**Инструкция по установке ПО «sDPI» и работе с ним**

[1. Общие сведения………………………………………………………………………………………...](#_Toc10728)2

[2. Ветки развертывания проекта, используемые для разработки и поддержки 2](#_Toc15772)

[3. Узлы, готового проекта 3](#_Toc32395)

[4. Инструкция по сборка проекта 3](#_Toc18572)

[5. Просмотр состояния клиента 4](#_Toc336)

[6. Доступные скрипты 4](#_Toc6941)

[7. Конфигурация решения 5](#_Toc10266)

[8. В случае падения 15](#_Toc15852)

9. Контакты …………………...…………………………………………………………………………17

1. Общие сведения.
В основном проекте sDPI содержится несколько подпроектов:

\* sdpi - основной проект, сам sDPI

\* collector - сюда sDPI отправляет биллинг, детальную статистику; CGNAT отправляет сюда

сообщения о трансляциях

\* control - утилита для просмотра состояния sDPI - можно смотреть

общую статистику по сессиям, траффику и т.д., а также статистику для конкретного клиента, в т.ч.

текущую скорость (в т.ч. до и после шейпинга)

\* lib - исходники с кодом, общим для всех подпроектов

\* splitter - разделяет логи детальной статистики коллеткора по файлам

1. Ветки развертывания проекта, используемые для разработки и поддержки.

\* master - самая стабильная (и старая) ветка, в нее добавляется код, который долго и без явных

ошибок работает в продукте

\* developer - ветка, которая запускается и работает в продукте, может содержать еще не

выявленные ошибки

\* cgnat\_statistics - ветка с коллеткором для CGNAT-а

\* sdpi\_cgnat\_merging - ветка с проектом CGNAT, встроенным в sDPI (copy-paste оригинального

CGNAT с поправкой на структуру sDPI); - используется в тестовом режиме

\* cgnat\_nextgen - ветка с CGNAT-ом, встроенном в sDPI (сам CGNAT переписан практически с

нуля, лишь частично заимствован из проекта CGNAT); - используется только в тестовом режиме

\* client\_instant\_detection - отдельный поток для определения клиента для сессии, поиск

клиента осуществляется непосредственно при создании сессии; запущено на sdpi-asbest и sdpi-pervouralsk. Так же добавлен snmp-хэндлер для статистики по правилу Soft-Block, и возможность менять параметры РКН-блокировки через IPC-команды.

\* dpi\_pkt\_t-list - убраны struct pkt\_queue\_2; dpi\_pkt\_t выстраиваются в список с помощью внутреннего указателя

\* rkn - вставлен механизм РКН-блокировок из DSF

\* useragents - добавлен парсинг поля Useragent в теле HTTP-запроса, отправка его в QoE;

\* flow-allocators - сессии аллоцируются из пулового аллокатора; - используется в тестовом режиме

\* dpdk - переход с PF\_RING (ZC) на DPDK (на данный момент версии 19.11 LTS); основная

обработка траффика дописана

1. Узлы, готового проекта.

\* sdpi-ekb1 (10.205.7.58) - главный узел в Екатеринбурге. 2-сокетный сервер, 64 ядра, почти 200 ГБ памяти.

На нем запускаем новые версии в последнюю очередь, когда на всем остальном уже протестили

как следует. Сейчас запущена ветка useragents

\* sdpi-ekb2 (10.205.7.40) - второй узел в Екатеринбурге. Обрабатывает траффик перед CGNAT1. Сейчас запущена ветка useragents

\* sdpi-asbest (10.66.246.200) - узел в Асбесте. Обрабатывает траффик перед Асбестовским

CGNAT. Сейчас запущена ветка client-instant-detection

\* sdpi-pervouralsk (10.210.192.215) - узел в Первоуральске. Сейчас запущена ветка client-instant

detection

1. Инструкция по сборка проекта.

Каждый проект собирается с помощью CMake (кроме sdpi в ветке dpdk - там используется Make с

инструментами сборки из DPDK).

Команды для сборки:

cmake -Bbuild -H.

cd build

make

В CMakeLists.txt каждого проекта можно добавить/убрать нудные дефайны. При сборке на проде

нужно обязательно проверить, с какими дефайнами собирался проект, а также какие изменения в

исходниках были сделаны, при необходимости, нужно будет их воспроизвести, после того, как

будет сделан git pull:

cd /src/sDPI

git diff

sdpi:

Средства управления с помощью команд расписаны в wiki-ess, здесь укажу, что основное

взаимодействие с sDPI осуществляется через БД (Postgres).

Список клиентов, их адреса, пакеты (packages), правила и др., все это подтягивается из БД.

collector:

Собирает биллинг и статистику со всех узлов sDPI. Ночью splitter разделяет их по файлам и

кладет в gzip-архив по пути /data/traffic/sdpi/detail/татиclients///.

Файлы статистики с sDPI всегда должны быть хотя бы за последние 3 месяца. Все, что старше -

можно и нужно чистить.

Каждый день эти файлы за предыдущий день копируются на внешнее хранилище (подключено по

NFS; смонтировано в /storage). Когда оно переполняется, диск меняется и все по новой, главное

запомнить, за какую даты последнюю скинулись полные логи, и соответственно недостающей

закинуть на новый диск.

Файлы событий с CGNAT-коллектора хранятся 2 недели (возможно конфигурирование сроков хранения)

Скрипт idle\_logs\_cleaner.sh проверяет, есть ли файлы, которые можно удалить (с аргументом check), либо удаляет (с аргументом delete). Синтаксис вызова можно посмотреть в crontab-e. Все его действия логируются в STDOUT (в кроне идет перенаправление в syslog).

1. Просмотр состояния клиента

/usr/bin/sdpi\_ctl -Vh localhost

Чтобы посмотреть скорость:

show client 10.3.2.132 speed interval=1 count=10 in out shaper

Interval - интервал между отсчетами в секундах

count - количество отсчетов (бесконечный цикл пока не сделан, по Ctrl+C не остановится, так что сильно много не ставь)

in, out - направления (по стандарту в обе стороны будет показывать)

shaper - покажет также значения до шейпера (и другой обработки)

История команд работает - можно получить путем введения команды Help

1. Доступные скрипты

asn\_parser.py

asn\_updater.py - Обновляет список автономных систем в таблице traffic\_categories\_asns в базе данных sdpi.

backup\_all.sh - Делает бэкап конфигов, скриптов, бинарников и др.

bypass.init.d - Инициализирует утилиту bpctl\_util для перевода сетевых интерфейсов на bypass.

bypass\_off.sh - Снимает все сетевые интерфейсы с bypass

bypass\_on.sh - Переводит все сетевые интерфейсы на bypass

pf\_ring.init.d

- Имеет имя файла pf\_ring после установки и располагается в директории /etc/init.d/

- Для запуска требует указания ключа start. Пример:

/etc/init.d/pf\_ring start

- Загружает драйвер сетевой карты, идущий в поставке с pf\_ring, с привязкой RSS-очередей сетевых интерфейсов.

- Выделяет указанное в скрипте число больших страниц (hugepages).

rkn\_parser.py

sdpi.init.d

sdpi\_watchdog.sh

sdpi\_watchdog\_run.sh

update\_rkn.sh

- Обновляет список заблокированных Роскомнадзором IP/хостов/URLов в базе.

- Требует запуска из директории, в которой располагается.

1. Конфигурация решения

Описание настроек файла sdpi.conf

NUMA

NUMA подразумевает архитектуру для работы с разделяемой памятью, в которой разные блоки ядер CPU, называемых нодами NUMA работают каждый со своей памятью и своими сетевыми интерфейсами, из расчета распределения нагрузки между ними.

Узнать количество используемых нод numa можно введя команду:

# numactl --hardware

available: 2 nodes (0-1)

node 0 cpus: 0 1 2 3 4 5 12 13 14 15 16 17

node 0 size: 7977 MB

node 0 free: 93 MB

node 1 cpus: 6 7 8 9 10 11 18 19 20 21 22 23

node 1 size: 8078 MB

node 1 free: 1929 MB

node distances:

node 0 1

0: 10 21

1: 21 10

```

\*\*main\_numa\_node\*\* - определяет основную ноду numa. Если нода одна или сетевые интерфейсы находятся на разных нодах - указываем значение 0, иначе указываем номер той ноды на которой находятся сетевые интерфейсы.

\*\*count\_numa\_nodes\*\* - определяет количество нод numa. Берем значение \*\*available\*\* выводимое командой \*\*numactl --hardware\*\*.

\*\*cores\_numa\_node0\*\* - номера используемых ядер на первой ноде, указанные вручную;

\*\*cores\_numa\_node1\*\* - номера используемых ядер на второй ноде, указанные вручную;

Также возможные значения для \*\*cores\_numa\_node0\*\* и \*\*cores\_numa\_node1\*\* - \*\*even\*\* и \*\*odd\*\* соответственно, означающие все четные и все нечетные ядра CPU.

\*\*working\_cores\*\* - определяет количество ядер, используемых под RSS-очереди.

По умолчанию или если значение параметра \*\*working\_cores\*\* равно 0, sDPI оставляет одно ядро под нужды ОС, одно ядро с основной ноды использует для обнаружения клиентов, одно ядро с основной ноды использует для очистки простаивающих сессий, одно ядро для обнаружения сетевых атак, оставшиеся ядра используются для RSS-очередей.

Значение параметра \*\*working\_cores\*\* не должно превышать общее количество ядер CPU.

Желательно чтобы количество свободных на каждой ноде ядер, было больше или равно кол-ву RSS-очередей работающих на тех же нодах сетевых интерфейсов для получения максимальной производительности.

Узнать количество RSS-очередей на отдельном сетевом интерфейсе можно введя команду:

cat /proc/interrupts|grep <имя интерфейса>

Пример:

# cat /proc/interrupts|grep eth2

58: 3973818 1561590 438857 38529 IR-PCI-MSI-edge eth2-TxRx-0

59: 3 0 2 0 IR-PCI-MSI-edge eth2

```

RSS-очереди(Receive Side Scaling Queues) - очереди сетевых интерфейсов, предназначенные для распределения нагрузки на множество ядер CPU при обработке сетевого трафика.

https://docs.microsoft.com/en-us/windows-hardware/drivers/network/introduction-to-receive-side-scaling

https://access.redhat.com/documentation/en-us/red\_hat\_enterprise\_linux/6/html/performance\_tuning\_guide/network-rss

### PF\_RING

\*\*max\_interface\_rss\_queues\*\* - максимальное количество используемых RSS-очередей на интерфейсах

\*\*network\_interfaces\*\* - список интерфейсов работающих со стороны Интернет

\*\*subscriber\_interfaces\*\* - список интерфейсов работающих со стороны клиента

### Database

\*\*pg\_conninfo\*\* - задает параметры подключения к базе данных, где:

\*\*host\*\* - имя хоста;

\*\*dbname\*\* - имя баз данных;

\*\*user\*\* - имя пользователя;

\*\*password\*\* - пароль.

\*\*db\_update\_period\*\* - период времени, через который перечитываются таблицы БД, в секундах.

### Collectors

\*\*collectors\*\* - список адресов коллекторов детальной статистики.

\*\*collectors\_billing\*\* - список адресов коллекторов статистики по биллингу.

\*\*sdpi\_ctl\_port\*\* - порт контроля sdpi.

\*\*billing\_stat\_timeout\*\* - таймаут отправки статистики по биллингу в секундах.

### Sessions

\*\*session\_timeout\*\* - таймаут сессий в секундах.

### Balancers

\*\*balancers\_network\_rx\_count\*\* - количество балансеров на прием трафика со стороны Интернет.

\*\*balancers\_network\_tx\_count\*\* - количество балансеров на отправку трафика со стороны Интернет.

\*\*balancers\_subscriber\_rx\_count\*\* - количество балансеров на прием трафика со стороны клиента.

\*\*balancers\_subscriber\_tx\_count\*\* - количество балансеров на отправку трафика со стороны клиента.

## Команды, применяемые без рестарта процесса

Выполнять команды нужно следующим образом:

# sdpi -C <command>

где вместо <command></command> одна из нижеперечисленных команд.

\*\*status\*\* - выводит статус sDPI

\*\*block rkn on\*\* - включает блокировку по спискам Роскомнадзора.

\*\*block rkn off\*\* - отключает блокировку по спискам Роскомнадзора.

\*\*block pkts send rst network on\*\* - включает отправку RST-пакета для блокировки соединения со стороны Интернет.

\*\*block pkts send rst network off\*\* - отключает отправку RST-пакета для блокировки соединения со стороны Интернет.

\*\*block pkts send rst subscriber on\*\* - включает отправку RST-пакета для блокировки соединения со стороны клиента.

\*\*block pkts send rst subscriber off\*\* - отключает отправку RST-пакета для блокировки соединения со стороны клиента.

\*\*redirects http pkts send rst network on\*\* - включает отправку RST-пакета при редиректе по http со стороны Интернет.

\*\*redirects http pkts send rst network off\*\* - отключает отправку RST-пакета при редиректе по http со стороны Интернет.

\*\*redirects http pkts send rst subscriber on\*\* - включает отправку RST-пакета при редиректе по http со стороны клиента.

\*\*redirects http pkts send rst subscriber off\*\* - отключает отправку RST-пакета при редиректе по http со стороны клиента.

\*\*redirects https pkts send rst network on\*\* - включает отправку RST-пакета при редиректе по https со стороны Интернет.

\*\*redirects https pkts send rst network off\*\* - отключает отправку RST-пакета при редиректе по https со стороны Интернет.

\*\*redirects https pkts send rst subscriber on\*\* - включает отправку RST-пакета при редиректе по https со стороны клиента.

\*\*redirects https pkts send rst subscriber off\*\* - отключает отправку RST-пакета при редиректе по https со стороны клиента.

\*\*redirects other pkts send rst network on\*\* - не используется.

\*\*redirects other pkts send rst network off\*\* - не используется.

\*\*redirects other pkts send rst subscriber on\*\* - не используется.

\*\*redirects other pkts send rst subscriber off\*\* - не используется.

\*\*print debug ipv4 on\*\* - включает вывод отладочной информации по IPv4.

\*\*print debug ipv4 off\*\* - отключает вывод отладочной информации по IPv4.

\*\*print debug ipv6 on\*\* - включает вывод отладочной информации по IPv6.

\*\*print debug ipv6 off\*\* - отключает вывод отладочной информации по IPv6.

\*\*print debug ip\*\* - выводит отладочные сообщения c указанным ip.

\*\*parse ipv6 on\*\* - включает обработку пакетов IPv6.

\*\*parse ipv6 off\*\* - отключает обработку пакетов IPv6.

\*\*parse traffic on\*\* - включает обработку трафика.

\*\*parse traffic off\*\* - отключает обработку трафика.

\*\*shaper on\*\* - включает шейпер, ограничивающий скорость прохождения трафика.

\*\*shaper off\*\* - отключает шейпер, ограничивающий скорость прохождения трафика.

\*\*push packets on\*\* - включает отправку пакетов из шейпера.

\*\*push packets off\*\* - отключает отправку пакетов из шейпера.

\*\*redirect on\*\* - включает редирект пакетов.

\*\*redirect off\*\* - отключает редирект пакетов.

\*\*block on\*\* - включает блокировку соединений.

\*\*block off\*\* - отключает блокировку соединений.

\*\*create new sessions on\*\* - включает создание новых сессий.

\*\*create new sessions off\*\* - отключает создание новых сессий.

\*\*attack detection on\*\* - включает обнаружение сетевых атак.

\*\*attack detection off\*\* - отключает обнаружение сетевых атак.

\*\*ad print attack on\*\* - включает вывод сообщений при обнаружении сетевых атак.

\*\*ad print attack off\*\* - отключает вывод сообщений при обнаружении сетевых атак.

\*\*ad dos append on\*\* - включает добавление адресов с которых идет DoS-атака в список атакующих.

\*\*ad dos append off\*\* - отключает добавление адресов с которых идет DoS-атака в список атакующих.

\*\*ad ddos append on\*\* - включает добавление адресов с которых идет DDoS-атака в список атакующих.

\*\*ad ddos append off\*\* - отключает добавление адресов с которых идет DDoS-атака в список атакующих.

\*\*ad ddos network on\*\* - включает проверку наличия DDoS-атак со стороны Интернет.

\*\*ad ddos network off\*\* - отключает проверку наличия DDoS-атак со стороны Интернет.

\*\*ad ddos subscriber on\*\* - включает проверку наличия DDoS-атак со стороны клиента.

\*\*ad ddos subscriber off\*\* - отключает проверку наличия DDoS-атак со стороны клиента.

\*\*session rate limit network\*\*

\*\*session rate limit subscriber\*\*

\*\*session del period network\*\*

\*\*session del period subscriber\*\*

\*\*addr sessions limit network\*\*

\*\*addr sessions limit subscriber\*\*

\*\*addr sessions period\*\*

\*\*send statistics on\*\* - включает отправку статистики.

\*\*send statistics off\*\* - отключает отправку статистики.

\*\*send statistics detail uninitialised tcp sessions in on\*\*

\*\*send statistics detail uninitialised tcp sessions in off\*\*

\*\*send statistics detail uninitialised udp sessions in on\*\*

\*\*send statistics detail uninitialised udp sessions in off\*\*

\*\*send statistics detail uninitialised other sessions in on\*\*

\*\*send statistics detail uninitialised other sessions in off\*\*

\*\*send statistics detail uninitialised tcp sessions out on\*\*

\*\*send statistics detail uninitialised tcp sessions out off\*\*

\*\*send statistics detail uninitialised udp sessions out on\*\*

\*\*send statistics detail uninitialised udp sessions out off\*\*

\*\*send statistics detail uninitialised other sessions out on\*\*

\*\*send statistics detail uninitialised other sessions out off\*\*

\*\*send statistics billing uninitialised tcp sessions in on\*\*

\*\*send statistics billing uninitialised tcp sessions in off\*\*

\*\*send statistics billing uninitialised udp sessions in on\*\*

\*\*send statistics billing uninitialised udp sessions in off\*\*

\*\*send statistics billing uninitialised other sessions in on\*\*

\*\*send statistics billing uninitialised other sessions in off\*\*

\*\*send statistics billing uninitialised tcp sessions out on\*\*

\*\*send statistics billing uninitialised tcp sessions out off\*\*

\*\*send statistics billing uninitialised udp sessions out on\*\*

\*\*send statistics billing uninitialised udp sessions out off\*\*

\*\*send statistics billing uninitialised other sessions out on\*\*

\*\*send statistics billing uninitialised other sessions out off\*\*

\*\*detect protocol on\*\* - включает определение протокола.

\*\*detect protocol off\*\* - отключает определение протокола.

\*\*print pd debug on\*\* - включает вывод отладочных сообщений при определении протокола.

\*\*print pd debug off\*\* - отключает вывод отладочных сообщений при определении протокола.

\*\*protocol detection guess on\*\*

\*\*protocol detection guess off\*\*

\*\*detect icmp by iph protocol on\*\* - включает определение ICMP-протокола через заголовок ip.

\*\*detect icmp by iph protocol off\*\* - отключает определение ICMP-протокола через заголовок ip.

\*\*protocol detection parse https on\*\* - включает разбор пакетов с целью получения хоста, если протокол определен как https.

\*\*protocol detection parse https off\*\* - отключает разбор пакетов с целью получения хоста, если протокол определен как https.

\*\*protocol detection exclude ports on\*\*

\*\*protocol detection exclude ports off\*\*

\*\*session idle time\*\* - время простоя сессии в секундах.

\*\*session timeout ip protocols on\*\*

\*\*session timeout ip protocols off\*\*

\*\*tcp session timeout limit\*\*

\*\*tcp session closing timeout\*\*

\*\*tcp session rst timeout\*\*

\*\*tcp session fin timeout\*\*

\*\*tcp session syn timeout\*\*

\*\*tcp session closed timeout\*\*

\*\*udp session timeout\*\*

\*\*udp session timeout limit\*\*

\*\*udp session dns timeout\*\*

\*\*udp session bittorent timeout\*\*

\*\*other session timeout\*\*

\*\*other session timeout limit\*\*

\*\*icmp session timeout\*\*

\*\*statistics tcp idle timeout\*\*

\*\*statistics udp idle timeout\*\*

\*\*statistics icmp idle timeout\*\*

\*\*statistics other idle timeout\*\*

\*\*statistics timeout limit\*\*

\*\*statistics exclude protocol add\*\*

\*\*statistics exclude tc add\*\*

\*\*statistics exclude protocol rm\*\*

\*\*statistics exclude tc rm\*\*

\*\*ts nsleep fct con\*\*

\*\*ts nsleep fct loop\*\*

\*\*ts nsleep ad con\*\*

\*\*ts nsleep ad loop\*\*

\*\*traffic dump on\*\* - включает отправку дампа в формате libpcap

\*\*traffic dump off\*\* - отключает отправку дампа в формате libpcap

\*\*traffic dump open interface\*\* - задает с какого интерфейса нужно собирать дамп

\*\*traffic dump interface\*\*

\*\*traffic dump all on\*\* - включает отправку дампа всего

\*\*traffic dump all off\*\* - отключает отправку дампа всего

\*\*traffic dump unknown on\*\* - включает отправку дампа для неизвестного трафика

\*\*traffic dump unknown off\*\* - отключает отправку дампа для неизвестного трафика

\*\*traffic dump no limits on\*\* - включает отправку дампа по безлимиту

\*\*traffic dump no limits off\*\* - отключает отправку дампа по безлимиту

\*\*traffic dump block on\*\* - включает отправку дампа по блокировкам

\*\*traffic dump block off\*\* - отключает отправку дампа по блокировкам

\*\*traffic dump redirect on\*\* - включает отправку дампа по редиректам.

\*\*traffic dump redirect off\*\* - отключает отправку дампа по редиректам.

\*\*traffic dump speed limit on\*\* - включает отправку дампа для трафика с ограничением скорости

\*\*traffic dump speed limit off\*\* - отключает отправку дампа для трафика с ограничением скорости

\*\*traffic dump soft block on\*\* -

\*\*traffic dump soft block off\*\*

\*\*traffic dump unrecognized on\*\*

\*\*traffic dump unrecognized off\*\*

\*\*traffic dump ips on\*\* - включает отправку дампа по ip.

\*\*traffic dump ips off\*\* - отключает отправку дампа по ip.

\*\*traffic dump src ips on\*\* - включает отправку дампа по ip источника

\*\*traffic dump src ips off\*\* - отключает отправку дампа по ip источника

\*\*traffic dump dst ips on\*\* - включает отправку дампа по ip назначения

\*\*traffic dump dst ips off\*\* - отключает отправку дампа по ip назначения

\*\*traffic dump ports on\*\* - включает отправку дампа по портам

\*\*traffic dump ports off\*\* - отключает отправку дампа по портам

\*\*traffic dump src ports on\*\* - включает отправку дампа по портам источника

\*\*traffic dump src ports off\*\* - отключает отправку дампа по портам источника

\*\*traffic dump dst ports on\*\* - включает отправку дампа по портам назначения

\*\*traffic dump dst ports off\*\* - отключает отправку дампа по портам назначения

\*\*traffic dump protocols on\*\* - включает отправку дампа по протоколам

\*\*traffic dump protocols off\*\* - отключает отправку дампа по протоколам

\*\*package check limit network\*\*

\*\*package check limit subscriber\*\*

\*\*client change detect on\*\*

\*\*client change detect off\*\*

\*\*client change debug on\*\*

\*\*client change debug off\*\*

\*\*client change full debug on\*\*

\*\*client change full debug off\*\*

\*\*client change check limit\*\*

\*\*tcp arq time limit\*\*

\*\*bypass off automatically on\*\* - включает автоматический перевод сетевых интерфейсов с bypass при старте sdpi.

\*\*bypass off automatically off\*\* - отключает автоматический перевод сетевых интерфейсов с bypass при старте sdpi.

\*\*bypass on automatically on\*\* - включает автоматический перевод сетевых интерфейсов на bypass при получении сигнала SIGTERM, SIGKILL или SIGSEGV

\*\*bypass on automatically off\*\* - отключает автоматический перевод сетевых интерфейсов на bypass при получении сигнала SIGTERM, SIGKILL или SIGSEGV

\*\*pass empty subscriber ip pkt dump on\*\* - включает запись дампа в формате lipcap при разборе пакетов с пустым клиентом

\*\*pass empty subscriber ip pkt dump off\*\* - отключает запись дампа в формате lipcap при разборе пакетов с пустым клиентом

\*\*save dump pcap on\*\* - включает запись дампа в формате libpcap при разборе пакетов

\*\*save dump pcap off\*\* - отключает запись дампа в формате libpcap при разборе пакетов

\*\*crash test\*\*

\*\*update db connection\*\* - переподключает к БД через период времени в секундах, задаваемый параметром \*\*db\_update\_period\*\* в конфиге \*\*sdpi.conf\*\*

\*\*reread collectors\*\* - перечитывает значения параметров collectors и collectors\_billing из конфигурации sdpi.conf.

\*\*help\*\*, \*\*h\*\* - вывод справки

1. В случае падения

Если процесс упал и не запустился сам, то, в первую очередь нужно отключить вочдоги и включить байпас.

Комментим в кронтабе \*\*sdpi\_watchdog\_run.sh\*\*

# crontab -e

А также, если в нем есть строчки с процессами \*\*update\_rkn.sh\*\*, \*\*asn\_updater.py\*\* и \*\*rkn\_list\_access\_check.sh\*\*, то эти строки тоже пока комментируем

Можно также выключить \*\*collectd\*\*, чтобы не засорял логи:

# /etc/init.d/collectd stop

Смотрим список процессов:

# ps aux | grep sdpi

Убиваем \*\*sdpi\_watchdog.sh\*\*

# kill -9 ....

Проверяем, что байпас включен:

```

# bpctl\_util all get\_bypass

Если не все \*\*on\*\*/\*\*slave\*\*, то включаем байпас:

# /home/bypass/bypass\_on.sh

Еще раз убеждаемся, что байпас включен, и проверяем, что на интерфейсах (кроме \*\*eth0\*\*/\*\*eth0.X\*\*) нет трафика:

# bpctl\_util all get\_bypass

# ifstat -b 2

Если в списке процессов есть \*\*sdpi\*\*, то пытаемся его завершить:

# sdpi -t

Завершаться может не моментально, в логах должно мелькнуть сообщение \*\*Stop sdpi version X.Y.ZZZ\*\*

Если не завершается, то убиваем процесс \*\*sdpi\*\*

# kill -9 ...

Еще раз убеждаемся, что процесcов, связанных с \*\*sdpi\*\* не осталось:

# ps aux | grep sdpi

Если остался процесс sdpi или какой-нибудь watchdog, \*\*ТО ДАЛЬШЕ НЕ ИДЕМ, ПОВТОРЯЕМ ВЫШЕОПИСАННЫЕ ПУНКТЫ!!!\*\*

Удаляем pid-файл и точку монтирования hugepages:

# rm /var/run/sdpi.pid

# rm /dev/hugepages/\\*

Перезагружаем драйвера PF\_RING:

# /etc/init.d/pf\_ring start

Иногда, при внезапном падении процесса сетевые интерфейсы остаются "в использовании", и pf\_ring и драйвера не могут нормально перезапуститься. В таком случае может наблюдаться плавно растущее число ошибок на каком-нибудь интерфейсе. Если так случилось, то нужно повторить всю процедуру перезапуска, и в этом месте вручную потушить все используемые sDPI интерфейсы, например

# ifconfig eth4 down

После этого пробуем запустить sdpi:

# sdpi -d

Для Асбеста нужно добавить дополнительные аргументы:

# sdpi -d -b on -b off

Запуск может идти долго, в логах будут мелькать сообщения вида \*\*Opening interface subscriber RX zc:ethX@Y\*\*. Отслеживаем наличие процесса sdpi.

```# ps aux | grep sdpi

Если процесс не может запуститься и падает, то зовем технического специалиста. Если в логе отобразилось сообщение вида \*\*Connecting to billing collector tcp://192.168.100.23:5557...\*\*, то процесс запустился и готов обрабатывать траффик.

Вырубаем байпас:

# /home/bypass/bypass\_off.sh

Проверяем, что байпас выключен и что на интерфейсах появился трафик:

# bpctl\_util all get\_bypass

# ifstat -b 2

Запускаем вочдог, раскомментировав строку с \*\*sdpi\_watchodg\_run.sh\*\* в кронтабе:

# crontab -e

А также, если в нем закомментированы строчки с процессами \*\*update\_rkn.sh\*\*, \*\*asn\_updater.py\*\* и \*\*rkn\_list\_access\_check.sh\*\*, то эти строки тоже раскомментируем

Если выключали \*\*collectd\*\*, не забываем его включить:

# /etc/init.d/collectd start

Готово, мониторим трафик на интерфейсах и логи.

На всякий случай, сохраняем лог для разработчиков:

# cp /var/log/syslog ...

&nbsp;

[sdpi\_restart.txt](resources/cd6d4a44cad440cda20f3b543b08e73f.txt)

9. При возникновении вопросов по установке ПО «sDPI»обращайтесь к техническим специалистам:

Яковлев Артем +7 (343) 287-15-34