Обеспечение качества обслуживания для трафика (QoS)

Настоящий документ содержит описание способа совместного использования Продуктов компании ООО «С-Терра СиЭсПи» и Продуктов третьих производителей.
ООО «С-Терра СиЭсПи» осуществляет сопровождение настоящего сценария в части настроек Продуктов Компании. Упоминание наименований, продуктов, торговых марок третьих организаций исключительно неформально и не является поддержкой, рекомендацией либо рекламой. ООО «С-Терра СиЭсПи» не несет какой-либо ответственности в отношении работоспособности и использования этих Продуктов. Документ имеет статус вспомогательного материала, который может быть использован технологическими партнерами, компаниями-интеграторами, при разработке собственных решений.
Решения, разработанные на базе данного сценария, могут применяться в действующих сетях/системах только после тестовой и/или опытной эксплуатации.

Введение

Данная инструкция описывает настройку качества обслуживания для сетевого трафика в продукте «С-Терра Шлюз» версии 5.0.

Предварительные требования

Требования к квалификации администратора

Администратор должен обладать обширными знаниями в области сетевой информационной безопасности, иметь опыт работы с аналогичным оборудованием/программным обеспечением, знать и понимать следующие технологии и протоколы: PKI, IPsec, NAT, Firewall, routing, VRF, switching.

Администратор должен быть ознакомлен со сценарием «Организация защищенного взаимодействия между подсетями центрального офиса и филиалов посредством IPsec VPN туннеля (протокол IPv4)». Далее по тексту он будет называться базовым сценарием.

Требования к инфраструктуре

1. Требования к устройствам.

   1.1. «С-Терра Шлюз»: криптошлюзы GW1 и GW2.

      1.1.1 Устройства «С-Терра Шлюз» должны быть инициализированы (подробнее на http://doc.s-terra.ru раздел С-Терра Шлюз -> С-Терра Шлюз <версия> -> «Подключение ПАК и инициализация С-Терра Шлюз на вычислительных системах архитектуры Intel x86-64»).

      1.1.2 На устройствах «С-Терра Шлюз» должны быть установлены корневой сертификат и сертификат устройства.

   1.2. Защищаемые хосты (ОС Debian 12): Host1-b-gw1, Host1-b-gw2, Host2-b-gw1, Host2-b-gw2.

© 2003-2025 S-Terra CSP

1.2.1 На защищаемых хостах должны быть настроены IP-адреса.

1.3. Коммутаторы (ОС Debian 12): SW1 и SW2.

1.4. Маршрутизатор (ОС Debian 12): ISP1.

1.4.1 На маршрутизаторе должны быть настроены IP-адреса и включён forwarding.

Схема взаимодействия

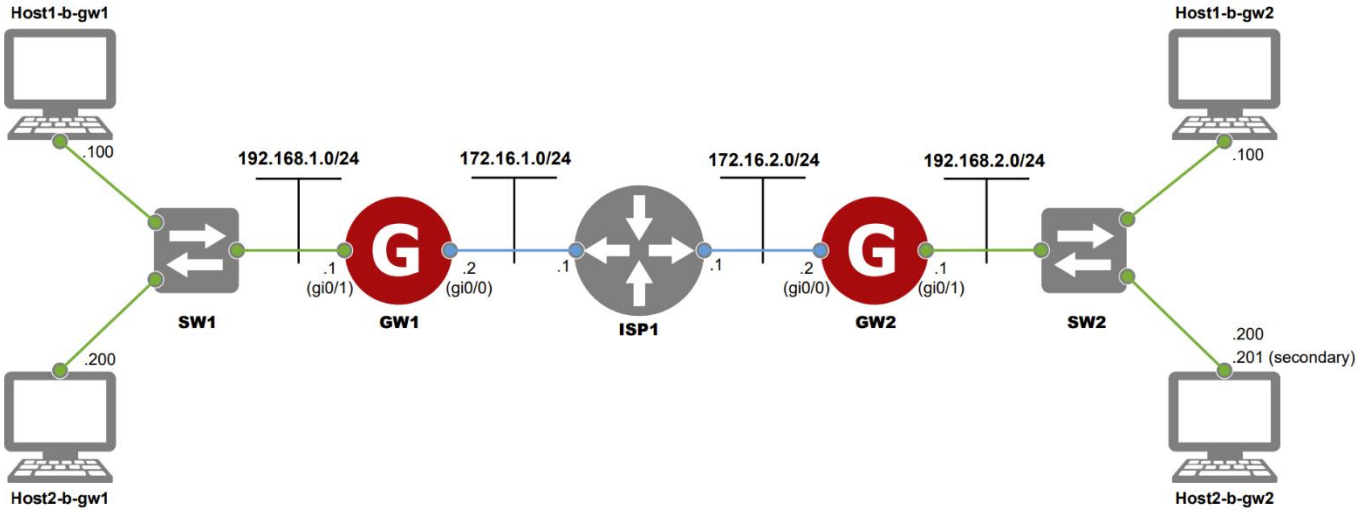


Рисунок 1. Схема взаимодействия

Общая логика работы

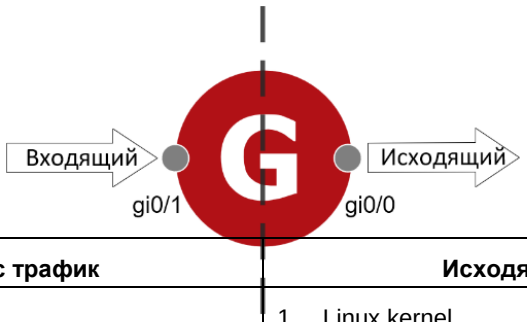
В Продукте «С-Терра Шлюз» поддерживается следующий список технологий, понимаемых под общим названием QoS:

- Маркирование (выставление DSCP/IP-precedence) IP-пакетов – осуществляется средствами самого Продукта.

По умолчанию IKE-пакеты маркируются DSCP равным 000001 (данное значение может быть изменено средствами Продукта). Также в Продукте «С-Терра Шлюз» версии 5.0 есть возможность включения/отключения переноса DSCP-метки из исходного незашифрованного пакета в зашифрованный при помощи команды `set dscp copy` в контексте настройки крипто-карты (старшие 6 бит – DSCP – переносятся, а младшие 2 бита – ECN – зануляются). По умолчанию перенос DSCP-метки из исходного незашифрованного пакета в зашифрованный включён. Будьте внимательны, в криптошлюзе версии 4.3 ToS-байт копировался целиком (то есть, все 8 бит – и DSCP, и ECN).

- Приоритизация промаркированных IP-пакетов – осуществляется средствами самого Продукта.
- Ограничение полосы пропускания (shaping) исходящего с интерфейса трафика – осуществляется средствами подсистемы traffic control ядра linux.

Порядок операций над сетевым трафиком на криптошлюзе



Входящий на интерфейс трафик	Исходящий из интерфейса трафик
1. Linux kernel. 1.1. подсистема QoS (traffic control): ограничение полосы пропускания (policing) и классификация трафика.	1. Linux kernel. 1.1. Подсистема Routing: статическая и динамическая маршрутизация. 1.2. Подсистема Firewall (netfilter): Output Stateful Packet Inspection, Output Stateless Filtering. 1.3. Подсистема NAT (netfilter): Source NAT.
2. ПО «С-Терра». 2.1. Предварительная обработка пакетов. 2.1.1 Разделение пакетов по приоритетным очередям в зависимости от значения ToS-байта в пакете и значения «pq_tos_mask» в драйвере Продукта. 2.2. Подсистема межсетевого экранирования (Firewall). 2.2.1 Инспектирование TCP, UDP, FTP трафика (INPUT Stateful Packet Inspection). 2.2.2 Проверка трафика по спискам доступа (INPUT ACL). 2.3. Подсистема шифрования (IPsec). 2.3.1 Расшифрование (если пакеты зашифрованы). 2.3.2 Фильтрация трафика в рамках IPsec. 2.4. Подсистема QoS: маркирование трафика.	2. ПО «С-Терра» 2.1. Предварительная обработка пакетов. 2.1.1 Разделение пакетов по приоритетным очередям в зависимости от значения ToS-байта в пакете и значения «pq_tos_mask» в драйвере Продукта. 2.2. Подсистема QoS: маркирование трафика. 2.3. Подсистема шифрования (IPsec). 2.3.1 Фильтрация трафика в рамках IPsec. 2.3.2 Зашифрование трафика (если требуется). 2.4. Подсистема межсетевого экранирования (Firewall). 2.4.1 Инспектирование TCP, UDP, FTP трафика (OUTPUT Stateful Packet Inspection). 2.4.2 Проверка трафика по спискам доступа (OUTPUT ACL).
3. Linux kernel. 3.1. Подсистема NAT (netfilter): Destination NAT. 3.2. Подсистема Routing: статическая и динамическая маршрутизация. 3.3. Подсистема Firewall (netfilter): Input Stateful Packet Inspection, Input Stateless Filtering.	3. Linux kernel. 3.1. Подсистема QoS (traffic control): ограничение полосы пропускания (shaping) и приоритизация трафика.

Рисунок 2. Схема прохождения трафика через криптошлюз

Маркирование

Маркирование в продукте «С-Терра Шлюз» осуществляется средствами самого Продукта и заключается в выставлении метки DSCP (Differentiated Services Code Point – информация о классификации, передаваемая в IP-заголовке в поле ToS) или IP-precedence (информация о классификации, передаваемая в IPv4-заголовке в поле ToS).

Приоритизация

Начиная с версии 5.0, в Продукте «С-Терра Шлюз» по умолчанию работают **две пары очередей** – одна пара очередей для входящих пакетов, другая пара для исходящих пакетов. Каждая пара очередей включает в себя приоритетную очередь (очередь для приоритетных пакетов) и неприоритетную очередь (очередь неприоритетных пакетов). Приоритетным является пакет, значение ToS-байта которого по маске `pq_tos_mask` в `drv_mgr` отлично от 0. Если на сетевой интерфейс криптошлюза (например, на LAN-интерфейс) попадают приоритетные пакеты, то криптошлюз отправит данные пакеты в приоритетную очередь соответствующей пары. Сначала на обработку и отправку выбираются приоритетные пакеты, и только при их отсутствии – неприоритетные. Размер очереди на обработку составляет по умолчанию 2000 пакетов и определяется параметром `pq_thread_q_size` в `drv_mgr` (параметр может принимать значения от 0 до 16383).

Приоритетная очередь в Продукте «С-Терра Шлюз» версии 5.0 включена по умолчанию. Если же по каким-то причинам требуется отключить её, то в файл `/etc/modprobe.d/vpndrvr.conf` нужно добавить строку `options vpndrvr use_priority_queue=0` и перезагрузить криптошлюз.

Схема работы двух пар очередей представлена на Рисунок 3. Логика обработки входящих пакетов следующая:

1. Сначала пакет попадает на сетевой интерфейс.
2. Далее, в блок «предварительная обработка» (где определяется, является ли пакет приоритетным или нет).
3. Затем, в зависимости от того, приоритетный пакет или нет, - в очередь (приоритетную или неприоритетную соответственно).
4. В конечном счёте, в блок «фильтрация, инспектирование, маркирование, шифрование», где производится обработка всех пакетов (в первую очередь, приоритетных).

Обработка исходящих пакетов производится аналогично вышеописанной для входящих пакетов.

© 2003-2025 S-Terra CSP

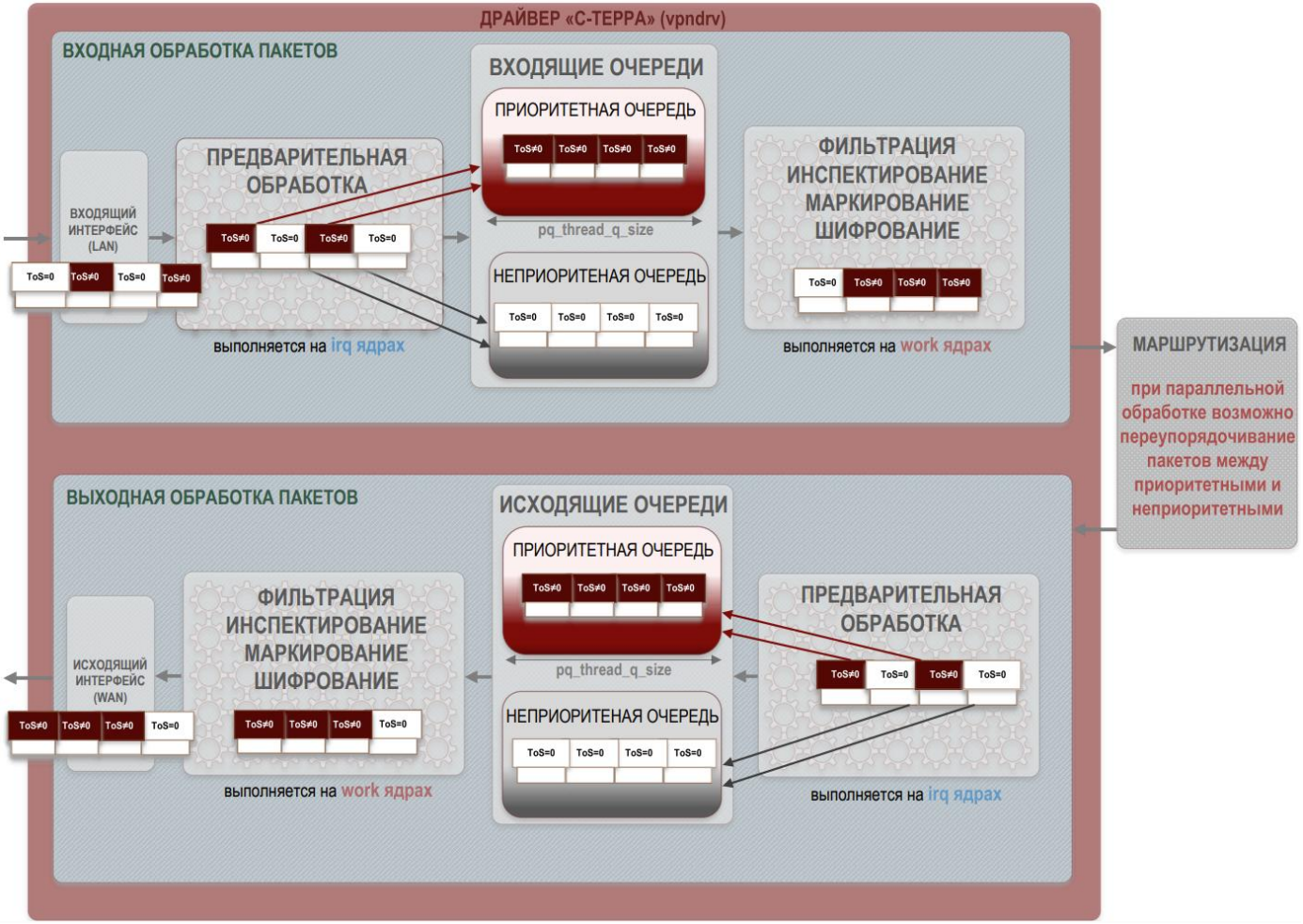


Рисунок 3. Схема обработки сетевого трафика на криптошлюзе

В рамках предварительной обработки может также выполняться **предварительная классификация** сетевых пакетов на основе кэша (настраивается при помощи параметра драйвера «qos_preclassify»). Данный механизм позволяет определить, есть ли в блоке фильтрации, инспектирования, маркирования и шифрования правило маркирования пакета (например, назначение или изменение ToS-байта). Если оно есть, то при предварительной обработке использовать не текущее значение ToS-байта для определения приоритетности пакета, а то значение ToS-байта, которое будет ему назначено при фильтрации, инспектирования, маркирования и шифрования. Таким образом, включение параметра драйвера «qos_preclassify» может использоваться в тех случаях, когда маркирование трафика происходит средствами самого Продукта и необходимо заранее проклассифицировать сетевой трафик, который вскоре будет промаркирован (то есть, заранее определить, в какую очередь попадёт трафик, который должен быть промаркирован Продуктом: приоритетную или неприоритетную). По умолчанию, qos_preclassify=0 (предварительная классификация отключена). Если qos_preclassify=1, то предварительная классификация будет работать в режиме «общий кэш». Если qos_preclassify=2, то предварительная классификация будет работать в режиме «расширенный кэш» (более производительный вариант).

Будьте внимательны! Предварительную классификацию в сценариях с VRF использовать нельзя (так как кэш в случае пересекающихся адресных пространств будет общим, что может привести к некорректной работе криптошлюза).

Продукт «С-Терра Шлюз» версии 5.0 поддерживает набор настроек приоритизации, который использовался в Продукте версии 4.3 (см. инструкции для «С-Терра Шлюз» версии 4.3). Для настройки приоритизации согласно инструкции для «С-Терра Шлюз» версии 4.3 необходимо отключить работу приоритетной очереди в Продукте версии 5.0, как указано выше.

Ограничение полосы пропускания (shaping)

Важно знать следующее – ограничение полосы пропускания на исходящем трафике в сторону IPsec-партнёра в подсистеме «traffic control» ядра linux происходит **после** его зашифрования, соответственно, ограничивать полосу пропускания трафика можно только на основе информации в зашифрованном IP-пакете. Этой информации не так много – туннельные IP-адреса и ToS-байт.

При ограничении полосы пропускания на исходящем интерфейсе (WAN-интерфейсе) передающего криптошлюза возможны срабатывания механизма защиты от атак воспроизведения пакетов на приемной стороне. Этот механизм включен по умолчанию. Атака воспроизведения пакетов выявляется путем сравнения sequence вновь принятого пакета с sequence предыдущего пакета. Если разница превышает предельно допустимую величину, недоступную для настройки, то вновь принятый пакет вопринимается, как воспроизведенный пакет из прошлого и отбрасывается.

Срабатывание защиты от атак воспроизведения на приемной стороне может происходить при импульсных перегрузках полосы пропускания на передающем криптошлюзе. Как было показано выше, ограничение полосы пропускания выполняется подсистемой QoS (traffic control) Linux kernel **после** зашифрования трафика и, соответственно, после назначения sequence зашифрованным пакетам, подлежащих отправке. При превышении ограничиваемым трафиком предела полосы пропускания, подсистема traffic control использует доступную пропускную способность смежных полос для своевременной отправки пакетов трафика, превышающего предел своей полосы. Но это возможно **только при наличии доступной пропускной способности** в полосе смежного трафика, например, когда смежные полосы пропускания не нагружены. Если доступная смежная полоса пропускания отсутствует, то пакеты из трафика, превышающего ограничение своей полосы пропускания, будут копиться в очереди в traffic control для последующей отправки. Накапливаемые в очереди пакеты будут иметь постепенно «устаревающие» sequence по сравнению с трафиком смежной полосы, который передается своевременно, если в ней нет перегрузки. Отсюда, на приемной стороне будут появляться пакеты, имеющие признаки атаки вопроизведения – это пакеты трафика, которые отправляются из очереди traffic control на передающей стороне, но которые не удалось отправить своевременно из-за перегрузки соответствующей полосы.

Срабатывание механизма защиты от атак воспроизведения на приемной стороне можно диагностировать по увеличению счетчика потерь «ipsec replay drops» в утилите «kstat_show». В Продукте «С-Терра Шлюз» версии 5.0 IPsec-окно (anti-replay window) увеличено до 2048 пакетов.

Для оценки причин возникновения потерь типа «ipsec replay drops» необходима информация об объемах трафика, передаваемого в каждой полосе пропускания в единицу времени. При наличии достаточной полосы пропускания для всех видов трафика потери из-за перегрузки отдельных полос пропускания не возникают. При выявлении перегрузок отдельных полос пропускания рекомендуется рассмотреть возможность уточнения настроек ограничений полос пропускания в соответствии с реальными потребностями в пропускной способности. Однако, данный путь не всегда возможен на практике из-за недостаточной предсказуемости объемов трафика. Если гарантировать выделение достаточных полос пропускания невозможно, то для предотвращения потерь типа «ipsec replay drops» из-за превышения полосы пропускания рекомендуется использовать один из следующих методов:

1. При возможности отказаться от shaping на WAN-интерфейсе и выполнять его ДО зашифрования трафика, например, на внешнем оборудовании (LAN-сегмент) или mGRE-интерфейсе.

2. При необходимости настройки shaping на WAN-интерфейсе настраивайте отдельные IPsec-туннели для каждого вида трафика, подлежащего shaping и потенциально подверженного перегрузкам.
3. Отключить механизм защиты от атак повторного воспроизведения пакетов на приемной стороне (команда `set security-association replay disable` в контексте настройки крипто-карты, см. пример в главе Приложение). Однако, с точки зрения безопасности данный функционал отключать не рекомендуется.

Задачи, реализуемые на стенде

Маркирование

Для проверки работоспособности маркирования необходимо:

1. Средствами криптошлюза GW1 выставить DSCP метку AF11 на немаркированный TCP-трафик между хостами Host1-b-gw1 и Host1-b-gw2.
2. Средствами криптошлюза GW1 выставить DSCP метку AF12 на немаркированный TCP-трафик между хостами Host2-b-gw1 и Host2-b-gw2.
3. Средствами криптошлюза GW1 выставить DSCP метку 0 на весь ICMP-трафик.
4. При помощи утилиты «klogview» на криптошлюзе GW1 убедиться в том, что маркирование работает корректно.

Приоритизация

Для того, чтобы убедиться в правильности работы приоритетной очереди, необходимо сгенерировать большое количество приоритетного и неприоритетного трафика, который бы практически максимально задействовал аппаратные мощности криптошлюза. В таком случае всю полосу пропускания должен занять приоритетный трафик (приоритетными считаются пакеты, имеющие ненулевое значение поля DSCP в IP-пакете).

Для проверки работоспособности приоритизации необходимо:

1. При помощи утилиты «iperf3» сгенерировать маркированный UDP-трафик между хостами Host1-b-gw1 и Host1-b-gw2.
2. При помощи утилиты «iperf3» сгенерировать немаркированный UDP-трафик между хостами Host2-b-gw1 и Host2-b-gw2.
3. При помощи утилиты «iperf3» на хостах Host1-b-gw1 и Host2-b-gw1 убедиться в том, что приоритизация работает корректно.

Ограничение полосы пропускания (shaping)

Для проверки работоспособности ограничения полосы пропускания необходимо:

1. Ограничить пропускную способность на внешнем интерфейсе до 50 Мбит/с (то есть, весь транзитный трафик, проходящий через криптошлюз в сторону IPsec-партнёра).
2. Ограничить пропускную способность для трафика с DSCP-меткой AF11 до 10 Мбит/с, гарантированно выделив под него 10 Мбит/с полосы пропускания (то есть, TCP-трафик между хостами Host1-b-gw1 – 192.168.1.100 и Host1-b-gw2 – 192.168.2.100).

3. Ограничить пропускную способность для трафика с DSCP-меткой AF12 до 10 Мбит/с, гарантированно выделив под него 10 Мбит/с полосы пропускания (то есть, TCP-трафик между хостами Host2-b-gw1 – 192.168.1.200 и Host2-b-gw2 – 192.168.2.200).
4. Ограничить пропускную способность для остального трафика до 30 Мбит/с, гарантированно выделив под него 10 Мбит/с полосы пропускания.
5. При помощи утилиты «iperf3» на хостах Host1-b-gw1 и Host2-b-gw1 убедиться в том, что ограничение полосы пропускания работает корректно.

Настройка стенда

Перед настройкой стенда необходимо выполнить предварительные требования к начальному состоянию «С-Терра Шлюз», которые описаны в главе «Предварительные требования».

Настройка оконечных устройств

На оконечных устройствах Host1-b-gw1, Host2-b-gw1, Host1-b-gw2, Host2-b-gw2 (ОС Debian 12) должны быть настроены IP-адреса и маршрутизация в соответствии с общей схемой стенда (см. рисунок 1).

Настройка коммутаторов

На коммутаторах SW1 и SW2 сетевые интерфейсы, коммутирующие шлюз GW1 и хосты Host1-b-gw1, Host2-b-gw1 и шлюз GW2 и хосты Host1-b-gw2, Host2-b-gw2, должны быть в одном VLAN в режиме portfast.

Настройка роутера ISP

На роутере ISP должны быть настроены IP-адреса в соответствии с общей схемой стенда (см. рисунок 1).

Настройка криптошлюза GW1

Предполагается, что базовые сетевые настройки и настройки шифрования уже выполнены (в главе «Приложение» представлена конфигурация cisco-like консоли). Ниже будет описано конфигурирование маркирования, приоритизации и ограничения полосы пропускания.

Настройка маркирования

Настройка маркирования осуществляется из cisco-like консоли криптошлюза.

1. Создайте необходимые списки доступа, которые описывают маркируемый трафик:

```
GW1#conf t
GW1(config)#ip access-list extended SET_DSCP_AF11
GW1(config-ext-nacl)#permit tcp host 192.168.1.100 Host 192.168.2.100
GW1(config-ext-nacl)#exit
GW1(config)#ip access-list extended SET_DSCP_AF12
GW1(config-ext-nacl)#permit tcp host 192.168.1.200 Host 192.168.2.200
GW1(config-ext-nacl)#exit
GW1(config)#ip access-list extended CLEAR_DSCP
GW1(config-ext-nacl)#permit icmp any any
GW1(config-ext-nacl)#exit
```

2. Опишите классы трафика на основе списков доступа:

```
GW1(config)#class-map match-all CLEAR_DSCP
GW1(config-cmap)#match access-group name CLEAR_DSCP
GW1(config-cmap)#exit
GW1(config)#class-map match-all SET_DSCP_AF11
GW1(config-cmap)#match access-group name SET_DSCP_AF11
GW1(config-cmap)#exit
GW1(config)#class-map match-all SET_DSCP_AF12
GW1(config-cmap)#match access-group name SET_DSCP_AF12
GW1(config-cmap)#exit
```

3. Опишите политику для трафика (выставление DSCP метки):

```
GW1(config)#policy-map DSCP_POLICY
GW1(config-pmap)#class SET_DSCP_AF11
GW1(config-pmap-c)#set dscp af11
GW1(config-pmap-c)#exit
GW1(config-pmap)#class SET_DSCP_AF12
GW1(config-pmap-c)#set dscp af12
GW1(config-pmap-c)#exit
GW1(config-pmap)#class CLEAR_DSCP
GW1(config-pmap-c)#set dscp default
GW1(config-pmap-c)#exit
GW1(config-pmap)#exit
```

4. Прикрепите политику для трафика к внутреннему интерфейсу на входящее направление:

```
GW1(config)#interface GigabitEthernet0/1
GW1(config-if)# service-policy input DSCP_POLICY
GW1(config-if)#exit
GW1(config)#end
```

Настройка приоритизации

Настройка приоритизации осуществляется из консоли linux bash.

1. Установите ToS-маску для определения приоритетности пакета (при значении маски в 252 из ToS-байта учитываются только первые шесть бит, то есть DSCP, если же необходимо выставить IP-precedence, то ToS-маска должна быть установлена в значение 192):

```
root@GW1:~# drv_mgr set pq_tos_mask 252
Value of "pq_tos_mask" is set to 252.
```

Определенные значения DSCP-метки задать нельзя.

Настройка ограничения полосы пропускания (shaping)

Сопоставление имени интерфейса в linux с именем интерфейса в cisco-like консоли можно посмотреть в файле `/etc/ifaliases.cf`.

Настройка ограничения полосы пропускания осуществляется из linux bash. Для того, чтобы осуществить ограничение полосы пропускания согласно главе «Ограничение полосы пропускания (shaping)», необходимо:

1. Удалить корневую дисциплину:

```
tc qdisc del dev eth0 root
```

Каждому сетевому интерфейсу в Linux назначается дисциплина (qdisc – queue discipline). Из дисциплин и строится вся система управления трафиком. Дисциплина – это алгоритм обработки очереди сетевых пакетов (дисциплин на одном сетевом интерфейсе может быть задействовано несколько). По умолчанию к каждому сетевому интерфейсу прикреплена так называемая корневая дисциплина (root qdisc). При этом каждый интерфейс имеет свою собственную корневую дисциплину.

2. Добавить корневую классовую дисциплину HTB к интерфейсу с классом по умолчанию 4 (трафик, который не попал под фильтры, направляется в этот класс):

```
tc qdisc add dev eth0 root handle 1:0 htb default 4
```

Классовая дисциплина HTB (Hierarchical Token Bucket) – алгоритм обработки сетевых пакетов, позволяющий производить заимствование полосы пропускания дочерними классами у родительского.

3. Ограничить скорость на интерфейсе до 50 Мбит/с:

```
tc class add dev eth0 parent 1:0 classid 1:1 htb rate 50mbit ceil 50mbit
```

Ключ “rate” позволяет установить минимальную гарантированную пропускную способность для класса. Ключ “ceil” – верхний порог.

4. Определить дочерние классы (1:2, 1:3, 1:4) и ограничить в них скорость:

```
tc class add dev eth0 parent 1:1 classid 1:2 htb rate 10mbit ceil 10mbit
tc class add dev eth0 parent 1:1 classid 1:3 htb rate 10mbit ceil 10mbit
tc class add dev eth0 parent 1:1 classid 1:4 htb rate 10mbit ceil 30mbit
```

5. При помощи фильтров направить трафик в нужные классы (в `tc` нельзя задать DSCP, поэтому указывается шестнадцатеричное значение ToS байта; чтобы перевести DSCP в ToS, нужно добавить справа к двоичному значению DSCP два нуля и перевести его в шестнадцатеричное представление: AF11 -> 001010 -> 00101000 -> 0x28, AF12 -> 001100 -> 00110000 -> 0x30). Пакеты с меткой AF11 (0x28) попадают в класс 1:2 (`flowid 1:2`), а пакеты с меткой AF12 (0x30) попадают в класс 1:3 (`flowid 1:3`), пакеты не попавшие под фильтры автоматически попадают в класс по умолчанию 1:4:

```
tc filter add dev eth0 parent 1:0 protocol ip prio 0 u32 match ip tos 0x28 0xfc
flowid 1:2
tc filter add dev eth0 parent 1:0 protocol ip prio 0 u32 match ip tos 0x30 0xfc
flowid 1:3
```

Команды, описанные выше, представлены в виде скрипта в главе «Настройки ограничения полосы пропускания для устройств GW1 и GW2» в приложении. Данные правила будут применяться к интерфейсу, который указан в скрипте (в данном сценарии шейпинг осуществляется на интерфейсе `eth0`).

Для того, чтобы воспользоваться скриптом, необходимо:

1. Добавить содержимое скрипта в файл `/etc/network/if-up.d/zshaper`:

```
root@GW1:~# vim.tiny /etc/network/if-up.d/zshaper
```

> Будьте внимательны, скрипты из директории `/etc/network/if-up.d/` выполняются в лексикографическом порядке, именно поэтому ниже создается скрипт с названием `zshaper` (для того, чтобы выполнение скрипта гарантированно осуществлялось в последнюю очередь при наличии других скриптов).

2. Дать файлу `/etc/network/if-up.d/shaper` права на исполнение:

```
root@GW1:~# chmod u+x /etc/network/if-up.d/zshaper
```

> Данный файл будет запускаться автоматически при старте системы во время конфигурирования интерфейса `eth0` (а также при выполнении команды `ifup eth0`). Если ограничение полосы настраивается на другом интерфейсе, то в рамках скрипта, указанного в главе «Настройки ограничения полосы пропускания для устройств GW1 и GW2», в строке 23 нужно заменить "eth0" на имя необходимого интерфейса).

3. Применить настройки ограничения полосы пропускания:

```
root@GW1:~# ifdown eth0; ifup eth0
```

Посмотреть статистику можно при помощи команд:

```
root@GW1:~# tc -s qdisc show dev eth0
root@GW1:~# tc -s filter show dev eth0
root@GW1:~# tc -s class show dev eth0
```

Настройка криптошлюза GW2

Настройка криптошлюза GW2 осуществляются аналогично настройкам криптошлюза GW1.

В главе «Приложение» представлена конфигурация cisco-like консоли.

Проверка работоспособности стенда

Для проверки работоспособности будут использованы утилиты «iperf3», «tcpdump» и «klogview». В качестве криптошлюза, на котором будут проводится проверки, выбран GW1.

Маркирование

1. Убедитесь, что трафик между хостами Host1-b-gw1 и Host1-b-gw2 маркируется верно (DSCP 001010 ->ToS 0x28):

   1.1. На устройстве Host1-b-gw2 запустите утилиту «iperf3» в режиме «сервер»:

```
root@Host1-b-gw2:~# iperf3 -s
```

```
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
```

   1.2. На устройстве Host1-b-gw1 запустите утилиту «iperf3» в режиме «клиент» и сгенерируйте TCP-трафик в сторону Host1-b-gw2:

```
root@Host1-b-gw1:~# iperf3 -c 192.168.2.100 -t 0
```

```
Connecting to Host 192.168.2.100, port 5201
[  5] local 192.168.1.100 port 58428 connected to 192.168.2.100 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  1.49 MBytes  12.5 Mbits/sec    0   83.5 KBytes
[  5]   1.00-2.00   sec  1.23 MBytes  10.3 Mbits/sec    0    141 KBytes
[  5]   2.00-3.00   sec  1.35 MBytes  11.4 Mbits/sec    0    197 KBytes
[  5]   3.00-4.00   sec  1.41 MBytes  11.9 Mbits/sec    0    255 KBytes
[  5]   4.00-5.00   sec  1.17 MBytes  9.80 Mbits/sec    0    311 KBytes
[  5]   5.00-6.00   sec  1.35 MBytes  11.3 Mbits/sec    0    367 KBytes
[  5]   6.00-7.00   sec  1.54 MBytes  12.9 Mbits/sec    0    424 KBytes
```

1.3. Проверьте, что правило маркирования срабатывает:

```
root@GW1:~# klogview -f all | grep SET_DSCP_AF11
```

```
filtration result for in packet 192.168.1.100:35762->192.168.2.100:5101, proto 6,
len 1500, if eth1: chain 101 "ClassificationChain:DSCP_POLICY", filter 150, event
id Classification:DSCP_POLICY:SET_DSCP_AF11:SET_DSCP_AF11, status PASS
filtration result for in packet 192.168.1.100:35762->192.168.2.100:5101, proto 6,
len 1500, if eth1: chain 101 "ClassificationChain:DSCP_POLICY", filter 150, event
id Classification:DSCP_POLICY:SET_DSCP_AF11:SET_DSCP_AF11, status PASS
```

1.4. Проверьте, что на внешнем интерфейсе (eth0) криптошлюза GW1 в зашифрованных пакетах выставлен ToS-байт со значением 0x28:

```
root@GW1:~# tcpdump -i eth0 -n -v 'ip[1] == 0x28' -c 2
```

```
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
09:00:54.605709 IP (tos 0x28, ttl 254, id 22448, offset 0, flags [none], proto
ESP (50), length 96)
    172.16.2.2 > 172.16.1.2: ESP(spi=0xbd6f457e,seq=0x9777), length 76
09:00:54.606427 IP (tos 0x28, ttl 255, id 39549, offset 1480, flags [none], proto
ESP (50), length 64)
    172.16.1.2 > 172.16.2.2: ip-proto-50
2 packets captured
3 packets received by filter
0 packets dropped by kernel
```

2. Проверка маркирования трафика между хостами Host2-b-gw1 и Host2-b-gw2 проводится аналогично проверке для хостов Host1-b-gw1 и Host1-b-gw2.

3. Проверьте, что значение DSCP сбрасывается в значение по умолчанию для ICMP-трафика.

3.1. Запустите утилиту «ping» на устройстве Host1-b-gw1 в сторону устройства Host1-b-gw2 с вручную выставленным значением ToS байта (0xb8):

```
root@Host1-b-gw1:~# ping 192.168.2.100 -Q 0xb8
```

```
PING 192.168.2.100 (192.168.2.100) 56(84) bytes of data.
64 bytes from 192.168.2.100: icmp_req=1 ttl=62 time=1.96 ms
64 bytes from 192.168.2.100: icmp_req=2 ttl=62 time=1.85 ms
64 bytes from 192.168.2.100: icmp_req=3 ttl=62 time=1.29 ms
```

3.2. Проверьте, что правило сброса метки срабатывает:

```
root@GW1:~# klogview -f all | grep CLEAR_DSCP
```

```
filtration result for in packet 192.168.1.100->192.168.2.100, proto 1, len 84, if
eth1: chain 109 "ClassificationChain:DSCP_POLICY", filter 160, event id
Classification:DSCP_POLICY:CLEAR_DSCP:CLEAR_DSCP, status PASS
```

```
filtration result for in packet 192.168.1.100->192.168.2.100, proto 1, len 84, if
eth1: chain 109 "ClassificationChain:DSCP_POLICY", filter 160, event id
Classification:DSCP_POLICY:CLEAR_DSCP:CLEAR_DSCP, status PASS
```

3.3. Проверьте, что на внешнем интерфейсе `eth0` криптошлюза GW1 в зашифрованных пакетах значение ToS-байта равно 0x00:

```
root@GW1:~# tcpdump -i eth0 -n -v 'ip[1] == 0x00' -c 2
```

```
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 bytes
10:11:09.651465 IP (tos 0x0, ttl 255, id 60244, offset 0, flags [none], proto ESP
(50), length 128)
    172.16.1.2 > 172.16.2.2: ESP(spi=0xe39ac051,seq=0x196c81), length 108
10:11:09.652439 IP (tos 0x0, ttl 254, id 48678, offset 0, flags [none], proto ESP
(50), length 128)
    172.16.2.2 > 172.16.1.2: ESP(spi=0xde6e0ba3,seq=0x5322), length 108
2 packets captured
2 packets received by filter
0 packets dropped by kernel
```

Приоритизация

1. Отключите ограничение полосы пропускания (если было настроено) на криптошлюзах для того, чтобы система шифрования была задействована полностью:

```
root@GW1:~# tc qdisc del dev eth0 root
```

Будьте внимательны, объём генерируемого трафика при проверке работы приоритизации должен быть выбран исходя из мощностей аппаратной платформы. Степень загрузки ядер процессора (в процентах от 0 до 100) в режиме реального времени можно посмотреть при помощи команды `watch cpupower monitor` (поле «Mperf», графа «C0»).

2. При помощи утилиты «iperf3» сгенерируйте маркированный UDP-трафик между хостами Host1-b-gw1 и Host1-b-gw2.

2.1. На устройстве Host1-b-gw2 запустите утилиту «iperf3» в режиме «сервер»:

```
root@Host1-b-gw2:~# iperf3 -s
```

2.2. На устройстве Host1-b-gw1 запустите утилиту «iperf3» в режиме «клиент» и сгенерируйте маркированный UDP-трафик в сторону устройства Host1-b-gw2 (например, с выставленным ToS-байтом в значение 136, что равно 0x88 в шестнадцатеричной системе или DSCP-метке AF41) с нагрузкой в 300 Мбит/с:

```
root@Host1-b-gw1:~# iperf3 -c 192.168.2.100 -B 192.168.1.100 -l 500 -u -b 300M -
t 0 -Z -S 136
```

3. При помощи утилиты «iperf3» сгенерируйте немаркированный UDP-трафик между хостами Host2-b-gw1 и Host2-b-gw2.

3.1. На устройстве Host2-b-gw2 запустите утилиту «iperf3» в режиме «сервер»:

```
root@Host2-b-gw2:~# iperf3 -s
```

3.2. На устройстве Host2-b-gw1 запустите утилиту «iperf3» в режиме «клиент» и сгенерируйте немаркированный UDP-трафик в сторону Host2-b-gw2 с нагрузкой в 300 Мбит/с:

```
root@Host2-b-gw1:~# iperf3 -c 192.168.2.200 -B 192.168.1.200 -l 500 -u -b 300M -
t 0 -Z
```

4. Сравните результаты тестов:

```
root@Host1-b-gw2:~# iperf3 -s
```

```
-----------------------------------------------------------
Server listening on 5201 (test #1)
-----------------------------------------------------------
Accepted connection from 192.168.1.100, port 58665
[  5] local 192.168.2.100 port 5201 connected to 192.168.1.100 port 59752
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total
Datagrams
[  5]   0.00-1.00   sec  19.3 MBytes   162 Mbits/sec  0.020 ms  7569/48121 (16%)
[  5]   1.00-2.00   sec  24.7 MBytes   207 Mbits/sec  0.012 ms  30/51798 (0.058%)
[  5]   2.00-3.00   sec  23.8 MBytes   200 Mbits/sec  0.020 ms  0/50001 (0%)
[  5]   3.00-4.00   sec  23.5 MBytes   197 Mbits/sec  0.027 ms  0/49206 (0%)
[  5]   4.00-5.00   sec  23.8 MBytes   200 Mbits/sec  0.018 ms  0/49965 (0%)
[  5]   5.00-6.00   sec  23.8 MBytes   200 Mbits/sec  0.030 ms  0/50014 (0%)
[  5]   6.00-7.00   sec  23.9 MBytes   200 Mbits/sec  0.022 ms  0/50030 (0%)
[  5]   7.00-8.00   sec  23.9 MBytes   200 Mbits/sec  0.010 ms  0/50029 (0%)
[  5]   8.00-9.00   sec  23.8 MBytes   200 Mbits/sec  0.018 ms  0/49965 (0%)
[  5]   9.00-10.00  sec  23.8 MBytes   200 Mbits/sec  0.065 ms  0/49990 (0%)
[  5]  10.00-11.00  sec  23.8 MBytes   200 Mbits/sec  0.065 ms  46/49960 (0.092%)
[  5]  11.00-12.00  sec  23.9 MBytes   200 Mbits/sec  0.073 ms  0/50041 (0%)
[  5]  11.00-12.00  sec  23.9 MBytes   200 Mbits/sec  0.073 ms  0/50041 (0%)
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total
Datagrams
[  5]   0.00-12.00  sec   289 MBytes   202 Mbits/sec  0.021 ms  7645/613768 (1.2%)
receiver
```

```
root@Host2-b-gw2:~# iperf3 -s
```

```
-----------------------------------------------------------
Server listening on 5201 (test #1)
-----------------------------------------------------------
Accepted connection from 192.168.1.200, port 42503
[  5] local 192.168.2.200 port 5201 connected to 192.168.1.200 port 38653
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total
Datagrams
[  5]   0.00-1.00   sec  11.5 MBytes  96.7 Mbits/sec  0.083 ms  23140/47327 (49%)
[  5]   1.00-2.00   sec  12.7 MBytes   107 Mbits/sec  0.077 ms  25040/51764 (48%)
[  5]   2.00-3.00   sec  12.9 MBytes   108 Mbits/sec  0.016 ms  23011/49992 (46%)
[  5]   3.00-4.00   sec  12.7 MBytes   106 Mbits/sec  0.044 ms  23363/49979 (47%)
[  5]   4.00-5.00   sec  12.7 MBytes   107 Mbits/sec  0.027 ms  23252/49988 (47%)
[  5]   5.00-6.00   sec  12.3 MBytes   103 Mbits/sec  0.054 ms  24194/49923 (48%)
[  5]   6.00-7.00   sec  12.6 MBytes   106 Mbits/sec  0.027 ms  23582/50107 (47%)
[  5]   7.00-8.00   sec  12.1 MBytes   101 Mbits/sec  0.031 ms  24582/49910 (49%)
[  5]   8.00-9.00   sec  14.5 MBytes   121 Mbits/sec  0.009 ms  20557/50926 (40%)
[  5]   8.00-9.00   sec  14.5 MBytes   121 Mbits/sec  0.009 ms  20557/50926 (40%)
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total
Datagrams
[  5]   0.00-9.00   sec   121 MBytes   113 Mbits/sec  0.010 ms  210721/463967
(45%)  receiver
```

5. Ознакомьтесь с выводом утилиты «kstat_show»:

```
root@GW1:~# kstat_show | grep low
```

```
low priority packets dropped:                    1433696
```

Можно заметить, что большую часть полосы пропускания занял именно приоритетный трафик от устройства Host1-b-gw1 к устройству Host1-b-gw2: скорость передачи 200 Мбит/с и 1.2% потерь. Неприоритетный трафик, идущий от устройства Host1-b-gw1 к устройству Host1-b-gw2, наоборот, идёт со скоростью около 100 Мбит/с и имеет примерно 45% потерь. Это обусловливается тем, что маркированный трафик попадает в приоритетную очередь криптошлюза, поэтому потерь при передаче практически не наблюдается.

Ограничение полосы пропускания (shaping)

1. Для того, чтобы включить ограничение полосы пропускания, воспользуйтесь командой:

```
root@GW1:~# ifdown eth0; ifup eth0
```

2. Запустите отправку TCP-трафика с устройства Host1-b-gw1 в сторону устройства Host1-b-gw2 (согласно настроенным правилам маркирования, трафик будет помечен криптошлюзом DSCP-меткой AF11):

   2.1. На устройстве Host1-b-gw2 запустите утилиту «iperf3» в режиме «сервер»:

```
root@Host1-b-gw2:~# iperf3 -s
```

   2.2. На устройстве Host1-b-gw1 запустите утилиту «iperf3» в режиме «клиент» и сгенерируйте TCP-трафик в сторону Host1-b-gw2:

```
root@Host1-b-gw1:~# iperf3 -c 192.168.2.100
```

   2.3. Ознакомьтесь с результатами тестирования на стороне сервера:

```
Connecting to Host 192.168.2.100, port 5201
[  5] local 192.168.1.100 port 58492 connected to 192.168.2.100 port 5201
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  5]   0.00-1.00   sec  1.11 MBytes  9.27 Mbits/sec    0   82.1 KBytes
[  5]   1.00-2.00   sec  1.12 MBytes  9.44 Mbits/sec    0    136 KBytes
[  5]   2.00-3.00   sec  1.00 MBytes  8.39 Mbits/sec    0    136 KBytes
[  5]   3.00-4.00   sec  1.12 MBytes  9.44 Mbits/sec    0    136 KBytes
[  5]   4.00-5.00   sec  1.12 MBytes  9.44 Mbits/sec    0    136 KBytes
[  5]   5.00-6.00   sec  1.00 MBytes  8.39 Mbits/sec    0    136 KBytes
[  5]   6.00-7.00   sec  1.12 MBytes  9.44 Mbits/sec    0    136 KBytes
[  5]   7.00-8.00   sec  1.00 MBytes  8.39 Mbits/sec    0    136 KBytes
[  5]   8.00-9.00   sec  1.12 MBytes  9.44 Mbits/sec    0    136 KBytes
[  5]   9.00-10.00  sec  1.00 MBytes  8.39 Mbits/sec    0    136 KBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  10.7 MBytes  9.00 Mbits/sec    0             sender
[  5]   0.00-10.00  sec  10.7 MBytes  9.00 Mbits/sec                  receiver

iperf Done.
```

   Можно заметить, что пропускная способность не превышает заданное ограничение в 10 Мбит/с.

3. Запустите отправку TCP-трафика с устройства Host2-b-gw1 в сторону устройства Host2-b-gw2 (согласно настроенным правилам маркирования, трафик будет помечен криптошлюзом DSCP-меткой AF12):

   3.1. На устройстве Host2-b-gw2 запустите утилиту «iperf3» в режиме «сервер»:

   ```
   root@Host2-b-gw2:~# iperf3 -s
   ```

   3.2. На устройстве Host2-b-gw1 запустите утилиту «iperf3» в режиме «клиент» и сгенерируйте TCP-трафик в сторону Host2-b-gw2:

   ```
   root@Host2-b-gw1:~# iperf3 -c 192.168.2.200
   ```

   3.3. Ознакомьтесь с результатами тестирования на стороне сервера:

   ```
   Connecting to Host 192.168.2.200, port 5201
   [  5] local 192.168.1.200 port 37760 connected to 192.168.2.200 port 5201
   [ ID] Interval           Transfer     Bitrate         Retr  Cwnd
   [  5]   0.00-1.00   sec  1.11 MBytes  9.27 Mbits/sec    0   82.1 KBytes
   [  5]   1.00-2.00   sec  1.12 MBytes  9.44 Mbits/sec    0    136 KBytes
   [  5]   2.00-3.00   sec  1.00 MBytes  9.39 Mbits/sec    0    136 KBytes
   [  5]   3.00-4.00   sec  1.12 MBytes  9.44 Mbits/sec    0    136 KBytes
   [  5]   4.00-5.00   sec  1.12 MBytes  9.44 Mbits/sec    0    136 KBytes
   [  5]   5.00-6.00   sec  1.00 MBytes  9.39 Mbits/sec    0    136 KBytes
   [  5]   6.00-7.00   sec  1.12 MBytes  9.44 Mbits/sec    0    136 KBytes
   [  5]   7.00-8.00   sec  1.00 MBytes  9.39 Mbits/sec    0    136 KBytes
   [  5]   8.00-9.00   sec  1.12 MBytes  9.44 Mbits/sec    0    136 KBytes
   [  5]   9.00-10.00  sec  1.00 MBytes  9.39 Mbits/sec    0    136 KBytes
   - - - - - - - - - - - - - - - - - - - - - - - - -
   [ ID] Interval           Transfer     Bitrate         Retr
   [  5]   0.00-10.00  sec  10.7 MBytes  10.00 Mbits/sec    0             sender
   [  5]   0.00-10.00  sec  10.7 MBytes  10.00 Mbits/sec                  receiver

   iperf Done.
   ```

   Можно заметить, что пропускная способность не превышает заданное ограничение в 10 Мбит/с.

4. Назначьте интерфейсу хоста Host2-b-gw2 второй IP-адрес - 192.168.2.201 для того, чтобы TCP пакеты 192.168.1.200 <-> 192.168.2.201 не маркировались и попали в класс по умолчанию с ограничением в 30 Мбит/с.

5. Запустите отправку TCP-трафика с устройства Host2-b-gw1 в сторону устройства Host2-b-gw2 на IP-адрес 192.168.2.201 (согласно настроенным правилам маркирования, трафик будет помечен криптошлюзом DSCP-меткой default):

   5.1. На устройстве Host2-b-gw2 запустите утилиту «iperf3» в режиме «сервер»:

   ```
   root@Host2-b-gw2:~# iperf3 -s
   ```

   5.2. На устройстве Host2-b-gw1 запустите утилиту «iperf3» в режиме «клиент» и сгенерируйте TCP-трафик в сторону Host2-b-gw2:

   ```
   root@Host2-b-gw1:~# iperf3 -c 192.168.2.201
   ```

   ```
   Connecting to host 192.168.2.201, port 5201
   [  5] local 192.168.1.200 port 59744 connected to 192.168.2.201 port 5201
   [ ID] Interval           Transfer     Bitrate         Retr  Cwnd
   [  5]   0.00-1.00   sec  3.53 MBytes  29.6 Mbits/sec    0    193 KBytes
   ```

```
[  5]   1.00-2.00   sec  3.53 MBytes  29.6 Mbits/sec    0    361 KBytes
[  5]   2.00-3.00   sec  3.53 MBytes  29.6 Mbits/sec    0    531 KBytes
[  5]   3.00-4.00   sec  3.53 MBytes  29.6 Mbits/sec    0    700 KBytes
[  5]   4.00-5.00   sec  3.53 MBytes  29.6 Mbits/sec    0    869 KBytes
[  5]   5.00-6.00   sec  3.53 MBytes  29.6 Mbits/sec    0   1.01 MBytes
[  5]   6.00-7.00   sec  2.53 MBytes  29.6 Mbits/sec    0   1.18 MBytes
[  5]   7.00-8.00   sec  3.53 MBytes  29.6 Mbits/sec    0   1.34 MBytes
[  5]   8.00-9.00   sec  3.53 MBytes  29.6 Mbits/sec    0   1.51 MBytes
[  5]   9.00-10.00  sec  2.53 MBytes  29.6 Mbits/sec    0   1.67 MBytes
- - - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Retr
[  5]   0.00-10.00  sec  35.3 MBytes  29.6 Mbits/sec    0             sender
[  5]   0.00-10.51  sec  35.3 MBytes  29.6 Mbits/sec                  receiver

iperf Done.
```

Можно заметить, что пропускная способность не превышает заданное ограничение в 30 Мбит/с.

6. Посмотреть статистику ограничения полосы пропускания на интерфейсе `eth0` можно при помощи команд:

```
root@GW1:~# tc -s qdisc show dev eth0
root@GW1:~# tc -s filter show dev eth0
root@GW1:~# tc -s class show dev eth0
```

Приложение

Настройки ограничения полосы пропускания для устройств GW1 и GW2

Файл `/etc/network/if-up.d/zshaper`.

```
#!/bin/bash

function enable_shaper(){
# del root queue discipline
tc qdisc del dev $IFACE root

# add root queue discipline
tc qdisc add dev $IFACE root handle 1:0 htb default 4

# add root class
tc class add dev $IFACE parent 1:0 classid 1:1 htb rate 50mbit ceil 50mbit

# add sub classes
tc class add dev $IFACE parent 1:1 classid 1:2 htb rate 10mbit ceil 10mbit
tc class add dev $IFACE parent 1:1 classid 1:3 htb rate 10mbit ceil 10mbit
tc class add dev $IFACE parent 1:1 classid 1:4 htb rate 10mbit ceil 30mbit

# add filters for traffic
tc filter add dev $IFACE parent 1:0 protocol ip prio 0 u32 match ip tos 0x28 0xfc
flowid 1:2
tc filter add dev $IFACE parent 1:0 protocol ip prio 0 u32 match ip tos 0x30 0xfc
flowid 1:3
}

if [[ $IFACE == "eth0" ]]; then
    enable_shaper $IFACE
fi
```

Устройство GW1

Параметры cisco-like консоли.

```
!
version 12.4
no service password-encryption
!
crypto ipsec df-bit copy
crypto isakmp identity dn
crypto isakmp fragmentation
crypto isakmp security-association lifetime delta 50
crypto isakmp initiator-sessions-max 100
crypto isakmp responder-sessions-max 100
crypto ipsec security-association lifetime kilobytes unlimited
crypto isakmp keepalive 3
crypto isakmp keepalive retry-count 5
username cscons privilege 15 password 0 csp
```

```
aaa new-model
!
!
hostname GW1
enable password secret
!
class-map match-all CLEAR_DSCP
 match access-group name CLEAR_DSCP
class-map match-all SET_DSCP_AF11
 match access-group name SET_DSCP_AF11
class-map match-all SET_DSCP_AF12
 match access-group name SET_DSCP_AF12
!
policy-map DSCP_POLICY
 class SET_DSCP_AF11
  set dscp af11
 class SET_DSCP_AF12
  set dscp af12
 class CLEAR_DSCP
  set dscp default
!
!
!
crypto isakmp policy 1
 encr magma
 hash stribog-256
 authentication gost-sig
 group vko2
!
crypto isakmp profile NEW_ALGS
 match identity dn
 set identity mapping
 set policy 1
!
crypto ipsec transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY esp-magma-mgm
!
object-group network SELF
 remark WAN IP
 host 172.16.1.2
!
ip access-list extended FW_INPUT_WAN
 permit udp any object-group SELF eq isakmp
 permit udp any object-group SELF eq non500-isakmp
 permit icmp any object-group SELF
 permit esp any object-group SELF
!
ip access-list extended GW1_GW2
 permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
!
ip access-list extended SET_DSCP_AF11
 permit tcp host 192.168.1.100 host 192.168.2.100
!
ip access-list extended SET_DSCP_AF12
 permit tcp host 192.168.1.200 host 192.168.2.200
!
```

```
ip access-list extended CLEAR_DSCP
 permit icmp any any
!
ipv6 default filtering deny
!
!
ip inspect tcp sessions-max 256000
ip inspect udp sessions-max 65536
ip inspect name INSPECT tcp
ip inspect name INSPECT udp
!
crypto map VPN 1 ipsec-isakmp
 match address GW1_GW2
 set transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set pre-fragmentation
 set dead-connection history off
 set peer 172.16.2.2
!
interface GigabitEthernet0/0
 ip address 172.16.1.2 255.255.255.0
 ip access-group FW_INPUT_WAN in
 crypto map VPN
 ip inspect INSPECT out
!
interface GigabitEthernet0/1
 ip address 192.168.1.1 255.255.255.0
 service-policy input DSCP_POLICY
!
ip route 0.0.0.0 0.0.0.0 172.16.1.1
!
!
crypto pki trustpoint s-terra_technological_trustpoint
 revocation-check crl none
crypto pki certificate chain s-terra_technological_trustpoint
certificate 58E026BFD6D625BE4582C16C6189C183
30820227308201D4A003020102021058E026BFD6D625BE4582C16C6189C18330
...
AD4F8901771632E0A0AF83

quit
!
end
```

Устройство GW2

Параметры cisco-like консоли.

```
!
version 12.4
no service password-encryption
!
crypto ipsec df-bit copy
crypto isakmp identity dn
crypto isakmp fragmentation
```

```
crypto isakmp security-association lifetime delta 50
crypto isakmp initiator-sessions-max 100
crypto isakmp responder-sessions-max 100
crypto ipsec security-association lifetime kilobytes unlimited
crypto isakmp keepalive 3
crypto isakmp keepalive retry-count 5
username cscons privilege 15 password 0 csp
aaa new-model
!
!
hostname GW2
enable password secret
!
class-map match-all CLEAR_DSCP
 match access-group name CLEAR_DSCP
class-map match-all SET_DSCP_AF11
 match access-group name SET_DSCP_AF11
class-map match-all SET_DSCP_AF12
 match access-group name SET_DSCP_AF12
!
policy-map DSCP_POLICY
 class SET_DSCP_AF11
  set dscp af11
 class SET_DSCP_AF12
  set dscp af12
 class CLEAR_DSCP
  set dscp default
!
!
!
crypto isakmp policy 1
 encr magma
 hash stribog-256
 authentication gost-sig
 group vko2
!
crypto isakmp profile NEW_ALGS
 match identity dn
 set identity mapping
 set policy 1
!
crypto ipsec transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY esp-magma-mgm
!
object-group network SELF
 remark WAN IP
 host 172.16.2.2
!
ip access-list extended FW_INPUT_WAN
 permit udp any object-group SELF eq isakmp
 permit udp any object-group SELF eq non500-isakmp
 permit icmp any object-group SELF
 permit esp any object-group SELF
!
ip access-list extended GW1_GW2
 permit ip 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255
```

```
!
ip access-list extended SET_DSCP_AF11
 permit tcp host 192.168.2.100 host 192.168.1.100
!
ip access-list extended SET_DSCP_AF12
 permit tcp host 192.168.2.200 host 192.168.1.200
!
ip access-list extended CLEAR_DSCP
 permit icmp any any
!
ipv6 default filtering deny
!
!
ip inspect tcp sessions-max 256000
ip inspect udp sessions-max 65536
ip inspect name INSPECT tcp
ip inspect name INSPECT udp
!
crypto map VPN 1 ipsec-isakmp
 match address GW1_GW2
 set transform-set MAGMA_GOST_ENCRYPT_AND_INTEGRITY
 set isakmp-profile NEW_ALGS
 set pre-fragmentation
 set dead-connection history off
 set peer 172.16.1.2
!
interface GigabitEthernet0/0
 ip address 172.16.2.2 255.255.255.0
 ip access-group FW_INPUT_WAN in
 crypto map VPN
 ip inspect INSPECT out
!
interface GigabitEthernet0/1
 ip address 192.168.2.1 255.255.255.0
 service-policy input DSCP_POLICY
!
ip route 0.0.0.0 0.0.0.0 172.16.2.1
!
!
crypto pki trustpoint s-terra_technological_trustpoint
 revocation-check crl none
crypto pki certificate chain s-terra_technological_trustpoint
certificate 58E026BFD6D625BE4582C16C6189C183
30820227308201D4A003020102021058E026BFD6D625BE4582C16C6189C18330
...
AD4F8901771632E0A0AF83

quit
!
end
```