ООО «С-Терра СиЭсПи»
124498, г. Москва, Зеленоград, Георгиевский проспект,
дом 5, помещение I, комната 33
Телефон: +7 (499) 940 9061
Факс: +7 (499) 940 9061
Эл.почта: information@s-terra.com
Сайт: http://www.s-terra.com

Программный комплекс CSP VPN Gate. Версия 3.11

Руководство администратора

Создание конфигурационного файла

РЛКЕ.00005-02 90 03

25.10.2013

Содержание

Создание конфигурационного файла

Создание локальной политики безопасности CSP VPN Gate возможно осуществить также 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

Синтаксические диаграммы верхнего уровня языка описания конфигурации

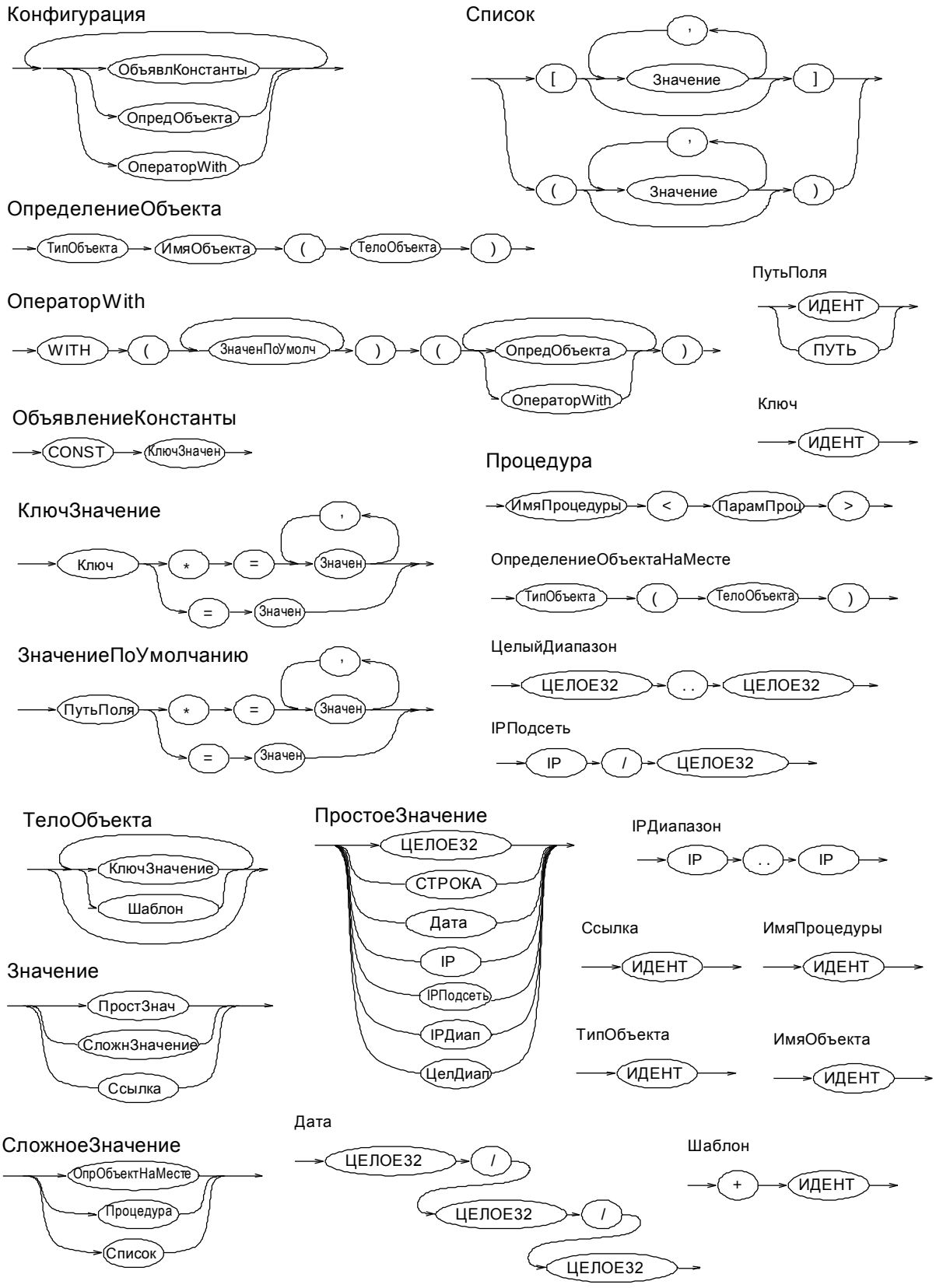


Рисунок 1

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.

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 ключевыми словами `with` и `const`. Внутри имен подстановок оператора `with` могут быть использованы символы '**.**'.

Примеры идентификаторов:

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

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
```


Значения типа 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
```

Ключевые слова

Ключевые слова ***with, const*** используются при создании специальных конструкций.

На диаграмме (Рисунок 1) эти ключевые слова написаны прописными (большими) буквами. В конфигурационном файле ключевые слова должны быть написаны строчными буквами.

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, ключевых слов const, with друг от друга.

Диапазоны значений

```
ProtocolID *= 20..30
IKECFGPool *= 192.168.13.17..192.168.13.127
```

Списки значений

При указании списка значений для какого-либо параметра перед знаком '=' должен стоять символ '*'. Если параметр может иметь список значений, но необходимо указать только одно, то символ '*' можно опустить.

```
GroupID *= MPDP_768, MODP_1024
```


Вложенные списки

Для описания вложенных списков могут использоваться круглые или квадратные скобки.

```
ContainedProposals *= (IPsec_ah_md5, IPsec_esp_des3), (IPsec_ah_md5,
IPsec_esp_idea)
```

Ссылки на структуры

```
LocalCredential *= cert1
```

Определение вложенных структур

```
Transform *= IKETransform(
     CipherAlg *= "DES-CBC"
     HashAlg *= "MD5"
     GroupID *= MODP_768
     LifetimeSeconds = 86400
     LifetimeKilobytes = 4608000
     LifetimeDerivedKeys = 100
)
```

Объявление структуры верхнего уровня

```
FilteringRule Client_Gate(
     LocalIPFilter* = FilterEntry( IPAddress *= 250.192.32.5 )
     PeerIPFilter* = FilterEntry( IPAddress *= 10.10.12.4)
     Action* = ( Client_Gate )
)
```

Специальные конструкции

Для упрощения описания повторяющихся параметров предусмотрена возможность использования именованных констант, значений по умолчанию 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 = FilterEntry(
     IPAddress *= 10.10.12.5
     ProtocolID = 6
     Port =80
     )
FilteringRule Filter_1 (
```



```
        PeerIPFilter* = A
        Action* = ( PASS)
        )
```

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.

Не допускается задавать одни и те же поля и в шаблоне и структуре, в которую он подставляется.

Пример описания:

```
const Transform_DES_MD5 = IKETransform(
        CipherAlg *= "DES-CBC"
        HashAlg *= "MD5"
        GroupID *= MODP_768
)
Transform *= IKETransform(
        + Transform_DES_MD5
        LifetimeSeconds = 86400
        LifetimeKilobytes = 4608000
)
Эквивалентное описание:
Transform *= IKETransform(
        CipherAlg *= "DES-CBC"
        HashAlg *= "MD5"
        GroupID *= MODP_768
        LifetimeSeconds = 86400
        LifetimeKilobytes = 4608000
)
```

Конструкция `WITH` используется для задания значений по умолчанию для полей структур, которые описываются внутри конструкции. После ключевого слова `'with'` указываются пути к полям и значения по-умолчанию для них. Путь к полю структуры может быть записан двумя способами:

первый вариант – это просто имя поля. В этом случае в каждую структуру, которая описана внутри `with`, будет добавлено указанное поле, если в структуре такого поля нет.

во втором варианте путь записывается в форме `тип_верх_ур.имя_поля1.имя_поля2 ... .имя_поляМ`. В этом случае `имя_поляМ` с указанным значением будет добавлено только для структур, которые указаны в качестве значения соответствующего поля структуры уровнем выше. Тип структуры, содержащей `имя_поля1` должен быть `тип_верх_ур`. Значения добавляются только в те структуры, которые определены непосредственно внутри других, а не в виде ссылки.

Значения добавляются только в том случае, если в структуре явно не указано других значений для поля.

Пример:

(* Указание значения по-умолчанию, используя полное имя поля (путь).*)

```
     with (
     (* Значение по-умолчанию для FilteringRule.LocalIPFilte *)
     FilteringRule.LocalIPFilter = FilterEntry(
                                        IPAddress = 10.0.16.84))
(
FilteringRule f0 (
     PeerIPFilter = FilterEntry(IPAddress= 192.168.12.11)
     Action = (DROP))
FilteringRule f1 (
     PeerIPFilter = FilterEntry(IPAddress= 192.168.19.22)
     Action = (PASS))
)
```

(* Таже конфигурация, сокращённая форма - задано значение по-умолчанию для всех структур верхнего уровня, независимо от типа.*)

```
with (
     LocalIPFilter = FilterEntry(IPAddress = 10.0.16.84))
(
FilteringRule f0 (
     PeerIPFilter = FilterEntry(IPAddress= 192.168.12.11)
     Action = (DROP))
FilteringRule f1 (
     PeerIPFilter = FilterEntry(IPAddress= 192.168.19.22)
     Action = (PASS))
)
(* Результирующая конфигурация*)
FilteringRule f0 (
     PeerIPFilter = FilterEntry(IPAddress= 192.168.12.11)
     LocalIPFilter = FilterEntry(IPAddress = 10.0.16.84)
     Action = (DROP)
)
FilteringRule f1 (
     PeerIPFilter = FilterEntry(IPAddress= 192.168.19.22)
     LocalIPFilter = FilterEntry(IPAddress = 10.0.16.84)
     Action = (PASS)
)
```


Структура конфигурации

Структуру конфигурации можно разделить на три логические части:

- Заголовок (GlobalParameters)
- Глобальные параметры протокола IKE (IKEParameters)
- Правила фильтрации (FilteringRules)

Структура конфигурации предполагает наличие только одного заголовка (GlobalParameters), одной структуры глобальных параметров протокола IKE (IKEParameters) и неограниченное количество правил фильтрации (FilteringRule).

Диаграмма структуры конфигурации и взаимосвязь между ее элементами

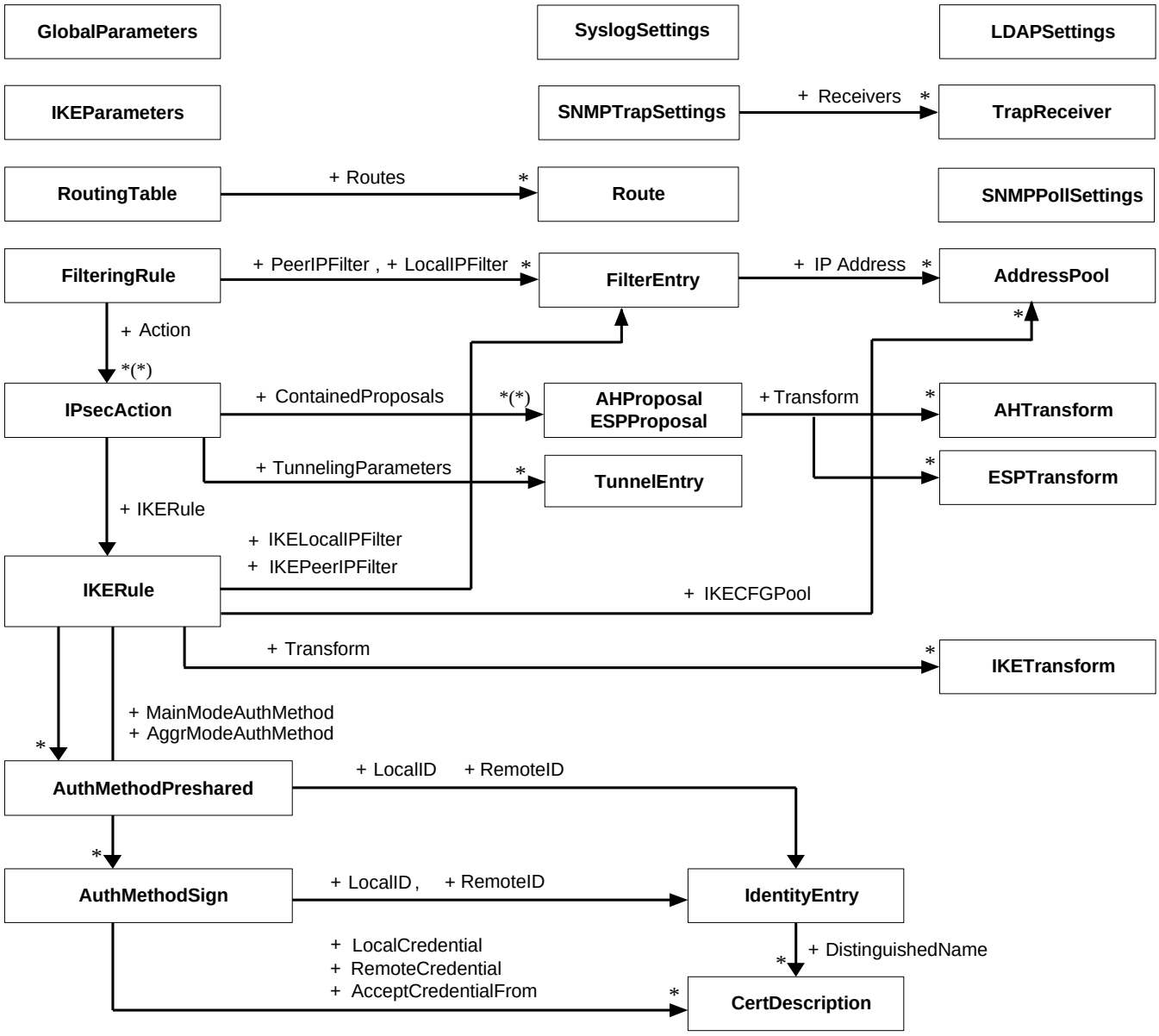


Рисунок 2

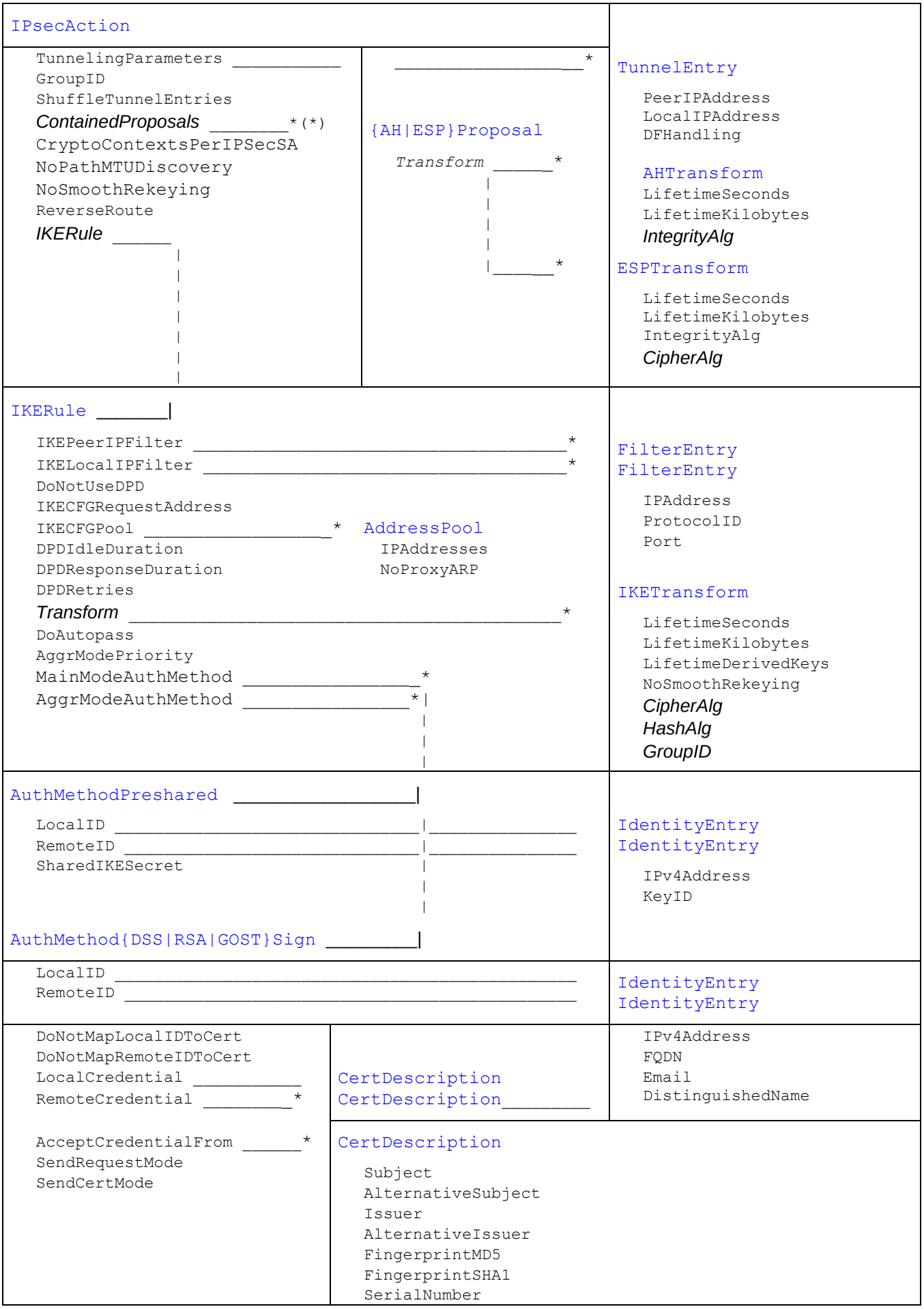
Пояснения к диаграмме:

- в прямоугольниках указаны имена структур данных, составляющих локальную политику безопасности
- стрелка обозначает отношение использования между структурами данных
- рядом со стрелкой указан атрибут структуры, который ссылается на используемую структуру
- '*' рядом со стрелкой обозначает, что атрибут содержит список используемых структур
- '*(*)' обозначает, что атрибут содержит список списков используемых структур.

Таблица 1

Структура конфигурации в табличном виде

GlobalParameters		LDAPSettings
Title Version Type Serial StartOfValidity EndOfValidity CRLHandlingMode LDAPLogMessageLevel SystemLogMessageLevel PolicyLogMessageLevel CertificatesLogMessageLevel		Server Port SearchBase ConnectTimeout ResponseTimeout HoldConnectTimeout DropConnectTimeout
IKEParameters		SysLogSettings
SendRetries RetryTimeBase RetryTimeMax SACreationTimeMax InitiatorSessionsMax ResponderSessionsMax BlacklogSessionsMax BlacklogSessionsMin BlacklogSilentSessions BlacklogRelaxTime IKECFGBindAddress		*Server* Facility
RoutingTable		
Routes _____*		Route|!Route *Destination* Gateway NetworkInterface Metric
SNMPPollSettings	SNMPTrapSettings	
LocalIPAddress Port ReadCommunity SysLocation SysContact	*Receivers* _____*	TrapReceiver|!TrapReceiver *IPAddress* Port *Community* Version SNMPv1AgentAddress
FilteringRule		
PeerIPFilter _____* FilterEntry LocalIPFilter _____* FilterEntry NetworkInterfaces RefuseTCPPeerInit Action _____ |*(*) | | | ..... | IPsecAction _____|	IPAddress _____ ProtocolID Port	AddressPool IPAddresses NoProxyARP


IPsecAction
TunnelingParameters
GroupID
ShuffleTunnelEntries
ContainedProposals *(*)
CryptoContextsPerIPSecSA
NoPathMTUDiscovery
NoSmoothRekeying
ReverseRoute
IKERule
{AH|ESP}Proposal
Transform *
TunnelEntry
PeerIPAddress
LocalIPAddress
DFHandling
AHTransform
LifetimeSeconds
LifetimeKilobytes
IntegrityAlg
ESPTransform
LifetimeSeconds
LifetimeKilobytes
IntegrityAlg
CipherAlg
IKERule
IKEPeerIPFilter *
IKELocalIPFilter *
DoNotUseDPD
IKECFGRequestAddress
IKECFGPool *
DPDIdleDuration
DPDResponseDuration
DPDRetries
Transform *
DoAutopass
AggrModePriority
MainModeAuthMethod *
AggrModeAuthMethod *
AddressPool
IPAddresses
NoProxyARP
FilterEntry
FilterEntry
IPAddress
ProtocolID
Port
IKETransform
LifetimeSeconds
LifetimeKilobytes
LifetimeDerivedKeys
NoSmoothRekeying
CipherAlg
HashAlg
GroupID
AuthMethodPreshared
LocalID
RemoteID
SharedIKESecret
AuthMethod{DSS|RSA|GOST}Sign
IdentityEntry
IdentityEntry
IPv4Address
KeyID
LocalID
RemoteID
IdentityEntry
IdentityEntry
DoNotMapLocalIDToCert
DoNotMapRemoteIDToCert
LocalCredential
RemoteCredential *
CertDescription
CertDescription
IPv4Address
FQDN
Email
DistinguishedName
AcceptCredentialFrom *
SendRequestMode
SendCertMode
CertDescription
Subject
AlternativeSubject
Issuer
AlternativeIssuer
FingerprintMD5
FingerprintSHA1
SerialNumber

В таблице жирным шрифтом выделены имена структур, а курсивом – обязательные атрибуты.

Название структуры в таблице также является ссылкой на описание этой структуры и ее атрибутов.

Знак "*" в конце атрибута конфигурационного файла означает, что значения данного атрибута представлены в виде списка. Если знак "*" не установлен, то предполагается, что вместо списка будет использовано только одно значение или одна ссылка.

Заголовок конфигурации. Структура GlobalParameters

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

Имя структуры GlobalParameters

Атрибуты
Title
Version
Type
Serial
StartOfValidity
EndOfValidity
CRLHandlingMode
LDAPLogMessageLevel
SystemLogMessageLevel
PolicyLogMessageLevel
CertificatesLogMessageLevel

Пример

```
GlobalParameters (
     Title = "Moon host LSP"
     Version = "3.11"
     Serial = "000000010000000E00000001"
     CRLHandlingMode = DISABLE
     LDAPLogMessageLevel = INFO
     SystemLogMessageLevel = INFO
     PolicyLogMessageLevel = INFO
     CertificatesLogMessageLevel = INFO
     )
```

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 | INCREMENTAL**

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

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

INCREMENTAL – конфигурация содержит изменения текущей политики.

Конфигурация такого типа может содержать любые структуры, однако, применяются изменения только для следующих структур: `RoutingTable, SNMPTrapSettings`.

Если изменения применяются к TEMPORARY конфигурации, то результирующая конфигурация будет записана в базу данных, и будет автоматически активизирована при следующем запуске шлюза безопасности.

Значение по умолчанию PERMANENT.

Атрибут Serial

Атрибут Serial определяет уникальный серийный номер конфигурации. В шлюзе безопасности 3.11 данное поле не используется.

Синтаксис Serial = СТРОКА

Значение строка содержит шестнадцатеричное представление серийного номера конфигурации

Значение по умолчанию пустая строка

Атрибут StartOfValidity

Атрибут StartOfValidity определяет момент времени, до которого конфигурация не может быть активизирована.

Синтаксис StartOfValidity = ДАТА

Значение 01/1/0000 – 31/12/9999

Значение по умолчанию ограничения отсутствуют на активизацию конфигурации

Атрибут EndOfValidity

Атрибут EndOfValidity определяет момент времени, после которого конфигурация не может быть активизирована.

Синтаксис EndOfValidity = ДАТА

Значение 01/1/0000 – 31/12/9999

Значение по умолчанию ограничения отсутствуют на активизацию конфигурации

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 LDAPLogMessageLevel

Атрибут LDAPLogMessageLevel задает текущий уровень детализации протоколирования для событий, связанных с доступом к LDAP-серверу.

Синтаксис

LDAPLogMessageLevel = **DEBUG|**
INFO|
NOTICE|
WARNING|
ERR|
CRIT|
ALERT|
EMERG

Значения уровни детализации протоколирования определены в RFC 3164[1].

[1] RFC 3164: The BSD syslog Protocol

Значение по умолчанию Если этот атрибут не указан или не загружена конфигурация, то действует уровень лога, установленный по умолчанию, который задается при помощи утилиты log_mgr set, описанной в документе «Специализированные команды». Если это значение не менялось, то оно равно DEBUG.

Атрибут SystemLogMessageLevel

Атрибут SystemLogMessageLevel задает текущий уровень детализации протоколирования для системных событий.

Синтаксис

SystemLogMessageLevel= **DEBUG|**
INFO|
NOTICE|
WARNING|
ERR|
CRIT|
ALERT|
EMERG

Значения уровни детализации протоколирования определены в RFC 3164.

Значение по умолчанию Если этот атрибут не указан или не загружена конфигурация, то действует уровень лога, установленный по умолчанию, который задается при помощи утилиты `log_mgr set`, описанной в документе «Специализированные команды». Если это значение не менялось, то оно равно DEBUG.

Атрибут PolicyLogMessageLevel

Атрибут PolicyLogMessageLevel задает текущий уровень детализации протоколирования для событий, связанных с применением локальной политики.

Синтаксис

PolicyLogMessageLevel= **DEBUG|**
INFO|
NOTICE|
WARNING|
ERR|
CRIT|
ALERT|
EMERG

Значения уровни детализации протоколирования определены в RFC 3164.

Значение по умолчанию Если этот атрибут не указан или не загружена конфигурация, то действует уровень лога, установленный по умолчанию, который задается при помощи утилиты log_mgr set, описанной в документе «Специализированные команды». Если это значение не менялось, то оно равно DEBUG.

Атрибут CertificatesLogMessageLevel

Атрибут CertificatesLogMessageLevel задает текущий уровень детализации протоколирования для событий, связанных с получением, обработкой сертификатов и их сохранением их в базе данных Продукта.

Синтаксис

CertificatesLogMessageLevel= **DEBUG|**
INFO|
NOTICE|
WARNING
ERR|
CRIT|
ALERT|
EMERG

Значения уровни детализации протоколирования определены в RFC 3164.

Значение по умолчанию Если этот атрибут не указан или не загружена конфигурация, то действует уровень лога, установленный по умолчанию, который задается при помощи утилиты `log_mgr set`, описанной в документе «Специализированные команды». Если это значение не менялось, то оно равно DEBUG.

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.

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.

Для прохождения LDAP-пакетов до каждого используемого шлюзом безопасности LDAP-сервера в политике необходимо задать фильтр вида:

```
        FilteringRule PassLdapTraffic(
        PeerIPFilter = FilterEntry(
             IPAddress = <LDAP-server IP-address from CRL Distibution
             Points extension>
             ProtocolID = 6
             Port = <LDAP-server port>)
        LocalIPFilter = FilterEntry(
             IPAddress = LOCAL_IP_ADDRESSES
             ProtocolID = 6)
        RefuseTCPPeerInit = TRUE
        Action = [PASS]
    )
```


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 3), необходимо указать атрибут SearchBase:

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

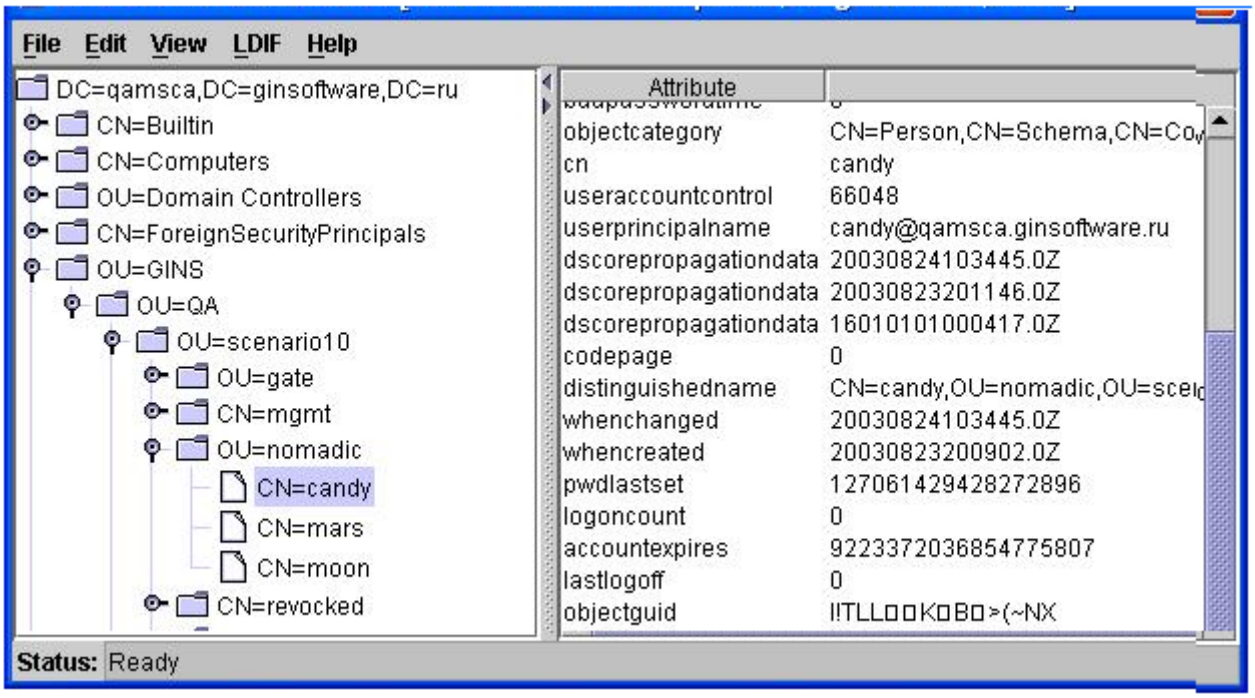


File Edit View LDIF Help
DC=qamsca,DC=ginsoftware,DC=ru
CN=Builtin
CN=Computers
OU=Domain Controllers
CN=ForeignSecurityPrincipals
OU=GINS
OU=QA
OU=scenario10
OU=gate
CN=mgmt
OU=nomadic
CN=candy
CN=mars
CN=moon
CN=revocked
Attribute
objectcategory CN=Person,CN=Schema,CN=Co
cn candy
useraccountcontrol 66048
userprincipalname candy@qamsca.ginsoftware.ru
dscorepropagationdata 20030824103445.0Z
dscorepropagationdata 20030823201146.0Z
dscorepropagationdata 16010101000417.0Z
codepage 0
distinguishedname CN=candy,OU=nomadic,OU=scel
whenchanged 20030824103445.0Z
whencreated 20030823200902.0Z
pwdlastset 127061429428272896
logoncount 0
accountexpires 9223372036854775807
lastlogoff 0
objectguid
Status: Ready

Рисунок 3

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	SendRetries
	RetryTimeBase
	RetryTimeMax
	SACreationTimeMax
	InitiatorSessionsMax
	ResponderSessionsMax
	BlacklogSessionsMax
	BlacklogSilentSessions
	BlacklogSessionsMin
	BlacklogRelaxTime
	IKECFGBindAddress

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

Атрибут SendRetries

Атрибут SendRetries устанавливает число попыток посылки IKE-пакетов партнеру.

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) и 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 SACreationTimeMax

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

Синтаксис SACreationTimeMax = 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.

Синтаксис InitiatorSessionsMax = ЦЕЛОЕ32

Значение число из диапозона 1-10000

Значение по умолчанию 30.

Атрибут ResponderSessionsMax

Атрибут ResponderSessionsMax определяет максимально допустимое количество одновременных обменов, проводимых VPN-устройством с одним неаутентифицированным партнером, в качестве ответчика. С таким партнером нет ни одного ISAKMP SA. Как только создается хотя бы один ISAKMP SA, данный атрибут ResponderSessionsMax перестает действовать.

Синтаксис ResponderSessionsMax = ЦЕЛОЕ32

Значения Целое число из диапозона 1..20

Значение по умолчанию 20.

Атрибут BlacklogSessionsMax

"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". В случае первой неуспешной IKE-сессии, инициированной со стороны партнера, партнер сразу же заносится в "черный список". BlacklogSessionsMax устанавливает число разрешенных одновременных IKE обменов, инициируемых неаутентифицированным партнером, только что попавшим в "черный список". При каждом следующем неудачном завершении IKE обмена число разрешенных одновременных IKE обменов для данного партнера снижается вдвое с округлением в меньшую сторону, вплоть до полного запрещения IKE трафика с данным партнером.

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[2].

Если значение BlacklogSessionsMax больше или равно ResponderSessionsMax, то атрибуту BlacklogSessionsMax присваивается значение ResponderSessionsMax-1.

Значение по умолчанию 16.

Атрибут BlacklogSessionsMin

Атрибут BlacklogSessionsMin позволяет установить минимальное число разрешенных одновременных IKE обменов, инициируемых неаутентифицированным партнером, находящимся в "черном списке".

Синтаксис BlacklogSessionsMin = ЦЕЛОЕ32

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

Если это значение больше, чем BlacklogSessionsMax, то атрибуту BlacklogSessionsMin присваивается значение BlacklogSessionsMax.

Значение по умолчанию 0 – нет ограничения снизу на активные обмены с партнером, находящимся в "черном списке".

Атрибут BlacklogSilentSessions

Атрибут BlacklogSilentSessions позволяет установить число активных обменов, инициированных партнером, находящимся в "черном списке", по достижении которого VPN-устройство перестает информировать партнера о причине отказа в создании IKE–контекста (ISAKMP SA).

Синтаксис BlacklogSilentSessions = ЦЕЛОЕ32

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

Если это значение больше, чем BlacklogSessionsMax, то атрибуту BlacklogSilentSessions присваивается значение BlacklogSessionsMax.

Значение по умолчанию 4.

[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".

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.

Атрибут IKECFGBindAddress

Атрибут IKECFGBindAddress задает политику распределения адресов из IKECFG-пула.

IKECFG-адрес выдается из множества адресов, заданных атрибутом IKECFGPool в активном правиле IKERule, на основе идентификатора партнера (IDii) после успешного окончания первой фазы IKE. Это позволяет распределять партнеров на основе их идентификационной информации на отдельные адресные пространства, для которых далее могут быть определены отдельные политики доступа.

При распределении адресов необходимо проверять, чтобы одному партнеру соответствовал один адрес из IKECFGPool. Данный атрибут задает совокупность параметров партнера, уникальность которых должна быть соблюдена при распределении адресов.

Синтаксис IKECFGBindAddress = **TO_PHASE1_ID | TO_PUBLIC_ADDRESS**

[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.

Значение	TO_PHASE1_ID – идентификатор партнера (IDii) первой фазы IKE ассоциируется только с одним адресом из IKECFG-пула. Если при установлении соединения обнаружено, что для партнера с таким же IDii, но с другими адресом и портом уже ранее был выдан адрес из IKECFG-пула, то конфликт разрешается посредством определения "жив" этот партнер или нет по протоколу DPD. Для этого включается таймер на 10 секунд и посылается Hello этому партнеру. Если этот партнер не отвечает, то найденный в IKECFG-пуле адрес выделяется для нового партнера с таким же IDii. Если партнер отвечает, то новому партнеру с таким же IDii адрес не выделяется и все обмены с новым партнером прекращаются TO_PUBLIC_ADDRESS – один идентификатор (IDii) может ассоциироваться с несколькими адресами из IKECFG-пула и тогда дифференциация производится по адресу и порту партнера[4]. При одновременном использовании разных IDii одним партнером, также выдается несколько адресов из IKECFG-пула.
Значение по умолчанию	TO_PHASE1_ID.

[4] Имееются в виду адрес и порт партнера, по которым было установлено соединение – ISAKMP SA. Дело в том, что при наличии между партнерами NAT-устройства IKE-обмен может начинаться с одной парой адрес-порт, а завершаться (с созданием SA) с другой. IPSec SA с применением UDP-инкапсуляции с этим партнером строится по той же паре адрес-порт, что и созданный ISAKMP SA.

Структура SNMPPollSettings

Структура задает настройки для выдачи информации SNMP-агентом по 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 локальный IPv4-адрес, на который можно получать запросы от SNMP-менеджера.

Синтаксис	LocalIPAddress = IP | **ANY**
Значения	IP –адрес – любой из 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.

При указании этой структуры в инкрементальной политике (если в структуре `GlobalParameters` атрибут `Type` = INCREMENTAL):

- каждая запись TrapReceiver, указанная в атрибуте `Receivers` добавляется в текущий список получателей SNMP-трапов. Запись в текущем списке, у которой поля IPAddress и Community совпадают с этими же полями новой записи, замещается на новую запись целиком;
- каждая запись !`TrapReceiver`, указанная в атрибуте `Receivers` удаляется из текущего списка получателей SNMP-трапов. Если указанной записи нет в списке, загрузка инкрементальной политики прекращается с соответствующей диагностикой.

Имя структуры SNMPTrapSettings

Атрибуты Receivers

Атрибут Receivers

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

Синтаксис Receivers* = `TrapReceiver` | !TrapReceiver

Значение по умолчанию не существует, атрибут обязательный.

Структура TrapReceiver | !TrapReceiver

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

Структура !TrapReceiver описывает одного получателя SNMP-трапов, который удаляется из текущего списка получателей SNMP-трапов.

Имя структуры	TrapReceiver | !TrapReceiver
Атрибуты	IPAddress
	Port
	Community
	Version
	SNMPv1AgentAddress

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 SNMPv1AgentAddress

Атрибут SNMPv1AgentAddress задает IP-адрес источника трап-сообщения, который прописывается в поле Agent address внутри SNMP-пакета. Этот атрибут указывается только для Version = V1.

Синтаксис SNMPv1AgentAddress = IP

Значение IP- адрес

Значение по умолчанию 0.0.0.0.

Структура SyslogSettings

Структура SyslogSettings позволяет задать настройки протокола SYSLOG. Если активной политикой является DDP (Default Driver Policy) или в LSP отсутствует структура SyslogSettings, то настройки протокола SYSLOG считываются из файла syslog.ini. Если этот файл недоступен, то протокол SYSLOG не используется. Структура SyslogSettings также позволяет отключить использование протокола SYSLOG.

В конфигурации должна быть только одна структура данного типа. Этой структуре имя не присваивается.

Имя структуры SyslogSettings

Атрибуты Server

Facility

Атрибут Server

Атрибут Server задает адрес SYSLOG-сервера, на который будут посылаться сообщения о протоклируемых событиях.

Синтаксис Server = IP **| NO_SYSLOG**

Значение IP – одиночный IP-адрес. Указание адреса 0.0.0.0 аналогично указанию константы NO_SYSLOG, которая приводит к отключению использования протокола SYSLOG. При указании адреса 127.0.0.1 сообщения посылаются на локальный хост.

NO_SYSLOG – указание этой константы отключает использование протокола SYSLOG.

Значение по умолчанию не существует, атрибут обязательный.

Атрибут Facility

Атрибут Facility позволяет задать источник сообщений протокола SYSLOG.

Синтаксис Facility = СТРОКА

Значение

LOG_KERN	система ядра
LOG_USER -	пользовательские программы
LOG_MAIL –	почтовая системв
LOG_DAEMON	прочие процессы
LOG_AUTH –	система авторизации и безопасности
LOG_SYSLOG	производятся самим SYSLOG
LOG_LPR –	подсистема печати
LOG_NEWS -	подсистема сетевых сообщений
LOG_UUCP -	подсистема UUCP
LOG_CRON –	системные часы
LOG_AUTHPRIV	
LOG_FTP	
LOG_NTP	
LOG_AUDIT	
LOG_ALERT	
LOG_CRON2	
LOG_LOCAL0	

LOG_LOCAL1
LOG_LOCAL2
LOG_LOCAL3
LOG_LOCAL4
LOG_LOCAL5
LOG_LOCAL6
LOG_LOCAL7 – значение по умолчанию

Значение по умолчанию LOG_LOCAL7.

Структура RoutingTable

Структура RoutingTable описывает таблицу маршрутизации. Таблица содержит записи, необходимые для работоспособности конфигурации, успешное добавление которых в таблицу проверяется на момент загрузки конфигурации.

Если таблица содержит записи, которые уже присутствуют в системной таблице маршрутизации или в которых указаны несуществующие интерфейсы, то загрузка конфигурации будет продолжена с соответствующей диагностикой.

При отгрузке конфигурации из системной таблицы маршрутизации будут удалены все указанные в ней записи маршрутизации, в том числе и существовавшие до загрузки этой конфигурации (например, добавленные командой `route add`).

В конфигурации допускается только один экземпляр этой структуры. Этой структуре не может быть присвоено имя.

При указании этой структуры в инкрементальной политике (если в структуре `GlobalParameters` атрибут `Type` = INCREMENTAL):

- каждая запись Route, указанная в атрибуте `Routes` , добавляется к таблицам маршрутизации операционной системы и текущей конфигурации. Запись маршрутизации, которая уже существует в системной таблице маршрутизации или указанная через несуществующий интерфейс, добавляются в таблицу маршрутизации конфигурации с соответствующей диагностикой. Все дублирующие записи изымаются из результирующей таблицы маршрутизации конфигурации.
- каждая запись !Route, указанная в атрибуте `Routes`, удаляется из текущей таблицы маршрутизации. Записи маршрутизации, которые отсутствуют в системной таблице маршрутизации, диагностируются.

Имя структуры	RoutingTable
Атрибуты	Routes

Атрибут Routes

Атрибут Routes содержит список записей таблицы маршрутизации.

Синтаксис	Routes* = `Route` | !Route
Значение по умолчанию	не существует, атрибут обязательный.

Структура Route | !Route

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

Структура Route описывает запись, которая добавляется к таблицам маршрутизации операционной системы и текущей конфигурации. Структура !Route описывает запись, которая удаляется из текущей таблицы маршрутизации.

Имя структуры	Route | !Route
Атрибуты	Destination
	Gateway
	NetworkInterface
	Metric

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.

Если пользователь хочет задать маршрут по умолчанию в LSP конфигурации, то надо отключить системные настройки маршрута по умолчанию. В противном случае, может возникнуть конфликт и маршрутизация не будет работать правильно (в поставляемых программно-аппаратных комплексах настройки маршрута по умолчанию отсутствуют). Для этого необходимо:

в ОС Solaris удалить файл /etc/defaultrouter, если он существует. Перезагрузить систему

в ОС Linux в файле /etc/sysconfig/network удалить значение во втором столбце строки с ключевым словом GATEWAY, если строка существует. Перезагрузить систему.

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

Синтаксис NetworkInterface = СТРОКА

Значение имя интерфейса

Значение по умолчанию используется значение из атрибута Gateway.

Атрибут Metric

Использовать этот атрибут не рекомендуется, так как в разных ОС имеет разный смысл и будет проигнорирован (например, в ОС Solaris метрика назначается интерфейсу, а не маршруту, как в данном атрибуте).

{Атрибут Metric задает метрику маршрута. В качестве метрики маршрута может использоваться любой показатель: длину маршрута, число промежуточных маршрутизаторов, надежность, задержка, затраты на передачу и др.}

Синтаксис Metric = ЦЕЛОЕ32

Значение Целое число из диапазона 1..255

Значение по умолчанию 1.

Правила пакетной фильтрации. Структура FilteringRule

Правила пакетной фильтрации содержат условия срабатывания правила и те действия, которые необходимо произвести с пакетом, в случае подпадания пакета под правило.

Порядок перечисления правил фильтрации сушественен, так как правила срабатывают в прямом порядке перечисления в конфигурации.

При получении TCP/IP пакета просматриваются правила в порядке указания в конфигурации и сравниваются параметры заголовка пакета, относящиеся к удаленному IP-хосту, до нахождения первого подходящего правила. Если правило не найдено – пакет уничтожается.

Правило считается подходящим, если в структуре FilteringRule в атрибутах PeerIPFilter и LocalIPFilter указаны параметры, совпадающие с параметрами в TCP/IP заголовке пакетов.

В случае выходящих пакетов параметры в атрибуте LocalIPFilter сравниваются с адресом источника пакета. Параметры в атрибуте PeerIPFilter сравниваются с адресом получателя пакета.

Для входящих пакетов параметры в атрибуте LocalIPFilter сравниваются с адресом получателя пакета. Параметры в атрибуте PeerIPFilter сравниваются с адресом источника пакета.

Структура FilterEntry формирует условие срабатывания конкретного правила пакетной фильтрации для партнеров по взаимодействию.

Имя структуры	FilteringRule
Атрибуты	PeerIPFilter
	LocalIPFilter
	NetworkInterfaces
	RefuseTCPPeerInit
	Action
	Firewall

Схематическое представление взаимосвязей структуры FilteringRule:

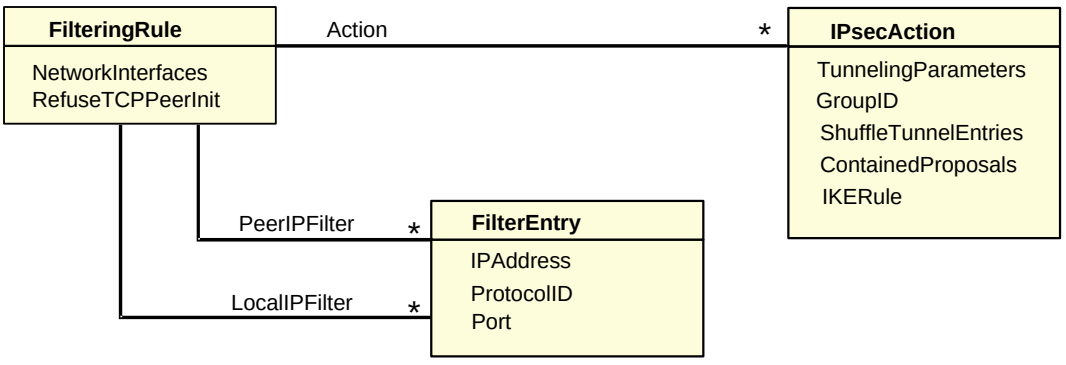


Рисунок 4

Атрибут PeerIPFilter

Атрибут PeerIPFilter описывает параметры удаленного хоста, которые

- в случае выходящих пакетов будут сравниваться с адресом получателя пакета
- в случае входящих пакетов будут сравниваться с адресом источника пакета.

Этот атрибут представляет собой список структур FilterEntry.

Синтаксис PeerIPFilter *= `FilterEntry`

Значение по умолчанию весь сетевой трафик.

Атрибут LocalIPFilter

Атрибут LocalIPFilter описывает параметры защищаемого хоста, а также защищаемых подсетей, которые:

- в случае выходящих пакетов будут сравниваться с адресом источника пакета
- в случае входящих пакетов будут сравниваться с адресом получателя пакета.

Этот атрибут представляет собой список структур FilterEntry.

Синтаксис LocalIPFilter *= `FilterEntry`

Значение по умолчанию весь локальный и транзитный трафик.

При использовании шлюза безопасности в качестве IKECFG-клиента, см. значение атрибута `FilterEntry`.

Атрибут NetworkInterfaces

Атрибут NetworkInterfaces задает список сетевых интерфейсов, на которые могут приходить пакеты от партнера (с которых могут уходить пакеты партнеру).

Синтаксис NetworkInterfaces* = СТРОКА

Значения Список логических имен сетевых интерфейсов. Интерфейсы должны быть указаны в кавычках (кроме служебного слова ANY).

Значение по умолчанию ANY – задействуются все интерфейсы.

Атрибут RefuseTCPPeerInit

Атрибут RefuseTCPPeerInit задает блокировку входящих TCP-соединений; используется как дополнительное ограничение к действию.

Синтаксис RefuseTCPPeerInit* = **TRUE | FALSE**

Значения TRUE – уничтожается первый входящий TCP-пакет соединения, в результате отвергаются все TCP-соединения, инициированные извне

FALSE – не производится никаких дополнительных действий

Значение по умолчанию **FALSE**

Атрибут Action

Атрибут Action описывает варианты действий, допускаемых VPN-устройством по взаимодействию с удаленным хостом.

Синтаксис Action *= (IPsecAction [, IPsecActionN]) | **(DROP)** | **(PASS)**

Значение Действия формируются в виде списка цепочек из правил создания SA:

в списке не должно быть одинаковых цепочек

в списке вместо цепочки могут использоваться зарезервированные слова PASS или DROP. При этом:

определяется действие, которое будет применено к пакету, подпадающему под это правило пакетной фильтрации, при отсутствии соответствующего SA

в списке допускается только одна цепочка заданная таким образом

порядок указания такой цепочки в списке не имеет значения

если в цепочке указано более одного правила, то все они, кроме последнего, должны иметь непустой атрибут TunnelingParameters (в IPsecAction).

Пример

[IPsecAction1] [IPsecAction2, IPsecAction3] [IPsecAction4] [PASS] [IPsecAction5]

Создание SA

Если устройство является инициатором соединения, то трафик будет обрабатываться в соответствии с первой цепочкой списка.

Если устройство является ответчиком, то правило обработки трафика выбирается путем сравнения каждой цепочки этого списка с каждой цепочкой своего списка и выбирается первая совпавшая цепочка.

Обработка трафика

Если выбранная цепочка состоит из двух правил [IPsecAction1, IPsecAction2] или более, то:

- исходящий трафик вначале обрабатывается контекстом, созданным по правилу IPsecAction2, а затем контекстом, созданным по правилу IPsecAction1
- для входящего трафика порядок применения контекстов обратный: вначале трафик обрабатывается контекстом, созданным по правилу IPsecAction1, а затем – IPsecAction2.

Значение по умолчанию (DROP)

Атрибут Firewall

Атрибут Firewall задает дополнительную фильтрацию пакетов по длине IP-пакета, смещению начала фрагмента пакета и контрольной сумме. Если длина пакета, смещение фрагмента пакета соответствуют заданным диапазонам значений в атрибуте Firewall – пакет уничтожается. Если контрольная сумма пакета совпадает с заданной в атрибуте Firewall – пакет уничтожается.

Синтаксис Firewall * = Firewall(

Name=”check_bad_ip_len_and_off”

Args *=Totallen1, Totallen2, Fragoff1, Fragoff2

),

Firewall(

Name=”check_bad_ip_checksum”

Args *=Checksum

),

Значения Totallen1, Totallen2 – задают диапазон длины IP-пакета в байтах

Fragoff1, Fragoff2 – задают диапазон смещения начала фрагмента внутри оригинального пакета в блоках по 8 байт

Checksum – задает значение контрольной суммы пакета.

Для исключения проверки по длине пакета, задайте диапазон 0, 65535.

Для исключения проверки по смещению фрагмента, задайте диапазон 0, 8191.

При получении пакета дополнительная проверка применяется только к открытым пакетам.

При задании атрибута Firewall сначала выполняется проверка на соответствие всем атрибутам структуры FilteringRule, а затем дополнительная проверка. При этом допускаются следующие значения в атрибутах:

PeerIPFilter, LocalIPFilter – IP-адреса, любые протоколы, Порты – не допускаются

Action – только значение [PASS]

NetworkInterfaces – ограничений нет

Firewall – дополнительные проверки.

В атрибуте Firewall проверки осуществляются последовательно. Если длина пакета и смещение начала фрагмента пакета попадают в соответствующие заданные диапазоны значений одновременно – пакет уничтожается, если нет – выполняется следующая проверка. При совпадении контрольной суммы – пакет уничтожается, при несовпадении - пакет пропускается, по следующим структурам FilteringRule проверка не выполняется.

Таким образом, пропускаются все пакеты, кроме тех, , у которых длина пакета и смещение фрагмента попадают в заданные диапазоны, или контрольная сумма равна заданной.

При уничтожении пакета выдается сообщение с указанием совпавших параметров, которое можно получить, задав команду klogview –l –f 0x80.

В LSP структуры FilteringRule с атрибутом Firewall должны быть расположены раньше, чем структуры FilteringRule с IPsec-фильтрами и портами.

Пример

```
FilteringRule f1 (
PeerIPFilter = FilterEntry(IPAddress = 10.0.16.2)
LocalIPFilter = FilterEntry(IPAddress = 0.0.0.0..255.255.255.255)
Action = [PASS]
NetworkInterfaces = "eth0"
Firewall *=
Firewall (
Name = "check_bad_ip_len_and_off"
Args *= 0, 200, 3, 02000h
),
Firewall (
Name = "check_bad_ip_checksum"
Args *= 5549h
))
```


Структура FilterEntry

Структура FilterEntry описывает параметры IP-заголовка пакета, которые будут вставлены в конкретное правило пакетной фильтрации для формирования условия его срабатывания. IP-заголовок получаемых/отправляемых пакетов будет сравниваться с задаваемыми правилами.

Имя структуры	FilterEntry
Атрибуты	IPAddress
	ProtocolID
	Port

Атрибут IPAddress

Атрибут IPAddress описывает список адресов, состоящий из пулов адресов, одиночных адресов, диапазонов адресов и подсетей, или всех локальных адресов устройства для срабатывания конкретного правила (FilteringRule).

Синтаксис	IPAddress *= (`AddressPool`, IP, IP..IP, IP/ЦЕЛОЕ32) | **LOCAL_IP_ADDRESSES**
Значения	AddressPool – множество адресов IKECFG пула
	IP-адрес
	IP..IP – диапазон IP-адресов
	IP/ЦЕЛОЕ32 – IP-адрес с маской
	LOCAL_IP_ADDRESSES – все локальные адреса устройства
Значение по умолчанию	всевозможные IP-адреса.

При использовании шлюза безопасности в качестве IKECFG-клиента, используйте значение `LocalIPFilter *= FilterEntry ( IPAddress *=` **LOCAL_IP_ADDRESSES).**

А при создании правила для СПДС «ПОСТ», когда шлюз безопасности выступает в качестве IKECFG-клиента, используйте значение `LocalIPFilter *= FilterEntry ()`.

Атрибут ProtocolID

Атрибут ProtocolID описывает список протоколов для срабатывания конкретного правила (FilteringRule).

Синтаксис	ProtocolID *= ЦЕЛОЕ32
Значение	целое число из диапазона 0..255.
	Значение 0 означает все сетевые протоколы.
Значение по умолчанию	все протоколы.

Атрибут Port

Атрибут Port описывает список идентификаторов портов для указанных протоколов объекта. Если атрибут ProtocolID отсутствует, то указанные порты будут применяться и к TCP(6), и к UDP(17).

Синтаксис	Port *= ЦЕЛОЕ32
Значение	целое число из диапазона 0..65535.

Значение 0 означает все порты для указанных протоколов.

Значение по умолчанию все порты.

Примечание В случае указания атрибута Port в отсутствии атрибута ProtocolID:

пакеты протоколов TCP и UDP подпадут под фильтр при условии совпадения порта;

пакеты протоколов, не относящихся к TCP и UDP, которые по IP-адресам попадают на данный фильтр и не отфильтровываются раньше, будут уничтожены с диагностикой "no matching filtering rule".

Структура AddressPool

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

Имя структуры	AddressPool
Атрибуты	IPAddresses
	NoProxyARP

Атрибут IPAddresses

Атрибут IPAddresses задает множество адресов в виде списка, состоящего из одиночных адресов и подсетей.

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

Пример:

IPAddresses* = `192.168.3.10, 192.168.3.15, 10.0.0.240/29`

Порядок выдачи IP-адресов из списка пулов по запросу клиентов при создании соединения по IKE-правилу описан в `Атрибуте IKECFGPool` структуры IKERule и приведен пример.

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 сети шлюза безопасности. В остальных случаях его использование не запрещено, однако лишено смысла.

Выданный по IKECFG внешнему устройству IP-адрес должен проксироваться с интерфейса защищаемой сети, чтобы пакеты от устройств этой сети, 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).

ARP-таблица модифицируется по мере создания/завершения соединений с партнерами. При загрузке и выгрузке конфигурации ARP-таблица очищается от proxy-записей, попадающих в объединение всех полей IKECFGAddressPool конфигурации.

Синтаксис	NoProxyARP = **FALSE | TRUE**
Значение	FALSE – для указанного множества адресов устройство выступает в роли ProxyARP
	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.

Структура IPsecAction

Структура IPsecAction задает правило создания контекста соединения для протоколов семейства IPsec. Этой структуре может быть присвоено имя.

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

Структура IPSecAction:

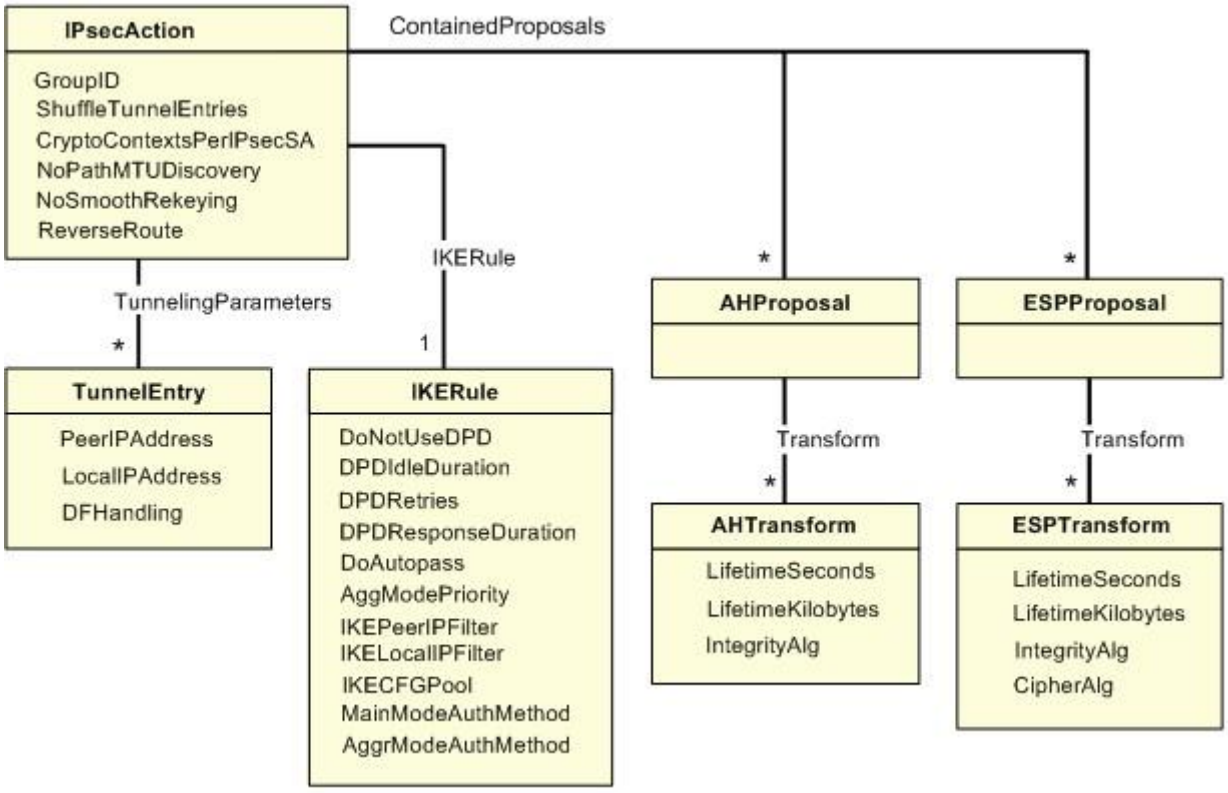


Рисунок 5

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 производиться 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. Если данный атрибут не указан в правиле, то количество контекстов задается параметром из файла agent.ini DefaultCryptoContextsPerIPSecSA Наличие нескольких криптографических контекстов позволяет распараллелить обработку пакетов одним IPsec SA.

Синтаксис CryptoContextsPerIPSecSA = ЦЕЛОЕ32

Значения Целое число из диапазона 1..128.

Значение по умолчанию 1.

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 VKO (GOST R 34.10-2001 [RFC4357]) либо 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* = **VKO_1B, MODP_768, MODP_1024, MODP_1536**

Значения VKO_1B – используется алгоритм VKO GOST R 34.10-2001

MODP_768 – длина ключа 768 бит – группа 1

MODP_1024 – длина ключа 1024 бита – группа 2

MODP_1536 – длина ключа 1536 бит – группа 5

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")
)
```

NoPathMTUDiscovery

Этот атрибут отключает алгоритм "Path MTU Discovery" (выявление максимального размера блока передачи, 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 блока, проходящий по всему каналу без фрагментации).

TRUE – берется значение MTU сетевого интерфейса, через который отправляется пакет.

Значение по умолчанию FALSE

Настройка MTU интерфейса на модуле описана в документе «Настройка шлюза».

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.[5]

Значение по умолчанию FALSE

Предупреждение: если CSP VPN Gate будет использоваться в роли ответчика, то рекомендуется отключить rekeying.

Атрибут ReverseRoute

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

Синтаксис ReverseRoute = **TRUE | FALSE**

Значения TRUE – RRI включен

FALSE – RRI выключен

Значение по умолчанию FALSE

Для вычисления обратного маршрута ID партнера[6] второй фазы IKE преобразуется в адрес и маску подсети. Полученные адрес и маска будут адресом назначения создаваемого маршрута RR.

ID партнера второй фазы IKE в политике безопасности шлюза описан в структуре FilteringRule в атрибуте PeerIPFilter, а ID локальный – в атрибуте LocalIPFilter:

Если `PeerIPFilter` содержит протоколы или порты, произвольный диапазон адресов, которые невозможно преобразовать в адрес и маску подсети, то маршрут RR не создается.

Для построения маршрута RR `LocalIPFilter` также не должен содержать протоколы и порты, указание диапазонов адресов тоже не желательно, так как может быть конфликт, описанный в пункте 2 Таблицы 1 документа «Использование RRI».

Маршрут RR будет построен, если LocalIPFilter и PeerIPFilter содержит адрес и маску подсети или хоста.

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:

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

[6] 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.

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

Вычисление маршрута:

- ID партнера[7] 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.

[7] 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.

Структура TunnelEntry

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

Имя структуры TunnelEntry

Атрибуты PeerIPAddress

LocalIPAddress

DFHandling

Атрибут PeerIPAddress

Атрибут PeerIPAddress описывает туннельный адрес. Этот адрес используется для двух целей – адрес получателя во внешнем IP-заголовке и адрес IKE-партнера, если последний не задан явно.

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.

Пример структуры IPsecAction

```
IPsecAction tunnel_ipsec_des_md5_action(
     TunnelingParameters *= TunnelEntry(
          PeerIPAddress = 192.168.2.1
          DFHandling = CLEAR
     )
     TransportModePriority = ALLOWED
     IKERule = ike_r
     GroupID *= MODP_768, MODP_1024
     ContainedProposals *= (ipsec_ah_md5, ipsec_esp_des),
     (ipsec_esp_des_md5)
)

ESPProposal ipsec_esp_des(
  Transform *= ESPTransform(
    CipherAlg = "DES-CBC"
  )
)

AHProposal ipsec_ah_md5(
  Transform *= AHTransform(
    IntegrityAlg* = "MD5-H96-HMAC"
  )
)

ESPProposal ipsec_esp_des_md5(
  Transform *= ESPTransform(
    CipherAlg = "DES-CBC"
    IntegrityAlg* = "MD5-H96-HMAC"
  )
)
IKERule ike_r(
    Transform* = IKETransform_1
    MainModeAuthMethod *= auth_ca_1
    DoAutopass = TRUE
    IKECFGPool* = Pool_1
)
AddressPool Pool_1(
    IPAddresses *= 10.10.11.224..10.10.11.254
    NoProxyARP = TRUE
```



```
)
IKETransform IKETransform_1(
    CipherAlg* = "AES-K192-CBC"
    HashAlg* = "MD5"
    GroupID* = MODP_1024
    LifetimeSeconds = 86400
)
AuthMethodRSASign auth_ca_1(
    LocalID                    =                    IdentityEntry(
       DistinguishedName*          =          USER_SPECIFIC_DATA
    )
    RemoteID = IdentityEntry(
       DistinguishedName* = USER_SPECIFIC_DATA
    )
    SendRequestMode = ALWAYS
    SendCertMode = ALWAYS
)
```

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.

Имя структуры	AHTransform
Атрибуты	LifetimeSeconds
	LifetimeKilobytes
	IntegrityAlg

Атрибут LifetimeSeconds

Атрибут LifetimeSeconds задает максимальное время существования контекста (SA) AH (в секундах).[8]

Синтаксис	LifetimeSeconds = ЦЕЛОЕ32
Значение	число из диапазона 1.. 4 294 967 295.
Значение по умолчанию	28800 (8 часов).

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 1.. 4 294 967 295.
Значение по умолчанию	нет ограничений на действие SA.

Атрибут IntegrityAlg

Атрибут IntegrityAlg задает набор предлагаемых/допустимых алгоритмов проверки целостности в рамках создаваемого контекста. Список должен содержать хотя бы один элемент.

Рекомендуется указывать не список алгоритмов, а только один алгоритм проверки целостности пакета.

Если же указан список алгоритмов и шлюз безопасности является инициатором соединения, то будет использоваться только первый элемент списка.

Если же существует необходимость задать несколько алгоритмов (их комбинацию) проверки целостности, то используйте альтернативный подход: в атрибуте Transform структуры

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

AHProposal укажите список структур AHTransform, а в каждой структуре AHTransform задайте только один алгоритм проверки целостности.

Синтаксис	IntegrityAlg* = СТРОКА
Значение	Возможные значения: "MD5-H96-KPDK" – Keyed MD5 "MD5-H96-HMAC" – HMAC MD5 (96 бит) "SHA1-H96-HMAC" – HMAC SHA-1 (96 бит) "GR341194CPRO1-H96-HMAC-254" – реализация ГОСТ Р 34.11-94 (96 бит)
Значение по умолчанию	не существует, атрибут обязательный.

Структура ESPTransform

Структура ESPTransform задает параметры контекста (SA) ESP.

Имя структуры	ESPTransform
Атрибуты	LifetimeSeconds
	LifetimaKilobytes
	CipherAlg
	IntegrityAlg

Атрибут LifetimeSeconds

Атрибут LifetimeSeconds задает максимальное время существования контекста в секундах.[9]

Синтаксис	LifetimeSeconds = ЦЕЛОЕ32
Значение	Целое число из диапазона 1.. 4 294 967 295.
Значение по умолчанию	28800 (8 часов).

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.. 4 294 967 295.
Значение по умолчанию	нет ограничений на действие SA.

Атрибут CipherAlg

Атрибут CipherAlg задает набор предлагаемых/допустимых алгоритмов шифрования трафика в рамках создаваемого контекста. Список должен содержать хотя бы один элемент.

Рекомендуется указывать не список алгоритмов, а только один алгоритм шифрования трафика.

Если же указан список алгоритмов и шлюз безопасности является инициатором соединения, то будет использоваться только первый элемент списка.

Если же существует необходимость задать несколько алгоритмов шифрования, то используйте альтернативный подход: в атрибуте Transform структуры ESPProposal укажите

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

список структур ESPTransform, а в каждой структуре ESPTransform задайте только один алгоритм шифрования.

Синтаксис CipherAlg* = СТРОКА

Значение Возможные значения:

"NULL" – NULL (данные не шифруются)

"DES-CBC_IV64" – DES в режиме CBC с явным IV длиной 64 бита

"DES-CBC_IV32" – DES в режиме CBC с явным IV длиной 32 бита

"DES-CBC" – DES в режиме CBC

"DES3-K168-CBC" – DES3 в режиме CBC

"IDEA-CBC" – IDEA 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

"G2814789CPRO1-K256-CBC-254" – реализация ГОСТ 28147-89 в режиме CBC с длиной ключа 256 бит

"G2814789CPRO1-K288-CNTMAC-253" – реализация ГОСТ 28147-89 в комбинированном режиме (гаммирование и вычисление имитовставки) в соответствии со спецификацией ESP_GOST-4M-IMIT (Методические рекомендации по использованию комбинированного алгоритма шифрования вложений IPsec ESP на основе ГОСТ 28147-89 (Проект) "rus-fedchenko-cpesp-ipsecme-gost-00-rt.pdf")

Значение по умолчанию не существует, атрибут обязательный.

Атрибут IntegrityAlg

Атрибут IntegrityAlg задает набор предлагаемых/допустимых алгоритмов проверки целостности пакета в рамках создаваемого контекста. Список должен содержать хотя бы один элемент.

Рекомендуется указывать не список алгоритмов, а только один алгоритм проверки целостности пакета.

Если же указан список алгоритмов и шлюз безопасности является инициатором соединения, то будет использоваться только первый элемент списка.

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.

Синтаксис IntegrityAlg* = СТРОКА

Значение Возможные значения:

"MD5-H96-KPDK" – Keyed MD5

"MD5-H96-HMAC" – HMAC MD5 (96 бит)

"SHA1-H96-HMAC" – HMAC SHA-1 (96 бит)

"GR341194CPRO1-H96-HMAC-65534" – реализация ГОСТ Р 34.11-94 (96 бит)

Значение по умолчанию при отсутствии в связке контекстов компонента AHProposal, список должен содержать хотя бы один элемент. Иначе функциональность проверки целостности пакетов возлагается на протокол AH.

Пример структуры ESPProposal

```
ESPTransform esp_trf_01(
     LifetimeSeconds = 28800
     LifetimeKilobytes = 4608000
     CipherAlg  *= "G2814789CPRO1-K256-CBC-254"
     IntegrityAlg *= "GR341194CPRO1-H96-HMAC-65534"
)
ESPTransform esp_trf_02(
     LifetimeSeconds = 28800
     LifetimeKilobytes = 4608000
     CipherAlg  *= "G2814789CPRO1-K256-CBC-254"
     IntegrityAlg *= "MD5-H96-HMAC"
)
ESPTransform esp_trf_03(
     LifetimeSeconds = 28800
     LifetimeKilobytes = 4608000
     CipherAlg  *= "G2814789CPRO1-K256-CBC-254"
     IntegrityAlg *= "SHA1-H96-HMAC"
)
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
	DoNotUseDPD
	DPDIdleDuration
	DPDResponseDuration
	DPDRetries
	IKECFGRequestAddress
	DoAutopass
	AggrModeAuthMethod
	MainModeAuthMethod
	AggrModePriority
	Transform
	IKECFGPool

Схематическое представление структуры IKERule и структур, на которые ссылаются атрибуты IKERule:

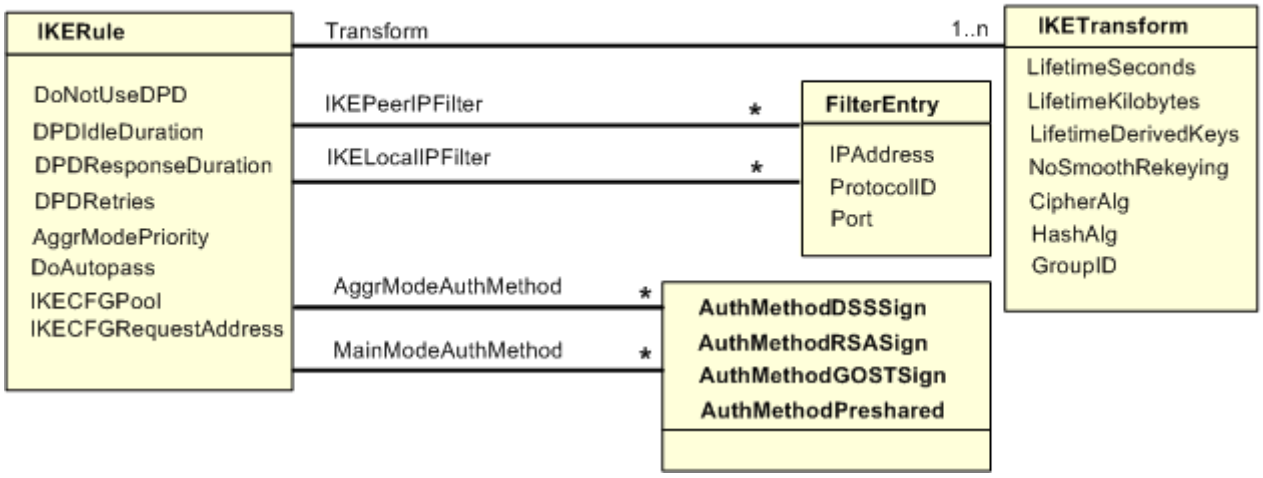


Рисунок 6

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* = `FilterEntry`

Значения В качестве элементов списка могут использоваться структуры `FilterEntry`, используемые в `FilteringRule`, либо самостоятельные структуры `FilterEntry`. При этом для каждого элемента:

- учитываются все единичные IP-адреса и диапазоны. При наличии хотя бы одного элемента без IP-адресов, принимается логика проверки IP-адреса IKE-партнера по умолчанию.
- значения протоколов игнорируются. При анализе входящего ISAKMP-пакета в качестве протокола всегда подразумевается UDP.
- значения портов игнорируются. При анализе входящего ISAKMP-пакета допускаются любые порты партнера.

Значение по умолчанию При проверке IP-адреса IKE-партнера учитываются все возможные значения IP-адресов, используемых в структурах `TunnelEntry` в списках `TunnelingParameters` всех возможных структур `IPsecAction`, которые в свою очередь используют текущее правило `IKERule`. Если в какой-либо из таких структур `TunnelEntry` не задано поле PeerIPAddress, то данное правило `IKERule` может быть использовано с любым партнером.

Если в таких `IPsecAction` поле `TunnelingParameters` отсутствует (транспортный режим), то IP-адрес IKE-партнера проверяется по полю `PeerIPFilter` всех структур `FilteringRule`, в которых используется данное правило `IPsecAction`.

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.

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

Синтаксис IKELocalIPFilter* = `FilterEntry`

Значения В качестве элементов списка могут использоваться структуры `FilterEntry`, используемые в `FilteringRule`, либо самостоятельные структуры `FilterEntry`. При этом для каждого элемента:

- учитываются все единичные IP-адреса и диапазоны. При наличии хотя бы одного элемента без IP-адресов, принимается логика проверки IP-адреса IKE-партнера по умолчанию.

значения протоколов игнорируются. При анализе входящего ISAKMP-пакета в качестве протокола всегда подразумевается UDP.

значения портов игнорируются. При анализе входящего ISAKMP-пакета допускаются следующие локальные порты:

согласно IKEParameters→DefaultPort (по умолчанию – 500)

4500 (используется для NAT Traversal).

Значение по умолчанию допускаются любые локальные IP-адреса.

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 IKECFGRequestAddress

Если шлюз безопасности выступает в качестве IKECFG-клиента, то используйте данный атрибут со значением TRUE для запроса адреса из IKECFG пула партнера.

Синтаксис IKECFGRequestAddress = **TRUE** | **FALSE**

Значение TRUE – агент является активным IKECFG-клиентом, т.е.агент инициирует посылку запроса на получение внутреннего IP-адреса у партнера сразу после создания IKE SA. Возможны следующие варианты дальнейшей работы агента:

Состояние партнера		Дальнейшие действия агента
Партнер является IKECFG-сервером	Успешно выделен IP-адрес из IKECFG-пула	a) Инициируется вторая фаза IKE b) Полученный адрес будет использован в качестве локального для пакетов, которые будут обрабатываться IPsec SA, созданными на основе исходного ISAKMP SA (включая его потомков – rekeying).
	Выдача адреса невозможна в ходе текущей IKECFG-сессии	Инициируется вторая фаза создания соединения, в ходе которой может быть инициирована новая IKECFG-сессия со стороны партнера для выдачи IP-адреса.
	Ошибка выдачи адреса (например, пул адресов исчерпан)	Создание соединения будет остановлено.
Партнер не является IKECFG-сервером		Инициируется вторая фаза создания соединения.

FALSE – агент является пассивным IKECFG-клиентом, т.е. IKECFG-сессия может быть проведена только по инициативе партнера, если он является IKECFG – сервером.

Значение по умолчанию FALSE

Примечание:

Если CSP VPN Gate используется как активный IKECFG-клиент, то в этом случае он не предназначен для обработки транзитного трафика (в атрибуте `LocalIPFilter` надо использовать LOCAL_IP_ADDRESSES или адреса локальной подсети).

Атрибут IKECFGPool

Атрибут IKECFGPool задает список пулов IP-адресов для данного правила. Задание пула адресов выполняется в структуре `AddressPool`. В случае срабатывания данного правила локальное устройство является IKECFG-сервером для партнера. Если в сработавшем правиле атрибут IKECFGPool не задан и партнер является активным IKECFG-клиентом, то создание соединения будет прекращено. Указанные пулы могут совпадать с пулами других правил IKERule, но пересечения по ним не допускаются.

Адреса, соответствующие списку IKECFG пулов, при загрузке LSP упорядочиваются в порядке возрастания. При IKECFG запросе клиенту выдается первый свободный адрес из этого упорядоченного списка. При запросе со стороны другого клиента, работающего с тем же списком пулов, ему будет выдан следующий свободный адрес и т.д. (см.Пример для атрибута IKECFGPool).

Синтаксис IKECFGPool* = `AddressPool`

Значение по умолчанию IKECFG не проводится.

Атрибут DoAutopass

Атрибут DoAutopass задает автоматическую генерацию фильтра для пропускания ISAKMP-трафика.

Синтаксис DoAutopass = **TRUE** | **FALSE**

Значение TRUE:

автоматически пропускать ISAKMP-пакеты, соответствующие атрибутам `IKEPeerIPFilter` и `IKELocalIPFilter`

при отсутствии IP-адреса в атрибуте `IKEPeerIPFilter`, IP-адреса для построения фильтра берутся из атрибута `TunnelingParameters` всех структур `IPsecAction`, ссылающихся на данное правило `IKE`

для структур `IPsecAction`, ссылающихся на данное правило `IKE`, в которых атрибут `TunnelingParameters` отсутствует, IP-адреса берутся из атрибутов `PeerIPFilter`, `LocalIPFilter` всех структур `FilteringRule`, имеющих ссылки на такие `IPsecAction`

FALSE:

не пропускать автоматически ISAKMP-трафик. Правило фильтрации (Filtering Rule) с действием PASS должно быть задано явно (вручную) для пропускания ISAKMP-трафика.

Значение по умолчанию FALSE.

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 предустановленного 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 предустановленного 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 групп параметров протокола ISAKMP для создания SA. Количество элементов списка не ограничего.

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

Пример для атрибута IKECFGPool

В примере представлены две структуры IKERule, каждая из которых имеет свой пул адресов.

В первом правиле IKE_cmap_1 пул состоит из одного диапазона адресов (списка одиночных адресов):

```
IKECFGPool* = pool_dyn
```

Во втором правиле IKE_cmap_2 пул состоит из списка диапазонов адресов:

```
IKECFGPool* = pool_dyn3, pool_dyn2, pool_dyn4
```

В этом правиле при загрузке LSP адреса будут упорядочены в порядке возрастания - `pool_dyn2, pool_dyn3, pool_dyn4,` и при запросе клиента сначала будет выдан первый адрес из диапазона `pool_dyn2,` при следующем запросе - второй адрес из этого же диапазона и т.д. Всегда будет выдаваться первый свободный адрес из упорядоченного списка адресов.

```
IKERule IKE_cmap_1 (
    IKEPeerIPFilter* = FilterEntry(IPAddress* = 10.10.2.102)
```



```
        DoNotUseDPD = TRUE
        IKECFGPool* = pool_dyn
        MainModeAuthMethod* = IKE_auth_cs_key_10_10_2_102
        AggrModeAuthMethod* = IKE_auth_cs_key_10_10_2_102
        AggrModePriority = FALSE
        DoAutopass = TRUE
    )
    AddressPool pool_dyn (
        IPAddresses* = 192.168.0.240..192.168.0.248
        NoProxyARP = FALSE
    )
    IKERule IKE_cmap_2 (
        IKEPeerIPFilter* = FilterEntry(IPAddress* = 10.10.2.10)
        DoNotUseDPD = TRUE
        IKECFGPool* = pool_dyn3, pool_dyn2, pool_dyn4
        MainModeAuthMethod* = IKE_auth_cs_key_10_10_2_10
        AggrModeAuthMethod* = IKE_auth_cs_key_10_10_2_10
        AggrModePriority = FALSE
        DoAutopass = TRUE
    )
    AddressPool pool_dyn3 (
        IPAddresses* = 192.168.3.220..192.168.3.228
        NoProxyARP = FALSE
    )
    AddressPool pool_dyn2 (
        IPAddresses* = 192.168.2.230..192.168.2.238
        NoProxyARP = FALSE
    )
    AddressPool pool_dyn4 (
        IPAddresses* = 192.168.4.210..192.168.4.218
        NoProxyARP = FALSE
    )
    AuthMethodPreshared IKE_auth_cs_key_10_10_2_102 (
        SharedIKESecret = "cs_key_10_10_2_102"
        RemoteID = IdentityEntry (IPv4Address* = 10.10.2.102)
    )
    AuthMethodPreshared IKE_auth_cs_key_10_10_2_10 (
        SharedIKESecret = "cs_key_10_10_2_10"
        RemoteID = IdentityEntry (IPv4Address* = 10.10.2.10)
    )
```


Структура IKETransform

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

Имя структуры	IKETransform
Атрибуты	LifetimeSeconds
	LifetimeKilobytes
	LifetimeDerivedKeys
	NoSmoothRekeying
	CipherAlg
	HashAlg
	GroupID

Атрибут LifetimeSeconds

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

Синтаксис	LifetimeSeconds = ЦЕЛОЕ32
Значение	Целое число из диапазона 1 .. 4 294 967 295.
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 .. 4 294 967 295.
Значение по умолчанию	нет ограничений на действие SA.

Атрибут LifetimeDerivedKeys

Атрибут LifetimeDerivedKeys 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 LifetimeDerivedKeys = ЦЕЛОЕ32

Значение Целое число из диапазона 1 .. 4 294 967 295.

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[10].

Значение по умолчанию FALSE

Предупреждение: если CSP VPN Gate будет использоваться в роли ответчика, то рекомендуется отключить rekeying.

Атрибут CipherAlg

Атрибут CipherAlg задает набор предлагаемых/допустимых алгоритмов шифрования для ISAKMP.

Рекомендуется указывать не список алгоритмов, а только один алгоритм шифрования.

Если же указан список алгоритмов и шлюз безопасности является инициатором соединения, то будет использоваться только первый элемент списка.

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* = СТРОКА

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

Значение	возможные значения: "DES-CBC" – DES в режиме CBC "IDEA-CBC" – IDEA в режиме CBC "DES3-K168-CBC" – DES3 в режиме CBC "AES-K128-CBC" – AES в режиме CBC с длиной ключа 128 бит "AES-K192-CBC" – AES в режиме CBC с длиной ключа 192 "AES-K256-CBC" – AES в режиме CBC с длиной ключа 256 "G2814789CPRO1-K256-CBC-65534" – реализация ГОСТ 28147-89 в режиме CBC с длиной ключа 256 бит
Значение по умолчанию	не существует, атрибут обязательный, список должен содержать хотя бы один элемент.

Атрибут HashAlg

Атрибут HashAlg задает набор предлагаемых/допустимых алгоритмов вычисления хэша для ISAKMP.

Рекомендуется указывать не список алгоритмов, а только один алгоритм хэширования.

Если же указан список алгоритмов и шлюз безопасности является инициатором соединения, то будет использоваться только первый элемент списка.

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* = СТРОКА
Значение	"MD5" "SHA1" "GR341194CPRO1-65534" – реализация ГОСТ Р 34.11-94
Значение по умолчанию	не существует, атрибут обязательный, список должен содержать хотя бы один элемент.

Атрибут GroupID

Атрибут GroupID описывает предлагаемые/допустимые параметры для выработки ключевого материала для ISAKMP. Используется алгоритм VKO (GOST R 34.10-2001 [RFC4357]) либо алгоритм Диффи-Хеллмана.

Рекомендуется указывать не список параметров, а только один элемент списка.

Если же указан список и шлюз безопасности является инициатором соединения, то будет использоваться только первый элемент списка.

Если же 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 число предлагаемых элементов списка должно быть равно единице. 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**
Значение	VKO_1B – используется алгоритм VKO GOST R 34.10-2001
	MODP_768 – стандартная Oakley-группа с длиной ключа 768 бит – группа 1
	MODP_1024 – стандартная Oakley-группа с длиной ключа 1024 бита – группа 2
	MODP_1536 – стандартная Oakley-группа с длиной ключа 1536 бит – группа 5
Значение по умолчанию	не существует, атрибут обязательный, список должен содержать хотя бы один элемент.

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.

Пример структуры IKERule

```
IKETransform ike_trf_01(
     LifetimeSeconds = 28800
     CipherAlg  *= "G2814789CPRO1-K256-CBC-65534"
     HashAlg *= "GR341194CPRO1-65534"
     GroupID *= MODP_1536
)
IKETransform ike_trf_02(
     LifetimeSeconds = 28800
     CipherAlg  *= "DES-CBC"
     HashAlg *= "GR341194CPRO1-65534"
     GroupID *= MODP_1024
)
IKETransform ike_trf_03(
     LifetimeSeconds = 28800
     CipherAlg  *= "AES-K128-CBC"
     HashAlg *= "GR341194CPRO1-65534"
     GroupID *= MODP_768
)
IKERule ike_rule(
     DoNotUseDPD = FALSE
     PDIdleDuration = 60
     DPDResponseDuration = 5
     DPDRetries = 3
```



```
        MainModeAuthMethod *= auth_method_01
        Transform *= ike_trf_01,ike_trf_02,ike_trf_03
        DoAutopass = TRUE
    )
```

Структуры для аутентификации

Схема данных структур AuthMethodDSSSign, AuthMethodRSASign, AuthMethodGOSTSign, AuthMethodPreshared, описывающих идентификационную информацию, предполагаемую к использованию при создании IKE контекста соединения, представлена на рисунке ниже.

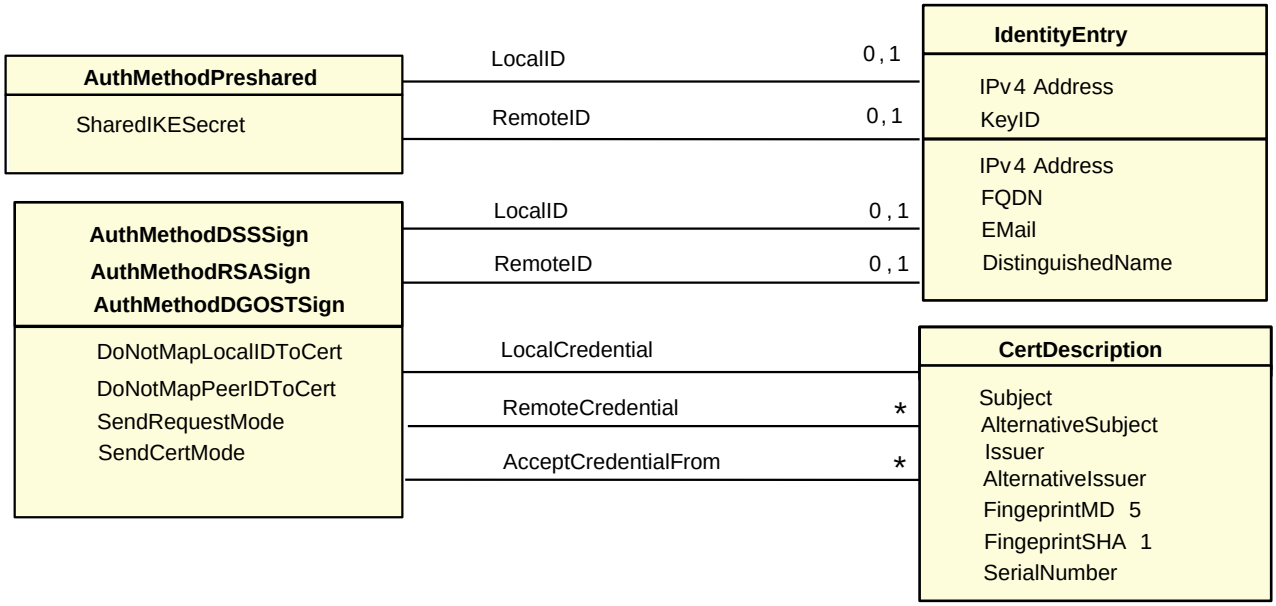


Рисунок 7

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.

AuthMethodDSSSign – аутентификация при помощи подписи, созданной с использованием алгоритма DSS

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

AuthMethodGOSTSign – аутентификация при помощи подписи, созданной с использованием алгоритма ГОСТ Р 34.10-2001.

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

Атрибут LocalID

Атрибут LocalID задает идентификационную информацию данного VPN-устройства.

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 характеристики сертификата данного VPN-устройства. В случае использования аутентификации на алгоритме ГОСТ Р 3410 локальный сертификат используется, если его секретный ключ доступен.

Синтаксис LocalCredential = CertDescription

Значение по умолчанию требования отсутствуют. Используется первый локальный сертификат.

Атрибут RemoteCredential

Атрибут RemoteCredential задает требуемые характеристики сертификата партнера по взаимодействию.

Синтаксис RemoteCredential* = CertDescription

Значение по умолчанию требования отсутствуют, допускается любой сертификат.

Атрибут AcceptCredentialFrom

Атрибут AcceptCredentialFrom задает требуемые характеристики 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 атрибута LocalID (используется первое значение) и CertDescription из

атрибута LocalCredential. Объединение строится по следующим правилам:

если LocalID задан зарезервированным словом `USER_SPECIFIC_DATA`, то результирующий CertDescription совпадает с исходным CertDescription из атрибута LocalCredential

если LocalID задан типом DistingueshedName, то:

если в исходном CertDescription задано поле Subject, то множество его атрибутов должно являться подмножеством атрибутов значения LocalID. Если это условие не выполняется, то соединение не установится;

результирующий CertDescription получается заменой или добавлением значения поля Subject из LocalID в исходном CertDescription;

если LocalID задан типом, отличным от DistingueshedName, то:

если в исходном CertDescription задано поле такого же типа, то их значения должны совпадать. Если это не выполняется, то соединение не установится;

результирующий CertDescription получается добавлением значения LocalID в исходный CertDescription;

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 на сертификат партнера.

Синтаксис SendRequestMode = **AUTO | NEVER | ALWAYS**

Значение AUTO – запрос высылается, если возможный сертификат партнера отсутствует

NEVER – запрос не высылается

ALWAYS – запрос высылается всегда

Значение по умолчанию AUTO

Атрибут SendCertMode

Атрибут SendCertMode определяет логику отсылки локального сертификата 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.

Структура AuthMethodPreshared

Структура AuthMethodPreshared задает аутентификационную информацию при использовании предустановленных (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 предустановленный секретный ключ.

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

Синтаксис	SharedIKESecret = СТРОКА
Значение	имя предустановленного (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 для `AuthMethodPreshared` – атрибут игнорируется;

для `AuthMethod{DSS|RSA|GOST}Sign:`

строки вида "host.domain". Шаблоны не допускаются.

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 для `AuthMethodPreshared` – атрибут игнорируется

для `AuthMethod{DSS|RSA|GOST}Sign`:

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 для `AuthMethodPreshared` – атрибут игнорируется

для `AuthMethod{DSS|RSA|GOST}Sign`:

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 строка, содержащая 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.

шаблоны не допускаются.

Для `AuthMethod{ DSS|RSA|GOST}Sign` – атрибут игнорируется.

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 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 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"
)
```


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

В шлюзе безопасности версии 3.11 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

1. 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.
2. 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.

3. 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.

Примеры конфигураций

Пример 1

Политика «Защищенное взаимодействие между двумя подсетями» (например, защищенный туннель между головным офисом и филиалом), а «в открытые сети (интернет) – только по HTTP».

```
GlobalParameters (
    Title = "LSP for Gate"
    Version = "3.11"
    Type = PERMANENT
    CRLHandlingMode = DISABLE
    LDAPLogMessageLevel = ERR
    SystemLogMessageLevel = ERR
    PolicyLogMessageLevel = ERR
    CertificatesLogMessageLevel = ERR
)
FilteringRule filter_1 (
    PeerIPFilter *= FilterEntry (
       IPAddress = 192.168.2.0/24
    )
    LocalIPFilter* = FilterEntry (
       IPAddress = 192.168.13.0/24
    )
    NetworkInterfaces* = "iprb0"
    RefuseTCPPeerInit = TRUE
    Action *= (tunnel_IPsec_des_md5_action)
)
FilteringRule filter_2 (
    PeerIPFilter* = FilterEntry (
       ProtocolID = 6
       Port = 80
    )
    LocalIPFilter* = FilterEntry (
       IPAddress = 192.168.13.0/24
       ProtocolID = 6
    )
    NetworkInterfaces* = "iprb0"
    Action* = (PASS)
)
```



```
FilteringRule filter_3 (
    # пропускать все на внутреннем интерфейсе шлюза
    NetworkInterfaces* = "iprb1"
    Action *= (PASS)
)

FilteringRule filter_4 (
     # запрещать трафик, неописанный явно
     Action *= (DROP)
)

IPsecAction tunnel_IPsec_des_md5_action(
   TunnelingParameters *= TunnelEntry(
      PeerIPAddress = 192.168.2.1
      LocalIPAddress = 192.168.31.15
      DFHandling = CLEAR
   )
   IKERule = ike_rule_mm_rsa_des_md5_dh1
   GroupID *= MODP_768, MODP_1024
   ContainedProposals      *=      (IPsec_ah_md5,      IPsec_esp_des),
                      (IPsec_esp_des_md5)
)

ESPProposal IPsec_esp_des(
    Transform *= ESPTransform(
       CipherAlg *= "DES-CBC"
       IntegrityAlg *= "MD5-H96-HMAC"
    )
)
AHProposal IPsec_ah_md5(
    Transform *= AHTransform(
       IntegrityAlg *= "MD5-H96-HMAC"
    )
)

ESPProposal IPsec_esp_des_md5(
    Transform* = ESPTransform(
       CipherAlg *= "DES-CBC"
       IntegrityAlg *= "MD5-H96-HMAC"
    )
)
```



```
IKERule ike_rule_mm_rsa_des_md5_dh1(
    DoNotUseDPD = TRUE
    DPDIdleDuration = 60
    DPDResponseDuration = 5
    DPDRetries = 3
    DoAutopass = TRUE
    Transform *= IKETransform(
      CipherAlg *= "DES-CBC"
      HashAlg *= "MD5"
      GroupID *= MODP_768
      LifetimeSeconds = 86400
      LifetimeKilobytes = 86400
      LifetimeDerivedKeys = 86400
    )
    MainModeAuthMethod *= AuthMethodRSASign(
       LocalID = IdentityEntry(
          DistinguishedName *= USER_SPECIFIC_DATA
       )
       RemoteID = IdentityEntry(
           DistinguishedName *= USER_SPECIFIC_DATA
       )
       LocalCredential = CertDescription(
           Issuer*    =    COMPLETE,    "    CN=S-Terra    CenterCA,
                              O=S-Terra, L=Moscow, C=RU"
           Subject* = TEMPLATE, "CN=S-Terra, OU=QA"
           SerialNumber = "617C9958FFFF"
       )
       RemoteCredential *= CertDescription(
           Subject*= "C=RU,O=OrgName,OU=qa0,CN=mars"
           AlternativeSubject       =       "EMAIL=information@sss.ru,
                              DNS= tester.sss.ru, IP =10.10.1.1"
           Issuer* = TEMPLATE, "CN=S-Terra, OU=QA"
           AlternativeIssuer = "IP=1.1.1.1"
           SerialNumber = "617C9958FFFF"
        )
    )
)
```

Пример 2

Пример конфигурации, разрешающей весь трафик:

```
GlobalParameters (
)
```



```
FilteringRule filter_1 (
   Action* = (PASS)
)
```

Пример 3

Пример конфигурации, разрешающей внешний трафик только по протоколу HTTP (разрешить доступ к внешним web – ресурсам):

```
GlobalParameters (
)
FilteringRule filter_1 (
   PeerIPFilter* = FilterEntry (
      ProtocolID = 6
      Port = 80
   )
   LocalIPFilter* = FilterEntry (
      IPAddress = LOCAL_IP_ADDRESSES
      ProtocolID = 6
   )
   NetworkInterfaces* = "interface_1"
   Action* = (PASS)
)
FilteringRule filter_2 (
   Action* = (DROP)
)
```

Пример 4

Пример конфигурации, разрешающей доступ к подсети 192.168.13.0 по защищенному каналу, а ко всем остальным ресурсам интернета – по открытому каналу:

```
GlobalParameters (
)
FilteringRule filter_1 (
   PeerIPFilter* = FilterEntry (
     IPAddress = 192.168.13.0/24
   )
   LocalIPFilter* = FilterEntry (
     IPAddress = LOCAL_IP_ADDRESSES
   )
   Action* = (IPsecESP_Tunnel)
)
```



```
FilteringRule filter_2 (
   Action* = (PASS)
)

IPsecAction IPsecESP_Tunnel(
    TunnelingParameters* = TunnelEntry(
       LocalIPAddress = 10.10.10.5
       PeerIPAddress = 192.168.13.2
       DFHandling=COPY
    )
    ContainedProposals* = (ESP_GOST)
    IKERule = IKE_GOST
)
ESPProposal ESP_GOST(
    Transform* = ESPTransform(
        IntegrityAlg* = "GR341194CPRO1-H96-HMAC-254"
        CipherAlg* ="NULL"
        LifetimeSeconds = 3600
     )
)
IKERule IKE_GOST(
    Transform* = IKETransform(
        CipherAlg* = "G2814789CPRO1-K256-CBC-65534"
        HashAlg* = "GR341194CPRO1-65534"
        GroupID* = MODP_768
     )
     MainModeAuthMethod* = auth_key
     DoAutopass = TRUE
)

AuthMethodPreshared auth_key(
    RemoteID = IdentityEntry( IPv4Address* =192.168.13.2)
    SharedIKESecret = "keygost"
)
```

