Easyhotspot и NetFlow

Настройка Easyhotspot для учета ресурсов, посещаемых клиентами хотспотов

составил **Дмитрий Харций**
г. Запорожье, **2020 г.**

dmitrykhn@aol.com

Оглавление

Введение

Цель данного Руководства — показать, как в программе Easyhotspot включить, настроить и использовать функцию сбора сведений о ресурсах, посещенных клиентами хотспотов*. Для этого используется протокол NetFlow. Пару цитат из вики [1]:

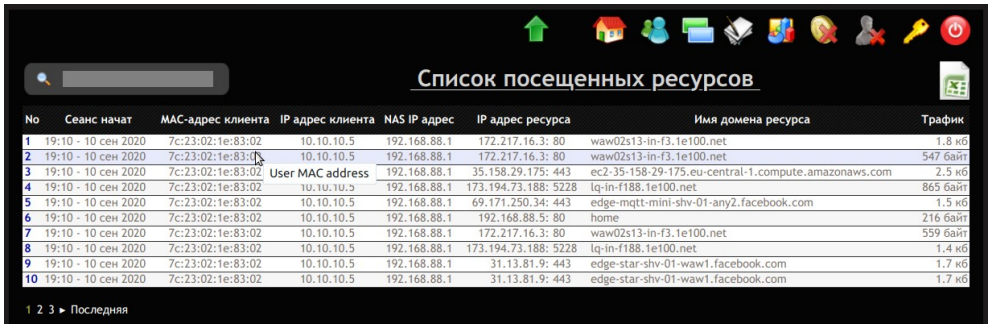
***NetFlow** — сетевой протокол, разработанный компанией Cisco Systems и предназначенный для учёта сетевого трафика. На сегодняшний день он «де-факто» является промышленным стандартом и поддерживается не только оборудованием Cisco, но и устройствами многих других изготовителей (например, Mikrotik). Кроме того, существуют свободные реализации П/О для *NIX-систем, использующего указанный протокол.*

Для сбора информации о трафике по протоколу NetFlow в системе должны присутствовать такие компоненты:

- ***Сенсор** — собирает статистику по проходящему через него трафику. Сенсор размещается непосредственно на пути прохождения трафика от клиентов (хотспотов) в сеть (интернет). Собранные сведения о трафике сенсор отсылает коллектору. Когда в системе используются «удаленные» роутеры (например, Mikrotik), то сбор данных осуществляют сенсоры, установленные непосредственно в них. Если же сервер Easyhotspot используется в качестве шлюза «локального» хотспота, то в таком случае сенсор должен быть установлен на самом сервере.*
- ***Коллектор** — собирает данные, поступающие со всех сенсоров, и помещает их в свое собственное (локальное) хранилище (файлы «дампов»).*
- ***Анализатор** — анализирует собранные коллектором данные и формирует пригодные для чтения человеком отчёты (часто в виде графиков).*

В биллинге Easyhotspot все происходит точно так же, как описано в википедии, за одним лишь небольшим исключением: его «Анализатор» (который далее в этом документе будет называться «***парсер***») не строит никаких графиков, а вместо этого просто вносит всю собранную информацию о трафике в базу данных программы. И затем, уже сам биллинг либо покажет вам собранные сведения в виде таблиц, либо позволит экспортировать их в файлы, которые вы сможете открывать и обрабатывать во внешних программах, в таких, например, как Microsoft Excel.

Пример отображения информации о трафике в программе Easyhotspot:



No	Сеанс начат	MAC-адрес клиента	IP адрес клиента	NAS IP адрес	IP адрес ресурса	Имя домена ресурса	Трафик
1	19:10 - 10 сен 2020	7c:23:02:1e:83:02	10.10.10.5	192.168.88.1	172.217.16.3: 80	waw02s13-in-f3.1e100.net	1.8 кб
2	19:10 - 10 сен 2020	7c:23:02:1e:83:02	10.10.10.5	192.168.88.1	172.217.16.3: 80	waw02s13-in-f3.1e100.net	547 байт
3	19:10 - 10 сен 2020	7c:23:02:1e:83:02	10.10.10.5	192.168.88.1	35.158.29.175: 443	ec2-35-158-29-175.eu-central-1.compute.amazonaws.com	2.5 кб
4	19:10 - 10 сен 2020	7c:23:02:1e:83:02	10.10.10.5	192.168.88.1	173.194.73.188: 5228	lq-in-f188.1e100.net	865 байт
5	19:10 - 10 сен 2020	7c:23:02:1e:83:02	10.10.10.5	192.168.88.1	69.171.250.34: 443	edge-mqtt-mini-shv-01-any2.facebook.com	1.5 кб
6	19:10 - 10 сен 2020	7c:23:02:1e:83:02	10.10.10.5	192.168.88.1	192.168.88.5: 80	home	216 байт
7	19:10 - 10 сен 2020	7c:23:02:1e:83:02	10.10.10.5	192.168.88.1	172.217.16.3: 80	waw02s13-in-f3.1e100.net	559 байт
8	19:10 - 10 сен 2020	7c:23:02:1e:83:02	10.10.10.5	192.168.88.1	173.194.73.188: 5228	lq-in-f188.1e100.net	1.4 кб
9	19:10 - 10 сен 2020	7c:23:02:1e:83:02	10.10.10.5	192.168.88.1	31.13.81.9: 443	edge-star-shv-01-waw1.facebook.com	1.7 кб
10	19:10 - 10 сен 2020	7c:23:02:1e:83:02	10.10.10.5	192.168.88.1	31.13.81.9: 443	edge-star-shv-01-waw1.facebook.com	1.7 кб

1 2 3 ▸ Последняя

Рис. 1 — Меню с информацией о трафике (посещенных ресурсах) в программе Easyhotspot

ВАЖНОЕ ПРИМЕЧАНИЕ:

- Использование протокола NetFlow предоставляет вам возможность сбора информации только об IP-адресах ресурсов, посещенных клиентами сетей хотспотов. Эта функция ***НЕ МОЖЕТ ПРЕДОСТАВИТЬ ИНФОРМАЦИЮ О ТОМ, КАКИЕ ИМЕННО СТРАНИЦЫ клиенты просматривали*** на удаленных сайтах! В лучшем случае, ***вам будут известны лишь адреса сайтов, посещенных клиентами*** (благодаря тому, что скрипт-«парсер» перед тем, как записать собранные сведения в базу, может выполнять т. н. «обратные DNS-запросы» с целью преобразования IP-адресов в «символьные имена доменов»).

Установка на сервер необходимых программ

Число программ, которые нужно будет установить на сервер, зависит от того, в каком режиме ваш сервер работает. Если сервер Easyhotspot обслуживает только удаленные («внешние») роутеры, то установить на него нужно NetFlow-коллектор (программу **nfdump** [2]) и скрипт-парсер (который будет переносить всю статистику непосредственно в базу данных). Если же сервер работает еще и в качестве шлюза локального хотспота (т. е., на сервере установлена программа Coova-Chilli), то в таком случае, в дополнение к двум вышеуказанным программам, установке подлежит еще и NetFlow-сенсор (программа **fprobe** [3]). Исходя из вышесказанного, определитесь для себя с ответом на вопрос, что именно вы будете устанавливать на ваш сервер.

Дополнительно, перед началом установки найдите такие сведения о своем хотспоте, которые вам понадобятся во время настройки функции сбора NetFlow-статистики, а именно:

- Адрес вашего сервера с биллингом Easyhotspot (вам нужно будет указать его в настройках NetFlow-сенсоров, чтобы они знали, куда отсылать собранные данные);
- Диапазоны IP-адресов, которые ваши хотспоты раздают своим клиентам (их нужно будет прописать в настройках парсера, чтобы тот не засорял базу «лишними данными», т. к., NetFlow-сенсоры на самом деле ведут себя как «еще те штирлицы» — а именно, «пишут все подряд»!). Если же у вас много хотспотов, у которых отличаются диапазоны адресов используемых DHCP-серверами, не огорчайтесь — скрипт парсера позволит вам указать в настройках любое нужное число отличающихся диапазонов адресов;

Установка NetFlow-коллектора nfdump

Nfdump — это не одна программа, а практически целый набор программ для сбора и обработки данных NetFlow. Устанавливаются они все вместе, одним пакетом, но в дальнейшем Easyhotspot использует из них только две — программу *nfcapd*, которая получает сетевые данные NetFlow и сохраняет их в «специальные» файлы («дампы») в своем «специальном» формате (собственно это и есть сам «коллектор»), и программу *nfdump*, которая считывает данные из этих «специальных» файлов и преобразует их в некий «стандартный» вид, приемлемый для дальнейшей обработки парсером.

НАПОМИНАЮ, ЧТО, ЕСЛИ ВЫ ХОТИТЕ СОХРАНЯТЬ СВЕДЕНИЯ NETFLOW, УСТАНОВКА ПРОГРАММЫ NFDUMP НУЖНА ВАМ В ЛЮБОМ СЛУЧАЕ ВНЕ ЗАВИСИМОСТИ ОТ ТОГО, БУДЕТ ВАШ БИЛЛИНГ ОБСЛУЖИВАТЬ ТОЛЬКО «ВНЕШНИЕ» РОУТЕРЫ, ИЛИ ЖЕ И ЛОКАЛЬНЫЙ ХОТСПОТ — ТОЖЕ (УСТАНОВЛЕННУЮ НА СЕРВЕРЕ ПРОГРАММУ COOVA-CHOLLI) !

Скрипт-инсталятор Easyhotspot, если вы устанавливаете биллинг именно с его помощью, просто задаст вам вопрос о том, хотите ли вы установить программу nfdump*:

```
Программа Easyhotspot может вести учет адресов, посещенных клиентами.
Для работы этой функции необходима установка на сервер программы т.н.
NetFlow-коллектора (службы nfcapd). С другой стороны, если вам не
нужен учет посещенных клиентами ресурсов, то тогда не рекомендуется
устанавливать эту программу (т.к. весь этот учет значительно повышает
нагрузку на сервер и радикально увеличивает число записей в базе
данных). Дополнительо УТОЧНЯЮ - nfcapd используется для сбора ВСЕХ
данных NetFlow, поступающих от любых источников (как от локального
хотспота (контроллера Coova-Chilli, установленного на самом сервере
Easyhotspot), так и от всех внешних роутеров)!

А теперь ответьте на вопрос:
Выполнить установку NetFlow-коллектора nfcapd на ваш сервер?
Да или нет - [Y/N]:
```

Рис. 2 — Предложение установить на сервер Easyhotspot программу nfdump

Чтобы установить ее, просто ответьте на данный вопрос утвердительно. Скрипт сам выполнит установку **и минимально необходимую настройку** программы.

Если же вы устанавливаете сервер биллинга ***вручную***, то вам нужно будет выполнить следующие действия:

1. Войдите в консоль (терминал) и введите такую команду:

   sudo apt-get install nfdump

2. После того, как установка пакета будет завершена, вам нужно настроить параметры программы. Для этого введите команду:

   sudo mcedit /lib/systemd/system/nfdump.service

3. В открывшемся файле найдите строку

   ExecStart=/usr/bin/nfcapd -D -l /var/cache/nfdump -P /var/run/nfcapd.pid -p 2055

4. Отредактируйте ее, чтобы в итоге она стала выглядеть следующим образом*:

   ExecStart=/usr/bin/nfcapd -T all -D -l /var/cache/nfdump -P /var/run/nfcapd.pid -p 2055 -x /root/parse_nfcap.pl

   (учтите, что в файле после вашего редактирования вышеприведенный код ДОЛЖЕН ПО ПРЕЖНЕМУ БЫТЬ ВПИСАН ОДНОЙ СТРОКОЙ, это он просто в данном документе в одну строку не поместился!)

5. Сохраните изменения (**F2**) и выйдите из редактора (**F10**).

6. Так как содержимое файла-«задачи» (nfdump.service) изменилось, введите команду:

   sudo systemctl daemon-reload

7. После этого перезапустите службу nfdump:

   sudo systemctl restart nfdump

На этом «ручная» установка и минимально необходимая настройка NetFlow-коллектора nfdump завершена. Вы можете проверить, все ли было выполнено правильно, введя команду:

sudo systemctl status nfdump

В ответ вы должны получить подобное сообщение:

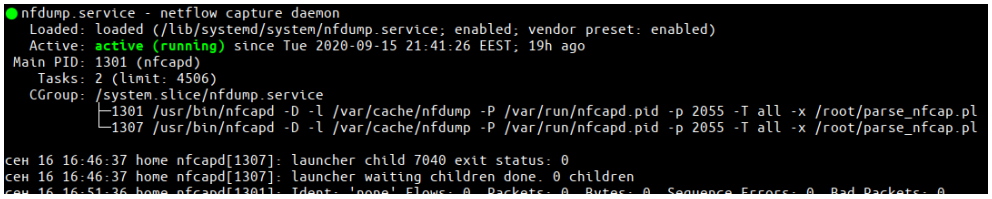


Рис. 3 — Результат проверки статуса программы nfdump

В нем особое внимание вы должны обратить на две вещи — статус (в приведенном примере он выделен зеленым цветом, сообщение «**active (running)**» в нем означает, что все ОК, программа работает) и на команду, которой nfdump был запущен (на приведенном скриншоте ее видно под строкой, начинающейся словом **CGroup)**, и ЧТО НАМ ВАЖНЕЙ ВСЕГО — она должна совпадать с той, которую вы вписали в файл, выполняя п. 4 инструкции выше.

ПРИМЕЧАНИЯ:

- Если параметры отличаются от требуемых, попробуйте выполнить проверку еще раз, но уже после полной перезагрузки сервера.
- Пусть вас не смущает использование то имени nfcapd, то имени nfdump применительно к одному и тому же программному продукту! Как уже было сказано выше — в (один и тот же) пакет NetFlow-коллектора nfdump входит сразу несколько программ.
- Значение параметров («ключей»), с которыми теперь будет запускаться служба nfcapd, будет рассмотрено далее — в разделе «*Настройка NetFlow-коллектора nfdump*».

Установка NetFlow-сенсора fprobe

Fprobe — это программа-«сенсор», которая собирает информацию о сетевом трафике хотспота и по протоколу NetFlow отсылает ее на «коллектор».

ВАЖНО: УСТАНОВКА ПРОГРАММЫ FPROBE НА СЕРВЕРЕ ТРЕБУЕТСЯ ТОЛЬКО В ТОМ СЛУЧАЕ, ЕСЛИ ВАШ СЕРВЕР EASYHOTSPOT ВЫСТУПАЕТ (В ТОМ ЧИСЛЕ) И В КАЧЕСТВЕ ШЛЮЗА ЛОКАЛЬНОГО ХОТСПОТА (Т.Е., НА СЕРВЕРЕ УСТАНОВЛЕНА ПРОГРАММА COOVA-CHILLI, И ТОЧКИ ДОСТУПА ПОДКЛЮЧАЮТСЯ НЕПОСРЕДСТВЕННО К ВЫХОДУ СЕРВЕРА)!

Если ваш сервер будет обслуживать только удаленные («внешние») роутеры, установка на сервере программы fprobe НЕ ТРЕБУЕТСЯ! Можете пропустить этот раздел и переходить к следующему. Но учтите — **вам обязательно нужно будет установить NetFlow-сенсоры на ваших роутерах**! Эти вопросы рассматриваются в разделе «*Установка и настройка NetFlow-сенсора на роутерах*».

Скрипт-инсталятор Easyhotspot, если вы устанавливаете биллинг именно с его помощью, и выберете вариант 1 или 2 (при которых на сервер устанавливается Coova-Chilli) просто задаст вам вопрос о том, желаете ли вы установить и программу fprobe:

```
Программа Easyhotspot может вести учет адресов, посещенных клиентами.
Так как в вашей системе используется Coova-Chilli, установленный на
самом сервере, то для работы этой функции необходима установка сенсора
NetFlow (программы fprobe). С другой стороны, если вам не нужен учет
посещенных клиентами ресурсов, эту программу устанавливать не реко-
мендуется (т.к. этот учет значительно повышает нагрузку на сервер, а
также радикально увеличивает число записей в базе данных биллинга).
На всякий случай, УТОЧНЯЮ - fprobe используется только для сбора
информации с ЛОКАЛЬНОГО хотспота (Coova-Chilli, установленного на
самом сервере Easyhotspot). Если же ваш биллинг обслуживает только
внешние роутеры, то установка программы fprobe на него не требуется!

А теперь ответьте на вопрос:
Выполнить установку NetFlow-сенсора fprobe на ваш сервер?
Да или нет - [Y/N]:
```

Рис. 4 — Предложение установить на сервер Easyhotspot программу fprobe

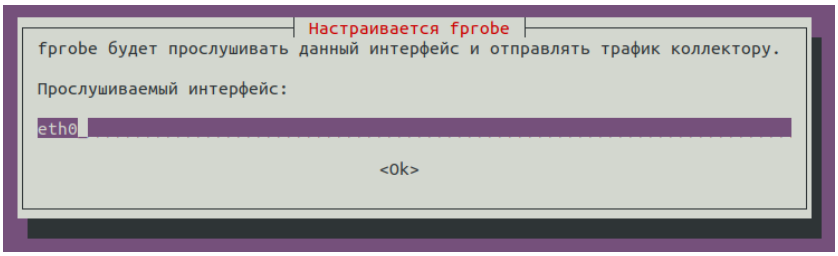
Чтобы установить ее, просто ответьте на данный вопрос утвердительно. Скрипт сам выполнит установку **и настройку** программы.

Если же вы устанавливаете сервер биллинга ***вручную***, то вам нужно будет выполнить следующие действия:

1. Войдите в консоль (терминал) и введите такую команду:

   ```
   sudo apt-get install fprobe
   ```

2. Во время установки пакета вам один за другим будут выведены два вопроса:



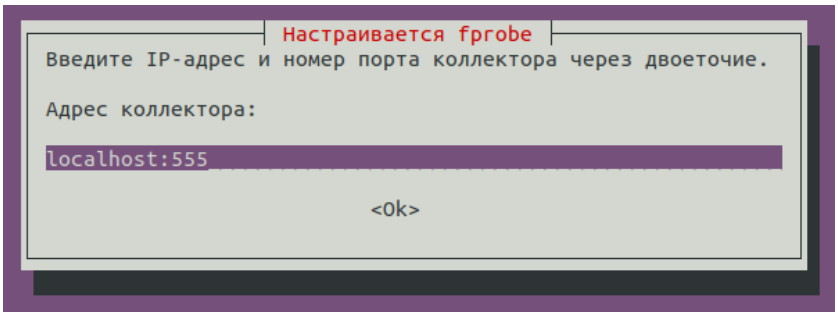


Рис. 5 — Вопросы, задаваемые во время установки пакета fprobe

3. В ответ на оба этих вопроса просто нажмите клавишу **Enter** на клавиатуре (параметры программы fprobe вы настроите позже, см. п. 4-7 далее).
4. После того, как установка пакета fprobe будет завершена, вам нужно настроить параметры программы. Для этого введите команду:

   ```
   sudo mcedit /etc/default/fprobe
   ```

5. В открывшемся файле найдите такие строки (*учтите, что они в файле не идут подряд одна за другой, в отличие от того, как это показано тут*):

   ```
   INTERFACE="eth0"
   FLOW_COLLECTOR="localhost:555"
   OTHER_ARGS="-fip"
   ```

6. Отредактируйте их, чтобы в итоге они стали выглядеть следующим образом*:

   ```
   NTERFACE="tun0"
   FLOW_COLLECTOR="localhost:2055"
   OTHER_ARGS="-fip -n7"
   ```

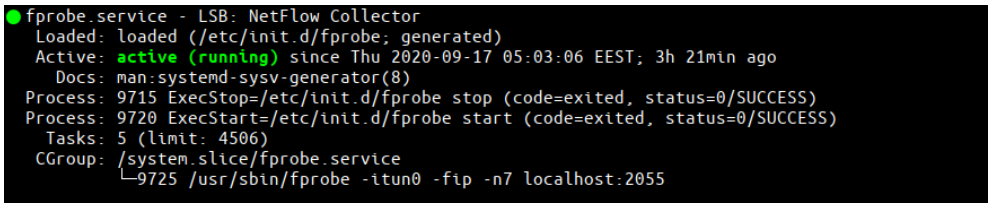
7. Сохраните изменения (**F2**) и выйдите из редактора (**F10**).
8. Чтобы внесенные вами изменения были задействованы в работе, перезапустите службу fprobe командой:

   ```
   sudo systemctl restart fprobe
   ```

На этом «ручная» установка и настройка NetFlow-сенсора fprobe завершена. Теперь Вы можете проверить, все ли было выполнено правильно, введя команду:

```
sudo systemctl status fprobe
```

В ответ вы должны получить подобное сообщение:



```
● fprobe.service - LSB: NetFlow Collector
   Loaded: loaded (/etc/init.d/fprobe; generated)
   Active: active (running) since Thu 2020-09-17 05:03:06 EEST; 3h 21min ago
     Docs: man:systemd-sysv-generator(8)
  Process: 9715 ExecStop=/etc/init.d/fprobe stop (code=exited, status=0/SUCCESS)
  Process: 9720 ExecStart=/etc/init.d/fprobe start (code=exited, status=0/SUCCESS)
    Tasks: 5 (limit: 4506)
   CGroup: /system.slice/fprobe.service
           └─9725 /usr/sbin/fprobe -itun0 -fip -n7 localhost:2055
```

Рис. 6 — Результат проверки статуса программы fprobe

В нем вас интересуют две вещи — статус (в приведенном примере он выделен зеленым цветом, сообщение «**active (running)**» означает, что все ОК, программа работает) и та команда, которой fprobe был запущен (ее видно на приведенном скриншоте под строкой, начинающейся словом **CGroup**, и ЧТО НАМ ВАЖНЕЙ ВСЕГО — в ней вы должны увидеть все те значения параметров, которые вы вписали в файл настроек, выполняя п. 6 инструкции выше).

ПРИМЕЧАНИЯ:

- Если параметры отличаются от требуемых, попробуйте выполнить проверку еще раз, но уже после полной перезагрузки сервера.
- Смысл параметров, с которыми запускается служба fprobe, будет рассмотрен далее — в разделе «*Настройки NetFlow-сенсора fprobe*».

Установка парсера

«Парсер» — это небольшой скрипт, написанный на языке perl, назначение которого — данные, собранные коллектором, внести в базу данных биллинга.

УСТАНОВКА ПАРСЕРА ДОЛЖНА ВЫПОЛНЯТЬСЯ ВАМИ В ЛЮБОМ СЛУЧАЕ, ЕСЛИ ВЫ ХОТИТЕ СОХРАНЯТЬ СТАТИСТИКУ NETFLOW, ВНЕ ЗАВИСИМОСТИ ОТ ТОГО, ОБСЛУЖИВАЕТ ВАШ БИЛЛИНГ ТОЛЬКО «ВНЕШНИЕ» РОУТЕРЫ, ИЛИ ЖЕ И ЛОКАЛЬНЫЙ ХОТСПОТ (УСТАНОВЛЕННУЮ НА СЕРВЕРЕ ПРОГРАММУ COOVA-CHOLLI) — ТОЖЕ!

В случае установки программы Easyhotspot ***скриптом-инсталятором***, если вы утвердительно ответите на вопрос об установке программы nfdump (см. рис. 2 в разделе «*Установка NetFlow-коллектора nfdump*»), парсер будет установлен автоматически. Вместе со скриптом парсера на сервер будет скопирован и файл с его настройками. Значения, вписанные в этот файл «по умолчанию», выбраны такими, что парсер сможет начать работать сразу же после установки. Но это лишь в том случае, если вы не меняли некоторых параметров ВСЕЙ СИСТЕМЫ биллинга, используемых в ней «по умолчанию». В любом случае, обязательно ознакомьтесь с разделом «*Настройки парсера*», и затем настройте его параметры в соответствии с ВАШЕЙ системой и ВАШИМИ пожеланиями.

Если же вы устанавливаете сервер биллинга ***вручную***, то вам нужно будет выполнить следующие действия:

1. Войдите в консоль (терминал) и введите такую команду:

   ```
   sudo cp /usr/src/easyhotspot/chillispot/fprobe/parse_nfcap.* /root/
   ```

   Фактически, команда должна скопировать в папку **/root** ДВА отдельных файла — **parse_nfcap.conf** (собственно, сам парсер) и **parse_nfcap.conf** (файл с его настройками).

2. Кроме того, файлу скрипта (парсеру) нужно присвоить т. н. «бит исполнения» (чтобы он при вызове не просто «считывался как текст», а именно выполнялся как программа). Для этого введите следующую команду:

   ```
   sudo chmod +x /root/parse_nfcap.pl
   ```

На этом «ручная» установка парсера завершена. Все, что было сказано выше про настройку парсера в случае установки биллинга скриптом в полной мере относится и к случаю его ручной установки в Easyhotspot. То есть, обязательно прочтите раздел «*Настройки парсера*», и затем настройте параметры парсера в соответствии с вашей системой и вашими пожеланиями.

Настройка программ

Каждая из программ, участвующих в процессе сбора NetFlow-статистики и сохранения ее в базу данных биллинга, имеет собственные настройки. Для успешной работы функции все они должны коррелировать между собой, а также с параметрами биллинга, установленного на вашем сервере.

Настройки NetFlow-коллектора nfdump

Как уже было сказано ранее, из всего набора программ, включенных в состав пакета nfdump, биллинг Easyhotspot использует только две — программу *nfcapd*, которая собственно и является «коллектором», и программу *nfdump*. Настройки «коллектора» *nfcapd* будут рассмотрены в этом разделе. А вот настройки программы *nfdump* в виде какого-либо отдельного файла на сервере отсутствуют, их динамически создает скрипт-парсер в момент запуска. Поэтому, настройки *nfdump* будут рассмотрены далее — в соответствующем разделе (см. «*Настройки парсера*»).

Полный список параметров, с которыми может быть запущен демон nfcapd, вы сможете прочесть по ссылке [4]. Ниже приведены лишь те из них, которые используются сервером Easyhotspot, а также еще парочка дополнительных.

Параметр	Описание и значение
-p	Номер порта, на котором демон слушает входящие подключения. В сервере Easyhotspot используется порт 2055. **НЕ МЕНЯЙТЕ** значение этого параметра, т. к. оно должно совпадать со значением, указанным в настройках «сенсора» (-ов).
-D	(заглавная буква «**D**») Режим «службы» (или «демона»). После запуска программа отключается от терминала, в котором она была запущена, и продолжает работать «в фоне» («в бэкграунде», как это делает любая «серверная служба»). **НЕ МЕНЯЙТЕ** значение этого параметра.
-l	(прописная буква «эль») Указывает папку, в которую nfcapd будет сохранять свои файлы со статистикой, полученной от «сенсоров». В Easyhotspot используется папка **/var/cache/nfdump**. **НЕ МЕНЯЙТЕ** значение этого параметра, т. к. оно должно совпадать с аналогичной настройкой скрипта-парсера.
-P	(заглавная буква «**P**») Имя т. н. pid-файла, по которому система определяет, что «демон» уже запущен. Оставьте значение, указанное по умолчанию.
-T	(заглавная буква «**T**») Указывает список т. н. «расширений» (или плагинов), которые используются при записи файлов со статистикой. В зависимости от выбранного списка меняется набор сведений, попадающих в файлы со статистикой (например, адреса/протоколы/порты/байты, и т. д.). В сервере Easyhotspot для этого параметра указано значение **all**, благодаря чему используются все возможные расширения, и в файлы со статистикой пишутся все сведения, поступившие от «сенсоров». **НЕ МЕНЯЙТЕ** значение этого параметра (а отбор того, что писать или не писать в базу, осуществит впоследствии уже парсер в соответствии со СВОИМИ настройками).
-x	Команда (скрипт) которую nfcapd вызывает на выполнение автоматически каждый раз после того, как сбросит на диск сервера очередной дамп статистики. У сервера Easyhotspot в этом параметре указан адрес скрипта-парсера: **/root/parse_nfcap.pl**. **НЕ МЕНЯЙТЕ** значение этого параметра!
-t	Определяет интервал в секундах, с которым nfcapd сбрасывает дампы с собранными сведениями на диск. По умолчанию (если параметр не указан явно), используется значение 300 (секунд!). Таким образом, по умолчанию nfcapd каждые 5 минут сбрасывает на диск новый файл со «свежеиспеченной» статистикой, и по окончании записи файла вызывает скрипт-парсер (указывается в параметре «**-x**»). После установки nfcapd скриптом-инсталятором равно как и после «ручной» установки программы, описанной в разделе «*Установка NetFlow-коллектора nfdump*», принудительно никакое значение для параметра нигде не указывается, и как следствие — оно равно «дефолтным» 300 секундам. В случае, если сенсоры ваших хотспотов сбрасывают «ну очень много» данных (трафик слишком большой), можете сократить этот период, чтобы не происходило переполнение памяти программы.
-B	(заглавная буква «**B**») Определяет (в байтах!) размер буфера программы nfcapd. Значение по умолчанию определяется параметрами Операционной Системы (ядра). При повышенном трафике (счет на Гигабайты) разработчики советуют устанавливать его значение максимально возможным (типовое значение — более 100k). После установки nfcapd скриптом-инсталятором равно как и после «ручной» установки программы, описанной в разделе «*Установка NetFlow-коллектора nfdump*», значение данного параметра не указывается, и как следствие — используется некое «дефолтное» его значение. В случае, если сенсоры ваших хотспотов сбрасывают слишком много данных (трафик слишком большой), вы можете попробовать изменить размер буфера, подобрать его приемлемое значение.

ПРИМЕЧАНИЕ:

- Два последних приведенных в таблице параметра (которые выделены серым фоном) при «дефолтной» установке и настройке сервера Easyhotspot НЕ НАСТРАИВАЮТСЯ (используют свои «дефолтные» значения), и в этом руководстве описаны лишь на случай, если вдруг вам потребуется оптимизация параметров системы ввиду повышенных нагрузок на нее!

Настройки NetFlow-сенсора fprobe

НАПОМИНАЮ — ЕСЛИ ВАШ СЕРВЕР EASYHOTSPOT НЕ ИСПОЛЬЗУЕТСЯ ВАМИ В КАЧЕСТВЕ ШЛЮЗА ЛОКАЛЬНОГО ХОТСПОТА, ЕСЛИ НА СЕРВЕРЕ НЕ УСТАНОВЛЕН COOVA-CHILLI, ЕСЛИ К ВЫХОДУ СЕРВЕРА НЕ ПОДКЛЮЧЕНЫ ТОЧКИ ДОСТУПА НАПРЯМУЮ, И Т.Д., И Т.П., ТО ПРОГРАММА FPROBE НА НЕМ НЕ УСТАНАВЛИВАЕТСЯ! Как следствие — выполнять настройку программы fprobe вам НЕ ТРЕБУЕТСЯ! Поэтому, можете смело пропускать этот раздел и переходить к следующему!

Полный список параметров, с которыми может быть запущена программа fprobe, вы сможете прочесть по ссылке [5]. Ниже приведены лишь те из них, которые используются сервером Easyhotspot.

Параметр	Описание и значение
`-i`	(В файле настроек* этот параметр записывается в «индивидуальную» строку `INTERFACE`) Указывает имя интерфейса, на котором сенсор должен будет собирать статистику. Программа позволяет не указывать этот параметр, но собирать всю статистику со всех интерфейсов сервера — лишено всякого смысла (imho). В сервере Easyhotspot используется интерфейс **tun0** (именно его и использует программа Coova-Chilli). **НЕ РЕКОМЕНДУЕТСЯ МЕНЯТЬ** значение этого параметра.
`адрес:порт`	(В файле настроек* данный параметр записывается в «индивидуальную» строку `FLOW_COLLECTOR`) В этом параметре указываются (разделенные двоеточием) соответственно адрес и порт месторасположения программы-«коллектора». Так как используемый биллингом Easyhotspot «коллектор» (программа nfcapd) размещается на том же самом компьютере, и слушает подключений на порту 2055, то в конфиге указано значение **localhost:2055**. **НЕ МЕНЯЙТЕ** значение этого параметра.
`-f`	(В файле настроек* этот параметр записывается в «общую» (с некоторыми другими параметрами) строку `OTHER_ARGS`). Параметр назначает фильтр (шаблон), с которым осуществляется отбор данных для статистики. Если параметр не указывается — сенсор собирает все подряд. Но так как fprobe использует весьма «примитивный» метод обнаружения пакетов, оставлять фильтр совсем пустым — плохая идея! В общем случае, рекомендуется использовать шаблон "`ip`" (чтобы в итоге параметр был указан как "`-fip`"). Дополнительные варианты для фильтров вы можете почитать либо на странице [5], либо в документации к **tcpdump**.
`-n`	(В файле настроек этот параметр записывается в «общую» (с другими) строку `OTHER_ARGS`) Указывает версию протокола NetFlow, используемую сенсором при отправке статистики на коллектор. Fprobe позволяет использовать версии 1,5 и 7. В сервере Easyhotspot (для локального сенсора fprobe) используется 7-я версия.

ПРИМЕЧАНИЕ:

- Речь идет о файле **/etc/default/fprobe**, редактирование которого было описано в разделе «Установка NetFlow-сенсора fprobe».

С прочими параметрами программы при необходимости вы сможете ознакомиться либо в документации, размещенной на сайте разработчиков [3], либо и на странице man-а к программе fprobe [5]. Может так статься, что эта информация понадобится вам в случае необходимости оптимизации параметров системы ввиду повышенных нагрузок на нее!

Настройки парсера

Настройки парсера определяются выполняемыми им функциями. Написанный на perl-е скрипт-парсер (файл **/root/parse_nfcap.pl**) выполняет такие действия:

- Считывает один за другим все файлы статистики (т. н. «дампы»), созданные коллектором nfcapd в указанной папке;
- Вызывает программу nfdump, которая «расшифровывает» сведения, записанные в файле дампа коллектором, параллельно фильтрует данные по ряду указанных критериев, и преобразует в формат, пригодный для дальнейшей работы с этой информацией;
- Подключается к базе данных Easyhotspot и записывает в нее «расшифрованные» данные статистики;

Все настройки парсера размещается в файле **/root/parse_nfcap.conf**, который после установки находится в одной папке с самим парсером. Чтобы изменить какие-нибудь (желаемые) настойки парсера, выполните следующее:

1. Войдите в консоль (терминал) и откройте файл настроек в редакторе с помощью вот такой команды:

   `sudo mcedit /root/parse_nfcap.conf`

2. Внесите измените в значения параметров на нужные вам (см. ниже раздел «*Параметры парсера*» с описаниями). При необходимости, вы можете как добавлять новые строки с нужными вам параметрами, так и удалять излишнее.
3. Сохраните изменения (**F2**) и выйдите из редактора (**F10**).

Вам ничего не нужно перезапускать после редактирования данного файла — новые параметры автоматически будут задействованы при следующих запусках скрипта парсера.

Параметры парсера

В данном разделе описаны те параметры, которые использует в своей работе скрипт-парсер, а также даны дополнительные разъяснения к ним (т. к., краткие описания смысла параметров присутствуют непосредственно в самом файле настроек). Также, с учетом того, что парсер для выполнения своих обязанностей в том числе вызывает еще и «внешнюю» программу *nfdump*, вам может пригодиться информация, размещенная на странице man-а указанной программы [**6**].

Совсем кратко о форматировании строк с параметрами. В строке указывается имя параметра, знак равно, и после знака равно — значение параметра. В качестве значения параметра скрипт парсера воспримет ВСЁ, ЧТО БУДЕТ НАПИСАНО СПРАВА ОТ ЗНАКА РАВНО! Поэтому, будьте внимательны, не пишите в строке с параметром все подряд (например, комментарии). Также, уточняю — имя параметра, знак равно и значение параметра ДОЛЖНЫ БЫТЬ НАПИСАНЫ В ОДНОЙ СТРОКЕ, если вы что-то перенесете на следующую строку, то это будет проигнорировано! С другой стороны, все строки из файла настроек, которые начинаются с символа «**#**» скрипт автоматически считает комментариями, и даже и не пытается анализировать, что там в них написано.

А теперь перейдем к самим параметрам:

`src = /var/cache/nfdump`

Параметр указывает имя папки, из которой парсер считывает файлы со статистикой (ту самую, в которую сохраняет свои данные программа-коллектор — nfcapd). Именно по этой причине, значение, указанное для параметра **`src`** тут, должно совпадать со значением параметра «**`-l`**», указанного в настройках коллектора nfcapd (см. раздел «*Настройки NetFlow-коллектора nfdump*»). При запуске парсера скрипт поочередно считывает один за другим файлы со статистикой из указанной папки, запускает их расшифровку (внешней) программой nfdump и полученные данные уже в «человеческом виде» вносит в базу

данных программы Easyhotspot. Важное уточнение — после успешной расшифровки данных из каждого отдельно взятого файла статистики скрипт тут же удаляет данный (исходный) файл из указанной папки. Такой механизм работы с файлами был выбран с учетом двух обстоятельств — во первых, так парсер гарантировано прочитает ВСЕ ФАЙЛЫ, КОТОРЫЕ НАЙДЕТ в папке с дампами, благодаря чему в базу в конечном итоге ПОПАДЕТ ВСЯ СТАТИСТИКА (даже если при каком-то из предыдущих вызовов скрипта произошел сбой и файл «прошлой» статистики остался не перенесенным в базу*[1]). А с другой стороны, «старые» декодированные файлы со статистикой тут же удаляются после их декодирования, что избавляет от необходимости использования для этого каких-либо дополнительных «механизмов» (например, «ротации лог-файлов»). Ну и последнее уточнение по работе с файлами статистики — на время обработки одного отдельно взятого файла парсер вносит его имя в специальный «файл блокировки»*[2], чтобы другой запущенный экземпляр парсера*[3] не внес в базу дубль той же самой информации.

```
host = localhost
base = easyhotspot
user = easyhotspot
pass = easyhotspot
```

Это параметры, с которыми скрипт парсера подключается к MySQL-базе данных программы Easyhotspot, чтобы перенести в нее собранную статистику. «**host**» — адрес компьютера, на котором расположен MySQL, «**base**» — имя базы данных, «**user**» — имя пользователя, который подключается к базе, и «**pass**» — пароль, с которым он подключается. Вы должны указать тут те параметры, которые использует для подключения к базе данных сама программа Easyhotspot (их вы можете прочесть, например, в файле настроек биллинга: **/var/www/easyhotspot/application/config/database.php**).

```
resolve = 1
```

Параметр определяет — должен ли скрипт парсера выполнять преобразование IP-адресов в имена доменов (отсылать DNS-запросы, и в базу записывать уже полученные на них ответы). Если значение параметра установлено как «`1`» (по умолчанию), то скрипт выполняет DNS-преобразование. Если же установить параметр равным «`0`», то скрипт не будет отсылать DNS-запросы, и в базу программы будут записываться «чистые» IP-адреса. Возможность отключения DNS-резольвинга предусмотрена по той причине, что процедура определения имен доменов по их IP-адресам замедляет работу парсера (пока для каждого адреса, попавшего в статистику, выполнишь DNS-запрос, пока получишь ответ — а время-то идет!). И может так оказаться, что на каких-нибудь хотспотах с очень большим числом клиентов, собираемая статистика в итоге будет иметь такие «нереальные» объемы, что скрипт просто не будет успевать с ней справляться! Вот именно для подобных случаев и предусмотрена возможность отключить резольвинг. Хотя, с другой стороны, у парсера есть встроенный «механизм», чтобы сократить затраты времени на выполнение этой задачи. С этой целью используется внутренний кеш*[4] скрипта: когда парсеру нужно преобразовать IP-адрес в имя домена, он сначала сверяется со своим кешем, и если там уже имеется ранее полученная запись — ответ берется из кеша, а запрос на DNS-сервер не отсылается.

```
direction = 1
```

Этот параметр указывает скрипту, нужно ли ему вносить в базу статистику о трафике «только в одну сторону» (от клиента в интернет), или же «в обе стороны» (как от клиента в интернет, так и из интернета к клиенту). По умолчанию в настройках указано значение «`1`», что означает — ТОЛЬКО ОТ КЛИЕНТА В ИНТЕРНЕТ! Чтобы в базу заносились сведения и об «обратном» трафике (который поступает уже из интернета к клиентам), у параметра нужно указать иное значение — «`2`». Будьте внимательны, не допускайте ошибок, параметр может иметь лишь два значения: **1** или **2** (арабские цифры), иные могут приводить к сбоям в работе скрипта! Учтите, что сбор сведений об «обратном» трафике, с одной стороны, замедляет работу скрипта (т. к., один и тот же дамп анализируется дважды, сначала

отбираются записи «туда», а потом — «обратно»). А с другой стороны, степень «информативности» обратных записей зачастую вызывает очень большие сомнения, т. к. один и тот же сайт только собственным содержимым может отвечать с целой пачки различных адресов (например, в целях балансировки нагрузки), а уж всевозможные там рекламы/банеры/счетчики/информеры и прочая подобная ерунда — так и вообще будут «валом валить» со всего света! Нужно ли вам весь этотм мусор логировать, или же «база не резиновая» — решайте сами!

debug = 0

Параметр указывает уровень логирования скрипта-парсера. По умолчанию указано значение «**0**», что означает — логирование выключено. Повышение этого уровня включает ведение парсером своего лога. Файл лога парсера (если он включен) размещается в папке **/tmp** и называется **netflow_parse.log**. Кроме **0** допускаются еще такие уровни: **1**, **2**, **3** и **4** (в качестве значения параметра разрешается писать ТОЛЬКО ЦЕЛЫЕ ЧИСЛА АРАБСКИМИ ЦИФРАМИ!). Каждый последующий уровень логирования (по мере возрастания числа) будет включать в себя и сведения, определенные для всех «более низких» уровней. Теперь пройдемся по самим уровням. «**1**» — в лог будут внесены сведения о времени и дате запуска скрипта, имени каждого обрабатываемого файла (дампа статистики), общем числе обработанных строк статистики (попавших в базу биллинга), числе записей DNS-кеша, само содержимое DNS-кеша, и продолжительность выполнения скрипта в секундах. «**2**» — в лог будет добавлена (целиком, уже сформированная) команда, которой парсер вызывал программу nfdump для расшифровки дампа (эта информация окажется полезной, если вы захотите увидеть, какие фильтры и в каком виде попали в команду дешифровки). «**3**» — в лог будут добавлены все результаты декодирования дампа (все строки, удовлетворившие критериям отбора, со всеми сведениями об этом трафике — адрес источника, адрес назначения, адрес NAS, маки, байты, и т. д., и т.п.). И последний допустимый вариант значения параметра — это «**4**». На нем мы задержимся подольше. Когда выбран такой уровень логирования, скрипт НЕ ДЕЛАЕТ ДВЕ ВЕЩИ — во первых, он не удаляет уже проанализированные файлы дампов «старой» статистики! А во вторых, чтобы повторно не вносить в базу данные из этих «старых» дампов, скрипт не удаляет имена этих, уже проанализированных файлов из ранее упоминавшегося файла блокировки*[2]! Собственно, данный уровень дебага (4-й) как раз и был введен в скрипт для сохранения «старых» дампов статистики (для возможности последующих «ручных» экспериментов над ними). Поэтому, использовать его на «боевых» серверах однозначно не рекомендуется! Но если уж вы его включите, то после того, как вернетесь назад, на любой более низкий уровень, обязательно удалите «файл блокировки» (чтобы парсер смог повторно проанализировать всю скопившуюся старую статистику и поудалять все старые дампы с компьютера).

network = 192.168.182.0/24,192.168.182.1

Параметры сети, для которой осуществляется сбор статистики. Важное уточнение по формату — фактически в строке указаны ДВА ПАРАМЕТРА, разделенных запятой, но ОТНОСЯЩИХСЯ К ОДНОЙ СЕТИ хотспота. В строке параметра ДО ЗАПЯТОЙ вы указываете сеть вашего хотспота (в формате «адрес сети/маска»), а ПОСЛЕ ЗАПЯТОЙ — адрес шлюза. Фактически — в строке параметра вы должны прописать настройки DHCP-сервера вашего хотспота: его шлюз и его диапазон адресов, раздаваемых клиентам! Параметр служит для того, чтобы в базу билинга не вносились лишние данные (так как, зачастую NetFlow-сенсоры собирают любую статистику, проходящую через все сетевые интерфейсы, включая и ту, сохранять которую нет ни малейшего смысла!). В итоге параметр попадает в команду для дешифрации статистики как указание отбирать «***только те пакеты, отправитель которых относится к указанной сети, но не является ее шлюзом***»*[5]. Если вам нужно собирать статистику с разных хотспотов, DHCP-серверы которых используют отличающиеся диапазоны адресов, выдаваемых клиентам и разные адреса шлюзов — не огорчайтесь! В файле настроек вы сможете столько раз указать данный параметр (каждый — отдельной строкой), сколько вам потребуется! Для наглядности, в файле настроек, который

поставляется с парсером по умолчанию, сразу указаны два «стандартных»*[6] значения параметра **network** — для «дефолтных» настроек хотспота, использующего Coova-Chilli, и для «дефолтных» настроек хотспота роутеров Mikrotik. Если вам нужен лишь один из этих диапазонов, или же какой-то абсолютно иной — отредактируйте файл и в начале строки с лишним просто поставьте знак «**#**» (либо полностью удалите ненужную строку).

`filter = proto tcp`

Параметр, указывающий какой-либо желаемый критерий для отбора статистики, которая должна попасть в базу программы. Сами фильтры — настолько «толстый лонгрид», что заслужили отдельный раздел в данной инструкции, который идет сразу же за этим (см. «*Фильтры парсера*»)! Здесь же я лишь опишу те принципы, по которым скрипт «собирает воедино» итоговую команду для расшифровки дампов. Во первых, строк с параметром «**filter**» вы можете указать в конфиге ровно столько, сколько вам их будет нужно (одна — хорошо, пять — тоже без проблем!). И первое, что вам нужно запомнить про «много строк с фильтрами», это то, что в итоговую команду все указанные вами фильтры скрипт «склеит» с оператором «**`and`**», означающим «***и***» (т. е., для в базу будут отбираться лишь те сведения, которые ОДНОВРЕМЕННО БУДУТ УДОВЛЕТВОРЯТЬ «**`условию 1 и условию 2 и условию 3`** и т. д.…»). Поэтому, составляя фильтры, в первую очередь думайте! Для примера представим, что вы написали два ОТДЕЛЬНЫХ условия: в первом указали отбирать запросы, идущие на 80-й порт, а во втором — на порт 443. Что получится в конечном итоге? А НИЧЕГО! В базу не попадет ни единой строки! А почему? Да потому, что команда фильтрации будет искать в дампах такие записи, в которых ЗАПРОСЫ ИДУТ И НА ПОРТ 80 И НА ПОРТ 443 ОДНОВРЕМЕННО (чего, естественно, быть не может!). Поэтому, если вам нужно выполнение каких-то условий в варианте «*или-или*», то такие условия вам нужно будет писать в одной строке параметра «**filter**». Причем, строку с таким фильтром вам нужно будет еще и брать в круглые скобки, т. к. к остальным критериям она все равно будут добавлена с оператором «**`and`**»! Для пущей наглядности разберем два фильтра, которые по умолчанию присутствуют в файле настроек парсера. В одной строке файла указано следующее: «**`filter = proto tcp`**». А в другой — «**`filter = (port 443 or port 80)`**». Итоговый отбор будет осуществляться по условию «**`proto tcp and (port 443 or port 80)`**», то есть — в базу попадут все запросы, которые по протоколу TCP направлялись на порт 80 или на порт 443. Надеюсь, понятно объяснил. НА КРАЙНИЙ СЛУЧАЙ — все свои пожелания по фильтрации пакетов вы можете ЗАПИСАТЬ В ОДНУ ЕДИНСТВЕННУЮ СТРОКУ параметра «**filter**» (если вам так будет проще).

ПРИМЕЧАНИЯ:

1. В случае нормальной работы скрипта порядок действий такой: парсер открыл дамп, выставил блокировку (см. следующий абзац), запустил анализ, записал в базу результат, удалил дамп, снял блокировку. Если же произошла ошибка, то именно благодаря месту размещения этого файла блокировки (если точно → **/tmp/netflow_parse.lock**), он будет автоматически удален после перезагрузки сервера. В результате парсер сможет прочесть и проанализировать старый дамп статистики и внести ее в базу.

2. Технически, файл блокировки представляет собой обычный текстовый файл, в который построчно заносятся имена файлов с дампами статистики, обрабатываемыми (всеми) парсерами, запущенными в данный момент. При запуске парсер проверяет, не внесено ли имя файла-дампа, который он собрался анализировать, в список файла блокировки. Если внесено, то скрипт переходит к следующему файлу статистики. После окончания обработки файла со статистикой его имя удаляется парсером из файла блокировки. Основное назначение данной блокировки — описано в следующем абзаце.

3. По умолчанию NetFlow-сенсор пять минут копит все поступающую на него статистику в своей памяти. Раз в пять минут он скидывает собранные данные в файл дампа и тут же запускает на выполнение скрипт-парсер. Продолжительность работы скрипта в первую очередь зависит от объема данных, попавших в этот дамп (т. к., парсер должен их считать,

прогнать через фильтры, распознать имя домена для каждого уникального IP-адреса из списка, и в конце концов — все это скинуть в базу данных!). Поэтому, вполне может так статься (если объем данных в дампе будет огромным), что обработка файла будет длиться дольше 5 минут. А тут NetFlow-сенсор, отсчитав свои очередные пять минут, снова скинет на диск новый файл дампа со статистикой, и запустит новый экземпляр (процесс) скрипта-парсера! Так вот именно для того, чтобы «второй» запущенный парсер не начал тоже анализировать «старый» дамп, над которым уже трудится «парсер №1», и используется файл блокировки. Парсер пишет в него имя того файла, который он анализирует в данный момент времени. А все другие парсеры сверяют имя файла, который они считали из папки со списком из файла блокировки...

4. Внутренний кеш парсера — это виртуальная память (часть ОЗУ сервера), выделенная ему Операционной Системой НА ВРЕМЯ ЕГО ВЫПОЛНЕНИЯ! То есть, кеш распознанных имен доменов у каждого запущенного экземпляра парсера — свой собственный! И хранится он в памяти только в течение того промежутка времени, пока скрипт запущен (выполняется). Как только скрипт завершил работу, эта память очищается. Каждый новый запущенный скрипт парсера заново создает свой кеш и заново наполняет его данными. Парсеры (если вдруг окажется, что в один и тот же момент их будет запущено несколько экземпляров) не могут обмениваться между собой информацией из данного кеша имен доменов.

5. Если в настройках парсера параметр **direction** установлен равным «**2**» (собирать статистику также и о данных, поступающих клиентам из интернета), то команда дешифрации дампов выполняется дважды для каждого файла. Первый проход отбирает данные по принципу «***только те пакеты, отправитель которых относится к указанной сети, но не является ее шлюзом***» (как и было сказано в описании параметра **network**). А вот во время второго прохода критерий отбора меняет свое «направление» на обратное: «***только те пакеты, получатель которых относится к указанной сети, но не является ее шлюзом***».

6. Параметры локальных сетей хотспотов вы можете посмотреть в их настройках. Для программы *Chillispot* в файле настроек **/etc/chilli.conf** вам нужно найти параметр **net**. Для контроллера *Coova-Chilli* — в файле **/etc/chilli/defaults** смотрите значение сразу двух параметров: **HS_NETWORK** и **HS_NETMASK**. В роутерах Mikrotik — открывайте меню **IP** → **DHCP Server** → **Networks**. Ну и последнее уточнение: естественно, файлы настроек нужно искать там, где установлен (и работает) контроллер вашего хотспота (captive portal). Т.е., если хотспот запущен где-нибудь в роутере, то и конфиг его нужно искать ТАМ, а не на сервере биллинга! А то бывают случаи, пишут мне потом вопросы…

Фильтры парсера

Фильтрация данных во время дешифровки дампов статистики В ПЕРВУЮ ОЧЕРЕДЬ СЛУЖИТ ЦЕЛИ СОКРАТИТЬ ОБЪЕМ ВСЕГО ТОГО «ШРОТА», КОТОРЫЙ В КОНЕЧНОМ ИТОГЕ ПОПАДЕТ В БАЗУ БИЛЛИНГА! Например, указав в фильтре условие отбирать запросы, выполненные только по протоколу TCP, вы избавите базу от абсолютно бесполезных записей о всевозможных попытках PING-а (равно как и обо всех иных запросах, выполненных с использованием других протоколов, отличающихся от TCP). В принципе, начать работу вы можете с фильтрами, вписанными в конфиг парсера по умолчанию, и уже потом по итогам увиденного решать для себя — оставить все как есть, или же настраивать их под свои собственные пожелания. В любом случае, внимательно ознакомьтесь с приведенной ниже информацией, чтобы иметь представление о том, какие вообще возможные варианты фильтрации данных вам предоставляет программа nfdump.

Фильтр — это некое выражение, указывающее программе УСЛОВИЕ ДЛЯ ОТБОРА из всего вороха статистики именно тех записей, которые должны быть перенесены в базу биллинга. В итоговую команду может быть объединено любое число нужных вам таких «условий», программа nfdump (которая собственно и будет заниматься выборкой) не ограничивает ваш «полет фантазии» в этом вопросе. Сама программа nfdump в команде ее запуска (если, допустим, вы выполняете ее в

консоли), позволяет «склеивать» различные условия между собой как операторами **and** (означает И), так и **or** (означает ИЛИ). НО, напоминаю вам, что в биллинге Easyhotspot команду вызова программы nfdump формирует скрипт парсера, который различные условия, написанные в отдельных строках своего файла настроек, «склеивает» самостоятельно, и использует при этом ТОЛЬКО оператор **and**! Так что, если вам будут нужны какие-либо условия с оператором **or**, то подобные условия вам нужно будет записывать В ОДНУ СТРОКУ ФИЛЬТРА (в файле настроек)! Более того — подобные условия вы должны будете заключать в круглые скобки, например, вот так: `(port 443 or port 80)`.

Дополнительные операторы, которые можно использовать в фильтрах

В строках фильтра вы можете использовать оператор **not**, означающий «НЕ», то есть, действие обратное условию указанному в фильтре. Для примера, запись «`not proto icmp`» указывает — отбирать сведения о тех соединениях, которые были выполнены НЕ ПО ПРОТОКОЛУ ICMP.

В ряде фильтров (будет указано дополнительно) можно использовать специальные операторы, указывающие «направление данных в потоке»: **src** (от слова *source*, означающего «источник» (откуда)) или **dst** (от слова *destination*, означающего «назначение» (куда)).

Некоторые фильтры служат для указания условий, сравнивающих разные параметры с какими-либо (числовыми) значениями. В качестве операторов сравнения допускается использовать такие: **=**, **==**, **>**, **<**, **EQ**, **LT**, **GT**, которые соответственно означают: «***равно***», «***равно***», «***больше***», «***меньше***», «***равно***», «***меньше***», «***больше***». Например, запись в фильтре: «`port > 1024`» указывает, что нужно отбирать лишь ту статистику, в которой номер порта больше 1024.

В фильтрах, использующих какие-либо числовые значения, разрешено использовать приставки-множители при написании чисел: **k**, **m** и **g**, означающие соответственно «**кило-**», «**мега-**» и «**гига-**». Кратность множителей — 1000. Пример записи с множителем в фильтре: «`bps > 10k`».

Собственно сами фильтры

И последняя ремарка перед тем, как приступить к описанию фильтров. В данном руководстве описаны лишь те фильтры, суть которых мне была понятна. Программа nfdump предлагает гораздо больше вариантов параметров, чем описано в этой инструкции. Поэтому, чтобы ознакомиться с полным перечнем фильтров программы nfdump, прочтите документацию к ней [6].

`inet`

Версия протокола TCP-IP. Для 4-й версии (IPv4) допускается указывать параметр как **inet** или **ipv4**, для 6-й (IPv6) — **inet6** или **ipv6**. Пример записи фильтра:

`inet6`

`proto`

Протокол соединения. В качестве значения параметра может быть указан как любой известный протокол из числа **tcp**, **udp**, **icmp**, **icmp6**, **gre**, **esp**, **ah**, пр., так и валидный номер протокола, например, **6**, **17** и т. п. Пример записи фильтра:

`proto udp`

`ip` (как альтернативный вариант — можно использовать для параметра имя `host`)*[1]

Адрес или хост. В качестве значения может быть указан как просто IP-адрес, так и «полное имя домена». Если вы укажете имя домена, то оно будет преобразовано программой в IP-адрес с помощью DNS-запроса. В параметре при необходимости вы можете использовать операторы, указывающие на «направление»: **src** или **dst**, а также оператор отрицания — **not**. Пример записи фильтра (с дополнительными операторами):

`not dst host 192.168.1.2`

net*[2]

Адрес сети и маска сети. Причем, маска может быть указана вами как в десятичном виде (например, «**255.255.255.0**»), так и в CIDR-формате (например, «**/24**»). В параметре при необходимости вы можете использовать операторы, указывающие на «направление»: **src** или **dst**. Пример записи фильтра:

```
net 192.168.88.0/24
```

port

Номер порта. Полная запись фильтра включает в себя: слово «**port**», оператор сравнения (см. описание выше) и число, обозначающее номер порта. Если оператор сравнения в фильтре не указан, автоматически принимается, что был использован **=**. Также в параметре при желании вы можете использовать операторы, указывающие на «направление»: **src** или **dst**. Кроме того, возможна запись вида **port in [список портов]**, где «список портов» — это разделенный пробелами список отдельных номеров портов. Пример записи фильтра:

```
src port > 1024
```

if

Номер интерфейса. Параметр указывает ID интерфейса, точнее — его номер, известный как «SNMP номер интерфейса». В фильтре допускается использовать дополнительные операторы: **in** и **out**, означающие соответственно «входной» и «выходной». Пример записи фильтра:

```
in if 3
```

flags

Флаги протокола TCP-IP. В параметре указывается набор флагов, из числа: **A** (ACK), **S** (SYN), **F** (FIN), **R** (Reset (сбросить буфер)), **P** (Push («вытолкнуть» данные из буфера в приложение)), **U** (Urgent (важные данные)), **X** (включены все флаги). Порядок написания флагов в фильтре не критичен. Статус флагов, не указанных в фильтре, СЧИТАЕТСЯ НЕ ВАЖНЫМ! Поэтому, чтобы, например, отобрать статистику только с флагом SYN, фильтр должен быть записан следующим образом (заодно, сойдет и как пример «составного фильтра, записанного в одну строку»):

```
(flags S and not flags AFRPU)
```

router ip

IP-адрес роутера, с которого была получена статистика. Пример записи фильтра:

```
router ip 123.123.123.123
```

mac

MAC-адрес (указывает любой валидный MAC-адрес, попавший в статистику). В фильтре допускается использование дополнительных комбинаций операторов, которые указывают «интерфейс» и «направление» (как это описано CISCO в спецификации NetFlow v9), а именно: **in src**, **in dst**, **out src**, **out dst**. Пример записи фильтра:

```
in src mac 9C:B7:0D:A2:47:8C
```

packets

Фильтрация записей по числу пакетов (обычно, один пакет — это от 1000 до 1500 байт, зависит от конкретной реализации сети). Полная запись фильтра включает в себя: слово «**packets**», оператор сравнения (см. описание выше), число (пакетов), и также может быть задействован «множитель» (также, см. выше). Если оператор сравнения в фильтре не указан, автоматически принимается, что был использован **=**. Пример записи фильтра:

```
packets > 1k
```

bytes

Фильтрация записей по количеству байт. Полная запись фильтра включает в себя: слово «**bytes**», оператор сравнения (см. описание выше), число (байт), и также может быть указан «множитель» числа (был описан ранее). Если оператор сравнения в фильтре не указан, автоматически принимается, что был использован знак **=**. Пример записи фильтра (данный пример предназначен для отфильтровки всех пустых IPv4 пакетов):

```
bytes 46
```

tos

Тип обслуживания (Type of Service, ToS). В параметре указывается число от 0 до 255, соответствующее выбранному типу обслуживания. В параметре должно быть указано «направление» с помощью оператора **src** или **dst**. Для совместимости с nfump версии 1.5.x, отсутствие оператора «направления» приравнивается к оператору **src** (т. е., запись вида **tos <число>** соответствует записи **src tos <число>**).

pps (вычисляемое значение)

Фильтрация записей по числу пакетов, переданных в секунду (т. е., по скорости обмена). Полная запись фильтра включает в себя: слово «**pps**», оператор сравнения (см. описание выше), число (пакетов в секунду). Также, вы можете использовать «множитель» числа (см. ранее). Если оператор сравнения в фильтре не указан явно, то автоматически принимается, что был использован знак **=**. Пример записи фильтра:

```
pps > 1k
```

duration

Фильтрация потоков по их длительности в миллисекундах. В фильтре должно быть указаны оператор сравнения и число. Если оператор сравнения в фильтре не указан явно, то автоматически принимается, что был использован знак **=**. Также, допускается использовать «множитель» для числа.

bps (вычисляемое значение)

Фильтрация записей по числу байт, переданных в секунду (фактически — по скорости обмена). Полная запись фильтра включает в себя: слово «**bps**», оператор сравнения (см. описание выше), число (байт в секунду). Также, вы можете использовать «множитель» числа (см. ранее). Если оператор сравнения в фильтре не указан явно, то автоматически принимается, что был использован знак **=**. Пример записи фильтра:

```
bps > 1k
```

bpp (вычисляемое значение)

Фильтрация потоков по числу байт в одном пакете. Полная запись фильтра включает в себя: слово «**bpp**», оператор сравнения (см. описание выше), число (байт в пакете). Также, вы можете использовать «множитель» числа (см. ранее). Если оператор сравнения в фильтре не указан явно, то автоматически принимается, что был использован знак **=**.

ПРИМЕЧАНИЯ:

1. Вам не нужно указывать в фильтрах параметр **ip** (или **host**) — его добавит в команду сам скрипт-парсер! Адрес шлюза сети хотспота будет взят им из параметра **network** (см. раздел «*Параметры парсера*»).
2. Вам не нужно указывать в фильтрах параметр **net** — его добавит в команду сам скрипт-парсер! Адрес сети хотспота (с маской) будет взят им из параметра **network** (см. раздел «*Параметры парсера*»).
3. С прочими параметрами фильтров (не попавшими в данное Руководство) вы можете ознакомиться в документацию к программе nfdump [6].

Настройка файервола на сервере биллинга

Сначала — как обычно, немного разъяснений. Дело в том, что описываемая ниже процедура вам может и не понадобиться в зависимости от того, как ваш сервер используется и как он настроен. Необходимость проведения дополнительной настройки файервола вашего сервера зависит от двух критериев:

1. Первый из них определяется тем, в каком режиме вы используете ваш сервер, а именно — обслуживает ли он «внешние» (размещенные удаленно от него) роутеры с хотспотами, установленными непосредственно в самих роутерах. Если нет (т. е., ваш сервер служит только шлюзом локального хотспота, для чего на нем установлена программа Coova-Chilli), то можете смело переходить к следующему разделу — вам ничего дополнительного делать с вашим файерволом не нужно! NetFlow-сенсор будет обращаться к NetFlow-коллектору «внутри» самого сервера (через т. н. localhost), следовательно, нет нужды в открытии каких-либо внешних портов в файерволе)!

2. Второй критерий — текущая настройка файервола сервера. В ряде случаев (например, при установке биллинга на VPS/VDS) «заградительные» функции файервола могут быть не активированы (иными словами, он и без проведения дополнительной настройки будет пропускать трафик от NetFlow-сенсоров «внешних» роутеров к NetFlow-коллектору).

Поэтому, если вы обслуживаете «внешние» роутеры, первым делом проверьте текущий статус вашего файервола. Чтобы проверить, как настроен ваш файервол, войдите в терминал (консоль) и введите такую команду:

```
sudo iptables -nvL
```

Теперь рассмотрим два возможных варианта ответа на нее.

Первый вариант ответа будет выглядеть таким образом:

```
Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

Рис. 7 — Результат проверки статуса файервола (когда «все открыто»)

В ответе мы видим информацию о трех «цепях» файервола (цепочках прохождения пакетов, по-английски цепь — это «*Chain*») → **Input** (вход), **Output** (выход) и **Forward** (переадресация). И сейчас нам из этих цепей (применительно к текущей задаче, а именно — получению данных NetFlow-коллектором от удаленных NetFlow-сенсоров) интересна лишь цепь **Input** (данные, поступающие извне вовнутрь («на вход») сервера). Мы видим, что в ней ОТСУТСТВУЮТ какие-либо правила (открывающие нужный нам 2055-й порт по протоколу UDP). И казалось бы — пакеты не пройдут, надо срочно «открывать порт»! Но перед тем как «бежать-спотыкаться», внимательно посмотрите на т. н. «политику» цепи **Input** (параметр «**policy**»). И как видно на приведенном скриншоте, там указано действие «**ACCEPT**» (т. е. — «принимать»)! А это значит, что в любом случае все пакеты, поступившие на вход сервера, попадут внутрь! То есть, прямым текстом — ЕСЛИ ВЫ ВИДИТЕ ИМЕННО ТАКОЙ ОТВЕТ НА КОМАНДУ КАК ПОКАЗАН НА РИС. 7, ТО ВАМ НЕ НУЖНО НИЧЕГО ДОПОЛНИТЕЛЬНО НАСТРАИВАТЬ В ВАШЕМ ФАЙЕРВОЛЕ! ДАННЫЕ ОТ NETFLOW-СЕНСОРОВ ВАШИХ «ВНЕШНИХ» РОУТЕРОВ БУДУТ И ТАК БЕСПРЕПЯТСТВЕННО ПРОХОДИТЬ К NETFLOW-КОЛЛЕКТОРУ!

Второй вариант ответа на вышеуказанную команду будет выглядеть иначе — аналогично тому, как показано ниже. Почему «аналогично»? Набор правил, управляющих файерволом сервера, может быть самым различным, и поэтому, ответ на команду тоже может отличаться — как числом строк,

так и их содержимым. Но вам сейчас важно понять те ключевые критерии, по которым вы сможете определить, что файервол вашего сервера нуждается в донастройке.

```
Chain INPUT (policy DROP 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
 294K   90M ACCEPT     all  --  *      *       0.0.0.0/0            0.0.0.0/0            state RELATED,ESTABLISHED
   35  2100 ACCEPT     tcp  --  enp1s0 *       0.0.0.0/0            0.0.0.0/0            tcp dpt:80 flags:0x17/0x02
    0     0 ACCEPT     tcp  --  enp1s0 *       0.0.0.0/0            0.0.0.0/0            tcp dpt:443 flags:0x17/0x02
    0     0 ACCEPT     tcp  --  enp1s0 *       0.0.0.0/0            0.0.0.0/0            tcp dpt:8081 flags:0x17/0x02
    0     0 ACCEPT     tcp  --  enp1s0 *       0.0.0.0/0            0.0.0.0/0            tcp dpt:8080 flags:0x17/0x02
    1    60 ACCEPT     tcp  --  enp1s0 *       0.0.0.0/0            0.0.0.0/0            tcp dpt:22 flags:0x17/0x02
  500 30000 ACCEPT     tcp  --  enp1s0 *       0.0.0.0/0            0.0.0.0/0            tcp dpt:10000 flags:0x17/0x02
   39  2340 ACCEPT     tcp  --  enp1s0 *       0.0.0.0/0            0.0.0.0/0            tcp dpt:20000 flags:0x17/0x02
  430 35316 ACCEPT     udp  --  enp1s0 *       0.0.0.0/0            0.0.0.0/0            udp dpt:137
  230 56738 ACCEPT     udp  --  enp1s0 *       0.0.0.0/0            0.0.0.0/0            udp dpt:138
    1    60 ACCEPT     tcp  --  enp1s0 *       0.0.0.0/0            0.0.0.0/0            tcp dpt:139 flags:0x17/0x02
    6   360 ACCEPT     tcp  --  enp1s0 *       0.0.0.0/0            0.0.0.0/0            tcp dpt:445 flags:0x17/0x02
    0     0 ACCEPT     tcp  --  enp1s0 *       0.0.0.0/0            0.0.0.0/0            tcp dpt:110 flags:0x17/0x02
    0     0 ACCEPT     tcp  --  enp1s0 *       0.0.0.0/0            0.0.0.0/0            tcp dpt:995 flags:0x17/0x02
    0     0 ACCEPT     tcp  --  enp1s0 *       0.0.0.0/0            0.0.0.0/0            tcp dpt:631 flags:0x17/0x02
   51  4944 ACCEPT     udp  --  enp1s0 *       0.0.0.0/0            0.0.0.0/0            udp dpt:514
    0     0 ACCEPT     tcp  --  enp1s0 *       0.0.0.0/0            0.0.0.0/0            tcp dpt:514 flags:0x17/0x02
    0     0 ACCEPT     udp  --  enp1s0 *       0.0.0.0/0            0.0.0.0/0            udp dpt:2055
    0     0 ACCEPT     udp  --  enp1s0 *       0.0.0.0/0            0.0.0.0/0            udp dpt:1812
    0     0 ACCEPT     udp  --  enp1s0 *       0.0.0.0/0            0.0.0.0/0            udp dpt:1813
    0     0 ACCEPT     udp  --  enp1s0 *       0.0.0.0/0            0.0.0.0/0            udp dpt:1814
    0     0 ACCEPT     udp  --  enp1s0 *       0.0.0.0/0            0.0.0.0/0            udp dpt:3799
    0     0 ACCEPT     tcp  --  enp1s0 *       0.0.0.0/0            0.0.0.0/0            tcp dpt:3799 flags:0x17/0x02
    4   336 ACCEPT     icmp --  enp1s0 *       0.0.0.0/0            0.0.0.0/0            icmptype 8
59414   13M REJECT     all  --  enp1s0 *       0.0.0.0/0            0.0.0.0/0            reject-with icmp-port-unreachable
    0     0 DROP       all  --  wlp3s0 *       0.0.0.0/0            0.0.0.0/0
  714 42840 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp dpt:80 flags:0x17/0x02
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp dpt:443 flags:0x17/0x02
    0     0 ACCEPT     tcp  --  *      *       0.0.0.0/0            0.0.0.0/0            tcp dpt:3990 flags:0x17/0x02
  119  9996 ACCEPT     icmp --  *      *       0.0.0.0/0            0.0.0.0/0            icmptype 8
 6061  466K ACCEPT     all  --  lo     *       0.0.0.0/0            0.0.0.0/0

Chain FORWARD (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
```

Рис. 8 — «Совсем другой» результат проверки статуса файервола

Как видите, в этом (втором) примере сразу же бросается в глаза множество строк с правилами, открывающими те или иные порты. Но, в первую очередь нас по прежнему интересует цепь **Input**. В этот раз в примере, показанном на рис. 8, мы видим, что «политика» цепи **Input** уже установлена как «**DROP**» (т. е., «сбрасывать»). Как следствие, файервол сервера не пропустит внутрь любые пакеты, для которых в списке ниже не будет «персонального разрешающего» правила с действием «**ACCEPT**»! Напоминаю, что NetFlow-сенсоры сбрасывают свои данные NetFlow-коллектору на порт 2055 по протоколу UDP. И в примере, показанном на рис. 8, УЖЕ ИМЕЕТСЯ правило, открывающее файервол для указанных целей. Это строка с таким вот содержанием:

```
0     0 ACCEPT     udp  --  enp1s0 *     0.0.0.0/0     0.0.0.0/0     udp dpt:2055
```

**Поэтому, честно говоря, рис. 8 — это все-таки скорее пример ответа файервола, КОТОРЫЙ УЖЕ БЫЛ НАСТРОЕН для приема NetFlow-трафика!*

И теперь, резюмируя, перечислим <u>условия, указывающие на необходимость настройки файервола</u> вашего сервера для приема NetFlow данных:

1. Ваш сервер должен будет получать статистику (NetFlow данные) от ВНЕШНИХ роутеров;
2. «Политика» цепи **Input** файервола вашего сервера установлена как «**DROP**» («сбрасывать» все не разрешенные пакеты, или же, проще говоря, его файервол «закрыт»);
3. В списке правил файервола сервера для цепи **Input** отсутствует правило, разрешающее прохождение UDP-трафика на 2055-й порт;

Чтобы открыть 2055-й UDP-порт вашего сервера выполните следующее:

1. Откройте терминал (консоль) и введите команду:
   ```
   sudo /sbin/iptables -I INPUT -p udp -m udp --dport 2055 -j ACCEPT
   ```
2. Проверьте, что правило было добавлено в список, введя команду:
   ```
   sudo iptables -nvL
   ```
   Вы должны увидеть, что в списке правил цепи **Input** появилось (добавилось в начало списка) правило про 2055-й UDP-порт, аналогичное показанному на рис. 8 выше.
3. Если все ОК (правило в списке присутствует), выполните экспорт текущих правил файервола в специальный файл, из которого сервер будет их загружать при каждой перезагрузке (старте Операционной Системы). Для этого введите команду:
   ```
   sudo /sbin/iptables-save > /etc/network/firewall.rules
   ```

На этом настройка файервола сервера завершена!

Установка и настройка NetFlow-сенсора на роутерах

В зависимости от того, какая именно прошивка используется на вашем роутере, действия по установке (если такое необходимо), настройке и запуску NetFlow-сенсора будут отличаться. Ниже рассмотрены варианты для трех наиболее популярных вариантов — DD-WRT, OpenWRT и Mikrotik.

Включение и настройка NetFlow-сенсора на роутере с прошивкой DD-WRT

В роутерах с прошивкой DD-WRT функция NetFlow названа несколько иначе — Rflow [7] (полагаю, что «собака тут порылась» в каких-то авторских правах компании Cisco или чем-то подобном). Но нам ведь важно не название функции, а результат ее работы! Нужно учесть еще один момент — сенсор Rflow в прошивке DD-WRT МОЖЕТ И ОТСУТСТВОВАТЬ! Такое может произойти, если вы на роутер установили вариант прошивки DD-WRT, «немного ужатый» с целью экономии места (такое практикуется для роутеров с малыми объемами внутренней флеш-памяти). Поэтому, если вдруг вы не можете в своем роутере найти параметры, которые описаны ниже, сверьтесь с таблицей [8] на предмет того, какие именно функции включены в вашу версию прошивки DD-WRT.

Непосредственно включение и настройка выполняются так:

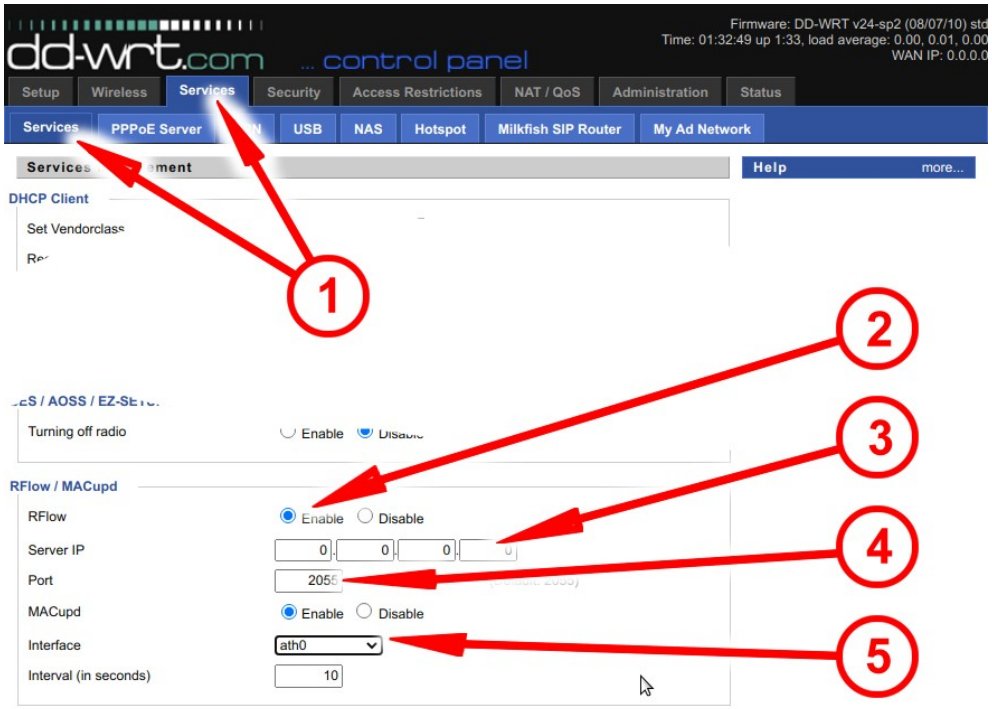


Рис. 9 — Настройка Rflow в прошивке DD-WRT

1. Войдите в веб-интерфейс роутера с прошивкой DD-WRT, откройте меню «**Service**», а в нем — закладку «**Service**» (стрелка-указатель ❶ на рис. 9).
2. Пролистайте вниз страницы и найдите там раздел «**RFlow / MACupd**».
3. В нем проставьте «птички» в полях «**Enable**» для параметров «**RFlow**» (стрелка-указатель ❷ на рис. 9) и «**MACupd**» (без стрелки-указателя).
4. В открывшемся поле «**Server IP**» впишите IP-адрес вашего сервера с биллингом Easyhotspot (стрелка-указатель ❸ на рис. 9).
5. Затем убедитесь, что в поле «**Port**» указано значение 2055 (стрелка-указатель ❹ на рис. 9). Если там указан иной номер порта, то измените его на 2055.
6. В выпадающем списке «**Interface**» выберите тот интерфейс, на котором работает ваш хотспот (стрелка-указатель ❺ на рис. 9). Обычно, посмотреть интерфейс можно, если

перейти в меню «**Service**», затем — в закладку «**Hotspot**» и там уже — в разделе настроек «**Chillispot**».

7. Для завершения нажмите кнопки «**Save**» и «**Apply Settings**», расположенные в самом низу страницы. Рекомендуется перезагрузить роутер после того, как вы сохраните настройки.

Включение и настройка NetFlow-сенсора на роутере с прошивкой OpenWRT

ВНИМАНИЕ: В прошивке OpenWRT по умолчанию НЕ УСТАНОВЛЕН NetFlow-сенсор! Поэтому, перед тем, как его включать и настраивать, сенсор нужно будет сначала установить! Учтите, что файловая система роутера должна иметь достаточное количество свободного места для установки дополнительной программы. И если объем флеш-памяти вашего роутера мал, то может оказаться, что устанавливать программу вам будет банально некуда…

И еще одно небольшое отступление перед тем, как перейти непосредственно к установке и настройке. Если установку программы **softflowd** (а именно так называется программа NetFlow-сенсора в прошивке OpenWRT) еще можно выполнить через веб-интерфейс, то при ее настройке вам все равно придется столкнуться с консолью потому, что веб-страницы с настройками этой функции в прошивке OpenWRT — не предусмотренно! Поэтому, в описании процедуры установки будут приведены еще и консольные команды, которые будут выполнять то же самое, что и показанные на скриншотах действия в веб-интерфейсе. А вот теперь — приступаем…

1. Войдите в веб-интерфейс роутера с прошивкой OpenWRT, щелкните в меню пункт «**System**», а в нем выберите пункт «**Software**» (стрелка-указатель ❶ на рис. 10 ниже):

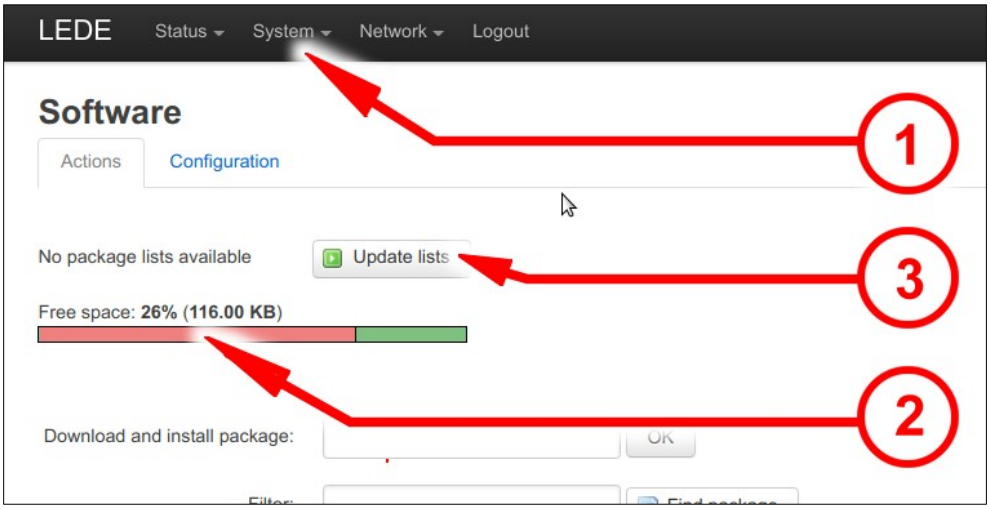


Рис. 10 — Подготовка к установке пакета softflowd в прошивке OpenWRT

2. Проверьте, что в файловой системе роутера осталось свободное место (стрелка-указатель ❷ на рис. 10).

3. Обновите список пакетов (программ), доступных в репозиториях на сервере OpenWRT в интернете. Для этого щелкните кнопку «**Update lists**» (стрелка-указатель ❸ на рис. 10).

   Если же установку пакета вы выполняете в консоли (роутера), то введите такую команду:

   ```
   opkg update
   ```

   ВАЖНО! Учтите, что ваш роутер в этот момент ДОЛЖЕН БЫТЬ ПОДКЛЮЧЕН К ИНТЕРНЕТУ! Иначе он просто не сможет связаться с сервером OpenWRT и не сможет скачать оттуда список доступных пакетов (программ). Да и саму программу softflowd роутер тоже будет устанавливать из интернета, с того же самого сервера OpenWRT. И если доступа в интернет у роутера в этот момент не будет, то все ваши попытки завершатся неудачей!

4. После того, как списки доступного П/О будут успешно получены роутером (стрелка-указатель ❹ на рис. 11 ниже), в поле «**Download and install package**» введите имя пакета (программы) которую хотите установить (***softflowd***), и щелкните кнопку «**OK**» (стрелка-указатель ❺ на рис. 11 ниже):

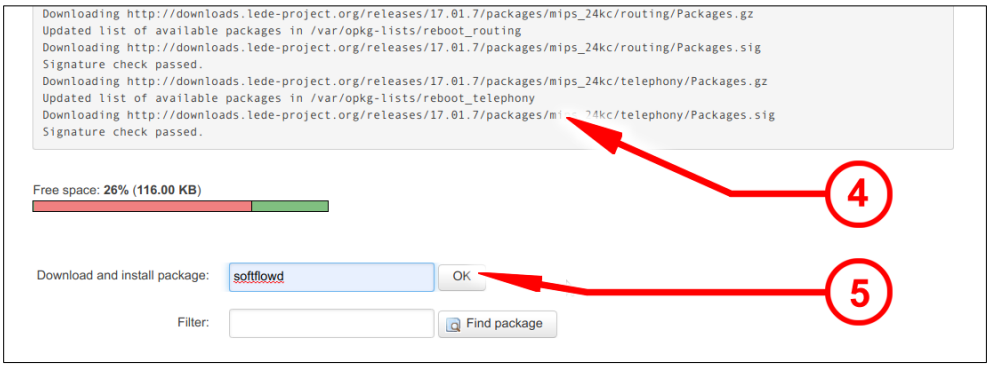


Рис. 11 — Установка пакета softflowd в прошивке OpenWRT

Если же установку пакета вы выполняете в консоли (роутера), то введите такую команду:

```
opkg install softflowd
```

После того, как установка пакета будет успешно завершена, роутер сообщит вам об этом (стрелка-указатель ❻ на рис. 12 ниже):

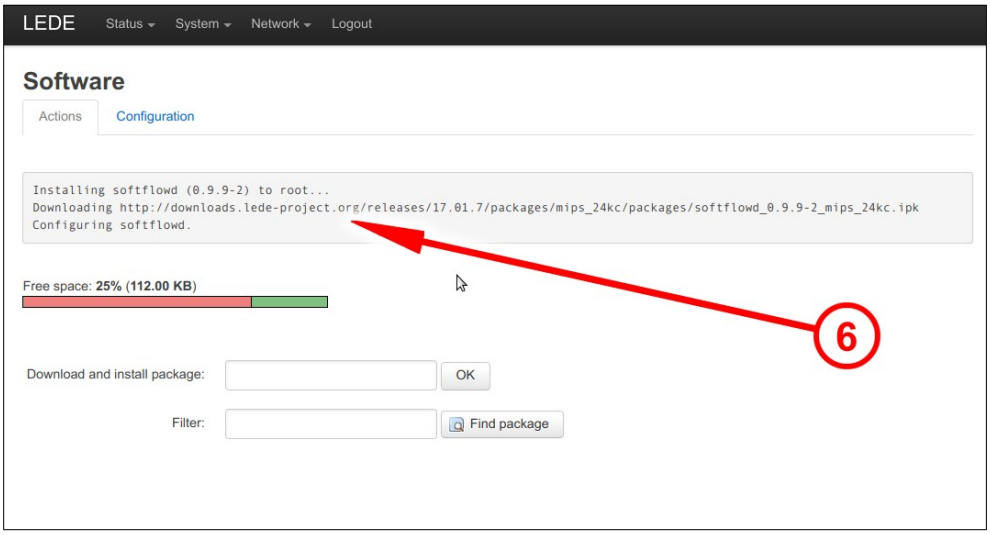


Рис. 12 — Установка пакета softflowd успешно завершена

Теперь программу **softflowd** нужно настроить. Как уже было сказано ранее, в веб-интерфейсе прошивки OpenWRT такой возможности не предусмотрено! Поэтому, вам нужно подключиться в консоль роутера. Вы можете сделать с помощью программы PuTTy, если работаете в ОС Windows [9]. Если же в Linux, то тогда вам подключиться из консоли собственного компьютера в консоль роутера (по ssh), думаю, особого труда не составит.

Чтобы «не заморачиваться» с установкой в роутер каких-либо дополнительных редакторов (и не расходовать понапрасну «бесценное» свободное место во флеш-памяти роутера) править файл настроек мы с вами будем с помощью консольного редактора **vi** (который, возможно, и нельзя отнести к числу удобных, но он включен в состав прошивки OpenWRT по умолчанию). Вам нужно запомнить «трюки» (иначе и не скажешь!) по работе в этом редакторе. Как только откроется редактор, вы будете находиться в «командном режиме». В нем вы не сможете изменить текст. Чтобы переключиться в «режим редактирования», нужно нажать на клавиатуре латинскую букву «`i`». Перейдя в режим редактирования, вы сможете править текст в файле: «стрелками» перемещаться к нужному месту, кнопками

«**Delete**» и «**Backspace**» — удалять старый текст, а новый — просто набирать на клавиатуре. То есть, все — как в любом «обычном» редакторе! Но вот для того, чтобы сохранить изменения и выйти из редактора, вам нужно будет снова вернуться в «командный режим»! Для этого на клавиатуре нужно нажать кнопку «**Esc**». После чего, чтобы сохранить отредактированный файл, на клавиатуре нужно набрать «**:w**», и затем «**Enter**»-ом подтвердите свое согласие (находясь уже в «командном режиме»). Ну и наконец, для того, чтобы выйти из редактора, нужно будет набрать «**:q**» (на всякий случай, еще раз напоминаю — выйти из редактора вы сможете только находясь в «командном режиме»). И вот теперь, когда вы «во всеоружии», мы наконец-то можем приступать к редактированию файла настроек!

5. Находясь в консоли роутера, введите команду:

```
vi /etc/config/softflowd
```

6. Файл с настройками программы **softflowd** будет открыт в редакторе **vi**. Отредактируйте его так, чтобы его содержимое стало выглядеть следующим образом*:

```
config softflowd
        option enabled        '1'
        option interface      'br-lan'
        option pcap_file      ''
        option timeout        ''
        option max_flows      '8192'
        option host_port      '192.168.88.5:2055'
        option pid_file       '/var/run/softflowd.pid'
        option control_socket '/var/run/softflowd.ctl'
        option export_version '9'
        option hoplimit       ''
        option tracking_level 'full'
        option track_ipv6     '0'
        option sampling_rate  '1'
```

ВАЖНО: Естественно, в настройках программы softflowd вашего роутера, вместо выделенного красным цветом IP-адреса моего тестового сервера укажите адрес вашего сервера с биллингом Easyhotspot!

Если в вашем роутере хотспот (Coova-Chilli) настроен так, что обслуживает не весь «мост», включающий в себя все LAN-интерфейсы роутера, а лишь какой-то один из них (отдельный интерфейс), то вместо выделенного в примере синим цветом «br-lan» вы должны будете указать его имя!

7. Сохраните изменения (см. «шпаргалку» выше ☺) и выйдите из редактора.

8. Теперь, чтобы внесенные вами изменения вступили в силу, программу **softflowd** нужно перезапустить. Для этого введите команду:

```
/etc/init.d/softflowd restart
```

9. По окончании настройки роутер рекомендуется перезагрузить.

Установка и настройка NetFlow-сенсора (программы softflowd) на роутере с прошивкой OpenWRT на этом завершена.

Включение и настройка NetFlow-сенсора на роутере Mikrotik

В RouterOS (прошивках роутеров Mikrotik) функция NetFlow-сенсора присутствует, как сейчас модно говорить, «из коробки» и называется она «**Traffic Flow**». Но перед тем, как ее включать и настраивать, вам нужно уточнить пару параметров вашего хотспота, запущенного в роутере (они понадобятся вам во время настройки). Вот с них и начнем!

Во первых, вам нужно узнать имя интерфейса, на котором запущен хотспот (этот параметр впоследствии вы должны будете указать в настройках сенсора в роутере, см. далее). Для этого в программе Winbox выполните следующее:

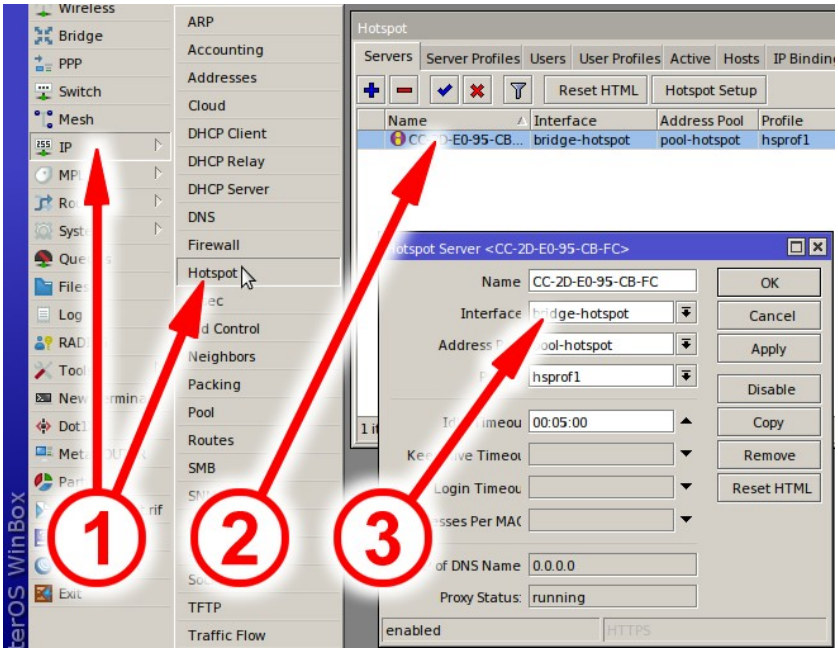


Рис. 13 — Параметры хотспота в роутере Mikrotik

1. Откройте меню «**IP**» и в нем выберите пункт «**Hotspot**» (стрелка-указатель ❶ на рис. 13).
2. В открывшемся окне в закладке «**Servers**» дважды щелкните сервер вашего хотспота (стрелка-указатель ❷ на рис. 13).
3. В поле «**Interface**» прочтите имя интерфейса, на котором запущен хотспот в вашем роутере (стрелка-указатель ❸ на рис. 13).

Кроме того, вам нужно узнать значения адреса и маски сети хотспота, а также адреса ее шлюза (эти параметры вы должны будете вписать в настройки скрипта-парсера на сервере биллинга (см. описание параметра `network` в раздел «*Параметры парсера*» ранее)):

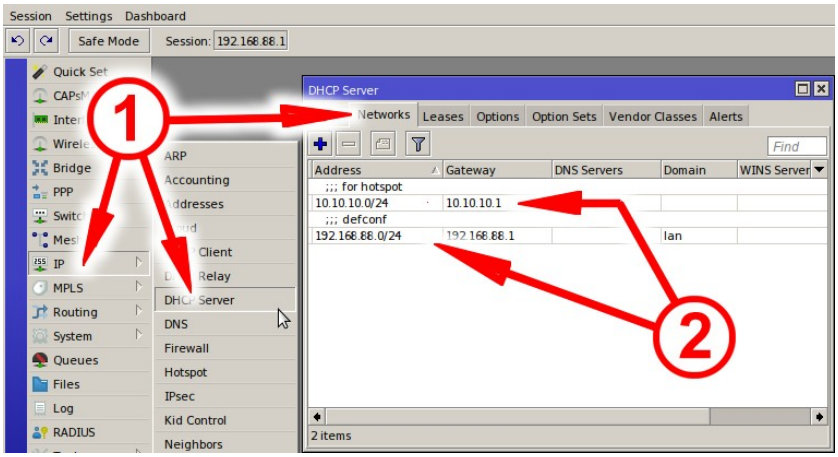


Рис. 14 — Сведения об адресе и шлюзе сети хотспота

1. Откройте меню «**IP**» и в нем выберите пункт «**DHCP Server**». В открывшемся окне щелкните закладку «**Networks**» (стрелка-указатель ❶ на рис.14).

2. В таблице в колонке «**Address**» вы сможете прочесть значения адреса и маски сети хотспота и адреса шлюза в колонке «**Gateway**» (стрелка-указатель ❷ на рис. 14).

   Обращаю ваше внимание на тот факт, что скриншот, приведенный на рис. 14, показывает настройки роутера, у которого сеть хотспота отделена от остальной LAN-сети, и поэтому на скриншоте ДВА разных диапазона сетей. Чаще всего (при «обычной» настройке, когда вся LAN-сеть является хотспотом) он там — один.

Поле того, как вы записали себе на шпаргалку параметры своего хотспота, пришло время включать и настраивать NetFlow-сенсор. Для этого выполните следующее:

1. В меню «**IP**» щелкните пункт «**Traffic Flow**» (стрелка-указатель ❶ на рис.15 ниже). Будет открыто окно с настройками NetFlow-сенсора (*Traffic Flow Settings*):

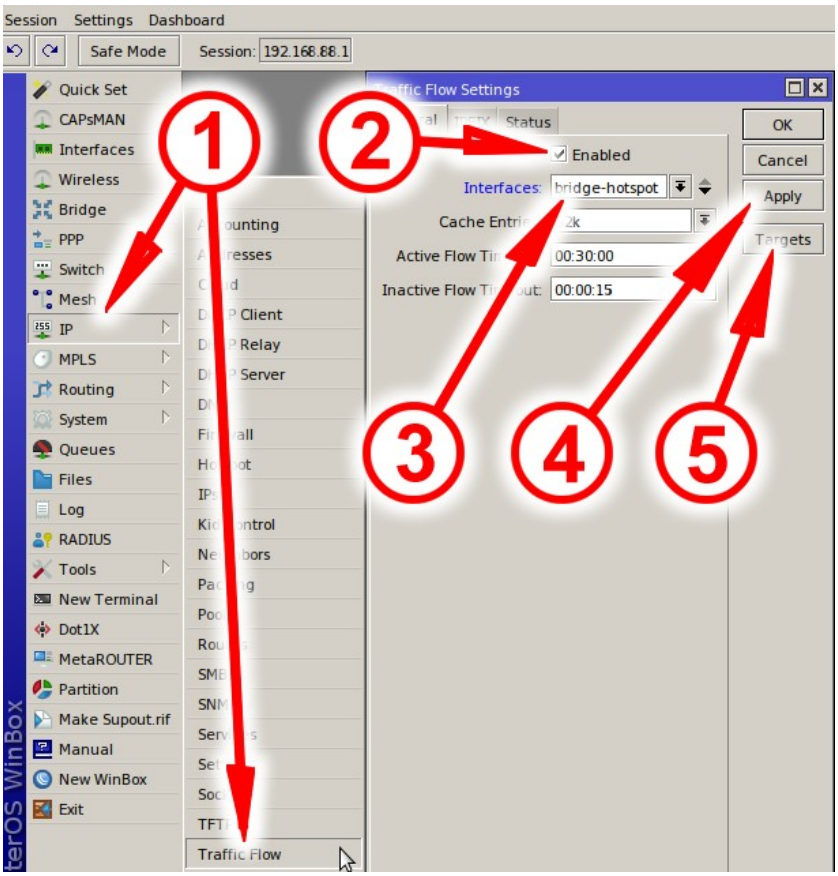


Рис. 15 — Включение NetFlow-сенсора в роутере Mikrotik

2. Поставьте «птичку» в поле «**Enabled**» (стрелка-указатель ❷ на рис. 15). В выпадающем списке «**Interfaces**» (стрелка-указатель ❸ на рис. 15) выберите тот интерфейс, на котором запущен хотспот в вашем роутере (см. показанный ранее рис. 13 и описание к нему).

3. Нажмите кнопку «**Apply**» (стрелка-указатель ❹ на рис. 15). В результате выполнения описанных действий в роутере будет включен NetFlow-сенсор. Теперь вам нужно указать — куда именно сенсор должен сбрасывать собранную им статистику (т. е. — где находится программа NetFlow-коллектора). На всякий случай напоминаю — NetFlow-коллектор у вас будет размещаться на сервере биллинга Easyhotspot. Вот его адрес вам теперь и нужно прописать в настройках. Для этого щелкните кнопку «**Targets**» (стрелка-указатель ❺ на рис. 15). Откроется меню настройки «целей» (*Traffic Flow Targets*):

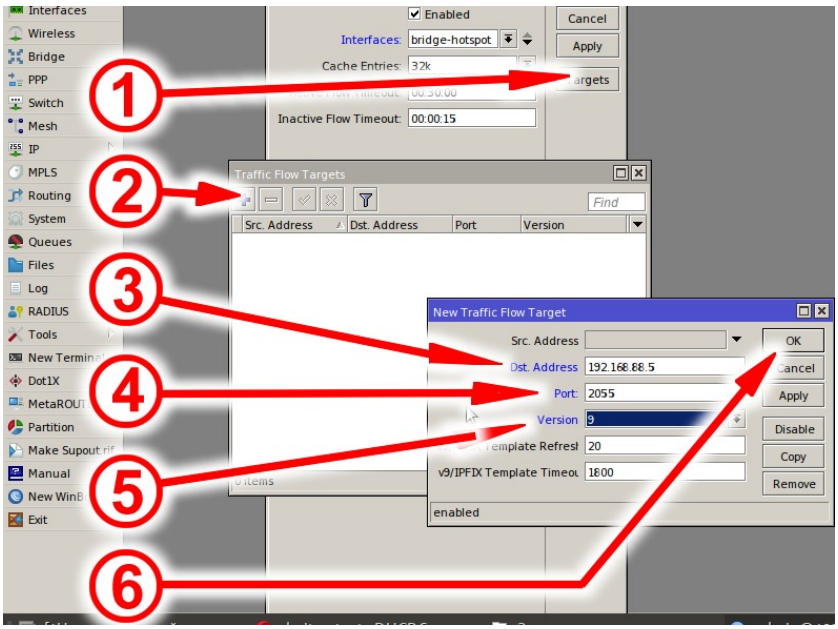


Рис. 16 — Включение NetFlow-сенсора в роутере Mikrotik

4. По умолчанию список «целей» в роутере — пустой. И вам нужно создать в нем новую запись (со сведениями о сервере, на котором размещен NetFlow-коллектор). Для этого нажмите кнопку «**+**» (стрелка-указатель ❷ на рис. 16). Откроется новое окошко со списком параметров — «**Traffic Flow Target**».

5. В этом окошке укажите адрес вашего сервера в поле «**Dst. Address**» (стрелка-указатель ❸ на рис. 16), порт 2055 в поле «**Port**» (стрелка-указатель ❹ на рис. 16) и в выпадающем списке «**Version**» выберите 9 (стрелка-указатель ❺ на рис. 16).

6. Нажмите кнопку «**OK**» (стрелка-указатель ❻ на рис. 16). Новая запись будет добавлена в список «целей».

На этом настройка NetFlow-сенсора в роутере Mikrotik завершена. Для «закрепления результата» роутер лучше перезагрузить.

Работа с NetFlow данными в биллинге Easyhotspot

Биллинг Easyhotspot предоставляет персоналу (кассирам, администраторам) такие возможности:

- просматривать данные о ресурсах, посещенных клиентами хотспотов;
- вести поиск в указанных данных;
- экспортировать эти данные в файлы «электронных таблиц», которые впоследствии могут быть обработаны в таких программах, как Microsoft Excel, Libreoffice Calc и тому подобных.

Как попасть в список ресурсов, посещенных клиентом

В веб-интерфейсе биллинга Easyhotspot «кнопка-иконка», предназначенная для перехода к списку ресурсов, посещенных клиентом, присутствует в двух местах. Данная кнопка обозначается значком «👁». При наведении курсора мыши на данную кнопку в меню появляется всплывающая подсказка «**Список посещенных ресурсов**».

Так где же располагаются эти две кнопки? Во первых — в списке сеансов пользователя (стрелка-указатель ❶ на рис.17):

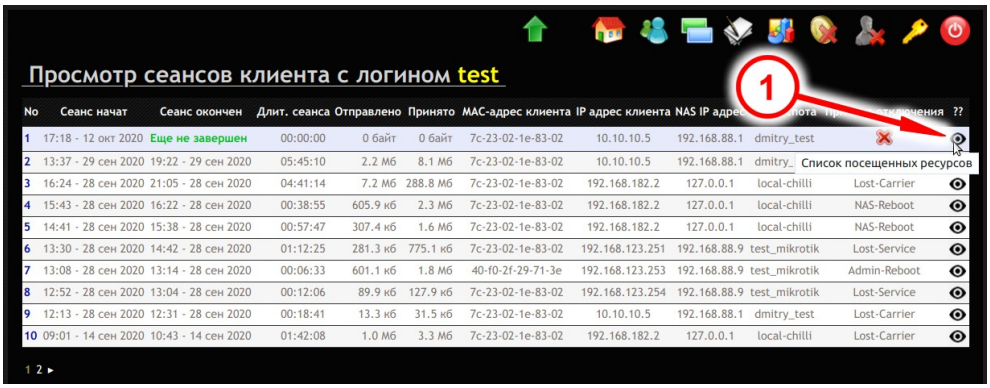


Рис. 17 — Кнопка для перехода к списку посещенных клиентом ресурсов

Причем, не важно, это т. н. «клиент с оплатой по счету» или же это ваучер, в списке сеансов кнопка «👁» будет присутствовать в любом из случаев.

Нажав кнопку «👁» вы попадете в список ресурсов, посещенных клиентом В ТЕЧЕНИЕ ИМЕННО ТОГО СЕАНСА ДОСТУПА В ИНТЕРНЕТ, В СТРОКЕ С КОТОРЫМ ВЫ НАЖАЛИ КНОПКУ!

Вторая кнопка находится в списке активных клиентов — тех, которые «прямо сейчас» подключены к интернету (стрелка-указатель ❷ на рис. 18):

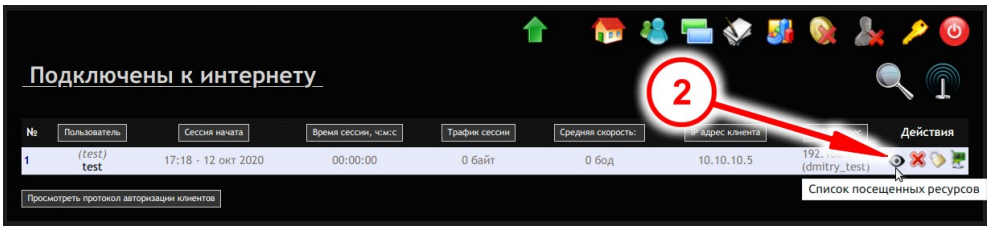


Рис. 18 — Кнопка для перехода к списку посещенных клиентом ресурсов

Нажав кнопку «👁» вы перейдете в список ресурсов, посещенных клиентом В ЕГО ТЕКУЩЕМ (АКТИВНОМ) СЕАНСЕ ДОСТУПА В ИНТЕРНЕТ!

Просмотр списка посещенных ресурсов

Собственно, внешний вид списка посещенных ресурсов мало чем отличается от любых других списков в биллинге Easyhotspot — все та же таблица с информацией:

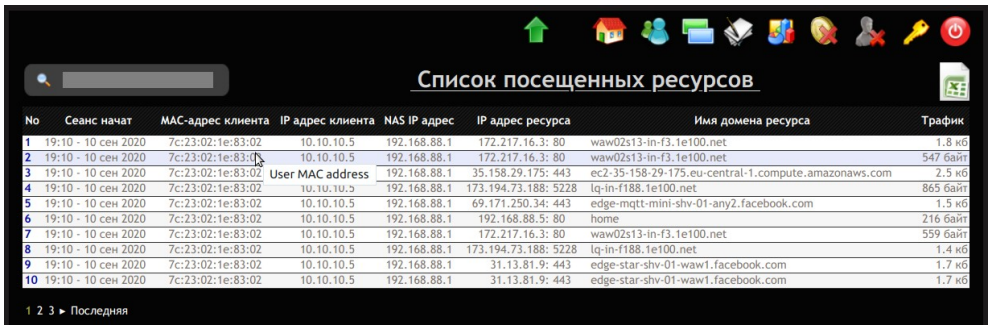


Рис. 19 — Список ресурсов, посещенных клиентом

Если список не умещается на одной странице, под ним будут размещены дополнительные кнопки для навигации по списку.

В списке присутствует такая информация:

Сеанс начат	Дата и время, когда начался обмен данными между клиентом и ресурсом*[1]
MAC-адрес клиента*[2]	Mac-адрес устройства, которым клиент пользовался во время сеанса доступа в интернет*[2]
IP адрес клиента	IP-адрес, выданный клиенту хотспотом.
NAS IP адрес	IP-адрес самого хотспота (либо «удаленного внешнего» роутера, или же программы Coova-Chilli, установленной на самом сервере)
IP адрес ресурса	IP-адрес того ресурса (сайта), с которым клиент обменивался данными
Имя домена ресурса*[3]	Доменное имя того ресурса (сайта), с которым клиент обменивался данными*[3]
Трафик	Объем данных, которыми обменялись клиент и ресурс

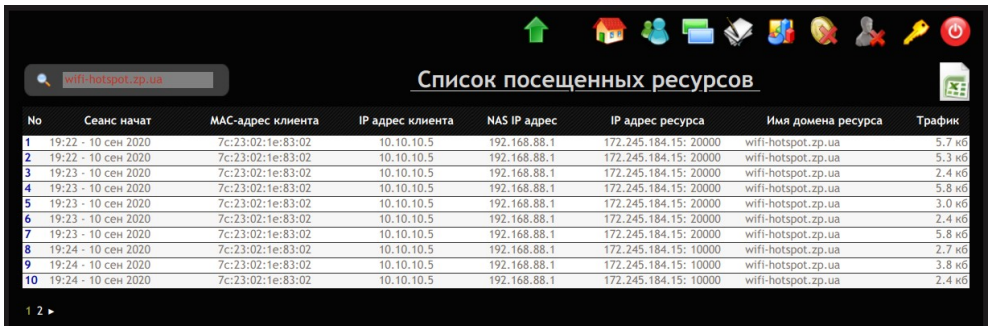
ПРИМЕЧАНИЯ:

- Напоминаю вам, что данная функция биллинга собирает сведения о том, что по английски называется как «***flow***» (т. е., поток, в данном случае — поток данных). И «литься» этот «поток» может какое-то «ощутимое» время. Но функция в Easyhotspot реализована так, что в базу попадают лишь сведения о дате и времени НАЧАЛА данного «потока». Сведения об его окончании игнорируются (в базу не вносятся).
- Сколько я не пытался, мне так и не удалось добиться того, чтобы установленный на (моем домашнем тестовом) сервере Coova-Chilli передавал эту информацию! Поэтому, если ваш хотспот построен таким образом, что сам сервер Easyhotspot является его шлюзом (и клиентов обслуживает установленный на сервере Coova-Chilli), данных о реальных mac-адресах клиентов в этой таблице вероятнее всего не будет!
- Наличие этой информации зависит от двух обстоятельств. Во первых, если в настройках скрипта-парсера параметр «`resolve`» был вами установлен равным «`0`» (см. раздел «*Параметры парсера*»), то биллинг не будет осуществлять преобразование IP-адресов в доменные имена совсем, и в этом столбце будут вписаны те же самые IP-адреса, что и в столбце «**IP адрес ресурса**». А во вторых, не все ресурсы в интернете имеют присвоенные

им имена доменов! И в таком случае, даже если в настройках скрита-парсера параметр «`resolve`» установлен равным «`1`», DNS-запрос все равно вернет пустой ответ! В таких случаях, в базу биллинга для подобных ресурсов также будут записываться их IP-адреса, а не (несуществующие) имена доменов.

Поиск по списку посещенных ресурсов

Программа Easyhotspot позволяет вам осуществлять поиск данных в списке ресурсов, посещенных клиентом хотспота. Для этого достаточно просто ввести желаемый поисковый критерий в поле поиска (слева вверху над списком) и нажать **Enter** на клавиатуре. В ответ программа выведет все записи, которые будут удовлетворять вашему запросу:



No	Сеанс начат	MAC-адрес клиента	IP адрес клиента	NAS IP адрес	IP адрес ресурса	Имя домена ресурса	Трафик
1	19:22 - 10 сен 2020	7c:23:02:1e:83:02	10.10.10.5	192.168.88.1	172.245.184.15: 20000	wifi-hotspot.zp.ua	5.7 кб
2	19:22 - 10 сен 2020	7c:23:02:1e:83:02	10.10.10.5	192.168.88.1	172.245.184.15: 20000	wifi-hotspot.zp.ua	5.3 кб
3	19:23 - 10 сен 2020	7c:23:02:1e:83:02	10.10.10.5	192.168.88.1	172.245.184.15: 20000	wifi-hotspot.zp.ua	2.4 кб
4	19:23 - 10 сен 2020	7c:23:02:1e:83:02	10.10.10.5	192.168.88.1	172.245.184.15: 20000	wifi-hotspot.zp.ua	5.8 кб
5	19:23 - 10 сен 2020	7c:23:02:1e:83:02	10.10.10.5	192.168.88.1	172.245.184.15: 20000	wifi-hotspot.zp.ua	3.0 кб
6	19:23 - 10 сен 2020	7c:23:02:1e:83:02	10.10.10.5	192.168.88.1	172.245.184.15: 20000	wifi-hotspot.zp.ua	2.4 кб
7	19:23 - 10 сен 2020	7c:23:02:1e:83:02	10.10.10.5	192.168.88.1	172.245.184.15: 20000	wifi-hotspot.zp.ua	5.8 кб
8	19:24 - 10 сен 2020	7c:23:02:1e:83:02	10.10.10.5	192.168.88.1	172.245.184.15: 10000	wifi-hotspot.zp.ua	2.7 кб
9	19:24 - 10 сен 2020	7c:23:02:1e:83:02	10.10.10.5	192.168.88.1	172.245.184.15: 10000	wifi-hotspot.zp.ua	3.8 кб
10	19:24 - 10 сен 2020	7c:23:02:1e:83:02	10.10.10.5	192.168.88.1	172.245.184.15: 10000	wifi-hotspot.zp.ua	2.4 кб

Рис. 20 — Результаты поиска по списку ресурсов, посещенных клиентом

ПРИМЕЧАНИЯ:

- Поиск осуществляется по сведениям, находящимся в таких колонках базы данных: «**MAC-адрес клиента**», «**IP адрес клиента**», «**NAS IP адрес**», «**IP адрес ресурса**» и «**Имя домена ресурса**»;
- Ключевая фраза поиска останется как «подсветка» в форме поиска, чтобы вы были в курсе, что именно пытались найти;
- Если сведения, удовлетворяющие поисковому запросу, не уместятся на одной странице, то под таблицей будут размещены дополнительные кнопки для навигации по списку;
- Результаты поиска учитываются при экспорте данных в файлы «электронных таблиц»;
- Если в базе не будет найдено записей, удовлетворяющих поисковому запросу, программа сообщит вам об этом, а таблица на странице будет отсутствовать;
- Если вы запустите поиск, не введя ключевую фразу, программа сообщит вам об этом, а в таблице будут показаны все данные так, как будто вы ничего и не искали.

Экспорт данных в файл

Программа позволяет вам экспортировать сведения о ресурсах, посещенных клиентами, в фал «электронных таблиц», которые впоследствии могут быть обработаны в таких программах, как Microsoft Excel, Libreoffice Calc и тому подобных.

Для того, чтобы экспортировать данные в файл, щелкните кнопку «», расположенную справа вверху над таблицей. Программа некоторое время потратит на формирование файла с данными, после чего браузер предложит вам либо сохранить этот файл, либо сразу же открыть его в какой-нибудь программе на ваш выбор.

ВАЖНО! Учтите, что при экспорте данных учитываются результаты поиска! То есть, если вы сначала выполните поиск, а потом нажмете кнопку экспорта, то в файл будут включены лишь те данные, которые удовлетворяют результатам вашего поиска.

Устранение неполадок

В биллинге полностью отсутствуют данные о ресурсах, посещавшихся клиентами

- **Возможно, прошло еще слишком мало времени**. Дело в том, что программы NetFlow-сенсоры хотспотов сбрасывают данные о потоках не сразу же после прохождения каждого байта от клиента к сайту или обратно, а какое-то время накапливают их в своих буферах. И лишь накопив некий объем этой информации, или обождав установленный промежуток времени, они связываются с программами NetFlow-коллекторами, чтобы «пачкой слить» им накопившуюся информацию. И точно так же поступают и сами программы NetFlow-коллекторы — полученную от сенсоров информацию они какое-то время хранят у себя в буферах, и лишь по прошествии установленного интервала времени сбрасывают ее на диск, параллельно вызывая скрипт-парсер для обработки этих файлов. По умолчанию в биллинге Easyhotspot указанные программы настраиваются на интервал в 5 (пять) минут! Если прошло меньше времени, то данные могут еще отсутствовать в базе биллинга (но не переживайте — они никуда не пропадут, они просто продолжают храниться в буферах программ, ожидая своего времени для «слива» в базу).

- **Не установлен или не запущен демон «NetFlow-сенсора» fprobe**. **ВАЖНО — эту проверку вам нужно выполнять только в том случае, если шлюзом вашего хотспота служит сам сервер Easyhotspot, для чего на нем установлен и запущен контроллер хотспота — программа Coova-Chilli! Если же ваш сервер обслуживает только удаленные («внешние») хотспоты (роутеры), просто пропустите этот пункт проверки!** Чтобы проверить, что fprobe установлен и работает, введите в консоли (терминале) следующую команду:

  ```
  sudo ps ax | grep fprobe
  ```

  В ответ (в одной из нескольких строк) вы должны получить нечто подобное:

  ```
  9373 ?  Ssl  0:00 /usr/sbin/fprobe -iwlp3s0 -fip localhost:2055
  ```

  Если же в ответе нет подобной строки (в которой будут указаны как сам файл программы с полным путем к нему, так и параметры его запуска), вам нужно повторить процедуры из разделов «*Установка NetFlow-сенсора fprobe*» и «*Настройки NetFlow-сенсора fprobe*», чтобы получить успешный результат.

- Не установлен или не запущен демон «NetFlow-коллектора» nfcapd. Чтобы проверить, что fprobe установлен и работает, введите в консоли (терминале) следующую команду:

  ```
  sudo ps ax | grep nfcapd
  ```

  В ответ вы должны получить нечто подобное (показанный в примере текст будет в одной строке, но самих подобных строк в ответе может быть несколько):

  ```
  1298 ? S  0:00 /usr/bin/nfcapd -D -l /var/cache/nfdump -P
  /var/run/nfcapd.pid -p 2055 -T all -x /root/parse_nfcap.pl
  ```

  Если же в ответе не будет таких строк (в которых будут указаны как сам файл программы с полным путем к нему, так и параметры его запуска), вам нужно повторить процедуры из разделов «*Установка NetFlow-коллектора nfdump*» и «*Настройки NetFlow-коллектора nfdump*», чтобы получить успешный результат.

  ВАЖНО: С ДРУГОЙ СТОРОНЫ, если данная проверка показывает, что демон nfcapd на сервере установлен и работает, а данных о «потоках» в базе все равно нет, проблема может заключаться в том, что **в файерволе сервера не открыты нужные порты** (в результате чего данные от внешних роутеров не могут поступать в сервер). Как проверить — см. следующий пункт.

- **На сервере не открыты порты, необходимые для передачи данных от NetFlow-сенсоров к NetFlow-коллектору** (на всякий случай, напоминаю, речь идет про протоколе UDP и порт 2055). Также напоминаю, что у сервера может быть два варианта настройки «политик», в зависимости от чего принудительное открытие необходимого порта может быть как нужным, так и не обязательным. Поэтому, сначала проверьте, какая именно из «политик» установлена в файерволе вашего сервера для цепи INPUT. Для этого введите в консоли (терминале) такую команду:

  ```
  sudo iptables -nvL
  ```

  В ответе, полученном на эту команду, вас интересует ПЕРВАЯ строка, в которой может быть написано либо так:

  ```
  Chain INPUT (policy DROP 0 packets, 0 bytes)
  ```

  … либо вот так:

  ```
  Chain INPUT (policy ACCEPT 0 packets, 0 bytes)
  ```

  Как видите, «политики» в приведенных примерах отличаются (для наглядности выделены красным цветом)! Так вот, если у вас «политика» установлена как «**ACCEPT**», то отдельное правило про 2055-й порт НЕ НУЖНО, т. к. файервол вашего сервера и так пропустит на вход все что угодно! Если же «политика» установлена как «**DROP**», то тогда проверьте, что 2055-й порт открыт. Для этого в консоли (терминале) введите следующую команду:

  ```
  sudo iptables -nvL | grep 2055
  ```

  В ответе вы должны получить строку, похожую на эту (порт и протокол выделены в данной инструкции красным цветом только для наглядности, в реальности этого не будет):

  ```
  2584  155K  ACCEPT  udp  --  * *  0.0.0.0/0  0.0.0.0/0  udp dpt:2055
  ```

  Такой ответ означает, что в файерволе порт 2055 по протоколу UDP открыт. Кстати, первые две цифры в строке с ответом будут означать число пакетов и число байт, которые прошли через файервол именно по этому разрешающему правилу. Если же строка есть (и порт открыт), но цифры там — нули, то это значит, что данные от NetFlow-сенсоров роутеров не поступают на вход сервера вообще (либо в роутерах что-то настроено не правильно, например, адрес сервера, либо провайдер блокирует трафик по 2055-му порту, либо ваш сервер за NAT-ом, см. следующий пункт). Если же подобный ответ отсутствует, а «политика» файервола вашего сервера для цепи **INPUT** установлена как «**DROP**», то вам нужно открыть порт. Как это сделать — читайте в разделе «*Настройка файервола на сервере биллинга*».

- **Ваш сервер Easyhotspot находится за NAT-ом какого-нибудь роутера** (например, сервер размещен у вас дома, в локальной сети, а к провайдеру эта ваша локальная домашняя сеть подключена через домашний роутер). В таком случае настройте в вашем роутере функцию «портфорвардинг» (может еще называться как «виртуальный сервер») чтобы данные, поступающие на порт 2055 роутера по протоколу UDP пересылались на сервер биллинга.

- **NetFlow-сенсоры в роутерах настроены не правильно** (например, в настройках указан не верный адрес вашего сервера с биллингом Easyhotspot). Пользуясь информацией из раздела «*Установка и настройка NetFlow-сенсора на роутерах*» (и дочерних), еще раз перепроверьте и при необходимости исправьте настройки NetFlow-сенсоров в роутерах.

- **NetFlow-сенсоры в роутерах не стартуют** (частный случай, обнаруженный в прошивках OpenWRT) — см. информацию из раздела «*Не стартует Netflow-сенсор softflowd в роутере с прошивкой OpenWRT*».

Функция сбора Netflow-статистики создает повышенную нагрузку на сервер

- В первую очередь, вам нужно учитывать простое и логичное обстоятельство — чем больше данных будет поступать со всех Netflow-сенсоров всех роутеров вашей сети, тем больше времени и вычислительных ресурсов сервер вынужден будет тратить на их обработку. Очень не желательно, чтобы продолжительность работы скрипта-парсера превышала тот интервал, с которым осуществляется его вызов. Это приведет к тому, что одновременно будут запущены несколько его экземпляров, что влечет за собой рост нагрузки на сервер. В таком случае рекомендуется уменьшить интервал, с которым NetFlow-коллектор nfdump выполняет сброс данных из памяти на диск (читайте про параметр «**-t**» в разделе «*Настройки NetFlow-коллектора nfdump*»). Это повлечет за собой более частый вызов скрипта-парсера (т. к., именно nfdump вызывает скрипт-парсер по окончании сброса дампов на диск), а следовательно — и меньший объем данных, которые парсер должен будет обработать за один запуск. То есть, вам нужно постараться добиться того, чтобы продолжительность выполнения скрипта стала меньше того интервала, с которым осуществляется его запуск. Увидеть, сколько именно времени скрипт-парсер тратит на обработку данных, вы сможете в файле **/tmp/netflow_parse.log**, если установите «уровень дебага» отличным от нуля (читайте про параметр «**debug**» в разделе «*Параметры парсера*»). Как только вы установите его равным «**1**» или выше, в лог-файле появятся вот такие строки:

  ```
  Execution time: 0.0951519012451172 s
  ```

  В этих строках как раз и показана продолжительность времени, в течение которой скрипт был запущен. Время приведено в секундах. Точка является разделителем дробной и целой частей значения (т. е., в показанном выше примере продолжительность работы скрипта составила менее одной десятой секунды).

  *ps. Не забудьте вернуть значение «**0**» параметру «**debug**» после того, как завершите проверку.*

- Тестирование мини-ПК (Celeron N2815 1.86ГГц / 4 Гб ОЗУ) с установленными на нем биллингом Easyhotspot, программами Coova-Chilli, fprobe, nfcapd и скриптом-парсером показало, что КРАЙНЕ НЕ ЖЕЛАТЕЛЬНО В РЕАЛЬНОЙ РАБОТЕ ОСТАВЛЯТЬ ВКЛЮЧЕННЫМ 4-й «уровень дебага» в настройках скрипта-парсера (см. раздел «*Параметры парсера*», параметр **debug**)! Это связано с тем, что при 4-м «уровне дебага» сервер не удаляет старые файлы с дампами статистики, из-за чего скрипт-парсер при каждом своем запуске вынужден «перелопачивать» весь этот ворох файлов! И не важно, что «старые» файлы он даже не считывает, а просто проверяет на факт наличия их в списке блокировки — на это уходят и время и ресурсы сервера! То есть, если вдруг вы решили, что вам это нужно, и вы включили 4-й «уровень дебага», то после того, как поэкспериментировали, посмотрели на данные в raw-формате и т.д. и т. п. — обязательно выключайте его (меняйте уровень на более низкий — с 3-го по 0-й)! После этого обязательно удалите все файлы из папки **/var/cache/nfdump** а также файлы **netflow_parse.log** и **parse_nfcap.lock** из папки **/tmp**! Уточняю — речь именно 4-й «уровень дебага», для более низких уровней вся эта «страшилка» не актуальна)!

Не стартует Netflow-сенсор softflowd в роутере с прошивкой OpenWRT

В ходе экспериментов на виртуальную машину была установлена прошивка OpenWrt версии 19.07.2 (r10947-65030d81f3). Попытки получить от нее Netflow-данные, следуя инструкциям, приведенным мной в разделе «*Включение и настройка NetFlow-сенсора на роутере с прошивкой OpenWRT*», **к успеху не привели!** «Гуглеж» проблемы привел к множественной информации о том, что в «свежих» версиях прошивки OpenWrt разработчики «что-то поломали» (вроде как, проблема заключается в неправильном «парсинге» файла конфигурации демона softflowd, из-за чего он и отказывается стартовать). В качестве «костыля» в найденных мной статьях (заметках в форумах), предлагалось добавить команду запуска softflowd в «специальный» файл **rc.local** (в него зачастую помещают команды, которые ОС должна выполнить при запуске). Проверка данного «костыля» также к успеху не привела. Поэтому, я предлагаю вам разместить команду для принудительного запуска softflowd в другом файле — **/etc/chilli/ipup.sh** (это тоже «специальный» файл скрипта, который обязательно выполняется после запуска Coova-Chilli). По крайней мере, проверка мною именно такого способа решения данной проблемы прошла успешно!

Для этого выполните следующее:

1. Подключитесь в консоль вашего роутера с прошивкой OpenWRT (например, с помощью программы PuTTy).
2. Сначала убедитесь, что softflowd действительно не работает. Для этого введите команду:

   ```
   ps |grep softflowd
   ```

   Вы должны получить либо «пустой» ответ, либо подобный:

   ```
   4985 root      1080 S    grep softflowd
   ```

   Любой из этих вариантов (пустой или показанный выше) означают, что демон не запущен!
3. Проверьте, что файл скрипта, который вам нужно создать, в роутере отсутствует (чисто на всякий случай). Для этого введите команду:

   ```
   ls /etc/chilli/ipup.sh
   ```

   Вы должны получить подобный ответ:

   ```
   ls: /etc/chilli/ipup.sh: No such file or directory
   ```
4. Раз искомый файл отсутствует, значит все ОК, и мы можем его создавать. Для этого введите команду:

   ```
   touch /etc/chilli/ipup.sh
   ```

   Ответ на данную команду ОТСУТСТВУЕТ (система «молча» создаст указанный вами файл)! Рекомендую вам повторно проверить существование этого файла. Для этого повторно выполните команду из пункта 3. Но в этот раз (после того, как нужный файл был создан), ответ должен быть уже иным (система просто выведет имя файла с полным путем к нему):

   ```
   /etc/chilli/ipup.sh
   ```
5. После того, как файл создан, вы должны его отредактировать. Для этого введите команду:

   ```
   vi /etc/chilli/ipup.sh
   ```

   Файл будет открыт в редакторе **vi**. Т.к., файл был только что создан, он БУДЕТ ПУСТЫМ (в нем не будет вообще ничего)! Еще раз перечитайте шпаргалку из п. 4 раздела «*Включение и настройка NetFlow-сенсора на роутере с прошивкой OpenWRT*» (где написано о том, как работать в этом «чудесном» редакторе).

   Также, прочитайте информацию по параметрам, которые должны были быть вписаны в файл настроек softflowd (см. п. 6 из раздела «*Включение и настройка NetFlow-сенсора на роутере с прошивкой OpenWRT*»). Вам нужны будут значения двух параметров — **interface** и **host_port**. (они «индивидуальны» для каждой инсталляции, и их значения, корректные

именно для вашего случая, вы должны будете вписать в команду запуска, приведенную ниже).

6. Впишите в файл скрипта ДВЕ такие (новые) строки:

```
#!/bin/sh
/usr/sbin/softflowd -i br-lan -v 9 -n 192.168.88.5:2055 -d &
```

   Естественно, вместо выделенных красным цветом значений параметров укажите верные ДЛЯ ВАШЕЙ СИСТЕМЫ значения!

7. Сохраните файл и выйдите из редактора.

8. Назначьте файлу аттрибут, указывающий, что он «исполняемый», командой:

```
chmod +x /etc/chilli/ipup.sh
```

   Ответа на эту команду тоже не будет.

9. Проверьте, что аттрибут «исполняемый» был вами успешно присвоен файлу. Для этого введите такую команду:

```
ls -l /etc/chilli/ipup.sh
```

10. Вы получите вот такой ответ:

```
-rwxr-xr-x   1 root   root   72 Oct 18 16:22 /etc/chilli/ipup.sh
```

   В этом ответе вас интересует блок аттрибутов файла (текст «`-rwxr-xr-x`») в самом начале строки. А в нем — факт наличия букв «`x`» (символ «`x`» был взят от слова «*execute*», которое переводится как «исполнение», «выполнение» и т. п.). Если буквы «`x`» в указанном блоке есть — значит все ОК, атрибут «исполняемый» был вами успешно присвоен файлу.

11. Перезагрузите роутер. После того, как он загрузится, снова подключитесь к его консоли, и еще раз проверьте — запущен ли softflowd командой, приведенной в п. 2 выше. Ответ на этот раз должен быть таким:

```
2107 root   3052 S   /usr/sbin/softflowd -i br-lan -v 9 -n
192.168.88.5:2055 -d

4985 root   1080 S   grep softflowd
```

   Подтверждением того факта, что softflowd работает, будет являться ПЕРВАЯ строка из показанного выше примера ответа (на самом деле, это будет именно одна строка, просто в данной инструкции ее содержимое в одну строку не уместилось). И если вы видите подобную строку у себя (с поправкой на те значения параметров, которые вы указали для своего роутера), то значит, все хорошо, softflowd РАБОТАЕТ!

В данных статистики отсутствуют mac-адреса клиентов

Тут я вынужден начать с грустного. Во всех моих экспериментах на домашнем тестовом сервере мне так и не удалось настроить NetFlow-сенсор, чтобы он сбрасывал данные о mac-ах, работая на хотспоте, который реализован с помощью программы CoovaChilli (речь о ситуации, когда сервер Easyhotspot выступает еще и шлюзом локального хотспота) ☹. Увы! Единственное оборудование, которое честно сбрасывало mac-адреса в NetFlow-статистике — это роутеры Mikrotik! Исходя из этого первого «глобального» препятствия, выросло и все остальное. А теперь по пунктам

Версия NetFlow-протокола, используемая сенсором, не может быть 5-й (и ниже). По крайней мере, если в роутере Mikrotik, от которого данные о mac-ах поступают, переключить NetFlow-протокол с 9-го на 5-й, то информация о mac-ах в статистике тут же пропадает! С другой стороны, и данная инструкция описывает, и скрипт-инсталлятор устанавливает на сервер биллинга программу сенсора fprobe, у которой на данном этапе «самая максимально свежая» поддерживаемая версия NetFlow-протокола — всего лишь 7-я! И вот о том, может ли эта 7-я версия передавать сведения о mac-ах — увы, не известно… С другой стороны, в ходе экспериментов на своем сервере я устанавливал две других программы NetFlow-сенсоров, которые поддерживали версии протокола 9-ю и выше. Речь идет про **softflowd** и **ipt_NETFLOW**. Они успешно устанавливались, правильно настраивались, включался 9-й протокол, статистика начинала поступать в биллинг, все ОК!… Но вот информации о mac-ах в статистике как не было, так и нет! А установка и настройка этих программ в плане «заморочек» разных — еще та (по сравнению с легко устанавливающейся fprobe)!

Проверить NetFlow-сенсоры в прошивках DDWRT и OpenWRT мне было банально не на чем, имеющийся роутер TP-Link WR741D оказался «слаб» для этого как в объеме памяти, так и в производительности процессора. Поэтому, все эксперименты, в которых mac-адреса успешно передавались в биллинг, проходили исключительно на роутерах Mikrotik.

Но и с Mikrotik-ами тоже есть нюанс! Дело в том, что если следовать документации к nfdump [6], то mac-адрес «источника» (гаджета клиента хотспота) указывается как значение поля **%ismc** (Input Src Mac Addr) в данных статистики. Но в роутерах Mikrotik mac-адрес клиента хотспота почему-то передается в поле **%odmc** (Output Dst Mac Addr)! И скрит-парсер сейчас написан именно с учетом данного обстоятельства! Так что, вполне может оказаться, что если кому-то таки удастся получить mac-и от Coova-Chilli, то ему придется еще и скрипт-парсер править, меняя в нем строку «**%odmc**» на «**%ismc**»…

Резюмируя — на данном этапе информация о mac-адресах клиентов в биллинге Easyhotspot присутствует только лишь в статистике, полученной от роутеров Mikrotik.

Ссылки

1. Netflow, материал из Википедии — свободной энциклопедии:
   https://ru.wikipedia.org/wiki/Netflow
2. Страница программы nfdump на сайте github:
   https://github.com/phaag/nfdump
3. Страница программы fprobe на сайте sourceforge:
   https://sourceforge.net/projects/fprobe/
4. Man программы nfcapd (на английском языке):
   https://manpages.ubuntu.com/manpages/bionic/en/man1/nfcapd.1.html
5. Man программы fprobe (на английском языке):
   https://manpages.ubuntu.com/manpages/bionic/en/man8/fprobe.8.html
6. Man программы nfdump (на английском языке):
   https://manpages.ubuntu.com/manpages/bionic/en/man1/nfdump.1.html
7. Краткая информация по использованию Rflow-коллектора на сайте DD-WRT:
   https://wiki.dd-wrt.com/wiki/index.php/
   Using_RFlow_Collector_and_MySQL_To_Gather_Traffic_Information
8. Таблица функций, включенных в ту или иную версию прошивки DD-WRT:
   https://wiki.dd-wrt.com/wiki/index.php/Version_Features#K2.4_Build_Features
9. Статья «Putty — вход в консоль Linux из Windows», размещенная в моем блоге:
   https://wifi-hotspot.zp.ua/wp/2009/02/putty-console-to-linux-from-windows/