Сетевые интерфейсы в «С-Терра Шлюз»

Назначение документа

Настоящий документ содержит описание логики работы с сетевыми интерфейсами в «С-Терра Шлюз» и «С-Терра Шлюз DP» версии 5.0 (далее Продукт).

Введение

Далее по тексту, в тех случаях, где изложенное одинаково применимо как к «С-Терра Шлюз», так и к «С-Терра Шлюз DP» будет использоваться название «С-Терра Шлюз».

На «С-Терра Шлюз» запрещено несанкционированное изменение состава ПАК. Любые изменения должны быть согласованы с производителем Продукта.

Продукт «С-Терра Шлюз» может иметь произвольное количество сетевых карт (NIC – англ. network interface controller/card), функционирующих, как на шине PCIe, так и на шине USB (USB-модемы, USB-Ethernet адаптеры).

Сетевые карты представлены в операционной системе (далее ОС) физическими сетевыми интерфейсами (например, eth0, usb0). При необходимости может быть создано произвольное количество виртуальных сетевых интерфейсов (например, 802.1Q интерфейс с VLAN ID 123 - eth0.123 или GRE интерфейс – gre1).

Каждая сетевая карта однозначно соответствует физическому сетевому интерфейсу в ОС. Виртуальный интерфейс может соответствовать физическому интерфейсу в ОС (а, следовательно, – сетевой карте), например 802.1Q интерфейс, или группе физических, например Bonding интерфейс, но может и не соответствовать ни одному физическому интерфейсу, например GRE интерфейс. Вопрос соответствия сетевых карт и виртуальных интерфейсов интерфейсам в ОС рассмотрен далее.

Функции создания, удаления и конфигурирования сетевых интерфейсов могут быть осуществлены как средствами ОС так при помощи Продукта с использованием консоли Cisco-like. Особенности создания, удаления и конфигурирования сетевых интерфейсов будут описаны далее.

К процедурам создания, удаления и конфигурирования сетевых интерфейсов нужно подходить крайне ответственно, так как ошибка может привести к утечке конфиденциальной информации или к несанкционированному доступу к защищаемому ресурсу.

В Продукте есть ограничение на общее количество сетевых интерфейсов (по умолчанию от 96 до 576 в зависимости от имен сетевых интерфейсов), процедура увеличения количества описана в главе «Особые случаиОсобые случаи».

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: routing, switching.

© 2003-2025 S-Terra CSP

Доверенные и недоверенные сетевые интерфейсы

В начале эксплуатации «С-Терра Шлюз» все сетевые интерфейсы, в зависимости от того, к какому сегменту сети они будут подключены, необходимо разделить на две группы. К группе недоверенных сетевых интерфейсов должны относиться интерфейсы недоверенного (публичного или WAN) сегмента. К группе доверенных сетевых интерфейсов должны относиться интерфейсы доверенного (защищаемого или LAN) сегмента.

Контролируемые и неконтролируемые сетевые интерфейсы

Помимо разделения сетевых интерфейсов на группы доверенных и недоверенных, существует разделение на два класса сетевых интерфейсов. Разделение интерфейсов на классы происходит с точки зрения наличия или отсутствия возможности подсистем СКЗИ и МЭ Продукта контролировать трафик на них в зависимости от конфигурации интерфейсов. Если подсистемы СКЗИ и МЭ могут контролировать трафик на сетевом интерфейсе, то такой интерфейс будет называться «контролируемым» и, соответственно, относиться к классу контролируемых, в противном случае – «неконтролируемым» и входить в класс неконтролируемых.

Сетевые интерфейсы, подключенные к недоверенным (например, WAN) сетям, не должны переводиться в неконтролируемое состояние.

Таблица 1 показывает возможные состояния интерфейса с точки зрения подконтрольности и допустимые переходы между ними.

Таблица 1. Классификация состояний интерфейса

Группа \ Класс	**Контролируемый**	**Неконтролируемый**
Доверенный	По умолчанию	Переключение по инициативе администратора для выполнения расширенных функций
Недоверенный	По умолчанию	Недопустимо

Настройки сетевых интерфейсов, при которых они могут стать неконтролируемыми с точки зрения подсистем СКЗИ и МЭ Продукта, описаны в главе «Добавление и удаление виртуальных сетевых интерфейсов».

Для «С-Терра Шлюз DP» все интерфейсы, которые не описаны в файле `/opt/VPNagent/etc/ipsm_dpdk.cfg`, считаются неконтролируемыми. Сопоставление интерфейсов в данном файле с сетевыми картами нужно осуществлять на основе адреса PCI шины (параметр `pci_id` в данном конфигурационном файле).

Политика безопасности и карта интерфейсов (файл /etc/ifaliases.cf)

В явном виде политика безопасности (LSP, local security policy) может быть назначена только на те сетевые интерфейсы, которые описаны в файле `/etc/ifaliases.cf`.

В файле `/etc/ifaliases.cf` отражено соответствие имени интерфейса в нотации Cisco и имени интерфейса в нотации Linux. Данный файл также называют картой интерфейсов.

© 2003-2025 S-Terra CSP

При ручном изменении файла `/etc/ifaliases.cf` нужно обязательно пересчитать хеш-сумму данного файла при помощи команды: `integr_mgr calc -f /etc/ifaliases.cf`, иначе не запустится служба `vpngate`.

Соответствие описывается посредством структуры "`interface`" с двумя обязательными полями `name` и `pattern`, где

- `name` – имя интерфейса в нотации Cisco, которое задается в LSP (в атрибуте LogicalName структуры NetworkInterface) и в консоли Cisco-like (список допустимых имен интерфейсов можно посмотреть в консоли Cisco-like при помощи команды «`interface ?`» в режиме конфигурирования);
- `pattern` – шаблон для имени интерфейса в нотации Linux.

При управлении Продуктом через Cisco-like консоль настоятельно не рекомендуется указывать неявные значения для `pattern`, например `eth*`. Данное правило не распространяется на интерфейс `default`. Если `pettern` задан неявно, то не будут работать функции конфигурирования сетевого интерфейса.

Файл `/etc/ifaliases.cf` обязательно должен содержать следующую строку (ее расположение относительно других строк неважно):

```
interface (name="default" pattern="*")
```

Пример:

```
interface (name="GigabitEthernet0/0" pattern="eth0")
interface (name="GigabitEthernet0/1" pattern="eth1")
interface (name="GigabitEthernet0/2" pattern="eth2")
interface (name="default" pattern="*")
```

При задании политики безопасности как посредством консоли Cisco-like, так и при помощи LSP, используется имя сетевого интерфейса в нотации Cisco из файла `/etc/ifaliases.cf`.

Пример для консоли Cisco-like:

```
Hub1#show running-config
```

```
...
interface GigabitEthernet0/0
 ip address 172.16.100.2 255.255.255.0
 ip access-group FW_INPUT_WAN in
 crypto map VPN
 ip inspect INSPECT out
...
```

Пример для LSP:

```
root@Hub1:~# lsp_mgr show
```

```
...
NetworkInterface (
    LogicalName = "GigabitEthernet0/0"
    InputFilter = FilterChain:FW_INPUT_WAN
    OutputFilter = InspectChain:INSPECT
    IPsecPolicy = IPsecPolicy:VPN
)
...
```

Консоль Cisco-like и сетевые интерфейсы

Консоль Cisco-like может работать только с теми сетевыми интерфейсами, которые описаны в файле `/etc/ifaliases.cf` (за исключением интерфейса с именем `default`, он используется для внутренних нужд Продукта). Виртуальные интерфейсы (GREv4, 802.1Q), созданные средствами консоли, автоматически добавляются в данный файл.

Средствами консоли можно посмотреть сводную информацию о сетевых интерфейсах, добавленных в Продукт, при помощи следующих команд:

- `show interfaces`
- `show ip interface`
- `show running-config`

Либо средствами ОС при помощи команд:

- `ip -d -s address`
- `ethtool <linux_iface_name>`
- `ethtool -S <linux_iface_name>`

Настройки сетевых интерфейсов консоль Cisco-like сохраняет в файл `/etc/network/interfaces`, а именно, в секцию, начало и конец которой помечены маркерами `###netifcfg-begin###` и `###netifcfg-end###`, например:

```
root@Hub1:~# cat /etc/network/interfaces
############################################################
# CAUTION: lines under special marker: ###netifcfg-*###
# contains autogenerated information. You can add/modify
# lines outside of those markers
############################################################

# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source /etc/network/interfaces.d/*

# loopback configuration
auto lo
iface lo inet loopback

###netifcfg-begin###

auto eth0
iface eth0 inet static
mtu 1500
address 172.16.100.2
netmask 255.255.255.0
broadcast 172.16.100.255

auto eth1
iface eth1 inet static
mtu 1500
address 192.168.100.1
netmask 255.255.255.0
broadcast 192.168.100.255

iface eth2 inet manual
mtu 1500
###netifcfg-end###
```

Содержимое секции начало и конец которой помечены маркерами `###netifcfg-begin###` и `###netifcfg-end###` является автоматически сгенерированным и запрещено для ручного изменения.

Важно отметить, что консоль, при выводе информации о сетевых интерфейсах по командам show, зачитывает настройки из системы, а не из файла `/etc/network/interfaces`, в данный файл происходит исключительно запись.

Общее описание процедуры добавления и удаления сетевых интерфейсов

Если интерфейс добавлен в Продукт, то данный интерфейс отображается в консоли Cisco-like, а также может быть описан в LSP. К добавленным в Продукт интерфейсам можно применять политику СКЗИ и правила МЭ. Важно понимать, что интерфейс, добавленный в Продукт, может быть неконтролируемым, то есть прикрепление политики СКЗИ и правил МЭ к таким интерфейсам будет возможно, но они не будут выполнены (подробнее см. главу «Добавление и удаление виртуальных сетевых интерфейсов»).

Если интерфейс не добавлен в Продукт, то данный интерфейс не отображается в консоли Cisco-like и не может быть описан в LSP.

Не для всех интерфейсов, добавленных в Продукт, можно изменять сетевые параметры (например MTU/IP адрес/состояние) при помощи Cisco-like консоли (подробнее описано в главе «Добавление и удаление виртуальных сетевых интерфейсов»).

При добавлении и удалении **физических** интерфейсов в/из Продукт(а) нужно использовать исключительно утилиту `netifcfg` (входит в состав Продукта), а для **виртуальных** интерфейсов нужно использовать либо консоль Cisco-like, либо делать это вручную посредством консоли Linux и файла `/etc/ifaliases.cf`. Далее будут описаны способы добавления интерфейсов в Продукт.

Физическим считается интерфейс, который представлен в виде сетевой карты (NIC), например на шине PCIe. Также к физическим относятся USB-модемы (подробнее про использование USB-модемов можно прочитать в документе «Использование USB-модемов в «С-Терра Шлюз»), и USB–Ethernet адаптеры, использующие драйвера CDC Ethernet, Asix AX88772/AX88179_178A, MOSCHIP 7830/7832/7730. Другие типы физических интерфейсов Продуктом не поддерживаются.

Для «С-Терра Шлюз» в виртуализированном исполнении физическими являются интерфейсы, которые указываются в параметрах виртуальной машины (**рекомендуемый** драйвер/тип адаптера `vmxnet3`).

Добавление и удаление физических сетевых интерфейсов

Добавление или удаление физических сетевых интерфейсов в/из Продукта для программно-аппаратных комплексов можно производить только для USB-модемов или Ethernet over USB сетевых карт. Для Продукта в виртуализированном исполнении разрешается добавлять только сетевые интерфейсы, которые поддерживаются гипервизором и разрешены к использованию разработчиком Продукта (см. соответствующую документацию по портале).

Для добавления или удаления **физических** сетевых интерфейсов в Продукт необходимо всегда использовать утилиту `netifcfg`. Утилита `netifcfg` входит в состав Продукта.

Описание утилиты netifcfg

Утилита `netifcfg` имеет множество опций. Посмотреть их можно при помощи команды:

```
root@Hub1:~# netifcfg -h
```

```
usage: netifcfg [-h]
                {enum,show,get-alias,map,add-device,remove-device,delete,delete-
all,save,save-all,version,check,reset}
                ...

positional arguments:
  {enum,show,get-alias,map,add-device,remove-device,delete,delete-all,save,save-
all,version,check,reset}
    enum                Print list of existing network interfaces
    show                show current list of persistent ifnames
                        '/etc/netifaces' and list of aliases
                        '/etc/ifaliases.cf' configuration
    get-alias           get alias name by ifname
    map                 Create new list of persistent ifnames '/etc/netifaces'
                        and list of aliases '/etc/ifaliases.cf' configuration
                        based on input file
    add-device          Add new physical interface to list of persistent
                        ifnames '/etc/netifaces' and list of aliases
                        '/etc/ifaliases.cf'
    remove-device       Remove physical interface from list of persistent
                        ifnames '/etc/netifaces' and list of aliases
                        '/etc/ifaliases.cf'
    delete              Remove interface(s) from network configuration
                        '/etc/network/interfaces'
    delete-all          Delete all interfaces from network configuration
                        '/etc/network/interfaces'
    save                Save current state of network interface(s) to network
                        configuration '/etc/network/interfaces'
    save-all            Save settings of all interfaces to network
                        configuration '/etc/network/interfaces'
    version             Show program version
    check               Check if interface's settings can be applied
    reset               Remove controlled interfaces from network
                        configuration '/etc/network/interfaces'

options:
  -h, --help            show this help message and exit
```

В документе будут рассмотрены только пять опций, а именно `show`, `add-device`, `remove-device`, `enum` и `map`.

Все опции, за исключением `show`, `add-device`, `remove-device`, `enum` и `map` **не предназначены** для применения администратором Продукта и используются для внутренних нужд.

Опция show

Опция `show` показывает информацию о сетевых интерфейсах, которые уже добавлены в Продукт.

Пример вывода:

```
root@Hub1:~# netifcfg show
```

```
INFO: list of persistent ifnames '/etc/netifaces':
pci:
  0000:03:00.0:
    if_name: eth0
    if_alias: GigabitEthernet0/0
  0000:0b:00.0:
    if_name: eth1
    if_alias: GigabitEthernet0/1
  0000:13:00.0:
```

```
    if_name: eth2
    if_alias: GigabitEthernet0/2
hotplug: {}
restricted: {}
INFO: list of aliases '/etc/ifaliases.cf':
GigabitEthernet0/0: eth0
GigabitEthernet0/1: eth1
GigabitEthernet0/2: eth2
default: '*'
```

Опция add-device

Опция `add-device` добавляет физический сетевой интерфейс в Продукт.

Под добавлением физического интерфейса в Продукт подразумевается следующее:

- осуществление привязки **физического** сетевого интерфейса к его PCI шине или USB порту (для USB модемов и Ethernet over USB интерфейсов). Цель данной привязки – обеспечение постоянного соответствия между именем сетевого интерфейса в нотации Linux и его PCI шиной, либо USB портом. Например, без данной привязки интерфейс `eth0` после перезагрузки, при некоторых обстоятельствах, зависящих от настроек ОС, может получить имя `eth1`, что в свою очередь может привести к нарушению политики безопасности из-за того, что политика СКЗИ или правила МЭ были прикреплены к `eth0`.
- внесение соответствующих изменений в карту интерфейсов (файл `/etc/ifaliases.cf`).

Параметры опции:

```
root@Hub1:~# netifcfg add-device -h
```

```
usage: netifcfg add-device [-h] [-n new_name] [-c ciscolike_alias] [-r] [-q]
                           [--use-macaddr]
                           ifname

positional arguments:
  ifname                Name of network inteface

options:
  -h, --help            show this help message and exit
  -n new_name, --name new_name
                        Rename interface to the specified name
  -c ciscolike_alias, --ciscolike ciscolike_alias
                        Use new_alias as Cisco-like name
  -r, --rename          Rename interface to expected name immediately
                        (Warning: interfaces will be brought down in the
                        process!)
  -q, --quiet           do not output info messages
  --use-macaddr         Add new hotplug interfaces using the current mac
                        address as a unique identifier instead of PCI ID.
```

Описание обязательных параметров:

- `ifname` – имя сетевого интерфейса, который требуется добавить в продукт, в нотации Linux.

Описание опциональных параметров:

- `-n` – новое имя сетевого интерфейса в нотации Linux;
- `-c` – имя сетевого интерфейса в нотации Cisco;
- `-r` – переименование сетевого интерфейса в нотации Linux при добавлении (без этой опции интерфейсу будет назначено имя, указанное в опции `-n`, при перезагрузке ОС;
- `-q` – не выводить информационные сообщения;

© 2003-2025 S-Terra CSP

- `--use-macaddr` – внутренний параметр, запрещен для использования.

Пример (добавление сетевого интерфейса в Продукт):

```
root@Hub1:~# netifcfg add-device unknwn_eth0 -n eth3 -r -c GigabitEthernet0/3
```

Неизвестные Продукту сетевые интерфейсы получают префикс «`unknwn_`».

```
INFO: Renamed interface unknwn_eth0 to eth3
INFO: Added new interface 'eth3' with alias 'GigabitEthernet0/3'
INFO: to both list of persistent ifnames '/etc/netifaces' and list of aliases
'/etc/ifaliases.cf'
INFO: Operation completed successfully
```

Опция remove-device

Опция `remove-device` удаляет физический сетевой интерфейс из Продукта.

Параметры опции:

```
root@Hub1:~# netifcfg remove-device -h
```

```
usage: netifcfg remove-device [-h] [-q] ifname

positional arguments:
  ifname       Name of network inteface

options:
  -h, --help   show this help message and exit
  -q, --quiet  do not print info messages
```

Описание обязательных параметров:

- `ifname` – имя сетевого интерфейса, который требуется удалить из Продукта, в нотации Linux.

Описание опциональных параметров:

- `-q` – не выводить информационные сообщения.

Пример (удаление сетевого интерфейса из Продукта):

```
root@Hub1:~# netifcfg remove-device eth3
```

```
INFO: Removed interface 'eth3' from list of persistent ifnames '/etc/netifaces'
INFO: Removed interface 'eth3' from list of aliases '/etc/ifaliases.cf'
INFO: Operation completed successfully
```

Опция enum

Опция `enum` собирает и анализирует информацию о всех **физических** сетевых интерфейсах и генерирует специальное описание, которое после анализа администратором подается на вход данной утилите с опцией `map` с целью добавления всех найденных физических интерфейсов в Продукт за один раз.

В выводе опции enum могут присутствовать интерфейсы, которые еще не добавлены в Продукт, для просмотра добавленных интерфейсов нужно пользоваться опцией show.

Рекомендуется пользоваться данной опцией исключительно для информационных целей.

Пример вывода `netifcfg enum`:

```
root@Hub1:~# netifcfg enum
```

```
pci:
  0000:03:00.0:
```

```
    if_name: eth0
    if_alias: GigabitEthernet0/0
  0000:0b:00.0:
    if_name: eth1
    if_alias: GigabitEthernet0/1
  0000:13:00.0:
    if_name: eth2
    if_alias: GigabitEthernet0/2
hotplug:
  pci-0000:00:16.0-usb-0:1.1:1.0:
    if_name: usb0
    if_alias: UsbEthernet0
restricted:
  0000:04:00.0:{}
  0000:05:00.0:{}
```

- В секции `pci` располагаются сетевые интерфейсы на шине PCI. Строки вида «`0000:XX:YY.Z`» являются адресом.
- В секции `hotplug` располагаются сетевые интерфейсы на шине USB.
- В секции `restrictred` располагаются сетевые интерфейсы на шине PCI, которые отданы под работу «С-Терра Шлюз DP» (интерфейсы с именем `vEthN`).

Пример вывода `netifcfg enum` в формате версии 4.3:

```
root@Hub1:~# netifcfg enum -t
```

```
#Unique_ID                      iface_type OS_name          Cisco-like_name
0000:00:14.0                    pci        eth0             GigabitEthernet0/0
0000:00:14.1                    pci        eth1             GigabitEthernet0/1
0000:00:14.2                    pci        eth2             GigabitEthernet0/2
0000:00:14.3                    pci        eth3             GigabitEthernet0/3
pci-0000:00:16.0-usb-0:1.1:1.0  hotplug    usb0             UsbEthernet0
```

Из примера выше видно, что вывод состоит из четырех колонок:

- `Unique_ID` – уникальный идентификатор сетевого интерфейса (адрес PCI шины, либо адрес USB порта);
- `Iface_type` – тип сетевого интерфейса (pci – физический интерфейс на PCI шине, hotplug – физический интерфейс на USB шине);
- `OS_name` – имя сетевого интерфейса в нотации Linux (имена интерфейсов в нотации Linux можно посмотреть при помощи команды `ip link show`);
- `Cisco-like_name` – имя сетевого интерфейса в нотации Cisco.

Опция map

Опция `map` для своей работы использует вывод утилиты `netifcfg`, запущенной с опцией `enum` и предназначена для разового добавления всех физических сетевых интерфейсов в Продукт. Данную опцию можно применять только на виртуализированных «С-Терра Шлюз» до прохождения инициализации и только, если по каким-то объективным причинам не устраивает результат автоматического добавления сетевых интерфейсов в процессе инициализации.

Пример №1. Добавление физического сетевого интерфейса в Продукт

Исходное состояние – инициализированный «С-Терра Шлюз» в боевой эксплуатации.

Ниже приводится пример добавления сетевого интерфейса в Продукт.

1. Добавьте сетевой интерфейс в настройках виртуальной машины для соответствующего гипервизора или подключите USB-модем, или Ethernet over USB адаптер в USB порт.

Для виртуализированного шлюза рекомендуется добавлять сетевые адаптеры только одного типа. Например, для гипервизора vSphere ESXi нужно использовать адаптеры с типом `vmxnet3`.

2. Войдите в Linux bash.
3. Посмотрите текущий список всех сетевых интерфейсов в нотации Linux:

```
root@sterragate:~# ip link show
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
group default qlen 10000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
2: unknwn_eth0: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode
DEFAULT group default qlen 1000
    link/ether 00:50:56:9e:4c:a5 brd ff:ff:ff:ff:ff:ff
    altname enp27s0
    altname ens256
3: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode
DEFAULT group default qlen 10000
    link/ether 00:50:56:9e:f3:7c brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    altname ens160
    altname GigabitEthernet0_0
4: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group
default qlen 1000
    link/ether 00:50:56:9e:d7:87 brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    altname ens192
    altname GigabitEthernet0_1
5: eth2: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group
default qlen 1000
    link/ether 00:50:56:9e:cd:c0 brd ff:ff:ff:ff:ff:ff
    altname enp19s0
    altname ens224
    altname GigabitEthernet0_2
```

Видно, что новый сетевой интерфейс добавился под именем **unknwn_eth0** (префикс unknwn говорит о том, что драйвером данного сетевого адаптера была предпринята попытка захвата имени `eth0`, но сработала защита, так как физический сетевой интерфейс ранее при помощи утилиты `netifcfg` был добавлен в Продукт (установлено соответствие между `eth0` и его адресом PCI шины или USB порта). Если бы попытки захвата не было, то в текущем примере интерфейс добавился бы с именем `eth3` (так как последний физический интерфейс до процедуры добавления имел имя `eth2`).

4. Добавьте сетевой интерфейс в Продукт под именем `GigabitEthernet0/3` с переименованием его из `unknwn_eth0` в `eth3`:

```
root@Hub1:~# netifcfg add-device unknwn_eth0 -n eth3 -r -c GigabitEthernet0/3
```

```
INFO: Renamed interface unknwn_eth0 to eth3
INFO: Added new interface 'eth3' with alias 'GigabitEthernet0/3'
INFO: to both list of persistent ifnames '/etc/netifaces' and list of aliases
'/etc/ifaliases.cf'
INFO: Operation completed successfully
```

В случае, если попытки захвата имени не было, и новый сетевой интерфейс добавился в систему под именем `eth3`, то добавьте его в Продукт без переименования:

```
root@Hub1:~# netifcfg add-device -c GigabitEthernet0/3 eth3
```

```
INFO: Added new interface 'eth3' with alias 'GigabitEthernet0/3'
```

```
INFO: to both list of persistent ifnames '/etc/netifaces' and list of aliases
'/etc/ifaliases.cf'
INFO: Operation completed successfully
```

5. Посмотрите на результат добавления:

```
root@sterragate:~# netifcfg show
```

```
INFO: list of persistent ifnames '/etc/netifaces':
pci:
  0000:03:00.0:
    if_name: eth0
    if_alias: GigabitEthernet0/0
  0000:0b:00.0:
    if_name: eth1
    if_alias: GigabitEthernet0/1
  0000:13:00.0:
    if_name: eth2
    if_alias: GigabitEthernet0/2
  0000:1b:00.0:
    if_name: eth3
    if_alias: GigabitEthernet0/3
hotplug: {}
restricted: {}
INFO: list of aliases '/etc/ifaliases.cf':
GigabitEthernet0/0: eth0
GigabitEthernet0/1: eth1
GigabitEthernet0/2: eth2
default: '*'
GigabitEthernet0/3: eth3
```

Видно, что процедура добавления выполнена успешно.

6. Перезагрузитесь:

```
root@sterragate:~# reboot
```

Пример №2. Использование enum и map на неинициализированном шлюзе в виртуализированном исполнении

Исходное состояние – неинициализированный «С-Терра Шлюз» после первого запуска.

Если шлюз находится в боевой эксплуатации и необходимо добавить новый сетевой интерфейс, то обратитесь к описанию опции `add-device`.

Для версии 4.3 и выше добавление физических сетевых интерфейсов будет выполнено в автоматическом режиме во время инициализации Продукта (при условии, что интерфейсы не были добавлены ранее).

Ниже приводится пример добавления сетевых интерфейсов в Продукт до инициализации.

1. Войдите в Linux bash.
2. Посмотрите текущий список всех сетевых интерфейсов в нотации Linux (физические интерфейсы будут иметь имена, соответствующие шаблонам `eth<N>` и `usb<N>`):

```
root@sterragate:~# ip link show
```

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN mode DEFAULT
group default qlen 10000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
```

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP mode
DEFAULT group default qlen 10000
    link/ether 00:50:56:9e:f3:7c brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    altname ens160
3: eth1: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group
default qlen 1000
    link/ether 00:50:56:9e:d7:87 brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    altname ens192
4: eth2: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state DOWN mode DEFAULT group
default qlen 1000
    link/ether 00:50:56:9e:cd:c0 brd ff:ff:ff:ff:ff:ff
    altname enp19s0
    altname ens224
```

3. Сгенерируйте описание для сетевых интерфейсов и убедитесь, что в списке присутствуют все физические сетевые интерфейсы:

```
root@sterragate:~# netifcfg enum > map.txt
root@sterragate:~# cat map.txt
```

```
pci:
  0000:03:00.0:
    if_name: eth0
    if_alias: GigabitEthernet0/0
  0000:0b:00.0:
    if_name: eth1
    if_alias: GigabitEthernet0/1
  0000:13:00.0:
    if_name: eth2
    if_alias: GigabitEthernet0/2
hotplug: {}
restricted: {}
```

Так как файл map.txt будет подан на вход утилите netifcfg с опцией map, то менять содержимое данного файла запрещено (чтобы исключить пользовательскую ошибку).

4. Добавьте физические сетевые интерфейсы в Продукт и перезагрузитесь:

```
root@sterragate:~# netifcfg map map.txt
```

```
INFO: Operation completed successfully. Restart your system to apply the changes
```

```
root@sterragate:~# reboot
```

5. Убедитесь, что файл `/etc/ifaliases.cf` создан и соответствует выводу `netifcfg enum`:

```
root@sterragate:~# cat /etc/ifaliases.cf
```

```
interface (name="GigabitEthernet0/0" pattern="eth0")
interface (name="GigabitEthernet0/1" pattern="eth1")
interface (name="GigabitEthernet0/2" pattern="eth2")
interface (name="default" pattern="*")
```

Автоматически сгенерированные строки для физических интерфейсов в данном файле менять запрещено. Можно добавлять свои, но только для **виртуальных интерфейсов**.

Если «С-Терра Шлюз» в аппаратном исполнении, то добавление сетевых интерфейсов в Продукт осуществляется на производстве компании ООО «С-Терра СиЭсПи». Для виртуализированных шлюзов добавление сетевых интерфейсов в Продукт не осуществляется, так как заранее неизвестно количество сетевых интерфейсов и их тип.

Добавление и удаление виртуальных сетевых интерфейсов

Продукт поддерживает множество виртуальных интерфейсов, но разрешенными к использованию считаются следующие (использование интерфейсов, не вошедших в разрешенный список, должно быть согласовано с ООО «С-Терра СиЭсПи»):

1. 802.1Q (VLAN) интерфейсы;
2. GRE IPv4 и GRE IPv6 интерфейсы;
3. Bonding (port channel) интерфейсы;
4. Loopback интерфейсы;
5. Bridge интерфейсы;
6. stVTI интерфейсы (настройка не описывается в данном документе);
7. VXLAN интерфейсы (настройка не описывается в данном документе).

Начиная с версии 5.0 появилась возможность добавить в Продукт средствами консоли cisco-like следующие виртуальные интерфейсы:

- VLAN (802.1Q) интерфейс на физическом сетевом интерфейсе (FastEthernet, GigabitEthernet, TenGigabitEthernet).
- GRE для версии IP v4 в режиме точка-точка (point-to-point) и точка-многоточка (point-to-multipoint).

При добавлении виртуальных интерфейсов через консоль cisco-like, не требуется вручную вносить изменения в какие-либо файлы.

При добавлении виртуальных интерфейсов через консоль Linux, **не требуется** использовать утилиту `netifcfg`. Достаточно создать интерфейс и внести его в файл `/etc/ifaliases.cf`.

Предупреждение. Если при создании интерфейсов получено сообщение:

```
sterragate systemd-udevd[1221]: gre1: Could not set Alias=, MACAddress= or MTU=,
ignoring: Invalid argument   Waiting for DAD... Done
```

это не считается ошибкой, а следует считать особенностью.

Особенности порядка включения сетевых интерфейсов

По умолчанию, сетевые интерфейсы, которые описываются в конфигурационных файлах в директории `/etc/network/interfaces.d/`, включаются до интерфейсов, описанных в `/etc/network/interfaces`. Интерфейсы, описанные в конфигурационных файлах в директории `/etc/network/interfaces.d/` включаются в лексикографическом порядке (по именам файлов).

В связи с этим следует не допускать ситуации, когда некоторые интерфейсы (например, VRF или Bridge) при поднятии включают в себя интерфейсы, которые на этот момент могут еще не существовать в системе (например, VTI или VLAN), так как это приведет к ошибкам конфигурации.

При необходимости, можно в файле `/etc/network/interfaces` строку `source /etc/network/interfaces.d/*` перенести в конец файла (ниже секции, обозначенной строками `###netifcfg-begin###` и `###netifcfg-end###`), также можно перенести описание интерфейса из директории `/etc/network/interfaces.d/` в файл `/etc/network/interfaces` (ниже секции, обозначенной строками `###netifcfg-begin###` и `###netifcfg-end###`).

© 2003-2025 S-Terra CSP

802.1Q (VLAN) интерфейсы

Интерфейсы 802.1Q являются виртуальными сетевыми интерфейсами. Данные интерфейсы работают в соответствии со стандартом IEEE 802.1Q.

Для «С-Терра Шлюз DP» настройка VLAN для интерфейсов, описанных в `/opt/VPNagent/etc/ipsm_dpdk.cfg` осуществляется через данный файл (параметр `vlan_identifier`).

Пример создания через консоль cisco-like

1. Создание.

   Ниже описан пример создания 802.1Q интерфейса с VLAN ID 4001 на GigabitEthernet0/1 (eth1). Интерфейс, на котором создается VLAN, должен быть включен.

   1.1. Войдите в консоль cisco-like и перейдите в режим конфигурирования:

   ```
   root@Hub1:~# su cscons
   Hub1#configure terminal
   Hub1(config)#
   ```

   1.2. Создайте и настройте VLAN интерфейс с ID 4001 на GigabitEthernet0/1:

   ```
   Hub1(config)#interface gigabitEthernet 0/1.4001
   Hub1(config-if)#ip address 192.168.3.1 255.255.255.0
   Hub1(config-if)#no shutdown
   Hub1(config-if)#end
   ```

   1.3. Убедитесь, что в системе появился соответствующий сетевой интерфейс:

   ```
   Hub1#run ip link
   ...
   8: eth1_4001@eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP mode DEFAULT group default qlen 10000
       link/ether 00:50:56:9e:d1:c0 brd ff:ff:ff:ff:ff:ff
   ...
   ```

   Обратите внимание, что VLAN интерфейсы, созданные консолью cisco-like, имеют следующий формат ethN_VLANID, а созданные консолью Linux – ethN.VLANID. Это сделано специально, чтобы визуально можно было отличить посредством чего был создан данный интерфейс.

   1.4. Убедитесь, что настройки интерфейса автоматически прописались в файл /etc/network/interfaces:

   ```
   Hub1#run cat /etc/network/interfaces
   ...
   auto eth1_4001
   iface eth1_4001 inet manual
   st_type VLAN
   st_state up
   st_parent_if eth1
   st_vlan_id 4001
   st_mtu 1500
   address 192.168.3.1
   netmask 255.255.255.0
   broadcast 192.168.3.255
   ...
   ```

Теперь интерфейс GigabitEthernet0/1.4001 должен быть доступен к заданию политики СКЗИ и правил МЭ при помощи консоли cisco-like и LSP.

2. Удаление.

2.1. Для удаления нужно выполнить следующую команду в консоли cisco-like:

```
Hub1#configure terminal
Hub1(config)#no interface gigabitEthernet 0/1.4001
Hub1(config)#end
```

Пример создания через консоль Linux

1. Создание.

802.1Q интерфейсы описываются в файлах /etc/network/interfaces.d/ethN.VLANID.

Ниже описан пример создания 802.1Q интерфейса с VLAN ID 4000 на eth1. Интерфейс, на котором создается VLAN, должен быть включен.

1.1. Отредактируйте файл /etc/network/interfaces.d/eth1.4000, добавив в него описание для 802.1Q интерфейса с VLAN ID 4000, например:

```
root@Hub1:~# vim.tiny /etc/network/interfaces.d/eth1.4000
```

```
auto eth1.4000
iface eth1.4000 inet manual
mtu 1500
```

Значение MTU для 802.1Q интерфейса не может быть больше, чем значение MTU соответствующего физического интерфейса.

1.2. Чтобы 802.1Q интерфейс eth1.4000 появился в системе, выполните команду ifup eth1.4000, после чего убедитесь, что данный интерфейс действительно доступен в системе:

```
root@Hub1:~# ifup eth1.4000
```

Если команда ifup завершилась с ошибкой, то ее можно запустить повторно с флагом –v для локализации проблемы, например: ifup –v <iface>.

```
root@Hub1:~# ip link show eth1.4000
```

```
5: eth1.4000@eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
state UP mode DEFAULT group default qlen 10000
    link/ether 00:50:56:9e:d1:c0 brd ff:ff:ff:ff:ff:ff
```

Видно, что 802.1Q интерфейс eth1.4000 присутствует в системе.

1.3. Добавьте 802.1Q интерфейс eth1.4000 в файл /etc/ifaliases.cf (выделено жирным):

```
root@Hub1:~# vim.tiny /etc/ifaliases.cf
```

```
interface (name="GigabitEthernet0/0" pattern="eth0")
interface (name="GigabitEthernet0/1" pattern="eth1")
interface (name="GigabitEthernet0/1.4000" pattern="eth1.4000")
interface (name="GigabitEthernet0/2" pattern="eth2")
interface (name="default" pattern="*")
```

- значение параметра `name` должно иметь следующий шаблон:

```
<имя физ. интерфейса в нотации Cisco>.<VLAN ID>
```

- значение параметра `pattern` должно совпадать с именем 802.1Q интерфейса в нотации Linux.

1.4. Пересчитайте контрольную сумму файла /etc/ifaliases.cf:

```
root@Hub1:~# integr_mgr calc -f /etc/ifaliases.cf
```

```
SUCCESS:  Operation was successful.
```

1.5. Перезапустите VPN сервис:

```
root@Hub1:~# systemctl restart vpngate.service
```

1.6. Проверьте наличие интерфейса в cisco-like консоли:

```
Hub1#show run
```

```
...
interface GigabitEthernet0/1.4000
 no ip address
!
...
```

Теперь интерфейс GigabitEthernet0/1.4000 должен быть доступен к заданию политики СКЗИ и правил МЭ при помощи консоли cisco-like и LSP. **Конфигурирование при помощи консоли cisco-like будет невозможно**.

2. Изменение параметров.

2.1. Выключите интерфейс ifdown ethN.VLANID, отредактируйте файл конфигурации /etc/network/interfaces.d/ethN.VLANID и включите интерфейс ifup ethN.VLANID.

3. Удаление.

3.1. Для удаления интерфейса нужно выключить интерфейс ifdown ethN.VLANID, удалить файл конфигурации /etc/network/interfaces.d/ethN.VLANID и удалить соответствующую строчку из /etc/ifaliases.cf (пересчитав контрольную сумму), перезапустить VPN сервис.

GRE интерфейсы

Интерфейсы GRE являются виртуальными сетевыми интерфейсами. Данные интерфейсы работают в соответствии со стандартом RFC 1701. Используются для организации Route Based VPN (поддерживают передачу Multicast трафика).

Интерфейсы GRE, настроенные через консоль cisco-like и консоль Linux, совместимы между собой.

В сценарии GRE over IPsec, при использовании динамических протоколов маршрутизации, например OSPF, в случае сбоя, создаются условия для утечки конфиденциальной информации. Чтобы избежать утечки, необходимо на сетевых интерфейсах из недоверенного сегмента заблокировать при помощи МЭ тот трафик, который должен быть инкапсулирован в GRE (трафик защищаемых подсетей). При невыполнении данной рекомендации утечка произойдет в том случае, если нарушится работа динамического протокола маршрутизации и маршрут через GRE интерфейс будет удален. В этом случае, трафик не будет инкапсулирован в GRE и, следовательно, не попадет под фильтр шифрования (который в данном сценарии, как правило, настраивается только на GRE).

Point-to-point GRE

Пример создания через консоль cisco-like

IPv4 over IPv4

1. Создание.

Ниже описан пример создания point-to-point GRE IP v4 over IP v4 интерфейса. Консоль cisco-like поддерживает только GRE для протокола IP v4.

1.1. Войдите в консоль cisco-like и перейдите в режим конфигурирования:

```
root@Hub1:~# su cscons
Hub1#configure terminal
Hub1(config)#
```

1.2. Создайте и настройте point-to-point GRE интерфейс Tunnel 0:

```
Hub1(config)#interface Tunnel 0
Hub1(config-if)#ip address 10.10.10.1 255.255.255.0
Hub1(config-if)#tunnel key 1
Hub1(config-if)#tunnel destination 172.16.1.2
Hub1(config-if)#no tunnel path-mtu-discovery
Hub1(config-if)#tunnel ignore-df
Hub1(config-if)#no shutdown
Hub1(config-if)#end
```

- Если параметр tunnel source не указан, то source адрес будет выбран на основании маршрутной информации до адреса в tunnel destination (можно воспользоваться командой run ip route get <адрес в tunnel destination> для выяснения source адреса).
- Параметры no tunnel path-mtu-discovery и tunnel ignore-df отключают работу PMTUd и копирование DF бита на GRE интерфейсах (это нужно для увеличения вероятности прохождения трафика, в случае его последующего шифрования).

1.3. Просмотрите текущую конфигурацию созданного интерфейса:

```
Hub1#show running-config
...
interface Tunnel0
 ip address 10.10.10.1 255.255.255.0
 mtu 1400
 tunnel key 1
 tunnel destination 172.16.1.2
 offloading tx-checksumming
 no tunnel path-mtu-discovery
 tunnel ignore-df
 tunnel ttl inherit
 tunnel tos 0
...
```

1.4. Убедитесь, что настройки интерфейса автоматически прописались в файл /etc/network/interfaces:

```
Hub1#run cat /etc/network/interfaces
...
auto gre_0
iface gre_0 inet manual
st_type GRE
st_state up
st_key 1
st_remote 172.16.1.2
st_txchecksumoffload enabled
st_gre_ttl inherit
st_gre_pmtudisc off
st_gre_ignore_df on
st_mtu 1400
address 10.10.10.1
netmask 255.255.255.0
broadcast 10.10.10.255
```

```
...
```

Обратите внимание, что GRE интерфейсы, созданные консолью cisco-like, имеют следующий формат gre_N, а созданные консолью Linux – greN. Это сделано специально, чтобы визуально можно было отличить посредством чего был создан данный интерфейс.

1.5. Убедитесь, что в системе появился соответствующий сетевой интерфейс:

```
Hub1#run ip -d link show dev gre_0
```

```
8: gre_0@NONE: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1400 qdisc noqueue state
UNKNOWN mode DEFAULT group default qlen 10000
    link/gre 0.0.0.0 peer 172.16.1.2 promiscuity 0  allmulti 0 minmtu 0 maxmtu 0
    gre remote 172.16.1.2 local any ttl inherit nopmtudisc ignore-df ikey
0.0.0.1 okey 0.0.0.1 addrgenmode random numtxqueues 1 numrxqueues 1 gso_max_size
65536 gso_max_segs 65535 tso_max_size 65536 tso_max_segs 65535 gro_max_size
65536
```

Теперь интерфейс Tunnel0 должен быть доступен к заданию политики СКЗИ и правил МЭ при помощи консоли cisco-like и LSP.

Для сценария IPsec over GRE (IPsec внутри GRE) политика прикрепляется непосредственно на GRE интерфейс, для сценария GRE over IPsec (GRE внутри IPsec) политика СКЗИ должна быть прикреплена к тому WAN интерфейсу, через который будут маршрутизироваться GRE пакеты с GRE интерфейса.

Для правильной работы сценария IPsec over GRE для версии Продукта младше, чем 5.0.25757, требуется задать следующее правило iptables для сброса служебных меток с пакетов:
`iptables -t mangle -A PREROUTING -j MARK --set-mark 0x00000000/0x80000000`
Для того, чтобы сохранить добавленное правило, необходимо обратиться к документу «Инструкция по настройке NAT и межсетевого экрана на «С-Терра Шлюз» при помощи утилиты «iptables».

2. Удаление.

2.1. Для удаления нужно выполнить следующую команду в консоли cisco-like:

```
Hub1#configure terminal
Hub1(config)#no interface Tunnel0
Hub1(config)#end
```

Пример создания через консоль Linux

IP v4 over IP v4

1. Создание.

Ниже описан пример создания point-to-point GRE IP v4 over IP v4 интерфейса. Point-to-point GRE интерфейсы описываются в файлах /etc/network/interfaces.d/greN.

Cisco-like консоль не поддерживает изменение параметров GRE интерфейса, созданного через консоль Linux.

1.1. Отредактируйте файл `/etc/network/interfaces.d/gre1`, добавив в него описание для point-to-point GRE интерфейса, например:

Использовать в качестве имени GRE интерфейса `gre0` запрещено, так как `gre0` используется для внутренних нужд драйвера GRE. В имени GRE интерфейса нельзя использовать точку.

```
root@Hub1:~# vim.tiny /etc/network/interfaces.d/gre1
```

```
auto gre1
iface gre1 inet static
address 10.10.10.1
netmask 255.255.255.0
pre-up ip link del gre1 2> /dev/null || true
pre-up ip tunnel add gre1 mode gre local 172.16.100.2 remote 172.16.1.2 ttl 64
tos inherit key 1
pre-up ip link set gre1 mtu 1400
pre-up ip link set gre1 multicast on
pre-up ip link set gre1 type gre nopmtudisc
pre-up ip link set gre1 type gre ignore-df
post-down ip link del gre1
```

- Если в строке «pre-up ip tunnel add gre1 mode gre …» не указан параметр local <IP>, то source адрес будет выбран в соответствии с таблицей маршрутизации (можно воспользоваться командой ip route get <адрес в remote> для выяснения source адреса).
- Перечисленные ниже параметры отключают работу PMTUd и копирование DF бита на GRE интерфейсах (это нужно для увеличения вероятности прохождения трафика, в случае его последующего шифрования):

```
pre-up ip link set gre1 type gre nopmtudisc
pre-up ip link set gre1 type gre ignore-df
```

1.2. Для того, чтобы GRE интерфейс gre1 появился в системе, выполните команду ifup gre1, после чего убедитесь, что данный интерфейс действительно доступен в системе:

```
root@Hub1:~# ifup gre1
```

Если команда ifup завершилась с ошибкой, то ее можно запустить повторно с флагом –v для локализации проблемы, например: ifup –v <iface>.

```
root@Hub1:~# ip -d link show dev gre1
```

```
20: gre1@NONE: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1400 qdisc noqueue
state UNKNOWN mode DEFAULT group default qlen 10000
    link/gre 172.16.100.2 peer 172.16.1.2 promiscuity 0  allmulti 0 minmtu 68
maxmtu 65507
    gre remote 172.16.1.2 local 172.16.100.2 ttl 64 tos inherit nopmtudisc
ignore-df ikey 0.0.0.1 okey 0.0.0.1 addrgenmode random numtxqueues 1 numrxqueues
1 gso_max_size 65536 gso_max_segs 65535 tso_max_size 65536 tso_max_segs 65535
gro_max_size 65536
```

Вывод свидетельствует о том, что GRE интерфейс gre1 присутствует в системе и сконфигурирован.

1.3. Добавьте GRE интерфейс gre1 в файл `/etc/ifaliases.cf` (выделено жирным):

```
root@Hub1:~# vim.tiny /etc/ifaliases.cf
```

```
interface (name="GigabitEthernet0/0" pattern="eth0")
interface (name="GigabitEthernet0/1" pattern="eth1")
interface (name="GigabitEthernet0/2" pattern="eth2")
interface (name="Tunnel1" pattern="gre1")
interface (name="default" pattern="*")
```

- значение параметра `name` должно быть задано в виде TunnelN.

- значение параметра `pattern` должно соответствовать имени GRE интерфейса в нотации Linux.

1.4. Пересчитайте контрольную сумму файла `/etc/ifaliases.cf`:

```
root@Hub1:~# integr_mgr calc -f /etc/ifaliases.cf

SUCCESS:  Operation was successful.
```

1.5. Перезапустите VPN сервис:

```
root@Hub1:~# systemctl restart vpngate.service
```

Теперь интерфейс `Tunnel1` должен быть доступен к заданию политики СКЗИ и правил МЭ при помощи консоли cisco-like и LSP, а также к конфигурированию средствами ОС.

Для сценария IPsec over GRE (IPsec внутри GRE) политика прикрепляется непосредственно на GRE интерфейс, для сценария GRE over IPsec (GRE внутри IPsec) политика СКЗИ должна быть прикреплена к тому WAN интерфейсу, через который будут маршрутизироваться GRE пакеты с GRE интерфейса.

2. Изменение параметров.

2.1. Выключите интерфейс ifdown greN, отредактируйте файл конфигурации /etc/network/interfaces.d/greN и включите интерфейс ifup greN.

3. Удаление.

3.1. Для удаления интерфейса нужно выключить интерфейс ifdown greN, удалить файл конфигурации /etc/network/interfaces.d/greN, и удалить соответствующую строчку из `/etc/ifaliases.cf` (пересчитав контрольную сумму), после чего перезапустить VPN сервис.

IP v4 over IP v6

Для «С-Терра Шлюз» в виртуализированном исполнении тип сетевого адаптера виртуальной машины должен быть vmxnet3. При использовании другого типа адаптера\драйвера будут портиться IPv6 пакеты.

1. Создание.

Ниже описан пример создания point-to-point GRE IP v4 over IP v6 интерфейса. Point-to-point GRE интерфейсы описываются в файлах /etc/network/interfaces.d/greN.

Cisco-like консоль не поддерживает изменение параметров GRE интерфейса, созданного через консоль Linux.

1.1. Отредактируйте файл `/etc/network/interfaces.d/gre1`, добавив в него описание для point-to-point GRE интерфейса, например:

Использовать в качестве имени GRE интерфейса `gre0` запрещено, так как `gre0` используется для внутренних нужд драйвера GRE. В имени GRE интерфейса нельзя использовать точку.

```
root@Hub1:~# vim.tiny /etc/network/interfaces.d/gre1
```

```
auto gre1
iface gre1 inet static
address 10.10.10.1/24
```

```
pre-up ip link del gre1 2> /dev/null || true
pre-up ip tunnel add gre1 mode ip6gre key 1 ttl 64 tos inherit remote
2001:db8::172:16:100:2
pre-up ip link set gre1 mtu 1400
pre-up ip link set gre1 multicast on
post-up ethtool -K gre1 tx-checksumming off > /dev/null
post-down ip link del gre1
```

- Если в строке «pre-up ip tunnel add gre1 mode ip6gre …» не указан параметр local <IP>, то source адрес будет выбран в соответствии с таблицей маршрутизации (можно воспользоваться командой ip route get <адрес в remote> для выяснения source адреса).

1.2. Для того, чтобы GRE интерфейс gre1 появился в системе, выполните команду ifup gre1, после чего убедитесь, что данный интерфейс действительно доступен в системе:

```
root@Hub1:~# ifup gre1
root@Hub1:~# ip -d link show dev gre1
```

```
16: gre1@NONE: <MULTICAST,NOARP,UP,LOWER_UP> mtu 1400 qdisc noqueue state
UNKNOWN mode DEFAULT group default qlen 10000
    link/gre6 :: brd 2001:db8::172:16:100:2 permaddr 36d2:5e00:2eb3::
promiscuity 0 allmulti 0 minmtu 0 maxmtu 0
    ip6gre remote 2001:db8::172:16:100:2 local any hoplimit 64 encaplimit 4
tclass inherit flowlabel 0x00000 ikey 0.0.0.1 okey 0.0.0.1 addrgenmode random
numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs 65535 tso_max_size
65536 tso_max_segs 65535 gro_max_size 65536
```

```
root@Hub1:~# ip add show dev gre1
```

```
16: gre1@NONE: <MULTICAST,NOARP,UP,LOWER_UP> mtu 1400 qdisc noqueue state
UNKNOWN group default qlen 10000
    link/gre6 :: brd 2001:db8::172:16:100:2 permaddr 36d2:5e00:2eb3::
    inet 10.10.10.1/24 brd 10.10.10.255 scope global gre1
       valid_lft forever preferred_lft forever
    inet6 fe80::a0a:a01/64 scope link
       valid_lft forever preferred_lft forever
    inet6 fe80::c0a8:6401/64 scope link
       valid_lft forever preferred_lft forever
    inet6 fe80::ac10:6402/64 scope link
       valid_lft forever preferred_lft forever
    inet6 fe80::7f00:1/64 scope host
       valid_lft forever preferred_lft forever
```

Наличие нескольких link-local адресов на GRE интерфейсе является известной особенностью и не является проблемой. Количество link-local зависит от количества IP адресов в системе.

Вывод свидетельствует о том, что GRE интерфейс gre1 присутствует в системе и сконфигурирован.

1.3. Добавьте GRE интерфейс gre1 в файл `/etc/ifaliases.cf` (выделено жирным):

```
root@Hub1:~# vim.tiny /etc/ifaliases.cf
```

```
interface (name="GigabitEthernet0/0" pattern="eth0")
interface (name="GigabitEthernet0/1" pattern="eth1")
interface (name="GigabitEthernet0/2" pattern="eth2")
interface (name="Tunnel1" pattern="gre1")
interface (name="default" pattern="*")
```

- значение параметра `name` должно быть задано в виде TunnelN.
- значение параметра `pattern` должно соответствовать имени GRE интерфейса в нотации Linux.

1.4. Пересчитайте контрольную сумму файла `/etc/ifaliases.cf`:

```
root@Hub1:~# integr_mgr calc -f /etc/ifaliases.cf
```

```
SUCCESS:  Operation was successful.
```

1.5. Перезапустите VPN сервис:

```
root@Hub1:~# systemctl restart vpngate.service
```

Теперь интерфейс `Tunnel1` должен быть доступен к заданию политики СКЗИ и правил МЭ при помощи консоли cisco-like и LSP, а также к конфигурированию средствами ОС.

Для сценария IPsec over GRE (IPsec внутри GRE) политика прикрепляется непосредственно на GRE интерфейс, для сценария GRE over IPsec (GRE внутри IPsec) политика СКЗИ должна быть прикреплена к тому WAN интерфейсу, через который будут маршрутизироваться GRE пакеты с GRE интерфейса.

2. Изменение параметров.

2.1. Выключите интерфейс ifdown greN, отредактируйте файл конфигурации /etc/network/interfaces.d/greN и включите интерфейс ifup greN.

3. Удаление.

3.1. Для удаления интерфейса нужно выключить интерфейс ifdown greN, удалить файл конфигурации /etc/network/interfaces.d/greN, и удалить соответствующую строчку из `/etc/ifaliases.cf` (пересчитав контрольную сумму), после чего перезапустить VPN сервис.

IP v6 over IP v6

Для «С-Терра Шлюз» в виртуализированном исполнении тип сетевого адаптера виртуальной машины должен быть vmxnet3. При использовании другого типа адаптера\драйвера будут портиться IPv6 пакеты.

1. Создание.

Ниже описан пример создания point-to-point GRE IP v6 over IP v6 интерфейса. Point-to-point GRE интерфейсы описываются в файлах /etc/network/interfaces.d/greN.

Cisco-like консоль не поддерживает изменение параметров GRE интерфейса, созданного через консоль Linux.

1.1. Отредактируйте файл `/etc/network/interfaces.d/gre1`, добавив в него описание для point-to-point GRE интерфейса, например:

Использовать в качестве имени GRE интерфейса `gre0` запрещено, так как `gre0` используется для внутренних нужд драйвера GRE. В имени GRE интерфейса нельзя использовать точку.

```
root@Hub1:~# vim.tiny /etc/network/interfaces.d/gre1
```

```
auto gre1
iface gre1 inet6 static
address 2001:db8::10:10:10:1/112
pre-up ip link del gre1 2> /dev/null || true
pre-up ip tunnel add gre1 mode ip6gre key 1 ttl 64 tos inherit remote
2001:db8::172:16:100:2
```

```
pre-up ip link set gre1 mtu 1400
pre-up ip link set gre1 multicast on
post-up ethtool -K gre1 tx-checksumming off > /dev/null
post-down ip link del gre1
```

- Если в строке «pre-up ip tunnel add gre1 mode ip6gre …» не указан параметр local <IP>, то source адрес будет выбран на основании адреса в src маршрута до адреса в remote (можно воспользоваться командой ip route get <адрес в remote> для выяснения source адреса).

1.2. Для того, чтобы GRE интерфейс gre1 появился в системе, выполните команду ifup gre1, после чего, убедитесь что данный интерфейс действительно доступен в системе:

```
root@Hub1:~# ifup gre1
root@Hub1:~# ip -d link show dev gre1
```

```
14: gre1@NONE: <MULTICAST,NOARP,UP,LOWER_UP> mtu 1400 qdisc noqueue state
UNKNOWN mode DEFAULT group default qlen 10000
    link/gre6 :: brd 2001:db8::172:16:100:2 permaddr f289:e792:f673::
promiscuity 0 allmulti 0 minmtu 0 maxmtu 0
    ip6gre remote 2001:db8::172:16:100:2 local any hoplimit 64 encaplimit 4
tclass inherit flowlabel 0x00000 ikey 0.0.0.1 okey 0.0.0.1 addrgenmode random
numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs 65535 tso_max_size
65536 tso_max_segs 65535 gro_max_size 65536
```

```
root@Hub1:~# ip add show dev gre1
```

```
14: gre1@NONE: <MULTICAST,NOARP,UP,LOWER_UP> mtu 1400 qdisc noqueue state
UNKNOWN group default qlen 10000
    link/gre6 :: brd 2001:db8::172:16:100:2 permaddr f289:e792:f673::
    inet6 2001:db8::10:10:10:1/112 scope global
       valid_lft forever preferred_lft forever
    inet6 fe80::c0a8:6401/64 scope link
       valid_lft forever preferred_lft forever
    inet6 fe80::ac10:6402/64 scope link
       valid_lft forever preferred_lft forever
    inet6 fe80::7f00:1/64 scope host
       valid_lft forever preferred_lft forever
```

Наличие нескольких link-local адресов на GRE интерфейсе является известной особенностью и не является проблемой.

Вывод свидетельствует о том, что GRE интерфейс gre1 присутствует в системе и сконфигурирован.

1.3. Добавьте GRE интерфейс gre1 в файл `/etc/ifaliases.cf` (выделено жирным):

```
root@Hub1:~# vim.tiny /etc/ifaliases.cf
```

```
interface (name="GigabitEthernet0/0" pattern="eth0")
interface (name="GigabitEthernet0/1" pattern="eth1")
interface (name="GigabitEthernet0/2" pattern="eth2")
interface (name="Tunnel1" pattern="gre1")
interface (name="default" pattern="*")
```

- значение параметра `name` должно быть задано в виде TunnelN.
- значение параметра `pattern` должно соответствовать имени GRE интерфейса в нотации Linux.

1.4. Пересчитайте контрольную сумму файла /etc/ifaliases.cf:

```
root@Hub1:~# integr_mgr calc -f /etc/ifaliases.cf
```

```
SUCCESS:  Operation was successful.
```

1.5. Перезапустите VPN сервис:

```
root@Hub1:~# systemctl restart vpngate.service
```

Теперь интерфейс `Tunnel1` должен быть доступен к заданию на нем политики СКЗИ и правил МЭ при помощи консоли cisco-like и LSP, а также к конфигурированию средствами ОС.

Для сценария IPsec over GRE (IPsec внутри GRE) политика прикрепляется непосредственно на GRE интерфейс, для сценария GRE over IPsec (GRE внутри IPsec) политика СКЗИ должна быть прикреплена к тому WAN интерфейсу, через который будут маршрутизироваться GRE пакеты с GRE интерфейса.

2. Изменение параметров.

2.1. Выключите интерфейс ifdown greN, отредактируйте файл конфигурации /etc/network/interfaces.d/greN и включите интерфейс ifup greN.

3. Удаление.

3.1. Для удаления интерфейса нужно выключить интерфейс ifdown greN, удалить файл конфигурации /etc/network/interfaces.d/greN, и удалить соответствующую строчку из /etc/ifaliases.cf (пересчитав контрольную сумму), после чего перезапустить VPN сервис.

Point-to-multipoint GRE

Пример создания через консоль cisco-like

IP v4 over IP v4

1. Создание.

Ниже описан пример создания point-to-multipoint GRE IP v4 over IP v4 интерфейса. Данный тип интерфейсов используются в сценарии DMVPN.

1.1. Войдите в консоль cisco-like и перейдите в режим конфигурирования:

```
root@Hub1:~# su cscons
Hub1#configure terminal
```

```
Hub1(config)#
```

1.2. Создайте и настройте point-to-multipoint GRE интерфейс Tunnel 0:

```
Hub1(config)#interface Tunnel 0
Hub1(config-if)#ip address 10.10.20.1 255.255.255.0
Hub1(config-if)#tunnel key 1
Hub1(config-if)#no tunnel path-mtu-discovery
Hub1(config-if)#tunnel ignore-df
Hub1(config-if)#tunnel mode gre multipoint
Hub1(config-if)#no shutdown
Hub1(config-if)#end
```

- В point-to-multipoint GRE параметр tunnel destination не задается, destination адрес будет получен по протоколу NHRP.
- Если параметр tunnel source не указан, то source адрес будет выбран на основании маршрутной информации до destination адреса, который будет получен по протоколу NHRP (зная этот адрес можно воспользоваться командой run ip route get <адрес> для выяснения source адреса).

- Параметры no tunnel path-mtu-discovery и tunnel ignore-df отключают работу PMTUd и копирование DF бита на GRE интерфейсах (это нужно для увеличения вероятности прохождения трафика, в случае его последующего шифрования).

1.3. Просмотрите текущую конфигурацию созданного интерфейса:

```
Hub1#show running-config
```

```
...
interface Tunnel0
 ip address 10.10.20.1 255.255.255.0
 mtu 1400
 tunnel mode gre multipoint
 tunnel key 1
 offloading tx-checksumming
 no tunnel path-mtu-discovery
 tunnel ignore-df
 tunnel ttl inherit
 tunnel tos 0
...
```

1.4. Убедитесь, что настройки интерфейса автоматически прописались в файл /etc/network/interfaces:

```
Hub1#run cat /etc/network/interfaces
```

```
...
auto mgre_0
iface mgre_0 inet manual
st_type GRE
st_state up
st_key 1
st_txchecksumoffload enabled
st_gre_ttl inherit
st_gre_pmtudisc off
st_gre_ignore_df on
st_mtu 1400
address 10.10.20.1
netmask 255.255.255.0
broadcast 10.10.20.255
...
```

Обратите внимание, что point-to-multipoint GRE интерфейсы, созданные консолью cisco-like, имеют следующий формат mgre_N, а созданные консолью Linux – mgreN. Это сделано специально, чтобы визуально можно было отличить каким образом был создан данный интерфейс.

1.5. Убедитесь, что в системе появился соответствующий сетевой интерфейс:

```
Hub1#run ip -d link show dev mgre_0
```

```
...
21: mgre_0@NONE: <NOARP,UP,LOWER_UP> mtu 1400 qdisc noqueue state UNKNOWN mode
DEFAULT group default qlen 10000
    link/gre 0.0.0.0 brd 0.0.0.0 promiscuity 0  allmulti 0 minmtu 0 maxmtu 0
    gre remote any local any ttl inherit nopmtudisc ignore-df ikey 0.0.0.1 okey
0.0.0.1 addrgenmode random numtxqueues 1 numrxqueues 1 gso_max_size 65536
gso_max_segs 65535 tso_max_size 65536 tso_max_segs 65535 gro_max_size 65536
...
```

Теперь интерфейс `Tunnel0` должен быть доступен к заданию политики СКЗИ и правил МЭ при помощи консоли cisco-like и LSP.

Для сценария IPsec over GRE (IPsec внутри GRE) политика прикрепляется непосредственно на GRE интерфейс, для сценария GRE over IPsec (GRE внутри IPsec) политика СКЗИ должна быть прикреплена к тому WAN интерфейсу, через который будут маршрутизироваться GRE пакеты с GRE интерфейса.

2. Удаление.

2.1. Для удаления нужно выполнить следующую команду в консоли cisco-like:

```
Hub1#configure terminal
Hub1(config)#no interface Tunnel0
Hub1(config)#end
```

Пример создания через консоль Linux

IP v4 over IP v4

1. Создание.

Ниже описан пример создания point-to-multipoint GRE IP v4 over IP v4 интерфейса. Point-to-multipoint GRE интерфейсы описываются в файлах /etc/network/interfaces.d/mgreN.

Cisco-like консоль не поддерживает изменение параметров GRE интерфейса, созданного через консоль Linux.

1.1. Отредактируйте файл `/etc/network/interfaces.d/mgre0`, добавив в него описание для point-to-multi GRE интерфейса, например:

```
root@Hub1:~# vim.tiny /etc/network/interfaces.d/mgre0
```

```
auto mgre0
iface mgre0 inet static
address 10.10.10.1
netmask 255.255.255.0
pre-up ip link del mgre0 2> /dev/null || true
pre-up ip tunnel add mgre0 mode gre ttl 64 tos inherit key 1
pre-up ip link set mgre0 mtu 1400
pre-up ip link set mgre0 multicast on
pre-up ip link set mgre0 type gre nopmtudisc
pre-up ip link set mgre0 type gre ignore-df
post-down ip link del mgre0
```

- В point-to-multipoint GRE в строке «pre-up ip tunnel add mgre0 mode gre …» параметр remote не задается, remote адрес будет получен по протоколу NHRP.
- Если в строке «pre-up ip tunnel add mgre0 mode gre …» не указан параметр local <IP>, то source адрес будет выбран на основании маршрутной информации до remote адреса, который будет получен по протоколу NHRP (зная этот адрес можно воспользоваться командой run ip route get <адрес> для выяснения source адреса).
- Перечисленные ниже параметры отключают работу PMTUd и копирование DF бита на GRE интерфейсах (это нужно для увеличения вероятности прохождения трафика, в случае его последующего шифрования):

```
pre-up ip link set gre1 type gre nopmtudisc
pre-up ip link set gre1 type gre ignore-df
```

1.2. Для того, чтобы GRE интерфейс mgre0 появился в системе, выполните команду ifup mgre0, после чего убедитесь, что данный интерфейс действительно доступен в системе:

```
root@Hub1:~# ifup mgre0
```

Если команда ifup завершилась с ошибкой, то ее можно запустить повторно с флагом –v для локализации проблемы, например: ifup –v <iface>.

```
root@Hub1:~# ip -d link show dev mgre0
```

```
22: mgre0@NONE: <MULTICAST,NOARP,UP,LOWER_UP> mtu 1400 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 10000
    link/gre 0.0.0.0 brd 0.0.0.0 promiscuity 0  allmulti 0 minmtu 68 maxmtu 65507
    gre remote any local any ttl 64 tos inherit nopmtudisc ignore-df ikey 0.0.0.1 okey 0.0.0.1 addrgenmode random numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs 65535 tso_max_size 65536 tso_max_segs 65535 gro_max_size 65536
```

Вывод свидетельствует о том, что GRE интерфейс mgre0 присутствует в системе и сконфигурирован.

1.3. Добавьте GRE интерфейс mgre0 в файл `/etc/ifaliases.cf` (выделено жирным):

```
root@Hub1:~# vim.tiny /etc/ifaliases.cf
```

```
interface (name="GigabitEthernet0/0" pattern="eth0")
interface (name="GigabitEthernet0/1" pattern="eth1")
interface (name="GigabitEthernet0/2" pattern="eth2")
interface (name="Tunnel0" pattern="mgre0")
interface (name="default" pattern="*")
```

- значение параметра `name` должно быть задано в виде TunnelN.
- значение параметра `pattern` должно соответствовать имени GRE интерфейса в нотации Linux.

1.4. Пересчитайте контрольную сумму файла `/etc/ifaliases.cf`:

```
root@Hub1:~# integr_mgr calc -f /etc/ifaliases.cf
```

```


SUCCESS:  Operation was successful.
```

1.5. Перезапустите VPN сервис:

```
root@Hub1:~# systemctl restart vpngate.service
```

Теперь интерфейс `Tunnel0` должен быть доступен к заданию политики СКЗИ и правил МЭ при помощи консоли cisco-like и LSP, а также к конфигурированию средствами ОС.

Для сценария IPsec over GRE (IPsec внутри GRE) политика прикрепляется непосредственно на GRE интерфейс, для сценария GRE over IPsec (GRE внутри IPsec) политика СКЗИ должна быть прикреплена к тому WAN интерфейсу, через который будут маршрутизироваться GRE пакеты с GRE интерфейса.

2. Изменение параметров.

2.1. Выключите интерфейс ifdown mgreN, отредактируйте файл конфигурации /etc/network/interfaces.d/mgreN и включите интерфейс ifup mgreN.

3. Удаление.

3.1. Для удаления интерфейса нужно выключить интерфейс ifdown mgreN, удалить файл конфигурации /etc/network/interfaces.d/mgreN, и удалить соответствующую строчку из /etc/ifaliases.cf (пересчитав контрольную сумму), после чего перезапустить VPN сервис.

IPv4 over IPv6

Для «С-Терра Шлюз» в виртуализированном исполнении тип сетевого адаптера виртуальной машины должен быть vmxnet3. При использовании другого типа адаптера\драйвера будут портиться IPv6 пакеты.

1. Создание.

   Ниже описан пример создания point-to-multipoint GRE IP v4 over IPv6 интерфейса. Point-to-multipoint GRE интерфейсы описываются в файлах /etc/network/interfaces.d/mgreN.

Cisco-like консоль не поддерживает изменение параметров GRE интерфейса, созданного через консоль Linux.

1.1. Отредактируйте файл `/etc/network/interfaces.d/mgre0`, добавив в него описание для point-to-multi GRE интерфейса, например:

```
root@Hub1:~# vim.tiny /etc/network/interfaces.d/mgre0
```

```
auto mgre0
iface mgre0 inet static
address 10.10.10.1/24
pre-up ip link del mgre0 2> /dev/null || true
pre-up ip tunnel add mgre0 mode ip6gre key 1 ttl 64 tos inherit
pre-up ip link set mgre0 mtu 1400
pre-up ip link set mgre0 multicast on
post-up ethtool -K mgre0 tx-checksumming off > /dev/null
post-down ip link del mgre0
```

- В point-to-multipoint GRE в строке «pre-up ip tunnel add mgre0 mode ip6gre …» параметр remote не задается, remote адрес будет получен по протоколу NHRP.
- Если в строке «pre-up ip tunnel add mgre0 mode ip6gre …» не указан параметр local <IP>, то source адрес будет выбран на основании маршрутной информации до remote адреса, который будет получен по протоколу NHRP (зная этот адрес можно воспользоваться командой run ip route get <адрес> для выяснения source адреса).

1.2. Для того, чтобы GRE интерфейс mgre0 появился в системе, выполните команду ifup mgre0, после чего убедитесь, что данный интерфейс действительно доступен в системе:

```
root@Hub1:~# ifup mgre0
root@Hub1:~# ip -d link show dev mgre0
```

```
11: mgre0@NONE: <MULTICAST,NOARP,UP,LOWER_UP> mtu 1400 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 10000
    link/gre6 :: brd :: permaddr 2a47:d421:b463:: promiscuity 0 allmulti 0 minmtu 0 maxmtu 0
    ip6gre remote any local any hoplimit 64 encaplimit 4 tclass inherit flowlabel 0x00000 ikey 0.0.0.1 okey 0.0.0.1 addrgenmode random numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs 65535 tso_max_size 65536 tso_max_segs 65535 gro_max_size 65536
```

Вывод свидетельствует о том, что GRE интерфейс mgre0 присутствует в системе и сконфигурирован.

1.3. Добавьте GRE интерфейс mgre0 в файл `/etc/ifaliases.cf` (выделено жирным):

```
root@Hub1:~# vim.tiny /etc/ifaliases.cf
```

```
interface (name="GigabitEthernet0/0" pattern="eth0")
interface (name="GigabitEthernet0/1" pattern="eth1")
interface (name="GigabitEthernet0/2" pattern="eth2")
interface (name="Tunnel0" pattern="mgre0")
```

```
interface (name="default" pattern="*")
```

- значение параметра `name` должно быть задано в виде TunnelN.
- значение параметра `pattern` должно соответствовать имени GRE интерфейса в нотации Linux.

1.4. Пересчитайте контрольную сумму файла `/etc/ifaliases.cf`:

```
root@Hub1:~# integr_mgr calc -f /etc/ifaliases.cf
```

```


SUCCESS:  Operation was successful.
```

1.5. Перезапустите VPN сервис:

```
root@Hub1:~# systemctl restart vpngate.service
```

Теперь интерфейс `Tunnel0` должен быть доступен к заданию политики СКЗИ и правил МЭ при помощи консоли cisco-like и LSP, а также к конфигурированию средствами ОС.

Для сценария IPsec over GRE (IPsec внутри GRE) политика прикрепляется непосредственно на GRE интерфейс, для сценария GRE over IPsec (GRE внутри IPsec) политика СКЗИ должна быть прикреплена к тому WAN интерфейсу, через который будут маршрутизироваться GRE пакеты с GRE интерфейса.

2. Изменение параметров.

2.1. Выключите интерфейс ifdown mgreN, отредактируйте файл конфигурации /etc/network/interfaces.d/mgreN и включите интерфейс ifup mgreN.

3. Удаление.

3.1. Для удаления интерфейса нужно выключить интерфейс ifdown mgreN, удалить файл конфигурации /etc/network/interfaces.d/mgreN, и удалить соответствующую строчку из /etc/ifaliases.cf (пересчитав контрольную сумму), после чего перезапустить VPN сервис.

IP v6 over IP v6

Для «С-Терра Шлюз» в виртуализированном исполнении тип сетевого адаптера виртуальной машины должен быть vmxnet3. При использовании другого типа адаптера\драйвера будут портиться IPv6 пакеты.

1. Создание.

Ниже описан пример создания point-to-multipoint GRE IP v6 over IPv6 интерфейса. Point-to-multipoint GRE интерфейсы описываются в файлах /etc/network/interfaces.d/mgreN.

Cisco-like консоль не поддерживает изменение параметров GRE интерфейса, созданного через консоль Linux.

1.1. Отредактируйте файл /`etc/network/interfaces.d/mgre0`, добавив в него описание для point-to-multi GRE интерфейса, например:

```
root@Hub1:~# vim.tiny /etc/network/interfaces.d/mgre0
```

```
auto mgre0
iface mgre0 inet6  static
address 2001:db8::10:10:10:1/112
pre-up ip link del mgre0 2> /dev/null || true
pre-up ip tunnel add mgre0 mode ip6gre key 1 ttl 64 tos inherit
pre-up ip link set mgre0 mtu 1400
pre-up ip link set mgre0 multicast on
```

```
post-up ethtool -K mgre0 tx-checksumming off > /dev/null
post-down ip link del mgre0
```

- В point-to-multipoint GRE в строке «pre-up ip tunnel add mgre0 mode ip6gre …» параметр remote не задается, remote адрес будет получен по протоколу NHRP.
- Если в строке «pre-up ip tunnel add mgre0 mode ip6gre …» не указан параметр local <IP>, то source адрес будет выбран на основании маршрутной информации до remote адреса, который будет получен по протоколу NHRP (зная этот адрес можно воспользоваться командой run ip route get <адрес> для выяснения source адреса).

1.2. Для того, чтобы GRE интерфейс mgre0 появился в системе, выполните команду ifup mgre0, после чего убедитесь, что данный интерфейс действительно доступен в системе:

```
root@Hub1:~# ifup mgre0
root@Hub1:~# ip -d link show dev mgre0
```

```
15: mgre0@NONE: <MULTICAST,NOARP,UP,LOWER_UP> mtu 1400 qdisc noqueue state UNKNOWN mode DEFAULT group default qlen 10000
    link/gre6 :: brd :: permaddr f6c7:2a7a:7012:: promiscuity 0  allmulti 0 minmtu 0 maxmtu 0
    ip6gre remote any local any hoplimit 64 encaplimit 4 tclass inherit flowlabel 0x00000 ikey 0.0.0.1 okey 0.0.0.1 addrgenmode random numtxqueues 1 numrxqueues 1 gso_max_size 65536 gso_max_segs 65535 tso_max_size 65536 tso_max_segs 65535 gro_max_size 65536
```

Вывод свидетельствует о том, что GRE интерфейс mgre0 присутствует в системе и сконфигурирован.

1.3. Добавьте GRE интерфейс mgre0 в файл `/etc/ifaliases.cf` (выделено жирным):

```
root@Hub1:~# vim.tiny /etc/ifaliases.cf
```

```
interface (name="GigabitEthernet0/0" pattern="eth0")
interface (name="GigabitEthernet0/1" pattern="eth1")
interface (name="GigabitEthernet0/2" pattern="eth2")
interface (name="Tunnel0" pattern="mgre0")
interface (name="default" pattern="*")
```

- значение параметра `name` должно быть задано в виде TunnelN.
- значение параметра `pattern` должно соответствовать имени GRE интерфейса в нотации Linux.

1.4. Пересчитайте контрольную сумму файла `/etc/ifaliases.cf`:

```
root@Hub1:~# integr_mgr calc -f /etc/ifaliases.cf
```

```


SUCCESS:  Operation was successful.
```

1.5. Перезапустите VPN сервис:

```
root@Hub1:~# systemctl restart vpngate.service
```

Теперь интерфейс `Tunnel0` должен быть доступен к заданию политики СКЗИ и правил МЭ при помощи консоли cisco-like и LSP, а также к конфигурированию средствами ОС.

Для сценария IPsec over GRE (IPsec внутри GRE) политика прикрепляется непосредственно на GRE интерфейс, для сценария GRE over IPsec (GRE внутри IPsec) политика СКЗИ должна быть прикреплена к тому WAN интерфейсу, через который будут маршрутизироваться GRE пакеты с GRE интерфейса.

2. Изменение параметров.

2.1. Выключите интерфейс ifdown mgreN, отредактируйте файл конфигурации /etc/network/interfaces.d/mgreN и включите интерфейс ifup mgreN.

3. Удаление.

   3.1. Для удаления интерфейса нужно выключить интерфейс ifdown mgreN, удалить файл конфигурации /etc/network/interfaces.d/mgreN, и удалить соответствующую строчку из /etc/ifaliases.cf (пересчитав контрольную сумму), после чего перезапустить VPN сервис.

Решение проблем при использовании GRE и IPsec

В случаях инкапсуляции поступающих пакетов в GRE, а, затем, в IPsec, пользователь может столкнуться с проблемой передачи пакетов большого размера (длина пакета значительно увеличивается при добавлении GRE и IPsec заголовков). По умолчанию, значение MTU на GRE-интерфейсе – 1400 байт. Значение MTU на физических интерфейсах – 1500 байт.

Может сложиться ситуация, при которой:

- Оконечные устройства генерируют пакеты, в которых выставлен флаг DF (do not fragmentation) в IP заголовке;
- Размер пакета больше, чем MTU GRE-интерфейса;
- ICMP-сообщение протокола PMTUD "Destination host Unreachable. Needed Fragment and DF set." по каким-то причинам до оконечного устройства не доходит.

В таком случае пакеты будут отбрасываться Шлюзом.

Существует несколько вариантов решения данной проблемы:

- снятие DF-флага с IP пакетов на «С-Терра Шлюз»;
- изменение значения MSS в TCP SYN пакетах для контроля максимального размера пакетов в соединении.

Процедура снятия DF-флага с поступающих IP пакетов

Процедуры, описанные ниже, должны быть применены на обоих устройствах «С-Терра Шлюз».

Для того, чтобы снять DF-флаг на поступающих в сетевой интерфейс пакетах при помощи утилиты tc (traffic control), необходимо:

1. Создать файл с названием `zzz_tc_rule_for_clear_df` в директории `/etc/network/if-pre-up.d/` (данная директория выбрана с целью применения настроек утилиты tc при поднятии сетевого интерфейса):

   ```
   root@Hub1-0:~# vim.tiny /etc/network/if-pre-up.d/zzz_tc_rule_for_clear_df
   ```

2. Скопировать содержимое ниже в этот файл:

```
#!/bin/bash

# This script, using the "tc" utility, allows you to clear the DF flag from
incoming IP packets.

INTERFACE="eth1"

if [[ "$IFACE" = $INTERFACE ]]; then
        tc qdisc add dev $INTERFACE ingress >/dev/null 2>&1
        tc filter del dev $INTERFACE parent ffff: handle 800::1 pref 1 protocol
ip u32 >/dev/null 2>&1
        tc filter add dev $INTERFACE parent ffff: pref 1 protocol ip handle
800::1 u32 match ip df action pedit munge ip df clear pipe action csum ip ||
echo "ERROR: Can not add rule to clear DF bit."
fi
```

2.1. Изменить значение переменной `INTERFACE` в зависимости от того, куда будут приходить пакеты с DF-флагом (обычно, это LAN-интерфейс, в данном случае – это «eth1»);

2.2. Дать файлу право на исполнение:

```
root@Hub1:~# chmod u+x /etc/network/if-pre-up.d/zzz_tc_rule_for_clear_df
```

2.3. Перезапустить интерфейс, на котором необходимо применить правила снятия DF-флага:

```
root@Hub1:~# ifdown eth1; ifup eth1
```

2.4. Убедиться в том, что правила применились:

```
root@Hub1:~# tc -s filter show parent ffff: dev eth1
```

```
filter protocol ip pref 1 u32 chain 0
filter protocol ip pref 1 u32 chain 0 fh 800: ht divisor 1
filter protocol ip pref 1 u32 chain 0 fh 800::1 order 1 key ht 800 bkt 0
terminal flowid not_in_hw (rule hit 71661 success 71661)
  match 00004000/00004000 at 4 (success 71661 )
        action order 1:  pedit action pipe keys 1
         index 1 ref 1 bind 1 installed 57 sec firstused 53 sec
         key #0  at 4: val 00000000 mask ffffbfff
        Action statistics:
        Sent 2165058738 bytes 1492685 pkt (dropped 0, overlimits 0 requeues 0)
        backlog 0b 0p requeues 0

        action order 2: csum (iph) action pass
        index 1 ref 1 bind 1 installed 57 sec firstused 53 sec
        Action statistics:
        Sent 2165058738 bytes 1492685 pkt (dropped 0, overlimits 0 requeues 0)
        backlog 0b 0p requeues 0
```

Данный скрипт будет выполняться при запуске/перезапуске Шлюза и при переходе назначенного в скрипте сетевого интерфейса в состояние «Up».

Изменение MSS при передаче TCP пакетов

Изменение MSS в TCP пакетах нужно настраивать на обоих устройствах «С-Терра Шлюз», как можно ближе к источнику трафика, как правило на LAN интерфейсах.

В примерах ниже, значение MSS будет выставлено в 1360 байт, что позволит оконечными устройствами генерировать TCP пакеты, не превышающие размер в 1400 байт на уровне L3.

Через консоль cisco-like

Для того, чтобы выставить нужное значение MSS, необходимо выполнить следующие настройки:

```
...
ip access-list extended FIX_MSS_ON_LAN_IN
 permit tcp any any
!
class-map CM_MSS_FIX
 match access-group name FIX_MSS_ON_LAN_IN
!
policy-map PM_MSS_FIX
 class CM_MSS_FIX
  set adjust-mss 1360
 !
interface GigabitEthernet0/1
 ...
 service-policy input PM_MSS_FIX
```

```
 ...
!
```

Через консоль Linux

Для того, чтобы выставить нужное значение MSS, необходимо воспользоваться командой:

```
root@Hub1:~# iptables -t mangle -A FORWARD -p tcp --tcp-flags SYN,RST SYN -j TCPMSS --set-mss 1360
```

Получить расширенную справку можно при помощи команды `man iptables-extensions`.

Для того, чтобы сохранить добавленное правило, необходимо обратиться к документу «Инструкция по настройке NAT и межсетевого экрана на «С-Терра Шлюз» при помощи утилиты «iptables».

Bonding (port channel) интерфейсы

Интерфейсы Bonding являются виртуальными сетевыми интерфейсами, которые объединяют несколько физических сетевых интерфейсов для увеличения отказоустойчивости и пропускной способности канала. Данные интерфейсы работают в соответствии со стандартом IEEE 802.3ad (если настроен протокол LACP).

Подробнее про Bonding интерфейсы описано в документе «Агрегирование сетевых интерфейсов на «С-Терра Шлюз».

Запрещено объединять в один Bonding интерфейс физические интерфейсы, входящие в разные группы – доверенные и недоверенные (например, запрещено объединять интерфейсы из LAN и WAN сегментов), так как физические интерфейсы, объединенные в Bonding становятся неконтролируемыми с точки зрения СКЗИ и МЭ, соответственно, политику СКЗИ и правили МЭ нужно применять на виртуальный Bonding интерфейс, который является логическим объединением физических интерфейсов.

Режим LACP

Пример создания через консоль Linux

1. Создание.

   Bonding интерфейсы описываются в файлах /etc/network/interfaces.d/bondN.

   Ниже описан пример объединения физических интерфейсов eth1 и eth2 в bonding интерфейс `bond0` с протоколом LACP.

   1.1. Выключите сетевые интерфейсы, которые будут входить в bonding интерфейс `bond0`, если они ранее были включены или настроены:

   ```
   root@Hub1:~# ifdown eth1
   root@Hub1:~# ifdown eth2
   ```

   1.2. Удалите в файле `/etc/network/interfaces` или в файлах, располагающихся в директории `/etc/network/interfaces.d/` строки с настройками для интерфейсов, которые будут входить в bonding интерфейс `bond0`, в данном примере это eth1 и eth2 (удалять строки с данными интерфейсами из файла `/etc/ifaliases` не нужно).

   1.3. Отредактируйте файл `/etc/network/interfaces.d/bond0`, добавив в него описание для Bonding (port channel) интерфейса, например:

   ```
   root@Hub1:~# vim.tiny /etc/network/interfaces.d/bond0
   ```

   ```
   auto bond0
   ```

```
iface bond0 inet static
address 192.168.2.1
netmask 255.255.255.0
mtu 1500
slaves eth1 eth2
bond_mode 802.3ad
bond_miimon 100
bond_xmit_hash_policy layer2+3
```

- **address <IP адрес>** – IP адрес логического интерфейса bond0;
- **netmask <маска>**– сетевая маска для IP адрес логического интерфейса;
- **mtu <значение>** – параметр для задания MTU интерфейса; значение по умолчанию – 1500; при изменении MTU интерфейса bond0, меняется MTU и объединяемых интерфейсов eth1 и eth2;
- **slaves <интерфейсы>** – имена физических интерфейсов, объединяемых в агрегированный канал;
- **bond_mode** – настройка, отвечающая за протокол, который обеспечивает работу агрегации каналов (в данном примере – это LACP, который основан на стандарте 802.3ad);
- **bond_miimon <интервал>** – параметр, определяющий интервал времени в миллисекундах. Через заданное время будет происходить проверка физических интерфейсов на их доступность;
- **bond_xmit_hash_policy** – параметр, определяющий уровень (в соответствии с эталонной моделью OSI), на котором будет происходить вычисление хеш-функции для последующей балансировки трафика.

При layer2+3 на втором уровне первые 5 байт MAC адреса НЕ учитываются при балансировке.

1.4. Для того, чтобы Bonding интерфейс `bond0` появился в системе, выполните команду `ifup bond0`, после чего убедитесь, что данный интерфейс действительно доступен в системе, а интерфейсы `eth1/eth2` добавлены в `bond0`:

```
root@Hub1:~# ifup bond0
root@Hub1:~# ip address show bond0
```

```
10: bond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc noqueue state
UP group default qlen 10000
    link/ether 00:50:56:9e:6e:cd brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.1/24 brd 192.168.2.255 scope global bond0
       valid_lft forever preferred_lft forever
    inet6 fe80::4389:22fe:93dd:bd03/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
```

```
root@Hub1:~# ip address show eth1
```

```
3: eth1: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc fq_codel master
bond0 state UP group default qlen 1000
    link/ether 00:50:56:9e:6e:cd brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    altname ens192
    altname GigabitEthernet0_1
```

```
root@Hub1:~# ip address show eth2
```

```
4: eth2: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc fq_codel master
bond0 state UP group default qlen 1000
    link/ether     00:50:56:9e:6e:cd     brd     ff:ff:ff:ff:ff:ff     permaddr
00:50:56:9e:92:46
    altname enp19s0
```

```
    altname ens224
    altname GigabitEthernet0_2
```

Видно, что Bonding интерфейс `bond0` присутствует в системе и сконфигурирован, а также интерфейсы `eth1/eth2` добавлены в `bond0`. Расширенную информацию о `bond0` можно посмотреть при помощи команд:

```
root@Hub1:~# cat /proc/net/bonding/bond0
root@Hub1:~# ip -d address show bond0
root@Hub1:~# ip -s address show bond0
```

1.5. Добавьте Bonding интерфейс `bond0` в файл `/etc/ifaliases.cf` (выделено жирным):

```
root@Hub1:~# vim.tiny /etc/ifaliases.cf
```

```
interface (name="GigabitEthernet0/0" pattern="eth0")
interface (name="GigabitEthernet0/1" pattern="eth1")
interface (name="GigabitEthernet0/2" pattern="eth2")
interface (name="Port-channel1" pattern="bond0")
interface (name="default" pattern="*")
```

- значение параметра `name` должно быть задано в виде Port-channelN;
- значение параметра `pattern` должно соответствовать имени Bonding интерфейса в нотации Linux.

1.6. Пересчитайте контрольную сумму файла /etc/ifaliases.cf:

```
root@Hub1:~# integr_mgr calc -f /etc/ifaliases.cf
```

```


SUCCESS:  Operation was successful.
```

1.7. Перезапустите VPN сервис:

```
root@Hub1:~# systemctl restart vpngate.service
```

Теперь интерфейс `Port-channel1` должен быть доступен к заданию на нем политики СКЗИ и правил МЭ при помощи консоли cisco-like и LSP. **Конфигурирование при помощи консоли cisco-like будет невозможно**.

2. Изменение параметров.

2.1. Выключите интерфейс ifdown bondN, отредактируйте файл конфигурации /etc/network/interfaces.d/bondN и включите интерфейс ifup bondN.

3. Удаление.

3.1. Для удаления интерфейса нужно выключить интерфейс ifdown bondN, удалить файл конфигурации /etc/network/interfaces.d/bondN и удалить соответствующую строчку из /etc/ifaliases.cf (пересчитав контрольную сумму), перезапустить VPN сервис.

Режим Active-Passive

Пример создания через консоль Linux

1. Создание.

Bonding интерфейсы описываются в файлах /etc/network/interfaces.d/bondN.

Ниже описан пример объединения физических интерфейсов eth1 и eth2 в bonding интерфейс `bond0` в режиме основной-резервный (Active-Passive).

1.1. Выключите сетевые интерфейсы, которые будут входить в bonding интерфейс `bond0`, если они ранее были включены или настроены:

```
root@Hub1:~# ifdown eth1
root@Hub1:~# ifdown eth2
```

1.2. Удалите в файле `/etc/network/interfaces` или в файлах, располагающихся в директории `/etc/network/interfaces.d/` строки с настройками для интерфейсов, которые будут входить в bonding интерфейс `bond0`, в данном примере это eth1 и eth2 (удалять строки с данными интерфейсами из файла `/etc/ifaliases` не нужно).

1.3. Отредактируйте файл `/etc/network/interfaces.d/bond0`, добавив в него описание для Bonding (port channel) интерфейса, например:

```
root@Hub1:~# vim.tiny /etc/network/interfaces.d/bond0
```

```
auto bond0
iface bond0 inet static
address 192.168.2.1
netmask 255.255.255.0
mtu 1500
slaves eth1 eth2
bond_mode active-backup
bond_miimon 100
```

- **address <IP адрес>** – IP адрес логического интерфейса bond0;
- **netmask <маска>**– сетевая маска для IP адрес логического интерфейса;
- **mtu <значение>** – параметр для задания MTU интерфейса; значение по умолчанию – 1500; при изменении MTU интерфейса bond0 меняется MTU и объединяемых интерфейсов eth1 и eth2;
- **slaves <интерфейсы>** – имена физических интерфейсов, объединяемых в агрегированный канал;
- **bond_mode** – настройка, отвечающая за протокол, который обеспечивает работу агрегации каналов (в данном примере – это active-backup);
- **bond_miimon <интервал>** – параметр, определяющий интервал времени в миллисекундах. Через заданное время будет происходить проверка физических интерфейсов на их доступность;

1.4. Для того, чтобы Bonding интерфейс `bond0` появился в системе, выполните команду `ifup bond0`, после чего убедитесь, что данный интерфейс действительно доступен в системе, а интерфейсы `eth1/eth2` добавлены в `bond0`:

```
root@Hub1:~# ifup bond0
root@Hub1:~# ip address show bond0
```

```
10: bond0: <BROADCAST,MULTICAST,MASTER,UP,LOWER_UP> mtu 1500 qdisc noqueue state
UP group default qlen 10000
    link/ether 00:50:56:9e:6e:cd brd ff:ff:ff:ff:ff:ff
    inet 192.168.2.1/24 brd 192.168.2.255 scope global bond0
       valid_lft forever preferred_lft forever
    inet6 fe80::4389:22fe:93dd:bd03/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
```

```
root@Hub1:~# ip address show eth1
```

```
3: eth1: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc fq_codel master
bond0 state UP group default qlen 1000
    link/ether 00:50:56:9e:6e:cd brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    altname ens192
    altname GigabitEthernet0_1
```

```
root@Hub1:~# ip address show eth2
```

```
4: eth2: <BROADCAST,MULTICAST,SLAVE,UP,LOWER_UP> mtu 1500 qdisc fq_codel master
bond0 state UP group default qlen 1000
```

```
    link/ether     00:50:56:9e:6e:cd     brd     ff:ff:ff:ff:ff:ff     permaddr
00:50:56:9e:92:46
    altname enp19s0
    altname ens224
    altname GigabitEthernet0_2
```

Видно, что Bonding интерфейс `bond0` присутствует в системе и сконфигурирован, а также интерфейсы `eth1/eth2` добавлены в `bond0`. Расширенную информацию о `bond0` можно посмотреть при помощи команд:

```
root@Hub1:~# cat /proc/net/bonding/bond0
root@Hub1:~# ip -d address show bond0
root@Hub1:~# ip -s address show bond0
```

1.5. Добавьте Bonding интерфейс `bond0` в файл `/etc/ifaliases.cf` (выделено жирным):

```
root@Hub1:~# vim.tiny /etc/ifaliases.cf
```

```
interface (name="GigabitEthernet0/0" pattern="eth0")
interface (name="GigabitEthernet0/1" pattern="eth1")
interface (name="GigabitEthernet0/2" pattern="eth2")
interface (name="Port-channel1" pattern="bond0")
interface (name="default" pattern="*")
```

- значение параметра `name` должно быть задано в виде Port-channelN;
- значение параметра `pattern` должно соответствовать имени Bonding интерфейса в нотации Linux.

1.6. Пересчитайте контрольную сумму файла `/etc/ifaliases.cf`:

```
root@Hub1:~# integr_mgr calc -f /etc/ifaliases.cf
```

```


SUCCESS:  Operation was successful.
```

1.7. Перезапустите VPN сервис:

```
root@Hub1:~# systemctl restart vpngate.service
```

Теперь интерфейс `Port-channel1` должен быть доступен к заданию политики СКЗИ и правил МЭ при помощи консоли cisco-like и LSP. **Конфигурирование при помощи консоли cisco-like будет невозможно**.

2. Изменение параметров.

2.1. Выключите интерфейс ifdown bondN, отредактируйте файл конфигурации /etc/network/interfaces.d/bondN и включите интерфейс ifup bondN.

3. Удаление.

3.1. Для удаления интерфейса нужно выключить интерфейс ifdown bondN, удалить файл конфигурации /etc/network/interfaces.d/bondN и удалить соответствующую строчку из /etc/ifaliases.cf (пересчитав контрольную сумму), перезапустить VPN сервис.

stVTI интерфейсы

Интерфейсы stVTI являются виртуальными сетевыми интерфейсами и используются для организации Route Based VPN по аналогии с GRE (GRE over IPsec). Не поддерживают передачу Multicast трафика. Могут быть помещены в VRF.

Пример создания через консоль Linux

1. Создание.

stVTI интерфейсы описываются в файлах `/etc/network/interfaces.d/st_vtiN`.

Имя создаваемого stVTI интерфейса должно быть вида `st_vtiN`, в противном случае входящие IPsec пакеты на таком интерфейсе будут отброшены драйвером Продукта.

1.1. Отредактируйте файл `/etc/network/interfaces.d/st_vti0`, добавив в него настройки для интерфейса `st_vti0`, например:

```
auto st_vti0
iface st_vti0 inet static
address 10.10.10.1/24
pre-up ip link del st_vti0 type dummy 2>/dev/null || true
pre-up ip link add st_vti0 type dummy
post-down ip link set dev st_vti0 down
```

1.2. Для того, чтобы stVTI интерфейс `st_vti0` появился в системе, выполните команду `ifup st_vti0`, после чего убедитесь, что данный интерфейс действительно доступен в системе:

```
root@Hub1:~# ifup st_vti0
root@Hub1:~# ip address show st_vti0
```

```
9: st_vti0: <BROADCAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
group default qlen 10000
    link/ether e2:32:af:83:8b:e3 brd ff:ff:ff:ff:ff:ff
    inet 10.10.10.1/24 brd 10.10.10.255 scope global st_vti0
       valid_lft forever preferred_lft forever
    inet6 fe80::bcbb:931:c0f:d43a/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
```

Вывод свидетельствует о том, что stVTI интерфейс `st_vti0` присутствует в системе и сконфигурирован.

1.3. Добавьте stVTI интерфейс `st_vti0` в файл `/etc/ifaliases.cf` (выделено жирным):

```
root@Hub1:~# vim.tiny /etc/ifaliases.cf
```

```
interface (name="GigabitEthernet0/0" pattern="eth0")
interface (name="GigabitEthernet0/1" pattern="eth1")
interface (name="GigabitEthernet0/2" pattern="eth2")
interface (name="stVTI0" pattern="st_vti0")
interface (name="default" pattern="*")
```

- значение параметра `name` должно быть задано в виде stVTIN;
- значение параметра `pattern` должно соответствовать имени stVTI интерфейса в нотации Linux.

1.4. Пересчитайте контрольную сумму файла `/etc/ifaliases.cf`:

```
root@Hub1:~# integr_mgr calc -f /etc/ifaliases.cf
```

```


SUCCESS:  Operation was successful.
```

1.5. Перезапустите VPN сервис:

```
root@Hub1:~# systemctl restart vpngate.service
```

Теперь интерфейс `stVTI0` должен быть доступен к заданию на нем политики СКЗИ и правил МЭ при помощи консоли cisco-like и LSP. **Конфигурирование при помощи консоли cisco-like будет невозможно**.

2. Изменение параметров.

Выключите интерфейс командой `ifdown st_vti0`, отредактируйте файл конфигурации `/etc/network/interfaces.d/st_vti0` и включите интерфейс командой `ifup st_vti0`.

3. Удаление.

   3.1. Для удаления интерфейса нужно выключить интерфейс командой `ifdown st_vti0`, удалить файл конфигурации `/etc/network/interfaces.d/st_vti0`, удалить соответствующую строчку из `/etc/ifaliases.cf` (пересчитав контрольную сумму), удалить интерфейс при помощи команды `ip link del dev st_vti0` и перезапустить VPN сервис.

VRF интерфейсы

VRF (Virtual Routing and Forwarding) является технологией для создания изолированных инстансов маршрутизации трафика в рамках одного устройства. К интерфейсу, **входящему в VRF**, можно прикреплять криптокарту только в том случае, если этот интерфейс является stVTI интерфейсом (подробнее в сценарии «Построение VPN соединения при наличии подсетей с пересекающимся адресным пространством и отказоустойчивостью»).

Пример создания через консоль Linux

1. Создание.

   VRF интерфейсы описываются в файлах в директории `/etc/network/interfaces.d/`.

   Ниже приведен пример создания VRF интерфейса (инстанса) с именем `vrf_blue` и своей таблицей маршрутизации и добавления входящих в него интерфейсов.

   1.1. Отредактируйте файл `/etc/network/interfaces.d/vrf_blue`, добавив в него настройки для интерфейса `vrf_blue`, например:

```
auto vrf_blue
iface vrf_blue inet manual
pre-up ip link del vrf_blue type vrf 2>/dev/null || true
pre-up ip route flush table 10 2>/dev/null || true
pre-up ip link add vrf_blue type vrf table 10
pre-up ip link set dev vrf_blue up
pre-up ip route add table 10 unreachable default metric 4278198272
pre-up ip link set dev eth0 master vrf_blue
pre-up ip link set dev eth1 master vrf_blue
post-down ip link set dev vrf_blue down
```

   - `pre-up ip route flush table 10` – предварительная очистка таблицы маршрутизации 10, которая будет использоваться данным VRF инстансом;
   - `pre-up ip route add table 10 unreachable default metric 4278198272` – добавление unreachable маршрута по умолчанию, чтобы трафик, для которого отсутствует маршрут не покидал пределы данного VRF инстанса;
   - `pre-up ip link set dev eth0 master vrf_blue` – добавление интерфейса eth0 в данный VRF инстанс.

   1.2. Для того, чтобы VRF интерфейс `vrf_blue` появился в системе, выполните команду `ifup vrf_blue`, после чего убедитесь, что данный интерфейс действительно доступен в системе:

```
root@Hub1:~# ifup vrf_blue
root@Hub1:~# ip address show vrf_blue
```

```
9: vrf_blue: <NOARP,MASTER,UP,LOWER_UP> mtu 65575 qdisc noqueue state UP group
default qlen 10000
    link/ether 0a:fe:37:bb:cc:ef brd ff:ff:ff:ff:ff:ff
```

1.3. Убедитесь, что интерфейсы, добавленные в данный VRF, действительно входят в него:

```
root@Hub1:~# ip address show eth0
```

```
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master vrf_blue state UP group default qlen 10000
    link/ether 00:50:56:9e:d3:8f brd ff:ff:ff:ff:ff:ff
    altname enp3s0
    altname ens160
    altname GigabitEthernet0_0
    inet 172.16.100.2/24 brd 172.16.100.255 scope global eth0
       valid_lft forever preferred_lft forever
    inet6 fe80::605c:3ff2:57df:437b/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
```

```
root@Hub1:~# ip address show eth1
```

```
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master vrf_blue state UP group default qlen 10000
    link/ether 00:50:56:9e:6b:21 brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    altname ens192
    altname GigabitEthernet0_1
    inet 192.168.100.1/24 brd 192.168.100.255 scope global eth1
       valid_lft forever preferred_lft forever
    inet6 fe80::ee31:332f:369f:59d8/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
```

2. Изменение параметров.

2.1. Выключите интерфейс командой `ifdown vrf_blue`, отредактируйте файл конфигурации `/etc/network/interfaces.d/vrf_blue` и включите интерфейс командой `ifup vrf_blue`.

3. Удаление.

3.1. Для удаления интерфейса нужно выключить интерфейс командой `ifdown vrf_blue`, удалить интерфейс при помощи команды `ip link del dev vrf_blue` и удалить файл конфигурации `/etc/network/interfaces.d/vrf_blue`.

Loopback интерфейсы

Пример создания через консоль Linux

Интерфейсы Loopback являются виртуальными сетевыми интерфейсами. Данные интерфейсы применяются для выставления заданного на Loopback интерфейсе IP адреса в качестве IP адреса источника для пакетов протоколов IKE/ESP/OSPF/BGP и прочих.

Интерфейсы Loopback не предназначены для маршрутизации через них IP трафика.

1. Создание.

Loopback интерфейсы описываются в файлах `/etc/network/interfaces.d/loopbackN`.

1.1. Отредактируйте файл `/etc/network/interfaces.d/loopback0`, добавив в него описание для Loopback интерфейса, например:

```
root@Hub1:~# vim.tiny /etc/network/interfaces.d/loopback0
```

```
auto loopback0
iface loopback0 inet static
address 172.30.1.1
netmask 255.255.255.255
```

```
pre-up ip link del loopback0 type dummy 2>/dev/null || true
pre-up ip link add loopback0 type dummy
post-down ip link set dev loopback0 down
```

1.2. Для того, чтобы Loopback интерфейс `loopback0` появился в системе, выполните команду `ifup loopback0`, после чего убедитесь, что данный интерфейс действительно доступен в системе:

```
root@Hub1:~# ifup loopback0
root@Hub1:~# ip address show loopback0
```

```
9: loopback0: <BROADCAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc noqueue state UNKNOWN
group default qlen 10000
    link/ether ae:a8:37:20:e0:bb brd ff:ff:ff:ff:ff:ff
    inet 172.30.1.1/32 brd 172.30.1.1 scope global loopback0
       valid_lft forever preferred_lft forever
    inet6 fe80::412e:39e3:e064:6683/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
```

Вывод свидетельствует о том, что Loopback интерфейс `loopback0` присутствует в системе и сконфигурирован.

1.3. Добавьте Loopback интерфейс `loopback0` в файл `/etc/ifaliases.cf` (выделено жирным):

```
root@Hub1:~# vim.tiny /etc/ifaliases.cf
```

```
interface (name="GigabitEthernet0/0" pattern="eth0")
interface (name="GigabitEthernet0/1" pattern="eth1")
interface (name="GigabitEthernet0/2" pattern="eth2")
interface (name="Loopback0" pattern="loopback0")
interface (name="default" pattern="*")
```

- значение параметра `name` должно быть задано в виде LoopbackN;
- значение параметра `pattern` должно соответствовать имени Loopback интерфейса в нотации Linux.

1.4. Пересчитайте контрольную сумму файла `/etc/ifaliases.cf`:

```
root@Hub1:~# integr_mgr calc -f /etc/ifaliases.cf
```

```


SUCCESS:  Operation was successful.
```

1.5. Перезапустите VPN сервис:

```
root@Hub1:~# systemctl restart vpngate.service
```

Теперь интерфейс `Loopback0` должен быть доступен в консоли cisco-like, но задавать на нем политику СКЗИ и правил МЭ при помощи как консоли cisco-like, так и LSP не имеет никакого смысла, так как IP пакеты через интерфейсы этого типа не могут маршрутизироваться. **Конфигурирование при помощи консоли cisco-like будет невозможно**.

2. Изменение параметров.

2.1. Выключите интерфейс ifdown loopbackN, отредактируйте файл конфигурации /etc/network/interfaces.d/loopbackN и включите интерфейс ifup loopbackN.

3. Удаление.

3.1. Для удаления интерфейса нужно выключить интерфейс ifdown loopbackN, удалить файл конфигурации /etc/network/interfaces.d/loopbackN, удалить соответствующую строчку из /etc/ifaliases.cf (пересчитав контрольную сумму), удалить интерфейс при помощи команды ip link del dev loopbackN и перезапустить VPN сервис.

Bridge интерфейсы

Интерфейсы Bridge являются виртуальными сетевыми интерфейсами, которые объединяют несколько физических сетевых интерфейсов для выполнения функций коммутатора (физические интерфейсы, добавленные в коммутатор, становятся портами коммутатора).

Без предварительного настройки функций контроля интерфейсов входящих в Bridge, (настройка описана в документе «Построение L2 VPN туннеля при помощи программного модуля «С-Терра L2») запрещено добавлять в один Bridge интерфейс физические интерфейсы, входящие в разные группы – доверенные и недоверенные (например, запрещено одновременно добавлять интерфейсы из LAN и WAN сегментов), так как физические интерфейсы, добавленные в Bridge, становятся неконтролируемыми с точки зрения СКЗИ и МЭ, соответственно, политику СКЗИ и правила МЭ нужно применять на виртуальный Bridge интерфейс, который является логическим объединением физических.

Пример создания через консоль Linux

1. Создание.

   Bridge интерфейсы описываются в файлах `/etc/network/interfaces.d/bridgeN`.

   Ниже будет описан пример настройки виртуального коммутатора (сетевой мост, бридж) `bridge0` с физическими интерфейсами `eth1` и `eth2` (порты коммутатора).

   1.1. Выключите сетевые интерфейсы, которые будут входить в виртуальный коммутатор `bridge0`, если они ранее были включены или настроены:

   ```
   root@Hub1:~# ifdown eth1
   root@Hub1:~# ifdown eth2
   ```

   1.2. Удалите в файле `/etc/network/interfaces` или в файлах, располагающихся в директории `/etc/network/interfaces.d/` строки с настройками для интерфейсов, которые будут входить в виртуальный коммутатор `bridge0`, в данном примере это eth1 и eth2 (удалять строки с данными интерфейсами из файла `/etc/ifaliases` не нужно).

   1.3. Отредактируйте файл `/etc/network/interfaces.d/bridge0`, добавив в него настройки для интерфейса `bridge0`, например:

   ```
   root@Hub1:~# vim.tiny /etc/network/interfaces.d/bridge0
   ```

   ```
   auto bridge0
   iface bridge0 inet static
   address 192.168.1.2
   netmask 255.255.255.0
   mtu 1500
   bridge_ports eth1 eth2
   ```

   - **address <IP адрес>** – IP адрес интерфейса bridge0;
   - **netmask <маска>**– сетевая маска для IP адрес интерфейса bridge0;
   - **mtu <значение>** – параметр для задания MTU интерфейса bridge0; значение по умолчанию – 1500; при изменении MTU интерфейса bridge0 значение MTU физических интерфейсов eth1 и eth2 меняется автоматический при подъеме интерфейса bridge0;
   - **bridge_ports <интерфейсы>** – имена физических интерфейсов, входящих в коммутатор.

   Примечание – дополнительную информацию по настройке можно найти в man при помощи команды: `man bridge-utils-interfaces`.

1.4. Для того, чтобы Bridge интерфейс `bridge0` появился в системе, выполните команду `ifup bridge0`, после чего убедитесь, что данный интерфейс действительно доступен в системе и он выступает коммутатором, а интерфейсы `eth1/eth2` добавлены в `bridge0`:

```
root@Hub1:~# ifup bridge0
root@Hub1:~# ip address show bridge0
```

```
16: bridge0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP
group default qlen 10000
    link/ether 66:48:d7:0f:98:ef brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.2/24 brd 192.168.1.255 scope global bridge0
       valid_lft forever preferred_lft forever
    inet6 fe80::ab3c:de43:a86e:2744/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
```

```
root@Hub1:~# brctl show bridge0
```

```
bridge name     bridge id               STP enabled     interfaces
bridge0         8000.6648d70f98ef       no              eth1
                                                        eth2
```

```
root@Hub1:~# ip address show eth1
```

```
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master
bridge0 state UP group default qlen 10000
    link/ether 00:50:56:9e:6e:cd brd ff:ff:ff:ff:ff:ff
    altname enp11s0
    altname ens192
    altname GigabitEthernet0_1
```

```
root@Hub1:~# ip address show eth2
```

```
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master
bridge0 state UP group default qlen 1000
    link/ether 00:50:56:9e:92:46 brd ff:ff:ff:ff:ff:ff
    altname enp19s0
    altname ens224
    altname GigabitEthernet0_2
```

Вывод свидетельствует о том, что Bridge интерфейс `bridge0` присутствует в системе и сконфигурирован, а также интерфейсы `eth1/eth2` добавлены в `bridge0`. Расширенную информацию о bridge0 можно посмотреть при помощи утилиты `brctl`.

1.5. Добавьте Bridge интерфейс `bridge0` в файл `/etc/ifaliases.cf` (выделено жирным):

```
root@Hub1:~# vim.tiny /etc/ifaliases.cf
```

```
interface (name="GigabitEthernet0/0" pattern="eth0")
interface (name="GigabitEthernet0/1" pattern="eth1")
interface (name="GigabitEthernet0/2" pattern="eth2")
interface (name="Bridge0" pattern="bridge0")
interface (name="default" pattern="*")
```

- значение параметра `name` должно быть задано в виде `BridgeN`.
- значение параметра `pattern` должно соответствовать имени Bridge интерфейса в нотации Linux.

1.6. Пересчитайте контрольную сумму файла `/etc/ifaliases.cf`:

```
root@Hub1:~# integr_mgr calc -f /etc/ifaliases.cf
```

```


SUCCESS:  Operation was successful.
```

1.7. Перезапустите VPN сервис:

```
root@Hub1:~# systemctl restart vpngate.service
```

Теперь интерфейс Bridge0 должен быть доступен к заданию на нем политики СКЗИ и правил МЭ при помощи консоли cisco-like и LSP, также к конфигурированию средствами ОС. **Конфигурирование при помощи консоли cisco-like будет невозможно.**

2. Изменение параметров.

   2.1. Выключите интерфейс ifdown bridgeN, отредактируйте файл конфигурации /etc/network/interfaces.d/bridgeN и включите интерфейс ifup bridgeN.

3. Удаление.

   3.1. Для удаления интерфейса нужно выключить интерфейс ifdown bridgeN, удалить файл конфигурации /etc/network/interfaces.d/bridgeN и удалить соответствующую строчку из /etc/ifaliases.cf (пересчитав контрольную сумму), перезапустить VPN сервис.

Получение сетевых параметров по DHCP

Консоль cisco-like не поддерживает настройку получения сетевых параметров по DHCP.

Настройку нужно производить через консоль Linux либо в файле `/etc/network/interfaces` **за пределами секции**, начало и конец которой помечены маркерами `###netifcfg-begin###` и `###netifcfg-end###`, либо в `/etc/network/interfaces.d/<iface>`

Пример настройки получения сетевых параметров по DHCP на интерфейсе eth2

Например, чтобы настроить получение сетевых параметров по DHCP на интерфейсе eth2 нужно выполнить следующие шаги.

1. Выключите сетевой интерфейс:

```
root@Hub1:~# ifdown eth2
```

2. Уберите из секции начало и конец которой помечены маркерами `###netifcfg-begin###` и `###netifcfg-end###`, которая находится в файле `/etc/network/interfaces`, все строки, соответствующие интерфейсу eth2.

3. Добавьте настройки в файл `/etc/network/interfaces`:

```
root@Hub1:~# vim.tiny /etc/network/interfaces
```

```
...
###netifcfg-begin###
...
###netifcfg-end###

auto eth2
iface eth2 inet dhcp
mtu 1500
```

4. Включите сетевой интерфейс:

```
root@Hub1:~# ifup eth2
```

```
Internet Systems Consortium DHCP Client 4.4.3-P1
Copyright 2004-2022 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/

Interface lo               is skipped: loopback interface
```

```
Interface lo               is skipped: loopback interface
Listening on Socket/eth2
Sending on   Socket/eth2
Sending on   Socket/fallback
DHCPDISCOVER on eth2 to 255.255.255.255 port 67 interval 5
DHCPOFFER of 10.0.106.31 from 10.0.0.17
DHCPREQUEST for 10.0.106.31 on eth2 to 255.255.255.255 port 67
DHCPACK of 10.0.106.31 from 10.0.0.17
Invalid domain list.
suspect value in domain_search option - discarded
Invalid domain list.
bound to 10.0.106.31 -- renewal in 9450 seconds.
```

5. Посмотрите выданный адрес:

```
root@Hub1:~# ip address show dev eth2
```

```
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 10000
    link/ether 00:50:56:9e:92:46 brd ff:ff:ff:ff:ff:ff
    altname enp19s0
    altname ens224
    altname GigabitEthernet0_2
    inet 10.0.106.31/16 brd 10.0.255.255 scope global dynamic eth2
       valid_lft 21552sec preferred_lft 21552sec
```

Пример изменения сетевых параметров, получаемых от DHCP сервера

В файле `/etc/dhcp/dhclient.conf` находятся параметры DHCP клиента.

Например, чтобы получать только IP v4 адрес нужно выполнить следующие шаги.

1. Замените в файле `/etc/dhcp/dhclient.conf`:

```
request subnet-mask, broadcast-address, time-offset, routers,
       domain-name, domain-name-servers, domain-search, host-name,
       dhcp6.name-servers, dhcp6.domain-search, dhcp6.fqdn, dhcp6.sntp-servers,
       netbios-name-servers, netbios-scope, interface-mtu,
       rfc3442-classless-static-routes, ntp-servers;
```

на:

```
request subnet-mask, broadcast-address;
```

2. Выключите и включите сетевой интерфейс, на котором настроено получение сетевых параметров по DHCP:

```
root@Hub1:~# ifdown ethN
root@Hub1:~# ifup ethN
```

Пример индивидуальной настройки DHCP клиента при использовании stdhcpclient

Утилита `stdhcpclient` предназначена для управления получением сетевых параметров по DHCP на указанном сетевом интерфейсе. Она позволяет использовать индивидуальные конфигурационные файлы DHCP клиента для конкретных интерфейсов, а не единые настройки для всех интерфейсов системы.

Например, чтобы настроить получение IP v4 адреса с желаемым временем аренды в 360 секунд с использованием `stdhcpclient` нужно выполнить следующие шаги.

1. Выключите сетевой интерфейс, на котором планируется использование индивидуальных параметров DHCP:

```
root@Hub1:~# ifdown eth2
```

2. Уберите из секции начало и конец которой помечены маркерами `###netifcfg-begin###` и `###netifcfg-end###`, которая находится в файле `/etc/network/interfaces`, все строки, соответствующие интерфейсу eth2.

3. Добавьте настройки в файл `/etc/network/interfaces`:

```
auto eth2
iface eth2 inet manual
post-up stdhcpclient start -i $IFACE
pre-down stdhcpclient stop -i $IFACE
mtu 1500
```

4. Создайте конфигурационный файл `/etc/dhcp/dhclient-eth2.conf` на основе шаблона `/opt/stmodems/docs/dhclient-usb0.conf.example`:

```
# Configuration file for /sbin/dhclient.
#
# This is a sample configuration file for dhclient. See dhclient.conf's
#       man page for more information about the syntax of this file
#       and a more comprehensive list of the parameters understood by
#       dhclient.
#
# Normally, if the DHCP server provides reasonable information and does
#       not leave anything out (like the domain name, for example), then
#       few changes must be made to this file, if any.
#

option rfc3442-classless-static-routes code 121 = array of unsigned integer 8;

send host-name = gethostname();
request subnet-mask, broadcast-address, time-offset, routers,
        domain-name, domain-name-servers, domain-search, host-name,
        dhcp6.name-servers, dhcp6.domain-search, dhcp6.fqdn, dhcp6.sntp-servers,
        netbios-name-servers, netbios-scope, interface-mtu,
        rfc3442-classless-static-routes, ntp-servers;

# Required for faster address synchronization when used IP passthrough
send dhcp-lease-time 360;
```

Примечание – если конфигурационный файл для интерфейса `eth2` не будет создан, утилита `stdhcpclient` завершит работу с ошибкой. Этот файл обязателен для корректной работы утилиты.

5. Включите сетевой интерфейс:

```
root@Hub1:~# ifup eth2
```

6. Проверьте, что получили IP v4 адрес с желаемым временем аренды:

```
root@Hub1:~# ip address show dev eth2
```

```
4: eth2: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 10000
    link/ether 00:50:56:9e:a2:12 brd ff:ff:ff:ff:ff:ff
    altname enp19s0
    altname ens224
    altname TenGigabitEthernet0_2
    inet 10.0.101.30/16 brd 10.0.255.255 scope global dynamic eth2
       valid_lft 358sec preferred_lft 358sec
```

```
    inet6 fe80::d99:f022:dfe7:13ac/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
```

Интерфейс для начальной настройки шлюза (mgmt)

Начиная с версии 5.0 появилась возможность подключиться к «С-Терра Шлюз» по SSH (root/s-terra) на адрес 192.168.4.1/24 (задан на интерфейсе eth0) до инициализации и выполнить первоначальную настройку. Данный функционал в «С-Терра Шлюз DP» отсутствует. Ограничения по применению описаны в правилах пользования.

Особые случаи

Увеличение количества допустимых сетевых интерфейсов

Если на «С-Терра Шлюз» настроено большое количество сетевых интерфейсов (>96) и в "klogview -f drop" наблюдаются сообщения "unknown physical network interface", например:

```
"dropped out packet 10.20.0.121:t8c0 -> 10.20.0.122:t8c0, proto 1, len 1028, if 195935983: unknown physical network interface"
```

то нужно увеличить хэш-таблицу сетевых интерфейсов в драйвере Продукта.

Пример:

```
root@Hub1:~# vim.tiny /etc/modprobe.d/vpndrvr.conf
```

```
...
options vpndrvr pcap_if_cache_bucket_size=2048
options vpndrvr pcap_if_uncached_bucket_size=2048
...
```

Максимальное количество хранимых сетевых интерфейсов для Шлюза, при равномерном распределении хэш-значений имен, вычисляется по формуле

`2 ^ pcap_if_cache_hash_bits * pcap_if_cache_bucket_size + pcap_if_uncached_bucket_size`

где:

`pcap_if_cache_hash_bits` - количество бит хэш-таблицы сетевых интерфейсов (в версии 5.0.25337 равно 4, не изменяется, изменить можно начиная с 5.0.25727)

`pcap_if_cache_bucket_size` - максимальное количество хранимых сетевых интерфейсов с одинаковым хэш-значением (число 8-2048, по умолчанию 32)

`pcap_if_uncached_bucket_size` - максимальное количество хранимых сетевых интерфейсов, для хэш-значения которых не осталось места в хэш-таблице (число 8-2048, по умолчанию 64).

Изменение MAC адреса интерфейса

В случаях, когда необходимо вручную назначить MAC адрес для интерфейса, воспользуйтесь следующим методом:

1. Создайте скрипт `/etc/network/if-pre-up.d/05-set-hwaddr` со следующим содержимым:

Важно! Название скрипта не должно содержать символ «.», в противном случае скрипт не будет работать.

```
#!/bin/sh

IFACE_TO_EDIT=eth0
HWADDR=00:08:da:63:e0:ad
```

```
IP=/usr/sbin/ip

if [ "$IFACE" != "$IFACE_TO_EDIT" ]; then
    exit 0
fi

$IP link set dev "$IFACE" address "$HWADDR"
```

Где:

- Параметр `IFACE_TO_EDIT` содержит имя интерфейса, чей MAC адрес следует изменить;
- Параметр `HWADDR` содержит MAC адрес, который будет назначен интерфейсу.

2. Установите права на исполнение:

```
root@Hub1:~# chmod a+x /etc/network/if-pre-up.d/05-set-hwaddr
```

3. Для применения изменений перезагрузите устройство или выключите/включите соответствующий интерфейс с помощью команд `ifdown` и `ifup`.