УТВЕРЖДЕНО

BY.РТНК.00001-04.1 34 01-9-ЛУ

Программно-аппаратный комплекс «Шлюз безопасности Bel VPN Gate 4.1»

Руководство администратора

Создание конфигурационного файла

BY.РТНК.00001-04.1 34 01-9

Листов 96

Подп. и дата	
Инв. № дубл.	
Взам. инд. №	
Подп. и дата	
Инд. № подл.	

2016

Содержание

Создание конфигурационного файла

Создание локальной политики безопасности для Bel VPN Gate возможно осуществить путем написания конфигурационного файла (в текстовом формате) для каждого устройства. Структуры конфигурационного файла предоставляют более широкие возможности для создания гибкой политики безопасности, чем возможности командной строки и графического интерфейса управления.

Созданную политику в виде конфигурационного файла нужно загрузить командой lsp_mgr load.

После загрузки конфигурационного файла, cisco-like конфигурация не изменится.

Описание грамматики LSP

Описание LSP представляет собой последовательное описание структур данных, определяемых типом, именем, списком параметров (полей) и их значений. Синтаксис языка определяет формат описания структур данных, базовые типы значений полей структур. Синтаксические конструкции позволяют описывать иерархические структуры данных, число уровней которых не ограничено.

Формальное описание синтаксиса LSP-языка в виде БНФ (Бэкуса—Наура форма) приведено ниже. В БНФ описании названия нетерминальных символов заключены в угловые скобки, имена терминалов написаны большими буквами. Кроме того, простые терминалы, ключевые слова и разделители, записаны в одинарных кавычках. В БНФ-описании используются следующие терминалы: ИДЕНТ, СТРОКА, DOTDOT, ЦЕЛОЕ32, ДАТА, ВРЕМЯ, IP.

<cfg_data>::= <top_level_form> | <cfg_data> <top_level_form>

<top_level_form>::= <object_def> | <constant>

<constant>::= `const` <key_value>

<object_def>::= ИДЕНТ ИДЕНТ `(` <key_value_or_template_list> `)`

| ИДЕНТ `(` <key_value_or_template_list> `)`

<key_value_list>::= <key_value> | <key_value> <key_value_list>

<key_value_or_template_list>::= <key_value_or_template_list>

|<key_value_or_template> <key_value_or_template_list>

<key_value_or_template>::= key_value | template

<key_value>::= <l_value> `=` <r_value_list>

<r_value_list>::= <r_value> | <r_value_list> `,` <r_value>

<r_value>::= ИДЕНТ | ИДЕНТ `<` `>`

| ИДЕНТ `<` <key_value_or_template_list> `>`

| ИДЕНТ `[` <r_value_list> `]`

| `(` <r_value_list> `)` | `(` `)`

| `[` <r_value_list> `]` | `[` `]`

| ИДЕНТ `(` <key_value_or_template_list> `)`

| СТРОКА

| ЦЕЛОЕ32 | ЦЕЛОЕ32 `..` ЦЕЛОЕ32

| ЦЕЛОЕ32 `/` ЦЕЛОЕ32 `/` ЦЕЛОЕ32

| - ЦЕЛОЕ32

| IP | IP `..` IP | IP `/` ЦЕЛОЕ32

| ДАТА

| ВРЕМЯ

<l_value>::= ИДЕНТ | ИДЕНТ `*`

<template>::= `+` ИДЕНТ

Терминальные символы

Терминальный символ ***ИДЕНТ*** обозначает идентификатор. Идентификатор состоит из латинских букв, цифр, символов ’_’, ':', '$' и ’–’. Он должен начинаться с латинской буквы или символа ’_’. Запрещено использование идентификаторов, совпадающих с ключевым словом `const`. В качестве типа структуры запрещается указывать идентификатор NULL.

Примеры идентификаторов:

```
Moscow-16
_WWW_
IKECFGRequestAddress
IKERule
```

Терминальный символ ***СТРОКА*** служит для обозначения строки, состоящей из любых символов, заключенных в двойные кавычки (***".."***) . Если внутри строки необходим символ двойной кавычки, то его следует дополнить слева символом '\' . Для использования символа '\' (back-slash)в строке, его нужно ставить два раза подряд ('\\' – двойной back-slash). Допустимо указывать и один back-slash, т.к. при перекодировании восстанавливается двойной back-slash.

Примеры задания значений типа СТРОКА:

```
Title = "Moon Gate LSP"
IntegrityAlg = "MD5-H96-KPDK"
X509SubjectDN *= "C=RU,O=OrgName,OU=qa0,CN=snickers0"
```

Терминальный символ ***ЦЕЛОЕ32*** представляет 32-битное целое число без знака. Число может быть записано в десятичной или шестнадцатеричной системе счисления. Во втором случае оно должно начинаться цифрой и заканчиваться буквой 'h' или 'H'. В шестнадцатеричном и десятичном представлении запись числа не может быть длиннее 10 символов, включая букву 'h'.

Примеры задания числовых значений параметров:

```
RetryTimeBase = 4
BlacklogSessionsMax = 16
LifetimeKilobytes = 0abcdh
```

Терминальный символ ***IP*** обозначает сетевой адрес четвертой версии IP-протокола. IP-адрес состоит из четырех чисел, разделенных точками, где каждое из чисел принадлежит диапазону от 0 до 255.

Пример задания IP-адреса:

```
PeerIPAddress = 192.168.2.1
```

Терминальный символ ***ДАТА***

Тип ***ДАТА*** представляется тремя целыми числами без знака, разделенными символом '/' – число/месяц/год.

Примеры задания даты:

```
StartOfValidity = 24/03/2004
EndOfValidity = 3/6/2004
```

Терминальный символ ***ВРЕМЯ***

Тип ***ВРЕМЯ*** представляется двумя числами, разделенными символом ':'. Время представляется в 24-часовом формате.

Примеры задания времени:

```
23:59 # без минуты полночь
1:1 # час ночи и одна минута
09:2 # 2 минуты десятого утра
01 : 02 # 2 минуты второго ночи
```

Терминальный символ ***DOTDOT*** обозначает две точки подряд, без разделителей ***".."***. Используется для указания диапазона значений.

Комментарии

Комментарии могут размещаться в любом месте текста между другими терминалами и являются разделителями, эквивалентными символам пробела. Вложения комментариев одного типа не допускаются. Поддерживаются следующие два вида комментариев:

Блочный. Начинается с символов "(*" и заканчивается символами "*)" или начинается символом '{' и заканчивается символом '}'.

Строковый. Начинается с символа '#', заканчивается символом перевода каретки <LF>.

Примеры задания комментариев:

```
20..30    # Диапазон чисел 20-30
Action *= (tunnel_IPsec_des_md5_action) (* будет описан ниже *)
```

Разделители

В качестве разделителей в LSP-языке могут быть использованы следующие символы: пробел, табуляция, <LF> и <CR>. Переходом на новую строку считается символ <LF>.

Разделители необходимы только для отделения терминалов ИДЕНТ, ЦЕЛОЕ32, IP, ключевого слова const друг от друга.

Значения полей структур

Значения полей структур (r_value) могут быть простого (базового) типа, например, целое число, текстовая строка, диапазон целых чисел, описанием или ссылкой на описание объекта, списком любых перечисленных значений или пустым списком.

Есть еще один возможный тип значения – процедура. Процедура определяется именем и набором именованных параметров со значениями, заключенными в угловые скобки или именем и списком неименованных параметров, заключенных в квадратные скобки.

Для описания списков могут использоваться круглые или квадратные скобки.

Примеры значений (r_value):

```
20..30                          # Диапазон чисел 20-30.
0.0.0.0..255.255.255.255        # Диапазон IP адресов.
4.3.2.0/24                      # подсеть с 4.3.2.0 с маской 255.255.255.0.
"abed"                          # Текстовая строка.
structure_ref                   # Ссылка на структуру "structure_ref".
[[a,b],[[k,l,m],x,y],4,c,6]     # Вложенные списки из ссылок на структуры
```

```
                                    # (a, b, k, l, m, x, y, c) и чисел (4, 6) .
[]                                  # Пустой список.
proc<x=10 y=24>                     # Процедура "proc" с параметрами x и y.
Filter(SourcePort = 500)            # Объект Filter со значением поля "SourcePort"
                                    # равным 500.
```

Определение объекта

Определение объекта (`object_def`) состоит из типа объекта, имени объекта и списка полей со значениями. Предварительного описания типов внутри языка не существует, описание экземпляра объекта и есть определение типа. Наличие необходимых полей и соответствие значений типу объекта определяется на этапе семантического разбора.

В приведенном ниже примере описан объект типа "Filter" с именем "hostA", который содержит одно поле с именем "DestinationIP" со значением простого типа (IPv4-адрес) равным 23.4.5.6.

Пример:

```
Filter hostA (DestinationIP = 23.4.5.6 )
```

Имя поля

Имя поля (`l_value`) является идентификатором. Значением поля может быть единственное значение или список значений.

Пример:

```
field1 = 1,2,3,4
field2 = 1
field3 = 1
```

В описании одного объекта не может быть двух полей с одинаковыми именами, но если значением поля является список, допускается альтернативный способ задания списка – повторение имени поля несколько раз.

Пример:

```
field* = 1
field* = 2
field* = 3, 4
```

что эквивалентно

```
field = 1,2,3,4
```

Для того чтобы отличить переопределение поля от списка, используется символ '*' после идентификатора. То есть при наличии '*', повторное описание поля будет интерпретировано как добавление элементов в список.

Это же правило действует при добавлении значений из шаблона.

Специальные конструкции

Для упрощения описания повторяющихся параметров предусмотрена возможность использования именованных констант и шаблонов.

В отличие от других конструкций языка, которые подвергаются семантическому анализу, константы и шаблоны полностью обрабатываются на этапе синтаксического разбора.

Описание каждой константы начинается с ключевого слова `const`, за которым следует имя константы и ее значение (или список значений). Значением константы может являться любая конструкция, которая может быть значением поля структуры. Использование константы заключается в подстановке ее имени вместо значения поля структуры.

Пример:

```
const A = 10
const structure = Filter(SourceIP = 1.1.1.1)
const c1 = 1,2,3
const c2 = 4,5,6
```

Описание объектов o1 и o2

```
Filter o1 ( DestinationPort* = c1,c2)
Filter o2 ( DestinationPort* = A )
```

эквивалентно нижеследующему описанию:

```
Filter o1 ( DestinationPort* = 1,2,3,4,5,6)
Filter o2 ( DestinationPort* = 10 )
```

Шаблон (`template`) является константой, единственное значение которой является структурой того типа, к которой этот шаблон будет применен. Для использования шаблона, внутри описания структуры необходимо написать символ '+' и имя константы за ним. Подстановка шаблона заключается в копировании всех полей из структуры, которая является значением константы, в структуру, в которую шаблон подставляется.

Если в структуре, куда подставляется шаблон, присутствует поле, описанное в шаблоне, то возможны следующие варианты:

- в шаблоне и в структуре поле имеет признак списка – *, тогда значения объединяются в единый список, причем порядок составления списков соответствует порядку перечисления полей и шаблонов в структуре
- если признак списка в одном из описаний отсутствует, то будет ошибка разбора.

Пример:

Описание шаблона:

```
const icmp = Filter(ProtocolID* = 1)
```

Пример использования:

```
Filter h_pl ( +icmp DestinationIP = 23.4.4.5 )
Filter icmp_and_tcp ( +icmp ProtocolID* = 6 )
```

Эквивалентные описания:

```
Filter h_pl ( ProtocolID = 1 DestinationIP = 23.4.4.5 )
Filter ping_and_tcp ( ProtocolID = 1,6 )
```

Структура конфигурации

Ниже в таблице представлен состав структур данных с указанием их полей.

Используются следующие обозначения:

- линия напротив поля структуры указывает на описание структуры, используемой в качестве значения;
- '*>' обозначает, что поле содержит список используемых структур;
- '**>' обозначает, что поле содержит список списков используемых структур;
- жирным шрифтом выделены обязательные поля структуры.

Для упрощения простые типы (число, строка, IP-адрес и т.д.) опущены.

GlobalParameters
- Title
- Version
- Type
- PreserveIPsecSA
- AllowNestedIPsec
- CRLHandlingMode
- FirewallLogPacketsThreshold
- FirewallLogTimeTHreshold
- FirewallLogStatesMax

RoutingTable
- **Routes**----------*> Route

FirewallParameters
- TCPEstablishedTimeout
- TCPFinTimeout
- TCPSynSentTimeout
- TCPSynRcvdTimeout
- TCPClosedTimeout
- TCPHalfOpenMax
- TCPHalfOpenLow
- TCPSessionRateMax
- TCPSessionRateLow
- TCPSessionsMax
- TCPStrictnessLevel

IKEParameters
- DefaultPort
- SendRetries
- RetryTimeBase
- RetryTimeMax
- SessionTimeMax
- InitiatorSessionsMax
- ResponderSessionsMax
- BlacklogSessionsMax
- BlacklogSessionsMin
- BlacklogSilentSessions
- BlacklogRelaxTime
- SALifetimeDelta
- FragmentSize
- LocalPort
- NATTLocalPort

SNMPTrapSettings
- Receivers----------*> TrapReceiver

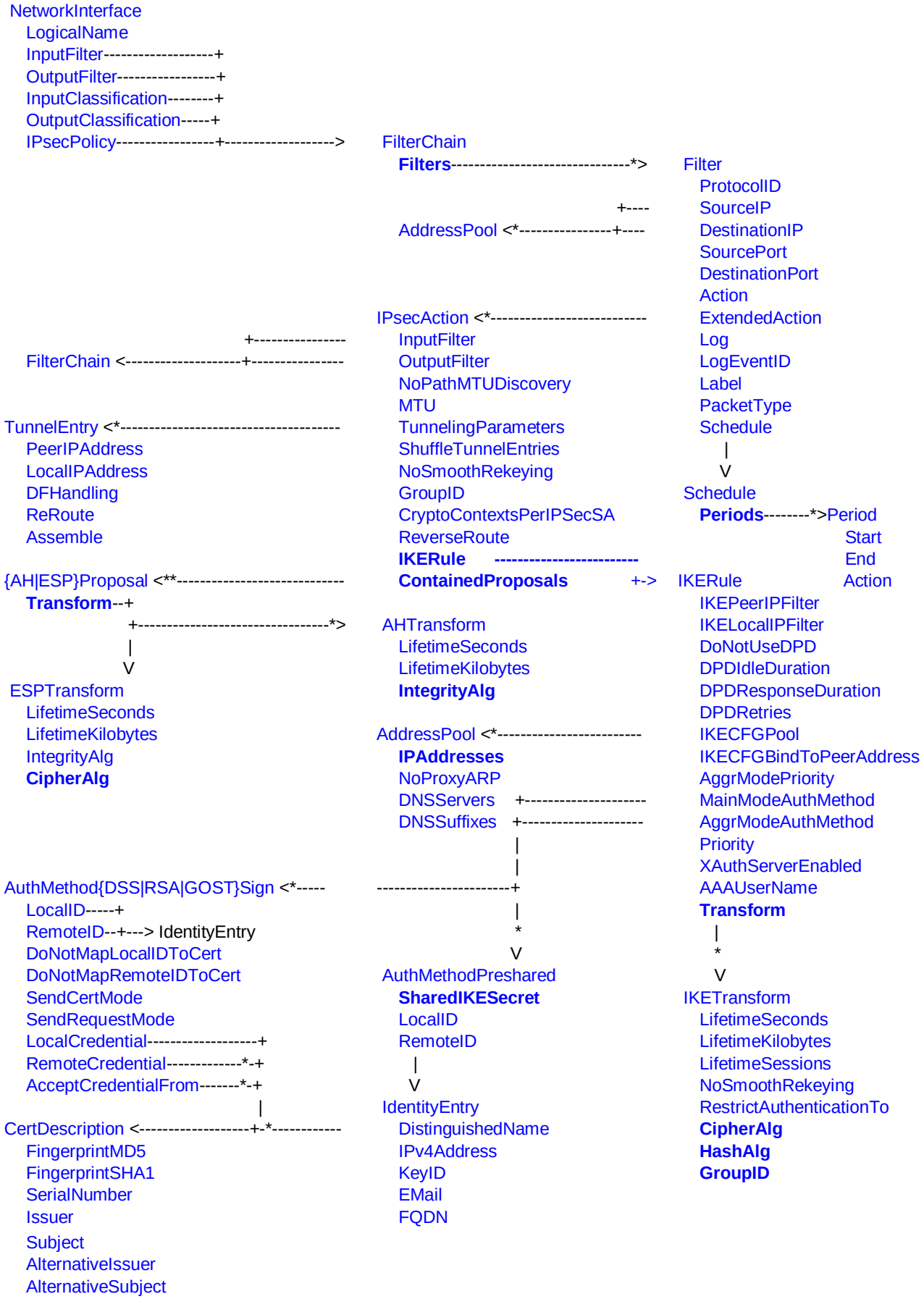
Route
- **Destination**
- Gateway
- NetworkInterface

LDAPSettings
- Server
- Port
- SearchBase
- ConnectTimeout
- ResponseTimeout
- HoldConnectTimeout
- DropConnectTimeout

SNMPPollSettings
- LocalIPAddress
- Port
- **ReadCommunity**
- SysLocation
- SysContact

TrapReceiver
- **IPAddress**
- Port
- **Community**
- Version
- LocalIPAddress

AAASettings
- RadiusServer
- Secret
- NonInteractivePassword
- NoProxyARP
- ResponseTimeout
- Retries



Заголовок конфигурации. Структура GlobalParameters

Заголовок конфигурации представляет собой структуру, описывающую общие параметры для всей политики. В конфигурации должна быть только одна структура данного типа. Имя этой структуре не присваивается.

Имя структуры	GlobalParameters
Атрибуты	Title
	Version
	Type
	CRLHandlingMode
	AllowNestedIPsec
	FirewallLogPacketsThreshold
	FirewallLogTimeThreshold
	FirewallLogStatesMax
	PreserveIPsecSA

Атрибут Title

Атрибут Title предназначен для краткого описания конфигурации (имя конфигурации).

Синтаксис	Title = СТРОКА
Значение	строка произвольного содержания
Значение по умолчанию	пустая строка

Атрибут Version

Атрибут Version определяет версию спецификации конфигурации.

Синтаксис	Version = СТРОКА
Значение	строка вида [0-9].[0-9]
Значение по умолчанию	пустая строка.

Атрибут Type

Атрибут Type специфицирует тип конфигурации, который определяет действия шлюза безопасности при ее активизации.

Синтаксис	Type = PERMANENT | TEMPORARY
Значения	PERMANENT – после успешной активизации конфигурации она сохраняется в базе Продукта, если она была активизирована из файла. При следующем запуске Продукта конфигурация будет автоматически активизирована из базы Продукта.

TEMPORARY – после успешной активизации, конфигурация не сохраняется в базе Продукта и используется только в текущем сеансе работы Продукта.

Значение по умолчанию PERMANENT.

Атрибут CRLHandlingMode

Атрибут CRLHandlingMode определяет режим обработки списка отозванных сертификатов (CRL).

Синтаксис CRLHandlingMode = DISABLE | OPTIONAL | BEST_EFFORT | ENABLE

Значения DISABLE – при проверке сертификата CRL не обрабатывается

OPTIONAL – при проверке сертификата CRL используется только в случае, если он был предустановлен в базу Продукта или получен (и обработан) в процессе IKE обмена и является действующим

BEST_EFFORT – при проверке сертификата CRL используется только в том случае, если он является действующим, если это не так, то CRL может быть получен посредством протокола LDAP (шлюз безопасности смотрит адрес LDAP-сервера сначала в поле CDP сертификата, а затем ищет структуру LDAPSettings). Если CRL получить не удалось – сертификат принимается.

ENABLE – при проверке сертификата обязателен действующий CRL, если это не так, то CRL может быть получен посредством протокола LDAP. Если CRL получить не удалось – сертификат не принимается.

Значение по умолчанию ENABLE.

Атрибут AllowNestedIPsec

Атрибут AllowNestedIPsec позволяет установить дополнительную фильтрацию для IPsec трафика.

Синтаксис AllowNestedIPsec = TRUE | FALSE

Значения TRUE – если входящий или исходящий пакет подпадает под IPsec-правило, для пакета применяется рекурсивный режим поиска правил (ниже поясняется, как это сказывается на обработке входящего и исходящего трафика).

Если AllowNestedIPsec имеет значение TRUE, то исходящий пакет после инкапсуляции подвергается повторному поиску правил IPsec, пока результат поиска не будет простым – PASS или DROP.

Для входящих пакетов AllowNestedIPsec включает симметричные проверки:

Перед декапсуляцией происходит IPsec-фильтрация. Если найдено правило фильтрации, к которому не привязан последний примененный к пакету SA, пакет уничтожается.

Если обрабатывается локальный IPsec-пакет, то он декапсулируется и происходит IPsec-фильтрация.

FALSE – включает упрощенную схему обработки пакетов, которая не предусматривает повторного поиска правил и потенциально работает быстрее.

Значение по умолчанию FALSE

Атрибут FirewallLogPacketsThreshold

Атрибут FirewallLogPacketsThreshold определяет количество пакетов, прошедших через соединение, при достижении которого форсируется вывод статистики по соединению в файл лога. Для сбора статистики по соединению необходимо, чтобы значение артрибута FirewallLogStatesMax не было равно 0.

Синтаксис	FirewallLogPacketsThreshold = ЦЕЛОЕ32
Значения	Целое число из диапазона 1..2147483647
Значение по умолчанию	Если значение не задано, механизм прерывания сбора статистики при превышении заданного количества пакетов выключен. Т.е. теоретически возможна ситуация, когда счетчик для подсчета пакетов будет превышен и накопление начнется снова.
Примечание	Вывод накопленной статистики в файл лога может произойти раньше указанного значения, если истек интервал времени для сбора статистики, заданный атрибутом FirewallLogTimeThreshold. После этого накопление статистики по соединению начнется заново.

Атрибут FirewallLogTimeThreshold

Атрибут FirewallLogTimeThreshold определяет время накопления статистики по текущему соединению. При достижении установленного значения происходит вывод накопленной статистики в файл лога. Для сбора статистики по соединению необходимо, чтобы значение артрибута FirewallLogStatesMax не было равно 0.

Синтаксис	FirewallLogTimeThreshold = ЦЕЛОЕ32
Значения	Целое число из диапазона 1..2147483647
Значение по умолчанию	300 секунд.
Примечание	Вывод накопленной статистики в файл лога может произойти раньше указанного значения, если будет достигнуто допустимое количество пакетов, заданное атрибутом FirewallLogPacketsThreshold. После этого накопление статистики по соединению начнется заново.

Атрибут FirewallLogStatesMax

Подсчет количества пакетов прошедших через каждое соединение происходит в ассоциированном объекте статистики, задаваемым кортежем: SourceIP, SourcePort, DestinationIP, DestinationPort, ProtocolID, LogEventID, Action.

Атрибут FirewallLogStatesMax задает максимальное количество объектов статистики, в которых накапливается информация по соединениям.

Синтаксис	FirewallLogStatesMax = ЦЕЛОЕ32
Значения	Целое число из диапазона 0..10000. Значение 0 говорит о том, что никакая информация накапливаться не будет, т.е. пакеты не обрабатываются, а в файл лога каждую минуту выводится информация о количестве пропущенных пакетов.
Значение по умолчанию	1500.

Атрибут PreserveIPsecSA

Атрибут PreserveIPsecSA позволяет задать сохранение IPsecSA при изменении конфигурационного файла.

Синтаксис PreserveIPsecSA = TRUE | FALSE

Значения TRUE – IPsec SA сохраняется при наличии следующих условий в момент изменения конфигурации (загрузки LSP):

существует список правил фильтрации – FilterChain, привязанный к тому же набору NetworkInterface, что и FilterChain, к которому был привязан IPsecAction, по которому данный SA построен;

в этом FilterChain для селектора SA находится подходящее правило пакетной фильтрации Filter с ExtendedAction = ipsec.

Для IPsec SA, оставшихся от предыдущей конфигурации, не работает заблаговременная смена ключевой информации (Smooth Rekeying) и не происходит уведомление партнера о разрыве соединения (отсылка Delete Payload). Delete Payload, присланные от партнера, обрабатываются корректно.

FALSE – все IPsec SA удаляются при любых изменениях в конфигурации следующих структур: IPsecAction, IKERule, AAASettings, фильтров NetworkInterface.IPsecPolicy, IKEParameters.LocalPort, IKEParameters.NATTLocalPort, GlobalParameters.AllowNestedIPsec, а также структур, на которые перечисленные ссылаются.

Значение по умолчанию FALSE.

Примечание IPsec SA, оставшиеся от старой конфигурации, в той или иной мере могут нарушать новую политику безопасности или быть неработоспособными из-за несоответствия новой LSP. Администратор должен учитывать данную особенность и в сомнительных ситуациях сбрасывать PreserveIPsecSA в FALSE.

Структура LDAPSettings

Структура LDAPSettings задает настройки протокола LDAP, который используется для получения сертификатов и списков отозванных сертификатов (CRL). В конфигурации может присутствовать только одна структура данного типа. Этой структуре имя не присваивается.

В случае отсутствия структуры:

- получение сертификатов посредством протокола LDAP невозможно
- если атрибут CRLHandlingMode структуры GlobalParameters имеет значение ENABLE или BEST_EFFORT, то CRL может быть получен посредством протокола LDAP только при наличии в сертификате, для которого производится проверка подписи, расширения CDP (CRL Distribution Point) с адресом LDAP-сервера.

LDAP-трафик должен быть учтен в правилах фильтрации, т.к. пакеты LDAP фильтруются наравне с остальным трафиком.

Имя структуры	LDAPSettings
Атрибуты	Server
	Port
	SearchBase
	ConnectTimeout
	ResponseTimeout
	HoldConnectTimeout
	DropConnectTimeout

Атрибут Server

Атрибут Server задает адрес LDAP-сервера, к которому производится запрос на поиск сертификатов. Указанный в этом атрибуте адрес не используется, если сертификат, для которого производится проверка подписи, содержит расширение CDP (CRL Distribution Point) с адресом LDAP-сервера.

Сначала делается попытка установить соединение по LDAP версии 2. Если эта попытка завершается с ошибкой LDAP_PROTOCOL_ERROR (наиболее вероятная причина – не поддерживается версия 2), то повторяется попытка установить соединение по LDAP версии 3.

Синтаксис	Server = IP
Значения	IP-адрес
Значение по умолчанию	LDAP-сервер не указан. Поведение шлюза безопасности аналогично случаю отсутствия структуры LDAPSettings в политике.

Атрибут Port

Атрибут Port задает порт LDAP-сервера. Если атрибут Server не задан или расширение сертификата CRL Distribution Point содержит адрес LDAP-сервера, то данный атрибут игнорируется.

Синтаксис	Port = ЦЕЛОЕ32
Значения	Целое число из диапазона 1..65535
Значение по умолчанию	389.

Атрибут SearchBase

Атрибут SearchBase задает имя (Distinguished Name, DN) корневого X.500-объекта, в поддереве которого производится поиск сертификатов и CRL на LDAP-сервере. Указанное имя дополняет запрос, созданный на основе имени из сертификата или CRL, позволяя находить соответствующий X.500-объект в случае, когда исходное имя в запросе является частью имени этого объекта. Для запроса на основе URL данное имя не используется.

Синтаксис SearchBase = СТРОКА

Значения строковое представление DN в соответствии с RFC2253. Относительные имена (Relative Distinguished Name, RDN) указываются в порядке от объекта к корню.

Значение по умолчанию поиск производится по имени, полученному из сертификата или CRL.

Атрибут ConnectTimeout

Атрибут ConnectTimeOut позволяет ограничить время (в секундах) создания TCP-соединения с LDAP-сервером.

Синтаксис ConnectTimeOut = ЦЕЛОЕ32

Значение Целое число из диапазона 1..6000

Значение по умолчанию не устанавливается, что приводит к тому, что время создания TCP-соединения с LDAP-сервером ограничивается установленным для ОС временем создания TCP-соединения.

Примечание Если в момент обращения к LDAP-серверу устройство, на котором он установлен, недоступно, то процесс создания TCP-соединения может занимать продолжительное время (до 3 минут, зависит от ОС). По этой причине могут наблюдаться внешние признаки зависания шлюза безопасности, и это может служить причиной неудачной попытки создания соединения.

Атрибут ResponseTimeout

Поиск посредством протокола LDAP может занимать достаточно продолжительное время, оно зависит от многих факторов, в том числе от масштаба запроса и характеристик канала передачи данных. Данный атрибут позволяет ограничить время (в секундах), в течение которого ожидается ответ от LDAP-сервера на единичный запрос.

Синтаксис ResponseTimeOut = ЦЕЛОЕ32

Значение Целое число из диапазона 2..6000

Значение по умолчанию 200

Атрибут HoldConnectTimeout

Атрибут HoldConnectTimeout устанавливает период времени, в течение которого держится установленное соединение к серверу на случай, если придет к нему повторный запрос.

Синтаксис HoldConnectTimeOut = ЦЕЛОЕ32

Значение Целое число из диапазона 0..6000

При значении 0 после обмена с LDAP-сервером соединение с ним сразу закрывается.

В виду наличия погрешности в одну секунду не рекомендуется выставлять значение в 1 секунду, поскольку это может привести в

некоторых случаях к немедленному закрытию соединения и к избыточному открытию нового соединения.

Значение по умолчанию 60

Атрибут DropConnectTimeout

Атрибут DropConnectTimeout устанавливает период времени, начиная с первой неудачной попытки создания соединения с LDAP-сервером, в течение которого новые попытки создания соединения с ним игнорируются.

Синтаксис DropConnectTimeOut = ЦЕЛОЕ32

Значение Целое число из диапазона 0..6000

При значении 0 в случае неудачной попытки установления соединения с LDAP-сервером новые попытки не игнорируются.

В виду наличия погрешности в одну секунду не рекомендуется выставлять значение в 1 секунду, поскольку это может привести в некоторых случаях к избыточным попыткам создания соединения.

Значение по умолчанию 5.

Пример

Пусть сертификат партнера имеет `Subject = "cn=candy,ou=nomadic"`.

Для поиска такого сертификата на LDAP-сервере (Active Directory – Рисунок 1), необходимо указать атрибут SearchBase:

```
LDAPSettings (
Server = 10.1.1.1
SearchBase="ou=scenario10,ou=QA,ou=GINS,dc=qamsca,dc=ginsoftware,
dc=ru"
)
```

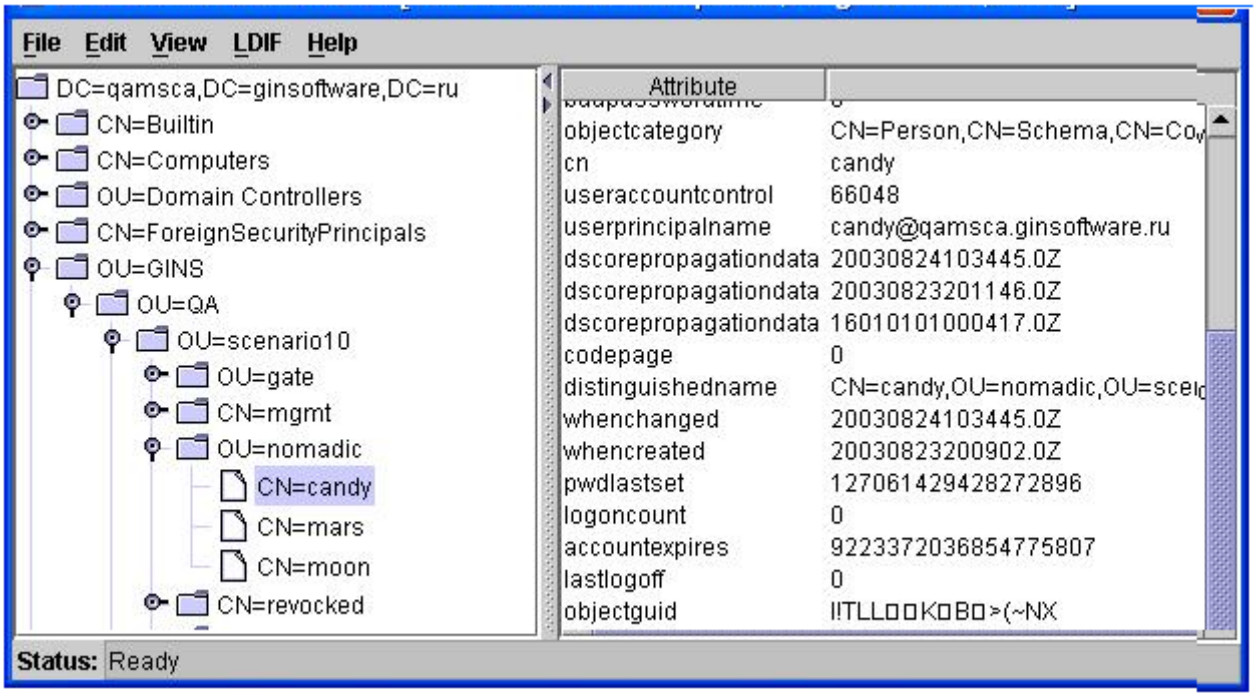


Рисунок 1

Структура IKEParameters

Структура IKEParameters описывает глобальные настройки протокола IKE. В конфигурации должна быть только одна структура данного типа. Этой структуре имя не присваивается.

Имя структуры	IKEParameters
Атрибуты	DefaultPort
	LocalPort
	NATTLocalPort
	SendRetries
	RetryTimeBase
	RetryTimeMax
	SessionTimeMax
	InitiatorSessionsMax
	ResponderSessionsMax
	BlacklogSessionsMax
	BlacklogSessionsMin
	BlacklogSilentSessions
	BlacklogRelaxTime
	SALifetimeDelta
	FragmentSize

Логику используемого механизма IKE-ретрансмиссий смотрите в разделе “Обработка пакетов – ретрансмиссии” Приложения.

Параметры с префиксом Blacklog задают поведение механизма так называемого “черного списка”. "Черный список" предназначен для защиты от DoS-атак (Denial of Service –отказ от обслуживания). "Черный список" минимизирует обработку IKE-пакетов от партнеров, находящихся в "черном списке".

Атрибут DefaultPort

Атрибут DefaultPort устанавливает порт партнера для протокола IKE, который будет использован по умолчанию. Данная настройка не меняет порт, который используется для NAT traversal.

Синтаксис	DefaultPort = ЦЕЛОЕ32
Значения	Целое число из диапазона 1..65535
Значение по умолчанию	500.

Атрибут LocalPort

Атрибут LocalPort устанавливает локальный порт, используемый протоколом IKE.

Синтаксис	LocalPort = ЦЕЛОЕ32

Значения Целое число из диапазона 1..65535.
Если укзано значение 0, выбирается свободный порт по алгоритму, реализованному в операционной системе.

Значение по умолчанию 500.

Атрибут NATTLocalPort

Атрибут NATTLocalPort устанавливает локальный порт для NAT Traversal, используемый протоколами IKE и IPsec.

Синтаксис NATTLocalPort = ЦЕЛОЕ32

Значения Целое число из диапазона 1..65535.
Если указано значение 0, выбирается свободный порт по алгоритму, реализованному в операционной системе.

Значение по умолчанию 4500.

Атрибут SendRetries

Атрибут SendRetries устанавливает число попыток посылки IKE-пакетов партнеру. Интервал между повторными посылками того же IKE-пакета с каждой новой попыткой увеличивается вдвое, начиная со значения параметра RetryTimeBase, но не может превышать значения атрибута RetryTimeMax.

Синтаксис SendRetries = ЦЕЛОЕ32

Значения Целое число из диапазона 1..30

Значение по умолчанию 5.

Атрибут RetryTimeBase

Атрибут RetryTimeBase позволяет установить начальный интервал в секундах между повторными попытками посылки IKE-пакетов партнеру. Если ответ не получен в течение начального интервала, то запрос посылается повторно и интервал между повторными попытками увеличивается в два раза. Этот интервал увеличивается в два раза до тех пор, пока:

- не будет получен ответ
- или значение интервала RetryTimeBase не достигнет значения RetryTimeMax (повторные попытки будут продолжаться с интервалом RetryTimeMax)
- или количество попыток не достигнет значения SendRetries.

Синтаксис RetryTimeBase = ЦЕЛОЕ32

Значения Целое число из диапазона 1..5

Значение по умолчанию 1.

Атрибут RetryTimeMax

Атрибут RetryTimeMax позволяет установить максимальный интервал в секундах между повторными попытками посылки IKE-пакетов партнеру. Если выставленное значение этого атрибута меньше, чем RetryTimeBase, то при загрузке конфигурации атрибуту RetryTimeMax присваивается значение RetryTimeBase.

Синтаксис RetryTimeMax = ЦЕЛОЕ32

Значения Целое число из диапазона 1..60

Значение по умолчанию 30.

Атрибут SessionTimeMax

Атрибут SessionTimeMax ограничивает время (в секундах) на каждую сессию IKE.

Синтаксис SessionTimeMax = ЦЕЛОЕ32

Значения Целое число из диапазона 10..300

Значение по умолчанию 60.

Атрибут InitiatorSessionsMax

Атрибут InitiatorSessionsMax устанавливает максимально допустимое количество одновременно инициируемых IKE-сессий для всех партнёров.
Если при достижении установленного порога локальное устройство требует инициации очередной IKE-сессии, то она откладывается в очередь ожидания. Отложенная IKE-сессия инициируются заново при завершении ранее активной IKE-сессии. Размер очереди ожидания не ограничен.

Синтаксис InitiatorSessionsMax = ЦЕЛОЕ32

Значение число из диапазона 1–10000

Значение по умолчанию 30.

Атрибут ResponderSessionsMax

Атрибут ResponderSessionsMax определяет максимально допустимое количество одновременных обменов, проводимых VPN-устройством со всеми партнерами в качестве ответчика.
Если локальное устройство имеет указанное количество незавершенных IKE-обменов в роли ответчика, то все входящие ISAKMP-пакеты, требующие установления новых обменов, игнорируются (без оповещения партнера).

Синтаксис ResponderSessionsMax = ЦЕЛОЕ32

Значения Целое число из диапазона 1–10000

Значение по умолчанию 20.

Атрибут BlacklogSessionsMax

BlacklogSessionsMax устанавливает.начальное число разрешенных одновременных IKE обменов, инициируемых одним партнером[1]. При каждом неудачном завершении IKE обмена,

[1]В данном случае партнер идентифицируется по паре ip:port. Пока партнер не аутентифицирован (т.е. с таким партнером на данный момент нет ни одного ISAKMP-соединения – SA), допустимое количество IKE-

число разрешенных одновременных IKE обменов для данного партнера снижается вдвое с округлением в меньшую сторону, вплоть до значения, устанавливаемого параметром BlacklogSessionsMin.

Примечание как только партнер заносится в "черный список", для него текущее значение разрешенных одновременно проводимых IKE обменов не только начинает уменьшаться в два раза после каждого неуспешного завершения обмена, но и увеличиваться на единицу по истечении каждого интервала времени BlacklogRelaxTime (описанного далее).

Синтаксис BlacklogSessionsMax = ЦЕЛОЕ32

Значения Целое число из диапазона $0..2^{32}-1$

Если значение равно 0, то "черный список" не используется.

Если значение BlacklogSessionsMax больше или равно ResponderSessionsMax, то атрибуту BlacklogSessionsMax присваивается значение ResponderSessionsMax.

Значение по умолчанию 16.

Атрибут BlacklogSessionsMin

Атрибут BlacklogSessionsMin позволяет установить минимальное число разрешенных одновременных IKE обменов, инициируемых неаутентифицированным партнером.

Синтаксис BlacklogSessionsMin = ЦЕЛОЕ32

Значения Целое число из диапазона $0..2^{32}-1$

Если значение равно 0, то для партнера, поведение которого привело к понижению числа разрешенных инициируемых им одновременных IKE обменов до значения BlacklogSessionsMin, игнорируется весь IKE-трафик, а все имеющиеся с ним недостроенные IKE-сессии уничтожаются (ситуация “Access denied”).

Если значение равно, либо превышает BlacklogSessionsMax, то число разрешенных одновременных IKE обменов, инициируемых неаутентифицированным партнером, не снижается (т.е. ”черный список“ отключен) [2].

Значение по умолчанию 0.

Атрибут BlacklogSilentSessions

Атрибут BlacklogSilentSessions позволяет установить число активных обменов, инициированных неаутентифицированным партнером, по достижении которого VPN-устройство перестает информировать партнера о причине неуспешного завершения инициированного им IKE-обмена.

обменов может снижаться в зависимости от того, насколько успешно завершаются IKE-обмены с этим партнером.

[2] При загрузке конфигурации с *отключенным* «черным списком» вся статистическая информация о «плохих» партнерах сбрасывается. Если же «черный список» *включен*, то к уже имеющейся накопленной статистике применяются новые параметры настроек «черного списка».

Синтаксис	BlacklogSilentSessions = ЦЕЛОЕ32
Значения	Целое число из диапазона $0..2^{32}-1$ Если это значение больше, чем BlacklogSessionsMax, то инициатор не ограничивается в таких оповещениях. Если значение равно 0 либо 1, то неаутентифицированный партнер никогда не оповещается о причинах ошибки инициированного им обмена.
Значение по умолчанию	4.

Атрибут BlacklogRelaxTime

Атрибут BlacklogRelaxTime устанавливает интервал времени (в секундах) релаксации "черного списка".

- За указанный период времени число разрешенных одновременных IKE обменов для каждого партнера, находящегося в "черном списке", увеличивается на единицу. По истечении следующего такого же интервала времени, текущие значения разрешенных одновременно проводимых IKE обменов для каждого партнера опять увеличивается на единицу и т.д. Этот интервал времени отсчитывается с момента последней загрузки конфигурации.
- Как только текущее значение разрешенных одновременно проводимых партнером IKE обменов начинает превышать значение BlacklogSessionsMax, такой партнер исключается из "черного списка".

Синтаксис	BlacklogRelaxTime = ЦЕЛОЕ32
Значения	Целое число из диапазона $0..2^{32}-1$ 0 – бесконечное время (партнер попадает в "черный список" навсегда).
Значение по умолчанию	120
Примечание	помимо механизма релаксации, партнер также может быть иключен из "черного списка" в следующих случаях: при перезапуске сервиса при загрузке конфигурации с отключенным "черным списком" (с атрибутом BlacklogSessionsMax = 0) при инициации IKE обмена со стороны локального VPN устройства с целью установления ISAKMP (IPsec) соединения[3] если партнеру удалось установить ISAKMP (IPsec) соединение с локальным VPN устройством, и тем самым партнер был успешно аутентифицирован.

[3] В данном случае считается, что локальное VPN устройство потенциально доверяет партнеру, с которым оно хочет установить соединение, и информация, накопленная в "черном списке", для такого партнера сбрасывается.

Атрибут SALifetimeDelta

Атрибут SALifetimeDelta позволяет установить случайный разброс во времени жизни IKE и IPsec SA. Этот атрибут рекомендуется использовать в случае массового пересоздания SA.

Синтаксис SALifetimeDelta = ЦЕЛОЕ32

Значения Целое число из диапазона 0..50

Значение – максимальный процент, на который может быть уменьшено время действия SA (LifetimeSeconds). Реальное значение определяется случайным образом от 0 до этого максимума.

Значение по умолчанию 0 – отключает механизм случайного изменения времени жизни IKE и IPsec SA.

Примечание Для респондера значение SALifetimeDelta будет использовано только при условии, если инициатор предлагает значение большее или равное локальному параметру LifetimeSeconds.

Если локальное значение IKETransform.LifetimeSeconds равно 0, то для данного правила SALifetimeDelta не используется.

Атрибут FragmentSize

Атрибут FragmentSize управляет функциональностью фрагментирования IKE-пакетов. Этот атрибут рекомендуется использовать в случае массового пересоздания SA.

Синтаксис FragmentSize = ЦЕЛОЕ32

Значения Целое число из диапазона 0..65535

Значение – максимальный размер результирующего IP-пакета[4] в байтах.
Значение 0 отключает функциональность фрагментирования IKE-пакетов. Партнёр о поддержке фрагментирования IKE-пакетов не оповещается, отсылаемые IKE-пакеты не фрагментируются, принимаемые фрагменты не собираются.
Ненулевое заданное значение может корректироваться в большую сторону таким образом, чтобы максимально возможный ISAKMP-пакет длиной 64Kb мог быть разбит на 255 фрагментов[5].

Значение по умолчанию 576.

[4] Следует учитывать, что операционная система сама устанавливает длину ip-заголовка, что может приводить к фактическому уменьшению длины ip-пакета с IKE-фрагментом на величину до 44 байт (максимально допустимый размер ip-заголовка – 64 байта, наиболее часто используемый – 20 байт).

[5] Это означает, что минимальная длина UDP-пакета с IKE-фрагментом не может быть менее 304 байт.

Структура SNMPPollSettings

Структура задает настройки для выдачи информации SNMP-агентом по запросу SNMP-менеджера. В конфигурации должна быть только одна структура данного типа. Этой структуре имя не присваивается.

Имя структуры	SNMPPollSettings
Атрибуты	LocalIPAddress
	Port
	ReadCommunity
	SysLocation
	SysContact

Атрибут LocalIPAddress

Атрибут LocalIPAddress задаёт список локальных IPv4-адрес, на которые можно получать запросы от SNMP-менеджера. Указание IP-адреса 0.0.0.0 эквивалентно указанию константы ANY.

Синтаксис	LocalIPAddress = IP | ANY
Значения	IP – список локальных IP-адресов
	ANY – все локальные IP-адреса
Значение по умолчанию	ANY

Атрибут Port

Атрибут Port задаёт порт, на который можно получать SNMP-запросы.

Синтаксис	Port = ЦЕЛОЕ32
Значение	целое число из диапазона 1..65535
Значение по умолчанию	161.

Атрибут ReadCommunity

Атрибут ReadCommunity играет роль пароля при аутентификации сообщений SNMP и разрешает SNMP-менеджеру чтение статистики из базы управления SNMP-агента.

Синтаксис	ReadCommunity = СТРОКА
Значение	произвольный формат
Значение по умолчанию	не существует, атрибут обязательный.

Атрибут SysLocation

Атрибут SysLocation содержит информацию о физическом расположении SNMP-агента.

Синтаксис	SysLocation = СТРОКА
Значение	произвольный формат, например "Building 3/Room 214"
Значение по умолчанию	пустая строка.

Атрибут SysContact

Атрибут SysContact содержит информацию о контактном лице, ответственном за работу SNMP-агента.

Синтаксис	SysContact = СТРОКА
Значение	произвольный формат, например e-mail, телефон и т.д.
Значение по умолчанию	пустая строка.

Структура SNMPTrapSettings

Структура задает настройки для выдачи агентом сообщений менеджеру о возникших событиях в виде SNMP-трапов. В конфигурации должна быть только одна структура данного типа. Этой структуре имя не присваивается. При отсутствии этой структуры, трап-сообщения не высылаются.

Имя структуры	SNMPTrapSettings
Атрибуты	Receivers

Атрибут Receivers

Атрибут Receivers задаёт список получателей SNMP-трапов и дополнительные настройки.

Синтаксис	Receivers* = `TrapReceiver`
Значение по умолчанию	не существует, атрибут обязательный.

Структура TrapReceiver

Структура TrapReceiver описывает одного получателя SNMP-трапов, который добавляется в текущий список получателей SNMP-трапов, и дополнительные настройки для трапов, отсылаемых ему.

Имя структуры	TrapReceiver
Атрибуты	IPAddress
	Port
	Community
	Version
	LocalIPAddress

Атрибут IPAddress

Атрибут IPAddress описывает IP-адрес получателя SNMP-трапов.

Синтаксис	IPAddress = IP
Значение	IP-адрес
Значение по умолчанию	не существует, атрибут обязательный.

Атрибут Port

Атрибут Port задает UDP-порт, на который SNMP-менеджеру будут высылаться трап-сообщения.

Синтаксис	Port = ЦЕЛОЕ32
Значение	целое число из диапазона 1..65535.
Значение по умолчанию	162.

Атрибут Community

Атрибут Community играет роль идентификатора отправителя трап-сообщения.

Синтаксис	Community = СТРОКА
Значение	произвольный формат
Значение по умолчанию	не существует, атрибут обязательный.

Атрибут Version

Атрибут Version указывает версию SNMP, в которой формируются трап-сообщения.

Синтаксис	Version = V1 | V2C
Значение	V1 – SNMP версии 1 V2C – SNMP версии 2c
Значение по умолчанию	V1.

Атрибут LocalIPAddress

Атрибут LocalIPAddress задает IP-адрес, с которого будут отправляться трап-сообщения. Можно вместо IP-адреса указать имя сетевого интерфейса.

Синтаксис	LocalIPAddress = IP | СТРОКА
Значение	Имя сетевого интерфейса. Имя должно совпадать с одним из имен NetworkInterface – NetworkInterface.LogicalName. Если указанному NetworkInterface соответствует несколько сетевых интерфейсов или адресов, то будет использован один адрес[6] IP-адрес. Если указано значение 0.0.0.0, адрес будет выбирать ОС в зависимости от адреса назначения.
Значение по умолчанию	0.0.0.0.

[6] Первый адрес первого подходящего интерфейса в соответствии с порядком выдачи интерфейсов и адресов библиотекой ni.

Структура RoutingTable

Структура RoutingTable задает маршруты, которые добавляются в системную таблицу маршрутизации.

При отгрузке конфигурации маршруты из системной таблицы маршрутизации будут удалены.

Если LSP создается автоматически (при задании политики через cs_console), то использовать структуру RoutingTable для управления маршрутами запрещено.

Предполагается, что пользователь не создает и не удаляет маршруты с теми же адресами назначения (Destination), что указаны в LSP. Совпадение маршрута по адресу назначения (Destination) с тем, что добавляет подсистема RRI, также может привести к ошибкам при создании или удалении маршрутов.

Если при добавлении маршрута в системную таблицу возникает ошибка, тем не менее, загрузка LSP продолжается, а соответствующее предупреждение передается через систему протоколирования.

В конфигурации допускается только один экземпляр этой структуры. Имя этой структуре не присваивается.

Имя структуры RoutingTable

Атрибуты Routes

Атрибут Routes

Атрибут Routes содержит список записей для добавления в таблицу маршрутизации.

Синтаксис Routes* = `Route`

Значение по умолчанию не существует, атрибут обязательный.

Структура Route

Структура Route описывает одну запись (маршрут) в таблице маршрутизации.

Имя структуры	Route
Атрибуты	Destination
	Gateway
	NetworkInterface

Атрибут Destination

Атрибут Destination задает адрес назначения (получателя) пакета.

Синтаксис	Destination = IP | IP/ЦЕЛОЕ32
Значение	IP-адрес IP/ЦЕЛОЕ32 – IP-адрес с маской подсети Для указания маршрута, который будет использоваться по умолчанию, IP-адрес и маска подсети должны иметь значение 0.0.0.0/0. Маршрут по умолчанию – маршрут, по которому будет отправлен пакет, если IP-адрес назначения, указанный в заголовке пакета, не совпадает ни с одним адресом назначения в таблице маршрутизации.
Значение по умолчанию	отсутствует, атрибут обязательный.

Атрибут Gateway

Атрибут Gateway задает IP-адрес устройства, на который нужно передать пакет для продвижения его к получателю пакета. Атрибут Gateway не может быть указан при наличии атрибута NetworkInterface.

Синтаксис	Gateway = IP
Значение	IP –адрес
Значение по умолчанию	используется значение из атрибута NetworkInterface.

Атрибут NetworkInterface

Атрибут NetworkInterface указывает имя выходного интерфейса шлюза безопасности (из файла ifaliases.cf), на который нужно передать пакет для продвижения его к получателю пакета. Использования шаблонов и списков значений для данного интерфейса в ifaliases.cf не допускается. Атрибут NetworkInterface не может быть указан при наличии атрибута Gateway.

Синтаксис	NetworkInterface = СТРОКА
Значение	имя интерфейса
Значение по умолчанию	используется значение из атрибута Gateway.

Структура AddressPool

Структура AddressPool задает множество адресов IKECFG-пула и связанные с ним свойства.

Имя структуры	AddressPool
Атрибуты	IPAddresses
	NoProxyARP
	DNSServers
	DNSSuffixes

Атрибут IPAddresses

Атрибут IPAddresses задает множество адресов в виде списка, состоящего из одиночных адресов, диапазонов адресов и подсетей. Адреса не должны совпадать или пересекаться между собой.

Синтаксис	IPAddresses* = IP, IP..IP, IP/ЦЕЛОЕ32
Значения	IP-адрес
	IP..IP – диапазон IP-адресов
	IP/ЦЕЛОЕ32 – IP-адрес с маской подсети
Значение по умолчанию	атрибут обязательный.

Атрибут NoProxyARP

Атрибут NoProxyARP задает режим работы устройства в роли ProxyARP для указанного множества адресов. Режим проксирования имеет смысл использовать, когда указанный пул адресов является подмножеством адресов защищаемой подсети шлюзом безопасности.

Выданный по IKECFG внешнему устройству IP-адрес должен проксироваться с интерфейса защищаемой подсети, чтобы пакеты от устройств этой подсети, предназначенные для исходного внешнего устройства, попадали на шлюз безопасности для их дальнейшей обработки и пересылки внешнему устройству, для этого также соответствующим образом должна быть задана таблица маршрутизации (см. Пример).

Добавление proxy-arp записей не гарантирует маршрутизацию пакетов на самом шлюзе безопасности. Для автоматического добавления маршрутов, proxy-arp можно использовать совместно с механизмом RRI (подробнее см. документ «Настройка шлюза», раздел «Общие настройки шлюза»).

При удалении SA, соответствующие proxy-arp записи удаляются.

Синтаксис	NoProxyARP = FALSE | TRUE
Значение	FALSE – для указанного множества адресов устройство выступает в роли ProxyARP. Если IP-адрес не попадает ни в одну из защищаемых локальных подсетей, proxy-arp запись не создается, и это не считается ошибкой.
	TRUE – адреса не проксируются.
Значение по умолчанию	FALSE

Пример

Описан случай, когда NoProxyARP обязан быть выставлен в FALSE (в иных случаях – это необязательно).

Топология сети:

```
------- 10.0.0.1/24 GW ====== ISP router === ... === nomadic
```

Обозначения:

`--------`	открытый трафик
`========`	защищенный трафик
`GW`	шлюз безопасности
`ISP router`	маршрутизатор провайдера
`.........`	все промежуточные хосты
`nomadic`	внешний пользователь.

Пул адресов выделен из внутренней сети, например, 10.0.0.240 – 10.0.0.247, его так же можно задать в форме 10.0.0.240/29. Здесь специально выбран диапазон, который укладывается в подсеть, чтобы удобнее было задавать запись в таблице маршрутизации. Но не стоит путать - адреса 10.0.0.240 и 10.0.0.247 не будут являться спец. адресами подсети.

Для указанной топологии в LSP необходимо указать:

```
AddressPool(
    IPAddresses = 10.0.0.240/29
    NoProxyARP = FALSE
)
Route(
    Destination = 10.0.0.240/29
    Gateway = ISP_router_IP_address
)
```

В результате получим:

- GW ответит на ARP-запрос Ethernet адреса для IP-адреса из пула от хоста из внутренней сети
- После попадания пакета на внутренний интерфейс GW с адресом назначения из пула, пакет будет перенаправлен в соответствии с указанной записью в маршрутной таблице на внешний интерфейс GW, где перед отправкой во вне он будет зашифрован.

Необходимо помнить:

Нельзя указывать маршрутизацию для 10.0.0.240/29 через внешний интерфейс GW, так как выделяемые адреса из пула будут привязываться в ARP-таблице к внешнему интерфейсу, и GW не будет отвечать на ARP-запросы для таких адресов с внутреннего интерфейса. Это расходится с практикой Cisco, где в таком случае запись делается через интерфейс.

Атрибут DNSServers

Атрибут DNSServers задает список адресов серверов DNS, передаваемых по протоколу IKECFG клиенту.

В случае отсутствия данного атрибута, адреса серверов клиенту не передаются.

Синтаксис	DNSServers* = IP
Значения	список IP-адресов DNS серверов. Допускаются любые значения адресов, кроме нулевого (0.0.0.0).
Значение по умолчанию	значение по умолчанию отсутствует.

Атрибут DNSSuffixes

Атрибут DNSSuffixes задает список суффиксов DNS, передаваемых по протоколу IKECFG клиенту. В случае отсутствия данного поля, суффиксы DNS клиенту не передаются.

Проверка на корректность значений при загрузке LSP и при отсылке по протоколу IKECFG не производится[7].

Синтаксис	DNSSuffixes* = СТРОКА
Значения	список суффиксов DNS
Значение по умолчанию	значение по умолчанию отсутствует.

Пример

```
AddressPool pool_dyn
(
   IPAddresses = 192.168.2.240..192.168.2.247
   DNSServers = 192.168.10.10
   DNSSuffixes = "s-terra"
)
```

[7] Корректность значений зависит от их применимости на операционной системе Клиента, о чем должен позаботиться администратор Шлюза безопасности.

Структура IPsecAction

Структура IPsecAction задает правило создания контекста соединения для протоколов семейства IPsec. В конфигурации таких структур может быть несколько. Этой структуре может быть присвоено имя.

Имя структуры	IPsecAction
Атрибуты	TunnelingParameters
	ShuffleTunnelEntries
	CryptoContextsPerIPSecSA
	GroupID
	ContainedProposals
	IKERule
	NoPathMTUDiscovery
	MTU
	NoSmoothRekeying
	ReverseRoute
	InputFilter
	OutputFilter

Атрибут TunnelingParameters

Атрибут TunnelingParameters описывает параметры внешнего IP-заголовка пакета, который добавляется в туннельном режиме IPsec. Если в TunnelingParameters указано более одного элемента, то элементы используются как альтернативные партнеры. Если не удалось установить IPsec-туннель с партнером, то производится попытка установить туннель со следующим партнером в списке, и так далее до окончания списка.

Синтаксис	TunnelingParameters* = `TunnelEntry`
Значение по умолчанию	используется транспортный режим.
Предупреждение:	если между партнерами обнаружен NAT, то создавать соединение в транспортном режиме нельзя.

Атрибут ShuffleTunnelEntries

Атрибут ShuffleTunnelEntries задает порядок применения структур TunnelEntry в атрибуте TunnelingParameters. Атрибут ShuffleTunnelEntries игнорируется, если атрибут TunnelingParameters не задан.

Синтаксис	ShuffleTunnelEntries = TRUE | FALSE
Значения	TRUE – при загрузке конфигурации туннели в списке TunnelingParameters перемешиваются случайным образом
	FALSE – при загрузке конфигурации туннели в списке TunnelingParameters применяются в порядке перечисления
Значение по умолчанию	FALSE

Атрибут CryptoContextsPerIPSecSA

Атрибут CryptoContextsPerIPSecSA задает количество открываемых криптографических контекстов на один IPsec SA, созданный по этому правилу IPsecAction. Наличие нескольких криптографических контекстов позволяет распараллелить обработку пакетов одним IPsec SA.

Значение CryptoContextsPerIPSecSA, превышающее общее число процессорных ядер[8] на 1 и более, включает оптимизированный алгоритм создания криптографических контекстов. Оптимизация заключается в привязке крипто-контекстов к ядрам процессора и может существенно повышать производительность IPsec на многопроцессорных[9] системах. Для однопроцессорных систем выставлять CryptoContextsPerIPSecSA не имеет смысла.

Синтаксис CryptoContextsPerIPSecSA = ЦЕЛОЕ32

Значения Целое число из диапазона 1..128.

Значение по умолчанию берется из файла agent.ini (параметр DefaultCryptoContextsPerIPSecSA).

Атрибут IKERule

Атрибут IKERule является ссылкой на правило создания контекста соединения для ISAKMP-инициатора.

Синтаксис IKERule = `IKERule`

Значение по умолчанию не существует, атрибут обязательный.

Атрибут GroupID

Атрибут GroupID задает параметры получения ключевого материала. Используется алгоритм Диффи-Хеллмана. Параметры задаются в виде списка. Если список не пуст, то для инициатора соединения ключевой материал всегда задаётся согласно первому компоненту списка. Для ответчика присланное предложение инициатора сравнивается последовательно со всеми элементами своего списка.

Синтаксис GroupID = MODP_768, MODP_1024, MODP_1536, BELTDH,NO_PFS

Значения MODP_768 – группа 1 – длина ключа 768 бит

MODP_1024 – группа 2 – длина ключа 1024 бита

MODP_1536 – группа 5 – длина ключа 1536 бит

BELTDH – протокол формирования общего ключа на основе эллиптических кривых согласно СТБ 34.101.66-2014

NO_PFS – обмен ключами во второй фазе IKE не используется/

[8] Данное значение гарантирует включение оптимизации, но она включается также начиная со значения, превышающего максимальный номер процессора, к которому привязана нитка обработчика в драйвере. То есть для гарантированного выключения оптимизации необходимо выставить число равное количеству ниток-обработчиков, задаваемому параметром драйвера "cpu_distribution".

[9] Подразумевается множество процессоров, видимых для операционной системы, независимо от топологии системы. Это могут быть логические процессоры (технологии hyperthreading и аналогичные), процессорные ядра, расположенные на одном кристалле.

Значение по умолчанию ключевой материал заимствуется из первой фазы IKE.

Атрибут ContainedProposals

Каждая из структур AHProposal и ESPProposal содержит список вариантов преобразований (transforms). Структуры AHProposal и ESPProposal могут группироваться, позволяя обрабатывать трафик комбинацией протоколов AH и ESP.

Атрибут ContainedProposals содержит список единичных структур AHProposal и ESPProposal или их пар в порядке убывания приоритета.

Синтаксис ContainedProposals *= Proposal

Proposal *= (`AHProposal [,ESPProposal]`) | `ESPProposal`

Значения Число элементов списка неограничено. Все элементы списка должны быть различными.

Один элемент списка содержит до двух преобразований с различными протоколами.

Если элемент списка содержит AHProposal и ESPProposal, то они должны следовать в указанном порядке.

Инициатор соединения посылает партнеру все варианты параметров защиты соединения, указанные в атрибуте ContainedProposals, с целью их согласования во время второй фазы IKE-сессии.

Ответная сторона присланные предложения инициатора соединения последовательно сравнивает с каждым элементом своего списка предложений и выбирает первое совпавшее. При переборе более приоритетным является список на стороне ответчика.

Параметры преобразований и комбинация протоколов AH и ESP определяют качество защиты соединения.

Запись `(ah1, esp1),(esp2),(ah3)` означает, что рассматриваются варианты контекстов:
либо связка `(ah1, esp1)`, либо proposal `esp2`, либо proposal `ah3`.

Значение по умолчанию не существует, атрибут обязательный.

Пример

```
ContainedProposals *=
     (ipsec_ah_md5, ipsec_esp_des3),(ipsec_ah_md5, ipsec_esp_idea)
     (* (AH(MD5) и ESP(DES3) или AH(MD5) и ESP(IDEA) *)

ContainedProposals *=
     (ipsec_ah_md5, ipsec_esp_des3),(ipsec_ah_md5)
     (* (AH(MD5) и ESP(DES3) или  AH(MD5) *)

ESPProposal ipsec_esp_idea(
     Transform *= ESPTransform(
     CipherAlg = "IDEA-CBC")
)
AHProposal ipsec_ah_md5(
     Transform *= AHTransform(
```

```
            IntegrityAlg* = "MD5-H96-HMAC")
    )
    ESPProposal ipsec_esp_des3(
        Transform *= ESPTransform(
            CipherAlg = "DES3-K168-CBC")
    )
```

Атрибут NoPathMTUDiscovery

Этот атрибут отключает алгоритм "Path MTU Discovery" (выявление максимального размера пакета, проходящего на всем пути от отправителя к получателю без фрагментации) для IPsec SA, создаваемых по данному правилу.

Синтаксис NoPathMTUDiscovery = TRUE | FALSE

Значения FALSE – производится обработка ICMP-сообщений типа destination unreachable/fragmentation needed, приходящих в ответ на IPsec-пакеты. На основе этих сообщений вычисляется эффективное значение MTU трассы (максимальный размер пакета, проходящий по всему каналу без фрагментации).

TRUE – ICMP-сообщения не обрабатываются, значение MTU вычисляется только из локальной конфигурации.

Значение по умолчанию FALSE

Атрибут MTU

Этот атрибут задает значение MTU для IPsec SA, создаваемых по данному правилу.

Значение MTU используется только для исходящих пакетов и для последнего SA, примененного к пакету (в случае вложенного IPsec значение MTU для внутреннего SA игнорируется).

Синтаксис: MTU = ЦЕЛОЕ32

Значения Целое число из диапазона 1..65535

Значение по умолчанию 0, это означает, что MTU определяется автоматически.

Значение MTU интерпретируется следующим образом:

- если пакет подвергается повторной маршрутизации (TunnelEntry.ReRoute = TRUE или пакет отправляется с IKECFG-интерфейса) и:
  - если в пакете выставлен DF-бит, то значение MTU интерфейса не учитывается, а значение IPsecAction.MTU рассматривается как значение MTU интерфейса
  - если в пакете DF-бит сброшен, то IPsecAction.MTU не используется
- если пакет отправляется без повторной маршрутизации, то выбирается минимальное из значений – MTU интерфейса и MTU в IPsecAction.

Если NoPathMTUDiscovery=FALSE, то указанное IPsecAction.MTU может быть скорректировано в меньшую сторону при вычислении MTU трассы.

Рекомендуется устанавливать значение MTU не менее 670 байт.

Атрибут NoSmoothRekeying

Атрибут NoSmoothRekeying задает режим "мягкой" смены ключевого материала.

Синтаксис NoSmoothRekeying = TRUE | FALSE

Значения TRUE – заблаговременная смена ключевого материала (rekeying) не проводится. При отсутствии подходящего IPsec соединения, новый IPsec SA создаётся только по запросу из ядра – при наличии исходящего IP-пакета, либо по инициативе партнера. В результате, во время создания нового IPsec SA IP-трафик приостанавливается, а при интенсивном трафике возможна потеря пакетов.

FALSE – заблаговременно, незадолго до окончания действия IPsec соединения, на его основе (с теми же параметрами) проводится IKE-сессия (Quick Mode) по созданию нового IPsec SA – rekeying. Rekeying не проводится, если за время существования старого SA под его защитой не было никакого трафика.[10]

Значение по умолчанию FALSE

Атрибут ReverseRoute

Атрибут ReverseRoute задает функциональность Reverse Route Injection (RRI). После установления защищенного соединения с удаленным партнером, при включенном механизме RRI в системную таблицу маршрутизации добавляется запись об обратном маршруте (подробнее см. документ «Настройка шлюза», раздел «Общие настройки шлюза»).

Синтаксис ReverseRoute = TRUE | FALSE

Значения TRUE – RRI включен

FALSE – RRI выключен

Значение по умолчанию FALSE

Для транспортного режима и для туннельного режима HOST<->HOST, где туннельный destination совпадает с destination из внутреннего заголовка, RRI допускается, но смысла не имеет.

Фильтры, к которым привязано правило с включенным RRI, не должны содержать портов, протоколов и диапазонов адресов в destination-части. Подробнее ограничения описаны в документе «Настройка шлюза», раздел «Общие настройки шлюза».

Алгоритм добавления маршрутов

Если для IPsecAction настройка ReverseRoute выставлена в FALSE, при создании SA по этому IPsecAction, дополнительных действий не предпринимается. Далее предполагается, что ReverseRoute выставлен в TRUE.

После построения IPsec SA вычисляется необходимый маршрут (RR). Основанием являются следующие данные:

- селектор SA (ID второй фазы IKE)

[10]Для проведения rekeying-а необходимо, чтобы время жизни обновляемого соединения было существенно больше времени, которое отводится на проведение IKE-сессии.

- адрес назначения туннельного заголовка SA (tdst)
- системная таблица маршрутизации (без учета маршрутов, добавленных подсистемой RRI).

Вычисление маршрута:

- ID партнера[11] второй фазы IKE преобразуются в адрес и маску подсети. Если это невозможно (ID является произвольным диапазоном, имеет протоколы и/или порты), то RR не создается. Полученные адрес и маска будут адресом назначения создаваемого маршрута.
- В системной таблице производится поиск туннельного адреса SA.
- Если правил не найдено (“Destination Unreachable”), RR не добавляется.
- Если найдено правило прямой маршрутизации через интерфейс, вычисленный маршрут будет через gateway tdst.
- Если найдено правило прямой маршрутизации через gateway GW, вычисленный маршрут будет через gateway GW.

Если маршрут успешно вычислен, проверяется следующее:

- Такой же маршрут был ранее добавлен подсистемой RRI для SA с тем же tdst. В этом случае увеличивается счетчик ссылок, маршрут не добавляется.
- Маршрут для SA с такими же ID второй фазы и tdst уже добавлен, но отличается. В этом случае существующий маршрут обновляется, увеличивается счетчик ссылок.
- Маршрут с такими же параметрами уже добавлен, но для SA с другим tdst. Маршрут не создается, счетчик ссылок не увеличивается.
- Маршрут, соответствующий ID партнера есть в системной таблице, но подсистемой RRI он не добавлялся. В этом случае маршрут не создается.

При удалении SA из ядра, счетчик ссылок соответствущего маршрута уменьшается, при обнулении счетчика маршрут удаляется.

Предупреждение недопустимо вручную изменять или удалять правила маршрутизации, которые автоматически формируются при использовании RRI.

Атрибут InputFilter

Атрибут InputFilter задает дополнительные правила фильтрации, присоединяемые к IPsec SA. InputFilter применяется к входящим пакетам после декапсуляции.

Если IPsecAction строит более одного SA для каждого направления, фильтрация все равно производится один раз. То есть, в комбинации ESP+AH правила фильтрации будут применены к ESP SA.

В случае вложенного IPsec, когда к пакету применяются SA, создаваемые по разным IPsecAction, пакет пройдет все InputFilter от каждого IPsecAction.

Синтаксис InputFilter = `FilterChain`

Значение по умолчанию дополнительная фильтрация входящих пакетов не производится.

[11] Поскольку протокол в ID второй фазы один для обоих партнеров, а порты без указания протокола смысла не имеют, присутствие портов и протоколов с обоих сторон не допускается.

Атрибут OutputFilter

Атрибут OutputFilter задает дополнительные правила фильтрации, присоединяемые к IPsec SA. OutputFilter применяется к исходящим пакетам до инкапсуляции.

Если IPsecAction строит более одного SA для каждого направления, фильтрация все равно производится один раз. То есть, в комбинации ESP+AH правила фильтрации будут применены к ESP SA.

В случае вложенного IPsec, когда к пакету применяются SA, создаваемые по разным IPsecAction, пакет пройдет все OutputFilter от каждого IPsecAction.

Синтаксис OutputFilter = `FilterChain`

Значение по умолчанию дополнительная фильтрация исходящих пакетов не производится.

Структура TunnelEntry

Структура TunnelEntry описывает параметры внешнего IP-заголовка пакета при использовании туннельного режима IPsec.

Имя структуры TunnelEntry

Атрибуты PeerIPAddress

LocalIPAddress

DFHandling

ReRoute

Assemble

Атрибут PeerIPAddress

Атрибут PeerIPAddress описывает туннельный адрес. Этот адрес используется для двух целей – адрес получателя во внешнем IP-заголовке и адрес IKE-партнера.

Синтаксис PeerIPAddress = IP

Значение по умолчанию если туннельный адрес используется как адрес получателя во внешнем IP-заголовке, то

для исходящего пакета берется адрес IKE партнера

если туннельный адрес используется как адрес IKE партнера, то:

для исходящего пакета берется адрес из IP-пакетов, вызвавших создание соединения

для входящего пакета принимается любой адрес.

Атрибут LocalIPAddress

Атрибут LocalIPAddress описывает туннельный адрес локального VPN-устройства.

Синтаксис LocalIPAddress = IP

Значение по умолчанию для исходящего пакета – любой из адресов сетевого интерфейса, с которого отправляется пакет.

Атрибут DFHandling

Атрибут DFHandling задает алгоритм формирования DF (Don't Fragment) бита внешнего IP-заголовка для туннельного режима IPsec.

Синтаксис DFHandling = COPY | SET | CLEAR

Значения COPY – копировать DF бит из внутреннего заголовка во внешний заголовок

SET – всегда устанавливать DF бит внешнего заголовка в 1

CLEAR – всегда сбрасывать DF бит внешнего заголовка в 0.

Значение по умолчанию COPY.

Атрибут ReRoute

Атрибут ReRoute указывает, что пакет будет подвергаться повторной маршрутизации. При использовании повторной маршрутизации может происходить повторная обработка пакета IPsec-драйвером, LSP должна создаваться с учетом этого. То есть, чтобы IPsec-пакеты с локального адреса пропускались при втором проходе. Указывать ReRoute имеет смысл для SA, заменяющих адрес назначения. Если по ходу обработки пакета адрес назначения не изменился, флаг ReRoute игнорируется.

Синтаксис ReRoute = TRUE | FALSE

Значения TRUE – исходящий пакет после цикла обработки не отправляется в драйвер сетевого интерфейса, а направляется в IP-драйвер для повторной маршрутизации. Такой пакет может попасть на повторную обработку IPsec драйвером, так что правила фильтрации должны учитывать и пропускать такие пакеты.

FALSE – указывает, что пакет не будет подвергаться повторной маршрутизации.

Значение по умолчанию FALSE

Атрибут Assemble

Атрибут Assemble указывает, что пакет будет собран из IP-фрагментов перед заворачиванием в IPsec. В транспортном режиме IPsec сборка пакетов перед инкапсуляцией производится всегда.

Синтаксис Assemble = TRUE | FALSE

Значения TRUE – означает, что пакет будет собран из IP-фрагментов перед заворачиванием в IPsec. Рекомендуется устанавливать при работе по защищенному соединению с предыдущими версиями Шлюза безопасности.

FALSE – указывает, что пакет не будет подвергаться сборке

Значение по умолчанию FALSE

Пример

```
IPsecAction CMAP_1
(
    TunnelingParameters *= TunnelEntry(
        PeerIPAddress = 10.0.2.103
        DFHandling=COPY
    ),
TunnelEntry(
        PeerIPAddress = 10.0.2.104
        DFHandling=COPY),
TunnelEntry(
        PeerIPAddress = 10.0.2.101
        DFHandling=COPY)
    ContainedProposals *= ( AH_AH_tr )
    IKERule = IKE_CMAP_1
)
```


Структуры AHProposal и ESPProposal

Структура AHProposal задает список криптографических преобразований (transforms) протокола AH в порядке убывания приоритета, которые допускаются для обработки трафика. Трафик – количество килобайт данных, обработанных данным контекстом.

Структура ESPProposal определяет список преобразований (transforms) протокола ESP в порядке убывания приоритета, которые допускаются для обработки специфицированного трафика.

Имя структуры	AHProposal
Атрибуты	Transform

Имя структуры	ESPProposal
Атрибуты	Transform

Атрибут Transform

Атрибут Transform задает список возможных групп параметров протокола AH (для структуры AHProposal) или ESP (для структуры ESPProposal), необходимых для создания SA, расположенных в порядке убывания их приоритета.

Синтаксис	Transform *= `AHTransform` # для структуры AHProposal
	Transform *= `ESPTransform` # для структуры ESPProposal
	Должен присутствовать хотя бы один трансформ.
Значение по умолчанию	не существует, атрибут обязательный.

Структура AHTransform

Структура AHTransform задает параметры контекста (SA) AH.

Неявные ограничения на количество обработанного трафика в пакетах, в байтах или на количество ошибок при проверке целостности пакетов могут содержаться в реализации конкретных криптографических алгоритмов.

Рекомендуется указывать такое время SA жизни в секундах, чтобы в основном удаление IPsec SA происходило по времени, а ограничение на объем трафика выбирать как дополнительную меру.

Имя	AHTransform
Атрибуты	LifetimeSeconds
	LifetimeKilobytes
	IntegrityAlg

Атрибут LifetimeSeconds

Атрибут LifetimeSeconds задает максимальное время существования контекста (SA) AH (в секундах).[12]

Синтаксис	LifetimeSeconds = ЦЕЛОЕ32
Значение	число из диапазона 0..2^{32}-1.
Значение по умолчанию	28800 (8 часов) [13].

Атрибут LifetimeKilobytes

Атрибут LifetimeKilobytes задает максимальное ограничение по трафику на действие контекста в килобайтах. Трафик – количество килобайт данных, обработанных данным контекстом.

Синтаксис	LifetimeKilobytes = ЦЕЛОЕ32
Значение	число из диапазона 0..2^{32}-1.
Значение по умолчанию	нет ограничений на действие SA[14]. (Замечание: количество IPsec пакетов, обработанных по одному IPsec SA, всегда ограничивается максимальным значением sequence number – 2^{32}-1 пакетов. При превышении максимального значения sequence number будет

[12] В случае использования связок (AH+ESP) в качестве элементов списка ContainedProposals структуры IPsecAction, соответствующие значения в AH- и ESP- трансформах уравниваются в меньшую сторону.

[13] В IOS по умолчанию 3600 (один час). При этом соответствующие атрибуты отсылаются в явном виде, и, более того, в предложениях от партнера требуется обязательное их наличие.

[14] В IOS по умолчанию принимается 4,608,000 (10 megabits per second for one hour). При этом соответствующие атрибуты отсылаются в явном виде, и, более того, в предложениях от партнера требуется обязательное их наличие.

запрошена смена ключей, как это делается в случае превышения ограничения в байтах.)

Примечание

Если в атрибуте IntegrityAlg задается алгоритм G2814789CPRO1-K256-MAC-255, то в этом случае максимальное допустимое значение LifetimeKilobytes – 4032 Кб.

При превышении указанного значения для созданного SA, в журнал протоколирования и на консоль будет выдано сообщение, что в созданном IPsec SA ограничение по трафику не соответствует допустимому ограничению для используемого криптографического алгоритма:

```
"SA traffic limit exceeds limitations imposed by the cryptograghic
algorithm"
```

Атрибут IntegrityAlg

Атрибут IntegrityAlg задает алгоритм проверки целостности в рамках создаваемого контекста.

Если же существует необходимость задать несколько алгоритмов (их комбинацию) проверки целостности, то используйте альтернативный подход: в атрибуте Transform структуры AHProposal укажите список структур AHTransform, а в каждой структуре AHTransform задайте только один алгоритм проверки целостности.

Синтаксис IntegrityAlg = "STB1176199-H96-HMAC-250"|"STB34101CIPH-K256-MAC-252"|"G2814789AV1-K256-MAC-251"|**"MD5-H96-HMAC"|"SHA1-H96-HMAC"**

Значение Возможные значения:

"STB1176199-H96-HMAC-250" – HMAC СТБ 1176.1-99 (96 бит)

"STB34101CIPH-K256-MAC-252" – имитовставка по СТБ 34.101.31 (64 бит)

"G2814789AV1-K256-MAC-251" – имитовставка по ГОСТ 28147 (64 бит) "MD5-H96-HMAC" – HMAC MD5 (96 бит)

"SHA1-H96-HMAC" – HMAC SHA-1 (96 бит)

Значение по умолчанию не существует, атрибут обязательный.

Структура ESPTransform

Структура ESPTransform задает параметры контекста (SA) ESP.

Неявные ограничения на количество обработанного трафика в пакетах, в байтах или на количество ошибок при проверке целостности пакетов могут содержаться в реализации конкретных криптографических алгоритмов.

Рекомендуется указывать такое время SA жизни в секундах, чтобы в основном удаление IPsec SA происходило по времени, а ограничение на объем трафика выбирать как дополнительную меру.

Имя	ESPTransform
Атрибуты	LifetimeSeconds
	LifetimeKilobytes
	CipherAlg
	IntegrityAlg

Атрибут LifetimeSeconds

Атрибут LifetimeSeconds задает максимальное время существования контекста в секундах.[15]

Синтаксис	LifetimeSeconds = ЦЕЛОЕ32
Значение	Целое число из диапазона 1..2^{32}-1.
Значение по умолчанию	28800 (8 часов) [16].

Атрибут LifetimeKilobytes

Атрибут LifetimeKilobytes задает максимальное ограничение по трафику на действие контекста в килобайтах. Трафик – количество килобайт данных, обработанных данным контекстом.

Синтаксис	LifetimeKilobytes = ЦЕЛОЕ32
Значение	Целое число из диапазона 1..2^{32}-1.
Значение по умолчанию	нет ограничений на действие SA[17]. (Замечание: количество IPsec пакетов, обработанных по одному IPsec SA, всегда ограничивается максимальным значением sequence number – 2^{32}-1 пакетов. При превышении максимального значения sequence number будет запрошена смена ключей, как это делается в случае превышения ограничения в байтах.)

[15] В случае использования связок (AH+ESP) в качестве элементов списка ContainedProposals структуры IPsecAction, соответствующие значения в AH- и ESP- трансформах уравниваются в меньшую сторону

[16] В IOS по умолчанию 3600 секунд. При этом соответствующие атрибуты отсылаются в явном виде, и, более того, в предложениях от партнера требуется обязательное их наличие.

[17] В IOS по умолчанию принимается 4,608,000 (10 megabits per second for one hour). При этом соответствующие атрибуты отсылаются в явном виде, и, более того, в предложениях от партнера требуется обязательное их наличие.

Примечание

Если используются алгоритмы G2814789CPRO1-K256-CBC-254, G2814789CPRO1-K256-MAC-65535 (атрибуты CipherAlg, IntegrityAlg), то в этом случае максимальное допустимое значение LifetimeKilobytes – 4032 Кб.

При превышении указанного значения для созданного SA, в журнал протоколирования и на консоль будет выдано сообщение, что в созданном IPsec SA ограничение по трафику не соответствует допустимому ограничению для используемого криптографического алгоритма:

```
"SA traffic limit exceeds limitations imposed by the cryptograghic
algorithm"
```

Атрибут CipherAlg

Атрибут CipherAlg задает алгоритм шифрования трафика в рамках создаваемого контекста.

Если же существует необходимость задать несколько алгоритмов шифрования, то используйте альтернативный подход: в атрибуте Transform структуры ESPProposal укажите список структур ESPTransform, а в каждой структуре ESPTransform задайте только один алгоритм шифрования.

Синтаксис	CipherAlg = "NULL"|"G2814789AV1-K256-CBC-250"|"STB34101CIPH-K256-CBC-252"|"DES-CBC"|"DES3-K168-CBC"|"AES-K128-CBC"|"AES-K256-CBC"
Значение	Возможные значения: "NULL" – NULL (данные не шифруются) "G2814789AV1-K256-CBC-250" – шифрование и расшифрование по ГОСТ 28147 в режиме CFB с длиной ключа 256 бит "STB34101CIPH-K256-CBC-252" – шифрование и расшифрование по СТБ 34.101.31-2011 в режиме CFB с длиной ключа 256 бит.DES в режиме CBC с явным IV длиной 32 бита "DES-CBC" – DES в режиме CBC "DES3-K168-CBC" – DES3 в режиме CBC "AES-K128-CBC" – AES в режиме CBC с длиной ключа 128 "AES-K192-CBC" – AES в режиме CBC с длиной ключа 192 "AES-K256-CBC" – AES в режиме CBC с длиной ключа 256
Значение по умолчанию	не существует, атрибут обязательный.

Атрибут IntegrityAlg

Атрибут IntegrityAlg задает алгоритм проверки целостности пакета в рамках создаваемого контекста.

Если же существует необходимость задать несколько алгоритмов проверки целостности (их комбинацию), то используйте альтернативный подход: в атрибуте Transform структуры ESPProposal укажите список структур ESPTransform, а в каждой структуре ESPTransform задайте только один алгоритм проверки целостности пакета.

Если CipherAlg имеет значение "NULL", IntegrityAlg указывать обязательно.

Синтаксис	IntegrityAlg = "STB1176199-H96-HMAC-65530"|"G2814789AV1-K256-MAC-65531"|"STB34101CIPH-K256-MAC-65532"|"MD5-H96-HMAC"|"SHA1-H96-HMAC"
Значение	Возможные значения: "STB1176199-H96-HMAC-65530" – HMAC СТБ 1176.1-99 (96 бит) "G2814789AV1-K256-MAC-65531" – имитовставка по СТБ 34.101.31 (64 бит) "STB34101CIPH-K256-MAC-65532" – имитовставка по ГОСТ 28147 (64 бит) "MD5-H96-HMAC" – HMAC MD5 (96 бит) "SHA1-H96-HMAC" – HMAC SHA-1 (96 бит)
Значение по умолчанию	если в атрибуте IntegrityAlg алгоритм не указан, а в атрибуте CipherAlg алгоритм указан, то проверка целостности пакета не производится.

Пример структуры ESPProposal

```
ESPTransform esp_trf_01(
    LifetimeSeconds = 28800
    LifetimeKilobytes = 4608000
    CipherAlg  = "G2814789AV1-K256-CBC-250"
    IntegrityAlg = "STB1176199-H96-HMAC-65530"
)
ESPTransform esp_trf_02(
    LifetimeSeconds = 28800
    LifetimeKilobytes = 4608000
    CipherAlg  = "STB34101CIPH-K256-CBC-252"
    IntegrityAlg = "G2814789AV1-K256-MAC-65531"
)
ESPTransform esp_trf_03(
    LifetimeSeconds = 28800
    LifetimeKilobytes = 4608000
    CipherAlg  = "NULL"
    IntegrityAlg = "STB34101CIPH-K256-MAC-65532"
)
ESPProposal ESP_1(
    Transform *= esp_trf_01,esp_trf_02,esp_trf_03
)
```

Структура IKERule

Структура IKERule описывает правило создания контекста соединения для протокола IKE.

Имя структуры	IKERule
Атрибуты	IKEPeerIPFilter
	IKELocalIPFilter
	DoNotUseCommitBit
	DoNotUseDPD
	DPDIdleDuration
	DPDResponseDuration
	DPDRetries
	AggrModeAuthMethod
	MainModeAuthMethod
	AggrModePriority
	Transform
	IKECFGPool
	IKECFGBindToPeerAddress
	Priority
	XAuthServerEnabled
	AAAUserName

Атрибут IKEPeerIPFilter

Атрибут IKEPeerIPFilter описывает список допустимых IP-адресов партнера, при которых применяется данное правило.

Атрибут используется шлюзом безопасности, выступающим в роли ответчика IKE-сессии, при проверке UDP-заголовка первого (входящего) пакета.

Для шлюза безопасности, выступающего в роли инициатора создания IKE-сессии, атрибут игнорируется.

Синтаксис	IKEPeerIPFilter* = IP, IP/ЦЕЛОЕ32
Значение	IP-адрес
	IP/ЦЕЛОЕ32 – IP-адрес с маской подсети
Значение по умолчанию	допускаются любые IP-адреса

Атрибут IKELocalIPFilter

Атрибут IKELocalIPFilter описывает список допустимых локальных IP-адресов, при которых применяется данное правило.

Атрибут используется шлюзом безопасности, выступающим в роли ответчика IKE-сессии при проверке UDP-заголовка первого (входящего) пакета.

Шлюз в роли инициатора IKE-сессии, имея адрес потенциального удаленного партнера, может определить, с какого адреса на локальной стороне будут уходить пакеты (используя таблицу маршрутизации), и если этот адрес не попадает под условия, описанные IKELocalIPFilter, то соединение не устанавливается.

Синтаксис IKELocalIPFilter* = IP, IP/ЦЕЛОЕ32

Значения IP-адрес

IP/ЦЕЛОЕ32 – IP-адрес с маской подсети

Значение по умолчанию допускаются любые локальные IP-адреса.

Атрибут DoNotUseCommitBit

Атрибут DoNotUseCommitBit применяется для совместимости со сторонними продуктами, которые некорректно обрабатывают Commit-Bit в заголовке ISAKMP пакета. В таких сторонних Агентах данный параметр должен быть выставлен в TRUE (подробнее см. RFC 2408).

Синтаксис DoNotUseCommitBit = TRUE | FALSE

Значение TRUE – отключает отсылку Commit-Bit в Quick Mode.
При данном значении респондер Quick Mode в заголовке 2-го ISAKMP пакета Commit-Bit не выставляет и дополнительный 4-й пакет не высылает; инициатор Quick Mode в заголовке 3-го ISAKMP пакета Commit-Bit не выставляет и дополнительный 4-й пакет не ожидает.

FALSE – Commit-Bit используется для сигнала синхронизации обмена ключа

Атрибут DoNotUseDPD

Атрибут DoNotUseDPD задает режим использования протокола DPD (Dead Peer Detection).

Синтаксис DoNotUseDPD = TRUE | FALSE

Значение TRUE – не использовать протокол DPD

FALSE – использовать протокол DPD

Значение по умолчанию FALSE.

Атрибут DPDIdleDuration

Атрибут DPDIdleDuration задает допустимый период времени отсутствия входящего трафика от партнера, по истечении которого, при наличии исходящего трафика, активируется DPD-сессия. Если атрибут DoNotUseDPD = TRUE, то атрибут DPDIdleDuration игнорируется.

Синтаксис DPDIdleDuration = ЦЕЛОЕ32

Значение 1..32767

Значение по умолчанию 60.

Атрибут DPDResponseDuration

Атрибут DPDResponseDuration задает время ожидания ответа от партнера на DPD запрос в секундах. Если атрибут DoNotUseDPD = TRUE, то атрибут DPDResponseDuration игнорируется.

Синтаксис DPDResponseDuration = ЦЕЛОЕ32

Значение 1..300

Значение по умолчанию 5.

Атрибут DPDRetries

Атрибут DPDRetries задает число попыток провести DPD обмен. Если все попытки закончились неудачей, защищенное соединение (IKE-контекст) считается "мертвым", и производится попытка создать его заново. Если атрибут DoNotUseDPD = TRUE, то атрибут DPDRetries игнорируется.

Синтаксис DPDRetries = ЦЕЛОЕ32

Значение 1..10

Значение по умолчанию 3.

Атрибут AggrModeAuthMethod

Атрибут AggrModeAuthMethod содержит список структур, определяющих способ и параметры аутентификации в агрессивном режиме IKE. В списке не должно быть задано двух одинаковых методов аутентификации.

Примечание: Хотя бы один из атрибутов AggrModeAuthMethod или MainModeAuthMethod должен быть задан.

Синтаксис AggrModeAuthMethod* = AuthMethodDSSSign | AuthMethodRSASign | FuthMethodGOSTSign | AuthMethodPreshared

Значение AuthMethodDSSSign – аутентификация DSA подписью

AuthMethodRSASign – аутентификация RSA подписью

AuthMethodGOSTSign – аутентификация при помощи подписи алгоритмом ГОСТ34.10

AuthMethodPreshared – аутентификация при помощи предопределенного ключа.

Значение по умолчанию При отсутствии MainModeAuthMethod атрибут является обязательным.

При наличии атрибута MainModeAuthMethod Aggressive Mode не проводится.

Атрибут MainModeAuthMethod

Атрибут MainModeAuthMethod содержит список структур, определяющих способ и параметры аутентификации в основном режиме IKE. В списке не должно быть задано двух одинаковых методов аутентификации.

Примечание: Хотя бы один из атрибутов AggrModeAuthMethod или MainModeAuthMethod должен быть задан.

Синтаксис	MainModeAuthMethod* = AuthMethodDSSSign | AuthMethodRSASign | FuthMethodGOSTSign | AuthMethodPreshared
Значение	AuthMethodDSSSign – аутентификация DSA подписью AuthMethodRSASign – аутентификация RSA подписью AuthMethodGOSTSign – аутентификация при помощи подписи алгоритмом ГОСТ3410 AuthMethodPreshared – аутентификация при помощи предопределенного ключа.
Значение по умолчанию	При отсутствии атрибута AggrModeAuthMethod атрибут является обязательным. При наличии атрибута AggrModeAuthMethod Main Mode не проводится.

Атрибут AggrModePriority

AggrModePriority задает режим использования Aggressive Mode. Атрибут используется только для инициатора в случае, если заданы значения MainModeAuthMethod и AggrModeAuthMethod одновременно. Атрибут игнорируется, если задан только один режим (Main Mode или Aggressive Mode)

Синтаксис	AggrModePriority = TRUE | FALSE
Значение	TRUE – Aggressive Mode является более приоритетным, инициатор начинает первую фазу IKE в "агрессивном" режиме. FALSE – Main Mode является более приоритетным, то инициатор начинает первую фазу IKE в "основном" режиме.
Значение по умолчанию	FALSE.

Атрибут Transform

Атрибут Transform задает список допустимых наборов криптографических параметров для ISAKMP SA. Количество элементов списка не ограничего.

Синтаксис	Transform* = `IKETransform`
Значение по умолчанию	не существует, атрибут обязательный.

Атрибут IKECFGPool

Атрибут IKECFGPool задает адреса, которые будут выдаваться партнерам по протоколу IKECFG.

Для структур IKERule с различающимися значениями поля IKECFGBindToPeerAddress использование в списках одинаковых объектов AddressPool не допускается.

Для всех несовпадающих объектов AddressPool в полях IKECFGPool структур IKERule адресные пространства не должны пересекаться.

Синтаксис	IKECFGPool* = `AddressPool`
Значение по умолчанию	Адреса партнерам по IKECFG не выдаются. На запросы партнёра о выдаче IKECFG-адреса отсылается отказ без удаления

настраиваемого соединения, что позволяет партнеру провести IKE Quick Mode и построить IPsec SA без использования IKECFG-адреса.

Примечание

Если предполагается использовать для авторизации и аутентификации RADIUS-сервер, то одновременно настраивать IKECFG параметры на отдельном Bel VPN Gate и на RADIUS-сервере крайне нежелательно:

1. Если будет задан локальный IKECFG пул (атрибут IKECFGPool), то в случае получения данных авторизации от RADIUS-сервера, будут задействованы данные из локального IKECFG пула, а данные от RADIUS-сервера будут игнорироваться.
2. Если RADIUS-сервер выдал Framed-IP-Address (Framed-IP-Address – атрибут RADIUS-сервера, соответствующий IKECFG-адресу, высылаемому Bel VPN Gate партнеру), который попадает в один из AddressPool, задействованных в LSP, то Bel VPN Gate игнорирует авторизационные данные от RADIUS-сервера и пытается установить соединение с партнёром без IKECFG.

Атрибут IKECFGBindToPeerAddress

IKECFGBindToPeerAddress задает один из двух вариантов параметров, по которым IKECFG-сервер идентифицирует клиентов.

При выдаче IKECFG адреса устанавливается однозначное соответствие выданного ip-адреса и клиента.

В случае отсутствия в правиле IKECFGPool, поле игнорируется.

Синтаксис IKECFGBindToPeerAddress = TRUE | FALSE

Значение TRUE – клиенты идентифицируются по адресу и порту[18] партнера.

FALSE – клиенты идентифицируются по партнёрскому ID первой фазы IKE.

Значение по умолчанию FALSE.

Атрибут Priority

Атрибут Priority задает порядок выбора IKE-правила ответчиком, если по параметрам, присланным партнером, подходят несколько правил. Из двух подходящих правил с разными приоритетами выберется то, у которого значение Priority меньше.

Порядок выбора IKE-правила из правил с одинаковым приоритетом не определен.

Синтаксис Priority = ЦЕЛОЕ32

[18] Имеются в виду адрес и порт партнера, видимые гейту, по которым построен ISAKMP SA. Следует учитывать, что при наличии между партнерами NAT-устройства IKE-обмен может начинаться с одной парой адрес-порт, а завершаться (с созданием ISAKMP SA) с другой. Именно конечная пара адрес-порт используется для идентификации партнёра в ISAKMP SA при выдаче IKECFG адреса. При установке значения в TRUE следует учитывать, что NAPT-устройства могут назначать разные ip-адреса и порты клиенту, что в свою очередь может привести к быстрому исчерпанию пула. В этом случае для ISAKMP и IPSec SA рекомендуется устанавливать короткие времена жизни для освобождения IKECFG адресов.

Значение Целое число из диапазона 1. 2^{32}-1.

Значение по умолчанию 2^{32}-1.

Пример

```
IKERule IKE_CMAP_1
(
    Transform* = IKETransform_1
    AggrModeAuthMethod  *= auth_ca
    MainModeAuthMethod  *= auth_ca
    DoAutopass          = TRUE
    DPDIdleDuration     = 30
    DPDResponseDuration = 2
    DPDRetries          = 5
    IKELocalIPFilter* = FilterEntry( IPAddress =  10.0.2.6)
)
```

Атрибут XAuthServerEnabled

Атрибут XAuthServerEnabled включает поддержку XAuth-сервера[19].

Синтаксис XAuthServerEnabled = TRUE | FALSE

Значение TRUE – Bel VPN Gate выступает в роли XAuth-сервера.
Для данного IKE правила шлюз требует поддержку метода аутентификации с проведением XAuth.
После успешного построения ISAKMP SA, Bel VPN Gate инициирует XAuth-сессию.

FALSE –Bel VPN Gate работает в обычном режиме, XAuth-обмены не проводятся.

Значение по умолчанию FALSE.

Атрибут AAAUserName

Атрибут AAAUserName задает способ получения идентификатора и пароля пользователя для аутентификации партнёра на RADIUS сервере.

Синтаксис AAAUserName = INTERACTIVE | IKE_ID | CERT_SUBJ_CN | CERT_SUBJ_OU | CERT_ALTSUBJ_EMAIL | CERT_ALTSUBJ_DNS

[19] В Bel VPN Gate 4.1 включено ограничение на использование XAuth с Bel VPN Client. Если Клиент присылает S-Terra VendorID, то для использования XAuth от него требуется дополнительный флаг – признак поддержки XAuth, зарезервированный для будущих версий Клиента. Принято, что Bel VPN Client до версии, в частности, 4.1 включительно, в полной мере XAuth не поддерживает, и IKERule с выставленным XAuthServerEnabled для такого Клиента выбрано не будет.

Значение

INTERACTIVE – для запроса в RADIUS сервер используются данные, заполненные партнёром в процессе проведения XAuth-сессии.

Требуется наличие параметра XAuthServerEnabled, выставленного в значение TRUE.
После успешного построения ISAKMP SA Клиенту отсылается XAuth-запрос на заполнение полей `user` и `password`, которые затем передаются на RADIUS сервер.
После успешной аутентификации на RADIUS сервере, Клиенту отсылается XAuth-сообщение об успешном проведении дополнительной аутентификации.

IKE_ID – для запроса в RADIUS сервер используются:

В качестве идентификатора пользователя – полное печатное значение IKE-идентификатора партнёра ISAKMP соединения[20].
В качестве пароля – содержимое хранилища, заданного параметром NonInteractivePassword структуры AAASettings.
При наличии параметра XAuthServerEnabled, выставленного в значение TRUE, после успешного построения ISAKMP SA Клиенту отсылается сообщение об успешной дополнительной аутентификации без предварительных запросов[21].

CERT_SUBJ_CN – для запроса в RADIUS сервер используются:

В качестве идентификатора пользователя – полное печатное значение поля `CommonName ("CN=…")` из описания `(Subject)` сертификата партнёра ISAKMP соединения, использованного при проверке подписи[22].
В качестве пароля – содержимое хранилища, заданного параметром NonInteractivePassword структуры AAASettings.
При наличии параметра XAuthServerEnabled, выставленного в значение TRUE, после успешного построения ISAKMP SA Клиенту отсылается сообщение об успешной дополнительной аутентификации без предварительных запросов[23].

CERT_SUBJ_OU – для запроса в RADIUS сервер используются:

В качестве идентификатора пользователя – полное печатное значение поля `OrganizationUnit ("OU=…")` из описания `(Subject)` сертификата партнёра ISAKMP соединения, использованного при проверке подписи[24].
В качестве пароля – содержимое хранилища, заданного параметром NonInteractivePassword структуры AAASettings.
При наличии параметра XAuthServerEnabled, выставленного в значение TRUE, после успешного построения ISAKMP SA

[20] Значение совпадает с IKE identity партнёра, передаваемому в лог, а также в выводе утилиты sa_mgr.

[21] Данное поведение реализовано для работы с Клиентами, которые ожидают от Гейта XAuth-запрос прежде чем проводить Quick mode.

[22] Значение совпадает со значением поля CN описания задействованного сертификата, передаваемому в лог, а также в выводе утилиты cert_mgr show для сертификата Клиента.

[23] Данное поведение реализовано для работы с Клиентами, которые ожидают от Гейта XAuth-запрос прежде чем проводить Quick mode.

[24] Значение совпадает со значением поля CN описания задействованного сертификата, передаваемому в лог, а также в выводе утилиты cert_mgr show для сертификата Клиента.

Клиенту отсылается сообщение об успешной дополнительной аутентификации без предварительных запросов[25].

CERT_ALTSUBJ_EMAIL – для запроса в RADIUS сервер используются:

В качестве идентификатора пользователя – полное печатное значение поля `EMail ("EMAIL=…")` из альтернативного описания `(Alternative subject)` сертификата партнёра ISAKMP соединения, использованного при проверке подписи[26].
В качестве пароля – содержимое хранилища, заданного параметром NonInteractivePassword структуры AAASettings.
При наличии параметра XAuthServerEnabled, выставленного в значение TRUE, после успешного построения ISAKMP SA Клиенту отсылается сообщение об успешной дополнительной аутентификации без предварительных запросов[27].

CERT_ALTSUBJ_DNS – для запроса в RADIUS сервер используются:

В качестве идентификатора пользователя – полное печатное значение поля `DNS ("DNS=…")` из альтернативного описания `(Alternative subject)` сертификата партнёра ISAKMP соединения, использованного при проверке подписи[28].
В качестве пароля – содержимое хранилища, заданного параметром NonInteractivePassword структуры AAASettings.
При наличии параметра XAuthServerEnabled, выставленного в значение TRUE, после успешного построения ISAKMP SA Клиенту отсылается сообщение об успешной дополнительной аутентификации без предварительных запросов[29].

Значение по умолчанию

Запрос в RADIUS сервер не производится.
При наличии параметра XAuthServerEnabled, выставленного в значение TRUE, после успешного построения ISAKMP SA Клиенту отсылается сообщение об успешной дополнительной аутентификации без предварительных запросов[30].

[25]Данное поведение реализовано для работы с Клиентами, которые ожидают от Гейта XAuth-запрос прежде чем проводить Quick mode.

[26] Значение совпадает со значением поля CN описания задействованного сертификата, передаваемому в лог, а также в выводе утилиты cert_mgr show для сертификата Клиента.

[27]Данное поведение реализовано для работы с Клиентами, которые ожидают от Гейта XAuth-запрос прежде чем проводить Quick mode.

[28] Значение совпадает со значением поля CN описания задействованного сертификата, передаваемому в лог, а также в выводе утилиты cert_mgr show для сертификата Клиента.

[29]Данное поведение реализовано для работы с Клиентами, которые ожидают от Гейта XAuth-запрос прежде чем проводить Quick mode.

[30]Данное поведение реализовано для работы с Клиентами, которые ожидают от Гейта XAuth-запрос прежде чем проводить Quick mode.

Структура AAASettings

Структура AAASettings описывает глобальные настройки внешнего сервиса аутентификации, авторизации и аккаунтинга. В конфигурации должна быть только одна структура данного типа. Имя этой структуре не присваивается. Пример настройки можно посмотреть в Приложении.

Имя структуры	AAASettings
Атрибуты	RadiusServer
	Secret
	NonInteractiveUserPassword
	NoProxyARP
	Retries
	ResponseTimeout

Атрибут RadiusServer

Атрибут RadiusServer задает адрес RADIUS-сервера, к которому производится запрос[31]. Для доступа используется протокол UDP, порт 1645.

Синтаксис	RadiusServer = IP
Значение	IP-адрес
Значение по умолчанию	не существует, атрибут обязательный.

Атрибут Secret

Атрибут Secret задает пароль доступа к RADIUS-серверу.

Синтаксис	Secret = СТРОКА
Значение	имя предопределенного (Preshared) ключа, хранящегося в базе данных Агента
Значение по умолчанию	не существует, атрибут обязательный.

Атрибут NonInteractiveUserPassword

Атрибут NonInteractiveUserPassword задает единый пароль пользователя в случае обращения к RADIUS-серверу для имён пользователя, полученных неинтерактивным способом.

Синтаксис	NonInteractiveUserPassword = СТРОКА
Значение по умолчанию	не существует, атрибут обязательный.

[31] Поскольку RADIUS протокол не отвечает достаточному уровню безопасности, пользователь сам должен обеспечить нахождение данного адреса в пределах доверяемого защищённого пространства.

Атрибут NoProxyARP

Атрибут NoProxyARP задает режим работы устройства в роли ProxyARP для указанного множества адресов. Добавление proxy-arp записей не гарантирует маршрутизацию пакетов на самом шлюзе безопасности.
Для автоматического добавления маршрутов, proxy-arp можно использовать совместно с механизмом RRI.
При удалении SA, соответствующие proxy-arp записи удаляются.

Синтаксис NoProxyARP = TRUE | FALSE

Значение FALSE – для IKECFG адресов, назначенных IKE-партнёрам RADIUS-сервером, в системную ARP-таблицу добавляются соответствующие "proxy-arp" записи. Но если IP-адрес не попадает ни в одну из локально подсоединенных сетей, proxy-arp запись не создается, и это не считается ошибкой.

TRUE – адреса не проксируются.

Значение по умолчанию FALSE.

Атрибут Retries

Атрибут Retries устанавливает число попыток посылки запросов на RADIUS-сервер.

Синтаксис Retries = ЦЕЛОЕ32

Значение Целое число из диапазона 1..10

Значение по умолчанию 4.

Атрибут ResponseTimeout

Атрибут ResponseTimeout устанавливает интервал в секундах между повторными попытками посылки запросов на RADIUS-сервер.

Синтаксис ResponseTimeout = ЦЕЛОЕ32

Значение Целое число из диапазона 1..65535

Значение по умолчанию 3.

Структура IKETransform

Структура IKETransform задает набор параметров, необходимых для создания ISAKMP SA.

Имя структуры	IKETransform
Атрибуты	LifetimeSeconds
	LifetimeKilobytes
	LifetimeSessions
	NoSmoothRekeying
	CipherAlg
	HashAlg
	GroupID
	RestrictAuthenticationTo

Атрибут LifetimeSeconds

Атрибут LifetimeSeconds задает время существования IKE-контекста (в секундах).

Синтаксис	LifetimeSeconds = ЦЕЛОЕ32
Значение	Целое число из диапазона $1..2^{32}-1^{32}$
Значение по умолчанию	нет ограничений на действие SA.

Для совместимости с IOS-партнером (Cisco) нужно всегда указывать в своем предложении атрибут LifetimeSeconds – время жизни в секундах и высылать это значение IOS-партнеру. В противном случае, IOS будет пытаться поместить в принятое предложение новый атрибут – время жизни SA по времени, которое IOS-ом будет установлено для создаваемого SA. Это является неприемлемым для шлюза безопасности и, будучи партнером IOS, он прекращает установление соединения.

Атрибут LifetimeKilobytes

Атрибут LifetimeKilobytes задает максимальное ограничение по трафику на действие контекста в килобайтах.

Синтаксис	LifetimeKilobytes = ЦЕЛОЕ32
Значение	Целое число из диапазона $1..2^{32}-1$
Значение по умолчанию	нет ограничений на действие SA.

[32] В IOS допустимы только значения в диапазоне 60-86400.

Атрибут LifetimeSessions

Атрибут LifetimeSessions задает ограничение по числу IPsec SA (числу успешных Quick Mode – QM), которые можно сделать с использованием одного IKE-контекста. Данный параметр не согласуется с партнерами в процессе создания соединения, поэтому при уничтожении ISAKMP SA по достижении этого ограничения партнеру всегда отсылается Delete payload.

Синтаксис LifetimeSessions = ЦЕЛОЕ32

Значение Целое число из диапазона $1..2^{32}-1$

Значение по умолчанию нет ограничений на действие SA по числу созданных под его защитой IPsec SA.

Атрибут NoSmoothRekeying

Атрибут NoSmoothRekeying задает режим "мягкой" смены ключевого материала.

Синтаксис NoSmoothRekeying = TRUE | FALSE

Значение TRUE – заблаговременная смена ключевого материала (rekeying) не проводится. При отсутствии подходящего ISAKMP SA, новый ISAKMP SA создаётся только по запросу из ядра на создание IPsec SA – при наличии исходящего IP-пакета, либо по инициативе партнера. В результате процесс создания IPsec SA существенно задерживается.

FALSE – заблаговременно, незадолго до окончания действия ISAKMP SA, на его основе (по тем же правилам и с теми же Identity) проводится IKE-сессия по созданию нового ISAKMP SA – rekeying. Rekeying не проводится, если за время существования старого SA под его защитой не было никакого трафика[33].

Значение по умолчанию FALSE

Атрибут CipherAlg

Атрибут CipherAlg задает алгоритм шифрования для ISAKMP.

Указывается только один алгоритм шифрования.

Если же существует необходимость задать несколько алгоритмов шифрования (их комбинацию), то используйте альтернативный подход: в атрибуте Transform структуры IKERule укажите список структур IKETransform, а в каждой структуре IKETransform задайте только один алгоритм шифрования (см. Пример структуры IKERule).

Синтаксис CipherAlg = "G2814789AV1-K256-CBC-65530"|"STB34101CIPH-K256-CBC-65532"|"DES-CBC"|"DES3-K168-CBC"|"AES-K128-CBC"|"AES-K192-CBC"|"AES-K256-CBC"

Значение "G2814789AV1-K256-CBC-65530" – шифрование и расшифрование по ГОСТ 28147 в режиме CFB с длиной ключа 256 бит

"STB34101CIPH-K256-CBC-65532" – шифрование и расшифрование по СТБ 34.101.31-2011 в режиме CFB с длиной ключа 256 бит.

[33] Для проведения rekeying необходимо, чтобы время жизни обновляемого соединения было существенно больше времени, которое отводится на проведение IKE-сессии.

"DES-CBC" – DES в режиме CBC

"DES3-K168-CBC" – DES3 в режиме CBC

"AES-K128-CBC" – AES в режиме CBC с длиной ключа 128

"AES-K192-CBC" – AES в режиме CBC с длиной ключа 192

"AES-K256-CBC" – AES в режиме CBC с длиной ключа 256

Значение по умолчанию не существует, атрибут обязательный.

Атрибут HashAlg

Атрибут HashAlg задает алгоритм вычисления хэша для ISAKMP[34]..

Указывается только один алгоритм хэширования.

Если же существует необходимость задать несколько алгоритмов хэширования, то используйте альтернативный подход: в атрибуте Transform структуры IKERule укажите список структур IKETransform, а в каждой структуре IKETransform задайте только один алгоритм хэширования (см. Пример структуры IKERule).

Синтаксис HashAlg = "STB1176199-65530"|"STB34101HASH-65532"

Значение "STB1176199-65530" – СТБ 1176.1-99

"STB34101HASH-65532" – СТБ 34.101.31-2011 (раздел 6.9 – хэширование)

Значение по умолчанию не существует, атрибут обязательный.

Атрибут GroupID

Атрибут GroupID описывает параметр для выработки ключевого материала для ISAKMP. Используется алгоритм Диффи-Хеллмана.

Если существует необходимость задать список, то используйте альтернативный подход: в атрибуте Transform структуры IKERule укажите список структур IKETransform, а в каждой структуре IKETransform задайте только один элемент списка (см. Пример структуры IKERule – MainMode).

При использовании атрибута GroupID для Aggressive Mode инициатор может предложить только одну Oakley группу. Это связано с тем, что в Aggressive Mode вычисление ключевых пар в соответствии с предлагаемым алгоритмом производится сразу, не дожидаясь ответа от партнера.

Синтаксис GroupID = VKO_1B|MODP_768|MODP_1024|MODP_1536|BELTDH

Значение MODP_768 – стандартная Oakley-группа с длиной ключа 768 бит – группа 1

MODP_1024 – стандартная Oakley-группа с длиной ключа 1024 бита – группа 2

[34] Если в правиле IKERule, использующем данный IKETransform указан метод аутентификации типа AuthMethodGOSTSign, то алгоритм вычисления хэша для ISAKMP *не может быть указан* MD5 или SHA1.

MODP_1536 – стандартная Oakley-группа с длиной ключа 1536 бит – группа 5

BELTDH – протокол формирования общего ключа на основе эллиптических кривых согласно СТБ 34.101.66-2014

Значение по умолчанию не существует, атрибут обязательный.

Примечание

Стоит отметить, что в правиле IKE (IKERule) предоставление партнеру выбора различных элементов списка возможно только в основном режиме IKE (MainMode). Если правило IKE предусматривает агрессивный режим (присутствует структура AggrModeAuthMethod), то в этом правиле IKERule во всех структурах IKETransform атрибут GroupID должен иметь только одно значение, и оно должно быть одинаковым во всех структурах IKETransform, т.е. должна быть указана одна и та же группа.

В ряде случаев это приводит к потере гибкости конфигурации и, следовательно, к применению не рекомендуется.

Атрибут RestrictAuthenticationTo

Атрибут RestrictAuthenticationTo определяет, с каким типом аутентификации может использоваться данный трансформ. Если не задано методов аутентификации подходящего типа, соответствующих используемому режиму (main/aggressive), данный трансформ не будет использован. Если для способа аутентификации в IKERule нет подходящего трансформа, произойдет ошибка разбора конфигурации.

Синтаксис RestrictAuthenticationTo = AuthMethodBELTSign|AuthMethodPreshared

Значение AuthMethodBELTSign – для аутентификации используется сертификат открытого ключа в формате X.509

AuthMethodPreshared – для аутентификации используется предопределенный ключ

Значение по умолчанию ограничение не установлено.

Пример структуры IKERule и IKETransform

```
IKETransform ike_trf_01(
 LifetimeSeconds = 28800
 CipherAlg = "G2814789AV1-K256-CBC-65530"
 HashAlg = "STB1176199-65530"
 GroupID = BELTDH
)
IKETransform ike_trf_02(
 LifetimeSeconds = 28800
 CipherAlg = "STB34101CIPH-K256-CBC-65532"
 HashAlg = "STB34101HASH-65532"
 GroupID = MODP_1536
)
IKETransform ike_trf_03(
 LifetimeSeconds = 28800
 CipherAlg = "STB34101CIPH-K256-CBC-65532"
 HashAlg = "STB34101HASH-65532"
```

```
 GroupID = MODP_1024
)
IKETransform ike_trf_04(
 LifetimeSeconds = 28800
 CipherAlg = "G2814789AV1-K256-CBC-65530"
 HashAlg = "STB34101HASH-65532"
 GroupID = MODP_768
)
IKERule ike_rule(
 DoNotUseDPD = FALSE
 DPDIdleDuration = 60
 DPDResponseDuration = 5
 DPDRetries = 3
 MainModeAuthMethod *= auth_method_01
 Transform *= ike_trf_01,ike_trf_02,ike_trf_03,ike_trf_04
 DoAutopass = TRUE
)
IdentityEntry auth_identity_01(
)
AuthMethodPreshared auth_method_01(
 SharedIKESecret = "dfd"
 LocalID = auth_identity_01
)
```


Структура AuthMethod{DSS|RSA|GOST}Sign

Указанная структура задает аутентификационную информацию при использовании сертификатов. Алгоритм (RSA, DSA, GOST), указанный в названии структуры, является криптографическим алгоритмом аутентификации сторон.

AuthMethodDSASign – аутентификация при помощи подписи, созданной с использованием алгоритма DSA.

AuthMethodRSASign – аутентификация при помощи подписи, созданной с использованием алгоритма RSA.

AuthMethodGOSTSign – аутентификация при помощи подписи, созданной с использованием алгоритма СТБ 1176.2-99/ СТБ 34.101.45-2013.

Имя структур	AuthMethodDSSSign
	AuthMethodRSASign
	AuthMethodGOSTSign
Атрибуты	LocalID
	RemoteID
	LocalCredential
	RemoteCredential
	AcceptCredentialFrom
	DoNotMapLocalIDToCert
	DoNotMapRemoteIDToCert
	SendRequestMode
	SendCertMode

Атрибут LocalID

Атрибут LocalID задает идентификационную информацию, посылаемую партнеру в первой фазе IKE.

Синтаксис	LocalID = `IdentityEntry`
Значение	В структуре IdentityEntry допускается задание одного значения одному из идентификаторов типа `IPv4Address`, `FQDN`, `EMail`, `DistinguishedName`. При задании значения идентификатору `DistinguishedName` использование в строке Subject зарезервированного слова TEMPLATE недопустимо. При задании значения идентификатору `IPv4Address` использование диапазона IP-адресов недопустимо. Если значение задано зарезервированным словом USER_SPECIFIC_DATA, то в качестве идентификатора будет использовано соответствующее значение из локального сертификата. Если в сертификате соответствующее значение отсутствует, то ISAKMP-сессия будет прервана.
Значение по умолчанию	первый IP-адрес сетевого интерфейса, с которого отсылаются ISAKMP-пакеты партнеру.

Атрибут RemoteID

Атрибут RemoteID задает требования к идентификационной информации партнера.

Синтаксис RemoteID = `IdentityEntry`

Значение В структуре IdentityEntry допускается задание нескольких идентификаторов типа `IPv4Address`, `FQDN`, `EMail`, `DistinguishedName`.

Значение по умолчанию принимается любой ID партнера.

Атрибут LocalCredential

Атрибут LocalCredential задает требуемые параметры сертификата данного VPN-устройства.

Синтаксис LocalCredential = `CertDescription`

Значение по умолчанию требования отсутствуют. Используется любой локальный сертификат.

Атрибут RemoteCredential

Атрибут RemoteCredential задает требуемые параметры сертификата партнера по взаимодействию.

Синтаксис RemoteCredential* = `CertDescription`

Значение по умолчанию требования отсутствуют, допускается любой сертификат.

Атрибут AcceptCredentialFrom

Атрибут AcceptCredentialFrom задает требуемые параметры CA сертификата, удостоверяющего подлинность сертификата партнера.

Синтаксис AcceptCredentialFrom* = `CertDescription`

Значение по умолчанию используется любой из тех CA, которому мы доверяем.

Атрибут DoNotMapLocalIDToCert

Атрибут DoNotMapLocalIDToCert задает режим использования локального идентификатора при поиске локального сертификата.

Синтаксис DoNotMapLocalIDToCert = TRUE | FALSE

Значение TRUE – при поиске локального сертификата используются описания сертификатов, указанные в атрибуте LocalCredential. Значение атрибута LocalID игнорируется

FALSE – при поиске локального сертификата используется список CertDescription. Каждый элемент этого списка является объединением полей атрибутов LocalID и LocalCredential. Объединение строится по следующим правилам:

если LocalID задан зарезервированным словом `USER_SPECIFIC_DATA`, то используется CertDescription в том виде, как он задан в LocalCredential

Если значение LocalID не противоречит LocalCredential, оно является дополнительным критерием поиска сертификата.

Если значение LocalID противоречит LocalCredential, соединение не построится.

Значение по умолчанию FALSE

Атрибут DoNotMapRemoteIDToCert

Атрибут DoNotMapRemoteIDToCert задает режим использования идентификатора партнера при поиске его сертификата.

Синтаксис DoNotMapRemoteIDToCert = TRUE | FALSE

Значение TRUE – при поиске сертификата партнера используются описания сертификатов, указанные в атрибуте RemoteCredential, значение атрибута RemoteID игнорируется.

FALSE – при поиске сертификата партнера используется список CertDescription. Каждый элемент этого списка является объединением присланного идентификатора партнера и CertDescription из атрибута RemoteCredential. Правила объединения совпадают с ранее описанными правилами в атрибуте DoNotMapLocalIDToCert.

Значение по умолчанию FALSE.

Атрибут SendRequestMode

Атрибут SendRequestMode определяет логику отсылки запроса сертификата партнера.

Синтаксис SendRequestMode = AUTO | NEVER | ALWAYS

Значение AUTO – запрос высылается, если сертификат партнера не доступен локально или не может быть однозначно определено, каким сертификатом воспользуется партнер.

NEVER – запрос не высылается.

ALWAYS – запрос высылается всегда

Значение по умолчанию AUTO

Атрибут SendCertMode

Атрибут SendCertMode определяет логику отсылки локального сертифика в процессе первой фазы IKE на запрос партнера. В своем запросе партнер может указать, какому CA сертификату он доверяет. Если такой сертификат не найден, то он не отсылается.

Синтаксис SendCertMode = AUTO | NEVER | ALWAYS| CHAIN

Значение AUTO – автоматически определяется, когда необходима отсылка локального сертификата партнеру:

если партнер не прислал запроса, то сертификат не отсылается

если партнер прислал запрос и соответствующий сертификат был найден, то партнеру высылается либо сертификат, либо найденная цепочка сертификатов

если партнер прислал запрос и этот запрос не был удовлетворен, то сертификат не высылается.

NEVER – сертификат не высылается

ALWAYS –сертификат высылается всегда

CHAIN – сертификат высылается всегда, причем в составе с цепочкой доверительных CA:

Имеется ввиду цепочка сертификатов, построенная от локального сертификата до CA, который удовлетворяет описанию, присланному партнером в запросе. В общем случае это CA, удовлетворяющий запросу партнера, произвольное количество промежуточных CA и локальный сертификат.

Значение по умолчанию AUTO

Пример

```
AuthMethodGOSTSign auth_ca
(
    LocalID = IdentityEntry( DistinguishedName* = USER_SPECIFIC_DATA
)
    RemoteID = Identity_id2
    AcceptCredentialFrom*= ca
    SendRequestMode = ALWAYS
    SendCertMode = ALWAYS
)
```

Структура AuthMethodPreshared

Структура AuthMethodPreshared задает аутентификационную информацию при использовании предопределенных (Preshared) ключей.

Имя структуры	AuthMethodPreshared
Атрибуты	LocalID
	RemoteID
	SharedIKESecret

Атрибут LocalID

Атрибут LocalID задает идентификационную информацию данного VPN-устройства.

Синтаксис	LocalID = `IdentityEntry`
Значение	В структуре IdentityEntry допускается задание только одного идентификатора с одним значением. При задании значения атрибуту `IPv4Address` использование диапазона IP-адресов недопустимо. Использование зарезервированного слова USER_SPECIFIC_DATA не допускается.
Значение по умолчанию	локальный IP-адрес из IKE-пакета.

Атрибут RemoteID

Атрибут RemoteID задает требования к идентификационной информации партнера.

Синтаксис	RemoteID = `IdentityEntry`
Значение	В структуре IdentityEntry допускается задание нескольких идентификаторов разных типов.
Значение по умолчанию	принимается любой ID партнера.

Атрибут SharedIKESecret

Атрибут SharedIKESecret определяет ссылку на предопределенный секретный ключ.

В атрибуте указывается имя предопределенного (Preshared) ключа, хранимого в базе Продукта и импортированного утилитой `key_mgr import`.

Синтаксис	SharedIKESecret = СТРОКА
Значение	имя предопределенного (Preshared) ключа
Значение по умолчанию	не существует, атрибут обязательный.

Структура IdentityEntry

Структура IdentityEntry описывает идентификационную информацию. Варианты задания этой структуры приведены в описаниях структур `AuthMethodPreshared` и `AuthMethod{DSS|RSA|GOST}Sign`.

Имя структуры IdentityEntry

Атрибуты IPv4Address – IPv4 адрес

FQDN – FQDN хоста

EMail – EMail пользователя

DistinguishedName – DN в формате X509Subject

KeyID – идентификатор ключа

Если структура IdentityEntry используется определенным методом аутентификации, то атрибуты, не соответствующие данному методу, игнорируются. Атрибуты, используемые для определенных методов аутентификации:

`AuthMethodPreshared`

- `IPv4Address`
- `KeyID`

`AuthMethod{DSS|RSA|GOST}Sign`

- `IPv4Address`
- FQDN
- `EMail`
- `DistinguishedName`.

Атрибут IPv4Address

Атрибут IPv4Address задает описание идентификатора по указанным IP-адресам.

Синтаксис для данного VPN устройства:

IPv4Address = IP | USER_SPECIFIC_DATA

для партнера:

IPv4Address *= IP | IP..IP | IP/ЦЕЛОЕ32 | USER_SPECIFIC_DATA

Значения для данного VPN устройства:

IP – один IP-адрес

для партнера:

IP – список IP-адресов

IP..IP – список диапазонов IP-адресов

IP/ЦЕЛОЕ32 – список подсетей с IP-адресом и маской

Если задано значение `USER_SPECIFIC_DATA`, то берется первый IP-адрес из расширения Subject Alternative Name локального сертификата, используемого для подписи. Если IP-адрес в сертификате отсутствует, то соединение не создается.

Если заданы диапазоны IP-адресов либо подсети, то это означает, что принимается любой Identity типа IP-адрес, если значение IP,

присланное партнером в таком Identity, попадает в указанный диапазон, либо подсеть.

Значение по умолчанию используются другие атрибуты.

Атрибут FQDN

Атрибут FQDN (Fully Qualified Domain Name – полностью определенное доменное имя) задает описание идентификатора хоста по указанным DNS именам.

Синтаксис FQDN* = СТРОКА | USER_SPECIFIC_DATA

Значения строки вида "host.domain".

если задано значение USER_SPECIFIC_DATA, то при проверке/отсылке Identity используется поле **DNS** расширения Subject Alternative Name соответствующего сертификата, используемого соответственно для проверки/формирования подписи.

Значение по умолчанию используются другие атрибуты.

Атрибут EMail

Атрибут EMail задает описание идентификатора по указанным Email-адресам.

Синтаксис EMail* = СТРОКА | USER_SPECIFIC_DATA

Значения строки вида "user@host.domain". Шаблоны не допускаются.

если задано значение USER_SPECIFIC_DATA, то при проверке/отсылке Identity используется поле **EMail** расширения Subject Alternative Name сертификата, используемого соответственно для проверки/формирования подписи.

Значение по умолчанию используются другие атрибуты.

Атрибут DistinguishedName

Атрибут DistinguishedName задает описание идентификатора по указанным DN (уникальное имя в формате X509Subject.).

Синтаксис DistinguishedName* = `CertDescrition` | USER_SPECIFIC_DATA

Значения в каждой структуре CertDescription допускается использование только поля Subject

если задано значение USER_SPECIFIC_DATA, то при проверке/отсылке Identity используется полное описание раздела **Subject Name** сертификата, используемого соответственно для проверки/формирования подписи.

Значение по умолчанию используются другие атрибуты

Атрибут KeyID

Атрибут KeyID задает описание Identity по указанным идентификаторам Preshared ключей

Синтаксис KeyID* = СТРОКА

Значение строки, содержащие шестнадцатеричное представление идентификаторов ключей.

Для `AuthMethodPreshared`:

рекомендуется при составлении идентификатора ключа использовать шестнадцатеричное представление только печатных символов без пробела: Именно такое ограничение существует при формировании конфигурации IOS.

Значение по умолчанию используются другие атрибуты.

Пример

```
AuthMethodPreshared auth_key (
     RemoteID = IdentityEntry(
          IPv4Address *= 192.168.13.117, 192.168.13.118
     )
     SharedIKESecret = "cskey"
)
```

Структура CertDescription

Структура CertDescription используется для задания собственного идентификатора и идентификатора партнера, для задания характеристик локального и сертификата партнера.

Для задания СТРОКИ в атрибутах этой структуры смотрите формат DN в разделе "Формат задания DistinguishedName в LSP" в Приложении.

Имя структуры	CertDescription
Атрибуты	Subject
	AlternativeSubject
	Issuer
	AlternativeIssuer
	FingerprintMD5
	FingerprintSHA1
	SerialNumber

Атрибут Subject

Атрибут Subject задает значение/шаблон поля Subject сертификата.

Синтаксис	Subject* = TEMPLATE | COMPLETE, СТРОКА
Значение	TEMPLATE – флаг, при котором указанная строка представляет собой незаконченное значение поля Subject сертификата.При поиске и сравнении, поле Subject сертификата должно содержать указанную строку
	COMPLETE – флаг, при котором указанная строка представляет собой законченное значение поля Subject сертификата. При поиске и сравнении поле Subject сертификата должно совпадать с указанным множеством атрибутов и их значениями (с точностью до порядка указания в сертификате) в строке.
Предупреждение	DN в строке должен быть задан точно так же, как он задан в сертификате: необходимо строго соблюдать количество пробелов и регистр символов.
Значение по умолчанию	если задана строка, но опущен флаг TEMPLATE или COMPLETE , то по умолчанию он равен COMPLETE;
	если не задана строка, то поле Subject сертификата принимает любые значения.

Пример:

Допустимые варианты:

```
Subject* = TEMPLATE, "ou=eng"
Subject* = "ou=eng", TEMPLATE
Subject* = COMPLETE, "c=RU,o=co.,ou=eng,cn=engineer"
Subject* = "c=RU, o=co, ou=eng, cn=engineer"
```

Недопустимые варианты:

```
Subject *= TEMPLATE, "ou=eng", COMPLETE
Subject *= "ou=eng", "ou=qa"
```

Атрибут AlternativeSubject

Атрибут AlternativeSubject задает шаблон Alternative Subject Extension сертификата.

Синтаксис AlternativeSubject = СТРОКА

Значение по умолчанию любое значение Alternative Subject Extension сертификата.

Атрибут Issuer

Атрибут Issuer задает значение/шаблон поля Issuer сертификата.

Синтаксис Issuer* = TEMPLATE | COMPLETE, СТРОКА

Значение TEMPLATE – флаг, при котором указанная строка представляет собой незаконченное значение поля Issuer сертификата. При поиске и сравнении, поле Issuer сертификата должно содержать указанную строку.

COMPLETE – флаг, при котором указанная строка представляет собой законченное значение поля Issuer сертификата. При поиске и сравнении, поле Issuer сертификата должно совпадать с указанным множеством атрибутов и их значениями (с точностью до порядка указания в сертификате) в строке.

Предупреждение DN в строке должен быть задан точно также, как он задан в сертификате: необходимо строго соблюдать количество пробелов и регистр символов.

Значение по умолчанию если задана строка, но опущен флаг TEMPLATE или COMPLETE , то по умолчанию он равен COMPLETE;

если не задана строка, то поле Issuer сертификата принимает любые значения.

Атрибут AlternativeIssuer

Атрибут AlternativeIssuer задает шаблон Alternative Issuer Extension сертификата.

Синтаксис AlternativeIssuer = СТРОКА

Значение по умолчанию любое значение Alternative Issuer Extension сертификата.

Атрибут FingerprintMD5

Атрибут FingerprintMD5 задает значение хеш-функции алгоритма MD5 по бинарному представлению сертификата.

Синтаксис FingerprintMD5 = СТРОКА

Значение шестнадцатеричная запись значения хэш-функции, длина строки должна быть равна 32 символам.

Значение по умолчанию любое значение хэш-функции.

Атрибут FingerprintSHA1

Атрибут FingerprintSHA1 задает значение хеш-функции алгоритма SHA1 по бинарному представлению сертификата.

Синтаксис FingerprintSHA1 = СТРОКА

Значение шестнадцатеричная запись значения хэш-функции, длина строки должна быть равна 40 символам.

Значение по умолчанию любое значение хэш-функции.

Атрибут SerialNumber

Атрибут SerialNumber задает значение серийного номера сертификата.

Синтаксис SerialNumber = СТРОКА

Значение шестнадцатеричная запись серийного номера.

Значение по умолчанию любое значение серийного номера.

Пример

```
RemoteCredential* = CertDescription(
Issuer* = COMPLETE, "CN=S-Terra CenterCA, O=S-Terra, L=Moscow, C=RU"
Subject* = TEMPLATE, "CN=S-Terra, OU=QA"
AlternativeSubject = "EMAIL=inform@s-terra.com, DNS= tester.s-terra.com, IP =10.10.10.10"
SerialNumber = "567A99991E1F"
)
```


Структура FirewallParameters

Структура FirewallParameters описывает глобальные параметры межсетевого экрана. В конфигурации должна быть только одна структура данного типа. Этой структуре имя не присваивается.

Имя структуры FirewallParameters

Атрибуты
TCPSynSentTimeout
TCPSynRcvdTimeout
TCPFinTimeout
TCPClosedTimeout
TCPEstablishedTimeout
TCPHalfOpenMax
TCPHalfOpenLow
TCPSessionRateMax
TCPSessionRateLow
TCPSessionsMax
TCPStrictnessLevel

Атрибуты TCPSynSentTimeout, TCPSynRcvdTimeout, TCPFinTimeout, TCPClosedTimeout, TCPEstablishedTimeout

Атрибуты устанавливают время жизни записи о соединении. Межсетевой экран определяет состояние TCP-соединения для каждого из партнеров и, в зависимости от этого, выставляет время жизни записи о соединении. В таблице приведены стандартные названия для состояний TCP и соответствующие параметры, задающие время жизни.

Синтаксис Атрибут = ЦЕЛОЕ32

Значения Целое число из диапазона 1..65535

Значение по умолчанию см. таблицу.

Состояние	Параметр LSP	Параметр в drv_mgr (только для просмотра)	Значение по умолчанию (сек.)
CLOSED, LISTEN	TCPClosedTimeout	fw_tcp_closed_ttl	30
SYNSENT	TCPSynSentTimeout	fw_tcp_synsent_ttl	30
SYNRCVD	TCPSynRcvdTimeout	fw_tcp_synrcvd_ttl	30
ESTAB	TCPEstablishedTimeout	fw_tcp_estab_ttl	3600
FINWAIT-1, FINWAIT-2, CLOSING, TIMEWAIT, LASTACK, CLOSED	TCPFinTimeout	fw_tcp_fin_ttl	5

Значение TCPEstablishedTimeout может быть переопределено для конкретного правила фильтрации (см. Filter.ExtendedAction)

Атрибут TCPHalfOpenMax

Атрибут TCPHalfOpenMax задает максимальное разрешенное количество одновременно существующих полуоткрытых сеансов, по достижении которого Шлюз безопасности начинает их удаление.
При превышении данного предела новые соединения будут создаваться только за счет уничтожения полуоткрытых сеансов, созданных ранее. Таким образом, после превышения TCPHalfOpenMax полуоткрытые сеансы будет удаляться, пока их количество не достигнет значения, заданного атрибутом TCPHalfOpenLow. Далее вновь допускается увеличение количества полуоткрытых сеансов.

Синтаксис TCPHalfOpenMax = ЦЕЛОЕ32

Значения Целое число из диапазона 0..1000000

Значение по умолчанию 500

Атрибут TCPHalfOpenLow

Атрибут TCPHalfOpenLow задает количество одновременно существующих полуоткрытых сеансов, которое считается нормальным. В случае превышения максимального числа полуоткрытых сеансов, заданных атрибутом TCPHalfOpenMax, полуоткрытые сеансы будут уничтожаться до заданного предела.

Синтаксис TCPHalfOpenLow = ЦЕЛОЕ32

Значения Целое число из диапазона 0..1000000

Значение по умолчанию 400

Атрибут TCPSessionRateMax

Атрибут TCPSessionRateMax задает верхнюю границу на количество новых контекстов соединений, создаваемых за минуту. Если частота появления новых контекстов соединений достигнет TCPSessionRateMax, то Шлюз безопасности начнет их удаление до тех пор, пока частота появления новых контекстов соединений не уменьшится до величины, заданной атрибутом TCPSessionRateLow.

Синтаксис TCPSessionRateMax = ЦЕЛОЕ32

Значения Целое число из диапазона $0..2^{32}-1$

Значение по умолчанию 500 новых контекстов соединений в минуту.

Атрибут TCPSessionRateLow

Атрибут TCPSessionRateLow задает нижнюю границу на количество новых контекстов соединений, создаваемых за минуту, по достижении которой, Шлюз безопасности прекращает их удаление.

Синтаксис TCPSessionRateLow = ЦЕЛОЕ32

Значения Целое число из диапазона $0..2^{32}-1$

Значение по умолчанию 400 новых контекстов соединений в минуту.

Атрибут TCPSessionsMax

Атрибут TCPSessionsMax задает максимальное разрешенное количество TCP-соединений. При превышении данного предела новые TCP-соединения будут отвергаться.

Синтаксис TCPSessionsMax = ЦЕЛОЕ32

Значения Целое число из диапазона 0..1000000

Значение по умолчанию 65536.

Атрибут TCPStrictnessLevel

Атрибут TCPStrictnessLevel используется для задания уровня "жесткости" к различным ситуациям, которые воспринимаются firewall как ошибочные.

Синтаксис TCPStrictnessLevel = ЦЕЛОЕ32

Значения Целое число из диапазона 0..6

Значение по умолчанию 3.

В следующей таблице приведены основные отличия в поведении при различных значениях TCPStrictnessLevel. Показана зависимость выполнения таких действий как «уничтожение пакета» и «отказ в изменении состояния соединения» от уровня, заданного TCPStrictnessLevel и результата анализа заголовка TCP пакета.

Значение	Уничтожение пакета	Отказ в изменении состояния соединения[35]
0	Пакеты не уничтожаются firewall	При некорректном TCP заголовке (проверяется соответствие длины пакета, TCP заголовка, checksum)
1	При некорректном TCP заголовке	При некорректном TCP заголовке
2	При некорректном TCP заголовке	При некорректном TCP заголовке или при sequence, несоответствующем состоянию партнера
3	При некорректном TCP заголовке	При некорректном TCP заголовке, или при sequence, несоответствующем состоянию партнера, или при несоответствии флагов заголовка состоянию партнера
4	При некорректном TCP заголовке или при несоответствии флагов заголовка состоянию партнера	При некорректном TCP заголовке, или при sequence, несоответствующем состоянию партнера, или при несоответствии флагов заголовка состоянию партнера
5	При некорректном TCP заголовке, или при несоответствии флагов заголовка состоянию партнера, или при sequence, несоответствующему состоянию партнера	При некорректном TCP заголовке, или при sequence, несоответствующем состоянию партнера, или при несоответствии флагов заголовка состоянию партнера
6	При некорректном TCP заголовке, или при несоответствии флагов заголовка состоянию партнера, или при sequence, несоответствующему состоянию партнера, или при приеме SYN для установившегося соединения	При некорректном TCP заголовке, или при sequence, несоответствующем состоянию партнера, или при несоответствии флагов заголовка состоянию партнера, или при получении первым пакетом не SYN, или при приеме SYN для установившегося соединения

[35] Например, не пролонгируется существование записи о соединении.

Структура NetworkInterface

Структура NetworkInterface описывает логический сетевой интерфейс, который может соответствовать нескольким сетевым интерфейсам системы. В структуре описываются действия, которые должны быть выполнены с пакетом, при его прохождении через этот интерфейс.

Структуре NetworkInterface имя не присваивается. В конфигурации допускается описание нескольких экземпляров данной структуры, которые будут отличаться значением поля LogicalName.

Имя структуры	NetworkInterface
Атрибуты	LogicalName InputFilter OutputFilter InputClassification OutputClassification IPsecPolicy

Атрибут LogicalName

Атрибут LogicalName задает логическое имя интерфейса. Это имя указано как алиас в файле etc/ifaliases.cf, описывающем системные интерфейсы. Если алиас не соответствует каким-либо сетевым интерфейсам на момент загрузки LSP, то будет выдано предупреждение.

Синтаксис	LogicalName = СТРОКА
Значение	логическое имя интерфейса
Значение по умолчанию	Значение "default" в файле ifaliases.cf определяется как "*", что интерпретируется как "остальные сетевые интерфейсы".

Атрибут InputFilter

Атрибут InputFilter задает правила как stateless (пакетной), так и stateful (контекстной) фильтрации для входящих пакетов через данный интерфейс. Пакет, не попавший ни под одно из заданных правил фильтрации, удаляется. Если фильтры на интерфейсе не заданы, то пропускается любой пакет.

Синтаксис	InputFilter = `FilterChain`
Значение по умолчанию	входящие пакеты не фильтруются.

Атрибут OutputFilter

Атрибут OutputFilter задает правила как stateless (пакетной), так и stateful (контекстной) фильтрации для исходящих пакетов через данный интерфейс. Пакет, не попавший ни под одно из заданных правил фильтрации, удаляется. Если фильтры на интерфейсе не заданы, то пропускается любой пакет.

Синтаксис	OutputFilter = `FilterChain`
Значение по умолчанию	исходящие пакеты не фильтруются.

Атрибут InputClassification

Атрибут InputClassification задает правила классификации и выставления значения поля TOS в IP-заголовке входящих пакетов через данный интерфейс. Классификация и маркирование производится на открытых пакетах, то есть после IPsec декапсуляции. Входящий пакет, не попавший ни под одно из заданных правил, не классифицируется и пропускается в неизменном виде.

Синтаксис InputClassification = `FilterChain`

Значение по умолчанию классификация и маркирование пакетов не производится.

Атрибут OutputClassification

Атрибут OutputClassification задает правила классификации и выставления значения поля TOS в IP-заголовке исходящих пакетов через данный интерфейс. Классификация и маркирование производится на открытых пакетах, то есть до IPsec инкапсуляции. В случае туннелирования значение поля TOS копируется из внутреннего IP-заголовка во внешний. Исходящий пакет, не попавший ни под одно из заданных правил, не классифицируется и пропускается в неизменном виде.

Синтаксис OutputClassification = `FilterChain`

Значение по умолчанию классификация и маркирование пакетов не производится.

Атрибут IPsecPolicy

Атрибут IPsecPolicy задает правила защиты пакетов с помощью IPsec. В фильтрах описывается исходящий трафик, но фильтрация производится симметрично для входящего и исходящего трафика. То есть при обработке входящего трафика на сетевом интерфейсе SourceIP, SourcePort сравнивается с соответствующими полями заголовков пакета destination IP address, destination UDP port, destination TCP port, а DestinationIP и DestinationPort сравниваются с полями заголовков пакета source IP address, source UDP port, source TCP port. При обработке исходящего трафика на сетевом интерфейсе понятия source и destination в конфигурации соответствуют понятиям source и destination в пакете.

Синтаксис IPsecPolicy = `FilterChain`

Значение по умолчанию IPsec не применяется.

Структура FilterChain

Структура FilterChain задает список правил пакетной и контекстной фильтрации («цепочка» правил). Этой структуре может быть присвоено имя.

Имя структуры FilterChain

Атрибуты Filters

Атрибут Filters

Атрибут Filters задает список правил фильтрации с условиями срабатывания каждого правила. Порядок обработки каждого правила соответствует порядку перечисления фильтров в LSP за исключением ситуаций, когда используются переходы (см. атрибут Action параметр STRING).

Синтаксис Filters *= `Filter`

Пример

```
FilterChain IPsecPolicy:DMAP (
    Filters = Filter (
        ProtocolID = 17
        SourcePort = 500, 4500
        Action = PASS
        PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
    ),
    Filter (
        SourceIP = 192.168.2.0/24
        DestinationIP = 192.168.2.240/29
        Action = PASS
        ExtendedAction = ipsec< sa = IPsecAction:DMAP:1:dmap:1 >
        LogEventID = "IPsec:Protect:DMAP:1:dmap:1:client"
    )
)
```


Структура Filter

Структура Filter задает правило пакетной и контекстной фильтрации.

Имя структуры Filter

Атрибуты
SourceIP
DestinationIP
ProtocolID
SourcePort
DestinationPort
PacketType
Action
ExtendedAction
LogEventID
Schedule
Log
Label

Атрибут SourceIP

Атрибут SourceIP задает возможные значения поля Source Address в IPv4-заголовке пакета[36].

Синтаксис SourceIP *= `AddressPool`, IP, IP/ЦЕЛОЕ32

Значение AddressPool – множество адресов IKECFG пула

IP-адрес

IP/ЦЕЛОЕ32 – IP-адрес с маской

Значение по умолчанию допускается любое значение поля Source Address в IPv4-заголовке пакета.

Атрибут DestinationIP

Атрибут DestinationIP задает возможные значения поля Destination Address в IPv4-заголовке пакета[37].

Синтаксис SourceIP *= `AddressPool` | IP | IP..IP | IP/ЦЕЛОЕ32

Значение AddressPool – множество адресов IKECFG пула

IP-адрес

IP..IP – диапазон IP-адресов

IP/ЦЕЛОЕ32 – IP-адрес с маской

Значение по умолчанию допускается любое значение поля Destination Address в IPv4-заголовке пакета.

[36] Для цепочки фильтров IPsec правил используется особая интерпретация. См. описание IPsecPolicy.

[37] Для цепочки фильтров IPsec правил используется особая интерпретация. См. описание IPsecPolicy.

Атрибут ProtocolID

Атрибут ProtocolID задает возможные значения поля Protocol в IPv4-заголовке.

Синтаксис ProtocolID *= ЦЕЛОЕ32 | ЦЕЛОЕ32..ЦЕЛОЕ32

Значение целое число из диапазона 0..255.

Значение 0 означает все сетевые протоколы.

Значение по умолчанию любое значение поля Protocol в IPv4-заголовке.

Атрибут SourcePort

Атрибут SourcePort описывает список идентификаторов портов для указанных протоколов объекта[38]. Если значение ProtocolID не TCP и не UDP, то данный фильтр не совпадет, поиск фильтров продолжится. Если в ProtocolID заданы оба протокола – TCP и UDP, то указанные порты допускаются в сочетании с любым из двух протоколов.

Синтаксис SourcePort *= ЦЕЛОЕ32 | ЦЕЛОЕ32..ЦЕЛОЕ32

Значение целое число из диапазона 0..65535.

Значение 0 означает все порты для указанных протоколов.

Значение по умолчанию допускается любое значение поля Source Port в UDP либо TCP заголовке пакета.

Атрибут DestinationPort

Атрибут DestinationPort описывает список идентификаторов портов для указанных протоколов объекта[39]. Если значение ProtocolID не TCP и не UDP, то данный фильтр не совпадет, поиск фильтров продолжится. Если в ProtocolID заданы оба протокола – TCP и UDP, то указанные порты допускаются в сочетании с любым из двух протоколов.

Синтаксис DestinationPort *= ЦЕЛОЕ32 | ЦЕЛОЕ32..ЦЕЛОЕ32

Значение целое число из диапазона 0..65535.

Значение 0 означает все порты для указанных протоколов.

Значение по умолчанию допускается любое значение поля Destination Port в UDP либо TCP заголовке.

Атрибут PacketType

Атрибут PacketType задает список типов пакетов, для которых данное правило может сработать.

Синтаксис PacketType *= ЦЕЛОЕ32

Значение TRANSIT – транзитные пакеты, которые не предназначены для данного хоста и не созданы данным хостом.

LOCAL_BROADCAST – broadcast в локальной подсети (т.е. без учета универсальных broadcast 255.255.255.255, 0.0.0.0). Данное значение используется только для входящих пакетов.

[38] Для цепочки фильтров IPsec правил используется особая интерпретация. См. описание IPsecPolicy.

[39] Для цепочки фильтров IPsec правил используется особая интерпретация. См. описание IPsecPolicy.

LOCAL_UNICAST – обычные пакеты, принятые/отправленные хостом, на котором загружена LSP.

LOCAL_MISDICRECTED – пакеты, принятые/отправленные с интерфейса, на котором адрес получателя/отправителя не зарегистрирован.

Значение по умолчанию любой тип пакетов.

Атрибут Action

Атрибут Action задает действие, которое должно быть применено к пакету, при выполнении условий срабатывания правила.

Синтаксис Action = PASS | DROP | STRING

Значение

PASS – прекращается поиск правил фильтрации, выполняется действие, описанное в `ExtendedAction`. Если `ExtendedAction` отсутствует, пакет пропускается для дальнейшей обработки.

DROP – прекращается поиск правил фильтрации, не выполняется действие, описанное в `ExtendedAction`, пакет уничтожается.

STRING – в кавычках указывается строчка и в случае срабатывания данного правила должен продолжиться поиск правил, начиная с того, у которого значение атрибута `Label` совпадает с указанной здесь строчкой. Правило фильтрации, на который происходит переход, должно присутствовать в том же списке правил (FilterChain) и располагаться в списке после фильтра, откуда происходит переход.

Значение по умолчанию PASS.

Атрибут ExtendedAction

Атрибут ExtendedAction задает дополнительные условия для срабатывания правила и/или дополнительные действия, которые должны быть применены при выполнении условий срабатывания правила. Условия (действия) задаются в виде синтаксической конструкции "процедура". То есть указывается имя и именованные параметры в угловых скобках.

Синтаксис

ExtendedAction = процедура

ExtendedAction = inspect_tcp <…>

ExtendedAction = inspect_ftp <…>

ExtendedAction = tcp_flags <…>

ExtendedAction = classify_mark <...>

ExtendedAction = ipsec <...>

ExtendedAction = bit_check <…>

Значение

inspect_tcp – отслеживает состояние TCP-соединения, делает некоторые проверки на корректность заголовка, меняет время жизни записи о соединении в соответствии с текущим состоянием соединения. Для пропуска пакетов в обе стороны, добавляются дополнительные правила фильтрации во входящую и исходящую цепочки правил интерфейса, на котором сработала процедура tcp. Дополнительные правила удаляются вместе с записью о соединении.

Для совместимости с IOS CBAC на остальные интерфейсы, где присутствуют цепочки фильтрации, добавляются правила для пропуска пакетов по данному соединению. При этом обновление записи происходит только при обработке пакета на том интерфейсе, где создался контекст.

inspect_ftp – дополнительно отслеживает некоторые команды FTP, создает правила для пропуска соединения для данных FTP, определяет и блокирует некоторые подозрительные команды, которые могут являться атакой на FTP сервер.

Параметры для inspect_tcp и inspect_ftp

Имя параметра	Тип	Значения	По умолчанию
flags	список значений ЦЕЛОЕ32	AUDIT, NOALERT	включены предупреждения, отключен аудит
timeout	ЦЕЛОЕ32		берется из FirewallParameters.TCPEstablishedTimeout

AUDIT – формировать сообщения при закрытии состояния со статистической информацией.

NOALERT – не формировать сообщения о потенциальных атаках (попытках взлома).

timeout – время хранения информация о неактивном соединении, этот параметр переопределяет время жизни установившегося соединения.

tcp_flags – дополнительная фильтрация пакетов по флагам TCP-заголовка, без сохранения какой-либо информации о соединении. Правило, в котором присутствует tcp_flags, считается подходящим, только если протокол TCP и флаги TCP-заголовка пакета соответствуют заданным параметрам.

Параметры для tcp_flags

Имя параметра	Тип	Значения	По умолчанию
set	список значений ЦЕЛОЕ32	SYN, FIN, ACK, PSH, RST, URG	нет требований
clear	список значений ЦЕЛОЕ32	SYN, FIN, ACK, PSH, RST, URG	нет требований
any_set	список значений ЦЕЛОЕ32	SYN, FIN, ACK, PSH, RST, URG	нет требований
any_clear	список значений ЦЕЛОЕ32	SYN, FIN, ACK, PSH, RST, URG	нет требований

set – флаги, которые обязательно должны быть выставлены.

clear – флаги, которые должны быть сброшены.

any_set – любой из указанных флагов может быть выставлен для совпадения.

any_clear – любой из указанных флагов может быть сброшен для совпадения.

Флаги задаются константами, значение которых соответствует кодированию в заголовке TCP. Если флаги заданы списком, значения объединяются операцией "логическое или". Можно задать несколько флагов сразу одним числом. Например, следующие записи эквивалентны:

```
14h
ACK, RST
4, 10h
4, 4, 4, 16
```


Флаги, выставленные в TCP-заголовке пакета, должны совпадать с флагами, заданными по set, clear и одному из any_set или any_clear.

classify_mark – проверяет и/или выставляет TOS-байт в IP-пакетах.

Параметры для classify_mark

Имя параметра	Тип	Значения	По умолчанию
tos_set	ЦЕЛОЕ32	0-255	0
tos_set_mask	ЦЕЛОЕ32	0-255	0, значение TOS-байта не меняется
tos_match	список значений ЦЕЛОЕ32	0-255	байт TOS не влияет на совпадения фильтра
tos_match_mask	ЦЕЛОЕ32	0-255	маска должна быть отлична от нуля, если список tos_match не пуст

tos_set, tos_set_mask – если маска не нулевая, то в TOS-байте заголовка пакета будут выставлены биты, соответствующие tos_set.

tos_match, tos_match_mask – задают дополнительные ограничения на совпадение фильтра. Фильтр будет считаться подходящим только в том случае, если одно из значений tos_match совпадет со значением TOS-байта пакета в битах, ограниченных tos_match_mask.

ipsec – указывает, что пакет должен быть обработан с помощью IPsec.

Параметры для ipsec

Имя параметра	Тип	Значения	По умолчанию
sa	список IPsecAction		обязательное поле
fallback_action	ЦЕЛОЕ32	REQUEST_SA,PASS,DROP	REQUEST_SA
sa_requests_max	ЦЕЛОЕ32		8
packets_waiting_max	ЦЕЛОЕ32		8

sa – список IPsec-правил, которые могут быть использованы для создания SA, прикрепленных к данному правилу. Инициатор всегда использует первое IpsecAction.

fallback_action – действие, выполняемое в случае отсутствия SA. По умолчанию – REQUEST_SA.

REQUEST_SA – посылать запрос в демон, ставить пакет в очередь

PASS – пропускать пакет без IPsec-обработки

DROP – уничтожать пакет.

Входящие пакеты, попадающие на действие с флагом REQUEST_SA, также будут уничтожены.

sa_requests_max – максимальное количество неотвеченных запросов на создание SA bundle, отправленных в демон по данному правилу.
Есть и общее ограничение на количество запросов – значение можно задать через drv_mgr – параметр ipsec_breq_max (значение по умолчанию – 1000). Текущее количество запросов доступно через drv_mgr – параметр ipsec_breq_count.

packets_waiting_max – размер очереди в пакетах, ожидающих приход SA bundle.

bit_check – задает дополнительную фильтрацию по любым значимым полям IP-заголовка пакета и полям данных. Поля задаются в виде диапазона битов.

Параметры для bit_check

Имя параметра	Тип	Значения	По умолчанию
origin		IP_HDR – смещение считается от начала пакета (начала IP-заголовка) или IP_DATA – смещение считается начиная с первого байта данных после IP-заголовка.	IP_HDR
bit-range	диапазон битов		
operation		EQUAL (равно), LESS (меньше), GREATER (больше)	EQUAL
value	неотрицательное число		

origin – начальное смещение для bit-range.

bit-range – диапазон битов, которые проверяется. Задается в формате bit_offset1..bit_offset2, где bit_offset – неотрицательные смещения в пакете относительно origin в битах. Допускается, чтобы bit_offset1 и bit_offset2 совпадали, но второе смещение должно быть не меньше первого. Диапазон должен закрывать не более 32 бит, следовательно, разница bit_offset2-bit_offset1 не должна быть больше 31.

operation – операция сравнения. Операция выполняется над значением, извлеченным из пакета по адресу bit-range и значением value. Данные пакета интерпретируются как неотрицательное число, bit_offset1 является старшим битом числа, bit_offset2 – младшим битом числа.

value – значение, с которым сравниваются данные пакета.

Если описано несколько операций сравнения, то они выполняются последовательно. Если на какой-то из операций условия не совпали, пакет считается неподходящим под условия bit_check. Так, если необходимо, можно проверить длину IP-пакета, а потом производить сравнение данных за пределами IP-заголовка.

bit_check влияет именно на совпадение фильтра, а не приводит к каким-то дополнительным действиям, если совпадение обнаружено. Таким образом, поля Action, Log интерпретируются после проверки bit_check.

Если смещение bit_offset2 выходит за пределы пакета, пакет будет уничтожен.

Пакеты, подвергаемые проверке bit_check, не проходят сборку (IP reassembly). Но если важно, чтобы пакет не был собран до выполнения bit_check, необходимо помещать фильтры с bit_check вначало цепочки фильтров – другие фильтры могут вызывать сборку пакетов (например, фильтрация по TCP или UDP портам). Кроме того, действия inspect_tcp или inspect_ftp могут привести к сборке пакета, даже если фильтры с этими действиями стоят в цепочке позже, чем bit_check.

Пример

В первом правиле задано, не пропускать пакеты, у которых длина заголовка пакета больше 5 – 4 битовое поле (4..7) Header length имеет значение больше 5.

Во втором правиле указано: уничтожать пакеты протокола 17, у которых номер порта "Destionation Port" свыше 300, а значение "Destination IP" 7.7.7.212.

Filter (

```
        ExtendedAction = bit_check[[4..7, GREATER, 5]]
        Action = DROP
        LogEventID = "\"options in IP header\""
    ), Filter (
        ProtocolID = 17
        ExtendedAction = bit_check [[128..159, 070707D4h],[IP_DATA,16..31,LESS,300]]
        LogEventID = "\"special packet\""
        Action = DROP
    )
```

Допустимые значения ExtendedAction для разных применений FilterChain приведены в нижеследующей таблице.

	inspect_tcp inspect_ftp	tcp_flags	classify_ mark	bit_check	ipsec
NetworkInterface.InputFilter NetworkInterface.OutputFilter	+	+	+	+	–
IPsecAction.InputFilter IPsecAction.OutputFilter	–	+	+	+	–
NetworkInterface.InputClassification NetworkInterface.OutputClassification	–	+	+	+	–
NetworkInterface.IPsecPolicy	–	–	–	–	+

Если ExtendedAction не соответствует применению FilterChain, выдается ошибка разбора конфигурации.

Значение по умолчанию Отсутствуют специальные действия над пакетом.

Атрибут LogEventID

Атрибут LogEventID задает идентификатор, который передается в сообщения аудита, связанные с данным фильтром. При наличии LogEventID сообщения о подпадании пакета под фильтр отправляются в журнал аудита.

Синтаксис LogEventID = СТРОКА

Значение фактически LogEventID можно считать именем фильтра, но требования на уникальность отсутствуют.

Значение по умолчанию неименованное правило.

Атрибут Schedule

Атрибут Schedule задает временные диапазоны, в которые данный фильтр активен. В другое время фильтр не активен и не учитывается при фильтрации пакетов.

Деактивация фильтра ExtendedAction = inspect_* приводит к прекращению отслеживания соединений по данному правилу и уничтожению динамически созданных фильтров.

Нельзя указывать временные диапазоны для фильтров, привязанных к NetworkInterface.IPsecPolicy.

Синтаксис Schedule = `Schedule`

Значение по умолчанию нет ограничений по времени действия фильтра.

Атрибут Log

Атрибут Log включает/выключает генерацию данных аудита по данному фильтру. Если генерация данных аудита включена и пакет подпадает под фильтр, то в журнал аудита будет передано об этом сообщение.

Синтаксис Log = TRUE | FALSE

Значение TRUE – включает генерацию данных аудита по данному фильтру

FALSE – выключает аудит по данному фильтру.

Значение по умолчанию FALSE.

Атрибут Label

Атрибут Label задает метку, которая при совпадении со строкой в атрибуте Action другого правила, говорит о том, что с данного правила можно продолжить поиск правил (см. атрибут Action параметр STRING).

Синтаксис Label = СТРОКА

Значение по умолчанию Метка отсутствует.

Структура Schedule

Структура Schedule описывает график активности правил.

Структура может быть именована, что позволяет делать ссылки из нескольких правил на один график.

Имя структуры Schedule

Атрибуты Periods

Атрибут Periods

Атрибут Periods задает список временных интервалов. Если текущее время попадает в заданный интервал, то правило, для которого задан интервал, в данный момент считается активным.

Если в списке есть пересекающиеся интервалы с противоречащими действиями (`Period.Action`), то используется интервал, который раньше в списке.

Синтаксис Periods *= `Period`

Значение по умолчанию ограничение по времени не применяется.

Структура Period

Структура Period описывает временной диапазон – периодический или абсолютный.

Если атрибуты Start или End содержит абсолютную дату (тип ДАТА представляется тремя целыми числами без знака, разделенными символом '/' – число/месяц/год), то интервал считается абсолютным, иначе он периодический. Для абсолютных интервалов допускается только указание абсолютной даты и времени. Буквенные обозначения дней недели и месяцев запрещены.

Время соответствует локальному времени, в соответствии с настройками операционной системы.

Интервалы отслеживаются с максимальным опозданием в 1 минуту, но в случае крайней загруженности ОС (т.е. невозможности выполнения приложений в течение длительного времени), отслеживание графиков может задерживаться более 1 минуты.

Имя структуры Period

Атрибуты Start
End
Action

Атрибут Start

Атрибут Start задает начало временного интервала.

Синтаксис Start *= ЦЕЛОЕ32, ДАТА, ВРЕМЯ

Значение SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY,

JANUARY, FEBRUARY, MARCH, APRIL, MAY, JUNE, JULY, AUGUST, SEPTEMBER, OCTOBER, NOVEMBER, DECEMBER,

дата, время, день месяца (1..31), END_OF_MONTH (последний день месяца (для релиза 14101 это значение недоступно)).

Периодические интервалы

Для периодических интервалов атрибут Start является обязательным и определяющим для указания временного интервала.

Действует следующий порядок:
если в Start указан месяц, то периодичность год,
если в Start указан день месяца, периодичность – месяц,
если в Start указан день недели, периодичность – неделя,
если в Start указано только время, период – день.

Значение, указанное в Start, может быть больше значения, указанного в End. При этом интервал инвертируется – End переносится на следующий год, месяц, неделю или день в зависимости от периодичности.

Допускается указание нескольких значений, например, месяца и дня месяца. Но делать это надо с осторожностью. Если, например, указанного числа в месяце нет, то период будет пропущен.
День недели нельзя указывать вместе с месяцем или числом одновременно. Остальные комбинации допускаются.

Действуют следующие правила дополнения:
Если время не указано, то берется начало дня (00:00).
Если месяц указан, но не указано число, то берется первое число.

Значение по умолчанию для абсолютных интервалов – начало летоисчисления.
Для периодических интервалов поле обязательно.

Атрибут End

Атрибут End задает конец временного интервала.

Синтаксис End* = ЦЕЛОЕ32, ДАТА, ВРЕМЯ

Значение SUNDAY, MONDAY, TUESDAY, WEDNESDAY, THURSDAY, FRIDAY, SATURDAY,

JANUARY, FEBRUARY, MARCH, APRIL, MAY, JUNE, JULY, AUGUST, SEPTEMBER, OCTOBER, NOVEMBER, DECEMBER,

дата, время, день месяца (1..31), END_OF_MONTH (последний день месяца (для релиза 14101 это значение недоступно)).
Если указано время, то включается последняя минута. Так прекращение интервала для End = 14:20 будет не ранее 14:21.

Абсолютные интервалы

Если не указана дата, а только время, то дата для End принимается равной дате для Start.

Периодические интервалы

Действуют следующие правила дополнения:
если время не указано, то берется конец дня (23:59),
если месяц не указан, но указан в Start – период оканчивается в месяц с именем, указанным в Start,
если день месяца не указан, а в Start указан день месяца или месяц – период оканчивается в последний день месяца (в релизе 14101 надо явно указывать число в End, если в Start указан месяц или число),
если не указан день недели, но день недели есть в Start, период оканчивается в день недели, указанный в Start[40].

Действуют следующие ограничения:
день недели нельзя указывать вместе с месяцем или числом одновременно,
нельзя указывать день месяца больше 28, если месяц не задан явно или месяц – февраль[41],
нельзя указывать величины большего порядка, чем в Start – т.е. если в Start не указан месяц, то в End нельзя указать месяц.

[40] Если End указывает на более раннее время дня, чем Start, то интервал будет длиться до соответствующего дня следующей недели.

[41] Допустимо указывать 29 февраля, как отдельный день – Start и End оба указывают на 29 февраля. В этом случае период будет активен один день за 4 года.

Значение по умолчанию для абсолютных интервалов, отсутствие End считается отсутствием ограничения по времени. Причем если End отсутствует, Start обязательно должен быть указан.
Для периодических интервалов отсутствие End интерпретируется как конец дня, если Start не содержит указание месяца и/или числа. Если в Start указан месяц и/или число, End выставляется на конец месяца.

Примеры абсолютных интервалов:

Period a (Start = 23/12/2009 End = 8/9/2016, 22:30)

Period b (End = 08/09/ 2007, 2:30)

Period c (Start = 2:00, 5 /6/15)

Примеры периодических интервалов:

Period a (Start = 2, JANUARY End = 10)	# со второго по десятое января каждого года
Period b (Start = 12:00 End = 14:00)	# каждый день с 12 до двух дня
Period c (Start = 10, 10:00 End = 14:00)	# с 10 числа каждого месяца до 14 часов #последнего дня месяца
Period d (Start = MONDAY End = FRIDAY, 17:00)	# с понедельника до 17:00 пятницы каждую #неделю
Period e (Start = APRIL, 1, 15:00 End = APRIL, 1, 14:00)	# весь год кроме 1 часа 1 #апреля
Period f (Start = MONDAY, 18:30 End = 17:30)	# с понедельника 18:30 по следующий #понедельник 17:30

Атрибут Action

Атрибут Action задает активность правила в указанный период.

Синтаксис Action = ENABLE | DISABLE

Значение ENABLE – временной интервал считается интервалом активности для правила фильтрации;

DISABLE – в указанный временной интервал правило неактивно и не учитывается при фильтрации пакетов.

Значение по умолчанию ENABLE.

Приложение

Формат задания DistinguishedName (GeneralNames) в LSP

Текстовое представление DN

Текстовое представление DistinguishedName (GeneralNames), далее просто имени, задается в соответствии с RFC2253:

```
distinguishedName = [name]; may be empty string

name  name-component *("," name-component)

name-component = attributeTypeAndValue *("+" attributeTypeAndValue)

attributeTypeAndValue = attributeType "=" attributeValue

attributeType = (ALPHA 1*keychar) / oid
keychar = ALPHA / DIGIT / "-"

oid = 1*DIGIT *("." 1*DIGIT)

attributeValue = string

string = *( stringchar / pair )
         / "#" hexstring
         / QUOTATION *( quotechar / pair ) QUOTATION; only from v2

quotechar = <any character except "\" or QUOTATION >

special = "," / "=" / "+" / "<" /  ">" / "#" / ";"

pair = "\" ( special / "\" / QUOTATION / hexpair )
stringchar =<any character except one of special, "\" or QUOTATION>

hexstring = 1*hexpair
hexpair = hexchar hexchar

hexchar = DIGIT / "A" / "B" / "C" / "D" / "E" / "F"
          / "a" / "b" / "c" / "d" / "e" / "f"
```



```
ALPHA = <any ASCII alphabetic character>;(decimal 65-90 and 97-122)
DIGIT = <any ASCII decimal digit>       ;(decimal 48-57)
QUOTATION = <the ASCII double quotation mark character '"' decimal
34>
```

Дополнения и отступления от RFC2253

Имеются следующие дополнения и отступления от RFC2253:

- символ "/" является разделителем компонент имени, т.е. допустим следующий синтаксис:

  ```
  name = name-component *("/" name-component)
  ```

  - для того, чтобы использовать этот символ как значащий, его необходимо проэскейпить.
- распознаются следующие сокращения типов атрибутов (attributeType) DistinguishedName:

X.500 Attribute Type	Сокращение
countryName	C
stateName	ST
localityName	L
organizationName	O
organizationalUnitName	OU
commonName	CN
title	T
surname	SN
givenName	GN
initials	I
streetAddress	STREET
nameQualifier	NQ
generationQualifier	GQ
userid	UID
domainComponent	DC

  - регистр, в котором записано сокращение, не имеет значения.
- Строковое задание GeneralNames сведено к синтаксису, описанному в RFC2253. Распознаются следующие сокращения типов атрибутов имени GeneralNames:

Тип атрибута	Сокращение
otherName	OTHERNAME
rfc822Name	EMAIL
dNSName	DNS
directoryName	DN
uniformResourceIdentifier	URI
iPAddress	IP
registeredID	RID

  - регистр, в котором записано сокращение, не имеет значения
  - задание атрибутов `x400Address` и `ediPartyName` в строковом представлении не поддерживается.

- Согласно RFC2253 символы ′"′(кавычки) и ′\′(back-slash) являются служебными. Согласно описанию Терминального символа СТРОКА, при задании любого строкового значения в LSP указанные символы так же используются как служебные. Поэтому:
  - каждая отдельно стоящая кавычка в строковом представлении должна быть дополнена слева символом '\' в LSP
  - каждое сочетание '\"' в строковом представлении должно быть дополнено слева '\\' в LSP.

Примеры

Имя в сертификате	Строковое представление	В LSP
`O=Sergey, Danila and company`	`O=Sergey\, Danila and company`	`Subject="O=Sergey\, Danila and company"`
`O=JSC "Horns and hoofs"`	`O=JSC \"Horns and hoofs\"`	`Subject="O=JSC \\\"Horns and hoofs\\\""`
`CN=Device#4`	`CN="Device#4"`	`Subject="CN=\"Device#4\""`

Обработка пакетов – ретрансмиссии

Используемый механизм IKE-ретрансмиссий находится в общей концепции, согласно которой инициатор, исходя из наличия собственных ресурсов, проявляет настойчивость и добивается чего-то от ответчика, а ответчик, во первых, не доверяет инициатору насколько это возможно, во-вторых, по-максимуму бережет собственные ресурсы.

Инициатор, в большинстве случаев, являясь активной стороной, посылает очередной пакет IKE-обмена и затем перепосылает его (в соответствии с настройками ретрансмиссий – атрибуты SendRetries, RetryTimeBase и RetryTimeMax) до тех пор, пока не получит ответный пакет от ответчика.

Таким образом, инициатор выполняет работу за двоих:

- если исходящий от инициатора пакет не дошел до ответчика, то ответчик его не обработает и, соответственно, никак не ответит инициатору. Но исходящий пакет инициатором может быть перепослан (возможно, с n-ой попытки), ответчик его получит, обработает и отошлёт ответ
- если же проблема возникла на обратном пути (т.е. пакет от ответчика потерялся на пути к инициатору), то для инициатора эта ситуация детектируется точно так же, как и первая – то есть инициатор ответного пакета ждал, но за отведенный timeout так и не дождался. Тогда инициатор перепосылает свой последний исходящий пакет, ответчик снова его получает, распознает его как совпадающий с последним пакетом от инициатора, т.е. ретрансмиссию, и в ответ перепосылает свой последний пакет.

События для перепосылки:

- для стороны, выполняющей активную роль в ретрансмиссиях, событием для перепосылки своего последнего пакета является таймер и отсутствие ответа от партнера
- для пассивной стороны событием для перепосылки своего последнего пакета является получение ретрансмиссии от партнера.

В сценариях IKE, в которых ответчик обрабатывает последний пакет (Aggressive Mode и Quick Mode без поддержки Commit Bit), ответчик становится активной стороной при ожидании

последнего пакета обмена. В этих случаях инициатор уже не может выполнять активную роль, так как он в любом случае по сценарию не получает ответный пакет.

Пример настройки RADIUS-клиента

```
#.....

AAASettings (
  RadiusServer = 10.0.8.187               # no default!
  Secret = <secret_name_in_keydb>         # no default!
  NonInteractiveUserPassword = <passwd_name_in_keydb>   # password
for non-interactive xauth
  SendAccounting = TRUE
  ResponseTimeout = 3                     # in seconds
  Retries = 4

)
#.....

IKERule name (
#.....
  IKECFGPool = pool_1   # use local IKECFG server.

  XAuthServerEnabled = [FALSE] | TRUE    # demand IKE XAuth
authentication type

  AAAUserName = { [<NONE>] |    # no XAuth: common; XAuth: send
SET(OK)
                  INTERACTIVE | # no XAuth: <not allowed>; XAuth:
request user/password
    # Below: use NonInteractivePassword to access RADIUS-server;
XAuth: just send SET(OK|FAIL)
                  IKE_ID             |  # use printable IKE Identity
as user
                  CERT_SUBJ_CN       |  # use generic CN field from
partner's certificate
                  CERT_SUBJ_OU       |  # use generic OU field from
partner's certificate
                  CERT_ALTSUBJ_EMAIL |  # use generic EMail field
from partner's certificate
                  CERT_ALTSUBJ_DNS }    # use generic DNS field from
partner's certificate
)
```