

# Содержание

<b>Описание статистики NAT</b> .....	3
<b>Вывод статистики в <i>fastdpi_stat.log</i></b> .....	3
Общее для всех профилей .....	3
Профиль CG-NAT .....	3
Информация по преобразователю данных .....	4
Профиль 1:1 .....	5
Вывод статистики по белым адресам .....	5
<b>Легенда команды просмотра статистики NAT профиля через <i>fdpi_ctrl</i></b> .....	7
Легенда команды просмотра статистики dump NAT профиля .....	9
<b>Мониторинг свободных/занятых портов на белых адресах</b> .....	10



# Описание статистики NAT

Вывод статистики NAT возможен в stat\_log и через fdpi\_ctrl.

Вывод информации по очередям возможно сделать dump NAT профиля через fdpi\_cli.

## Вывод статистики в fastdpi\_stat.log

Задается с помощью параметра dbg\_log\_mask в /etc/dpi/fastdpi.conf.

Для применения требуется релоад сервиса: service fastdpi reload **Пример:**

```
dbg_log_mask=0x2000000
```

0x40000 - вывод в alert лог детализации инициализации NAT (профили, белые адреса и прочее)

0x100000 - вывод статистики по блокам белых адресов (если задано 0x2000000)

0x2000000 - вывод статистики по NAT.

## Общее для всех профилей

```
[STAT  ][2021/12/22-17:33:17:513859] NAT statistics : itrnsl=0, iprof=2,
profile 'cgnat', ntype=0, ref_cnt=1, cidr=94.140.198.84/30
[STAT  ][2021/12/22-17:33:17:513869] NAT statistics : itrnsl=1, iprof=3,
profile 'nat1_1', ntype=1, ref_cnt=1, cidr=16.35.121.0/24
```

itrnsl - внутренний индекс преобразователя серый <--> белый профиля.

iprof - внутренний индекс профиля

profile - имя профиля

ntype - тип профиля (0 - cgnat, 1 - 1:1)

ref\_cnt - счетчик ссылок использования преобразователя профилями

(Профили могут использовать один набор бесклассовых IP сетей, но разное количество ограничений на сессии)

cidr - список бесклассовых белых адресов профиля

## Профиль CG-NAT

```
[STAT  ][2021/12/22-17:33:19:252622] NAT statistics : itrnsl=0, iprof=2,
profile 'cgnat', ntype=0, ref_cnt=1, cidr=94.140.198.84/30
k=0, itrnsld=0, cidr=94.140.198.84/30
    total TCP : 30/20/0/7/17/ 0/0/0 50/20/0/50/0 5516/8/121
    actual TCP : 0/0/0/0/0 0/0/0 0/0/0/0/0 0/0/0
    total UDP : 13/4/0/13/1/ 0/0/0 17/4/11/17/0 28/1/3
    actual UDP : 0/0/0/0/0 0/0/0 0/0/0/0/0 0/0/0
    total GRE : 0/0
```

## Информация по преобразователю данных

```
k=0, itrnsld=0, cidr=94.140.198.84/30
k      - номер itrnsld
itrnsld - внутренний индекс данных преобразователя - того, кто обслуживает
бесклассовые IP сети
cidr   - конкретная бесклассовая IP сеть

total  - суммарная статистика
actual - статистика изменений счетчиков за период вывода статистики (параметр
delta_alarm, default 15 секунд)

total TCP : 30/20/0/7/17 0/0/0 50/20/0/50/0 5516/8/121
```

### Четыре группы:

1 группа - операции с портами белых адресов

30/20/0/7/17:

30 - выделение нового белого порта  
20 - повторное использование белого порта  
0 - ошибки получения нового белого порта  
7 - выполняет декремент количества сессий абонента по освобождению flow  
17 - выполняет декремент количества сессий абонента по повторному использованию белого порта

2 группа - общая статистика

0/0/0:

0 - рассчитали CRC по IP при обращении для выделения белого адреса. Должно быть == 0  
0 - превышение количества сессий для абонентов  
0 - разные белые адреса в flow и преобразователях - Должно быть == 0

3 группа - статистика по кэшу *серый* -> *белый*

50/20/0/50/0:

50 - добавлено записей в кэш  
20 - удалено записей из кэша  
0 - найдено записей в кэше при выделении нового белого порта  
50 - не найдено по серому белый адрес  
0 - ошибки добавления в кэш

4 группа - статистика преобразования *белый* -> *серый* (*inet*->*subs*)

5516/8/121:

5516 - успешно проведена трансляция белый --> серый  
8 - порт не попадает в выделенный диапазон белых портов

121 - не найдена трансляция белый --> серый

Для TCP/UDP и total/actual статистика одинаковая.

Для GRE - это GRE по умолчанию (когда не смогли в туннельном протоколе "точка-точка" найти сессию). На белый адрес может быть создана только одна такая сессия.

```
total GRE : 0/0
0 - использован адрес
0 - количество попыток создания сессий на уже занятый белый адрес
```

## Профиль 1:1

```
[STAT    ][2021/12/22-17:17:28:749622] NAT statistics : itrnsl=1, iprof=3,
profile 'nat1_1', ntype=1, ref_cnt=1, cidr=16.35.121.0/24
      k=0, itrnsld=1, cidr=16.35.121.0/24
      total 256/256/0/0/0/0 0/0
```

Статистика 2 группы:

1 группа - 256/256/0/0/0/0

2 группа - 0/0

### Пример:

dbg\_log\_mask=0x2100000

## Вывод статистики по белым адресам

```
[STAT    ][2021/12/22-21:14:48:385991] NAT statistics : itrnsl=0, iprof=2,
profile 'cgnat', ntype=0, ref_cnt=1, cidr=94.140.198.84/30
      k=0, itrnsld=0, cidr=94.140.198.84/30
      total TCP : 26/4/0/4/2/ 0/0/0 30/4/0/30/0 3045/1/36
      actual TCP : 0/0/0/0/0 0/0/0 0/0/0/0/0 0/0/0
      TCP whiteblk ip_mask=0x0, nwhaddr=2
      whip=94.140.198.84 : sb=64, lsb=64, nb=1008,
whpa=64512, whpb=0, whpf=64512, awhb=4, fwhb=1004, puwhb=0.40%
      thr=0, ublock=1, uport=0
      thr=1, ublock=1, uport=0
      thr=2, ublock=1, uport=0
      thr=3, ublock=1, uport=0
      whip=94.140.198.86 : sb=64, lsb=64, nb=1008,
whpa=64512, whpb=26, whpf=64486, awhb=4, fwhb=1004, puwhb=0.40%
      thr=0, ublock=1, uport=0
      thr=1, ublock=1, uport=0
      thr=2, ublock=1, uport=13
      thr=3, ublock=1, uport=13
```

TCP whiteblk ip\_mask=0x0, nwhaddr=2 :

ip\_mask - маска адресов  
nwhaddr - количество белых адресов, которые попадают под маску

whip=94.140.198.84 : sb=64 ( 64 ), nb=1008, whpa=64512, whpb=0, whpf=64512, awhb=4, fwhb=1004, puwhb=0.40%

whip=94.140.198.84 - белый адрес  
sb=64 - размер блока портов  
lsb=64 - размер последнего блока  
nb=1008 - количество блоков портов  
whpa=64512 - всего портов  
whpb=0 - занято портов  
whpf=64512 - свободно портов  
awhb=4 - выдано блоков  
fwhb=1004 - свободно блоков  
puwhb=0.40% - процент занятости блоков

### Добавлено в версии 12.1.0

whp\_salfs - сколько портов в 'короткой' очереди  
whp\_lalfs - сколько портов в 'длинной' очереди  
whp\_ruse - сколько портов может быть использовано повторно  
whp\_ruse\_salfs - сколько портов может быть использовано повторно из 'короткой' очереди  
whp\_ruse\_lalfs - сколько портов может быть использовано повторно из 'длинной' очереди  
whp\_dthr - сколько портов созданы в одном рабочем потоке, но использовались в другом  
whp\_dthr\_salfs - сколько портов созданы в одном рабочем потоке, но использовались в другом из 'короткой' очереди  
whp\_dthr\_lalfs - сколько портов созданы в одном рабочем потоке, но использовались в другом из 'длинной' очереди

В рамках белого адреса видно распределение захваченных портов/блоков по рабочим потокам

thr=0, ublock=1, uport=0  
thr=0 - номер рабочего потока  
ublock=1 - использовано блоков белых портов  
uport=0 - использовано блоков белых портов

thr\_salfs - сколько портов в 'короткой' очереди  
thr\_lalfs - сколько портов в 'длинной' очереди  
thr\_ruse - сколько портов может быть использовано повторно  
thr\_ruse\_salfs - сколько портов может быть использовано повторно из 'короткой' очереди  
thr\_ruse\_lalfs - сколько портов может быть использовано повторно из 'длинной' очереди  
thr\_dthr - сколько портов созданы в одном рабочем потоке, но использовались в другом

thr\_dthr\_salfs - сколько портов созданы в одном рабочем потоке, но использовались в другом из 'короткой' очереди  
thr\_dthr\_lalfs - сколько портов созданы в одном рабочем потоке, но использовались в другом из 'длинной' очереди

Формат вывода сохранился

```
fdpi_ctrl list status --service 11 --ip 192.168.4.20
Autodetected fastdpi params : dev='em1', port=29001
connecting 94.140.198.68:29001 ...

=====
192.168.4.20    crcip=0xd649d853          ntype=0          profile='cgnat'
itrnsl=0 itrnsl=0          whiteip=94.140.198.86  sess_tcp=127
sess_udp=108   indmtd=4
```

**Вывод:**

```
192.168.4.20    - серый IP
crcip=0xd649d853 - CRC серого IP
ntype=0         - тип NAT: 0 - cgnat, 1 - 1:1
profile='cgnat' - имя профиля
itrnsl=0       - внутренний индекс преобразователя серый <--> белый профиля.
itrnsl=0       - внутренний индекс данных преобразователя
whiteip=94.140.198.86 - белый адрес
sess_tcp=127   - количество сессий TCP
sess_udp=108   - количество сессий UDP
indmtd=4       - внутренний индекс данных абонента (метаданных абонента)

fdpi_ctrl list status --service 11 --ip 192.168.4.20 --outformat=json
fdpi_ctrl list status --service 11 --ip 192.168.4.20 --outformat=json | jq
.

fdpi_ctrl list all status --service 11
fdpi_ctrl list all status --service 11 --outformat=json
```

Формат аналогично

## Легенда команды просмотра статистики NAT профиля через fdpi\_ctrl

Команда:

```
fdpi_ctrl list status --service 11 --profile.name cgnat
```

Вывод:

```
ntype=0          profile='test_nat_cgnat'          itrnsl=0          itrnsl=1
```

```

        itrnsld=0          cidr=94.140.198.84/30
                        proto=TCP          ip_mask=0x0          nwhaddr=2
                                proto=TCP          ip_mask=0x0          whip=94.140.198.84
sb=64   lsb=64   nb=1008 whpa=64512          whpb=0   whpf=64512          awhb=4
fwhb=1004          puwhb=0.40%          whp_salfs=0          whp_lalfs=0          whp_ruse=0
whp_ruse_salfs=0          whp_ruse_lalfs=0          whp_dthr=0
whp_dthr_salfs=0          whp_dthr_lalfs=0
                                nthr=0   ublock=1          uport=0 thr_salfs=0
thr_lalfs=0          thr_ruse=0          thr_ruse_salfs=0          thr_ruse_lalfs=0
thr_dthr=0          thr_dthr_salfs=0          thr_dthr_lalfs=0
                                nthr=1   ublock=1          uport=0 thr_salfs=0
thr_lalfs=0          thr_ruse=0          thr_ruse_salfs=0          thr_ruse_lalfs=0
thr_dthr=0          thr_dthr_salfs=0          thr_dthr_lalfs=0

```

Легенда:

```

ntttype          - тип профиля (0 - cgnat, 1 - 1:1)
profile          - имя профиля
itrnsld          - внутренний индекс преобразователя серый<-->белый профиля
nitrnsld          - количество данных преобразователей профиля (количество
бесклассовых IP сетей)
itrnsld          - внутренний индекс данных преобразователя - того, кто обслуживает
бесклассовые IP сети
cidr              - конкретная бесклассовая IP сеть
proto            - протокол TCP/UDP
ip_mask          - маска адресов
nwhaddr          - количество белых адресов, которые попадают под маску или под CRC
(зависит от параметра rx_dispatcher)
whip            - белый адрес
sb              - размер блока выделяемых портов.
lsb             - размер последнего блока
nb              - количество блоков портов
whpa            - всего портов
whpb            - занято портов
whpf            - свободно портов
awhb            - выдано блоков
fwhb            - свободно блоков
puwhb           - процент занятости блоков
whp_salfs       - находится в 'короткой' очереди
whp_lalfs       - находится в 'длинной' очереди
whp_ruse        - может быть использовано
whp_ruse_salfs  - может быть использовано в 'короткой' очереди
whp_ruse_lalfs  - может быть использовано в 'длинной' очереди
whp_dthr        - количество элементов ithr_owner != ithr по очередям
whp_dthr_salfs  - количество элементов ithr_owner != ithr по 'short' очереди
whp_dthr_lalfs  - количество элементов ithr_owner != ithr по 'long' очереди

```



Описание параметра `rx_dispatcher` [по ссылке](#)

## Легенда команды просмотра статистики dump NAT профиля

Команда:

```
fdpi_cli nat dump whaddr queue test_nat_cgnat
```

Вывод:

```
profile='test_nat_cgnat' itrnsl=0
  cidr='94.140.198.84/30' itrnsld=0
    whip=94.140.198.86
      proto=TCP
        entryp :
          ithr=0, ihead=0, itail=0
          ithr=1, ihead=0, itail=0
          ithr=2, ihead=133, itail=265
          ithr=3, ihead=193, itail=327
        data :
          sind=129, inext=257, iprev=258,
whport=1152, graddr=192.168.4.20:60637 tml='2023/03/06 16:28:09,
-00:00:10.657 (7472516905147512 ticks)', lifetime=120, canreuse=0, ialf=1,
imtd=516, iown=2, ilst=2, subproto=0, decr_sess=0, ind_gcache_slice=1,
igcache=40
          sind=130, inext=151, iprev=148,
whport=1153, graddr=192.168.4.20:52553 tml='2023/03/06 16:27:50,
-00:00:29.455 (7472459405058624 ticks)', lifetime=30, canreuse=0, ialf=0,
imtd=516, iown=2, ilst=2, subproto=0, decr_sess=0, ind_gcache_slice=1,
igcache=1
```

Легенда:

profile	- имя профиля
itrnsl	- внутренний индекс преобразователя серый <--> белый профиля
cidr	- конкретная бесклассовая IP сеть
itrnsld	- внутренний индекс данных преобразователя - того, кто обслуживает бесклассовую IP сеть
whip	- белый адрес
proto	- протокол TCP/UDP

Реализация очереди используемых портов для белых адресов использует один массив - назовем его WHP, размером 0xffff. На его основе построен список используемых портов для рабочего потока. Индекс 0 - используется как заглушка (пусто).

Очереди потоков нельзя вывести в виде списка, так как в процессе происходит перемещение записей в очереди, из-за чего вывод может зациклиться. Поэтому массив WHP выводится as is для занятых записей.

```
entryp : задает точки входа в список белых портов рабочего потока
```

ithr - номер рабочего потока  
ihead - вершина списка  
itail - последний элемент списка

data : данные массива белых портов WHP (выводятся только занятые записи)

sind - индекс записи  
inext - индекс след. записи  
iprev - индекс пред. записи  
whport - белый порт  
graddr - серый адрес, которому назначен белый адрес  
tml - время последнего обращения к записи  
lifetime - тайм-аут в секундах время жизни записи (зависит от параметров для short/long очереди)  
canreuse - признак, что запись может быть использована повторно  
ialf - номер очереди обработки:  
en\_nalfs\_shrt = 0, # очередь с коротким временем жизни  
en\_nalfs\_long = 1, # долгоиграющая очередь  
indmtd - внутренний индекс данных абонента (метаданных абонента)  
iown - поток-владелец, который породил запись.  
ilst - номер потока, который последний обращался к записи  
subproto - для какого протокола выделена запись из UDP  
typedef enum en\_nat\_borrw\_udp: u\_int8\_t  
{  
    ennatborwu\_ORG = 0, # UDP/TCP  
    ennatborwu\_DFLTGRE = 1, # общий GRE  
    ennatborwu\_MAX = 2, # ICMP  
} en\_nat\_borrw\_udp\_t;  
decr\_sess - признак, что произошел декремент счетчика использования порта на сером адресе.  
ind\_gcache\_slice - индекс кэше-slice перекодировки 'серый --> белый'  
igcache - индекс в соответствующем кэше-slice перекодировки 'серый -> белый'

## Мониторинг свободных/занятых портов на белых адресах

Превентивный мониторинг позволит избежать проблем нехватки свободных портов и соответственно невозможности создания новых сессий, выделив дополнительные блоки белых адресов или уменьшив лимиты выделения портов при исчерпании ресурсов в текущем пуле.

**свободно портов на белом адресе** = whpf (не распределены) + whp\_ruse (готовы к переиспользованию)

**занято портов на белом адресе** = whph (распределены) - whp\_ruse (готовы к переиспользованию)