ОБНОВЛЕНО 16 МАР

Сценарии применения

У Досгейта есть множество разных сценариев применения.

Он может быть использован как:

- пограничное устройство сети в виде пакетного фильтра
- stateless firewall
- полноценное standalone решение для защиты от ДДоС-атак.

Кроме этого, Досгейт может быть использован и напрямую интегрирован для разгрузки различных DPI, WAF, bot management, или NGFW систем.

Всё это доступно за счет **единого конструктора правил**, с помощью которого администратор системы сам выбирает какие функции и инструменты он комбинирует между собой для достижения той или иной цели.

В данной документации приведены примеры, как Досгейт может быть использован в разных случаях.

Incorrect TCP flags

Досгейт позволяет администратору системы завести набор правил, по которому можно сбрасывать некорректные TCP-флаги. Это может быть полезно как в случае эксплуатации Досгейта как Анти-ДДоС решения, так и пакетного файрволла

Данный набор правил рекомендуется для применения в внешних TCP-сервисах, например, на игровых серверах которые используют протокол TCP, а также веб-приложениях, которые защищаются только с использованием dosgate без WAF, bot management систем, и др. ПО которое обрабатывает HTTP/HTTPs трафик на прикладном уровне

Набор правил

Для каждого правила написан комментарий статистики для корректной визуализации в Collectd.

```
# В примере используется arena first и профиль test
# Создание цепочки правил (chain)
dgctl -u chain://first/test -c insert tcp_flags_chain Chain to drop TCP packets
with malicious flags

# Наполнение цепочки правил
dgctl -u chain://first/test/tcp_flags_chain -c insert -- -m protocol tcp -m
tcpflags fin,syn/fin,sin -j STATS INVALID_TCP_FLAG -j DROP
dgctl -u chain://first/test/tcp_flags_chain -c insert -- -m protocol tcp -m
tcpflags syn,rst/syn,rst -j STATS INVALID_TCP_FLAG -j DROP
dgctl -u chain://first/test/tcp_flags_chain -c insert -- -m protocol tcp -m
tcpflags fin,rst/fin,rst -j STATS INVALID_TCP_FLAG -j DROP
dgctl -u chain://first/test/tcp_flags_chain -c insert -- -m protocol tcp -m
tcpflags fin/fin,ack -j STATS INVALID_TCP_FLAG -j DROP
dgctl -u chain://first/test/tcp_flags_chain -c insert -- -m protocol tcp -m
tcpflags fin,psh,urg/fin,psh,urg -j STATS INVALID_TCP_FLAG -j DROP
dgctl -u chain://first/test/tcp_flags_chain -c insert -- -m protocol tcp -m
tcpflags /sin,rst,ack -j STATS INVALID_TCP_FLAG -j DROP
dgctl -u chain://first/test/tcp_flags_chain -c insert -- -m protocol tcp -m
tcpflags /syn,fin,rst,ack,psh,urg,ece,ecr -j STATS INVALID_TCP_FLAG -j DROP
dgctl -u chain://first/test/tcp_flags_chain -c insert -- -m protocol tcp -m
tcpflags syn ! -m tcpmss 536-65535 -j STATS SUSPICIOS_MSS
dgctl -u chain://first/test/tcp_flags_chain -c insert -- -m protocol tcp -m
sport 21,80,443 -m dport 80,443 -j STATS WEB_INVALID -j DROP

# Добавление правила в профиль защиты об отправке всех TCP пакетов на обработку
через ранее созданную цепочку
dgctl -u profile://first/test -c insert -- -m protocol tcp -j GOTO
tcp_flags_chain
```

Anti-DDoS

В качестве примера мы используем следующие вводные данные: Досгейт является устройством защиты внешнего периметра сети, на котором расположена 1 подсеть /24.

Основное преимущество Досгейта - возможность детализированного описания сервисов, как их легитимных, так и потенциальных вредоносных параметров, по которым можно сбрасывать атакующие пакеты, без привязанности к каким-то конкретным контрмерам и ограниченным настройкам.

Досгейт работает как единый конструктор правил, с помощью которого можно быстро и легко описать существующие сервисы, комбинировать наборы данных и функции между собой для создания эффективной ДДоС-защиты.

Правила фильтрации, созданные в данном документе являются примером использования существующих инструментов, а не регламентируют конкретный тип или

порядок их использования.

В рамках сети работают как UDP, так и TCP сервисы. В рамках карты сервисов мы знаем, что:

- На IP-адресе .251 на порту TCP 8443 работает VPN, на порту TCP 22 работает SSH
- На IP-адресе .252 на порту TCP 443 работает веб-приложение с использованием HTTPs
- На IP-адресе .253 на порту UDP 53 и TCP 53 работает DNS-сервер
- На IP-адресах .251, .252, .253 - не живут никакие сервисы кроме описанных и они полностью под них зарезервированы

Помимо этих данных, мы знаем, что на остальных 252 адресах также время от времени появляются и удаляются внешние сервисы, которые тоже могут стать целью ДДоС-атаки, и мы должны защищать их в том числе.

В нашем примере arena имеет название `first`

Создание профиля защиты

```
dgctl -u arena://first -c add global all_services
```

Аллокация IP-адресов получателей к профилям

```
dgctl -u router://first/global -c insert -- x.x.x.0/24
```

Общие правила фильтрации

В рамках общих правил фильтрации мы сбрасываем самые популярные атаки, которые являются едиными для всех сервисов: VPN, WEB, DNS, SSH, и др.

Блокировка амплификаций

Сброс пакетов с вредоносных портов отправителя протокола UDP

```
dgctl -u profile://first/global -c insert -- -m protocol udp -m sport
19,6881,389,751,11211,1434,5353,137,111,17,27960,520,1900,27015,7001,3283,5683,37
```

```
-j STATS amplifications -j DROP
dgctl -u profile://first/global -c insert -- -m protocol udp -m sport 123 ! -m
len what net:packet len 76 -j STATS amplifications -j DROP
dgctl -u profile://first/global -c insert -- -m protocol udp -m dport 53 -m len
what net:packet len 1000-1500 -j STATS amplifications -j DROP
dgctl -u profile://first/global -c insert -- -m protocol udp -m sport 53 -m len
what net:packet len 1000-1500 -j STATS dns_flood -j DROP
```

Сброс всех протоколов кроме часто-используемых

В этом примере: TCP, UDP, ICMP, ESP, GRE, IPIP, AH

```
dgctl -u profile://first/global -c insert -- -m protocol udp,tcp,icmp -j MARK
how set value 2
dgctl -u profile://first/global -c insert -- -m protocol ah,esp,ipip,gre -j
MARK how set value 2
dgctl -u profile://first/global -c insert -- ! -m mark 2 -j STATS
incorrect_protocol -j DROP
```

Создание отдельных цепочек правил

Для каждого сервиса - своя индивидуальная цепочка правил, в которую пакет переходит из основного профиля

```
dgctl -u chain://first/global -c insert -- DNS
dgctl -u chain://first/global -c insert -- WEB
dgctl -u chain://first/global -c insert -- VPN_and_ssh
dgctl -u chain://first/global -c insert -- other
```

Маршрутизация сервисов

В зависимости от IP-получателя, пакет отправляется в специфичную для этого сервиса цепочку правил

```
dgctl -u profile://first/global -c insert -- ! -m dst
x.x.x.251,x.x.x.252,x.x.x.253 -j GOTO other
dgctl -u profile://first/global -c insert -- -m dst x.x.x.251 -j GOTO
VPN_and_ssh
dgctl -u profile://first/global -c insert -- -m dst x.x.x.252 -j GOTO WEB
dgctl -u profile://first/global -c insert -- -m dst x.x.x.253 -j GOTO DNS
```

Настройка цепочки правил other

Активация защиты по триггеру

В случае, если общий трафик, проходящий через это правило, составляет меньше 500 MBps, трафик пропускается сразу конечному получателю (bypass). Когда трафик становится выше 500 MBps - пакеты начинают проходить через остальные правила в профиле защиты (и в соединенных с профилем цепочках). В случае, если трафик стал ниже порога в 500 MBps - чтобы триггер перестал быть активным должно пройти еще 60 секунд (cooldown, для pulse wave атак)

```
dgctl -u chain://first/global/other -c insert -- -j RATELIMIT 2 key "" cooldown
60 bps 500m
dgctl -u chain://first/global/other -c insert -- -m verdict ratelimit conform -
j ACCEPT
```

TCP-авторизация

Методом RST для защиты от IP-spoofing атак, авторизация IP отправителя на 3600 секунд. TCP-авторизация

```
dgctl -u chain://first/global/other -c insert -- ! -m hmark id 2 status valid -
j TCPAUTH id 1 type hs atype hs
dgctl -u chain://first/global/other -c insert -- -m verdict tcpauth valid -j
HMARK id 2 value 2 lifetime 3600
```

Защита от ботнетов

В случае, если 1 IP отправителя превышает 2000 пакетов в секунду, IP-адрес отправителя из пакета заносится в метку "hmark" ID 1 (быструю таблицу данных) и хранится там установленный lifetime (в нашем примере, 600 секунд). Правило по сбросу

IP-адресов которые были заблокированы устанавливается на первое место после триггера о том что идет атака для ускорения работы системы, атаки ботнетами могут превышать сотни ГБит/с.

```
dgctl -u chain://first/global/other -c insert -i 3 -- -m hmark id 1 status
valid -j STATS 2000pps -j DROP
dgctl -u chain://first/global/other -c insert -- -j RATELIMIT 1 key "l3_src"
pps 2000
dgctl -u chain://first/global/other -c insert -- -m verdict ratelimit exceed -j
HMARK id 1 value 1 lifetime 600
```

ACL для TCP

```
dgctl -u chain://first/global/other -c insert -- -m protocol tcp -m sport 0-
1023 -j STATS bad_ports_tcp -j DROP
```

Защита от SYN flood ботами

Блокировка IP отправителя на 600 секунд при превышении 50 PPS SYN)

```
dgctl -u chain://first/global/other -c insert -- -m tcpflags syn/syn -j
RATELIMIT 3 key "l3_src" pps 50
dgctl -u chain://first/global/other -c insert -- -m verdict ratelimit exceed -j
HMARK id 1 value 2 lifetime 600
```

Сброс IP-фрагментации в UDP

```
dgctl -u chain://first/global/other -c insert -- -m protocol udp -m frag -j
STATS fragmentation -j DROP
```

Сброс QUIC-flood атак

```
dgctl -u chain://first/global/other -c insert -- -m protocol udp -m sport
80,443 -m dport 80,443 -j DROP
```

Гео-шейпинг

Если может быть применим, для всех стран кроме России, Беларуси, Казахстана, Узбекистана, Кыргызстана, Грузии, Армении, и Азербайджана до 150 MBps

```
dgctl -u chain://first/global/other -c insert -- ! -m geoip cntr
RU,BY,KZ,UZ,KG,GE,AM,AZ -j RATELIMIT 4 key "" bps 150m
dgctl -u chain://first/global/other -c insert -- -m verdict ratelimit exceed -j
STATS geo_shaping -j DROP
```

Настройка цепочки правил DNS

ACL

Зная какой сервис расположен на этом адресе, мы заранее сбрасываем все проходимые пакеты не отправленные на порт получателя 53, а также те пакеты - которые не содержат DNS header в пакете.

```
dgctl -u chain://first/global/DNS -c insert -- ! -m dport 53 -j DROP
dgctl -u chain://first/global/DNS -c insert -- ! -m dns -j DROP
```

Активация защиты по триггеру

В случае если общий трафик проходимый через это правило составляет меньше 100 MBps, трафик пропускается сразу конечному получателю (bypass). Когда трафик становится выше 100 MBps - пакеты начинают проходить через остальные правила в профиле защиты (и в соединенных с профилем цепочках). В случае если трафик стал ниже порога в 100 MBps - чтобы триггер перестал быть активным должно пройти еще 60 секунд (cooldown, для pulse wave атак)

```
dgctl -u chain://first/global/DNS -c insert -- -j RATELIMIT 2 key "" cooldown
60 bps 100
dgctl -u chain://first/global/DNS -c insert -- -m verdict ratelimit conform -j
ACCEPT
```

TCP-авторизация

Методом RST для защиты от IP-spoofing атак, авторизация IP отправителя на 3600 секунд. TCP-авторизация

```
dgctl -u chain://first/global/DNS -c insert -- ! -m hmark id 2 status valid -j
TCPAUTH id 1 type hs atype hs
dgctl -u chain://first/global/DNS -c insert -- -m verdict tcpauth valid -j
HMARK id 2 value 2 lifetime 3600
```

Защита от ботнетов

В случае, если 1 IP отправителя превышает 2000 пакетов в секунду, IP-адрес отправителя из пакета заносится в метку "hmark" ID 1 (быструю таблицу данных) и хранится там установленный lifetime (в нашем примере, 600 секунд). Правило по сбросу IP-адресов которые были заблокированы устанавливается на первое место после триггера о том что идет атака для ускорения работы системы, атаки ботнетами могут превышать сотни ГБит/с.

```
dgctl -u chain://first/global/DNS -c insert -i 5 -- -m hmark id 1 status valid
-j STATS 2000pps -j DROP
dgctl -u chain://first/global/DNS -c insert -- -j RATELIMIT 1 key "l3_src" pps
2000
dgctl -u chain://first/global/DNS -c insert -- -m verdict ratelimit exceed -j
HMARK id 1 value 1 lifetime 600
```

DNS geo shaping для всех стран кроме России до 100 MBps

```
dgctl -u chain://first/global/DNS -c insert -- ! -m geoip cntr RU -j RATELIMIT
id 8 key "l3_dst" bps 100m
dgctl -u chain://first/global/DNS -c insert -- -m verdict ratelimit exceed -j
DROP
```

Настройка цепочки правил VPN_and_ssh

ACL

```
dgctl -u chain://first/global/VPN_and_ssh -c insert -- -m dport 8443,22 -m
protocol TCP -j MARK how set value 2
dgctl -u chain://first/global/VPN_and_ssh -c insert -- ! -m mark 2 -j DROP
```

Активация защиты по триггеру

В случае если общий трафик проходимый через это правило составляет меньше 100 MBps, трафик пропускается сразу конечному получателю (bypass). Когда трафик становится выше 100 MBps - пакеты начинают проходить через остальные правила в профиле защиты (и в соединенных с профилем цепочках). В случае если трафик стал ниже порога в 100 MBps - чтобы триггер перестал быть активным должно пройти еще 60 секунд (cooldown, для pulse wave атак)

```
dgctl -u chain://first/global/VPN_and_ssh -c insert -- -j RATELIMIT 2 key ""
cooldown 60 bps 100m
dgctl -u chain://first/global/VPN_and_ssh -c insert -- -m verdict ratelimit
conform -j ACCEPT
```

TCP-авторизация

Методом RST для защиты от IP-spoofing атак, авторизация IP отправителя на 3600 секунд. TCP-авторизация

```
dgctl -u chain://first/global/VPN_and_ssh -c insert -- ! -m hmark id 2 status
valid -j TCPAUTH id 1 type hs atype hs
dgctl -u chain://first/global/VPN_and_ssh -c insert -- -m verdict tcpauth valid
-j HMARK id 2 value 2 lifetime 3600
```

Защита от ботнетов

В случае, если 1 IP отправителя превышает 2000 пакетов в секунду, IP-адрес отправителя из пакета заносится в метку "hmark" ID 1 (быструю таблицу данных) и хранится там установленный lifetime (в нашем примере, 600 секунд). Правило по сбросу IP-адресов, которые были заблокированы, устанавливается на первое место после триггера о том, что идет атака для ускорения работы системы, атаки ботнетами могут превышать сотни ГБит/с.

```
dgctl -u chain://first/global/VPN_and_ssh -c insert -i 4 -- -m hmark id 1
status valid -j STATS 2000pps -j DROP
```

```
dgctl -u chain://first/global/VPN_and_ssh -c insert -- -j RATELIMIT 1 key
"l3_src" pps 2000
dgctl -u chain://first/global/VPN_and_ssh -c insert -- -m verdict ratelimit
exceed -j HMARK id 1 value 1 lifetime 600
```

GEO shaping для всех стран кроме России на 100 MBps на каждый IP-получателя

```
dgctl -u chain://first/global/VPN_and_ssh -c insert -- ! -m geoip cntr RU -j
RATELIMIT id 8 key "l3_dst" bps 100m
dgctl -u chain://first/global/VPN_and_ssh -c insert -- -m verdict ratelimit
exceed -j DROP
```

Настройка цепочки правил WEB

ACL

```
dgctl -u chain://first/global/WEB -c insert -- -m dport 443 -m protocol TCP -j
MARK how set value 3
dgctl -u chain://first/global/WEB -c insert -- ! -m mark 3 -j DROP
```

Активация защиты по триггеру

В случае если общий трафик проходимый через это правило составляет меньше 200 MBps, трафик пропускается сразу конечному получателю (bypass). Когда трафик становится выше 200 MBps - пакеты начинают проходить через остальные правила в профиле защиты (и в соединенных с профилем цепочках). В случае если трафик стал ниже порога в 200 MBps - чтобы триггер перестал быть активным должно пройти еще 60 секунд (cooldown, для pulse wave атак)

```
dgctl -u chain://first/global/WEB -c insert -- -j RATELIMIT 2 key "" cooldown
60 bps 200m
dgctl -u chain://first/global/WEB -c insert -- -m verdict ratelimit conform -j
ACCEPT
```

TCP-авторизация

Методом RST для защиты от IP-spoofing атак, авторизация IP отправителя на 3600 секунд. TCP-авторизация

```
dgctl -u chain://first/global/WEB -c insert -- ! -m hmark id 2 status valid -j
TCPAUTH id 1 type hs atype hs
dgctl -u chain://first/global/WEB -c insert -- -m verdict tcpauth valid -j
HMARK id 2 value 2 lifetime 3600
```

Защита от ботнетов

В случае, если 1 IP отправителя превышает 2000 пакетов в секунду, IP-адрес отправителя из пакета заносится в метку "hmark" ID 1 (быструю таблицу данных) и хранится там установленный lifetime (в нашем примере, 600 секунд). Правило по сбросу IP-адресов которые были заблокированы устанавливается на первое место после триггера о том что идет атака для ускорения работы системы, атаки ботнетами могут превышать сотни ГБит/с.

```
dgctl -u chain://first/global/WEB -c insert -i 4 -- -m hmark id 1 status valid
-j STATS 2000pps -j DROP
dgctl -u chain://first/global/WEB -c insert -- -j RATELIMIT 1 key "l3_src" pps
2000
dgctl -u chain://first/global/WEB -c insert -- -m verdict ratelimit exceed -j
HMARK id 1 value 1 lifetime 600
```

Stateless Firewall

В качестве примера мы используем следующие вводные данные: Досгейт является устройством защиты внешнего периметра сети, через которое ходит трафик. После этого трафик попадает внутрь сети. Досгейт используется как промежуточный пакетный фильтр, выполняющий, в том числе функции файрволла.

В данном файле можно найти примеры использования Досгейта, как пакетного фильтра, для решения задач межсетевого экрана, ACL-фильтра, белых и черных списков, контроля доступа

В нашем примере, arena имеет название `first`, профиль имеет название `firewall`

Защита SSH-сервера от перебора паролей (bruteforce)

Блокировка IP-отправителя который перебирает пароль к SSH-сервису работающему на порту TCP 22 на 600 секунд, в случае если было больше 10 попыток установить соединение за минуту

```
# Проверка на IP-spoofing с помощью TCP-авторизации, чтобы атакующий не мог заблокировать легитимные IP-адреса от доступа к SSH

dgctl -u profile://first/firewall -c insert -- ! -m hmark id 1 status valid -j TCPAUTH id 1 type hs atype hs
dgctl -u profile://first/firewall -c insert -- -m verdict tcpauth valid -j HMARK id 1 value 1 lifetime 3600

# Блокировка IP-отправителя в случае если совершено больше 10 попыток соединения (определяется по пакету с SYN флагом в сторону TCP порта получателя 22) за 60 секунд с момента последней попытки

dgctl -u profile://first/firewall -c insert -- -m sdhmark id 1 status matches value 10-999 -j STATS ssh_bruteforce -j DROP
dgctl -u profile://first/firewall -c insert -- -m protocol tcp -m dport 22 -m tcpflags syn/syn -j SDHMARK id 1 how add value 1 lifetime 60
```

Сброс чужих IP-подсетей (тех, какие не занесены в белый список) в случае обращения к системным портам на сети

В этом примере системным является порт 22 по протоколам TCP и UDP. Это может применяться также на целые устройства на сети, например, когда нужно заблокировать доступ к определенному IP-получателя от всех кроме указанных масок, или даже стран (используя `-m geoip`)

```
# Создание локального префикс-сета, активного только для этого профиля

dgctl -u prefixset://first/firewall/ -c new internal_access

# Добавление IP-масок в префикс-сет
dgctl -u prefixset://first/firewall/internal_access -c insert -- 1.4.4.3/32 1,1.4.4.5/32 1,1.4.6.0/24 1

# Применение префикс-сета
dgctl -u prefixset://first/firewall/internal_access -c commit

# Создание правила, которое отрабасывает весь трафик на порт получателя 22 на протоколе UDP и TCP, в случае если IP-отправителя не находится в префикс-сете internal_access
dgctl -u profile://first/firewall -c insert -- -m dport 22 ! -m pset
```

```
internal_access class local what src value 1 -j STATS no_internal_access -j
DROP
```

Blackholing

В случае если нам нужно заблокировать доступ к определенному IP-адресу, или от определенных IP-адресов. В этом примере адреса находятся сразу в правиле, но они также могут забираться из созданного префикс-сета

```
dgctl -u profile://first/firewall -c insert -- -m dst 7.7.7.7 -j STATS
dst_7_7_7_7_blackhole -j DROP
dgctl -u profile://first/firewall -c insert -- -m src 5.5.5.5 -j STATS
src_5_5_5_5_blackhole -j DROP
```

Географический шейпинг

В некоторых случаях, требуется ограничить доступ к определенным ресурсам, например, только Российскими сетями. В этом примере мы ограничиваем трафик из всех стран кроме России в сторону указанных IP-масок получателей в 10 MBps.

```
dgctl -u profile://first/firewall -c insert -- ! -m geoip cntr RU -m dst
3.3.3.3,3.2.2.0/24,3.7.6.2/28 -j RATELIMIT 1 key "" bps 10m
dgctl -u profile://first/firewall -c insert -- -m verdict ratelimit exceed -j
STATS geo_shape -j DROP
```

Bandwidth enforcement

В сетевой инфраструктуре находится несколько групп клиентов, которые отправляют и получают трафик

Администратору системы необходимо ограничить полосу пропускания на входящий и исходящий трафик индивидуально для каждой группы клиентов

Такое может быть применимо, например, в операторе связи, который продает разные группы тарификации разным клиентам, или хостингу, который хочет ограничить полосу пропускания определенных виртуальных или серверных платформ

В этом примере, трафик на постоянной основе маршрутизируется через аппаратную платформу на которой установлено ПО. Создано две арены: одна на входящий трафик, другая на исходящий

`arena:in eth1 > eth2 | arena:out eth2 > eth1`

В данном примере также всего 2 группы клиентов, их может быть больше

Первая группа клиентов ограничивается в `100 MBps входящего трафика` и `100 MBps исходящего трафика`

Вторая группа клиентов ограничивается в `50 MBps входящего трафика` и `25 MBps исходящего трафика`

Создание профилей

```
dgctl -u arena://in -c add bandwidth
dgctl -u arena://out -c add bandwidth
```

Настройка маршрутизации всего трафика через эти профили

```
dgctl -u router://in/bandwidth -c insert -- 0.0.0.0/0
dgctl -u router://out/bandwidth -c insert -- 0.0.0.0/0
```

Создание префикс-сетов, как определителя группы клиентов

```
# Входящий трафик
dgctl -u prefixset://in/bandwidth/ -c new group1
dgctl -u prefixset://in/bandwidth/ -c new group2

# Исходящий трафик
dgctl -u prefixset://out/bandwidth/ -c new group1
dgctl -u prefixset://out/bandwidth/ -c new group2
```

Добавление IP-адресов в префикс-сеты

```
# Адреса первой группы
dgctl -u prefixset://in/bandwidth/group1 -c insert -- 1.1.1.0/24 1
dgctl -u prefixset://out/bandwidth/group1 -c insert -- 1.1.1.0/24 1

# Адреса второй группы
```

```
dgctl -u prefixset://in/bandwidth/group2 -c insert -- 1.1.2.0/24 1
dgctl -u prefixset://out/bandwidth/group2 -c insert -- 1.1.2.0/24 1
```

Правила для входящего трафика

```
# Ограничения для первой группы
dgctl -u profile://in/bandwidth -c insert -- -m pset group1 class local what
dst -j RATELIMIT 1 key "l3_dst" bps 100m
dgctl -u profile://in/bandwidth -c insert -- -m verdict ratelimit exceed -j
DROP
dgctl -u profile://in/bandwidth -c insert -- -j VERDICT clear

# Ограничения для второй группы
dgctl -u profile://in/bandwidth -c insert -- -m pset group2 class local what
dst -j RATELIMIT 1 key "l3_dst" bps 50m
dgctl -u profile://in/bandwidth -c insert -- -m verdict ratelimit exceed -j
DROP
dgctl -u profile://in/bandwidth -c insert -- -j VERDICT clear
```

Правила для исходящего трафика трафика

```
# Ограничения для первой группы
dgctl -u profile://out/bandwidth -c insert -- -m pset group1 class local what
src -j RATELIMIT 1 key "l3_dst" bps 100m
dgctl -u profile://out/bandwidth -c insert -- -m verdict ratelimit exceed -j
DROP
dgctl -u profile://out/bandwidth -c insert -- -j VERDICT clear

# Ограничения для второй группы
dgctl -u profile://out/bandwidth -c insert -- -m pset group2 class local what
src -j RATELIMIT 1 key "l3_dst" bps 25m
dgctl -u profile://out/bandwidth -c insert -- -m verdict ratelimit exceed -j
DROP
dgctl -u profile://out/bandwidth -c insert -- -j VERDICT clear
```