УТВЕРЖДЕН

BY.СЮИК.00371-02 34 01-ЛУ

КОМПЛЕКС ПРОГРАММНЫЙ РЕАЛИЗАЦИИ ПРОТОКОЛОВ IPSEC STRONGSWANCONT

Руководство оператора

BY.СЮИК.00371-02 34 01

Листов 37

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл.	

2021

№ изм.	*Подп.*	*Дата*

Литера O_1

АННОТАЦИЯ

Настоящий документ содержит сведения, необходимые оператору (администратору), для настройки, запуска и управления «Комплексом программным реализации протоколов IPsec strongSwanCont» BY.СЮИК.00371-02 (далее – strongSwanCont).

Для понимания материала, изложенного в документе, необходимы: знание основ криптографии в рамках нормативных документов Республики Беларусь, приведенных в ГОСТ 28147-89, СТБ 34.101.31-2020, СТБ 34.101.45-2013, СТБ 34.101.47-2017, СТБ 34.101.60-2014 и СТБ 34.101.66-2014; понимание принципов маршрутизации; знание набора протоколов IPsec; понимание ключевых моментов и общее представление политик обработки трафика в операционной системе (далее – ОС) Linux; умение администрирования ПЭВМ под управлением ОС Linux.

СОДЕРЖАНИЕ

1. НАЗНАЧЕНИЕ ПРОГРАММЫ

1.1. Программное обеспечение strongSwanCont предназначено для организации безопасного канала передачи данных, посредством аутентификации участников, обмена данными и согласования между ними параметров безопасности (используемых криптографических алгоритмов, сеансовых ключей и т.д.).

1.2. StrongSwanCont предназначен для решения задачи организации безопасного канала передачи данных путем их шифрования с использованием криптографических алгоритмов на основе протоколов IPsec.

1.3. Под IPsec понимается набор протоколов, предназначенных для обеспечения безопасности данных, передаваемых по сетевому протоколу IP. IPsec и протоколы, входящие в его состав, описаны в ряде стандартов. Сам IPsec определён в RFC 2401, а его составные части описаны в более чем 10 документах, отметим лишь те, в которых определяются основные протоколы, а именно: RFC 2402 (AH), RFC 2406 (ESP), RFC 2409 (IKE). Кратко рассмотрим эти протоколы:

– Authentication Header (AH) – обеспечивает целостность и подлинность передаваемых данных посредством применения криптографических алгоритмов контроля целостности и подлинности;

– Encapsulating Security Protocol (ESP) – обеспечивает конфиденциальность, целостность и подлинность передаваемых данных, посредством применения криптографических алгоритмов шифрования, контроля целостности и подлинности;

– Internet Key Exchange (IKE) – обеспечивает взаимную аутентификацию сторон обмена данными и согласование между ними параметров безопасности.

1.4. В настоящий момент существуют две версии протокола IKE – IKEv1 и IKEv2. StrongSwanCont поддерживает обе реализации.

Коммуникация между сторонами-участниками протокола IKE осуществляется при помощи пар сообщений «запрос-отклик». Взаимодействие всегда начинается с, так называемых, начальных обменов – IKE_SA_INIT и IKE_SA_AUTH. Начальные обмены, как правило, состоят из четырёх сообщений (по два на один обмен), но в некоторых сценариях работы протокола это число может расти.

При помощи первой пары сообщений (IKE_SA_INIT) стороны согласуют криптографические алгоритмы, осуществляет обмен специальными значениями *nonce* и выполняют протокол Диффи-Хеллмана.

При помощи второго обмена (IKE_SA_AUTH) стороны аутентифицируют сообщения предыдущего обмена, обмениваются идентификационной информацией (например, сертификатами открытого ключа), аутентифицируют друг друга. Компоненты сообщений

IKE_SA_AUTH шифруются и защищаются от модификации с использованием ключей, выработанных при обмене IKE_SA_INIT, это обеспечивает конфиденциальность, целостность и подлинность идентификационных данных сторон.

Успешное завершение этих обменов устанавливает между сторонами ассоциацию безопасности IKE (далее – IKE SA), содержащую общую (для сторон) секретную информацию, которая может быть использована для установления ассоциаций безопасности AH и/или ESP (далее – Child SA), и первую (в большинстве случаев – единственную) Child SA. В рамках одной IKE SA может содержаться более одной Child SA. В рамках IKE SA защищаются обменные сообщения IKE, в рамках Child SA защищается весь трафик сетевого уровня.

Помимо описанных начальных обменов, в IKE существуют обмены CREATE_CHILD_SA и INFORMATIONAL. Обмен CREATE_CHILD_SA может инициироваться любой из сторон после завершения начальных обменов и состоит из двух сообщений. Этот обмен предназначен для создания новых Child SA и обновления ключей, используемых IKE SA и Child SA. Обновление ключей ассоциации безопасности (далее – SA), осуществляется посредством создания новой SA и удаления старой.

Для обеспечения возможности передачи сторонами различных управляющих сообщений (например, сообщений об ошибках) введён обмен INFORMATIONAL. Этот обмен может осуществляться только после завершения начальных обменов и, как следствие, организации криптографической защиты сообщений обмена.

2. УСЛОВИЯ ПРИМЕНЕНИЯ ПРОГРАММЫ

Исходные тексты strongSwanCont написаны на языке программирования Си. StrongSwanCont может эксплуатироваться в ОС Linux. Скорость выполнения функций strongSwanCont зависит от производительности ПЭВМ. Минимальный состав программных средств, необходимых для функционирования strongSwanCont включает в себя:

– ОС Linux;

– «Библиотеку криптографических преобразований ContactCrypto32LE» BY.СЮИК.00365-04 (далее – библиотека **ContactCrypto32LE**);

– «Библиотеку взаимодействия с комплексом программно-аппаратным защиты информации от несанкционированного доступа «Барьер – USB» BY.СЮИК.00383-01 (далее - библиотека **libkeystorage**);

– свободно распространяемую библиотеку **libusb-1.0.20**;

– свободно распространяемую библиотеку **libgmp**.

StrongSwanCont не предъявляет особых требований к аппаратной платформе. Требования к аппаратной платформе устанавливаются ОС (минимальными требованиями для ее установки и работы).

3. ВЫПОЛНЕНИЕ ПРОГРАММЫ

3.1. Криптографическая поддержка

Программное обеспечение strongSwanCont обеспечивает организацию безопасного канала передачи данных посредством протокола IKE с использованием следующих средств криптографической защиты информации:

– выработки и проверки ЭЦП в соответствии с СТБ 34.101.45 (*bign-with-hbelt*) с функцией хэширования в соответствии с СТБ 34.101.31 (*elt-hash256*);

– взаимной аутентификации участников обмена и выработки общего ключа шифрования данных по протоколу BSTS и BPACE в соответствии с СТБ 34.101.66 (*bake-bsts*, *bake-bpace*), а также Диффи-Хеллмана в соответствии с СТБ 34.101.66 Приложение А;

– прозрачного автоматического шифрования/расшифрования информации в соответствии с алгоритмами: ГОСТ 28147 в режимах гаммирования – *gost28147-ctr*, и гаммирования с обратной связью – *gost28147-cfb*; СТБ 34.101.31 в режимах сцепления блоков – *belt-cbc256*, гаммирования с обратной связью – *belt-cfb256*, счетчика – *belt-ctr256*;

– контроля целостности и имитозащиты данных в соответствии с ГОСТ 28147-89 – *gost28147-mac*, СТБ 34.101.31 *belt-mac256*, СТБ 34.101.47 – *hmac-hbelt*;

– обновления ключа шифрования данных при помощи алгоритма преобразования ключа в соответствии с СТБ 34.101.31 – *belt-keyrep*.

В состав strongSwanCont входят:

1. Исполняемые модули.

2. Библиотеки.

3. Плагины.

Подробно структура strongSwanCont описана в документе «Комплекс программный реализации протоколов IPsec strongSwanCont. Описание программы» BY.СЮИК.00371-02 13 01.

3.2. Обращение к программе

StrongSwanCont предоставляет сценарий командной строки – *ipsec*, являющийся оболочкой над вызовами исполняемых модулей, входящих в состав strongSwanCont. Сценарий *ipsec* вызывается из командной строки следующим образом:

```
ipsec <command> [<argument>] [<options>]
```

Рассмотрим команды *ipsec* (*command*) с их аргументами (*argument*) и опциями (*options*):

– *start [<starter options>]* – вызов исполняемого модуля *starter*, где «*starter options*» – опциональные аргументы командной строки модуля *starter*;

– *stop* – отправка сигнала *SIGTERM* процессу *starter*;

– *restart [<starter options>]* – аналогично последовательной подаче команд *ipsec stop* и *ipsec start [<starter options>]* с интервалом в 2 секунды;

– *update* – отправка сигнала *SIGHUP* процессу *starter*;

– *reload* – отправка сигнала *SIGUSR1* процессу *starter*;

– *up <name>* – установка соединения с именем *<name>*; оболочка вызова *stroke up <name>*;

– *down <name>* – закрытые соединения с именем *<name>*; оболочка вызова *stroke down <name>*;

– *down <name>{n}* – закрытые *n*-ной Child SA соединения с именем *<name>*, поскольку *{n}* однозначно идентифицирует Child SA – имя (*name*) опционально (напр. *ipsec down conn{42}* или *ipsec down {42}*);

– *down <name>{*}* – закрытые всех Child SA соединения с именем *<name>*;

– *down <name>[n]* – закрытые *n*-ной IKE SA соединения с именем *<name>*, поскольку *[n]* однозначно идентифицирует IKE SA – имя (*name*) опционально (напр. *ipsec down conn[42]* или *ipsec down [42]*);

– *down <name>[*]* – закрытые всех IKE SA соединения с именем *<name>*;

– *route <name>* – установка в ядре ОС политики IPsec для соединения *<name>*. Первый пакет, соответствующий этой политике, синициирует установление соединения IKE. Оболочка вызова s*troke route <name>*;

– *unroute <name>* – удаление из ядра ОС политики IPsec для соединения *<name>*; оболочка вызова s*troke unroute <name>*;

– *status [<name>]* – вывод краткой информации о состоянии соединения *<name>* или, если аргумент отсутствует, обо всех соединениях. Оболочка вызова *stroke status [<name>]*;

– *statusall [<name>]* – вывод подробной информации о состоянии соединения *<name>* или, если аргумент отсутствует, обо всех соединениях. Оболочка вызова *stroke statusall [<name>]*;

– *version* – вывод версии strongSwanCont в виде:

Linux strongSwanCont U <версия strongSwanCont> /K <версия ядра Linux>;

– *--confdir* – вывод пути к директории, в которой хранятся настроечные файлы;

– *--directory* – вывод пути к директории, содержащей исполняемые модули StrongSwanCont;

– *--help* – вывод информацию о порядке использования сценария *ipsec*;

– *--versioncode* – вывод версии StrongSwanCont в виде:

U <версия StrongSwanCont> /K <версия ядра Linux>;

– *leases [<poolname> [<address>]]* – вывод состояния всех или выбранных пулов IP адресов; оболочка вызова *stroke leases [<poolname> [<address>]]*;

– *listaacerts [--utc]* – вывод списка X.509 *Authorization Authority* (AA) сертификатов, загруженных демоном *charon* из директории **ipsec.d/aacerts**. Оболочка вызова *stroke listaacerts*;

– *listacerts [--utc]* – вывод списка атрибутных сертификатов X.509, которые были загружены демоном *charon* из директории **ipsec.d/acerts**. Оболочка вызова *stroke listacerts*;

– *listalgs* – вывод списка поддерживаемых демоном *charon* криптографических алгоритмов; оболочка вызова *stroke listalgs*;

– *listcacerts [--utc]* – вывод списка корневых сертификатов X.509, которые были загружены демоном *charon* из директории **ipsec.d/cacerts** или получены во время обменов IKE. Оболочка вызова *stroke listcacerts*;

– *listcainfos [--utc]* – вывод информации о центрах сертификации, сконфигурированных в *ipsec.conf* (см. п. 4.1.1). Оболочка вызова *stroke listcainfos*;

– *listcerts [--utc]* – вывод списка оконечных X.509 сертификатов открытого ключа, которые были загружены демоном *charon* или полученны во время обменов IKE. Оболочка вызова *stroke listcerts*;

– *listcounters [<name>]* – вывод значений счётчиков IKE; оболочка вызова *stroke listcounters [<name>]*;

– *listcrls [--utc]* – вывод списков отзыва сертификатов, которые были загружены демоном *charon* из каталога **ipsec.d/crls** или получены по HTTP- или LDAP- из точки распространения COC. Оболочка вызова *stroke listcrls*;

– *listocsp [--utc]* – вывод кэшированной информации о статусе отзыва сертификатов, полученной с OCSP серверов. Оболочка вызова *stroke listocsp*;

– *listocspcerts [--utc]* – вывод списка полномочных OCSP сертификатов, которые были загружены демоном *charon* из каталога **ipsec.d/ocspcerts** или были получены от OCSP. Оболочка вызова *stroke listocspcerts*;

– *listplugins* – вывод списка плагинов, загруженных демоном *charon*. Оболочка вызова *stroke listplugins*;

– *listpubkeys [--utc]* – вывод списка открытых ключей, загруженных демоном *charon* в сыром виде. Оболочка вызова *stroke listpubkeys*;

– *listall [--utc]* – последовательно выполняет все, указанные выше, *list*- команды. Многие из *list*- команд могут быть вызваны с опцией *--utc*, переводящей все даты из локального времени в UTC. Оболочка вызова *stroke listall*;

– *rereadaacerts* – чтение сертификатов, содержащихся в каталоге **ipsec.d/aacerts**, и добавление их в список Authorization Authority (AA) сертификатов. Оболочка вызова *stroke rereadaacerts*;

– *rereadacerts* – чтение сертификатов, содержащихся в каталоге **ipsec.d/acerts**, и добавление их в список атрибутных сертификатов. Оболочка вызова *stroke rereadacerts*;

– *rereadcacerts* – чтение сертификатов, содержащихся в каталоге **ipsec.d/cacerts**, и добавление их в список корневых сертификатов. Оболочка вызова *stroke rereadcacerts*;

– *rereadcrls* – чтение списков отзыва сертификатов, содержащихся в каталоге **/etc/ipsec.d/crls**, и добавление их в соответствующий список. Оболочка вызова *stroke rereadcrls*;

– *rereadocspcerts* – чтение сертификатов, содержащихся в каталоге **ipsec.d/ocspcerts**, и добавление их в список OCSP сертификатов. Оболочка вызова *stroke rereadocspcerts*;

– *rereadsecrets* – чтение личных ключей и секретов, определенных в файле **ipsec.secrets** (см. п. 4.1.3), и добавление их в соответствующий список. Оболочка вызова *stroke rereadsecrets*;

– *secrets* – эквивалентно *ipsec rereadsecrets*;

– *rereadall* – последовательное выполнение всех *read-* команд, перечисленных выше. Оболочка вызова *stroke rereadall*;

– *resetcounters [<name>]* – сброс всех или привязанных к соединению *name* счётчиков IKE. Оболочка вызова *stroke resetcounters [<name>]*;

– *purgecerts* | *purgecrls* | *purgeocsp* – очистка кэшей сертификатов, списков отзыва, записей OCSP; оболочка вызовов *stroke purgecerts, stroke purgecrls* и *purgeocsp* соответственно;

– *purgeike* – удаление всех IKE SA не имеющих ни одной Child SA; оболочка вызова *stroke purgeike*.

3.3. Настройка программы

3.3.1. Общие сведения

Настройка параметров IPsec соединения strongSwanCont происходит, путем заполнения файл конфигурации **/usr/local/etc/ipsec.conf**.

Файл конфигурации **/usr/local/etc/ipsec.conf** состоит из трех типов разделов:

– **config setup** определяет общие параметры конфигурации

– **conn <name>** определяет соединение

– **ca <name>** определяет центр сертификации

Файл может содержать только один раздел **config setup**, но неограниченное количество разделов **conn** и **ca**.

Все параметры, принадлежащие разделу, должны иметь отступ не менее одного пробела или символа табуляции. Остальная часть строки после символа «#» рассматривается как комментарий. Комментарии в разделе также должны быть с отступом.

3.3.2. Повторное использование существующих параметров

Все секции **conn** и **ca** наследуют параметры, определенные в секциях **conn %default** или **ca %default**, соответственно.

Параметры, определенные в других разделах **conn** или **ca**, могут быть включены в раздел с параметром **also = othersection**. Включенный раздел может, в свою очередь, использовать ключевое слово **also** для включения других разделов.

Один и тот же параметр может быть определен несколько раз в одном и том же разделе, при этом будет использоваться последнее значение. Не имеет значения, определены ли параметры до или после оператора **also**, параметры в текущем разделе всегда переопределяют унаследованные параметры. Но если в одном и том же разделе используются множественные операторы **also**, их порядок имеет значение (настройки из раздела, включенного позже, переопределяют настройки из ранее включенных разделов). Также можно сбрасывать настройку, не назначая значения (например, **leftcert** =), будет применяться значение настройки по умолчанию, если оно есть, которое может использоваться для «удаления» унаследованных настроек, например от **conn %default**.

3.3.3. Пример заполнения файла ipsec.conf

```
server@server:~$ sudo nano /usr/local/etc/ipsec.conf

config setup
        charondebug = "ike 1, lib 1, cfg 1"

# Add connections here.
conn %default
        keyexchange = ikev2
        ikelifetime = 24h
        lifetime = 1h
        rekeymargin = 3m
        mobike = no
        ike = belt_cfb-belt_mac-prfbrng_ctr-modp2048-keyrep
        esp = belt_cfb-belt_mac
        left = 100.0.0.1
        leftsubnet = 10.0.0.0/24
        leftid = %any
        leftcert = cert00001.cer
        leftauth = eap-bsts
        auto = route
```

```
conn simple1
        right = 100.0.0.2
        rightsubnet = 20.0.0.0/24
        rightid = %any
        rightauth = eap-bsts
        rightsendcert = never

conn simple2
        also = simple1
        right = 100.0.0.3
        rightsubnet = 30.0.0.0/24
```

3.3.4. Параметры раздела config setup

cachecrls = yes | **no**

Если параметр включен, то список отозванных сертификатов (СОС) подгруженный через HTTP или LDAP будет кэшироваться в каталоге **/usr/local//etc/ipsec.d/crls/** под уникальным именем файла, полученным от удостоверяющего центра.

charondebug = <debug list>

Параметр устанавливает события аудита и количество потоков для отладки. Указывается в форме разделённого запятыми списка, содержащего в себе пары тип_аудита/уровень_аудита, например: **dmn 3, ike 1, net -1**.

Демон IKE поддерживает числовые уровни ведения журнала (от -1 до 4):

-1: абсолютно тихий (отключение аудита от источника);

0: очень простые журналы аудита (например, SA up / SA down);

1: общий поток управления с ошибками, по умолчанию, хорош для того, чтобы увидеть, что происходит;

2: более подробный поток управления, для отладки;

3: включение в журнал дампов в шестнадцатеричном формате (RAW);

4: включение в журнал чувствительного материала, приватных данных.

Каждое сообщение журнала также имеет источник, от которого оно получено для записи в журнал. В настройках могут быть указаны следующие источники:

app: приложения, кроме демонов;

asn: низкоуровневое кодирование / декодирование (ASN.1, X.509 и т. д.);

cfg: управление конфигурацией и плагинами;

chd: CHILD_SA/IPsec SA;

dmn: настройка / очистка / обработка сигналов основного демона;

enc: операции кодирование / декодирование, шифрования / расшифрования пакетов;

esp: сообщения библиотеки libipsec;

ike: IKE_SA/ISAKMP SA;

imc: контроль целостности;

imv: проверка целостности;

job: работа очереди / процессов и управление потоками;

knl: работа сетевого интерфейса ядра для IPsec;

lib: сообщения библиотеки libstrongwan;

mgr: управление IKE_SA, обработчик синхронизации для доступа IKE_SA;

net: сетевая связь в IKE;

pts: сервис доверенной платформы;

tls: сообщения библиотеки libtls;

tnc: доверенное сетевое соединение.

Такое количество источников журналирования при работе избыточно. Установка большого количества источников и высокого уровня журналирования приводит к усложнению поиска информации в журнале, в связи с его объемом.

charonstart = **yes** | no

Настройка автоматического запуска (демоном-диспетчером) IKE-демона **charon.** Значение по-умолчанию: "**yes**".

strictcrlpolicy = yes | ifuri | **no**

Параметр определяет, обновлять ли каждый раз список отозванных сертификатов при попытке аутентификации. IKE-v2 дополнительно понимает значение "**ifuri**", которое выставляется в "**yes**" в том случае, если указан хотя бы один ресурс URI для COC и выставляется в "**no**" в том случае, если ни один URI не известен.

uniqueids = **yes** | no | never | replace | keep

Параметр определяет должен ли конкретный идентификатор участника оставаться уникальным. Для каждой новой IKE_SA, будут заменены все старые значения идентификаторов на новые. Идентификаторы участников обычно являются уникальными, поэтому новый IKE_SA, использующий тот же идентификатор, почти всегда предназначен для замены старого.

Если значение параметра установлено в "**no**", то демон будет заменять старые IKE_SA, при получении сообщения с уведомлением "**INITIAL_CONTACT**", а если значение параметра установлено в "**never**", то демон будет просто полностью игнорировать эти уведомления.

Параметр также принимает значение "**replace**", которое идентично "**yes**", и значение "**keep**" для отклонения новых установок IKE_SA и сохранения ранее созданных.

3.3.5. Параметры раздела conn <name>

3.3.5.1. Общие параметры подключения

ah = <cipher suites>

Параметр определяет разделенный тире список алгоритмов AH, которые будут использоваться для соединения, например, **sha1-sha256-modp1024**.

В IKEv2 могут быть включены несколько алгоритмов одного типа (разделенных "-") в одно предложение. IKEv1 включает только первый алгоритм в предложении.

Можно использовать только ключевое слово **ah** или **esp**, пакеты AH + ESP не поддерживаются.

Набор шифров AH по умолчанию отсутствует, поскольку по умолчанию используется ESP. Демон добавляет свое предложение по умолчанию к сконфигурированному значению. Чтобы ограничить его настроенным предложением, в конце можно добавить восклицательный знак (!).

Примечание. В качестве респондента по умолчанию демон выбирает первое настроенное предложение, которое также поддерживается партнером. Чтобы указать респонденту принимать только определенные наборы шифров, можно использовать флаг строгого режима (!, Восклицательный знак), например, **sha256-sha512-modp2048!**.

aggressive = yes | **no**

Режим использования IKEv1 –агрессивный или основной (по умолчанию).

also = <name>

Включает параметры раздела **conn** <**name**>.

authby = **pubkey** | rsasig | ecdsasig | psk | secret | xauthrsasig | xauthpsk | never

Параметр определяет способ аутентификации strongSwanCont друг перед другом; допустимые значения: **secret**" или "**psk**" для общих секретных ключей, **pubkey**" (по умолчанию) для сигнатур с открытым ключом, а также синонимы "**rsasig**" для цифровых подписей RSA и

"**ecdsasig**" для сигнатур DSA эллиптической кривой, **never**" может использоваться, если согласование никогда не должно быть предпринято или принято.

Цифровые подписи во всех отношениях превосходят общие секреты. IKEv1 дополнительно поддерживает значения "**xauthpsk**" и "**xauthrsasig**", которые будут включать расширенную аутентификацию (XAuth) в дополнение к основному режиму IKEv1 на основе общих секретов или цифровых подписей RSA соответственно.

Этот параметр не рекомендуется для подключений IKEv2, поскольку двум узлам не требуется согласовывать метод проверки подлинности. Вместо этого рекомендуется использовать параметр **left** | **righttauth** для определения методов аутентификации.

auto = **ignore** | add | route | start

Параметр определяет, какая операция должна выполняться автоматически при запуске IPsec:

"**add**" загружает соединение без его запуска.

"**route**" загружает соединение и устанавливает ловушки ядра. Если трафик обнаружен между **leftsubnet** и **rightsubnet**, соединение установливается.

"**start**" загружает соединение и немедленно устанавливает его.

"**ignore**" игнорирует соединение. Это равносильно удалению соединения из файла конфигурации.

Относится только локально, другой конец не должен согласовывать это.

closeaction = **none** | clear | hold | restart

Параметр определяет действие, которое необходимо предпринять, если удаленный узел неожиданно закрывает CHILD_SA. Закрытие не должно использоваться, если одноранговый узел использует повторную аутентификацию или проверку уникальных идентификаторов, так как эти события могут инициировать определенное действие, когда это нежелательно.

compress = yes | **no**

Параметр устанавливает, производить ли сжатие содержимого IPComp для соединения (сжатие на уровне канала не работает с зашифрованными данными, поэтому для эффективности сжатие должно выполняться до шифрования). Значение "**yes**" заставляет демон применять сжатие. Значение "**no**" запрещает демону применять сжатие.

dpdaction = **none** | clear | hold | restart

Параметр управляет использованием протокола обнаружения мертвых узлов (DPD, RFC 3706), в котором периодически отправляются уведомления R_U_THERE (IKEv1) или пустые информационные сообщения (IKEv2) для проверки живости узла IPsec. Значения "**clear**", "**hold**", "**restart**" активируют DPD и определяют действие, которое нужно выполнить по таймауту: "**clear**" – соединение закрывается без дальнейших действий, "**hold**" – устанавливает политику прерываний, которая будет перехватывать соответствующий трафик и пытается повторно согласовать соединение по требованию, "**restart**" – немедленно вызовет попытку пересмотреть соединение. По умолчанию установлено значение "**none**", что отключает активную отправку сообщений DPD.

dpddelay = **30s** | <time>

Параметр определяет интервал времени, с которым информационные сообщения R_U_THERE отправляются одноранговому узлу.

Они отправляются только в том случае, если другой трафик не получен. В IKEv2 значение "**0**" не отправляет никаких дополнительных информационных сообщений и использует только стандартные сообщения (например, для повторного определения ключа) для обнаружения мертвых узлов.

dpdtimeout = **150s** | <time>

Параметр определяет интервал времени ожидания, после которого все соединения с одноранговым узлом удаляются в случае неактивности. Это относится только к IKEv1. В IKEv2 время ожидания повторной передачи устанавливается по умолчанию, поскольку каждый обмен используется для обнаружения мертвых узлов.

Время ожидания повторной передачи в демоне IKE можно настроить глобально с помощью параметров strongswan.conf.

Следующие ключи используются для настройки поведения повторной передачи:

Параметр	Значение по умолчанию	Описание
charon.retransmit_tries	5	Количество повторных передач до принятия решения об отказе (n)
charon.retransmit_timeout	4.0	Время ожидания ответа в секундах (t)
charon.retransmit_base	1.8	База экспоненциального отката (b)

Относительное время ожидания повторной передачи (ΔT_{dpd}) рассчитывается по формуле:

$\Delta T_{dpd} = t \cdot b^{(n-1)}$

Таким образом:

Количество повторных передач	**Формула**	**Относительное время ожидания**	**Абсолютное время ожидания**
1	$4 \cdot 1{,}8^0$	4 с	4 с
2	$4 \cdot 1{,}8^1$	7 с	11 с
3	$4 \cdot 1{,}8^2$	13 с	24 с
4	$4 \cdot 1{,}8^3$	23 с	47 с
5	$4 \cdot 1{,}8^4$	42 с	89 с
и т. д.	$4 \cdot 1{,}8^5$	76 с	165 с

eap_identity = <id>

Параметр определяет идентификатор, который клиент использует для ответа на запрос идентификатора EAP. Если на сервере установлена аутентификация по EAP, идентификатор будет использоваться для идентификации однорангового узла во время аутентификации по EAP. Если значение не определено, в качестве идентификатор EAP будет использоваться идентификатор IKEv2.

esp = <cipher suites>

Параметр определяет разделенный тире список алгоритмов шифрования / аутентификации ESP, которые будут использоваться для соединения, например **aes128-sha256**. Обозначение: **алгоритм шифрования-алгоритм контроля целостности[-группа Диффи-Хеллмана][- esnmode]**.

В IKEv2 могут быть включены несколько алгоритмов одного типа (разделенных "-") в одно предложение. IKEv1 включает только первый алгоритм в предложении. Можно использовать только ключевое слово ah или esp, пакеты AH + ESP не поддерживаются.

По умолчанию (параметр **esp** не включен в файл **ipsec.conf**)используется **belt_cfb-belt_mac**. Демон добавляет свое предложение к этому значению по умолчанию или к настроенному значению. Чтобы ограничить его настроенным предложением, в конце можно добавить восклицательный знак (!).

Примечание. В качестве респондента по умолчанию демон выбирает первое настроенное предложение, которое также поддерживается партнером. Чтобы указать респонденту принимать только определенные наборы шифров, можно использовать флаг строгого режима (!, Восклицательный знак), например: **aes256-sha512-modp4096!**.

В связи с этим, если указать неизвестное значение (указать неподдерживаемый алгоритм или допустить ошибка в написании параметра), будет принято первое поддерживаемое значение.

Если указана dh-группа, пересмотр CHILD_SA и начальное согласование включают отдельный обмен по протоколу Диффи-Хеллмана. Однако для IKEv2 ключи CHILD_SA, созданные неявно с помощью IKE_SA, всегда будут получены из материала ключа IKE_SA. Таким образом, любая указанная здесь группа DH будет применяться только в том случае, если CHILD_SA позднее будет переименован или создан с отдельным обменом CREATE_CHILD_SA. Следовательно, несоответствие предложения может не сразу быть замечено при создании SA, но может позже привести к сбою повторного запроса.

Допустимыми значениями для **esnmode** являются "**esn**" и "**noesn**". Указание обоих согласовывает поддержку расширенного порядкового номера с партнером, по умолчанию "**noesn**".

Для соответствия использования strongSwanCont нормативным актам Республики Беларусь рекомендуется использовать следующие значения параметра esp.

EALG	
belt_cbc	алгоритм шифрования СТБ 34.101.31-2020 в режиме сцепления блоков
belt_cfb	алгоритм шифрования СТБ 34.101.31-2020 в режиме гаммирования с обратной связью
belt_ctr	алгоритм шифрования СТБ 34.101.31-2020 в режиме счётчика
belt_cbc_legacy	алгоритм шифрования СТБ 34.101.31-2020 в режиме сцепления блоков (для совместимости с первой версией ПАК «БАС»)
belt_cfb _legacy	алгоритм шифрования СТБ 34.101.31-2020 в режиме гаммирования с обратной связью (для совместимости с первой версией ПАК «БАС»)
belt_ctr legacy	алгоритм шифрования СТБ 34.101.31-2020 в режиме счётчика (для совместимости с первой версией ПАК «БАС»)
IALG	
belt_mac	алгоритм выработки иммитовставки СТБ 34.101.31-2020
belt_hmac	алгоритм ключезависимого хэширования СТБ 34.101.47-2017
belt_mac_legacy	алгоритм выработки иммитовставки СТБ 34.101.31-2020 (для совместимости с первой версией ПАК «БАС»)
belt_hmac_legacy	алгоритм ключезависимого хэширования СТБ 34.101.47-2017 (для совместимости с первой версией ПАК «БАС»)
DHGROUP	
ecp256bign	Алгоритм Диффи-Хеллмана с соответствии с СТБ 34.101.66-2014 Приложение А.
modp2048	(для совместимости с первой версией ПАК «БАС»)
Примечания: – жирным выделены алгоритмы, используемые по умолчанию; – курсивом выделены первые поддерживаемые значения.	

forceencaps = yes | **no**

Параметр устанавливает принудительную инкапсуляцию UDP для пакетов ESP, даже если ситуация с NAT не обнаружена. Актуально для IKEv1. Это может помочь преодолеть ограничительные брандмауэры.

fragmentation = **yes** | force | no

Параметр определяет, использовать ли фрагментацию IKE.

Фрагментированные сообщения, отправленные одноранговым узлом, всегда обрабатываются независимо от значения этого параметра (даже если установлено значение "**no**").

Если установлено значение "**yes**" и одноранговый узел поддерживает его, IKE-сообщения большого размера будут отправляться.

Если установлено значение "**force**" (поддерживается только для IKEv1), исходное сообщение IKE уже будет фрагментировано при необходимости.

ike = <cipher suites>

Параметр определяет разделенный тире список используемых алгоритмов шифрования / аутентификации IKE / ISAKMP SA, например, **aes128-sha256-modp3072**. Обозначение: **алгоритм шифрования-алгоритм контроля целостности[-PRF – датчик (псевдо)случайных чисел]-группа Диффи-Хеллмана[-алгоритм преобразования ключа]**.

В IKEv2 могут быть включены несколько алгоритмов и предложений, например **aes128-aes256-sha1-modp3072-modp2048,3des-sha1-md5-modp1024**.

Если PRF не настроен, алгоритм PRF определяется из алгоритма контроля целостности.

По умолчанию (параметр **ike** не включен в файл **ipsec.conf**) используется **belt_cfb-belt_mac-prg_brng_ctr_-ecp256bign**. Демон добавляет свое предложение по умолчанию к этому значению или к настроенному значению. Чтобы ограничить его настроенным предложением, в конце можно добавить восклицательный знак (!).

Примечание. В качестве респондента по умолчанию оба демона принимают первое поддерживаемое предложение, полученное от партнера. Чтобы указать респонденту принимать только определенные наборы шифров, можно использовать флаг строгого режима (!, Восклицательный знак), например: **aes256-sha512-modp4096!**.

В связи с этим, если указать неизвестное значение (указать неподдерживаемый алгоритм или допустить ошибка в написании параметра), будет принято первое поддерживаемое значение.

Для соответствия использования ПАК «БАС» нормативным актам Республики Беларусь рекомендуется использовать следующие значения параметра ike.

EALG	
belt_cbc	алгоритм шифрования СТБ 34.101.31-2020 в режиме сцепления блоков
belt_cfb	алгоритм шифрования СТБ 34.101.31-2020 в режиме гаммирования с обратной связью
belt_ctr	алгоритм шифрования СТБ 34.101.31-2020 в режиме счётчика
belt_cbc_legacy	алгоритм шифрования СТБ 34.101.31-2020 в режиме сцепления блоков (для совместимости с первой версией ПАК «БАС»)
belt_cfb _legacy	алгоритм шифрования СТБ 34.101.31-2020 в режиме гаммирования с обратной связью (для совместимости с первой версией ПАК «БАС»)
belt_ctr legacy	алгоритм шифрования СТБ 34.101.31-2020 в режиме счётчика (для совместимости с первой версией ПАК «БАС»)
IALG	
belt_mac	алгоритм выработки иммитовставки СТБ 34.101.31-2020
belt_hmac	алгоритм ключезависимого хэширования СТБ 34.101.47-2017
belt_mac_legacy	алгоритм выработки иммитовставки СТБ 34.101.31-2020 (для совместимости с первой версией ПАК «БАС»)
belt_hmac_legacy	алгоритм ключезависимого хэширования СТБ 34.101.47-2017 (для совместимости с первой версией ПАК «БАС»)
PRF	
prfbrng_ctr	алгоритм выработки псевдослучайных чисел СТБ 34.101.47-2017 в режиме счётчика
prfbrng_hmac	алгоритм выработки псевдослучайных чисел СТБ 34.101.47-2017 в режиме HMAC
DHGROUP	
ecp256bign	Алгоритм Диффи-Хеллмана с соответствии с СТБ 34.101.66-2014 Приложение А.
modp2048	(для совместимости с первой версией ПАК «БАС»)
KEYREP	
keyrep	алгоритм преобразования ключа СТБ 34.101.31-2020
Примечания: – жирным выделены алгоритмы, используемые по умолчанию; – курсивом выделены первые поддерживаемые значения.	

ikedscp = **000000** | <DSCP field>

Точка кода дифференцированных услуг (DSCP, Differentiated Services Code Point) для установки исходящих пакетов IKE, отправленных с этого соединения. Значение представляет собой шестизначную двоично-кодированную строку, определяющую устанавливаемую кодовую точку в соответствии с RFC 2474.

ikelifetime = **3h** | <time>

Параметр устанавливает время жизни ключей соединения (ISAKMP или IKE SA). После истечения которого, происходит повторное согласование.

installpolicy = **yes** | no

Параметр решает, установлены ли политики IPsec в ядре демоном charon для данного соединения.

Позволяет мирное сотрудничество, например с демоном Mobile IPv6 mip6d, который хочет управлять политиками ядра.

keyexchange = **ike** | ikev1 | ikev2

Параметр устанавливает метод обмена ключами; какой протокол следует использовать для инициализации соединения.

Значение "**ike**" по умолчанию является синонимом "**ikev2**".

keyingtries = **3** | <number> | %forever

Параметр устанавливает сколько попыток (положительное целое число или **%forever**) следует предпринять, чтобы договориться о соединении или заменить его, прежде чем отказаться (по умолчанию "**3**"). Значение "**%forever**" устанавливает бесконечное количество попыток соединения. Относится только локально, другой конец не должен согласовывать это.

lifebytes = <number>

Параметр устанавливает количество байтов, переданных через IPsec SA до истечения срока его действия. Для обеспечения высокого уровня гарантии безопасности данных, необходимо использовать значения, обеспечивающие квоту ключа. Рекомендуется использовать следующее значение параметра lifebytes = 11000000000000.

lifepackets = <number>

Параметр устанавливает количество пакетов, переданных через IPsec SA до истечения срока его действия.

lifetime = **1h** | <time>

Параметр устанавливает время жизни соединения (набор ключей шифрования / аутентификации для пользовательских пакетов) от успешного согласования до истечения срока действия; допустимыми значениями являются целое число, необязательно, за которым следует "**s**" (время в секундах) или десятичное число, за которым следуют "**m**", "**h**" или "**d**" (время в минутах, часах или днях соответственно) (по умолчанию 1 час, максимум 24 часа).

Обычно соединение повторно согласовывается (через канал ключей) до истечения срока его действия. Два конца не обязательно должны точно согласовывать время жизни, хотя, если они этого не делают, на конце, который думает, что время жизни больше, будет некоторый беспорядок замененных соединений.

Подробнее о смене ключей описано в Приложении А.

marginbytes = <number>

Параметр устанавливает количество байт оставшихся до истечения срока действия SA IPsec (см. lifebytes). При их достижении должны начинаться попытки согласования смены ключей.

marginpackets = <number>

Параметр устанавливает количество пакетов оставшихся до истечения срока действия SA IPsec (см. lifepackets). При их достижении должны начинаться попытки согласования смены ключей.

margintime = **9m** | <time>

rekeymargin = **9m** | <time>

Параметр устанавливает время до истечения срока действия соединения или истечения срока действия ключей. При его достижении должны начинаться попытки согласования смены ключей; допустимые значения, как для lifetime (по умолчанию "**9 м**"). Относится только локально, другой конец не должен согласовывать это. Параметр имеет два эквивалентных наименования: margintime и rekeymargin.

mark = <value>[/<mask>]

Параметр устанавливает метку XFRM для входящей и исходящей IPsec SA политики.

Если маска отсутствует, то принимается маска по умолчанию 0xffffffff.

mark_in = <value>[/<mask>]

Параметр устанавливает метку XFRM для входящей политики.

Если маска отсутствует, то принимается маска по умолчанию 0xffffffff.

mark_out = <value>[/<mask>]

Параметр устанавливает метку XFRM для исходящего IPsec SA и политики.

Если маска отсутствует, то принимается маска по умолчанию 0xffffffff.

mobike = **yes** | no

Параметр разрешает протокол IKEv2 MOBIKE, определенный в RFC 4555. Если установлено значение "**no**", демон charon не будет активно предлагать MOBIKE в качестве инициатора и игнорировать уведомление MOBIKE_SUPPORTED в качестве ответчика.

modeconfig = push | **pull**

Параметр определяет, какой режим используется для назначения виртуального IP-адреса. В настоящее время актуально только для KEv1, поскольку IKEv2 всегда использует данные конфигурации в режиме "**pull**". Шлюзы Cisco VPN обычно работают в режиме "**push**".

Этот параметр должен быть одинаковым с обеих сторон.

reauth = **yes** | no

Параметр устанавливает, должно ли повторное создание IKE_SA также повторно аутентифицировать одноранговый узел. В IKEv1 повторная аутентификация всегда выполняется.

В IKEv2 при значении "**no**" пересогласование ключей происходит без удаления SA IPsec, при значении "**yes**" (по умолчанию) новый IKE_SA создается с нуля и пытается воссоздать все SA IPsec.

rekey = **yes** | no

Параметр устанавливает, следует ли пересмотреть соединение, когда оно истекает. Оба конца не должны согласовываться, хотя значение "**no**" препятствует тому, чтобы демон запросил повторное согласование, оно не препятствует ответу на повторное согласование, запрошенное с другого конца, поэтому "**no**" будет в значительной степени неэффективным, если оба конца не согласятся с ним.

rekeyfuzz = **100%** | <percentage>

Параметр устанавливает максимальный процент, на который необходимо произвольно увеличивать маржинальные байты, маржинальные пакеты и маржинальное время для

рандомизации интервалов повторного ввода (важно для хостов с большим количеством соединений); допустимые значения - целое число, которое может превышать 100, за которым следует «%».

Значение "**marginTYPE**" после этого случайного увеличения не должно превышать "**lifeTYPE**" (где "**TYPE**" - один из вариантов: "**bytes**", "**packets**" или "**time**").

Значение "**0%**" будет подавлять рандомизацию. Относится только локально, другой конец не должен согласовывать это.

reqid = <number>

Параметр устанавливает идентификатор reqid для данного соединения в предварительно сконфигурированное фиксированное значение.

tfc = <value>

Параметр устанавливает количество байтов для заполнения данных полезной нагрузки ESP. Конфиденциальность трафика в настоящее время поддерживается в IKEv2 и применяется только к исходящим пакетам. Специальное значение % mtu заполняет ESP пакеты данными равными размеру MTU.

type = **tunnel** | transport | transport_proxy | passthrough | drop

Параметр устанавливает тип соединения; в настоящее время допустимыми значениями являются "**tunnel**", обозначающий туннель точка-точка, точка-сеть или сеть-сеть; "**transport**", обозначающий транспортный режим точка-точка; "**transport_proxy**", обозначающий специальный режим прокси-сервера Mobile IPv6; "**passthrough**", означающий, что обработка IPsec вообще не должна выполняться; "**drop**", означающий, что пакеты должны быть отброшены.

xauth = **client** | server

Параметр указывает роль в протоколе XAuth, если активирована с помощью authby = xauthpsk или authby = xauthrsasig.

xauth_identity = <id>

Параметр определяет идентификатор / имя пользователя, которое клиент использует для ответа на запрос XAuth. Если не определено, идентификатор IKEv1 будет использоваться как идентификатор XAuth.

3.3.5.2. Параметры конечной точки

Описания соединений определяются в терминах левой (**left**) конечной точки и правой (**right**) конечной точки. Например, два параметра **leftid** и **rightid** определяют идентификатор левой и правой конечной точки. Для каждого описания соединения делается попытка выяснить, должна ли локальная конечная точка действовать как левая или правая. Это делается путем сопоставления IP-адресов, определенных для обеих конечных точек, с IP-адресами, назначенными для локальных сетевых интерфейсов. Если совпадение найдено, то соответствующая роль (слева или справа) будет считаться «локальной». Если во время запуска совпадений не найдено, «**left**» считается «**local**».

left|right = <ip address> | <fqdn> | **%any** | range | subnet

Параметр определяет IP-адрес интерфейса общедоступной сети участника или одно из нескольких значений.

Значение "**%any**" для локальной конечной точки означает адрес, который должен быть заполнен (автоматическим вводом) во время согласования. Если локальный одноранговый узел инициирует настройку соединения, к таблице маршрутизации будет предложено определить правильный локальный IP-адрес. В случае, если локальный узел отвечает на настройку соединения, любой IP-адрес, назначенный локальному интерфейсу, будет принят.

Префикс "**%**" перед полностью определенным доменным именем или IP-адресом неявно установит **left|rightallowany = yes**.

Если "**%any**" используется для удаленной конечной точки, это буквально означает разрешение подключения любого IP-адреса.

Соединения могут быть ограничены конкретным диапазоном хостов. Для этого можно указать диапазон (10.1.0.0-10.2.255.255) или подсеть (10.1.0.0/16), а несколько адресов, диапазонов и подсетей можно разделить запятыми. Также можно свободно комбинировать эти элементы. Для инициирования соединения требуется, по крайней мере, одна запись адрес / диапазон / подсеть.

left|rightallowany = yes | **no**

Модификатор параметра **left | right**, который ведет себя как "**% any**", хотя конкретный IP-адрес был назначен. Рекомендуется для динамических IP-адресов, которые могут быть разрешены DynDNS при запуске или обновлении IPsec.

left|rightauth = <auth method>

Способ аутентификации, используемый локально (**left**) или требуемый от удаленной (**right**) стороны. Приемлемыми значениями являются: "**pubkey**" для аутентификации с открытым ключом (алгоритм Диффи-Хелманна СТБ 34.101.66-2014 Приложение А), "**psk**" для аутентификации с предварительным общим ключом, "**eap**" для использования расширяемого протокола аутентификации и "**xauth**" для расширенной аутентификации IKEv1.

StrongSwanCont поддерживает аутентификацию и выработку общего ключа в соответствии с протоколом **BSTS**, требования к которому установлены в п. 7.5 СТБ 34.101.66-2014 и **BPACE**, требования к которому установлены в п. 7.6 СТБ 34.101.66-2014.

Для включения аутентификации с помощью протокола BSTS используется значение "**eap-bsts**", BPACE используется значение "**eap-bpace**".

Механизмы аутентификации **pubkey**, **eap-bsts**, **eap-bpace** являются рекомендуемыми для соответствия нормативным актам Республики Беларусь.

EAP является клиент-серверным протоколом аутентификации. За выбором конкретного механизма EAP отвечает сервер, клиент лишь запрашивает аутентификацию при помощи EAP. В связи с этим на одной стороне (сервере) параметр должен принимать значение "**eap-bsts**", а на другой (клиенте) "**eap**".

В случае "**eap**" в strongSwanCont могут быть добавлены дополнительные методы EAP: eap-aka, eap-gtc, eap-md5, eap-mschapv2, eap-peap, eap-sim, eap-tls, eap-ttls, eap-dynamic и eap-radius. Однако, они не включены к комплект поставки.

left|rightauth2 = <auth method>

То же, что **left|righttauth**, но определяет дополнительный обмен аутентификацией. В IKEv1 только "**xauth**" может использоваться во втором раунде аутентификации. IKEv2 поддерживает несколько полных раундов аутентификации с использованием множественных обменов аутентификацией, определенных в RFC 4739. Это позволяет, например, выполнить отдельную аутентификацию хоста и пользователя.

left|rightca = <issuer dn> | %same

Параметр определяет отличительное имя центра сертификации, которое должно лежать на пути доверия, идущем от сертификата **left|righ** участника до корневого центра сертификации. Значение "**%same**" означает, что должно быть повторно использовано значение, настроенное для другого участника.

left|rightca2 = <issuer dn> | %same

Параметр определяет то же, что **left|rightca**, но для второго раунда аутентификации (только IKEv2).

left|rightcert = <path>

Параметр указывает путь к сертификату X.509 **left|right** участника. Файл может быть закодирован в формате PEM или DER. Также поддерживаются сертификаты OpenPGP.

Оба абсолютных пути или пути относительно **/usr/local/etc/ipsec.d/certs** принимаются. По умолчанию **left|rightcert** устанавливает **left|righttid** для отличительного имени субъекта сертификата. ID **left|right** участника можно переопределить, указав значение **left|righttid**, которое должно быть подтверждено сертификатом.

Параметр может быть представлен в виде списка, где через запятую можно указывать несколько путей сертификатов.

Демон выбирает сертификат на основе полученных запросов на сертификат, если это возможно, перед применением первого.

left|rightcert2 = <path>

Параметр определяет то же, что **left|rightcert**, но для второго раунда аутентификации (только IKEv2).

left|rightcertpolicy = <OIDs>

Разделенный запятыми список OID политики сертификата, который должен иметь сертификат партнера. OID указываются с использованием числового точечного представления.

left|rightdns = <servers>

Разделенный запятыми список адресов DNS-серверов для обмена в качестве атрибутов конфигурации. В инициаторе - это фиксированный адрес сервера IPv4 / IPv6 или %config4 / %config6 для запроса атрибутов без адреса.

На респонденте разрешены только фиксированные адреса IPv4 / IPv6, которые определяют DNS-серверы, назначенные клиенту.

left|rightfirewall = yes | **no**

Параметр устанавливает, выполняет ли **left|right** участник фильтрацию (включая masquerading) с использованием iptables для трафика из **left|rightsubnet** (который должен быть отключен для пересылки в другую подсеть) после установления соединения. Не может использоваться в том одном описании соединения вместе с параметром **left|rightupdown**. Реализовано в качестве параметра для скрипта ipsec_updown по умолчанию. Относится только локально, другой конец не должен согласовывать это.

Если один или оба strongSwanCont выполняют фильтрацию (возможно, включая masquerading), и это задается с использованием параметров брандмауэра, туннели, установленные с помощью IPsec, освобождаются от него, так что пакеты могут проходить без изменений через туннели. (Это означает, что все подсети, подключенные таким образом, должны иметь различные непересекающиеся блоки адресов подсети.) Это делается с помощью сценария ipsec_updown по умолчанию.

В ситуациях, требующих большего контроля, для пользователя может быть предпочтительным предоставить свой собственный скрипт, который вносит соответствующие изменения в его систему.

left|rightgroups = <group list>

Разделенный запятыми список имен групп. Если параметр **left|rightgroups** присутствует, то узел должен быть членом хотя бы одной из групп, определенных параметром. Группы могут использоваться вместе с плагином eap-radius.

left|rightgroups2 = <group list>

Параметр определяет то же, что **left|rightgroups**, но для второго раунда аутентификации (только IKEv2).

left|righthostaccess = yes | **no**

Параметр вставляет пару правил iptables INPUT и OUTPUT, используя скрипт ipsec_updown по умолчанию, что позволяет получить доступ к самому хосту в случае, когда внутренний интерфейс хоста является частью согласованной клиентской подсети.

left|rightid = <id>

Параметр устанавливает имя, под которым **left|right** участник должен быть идентифицирован для аутентификации (идентификатор); по умолчанию используется **left|right** или секция **subject** сертификата, подключенного с помощью **left|rightcert**. Если настроен **left|rightcert**, идентификатор должен быть подтвержден сертификатом, то есть он должен полностью соответствовать секции **subject** сертификата или одному из значений, содержащихся в расширении **subjectAltName**.

Это может быть IP-адрес, полное доменное имя, адрес электронной почты или отличительное имя, для которого тип идентификатора определяется автоматически, и строка преобразуется в соответствующую кодировку.

В определенных особых ситуациях приведенный выше анализ идентичности может быть неадекватным или привести к неверному результату.

Параметр для IKEv2 может опционально, включать "**%**" в качестве префикса перед идентификатором. Если он установлен, то демону запрещается отправлять ID в своем запросе IKE_AUTH и позволяет ему проверять соответствие установленных настроек и секции **subject** или **subjectAltName**, содержащихся в сертификате респондента (в противном случае он сравнивается только с идентификатором, возвращенным респондентом). ID, отправленный инициатором, может помешать ответчику найти конфигурацию, если он настроил другое значение для **leftid**.

left|rightid2 = <id>

Идентификатор для второго раунда аутентификации (только IKEv2). По умолчанию принимает значение **left|rightid**.

left|rightikeport = <port>

UDP-порт, который участник использует для связи IKE. Если не указано, используется порт 500 совместно с портом 4500, если обнаружен NAT или включен MOBIKE.

Указание локального порта IKE, отличного от значения по умолчанию, дополнительно требует реализации сокета, который прослушивает этот порт.

left|rightprotoport = <protocol>/<port>

Параметр устанавливает селектор трафика на один протокол и / или порт. Избыточный параметр, так как информация о протоколе / порте может быть определена для каждой подсети непосредственно в **left|rightsubnet**.

left|rightsendcert = never | no | **ifasked** | always | yes

Параметр устанавливает необходимость пересылки сертификата партнеру. Допустимые значения: "**never**" или "**no**", "**always**" или "**yes**", и "**ifasked**", последнее означает, что узел должен отправить запрос сертификата (CR), чтобы получить сертификат взамен.

leftsourceip = %config4 | %config6 | <ip address>

Внутренний IP-адрес для использования в туннеле, также известный как виртуальный IP-адрес.

Если в качестве значения установлено одно из следующих: "**%config**", "**%cfg**", "**%modeconfig**" или "**%modecfg**", то адрес запрашивается у партнера (из семейства адресов туннеля).

Список может принимать несколько значений адреса, перечисленных через запятую, при указании %config4 или %config6 адрес из данного семейства адресов будет запрашиваться явно.

Если IP-адрес настроен, он будет запрошен у респондента, который может ответить другим адресом.

rightsourceip = %config | <network>/<netmask> | <from>-<to> | %poolname

IP-адрес внутреннего источника для использования в туннеле для удаленного узла. Если установлено значение на стороне ответчика, инициатор должен предложить адрес, который затем возвращается. Также поддерживаются пулы адресов, выраженные как <network>/<netmask>.

Список IP-адресов / пулов, разделенных запятыми, принимается, например, для определения пулов различных семейств адресов.

left|rightsubnet = <ip subnet>[[<proto/port>]][,...]

Параметр устанавливает частную подсеть, находящуюся позади **left|right** участника, выраженную как сеть / маска сети; если опущено, то, по сути, предполагается, что **left/32|128**, что означает, что соединения устанавливается только между **left|right** участниками.

Сконфигурированные подсети одноранговых узлов могут отличаться, протокол сужает их до наибольшей общей подсети.

IKEv2 поддерживает несколько подсетей, разделенных запятыми, IKEv1 интерпретирует только первую подсеть такого определения. Это связано с ограничением протокола IKEv1, который допускает только одну пару подсетей на CHILD_SA. Таким образом, для туннелирования нескольких подсетей должна быть определена своя запись **conn** для каждой пары подсетей.

Необязательная часть после каждой подсети, заключенная в квадратные скобки, определяет протокол / порт для ограничения селектора для этой подсети. Например:

leftsubnet=10.0.0.1[tcp/http],10.0.0.2[6/80] или

leftsubnet=fec1::1[udp],10.0.0.0/16[/53].

Вместо того, чтобы пропускать значение, может быть использовано "**%any**" для того же эффекта. Например:

leftsubnet=fec1::1[udp/%any],10.0.0.0/16[%any/53]

Вместо указания подсети можно использовать "**%dynamic**", чтобы заменить ее адресом IKE, что будет иметь тот же эффект, что и полное исключение **left|rightsubnet**. "**%dynamic**" может использоваться для определения нескольких динамических селекторов, каждый из которых имеет потенциально различное определение протокола / порта.

left|rightupdown = <path>

Параметр устанавливает путь к скрипту, который необходимо запустить для настройки маршрутизации и / или межсетевого экрана при изменении состояния соединения (по умолчанию ipsec_updown). Относится только локально, другой конец не должен согласовывать это.

3.3.6. Параметры раздела ca <name>

Разделы **ca** - это необязательные разделы, которые можно использовать для назначения специальных параметров удостоверяющего центра (CA).

Поскольку демон автоматически импортирует корневые сертификаты из **/usr/local/etc/ipsec.d/cacerts**, нет необходимости явно добавлять раздел **ca**, если вы не хотите назначать специальные параметры (например, список отозванных сертификатов).

also = <name>

Включает параметры раздела **ca** <**name**>.

cacert = <path>

Параметр определяет путь к корневому сертификату либо относительно **/usr/local/etc/ipsec.d/cacerts**, либо как абсолютный путь.

crluri = <uri>

Параметр определяет точку распространения списка отозванных сертификатов (ldap, http или URI файла).

crluri1 = <uri>

Параметр-синоним **crluri**.

crluri2 = <uri>

Параметр определяет альтернативную точку распространения списка отозванных сертификатов (ldap, http или file URI).

ocspuri = <uri>

Параметр определяет точку распространения OCSP (URI OCSP сервера).

ocspuri1 = <uri>

Параметр-синоним **ocspuri**.

ocspuri2 = <uri>

Параметр определяет альтернативную точку распространения OCSP (URI OCSP сервера).

3.3.7. Настроечный файл ipsec.secrets

Настроечный файл **ipsec.secrets** создаётся автоматически при первом запуске strongSwanCont. Он предназначен для задания личных ключей и секретов.

Одна строка этого файла соответствует одному секретному параметру. Строки имеют следующий вид:

```
[ID] : [TYPE] [SECRET],
```

где: *ID* – идентификатор владельца секретного параметра (не указывается при использовании «для всех»);

TYPE – тип секретного параметра (EAP – при использовании eap-bpace);

SECRET – секретный параметр.

Отдельно необходимо упомянуть способ задания ПАК «Барьер – USB» в качестве хранилища личного ключа. Для этого в файл **ipsec.secrets** заносится строка вида:

```
: USBBAR<путь к защищённому паролю доступа>
```

Для того, чтобы задать файлы ключевых контейнеров в качестве хранилища личного ключа необходимо в файле **ipsec.secrets** указать строку вида:

```
: BPKI PRIV_KEY=<путь к контейнеру личного ключа>
SHARE_PWD=<путь к защищённому паролю доступа к частичным секретам>
SHARE_SECR1=<путь к контейнеру первого частичного секрета>
SHARE_SECR2=<путь к контейнеру второго частичного секрета>
...
SHARE_SECRN=<путь к контейнеру N-го частичного секрета>
```

4. СООБЩЕНИЯ ОПЕРАТОРУ

В процессе работы прикладное программное обеспечение strongSwanCont взаимодействует с оператором по средствам командной строки, принимая команды. strongSwanCont не выдает сообщения оператору. StrongSwanCont выполняет журналирования своей работы.

В случае ошибки на этапе выполнения согласования ключей в системном журнале появится запись об ошибке, с подробным описанием области ее возникновения и причины. Для отображения полной информации об ошибке можно установить более глубокий режим журналирования.

ПРИЛОЖЕНИЕ А

Квоты ключа

Ключи, согласованные для IKE и IPsec / CHILD SA, следует использовать только в течение ограниченного времени и для защиты ограниченного объема данных. Это означает, что каждая SA должна исчезнуть через определенное время жизни. Чтобы избежать прерываний, можно согласовать замену SA до того, как это произойдет, это называется «сменой ключей».

Алгоритмы шифрования остаются надежными до тех пор, пока соблюдаются квоты для используемых в них ключей. Квота ключа – это максимальный объем данных, которые разрешается зашифровывать на этом ключе. Квота задается количеством блоков зашифрованных сообщений, которые нельзя превысить.

В СТБ 34.101.31-2020 определены три уровня гарантий безопасности шифрования данных при использовании одного ключа.

Таблица А.1

Уровень гарантий	Количество безопасно обрабатываемых при использовании алгоритма					
	Belt-cbc		Belt-cfb		Belt-ctr	
	Блоков	Данных	Блоков	Данных	Блоков	Данных
Средний	2^{48}	4096 Тбайт	2^{48}	4096 Тбайт	$2^{48}\sqrt{\frac{2}{3}}$	3344 Тбайт
Высокий	2^{40}	16 Тбайт	2^{40}	16 Тбайт	$2^{40}\sqrt{\frac{2}{3}}$	13 Тбайт
Максимальный	2^{32}	64 Гбайт	2^{32}	64 Гбайт	$2^{32}\sqrt{\frac{2}{3}}$	52 Гбайт
Примечание: При расчетах применялись следующие соображения: 1 Кбайт = 1024 байт; 1 Мбайт = 1024 Кбайт; 1 Гбайт = 1024 Мбайт; 1 Тбайт = 1024 Гбайт;						

Настройки по умолчанию предполагают смену ключей (rekey) раз в час или при достижении объема данных в 10 Тбайт (lifebytes = 11000000000000), а также смену с повторной аутентификацией (reauth) раз в сутки.

Учитывая, что максимальный теоретический объем данных переданных через 1000 Мбит/с Ethernet сеть составляет 440 Гбайт/час или 10 Тбайт/сутки, а через 10 Гбит/с Ethernet сеть – 4,4 Тбайт/час или 105 Тбайт/сутки, смена ключей раз в час обеспечит высокий уровень гарантии безопасности данных при использовании любого поддерживаемого алгоритма в соответствии с СТБ 34.101.34-2020 в сетях 1000 Мбит/с Ethernet и 10 Гбит/с Ethernet, а смена ключей после обработки 10 Тбайт трафика – в любых сетях, вне зависимости от скорости передачи данных.

При необходимости обеспечить максимальный уровень гарантии безопасности необходимо установить соответствующее значение в параметр lifebytes, чтобы указать максимальный размер данных, которые можно обрабатывать до смены ключа. При этом ключ будет сменен по событию, которое наступит раньше: истечение времени lifetime или обработка порогового количества данных lifebytes.

Например, при использовании алгоритмов Belt-cbc или Belt-cfb необходимо установить порог в 64 Гбайт:

```
lifebytes = 68719476736
```

При использовании алгоритма Belt-ctr необходимо установить порог в 52 Гбайт:

```
lifebytes = 55834574848
```

Лист регистрации изменений									
Изм	Номера листов (страниц)				Всего листов (страниц) в докум.	№ документа	Входящий № сопроводительного докум. и дата	Подп.	Дата
	измененных	замененных	новых	аннулированных					