Radius, но вы не предполагаете настраивать взаимодействие СКАТ с биллингом через него и имеете динамические IP адреса, необходимо использовать [Radius-монитор](#), который добавит связку IP-Login в UDR.

1. Проводим интеграцию с биллингом

Вариант интеграции зависит от того, обладает ли биллинг возможностью управления оборудованием по событиям или нет.

1а. Биллинг умеет управлять оборудованием по событиям: создание абонента, смена тарифного плана, блокировка

В этом случае выбираем тип оборудования с управлением по SSH/RSN⁴⁾ или с помощью выполнения локальных скриптов и заносим в настройки соответствующих команд (или скриптов) команды подключения (смены) тарифного плана:

```
fdpi_ctrl load --policing ${rateplan}.cfg --ip ${ip_address}  
или  
fdpi_ctrl load --policing ${rateplan}.cfg --login ${login}
```

где

- \${rateplan} - переменная куда биллинг подставит имя тарифного плана абонента rate_10M
- \${ip_address} - сюда биллинг подставит ip адрес 192.168.0.1 абонента (для абонентов с фиксированным ip)
- \${login} - сюда биллинг подставит login абонента dom1kv2 (для абонентов с динамическим ip, несколькими ip, или просто мы хотим управлять по login)

1б. Биллинг не умеет управлять оборудованием по событиям

Настроим выгрузку данных из биллинга по расписанию в crontab. В файлы с именами имя_тарифного_плана.lst выгружаем из биллинга список абонентов с соответствующими тарифными планами (список может содержать ip или login) и запускаем загрузку этих данных в dpi

```
fdpi_ctrl load --policing rate_10M.cfg --file rate_10M.lst
fdpi_ctrl load --policing rate_20M.cfg --file rate_20M.lst
...
или (для всех сразу)
for rateplan in *.cfg; do fdpi_ctrl load --policing "$rateplan" --file
"${rateplan%%.*}".lst; done
```

2. Помещаем абонента в Captive Portal⁵⁾

```
fdpi_ctrl load --policing blocked.cfg --ip ${ip_address}
fdpi_ctrl load --service 5 --ip ${ip_address}
```

3. После оплаты отключаем абонента Captive Portal и восстанавливаем его тарифный план

```
fdpi_ctrl load --policing ${rateplan}.cfg --ip ${ip_address}
fdpi_ctrl del --service 5 --ip ${ip_address}
```

¹⁾

Внимание если указываете https сайт, то обязательно требуется данный домен внести в список SNI иначе домен будет заблокирован

²⁾

доп. параметры можно дописать (по правилам http) только после ? или &, их надо обязательно указывать в url для белого списка и тут надо подумать за dpi, иначе dpi припишет /?

³⁾

проверка по ip:port или sname осуществляется если в запросе отсутствуют url или sni

⁴⁾

При необходимости можно доустановить на dpi дополнительное ПО, совместимое с OS Linux, для расширения возможностей удаленного управления, например, telnet сервер.

⁵⁾

Если событийное управление не поддерживается, то делаем через выгрузку заблокированных и разблокированных абонентов в файл blocked.lst и unblocked.lst