УТВЕРЖДЕНО
BY.РТНК.45000 34 01-2-ЛУ

Программные продукты Bel VPN
Руководство пользователя
Форматы и расширения

BY.РТНК.45000 34 01-2

Листов 46

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

2020

Содержание

Описание формата ini-файлов

Формат допустимых строк

1. Все `ini`-файлы, используемые в продуктах Bel VPN, могут содержать следующие типы строк:
   - *Строка имени секции* – строка, обозначающая начало новой секции переменных
   - *Строка описания переменной* – строка, содержащая имя и значение переменной
   - *Строка комментариев* – строка, содержащая комментарии или пустая строка.

Строка имени секции имеет следующий формат:

```
[SECTION_NAME]
```

Пробелы и символы табуляции, стоящие до открывающей и после закрывающей квадратных скобок, игнорируются. Именем секции считается строка, помещенная между квадратными скобками (любой символ является значимым). Допускается пустое имя секции – `[]`.

Строка описания переменной имеет следующий формат:

```
var_name = var_value
```

Пробелы и знаки табуляции, стоящие до и после `var_name` и `var_value` игнорируются.

Строка комментариев имеет следующий формат:

```
! comment string
```

Пробелы и знаки табуляции, стоящие до символа `!` игнорируются.

Пустая строка (не содержащая ничего, кроме пробелов и знаков табуляции) приравнивается к строке комментария.

Любая строка, не удовлетворяющая ни одному типу описанных строк, является ошибочной, и будет приводить к ошибке чтения `ini`-файла.

1. Повторяющиеся имена секций.

В файле допустимо многократное использование *строки имени секции* с одним и тем же именем секции. В этом случае каждая последующая секция считается продолжением предыдущих (*строки описания переменных* объединяются)

2. Повторяющиеся имена переменных.

В файле допустимо многократное использование *строки описания переменной* с одним и тем же именем переменной. В том случае, если эти строки принадлежат к одной секции – действительным считается значение, описанное последней строчкой (*строки описания переменных* накладываются).

3. Переменные, не принадлежащие ни одной из секций.

В файле допускается указание *строк описания переменных*, не принадлежащих ни одной секции (в начале файла идут строки переменных без задания имени секции). Этот случай эквивалентен заданию секции с пустым именем. Следующие ситуации эквивалентны:

```
File01.ini
 Var01=value01
 Var02=value02
 Vat03=valeue03

File02.ini
 []
 Var01=value01
 Var02=value02
 Vat03=valeue03
```

4. Перезапись ini-файла.

В процессе работы некоторые `ini`-файлы могут модернизироваться продуктом. При этом, все комментарии и пустые строки будут сохранены.

- В случае повторяющихся имен секций подобные секции будут объединены в одну (первая дополняется переменными последующих). В этом случае комментарии, принадлежащие секциям, объединяются.
- В случае повторяющихся имен переменных (принадлежащих одной секции) будет оставлено только последнее ее описание. В этом случае комментарии, принадлежащие переменной, объединяются.

5. Принадлежность строк комментариев.

Любая *строка комментариев*, расположенная перед *строкой имени секции* или *строкой описания переменной*, считается принадлежащей этой строке.

Конвертор

В данной главе описана внутренняя логика работы конвертора VPN агента из Cisco-like конфигурации в Native-конфигурацию, как управлять конвертированием с помощью INI-файла, формирование имен структур при конвертировании, а также указан список сообщений, предупреждений и ошибок, которые могут быть посланы при протоколировании событий.

Основная логика работы

1. Конвертор выполнен в виде динамической библиотеки s_converter.dll (Win32) / libs_converter.so (Solaris). Также используются некоторые вспомогательные файлы и агентские библиотеки.
6. Конвертор работает в рамках программы cs_console.
7. При выходе из конфигурационного режима, если в конфигурацию были внесены какие-то изменения, cs_console вызывает конвертор и передает ему внутреннее представление Cisco-конфигурации.
8. Во время работы конвертора используются настройки конвертора, описанные в разделе . Некоторые из настроек могут редактироваться пользователем. В результате работы конвертора формируется LSP в Native-формате.
9. Логика формирования имен структур в Native-конфигурации представлена в разделе «Формирование имен структур LSP при конвертировании».
10. Далее происходит попытка загрузки LSP в Native-формате в агента.

Если по каким-то причинам произошла ошибка при загрузке, Native-конфигурация пишется в файл `erroneous_lsp.txt`, расположенный в каталоге `/var/belvpn`.

11. В конце работы выдается результат (успех/неуспех) обратно в cs_console.

Ограничения на конвертор

1. Поддерживается набор команд, определенный в документе .
12. В Cisco используется примерно следующая логика работы с access list:

`<interface_acl> -> <crypto_map_acl> -> <interface_acl>`,

где `<interface_acl>` – access list в интерфейсе,

а `<crypto_map_acl>` – access list в crypto map.

В конверторе используется логика разворачивания access list-ов в сквозную модель правил. При этом возможны некоторые несоответствия и несовместимости (подробнее см).

- В правилах в access list в маске подсети допускается указание только непрерывной линейки из установленных битов в конце (т.е. 00…01…1, например 0.0.0.255 0.0.0.63 и т.п.). Не допускается разрывов в полях установленных и сброшенных битов (например, маски вида 0.255.0.255). В случае появления запрещенной маски, конвертация завершается с ошибкой .

13. Ряд ограничений на `ca trustpoint`:

- enrollment игнорируется (только ручное задание сертификатов).
- Читаются только CA-сертификаты, локальные сертификаты игнорируются.

    - Небольшое пояснение: в Cisco по команде `crypro ca certificate chain` показываются CA сертификаты и локальные сертификаты. Через эту команду все сертификаты можно посмотреть, удалить и ввести CA сертификаты. Однако, локальные сертификаты нельзя ввести таким образом (они будут неработоспособны без секретного ключа). В `cs_console` данная команда используется только для работы с CA сертификатами.

- Под обозначением RSA-сертификатов (другие в Cisco не используются) могут использоваться RSA, СТБ и DSA-сертификаты.
- В Cisco в пределах одного `trustpoint` могут вписываться только сертификаты из одной цепочки. В `cs_converter` допускаются любые CA сертификаты.
- Задается строгое соответствие: RSA CA сертификат подписывает только RSA-сертификаты, СТБ CA сертификат подписывает только СТБ сертификаты, DSA CA сертификат подписывает только DSA-сертификаты.
- Следует учитывать, что в конфигурации не задается точных критериев выбора локального сертификата (в терминах Native LSP задается USER_SPECIFIC_DATA). В связи с этим возможны ситуации, при которых не установится соединение, если присутствуют больше одного локального сертификата, подписанного разными CA.
    - Пример подобной ситуации: у партнера не прописана посылка Certificate Request, и партнер ожидает от агента конкретный сертификат (который действительно присутствует), но агент по своим критериям выбирает другой сертификат, который не подходит партнеру.
    - Как правило, таких проблем не возникает, если соблюдаются следующие условия:
        - У обоих партнеров прописана отсылка Certificate Request. По умолчанию конвертер именно так и делает. Cisco в большинстве случаев поступает также.
        - Не используется `Aggressive Mode` при работе с сертификатами (экзотический случай).
        - У партнера должны быть явно указаны CA-сертификаты, которыми может быть подписан локальный сертификат агента. В Native LSP агента – атрибут `AcceptCredentialFrom` (cs_converter вписывает все CA-сертификаты, лежащие в базе). В Cisco – должен быть прописан подходящий `trustpoint`.

14. Ограничение на LDAP url: допускается только задание IP-адреса и, возможно, порта. Если задано DNS-name – данный url игнорируется.
15. Допускается только одно ISAKMP правило для одного IPsec-правила.
16. Если для данного crypto-map удалось подобрать несколько ISAKMP policy с разными Transform и методами аутентификации, то формируется одна IKERule, в которой пишутся ВСЕ трансформы и методы аутентификации, что приводит к несколько иной логике (т.е. теряется связь между трансформами и методами аутентификации).

Пример

#Фрагмент исходной конфигурации:

```
crypto isakmp policy 1
encr des
hash md5
authentication rsa-sig

crypto isakmp policy 2
encr 3des
hash sha
authentication pre-share
group 2
```

```
crypto isakmp policy 3
encr aes 128
hash md5
authentication rsa-sig
```

#Фрагмент Native-LSP (в ситуации, когда подходят все три policy):

```
AuthMethodRSASign auth_ca(
...
)
AuthMethodPreshared IKE_auth_key_192_168_11_110(
...
)
IKERule IKE_router_mc_fastethernet0_0_crypto_1(
    Transform* = IKETransform(
        CipherAlg   *= "DES-CBC"
        HashAlg     *= "MD5"
        GroupID     *= MODP_768
        LifetimeSeconds = 86400
    ),
    IKETransform(
        CipherAlg   *= "DES3-K168-CBC"
        HashAlg     *= "SHA1"
        GroupID     *= MODP_1024
        LifetimeSeconds = 86400
    ),
    IKETransform(
        CipherAlg   *= "AES-K128-CBC-7"
        HashAlg     *= "MD5"
        GroupID     *= MODP_768
        LifetimeSeconds = 86400
    )
    MainModeAuthMethod  *= auth_ca, IKE_auth_key_192_168_11_110
    AggrModeAuthMethod  *= auth_ca, IKE_auth_key_192_168_11_110
    DoAutopass          = TRUE
)
```

17. В фильтрах, в которых прописан локальный адрес ANY, в Native-LSP прописывается диапазон 0.0.0.0..255.255.255.255.
18. В IKERule может прописываться параметр IKEPeerIPFilter. Используются следующие правила:
    - Источник информации выбирается по приоритетам: если удается получить фильтр на основе информации из очередного пункта, следующие пункты игнорируются (например, если есть set peer, другие пункты игнорируются).
    - Источники информации для IKEPeerIPFilter:

- Команда set peer, если она присутствует в crypto map (всегда для статических crypto map; редко для динамических crypto map).
- Только для структур IKERule, не имеющих ссылок на AuthMethod...Sign (используются только preshared keys; без сертификатов): адрес из команды crypto isakmp key ... address (или ...hostname в случае, когда мы можем связать указанный hostname с IP-адресом через команду ip host).
- Только при использовании транспортного режима: permit-записи в crypto ACL.

19. Если в crypto map прописаны несколько peer, каждый из которых аутентифицируется по preshared key, то используется следующий подход:
    - прописывается туннель и аутентификация для первого по счету peer
    - для остальных peer-ов проверяются preshared keys:
        - если preshared key совпадает с ключом для первого peer, то этот peer прописывается в качестве туннеля;
        - если preshared key не совпадает с ключом для первого peer, то для данного peer формируется отдельный `AuthMethodPreshared`, `IKERule` и `IPsecAction`.

Следует учитывать, что подобная конфигурация приведет к тому, что работа со вторым peer будет возможна только в качестве ответчика. В качестве инициатора работа возможна только с первым peer.

Рекомендуется по возможности избегать таких ситуаций. Для этого, в случае указания в crypto map нескольких peers, следует либо использовать аутентификацию по сертификатам либо, в случае использования аутентификации на preshared keys, использовать одинаковый ключ для всех peers, перечисленных в одной crypto map.

В подобной ситуации выдается сообщение

Если присутствует подобная конфигурация с несовпадающими preshared ключами и, кроме того, существует аутентификация на сертификатах; тогда к вышеперечисленным наборам `AuthMethodPreshared, IKERule и IPsecAction` добавится еще один, описывающий аутентификацию на сертификатах. При этом в `IPsecAction` будут прописаны все peers.

20. Существует специфический подход в случае, если в `crypto map set` присутствует несколько `crypto maps`, а в их `crypto-map-acls` существуют пересечения по адресам, причем в части правил присутствует `permi`t, а в других правилах – `deny`. Подробнее логика конвертирования для данной ситуации описана в разделе .
21. Существуют особенности в настройке маршрутизации:

Если добавить из консоли `routing,` который уже присутствует в системной таблице маршрутизации, то он будет добавлен в текущую конфигурацию с диагностикой в файле лога.

При отгрузке сконвертированной конфигурации (по любой причине), из системной таблицы маршрутизации будут удалены все записи, добавленные из консоли, которые также могли существовавать и до запуска консоли (например, добавленные с помощью команды `route add`).

22. Существуют дополнительные команды, которые отсутствуют у Cisco:
    - команда `set pool` – задает IKE-CFG pool, привязанный к конкретной `crypto map`. Работает в конфигурационном режиме `crypto map` и `crypto dynamic-map`.

особый случай – команда `set pool <none>` – убирает для конкретной `crypto map` или `crypto dynamic-map` настройки IKE-CFG, которые, возможно, выставлены с помощью команды `crypto map client configuration address` или `crypto dynamic-map client configuration address` (они работают для `crypto map set`).

- команда `crypto dynamic-map client configuration address`. Работает аналогично команде `crypto map client configuration address`, но для `dynamic map set`.

23. Команды для задания ограничений по трафику и времени имеют больший диапазон, чем в Cisco:

- `security-association lifetime kilobytes`: в Cisco `2560-536870912`, у нас – `1-4294967295`.
- `security-association lifetime seconds`: в Cisco `120-86400`, у нас – `1-4294967295`.
- `IKE lifetime (seconds)`: в Cisco 60-86400, у нас – `1-4294967295`.

Примечание: Cisco-like консоли как и в Cisco отсутствует возможность убрать ограничения по трафику и времени (unlimited).

24. Существуют особенности при настройке шлюза для работы с мобильным клиентом. Можно использовать один из двух подходов:
    - с точки зрения логики настройки продуктов Bel VPN, в acl, привязанном к crypto dynamic map, в качестве remote-адресов для мобильных клиентов необходимо указывать any. Таким образом указывается, что допускается любой физический адрес мобильного клиента.
    - с точки зрения логики настройки Cisco, в acl, привязанном к crypto dynamic map, в качестве remote-адресов для мобильных клиентов указывается пул, из которого роутер раздает адреса мобильным клиентам. Таким образом указывается, что область действия этой crypto dynamic map распространяется только для мобильных клиентов из пула:

В сконвертированной LSP, в структуре IPsecAction в атрибуте TunnelEntry отсутствует поле PeerIPAddress, и в качестве IKE-партнера для шлюза может выступать мобильный клиент с любым физическим адресом.

В тоже время, если мобильный клиент является пассивным IKE-CFG клиентом (IKECFGRequestAddress=FALSE – не инициирует посылку запроса на получение адреса из пула), то защищенное соединение построено не будет. Присылаемый мобильным клиентом его физический адрес в качестве identity QM не подходит под соответствующее правило фильтрации и QM шлюзом отвергается.

Для успешного создания соединения при такой конфигурации шлюза мобильный клиент должен выступать в качестве активного IKE-CFG клиента (IKECFGRequestAddress=TRUE). В этом случае в качестве identity QM используется выданный клиенту адрес из пула и соответствующее правило фильтрации на шлюзе срабатывает.

25. Если задается dynamic map без указания set peer (обычная ситуация), то формируются цепочки правил для всех возможных вариантов аутентификации. Например, если заданы несколько preshared keys для разных хостов, то будут сформированы правила для всех этих preshared keys и соответствующих им хостов.

Если задается dynamic map с указанием set peer (экзотический, но допустимый вариант), то данная dynamic map конвертируется аналогично static map.

26. В TunnelingParameters прописывается значение DFHandling:
    - используется значение crypto ipsec df-bit для интерфейса, если оно присутствует в конфигурации
    - в противном случае используется глобальное значение crypto ipsec df-bit.
27. Может генерироваться несколько структур IKERule, каждая из которых подходит по присланным партнером параметрам.

Для выбора конкретного IKE-правила используется параметр Priority. Особенности его формирования:

- У каждой структуры IKERule уникальное значение параметра.
- Если структура IKERule порождена только из динамических криптокарт, значение параметра Priority гарантировано будет больше, чем у любой из структур, порожденных из статических криптокарт.
    - Присутствуют два диапазона значений: для IKERule, порожденных от динамических криптокарт; и для IKERule, порожденных от статичсеких криптокарт.

- Если структура IKERule порождена из нескольких криптокарт (происходит объединение правил), значение параметра Priority будет минимально возможным среди указанных криптокарт.
    - В частности: если данная структура IKERule порождена как из статической, так и из динамической криптокарты, то параметр Priority будет задан из диапазона значений, предназначенных для статических криптокарт.
- В описании последующих правил для простоты предполагается, что каждая криптокарта порождает отдельную структуру IKERule (не происходит объединение правил):
    - если две статические криптокарты входят в один набор, привязанный к данному сетевому интерфейсу, значение Priority криптокарты с меньшим номером будет также меньшим
    - если две статические криптокарты привязаны к разным сетевым интерфейсам, значение Priority будет меньше у криптокарты, привязанной к интерфейсу, который описан раньше в файле ifaliases.cf
    - аналогичная ситуация с двумя динамическими криптокартами.

28. Возможна конфликтная ситуация при выставлении aggressive mode:

- Имеется криптокарта, в которой прописаны два peer, для одного из которых выставлена aggressive mode, а для другого – нет. Например:

```
crypto map cmap 10 ipsec-isakmp
! ...
 set peer 192.168.10.11
 set peer 192.168.10.12
! ...
crypto isakmp peer address 192.168.10.11
 set aggressive-mode client-endpoint ipv4-address 192.168.10.11
```

- Конвертирование такой конфигурации может прерываться с ошибкой MSG_ID_CSCONV_CRYPTO_MAP_PH1_MODE_MISMATCH (0x00733108).
    - Примечание: ошибка не появляется, если в результате конвертирования для каждого из peer-ов генерируются отдельные структуры IKERule.
- Следует учесть, что данное поведение отличается от поведения конвертора версии 3.1: там в подобной ситуации выставлялся параметр AggrModePriority = TRUE.

Логика запуска конвертора

1. При старте продукта проверяются следующие изменения:
   - Добавление или удаление сертификата в базе локальных настроек.
   - Удаление preshared ключа, заданного в Cisco-like конфигурации.

Во всех этих случаях выдается сообщение в лог [] и выставляется флаг изменения Cisco-like конфигурации. В результате конвертор будет вызван при выходе из режима конфигурирования даже в том случае, если пользователь не сделал никаких изменений в Cisco-like конфигурации.

Следует отметить, что проверка на данные ситуации делается только при старте консоли. При этом проверка LSP делается при каждом выходе из режима конфигурирования (подробнее см. ниже).

29. При выходе из режима конфигурирования сначала проверяется источник текущей LSP. Возможны следующие варианты:.

Результат проверки источника текущей LSP	Предпринимаемые действия
LSP не загружена	Вызывается конвертор
Загружена LSP не из cs_converter	Вызывается конвертор. Если конвертирование и загрузка новой LSP прошли успешно, то предыдущая LSP сохраняется в файл /var/belvpn/non_cscons.lsp. Об этом выдается сообщение в лог (см., если по каким-либо причинам не удалось сохранить файл – см. сообщение)
Загружена LSP из cs_converter	Проверяется флаг изменения Cisco-like конфигурации (он мог быть выставлен на старте продукта, см. 1; либо если пользователь внес изменения в Cisco-like конфигурацию в данном сеансе). Если флаг выставлен, вызывается конвертор. В противном случае ничего не делается.

30. Если при выходе из конфигурационного режима происходит конвертирование конфигурации, и это конвертирование завершается с ошибкой; то на консоль выдается сообщение об ошибке: "`LSP conversion failed. You can use the "show load-message" command to obtain the additional information.`" ("Конвертирование LSP завершилось с ошибкой. Вы можете использовать команду "`show load-message`" для получения дополнительной информации").
31. Если конвертирование завершилось успешно, но с предупреждениями MSG_ID_CSCONV_LSP_LOAD_WITH_ONE_WARNING (00734101) или MSG_ID_CSCONV_LSP_LOAD_WITH_WARNINGS (00734102), сигнализирующими о том, что при загрузке сгенерированной LSP были выданы предупреждения (со стороны демона vpnsvc), то на консоль выдается сообщение вида:

LSP conversion complete with additional message(s):
LSP successfully loaded with warning: <LSP_load_warning>

где <LSP_load_warning> – предупреждение со стороны демона vpnsvc по поводу загруженной LSP.

- Если вместе с указанными предупреждениями присутствуют другие предупреждения, относящиеся непосредственно к конвертору, они также будут выданы на консоль.

- Все выданные на консоль предупреждения можно повторно выдать на консоль по команде show load-message.

32. Если конвертирование завершилось успешно, но с предупреждениями, относящимися непосредственно к конвертору (отсутствуют предупреждения, перечисленные в предыдущем пункте), то на консоль не выдается никакой дополнительной информации. Однако по команде `show load-message` будут выданы все предупреждения конвертора.
33. При успешном конвертировании конфигурации флаг изменения Cisco-like конфигурации сбрасывается.

- При повторном входе в режим конфигурирования конвертор не будет запущен, если перед этим не было внесено каких-либо изменений в Cisco-like конфигурацию.

Алгоритм работы конвертора

1. Подготовительный этап:

Инициализация лога.

Инициализация локальных настроек.

34. Написание служебной информации в комментариях:

Фраза `"This is automatically generated LSP"`.

Дата и время конвертации.

35. Создание заголовка конфигурации:

Служебная информация (версия и т.п.).

Настройки LDAP и CRL-processing.

Глобальные настройки лога.

36. Обработка интерфейсов. Подробнее см. . При этом формируются правила фильтрации, в том числе и IPsec (APPLY).
37. Для APPLY правила происходит поиск подходящего ISAKMP правила:

Сначала делается попытка найти подходящее правило на `Preshared key`.

Также берется первое по счету правило `rsa-sig`.

38. Если подобрано правило на `Preshared key` – прописывается аутентификационная информация с соответствующим именем ключа.

Для `main mode LocalID` прописывается в зависимости от команды `crypto isakmp identity`:

- Если `crypto isakmp identity address` (вариант по умолчанию), а также в случае `crypto isakmp identity dn` (вариант, неприменимый для `preshared keys`), `LocalID` в конфигурацию не пишется (обозначает – использовать локальный IP-адрес).
- Если `crypto isakmp identity hostname`, пишется:
  `LocalID = IdentityEntry( KeyID *= "<local_id>" )`
  где `<local_id>` – представление в виде Hex-string полного DNS-адреса, составленного из локального `hostname`, заданного с помощью команды `hostname <…>` и доменного имени, заданного с помощью команды `ip domain name <…>`. Если доменное имя отсутствует, или в `hostname` присутствует хотя бы одна точка, `<local_id>` составляется только из `hostname`.

`RemoteID` прописывается по следующим правилам:

- Если ключ привязан к IP-адресу с помощью команды `crypto isakmp key <…> address <…>`,

то формируется правило с аутентификацией по данному ключу:

```
AuthMethodPreshared <...> (
    [ LocalID  = IdentityEntry( KeyID *= "<local_id>" ) ] –
    если необходимо
    RemoteID = IdentityEntry( IPv4Address *= <peer_ip> )
    SharedIKESecret = <...>
)
```

- Если ключ привязан к `hostname` с помощью команды `crypto isakmp key <…> hostname <…>`, а `hostname` привязан к IP-адресу с помощью команды `ip host <hostname> <ip-addr>`, то формируется правило с аутентификацией по `hostname и IP-address`:

```
AuthMethodPreshared <...> (
    [ LocalID  = IdentityEntry( KeyID *= "<local_id>" ) ] –
    если необходимо
    RemoteID = IdentityEntry(
    IPv4Address *= <peer_ip> KeyID *= "<peer_id>" )
    SharedIKESecret = <...>
)
```

где `<peer_id>` – представление в виде Hex-string hostname-а из команды `crypto isakmp key <…> hostname <…>`.

- Если ключ привязан к `hostname` с помощью команды `crypto isakmp key <…> hostname <…>` и используется динамический `crypto map`, формируется правило с аутентификацией по `hostname`:

```
AuthMethodPreshared <...> (
    [ LocalID  = IdentityEntry( KeyID *= "<local_id>" ) ] –
    если необходимо
    RemoteID = IdentityEntry( KeyID *= "<peer_id>" )
    SharedIKESecret = <...>
)
```

- Если удается подобрать дополнительные IP-адреса и hostname, для которых задаются те же самые ключи, тогда эти IP-адреса и KeyID (из hostname-ов) добавляются в RemoteID.

39. Если подобрано правило `RSA sig` – прописывается правило аутентификации в виде:

```
{ AuthMethodRSASign | AuthMethodDSSSign | AuthMethodGostSign } auth_ca(
    LocalID        =  IdentityEntry( ... )
    [ RemoteID       =  IdentityEntry( ... ) ] |
    [ DoNotMapRemoteIDToCert = TRUE ]
    AcceptCredentialFrom      *= CertDescription(
       X509IssuerDN  *= "..."
       SerialNumber  =  "..."
       X509SubjectDN *= "..."
    )
    SendRequestMode =  { AUTO | NEVER | ALWAYS }
    SendCertMode    =  { AUTO | NEVER | ALWAYS }
)
ModeAuthMethod *= auth_ca
```

- `LocalID для main mode` формируется по следующим правилам:

- Если `crypto isakmp identity hostname`, пишется:
  `LocalID = IdentityEntry( FQDN* = USER_SPECIFIC_DATA )`
- Если `crypto isakmp identity dn:`
  `LocalID =IdentityEntry( DistinguishedName* = USER_SPECIFIC_DATA )`
- Если `crypto isakmp identity address:`
  `LocalID = IdentityEntry(IPv4Address* = USER_SPECIFIC_DATA)`

- `RemoteID` формируется по следующим правилам:
  - Если к crypto map привязана identity, в которой прописан один или несколько dn, пишется:
    ```
    RemoteID = IdentityEntry (
      DistinguishedName* = CertDescription(Subject* =“<dn1>”),
                           CertDescription( Subject*
    =“<dn2>”), ...
      FQDN* = “<fqdn1>”, “<fqdn2>” ...
    )
    ```
  - Если к crypto map не привязана `identity`, то пишется
    `DoNotMapRemoteIDToCert = TRUE`

    данная логика предназначена для того, чтобы в качестве remote identity работал IP-адрес и в том случае, когда он отсутствует в сертификате (типичная ситуация). Это бывает полезно при использовании разных типов аутентификации в пределах одной конфигурации – на сертификатах и на preshared keys.
- Если в LSP уже присутствует идентичная структура `AuthMethod…Sign`, новая структура не пишется. В структуре `IKERule` пишется ссылка на уже существующую структуру.

40. В зависимости от наличия команды `crypto isakmp keepalive` и ее параметров в `IKERule` прописываются настройки `DPD`:

Если команда `crypto isakmp keepalive` не задана (по умолчанию), прописывается `DoNotUseDPD = TRUE`.

Если команда задана, пишутся параметры:

```
DPDIdleDuration = <secs>
DPDResponseDuration = <retries>
DPDRetries = <dpd_retries>
```

где `<secs>` – первый аргумент команды;
`<retries>` – второй аргумент команды;
`<dpd_retries>` – параметр `dpd_retries` из файла `cs_conv.ini`.
См. описание настройки `_retries`.

Внимание!!! Поведение по умолчанию отличается от настроек DPD в версии 2.0. Там всегда (вне зависимости от `crypto isakmp keepalive`) выставлялись настройки DPD по умолчанию, характерные для Native-LSP: `DPDIdleDuration = 60; DPDResponseDuration = 5; DPDRetries = 3`.
Сейчас по умолчанию DPD выключен.

41. Если в LSP уже присутствует идентичная структура `IKERule`, новая структура не пишется. В структуре `IPsecAction` пишется ссылка на уже существующую структуру.
42. Если в LSP уже присутствует идентичная структура `IPsecAction`, новая структура не пишется.
43. В конце конвертирования делается попытка загрузить сформированную конфигурацию.

Если конфигурацию не удалось загрузить, она сохраняется в файле `erroneous_lsp.txt` (с выдачей сообщения в лог). Файл расположен в каталоге `/var/belvpn`.

Внутренние настройки консоли и конвертора

1. Внутренние настройки конвертора хранятся в файле cs_cons_reg.ini, расположенном в каталоге агента.

44. Данный файл используется для хранения внутренних настроек консоли и конвертора. Он автоматически модифицируется при запуске консоли. Редактирование этого файла вручную не рекомендуется.

45. Если файл отсутствует на момент старта консоли, он автоматически создается.

46. Формат файла:

    - Обычный текстовый файл в кодировке ASCII. Используется кодирование окончания строк принятое для операционной системы (Windows/UNIX).
    - Пустые строки и строки, начинающиеся с `!` (восклицательный знак) игнорируются.
    - Файл состоит из секций. Каждая секция начинается с названия секции, заключенного в квадратные скобки.

47. В настоящее время присутствует одна секция: описание перекодировки интерфейсов из формата Cisco в Native формат агента:

Название секции: `[interface_list]`

Формат строк: <Native_interface_name> = <Cisco_interface_name>. Например: `I0 = 0/0`

Данная секция редактируется автоматически при старте консоли:

- Если в Cisco-like конфигурации и в INI-файле не описан native interface (например, первый старт консоли, то этому интерфейсу присваивается свободное <Cisco_interface_name>. Этот интерфейс добавляется в Cisco-like конфигурацию и в INI-файл.
- Если здесь описан native interface, который отсутствует в агенте, то этот интерфейс удаляется как из INI-файла, так и из Cisco-like конфигурации.
- Если в текущей Cisco-like конфигурации присутствует интерфейс, не описанный в INI-файле (нештатная ситуация), этот интерфейс удаляется из Cisco-like конфигурации.
- Если в INI-файле присутствует интерфейс, которого нет в текущей Cisco-like конфигурации (нештатная ситуация), этот интерфейс удаляется из INI-файла.

48. В случае, если произошли какие-либо изменения, описанные в предыдущем пункте, то обновленный файл сохраняется. Если сохранение по тем или иным причинам не удалось, то в лог выдается сообщение об ошибке .

Управление конвертором с помощью INI-файла

1. Настройки конвертора хранятся в INI-файле `cs_conv.ini`, расположенном в каталоге агента.
2. INI-файл хранит служебную информацию, необходимую для конвертора, включающую в себя все пользовательские настройки.
3. Формат файла:
    - обычный текстовый файл в кодировке ASCII. Используется кодирование окончания строк, принятое для операционной системы (Windows/UNIX)
    - пустые строки и строки, начинающиеся с ! (восклицательный знак) игнорируются
    - файл состоит из нескольких секций. Каждая секция начинается с названия секции, заключенного в квадратные скобки. Например: `[interface_list]`.

49. Описание секций файла:

- Описание отдельных глобальных настроек:

Название секции: `[global_settings]`

- Формат строк:

`ike_autopass = {on|off}` – включение/выключение прописывания ike autopass в конфигурации. По умолчанию – `on`. Пользователю редактировать данный параметр не рекомендуется, за исключением случаев, когда надо построить конфигурацию с IPsec
Следует учесть, что данная настройка отличается от похожей настройки для версии 3.1: там она открывала полный доступ для IKE пакетов. В текущей версии данная настройка обеспечивает прохождение IKE пакетов в обход IPsec policy, но не в обход правил firewall. Необходимо следить, чтобы были прописаны соответствующие правила пакетной фильтрации для беспрепятственного прохождения IKE пакетов.

`dpd_retries = {1-10}` – количество попыток проведения DPD-обмена. По умолчанию – 5. Пользователь может настраивать данный параметр для получения оптимального количества DPD-retries.

`tunnel_local_ip = {on|off}` – включение/выключение прописывания локального IP-адреса в структуре `TunnelEntry`. По умолчанию – `off`. Пользователю редактировать данный параметр не рекомендуется: только при возникновении ситуаций, когда прописывание локального адреса необходимо (пока не выявлено).

`ipsec_reassemble_ip = {on|off}` – включение/выключение прописывания параметра `Assemble=TRUE` в структуре `TunnelEntry`. По умолчанию – on. Значение `on` означает, что пакет будет собран из IP-фрагментов перед IPsec инкапсуляцией (как и в предыдущих версиях шлюза безопасности).
При значении `off` могут появиться проблемы при работе по защищенному соединению с предыдущими версиями агента. Если защищенная связь с предыдущими версиями шлюза безопасности не требуется, то данный параметр можно отключить для улучшения производительности IPsec.

`preserve_ipsec_sa = {on|off}` – включение/выключение прописывания параметра `PreserveIPsecSA=TRUE` в структуре `GlobalParameters`. По умолчанию – `off`. Значение `off` – при любых изменениях в конфигурации IPsecAction, IKERule фильтров NetworkInterface.IPsecPolicy удаляются все IPsec SA. Значение `on` означает, что в момент изменения конфигурации IPsec SA будет сохранен при соблюдении некоторых условий (подробнее см. документ).
Внимание! При использовании данного режима следует соблюдать осторожность: IPsec SA, оставшиеся от старой конфигурации в той или иной мере могут нарушать новую политику безопасности или быть неработоспособными из-за несоответствия новой LSP. Если нет уверенности в корректности использования данной настройки, рекомендуется оставить значение `off`.

Описание перекодировки алгоритмов:
Название секции: `[algorithm_list]`
Редактирование пользователем данной секции не рекомендуется.

- Формат строки:

```
<Generic_algorithm_name> = { SignByCaType | native:<Native_algorithm_name> }
gui:<GUI_name> [gui_default] [optional:<optional_part>] [show:"<Show_description>"]
[default] console:<console_description> [console2:<console2_description>]
```

где
<Generic_algorithm_name> – имя метода аутентификации, алгоритма или группы в стиле cs_console..

Формат <Generic_algorithm_name>:
Сначала идет префикс:

Префикс	Режим конфигурирования	Команда или префикс IPsec transform
ike-auth-	IKE policy	auth
ike-hash-		hash
ike-cipher-		encr
ike-group-		group
ah-integrity-	IPsec transform	ah-

esp-integrity-		esp-
esp-cipher-		esp-
pfs-group-	Crypto map config	set pfs

После префикса идет параметр команды или суффикс IPsec transform. Примеры:

команда hash md5 – `<Generic_algorithm_name>=ike-hash-md5`

описание алгоритма в IPsec transform esp-3des – `<Generic_algorithm_name>=esp-cipher-3des`
(префикс “esp-cipher-“ берется по типу алгоритма в стиле Cisco-like, суффикс “3des” берется из синтаксиса “esp-3des”).

Если присутствует символ _ (подчеркивание), то он обозначает, что данный алгоритм задается двумя словами.
Пример:
команда “encr aes 192” – `<Generic_algorithm_name>=ike-cipher-aes_192`
Существует требование: `<Generic_algorithm_name>=“esp-null”` обязательно должно обозначать ESP CipherAlg=”NULL” (задается стандартным образом, пример см. ниже). На другие имена жесткие требования по соответствию алгоритмов отсутствуют.

SignByCaType – зарезервированное слово, обозначающее режим выбора метода аутентификации в зависимости от типа CA. Используется только для описания метода аутентификации. Является альтернативой для параметра <Native_algorithm_name>. Подробнее по логике работы с сертификатными методами аутентификации см. ниже.

<Native_algorithm_name> – имя метода аутентификации, алгоритма или группы в стиле LSP. Не может содержать кавычек, пробелов и табуляций. Параметр обязательный, но может отсутствовать, если задано ключевое слово SignByCaType. Если ключевое слово SignByCaType не используется, то метод аутентификации описывается именем структуры AuthMethod. Остальные значения задаются строковым значением соответствующего параметра в LSP.

<GUI_name> – идентификатор, представляющий данный алгоритм в GUI. Не может содержать кавычек, пробелов и табуляций.

gui_default – спецификатор, отмечающий значение, которое в GUI выбирается как значение по умолчанию при создании новой ISAKMP Policy или IPsec Transform Set.

Optional – после ключевого слова optional и двоеточия пишется опциональная часть команды, которая может быть введена или опущена при вводе. При выводе по команде show running-config данная часть команды никогда не показывается. Не может содержать кавычек, пробелов и табуляций.

Пример таких команд:

```
encr aes [128]
crypto ipsec transform-set <...> esp-aes [128]
```

Опциональная часть “`128`” может вводиться или нет. При выводе по show running-config “`128`” никогда не выводится.

<Show_description> – текстовое описание метода аутентификации, алгоритма или группы, которое показывается по команде show crypto isakmp

policy. Применимо для ISAKMP алгоритмов и групп. Обязательно должно быть заключено в кавычки. Может содержать пробелы или табуляции. Не может содержать кавычки внутри.

Для группы описание должно начинаться с одного из следующих текстов (дефис здесь обозначает пробел):

для Diffie-Hellman групп:
`Diffie-Hellman group:---`

для групп, отличных от Diffie-Hellman (например, VKO):
`Oakley group:-----------`

default – спецификатор используется для ISAKMP алгоритмов и групп. Он обозначает значения по умолчанию для объекта crypto isakmp policy. Также спецификатор применяется для PFS групп: в этом случае имеет специальный смысл – значение, выставляемое командой “set pfs” без параметров. Всего в файле должны быть четыре записи со спецификатором default: ISAKMP hash, encryption, group, а также PFS group.

Появление лишних записей default или отсутствие ее для какого-то из типов алгоритмов рассматривается как ошибка – испорченный ресурсный файл (“ERROR: Could not initialize resources.”). Консоль не запускается.(подробнее см. документ)

<console_description> – текстовое описание метода аутентификации, алгоритма или группы в cs_console, которое показывается при вызове подсказки во время редактирования конфигураци. Обязательно должно быть заключено в кавычки. Может содержать пробелы или табуляции. Не может содержать кавычки внутри.

<console2_description> – текстовое описание второй части алгоритма в cs_console. Небязательный параметр, используется для AES алгоритмов. Например:
esp-cipher-aes_256 = native:AES-K256-CBC-12 gui:"ESP AES 256" console:"ESP transform using AES cipher" console2:"256 bit keys."

Список поддерживаемых в настоящее время значений:

ike-auth-belt-sig
ike-auth-pre-share
ike-auth-rsa-sig
ike-auth-dss-sig
ike-auth-sign
ike-hash-stb1176
ike-hash-belt
ike-cipher-gost28147-cbc
ike-cipher-belt
ike-hash-sha
ike-hash-md5
ike-cipher-des
ike-cipher-3des
ike-cipher-aes
ike-cipher-aes_192
ike-cipher-aes_256
ike-group-1
ike-group-2
ike-group-beltdh
ike-group-5
ah-integrity-stb1176
ah-integrity-belt-mac
ah-integrity-gost28147-mac
ah-integrity-sha-hmac

ah-integrity-md5-hmac
esp-integrity-stb1176
esp-integrity-gost28147-mac
esp-integrity-belt-mac
esp-integrity-sha-hmac
esp-integrity-md5-hmac
esp-cipher-gost28147-cbc
esp-cipher-belt
esp-cipher-des
esp-cipher-3des
esp-cipher-aes
esp-cipher-aes_192
esp-cipher-aes_256
esp-cipher-null
pfs-group-group1
pfs-group-group2
pfs-group-beltdh
pfs-group-group5

- Описание логики работы с `Cert request` и посылки сертификатов в процессе IKE:

Название секции: `[auth_cert]`

Редактирование пользователем данной секции, как правило, не требуется, но возможно, при необходимости, изменить логику работы с Cert request и посылки сертификатов в процессе IKE.

- Формат строк:

  `<param_name>=<param_val>`

  где
  `<param_name>`:

  `send_request`. По умолчанию `<param_val>`= ALWAYS.
  `send_cert`. По умолчанию `<param_val>`= ALWAYS.

- Описание переменных окружения, выставляемых для процесса cs_console:

Название секции: `[env]`

- Формат строк: `<env_var_name>=<env_var_val>`

  Можно задавать любые перемнные окружения. На практике, как правило, нужно для выставления переменной PATH.

- Редактирование пользователем данной секции, как правило, не требуется, но возможно при необходимости изменить переменную PATH или добавить какие-то свои переменные окружения (м.б. полезно для команды run).

50. Варианты файлов, поставляемых в составе продукта:

Пример INI-файла **cs_conv.ini** для Bel VPN Gate; стандартный режим; поставляется в составе продукта; вариант исполнения Linux):

```
[global_settings]
ike_autopass        = on
dpd_retries         = 5
tunnel_local_ip     = off
ipsec_reassemble_ip = on
preserve_ipsec_sa   = off

[algorithm_list]
```

```
ike-auth-belt-sig   = native:AuthMethodBELTSign gui:"STB 1176.2 /
34.101.45 Signature" gui_default show:"STB 1176.2 / 34.101.45 Signature"
default console:"STB 1176.2 / 34.101.45 Signature"

ike-auth-pre-share  = native:AuthMethodPreshared gui:"Pre-Shared Key"
show:"Pre-Shared Key" console:"Pre-Shared Key"

ike-auth-rsa-sig    = native:AuthMethodRSASign gui:"RSA Signature"
show:"Rivest-Shamir-Adleman Signature" console:"Rivest-Shamir-Adleman
Signature"

ike-auth-dss-sig    = native:AuthMethodDSSSign gui:"DSS Signature"
show:"Digital Signature Standard" console:"DSS Signature"

ike-auth-sign       = SignByCaType gui:"Signature selected by CA type"
show:"Signature selected by CA type" console:"Signature selected by CA
type"

ike-hash-stb1176    = native:STB1176199-65530 gui:"STB 1176.1" show:"STB
1176.1" console:"STB 1176.1 Hash"

ike-hash-belt       = native:STB34101HASH-65532 gui:"STB 34.101.31"
gui_default show:"STB 34.101.31" default console:"STB 34.101.31 Hash"

ike-cipher-gost28147-cbc    = native:G2814789AV1-K256-CBC-65530 gui:"GOST
28147-89" show:"GOST 28147-89" console:"GOST - GOST 28147-89 Encryption."

ike-cipher-belt     = native:STB34101CIPH-K256-CBC-65532 gui:"STB
34.101.31" gui_default show:"STB 34.101.31" default console:"STB
34.101.31 Encryption."

ike-hash-sha        = native:SHA1 gui:"SHA1" show:"Secure Hash Standard"
console:"Secure Hash Standard"

ike-hash-md5        = native:MD5 gui:"MD5" show:"Message Digest 5"
console:"Message Digest 5"

ike-cipher-des      = native:DES-CBC gui:"DES" show:"DES - Data
Encryption Standard (56 bit keys)." console:"DES - Data Encryption
Standard (56 bit keys)."

ike-cipher-3des     = native:DES3-K168-CBC gui:"3DES" show:"Three key
triple DES" console:"Three key triple DES"

ike-cipher-aes      = native:AES-K128-CBC-7 gui:"AES 128" optional:128
show:"AES - Advanced Encryption Standard (128 bit keys)." console:"AES -
Advanced Encryption Standard." console2:"128 bit keys."

ike-cipher-aes_192  = native:AES-K192-CBC-7 gui:"AES 192" show:"AES -
Advanced Encryption Standard (192 bit keys)." console:"AES - Advanced
Encryption Standard." console2:"192 bit keys."

ike-cipher-aes_256  = native:AES-K256-CBC-7 gui:"AES 256" show:"AES -
Advanced Encryption Standard (256 bit keys)." console:"AES - Advanced
Encryption Standard." console2:"256 bit keys."

ike-group-1         = native:MODP_768 gui:"D-H Group 1 (768-bit modp)"
show:"Diffie-Hellman group:   #1 (768 bit)" console:"Diffie-Hellman group
1"

ike-group-2         = native:MODP_1024 gui:"D-H Group 2 (1024-bit modp)"
show:"Diffie-Hellman group:   #2 (1024 bit)" console:"Diffie-Hellman
group 2"

ike-group-beltdh    = native:BELTDH gui:"STB 34.101.66" gui_default
show:"Oakley group:           STB 34.101.66" default console:"STB
34.101.66"

ike-group-5         = native:MODP_1536 gui:"D-H Group 5 (1536-bit modp)"
show:"Diffie-Hellman group:   #5 (1536 bit)" console:"Diffie-Hellman
group 5"

ah-integrity-stb1176    = native:STB1176199-H96-HMAC-250 gui:"AH STB
1176.1 HMAC96" console:"AH STB 1176.1 HMAC96 transform"

ah-integrity-belt-mac   = native:STB34101CIPH-K256-MAC-252 gui:"AH STB
34.101.31 MAC" console:"AH STB 34.101.31 MAC transform"
```

```
ah-integrity-gost28147-mac = native:G2814789AV1-K256-MAC-251 gui:"AH GOST
28147-89 MAC" console:"AH GOST 28147-89 MAC transform"

ah-integrity-sha-hmac   = native:SHA1-H96-HMAC gui:"AH SHA HMAC"
console:"AH-HMAC-SHA transform"

ah-integrity-md5-hmac   = native:MD5-H96-HMAC gui:"AH MD5 HMAC"
console:"AH-HMAC-MD5 transform"

esp-integrity-stb1176   = native:STB1176199-H96-HMAC-65530 gui:"ESP STB
1176.1 HMAC96" console:"ESP transform using STB 1176.1 HMAC96 auth"

esp-integrity-gost28147-mac = native:G2814789AV1-K256-MAC-65531 gui:"ESP
GOST 28147-89 MAC" console:"ESP transform using GOST 28147-89 MAC auth"

esp-integrity-belt-mac  = native:STB34101CIPH-K256-MAC-65532 gui:"ESP STB
34.101.31 MAC" console:"ESP transform using STB 34.101.31 MAC auth"

esp-integrity-sha-hmac  = native:SHA1-H96-HMAC gui:"ESP SHA HMAC"
console:"ESP transform using HMAC-SHA auth"

esp-integrity-md5-hmac  = native:MD5-H96-HMAC gui:"ESP MD5 HMAC"
console:"ESP transform using HMAC-MD5 auth"

esp-cipher-gost28147-cbc    = native:G2814789AV1-K256-CBC-250 gui:"ESP
GOST 28147-89" console:"ESP transform using GOST 28147-89 cipher"

esp-cipher-belt         = native:STB34101CIPH-K256-CBC-252 gui:"ESP STB
34.101.31" gui_default console:"ESP transform using STB 34.101.31 cipher"

esp-cipher-des      = native:DES-CBC gui:"ESP DES" console:"ESP transform
using DES cipher (56 bits)"

esp-cipher-3des     = native:DES3-K168-CBC gui:"ESP 3DES" console:"ESP
transform using 3DES(EDE) cipher (168 bits)"

esp-cipher-aes      = native:AES-K128-CBC-12 gui:"ESP AES 128"
optional:128 console:"ESP transform using AES cipher" console2:"128 bit
keys."

esp-cipher-aes_192  = native:AES-K192-CBC-12 gui:"ESP AES 192"
console:"ESP transform using AES cipher" console2:"192 bit keys."

esp-cipher-aes_256  = native:AES-K256-CBC-12 gui:"ESP AES 256"
console:"ESP transform using AES cipher" console2:"256 bit keys."

esp-cipher-null     = native:NULL gui:"ESP NULL" console:"ESP transform
w/o cipher"

pfs-group-group1    = native:MODP_768 gui:"D-H Group 1 (768-bit modp)"
console:"D-H Group1 (768-bit modp)"

pfs-group-group2    = native:MODP_1024 gui:"D-H Group 2 (1024-bit modp)"
console:"D-H Group2 (1024-bit modp)"

pfs-group-beltdh    = native:BELTDH gui:"STB 34.101.66" gui_default
default console:"STB 34.101.66"

pfs-group-group5    = native:MODP_1536 gui:"D-H Group 5 (1536-bit modp)"
console:"D-H Group5 (1536-bit modp)"


[auth_cert]

send_request            = ALWAYS

send_cert               = ALWAYS


[env]

PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin
```

Сообщения в логе при конвертировании

При работе конвертора могут посылаться сообщения в файл лога.

Формат строк сообщений:

```
<Date_Time> <Level:> <Message>, где Level – INFO, Warning или ERROR.
```

Пример сообщения:

```
Wed Oct 29 18:19:50 2003 INFO: LSP conversion complete. Warnings: 2
```

Список сообщений, предупреждений и ошибок, передаваемых по протоколу Syslog, представлен в таблице.

1. **Информационные сообщения**

	Сообщение	Комментарий
1.1	LSP conversion started	Начат процесс конвертирования
1.2	LSP conversion complete	Процесс конвертирования завершен успешно. Предупреждения не выдавались.
1.3	LSP conversion complete. Warnings: {1}	Процесс конвертирования завершен успешно. Выдано {1} предупреждений.
1.4	Previous user-defined LSP saved in file "{1}"	Предыдущая пользовательская LSP сохранена в файле “{1}”
1.5	Non-synchronized policy detected. Policy type: <type> где <type> один из: Can’t obtain DDP Drop All User-defined (Source: <source>), где <source>: Command-line utility Unknown (<n>).	Обнаружена несинхронизированная политика. Тип политики: <type> Примечание: <type>=”Can’t obtain” – обозначает, что в данный момент невозможно определить тип LSP (обычно не удается установить связь с сервисом vpnsvc (например, сервис был остановлен в процессе работы cs_console)) <type>=“User-defined (Source: Unknown(<n>))”, где <n> – внутренний номер источника, в нормальной ситуации появляться не должен (следует рассматривать как ошибку продукта).
1.6	Certificates or preshared keys were changed. Conversion required.	Сертификаты или Preshared keys были изменены. Требуется конвертирование. Примечание: может быть выдано при старте cs_console. Если в данной ситуации войти в режим конфигурирования, а затем выйти из него, не внеся никаких изменений в конфигурацию, то конвертор все равно будет вызван.

51. **Предупреждения**

	Сообщение	Комментарий
2.1	LDAP url "{1}" ignored. IP address and port allowed only.	Введенный LDAP url {1} проигнорирован, поскольку допускаются только IP-адрес и порт.
2.2	OUT access group in the interface "{1}" ignored. Only IN access group is used.	Проигнорирован access-group out в интерфейсе {1}, поскольку допускается только access-group in.

	Сообщение	Комментарий
2.3	Only one interface is used while host mode is on. Other interfaces ignored.	При включенном Host-режиме допускается только один интерфейс. Остальные интерфейсы игнорируются.
2.4	Only one CA certificate imported. Other certs ignored.	Импортирован только первый по списку CA-сертификат. End-User сертификаты и оставшиеся CA-сертификаты проигнорированы.
2.5	Crypto map "{1}" contains several peers. Peer(s) "{2}" ignored due to authentication information mismatch.	В crypto map {1} прописаны несколько peer-ов. Peer(s) {2} проигнорированы из-за того, что для них не совпадает аутентификационная информация.
2.6	Crypto map(s) "{1}" contain transform sets with different encapsulation modes. Tunnel mode is used.	Crypto maps {1} содержат transform sets, в которых заданы разные encapsulation режимы. Используется туннельный режим.
2.7	Crypto map(s) "{1}" contain transform sets with different encapsulation modes. Transport mode is used.	Crypto maps {1} содержат transform sets, в которых заданы разные encapsulation режимы. Используется транспортный режим.
2.8	Incorrect config detected. Policy conversion ignored	Обнаружена некорректная политика. Конвертирование политики не делается.
2.9	Crypto map set(s) "{1}" contain static crypto map(s) with priorities lower than dynamic.	Crypto map set(s) {1} содержат статические crypto map(s) с приоритетом ниже, чем у динамических
2.10	Crypto map "{1}" contains several peers with different preshared keys. This is not recommended.	Crypto map {1} содержит несколько peers с разными preshared keys. Это не рекомендуемая ситуация. Подробнее см. ”Ограничения на конвертер” Preshared keys.

52. **Ошибки**

	Сообщение	Комментарий
3.1	No interfaces were found in the INI file. Configure interfaces or set host mode to proceed.	Не заданы интерфейсы в INI-файле при выключенном Host-режиме. Необходимо настроить интерфейсы или включить Host-режим.
3.2	No interfaces were found in the configuration.	В импортируемой конфигурации не заданы интерфейсы. Конвертирование не имеет смысла.
3.3	Interface “{1}” not found in the INI file.	Интерфейс {1} не задан в INI-файле.
3.4	Interface "{1}" references to the empty access list "{2}".	Интерфейс “{1}” ссылается на пустой ACL “{2}”
3.5	Certificate parse failed	Не удалось разобрать введенный сертификат.
3.6	Could not convert {crypto map | dynamic crypto map template} “{1}”. Reason: <Reason> где <Reason>:	Невозможно сконвертировать crypto map (или dynamic crypto map template) “{1}”. Причина: <Причина> где <Причина> одна из:

	Сообщение	Комментарий
	There is no isakmp policy. There is no CA or appropriate preshared key. Also isakmp policy can have wrong type (rsa-sig or pre-share). There is no peer. There are no transform sets. Crypto map is incomplete. Reference to the empty access list "{2}" for a clear-text packets filtration. {crypto map | dynamic crypto map template} "<name2> <idx2>" references to the same pool ("<pool-name>") but the additional parameters to send to client are different. Unknown.	Отсутствует isakmp policy. Отсутствует CA или подходящий Preshared Key, либо isakmp policy неправильного типа (rsa-sig или pre-share). Отсутствует peer. Отсутствуют transform sets Crypto map неполная (не хватает crypto ACL, transform set или peer). Ссылка на пустой ACL “{2}” для Clear-Text фильтрации (внутри защищенного туннеля). crypto map (или dynamic crypto map template) "<name2> <idx2>" ссылается на тот же самый пул ("<pool-name>"), но дополнительные параметры, отсылаемые клиенту, различаются. Неизвестная причина.
3.7	LSP load failed	Не удалось загрузить сформированную LSP
3.8	Unsupported network wildcard "{1}"	Не поддерживается данный формат маски подсети
3.9	LSP conversion failed	Произошла некоторая невыясненная ошибка
3.10	Could not save previous user-defined LSP in file "{1}"	Не удалось сохранить предыдущую пользовательскую LSP в файл {1}
3.11	Could not save internal settings in file "{1}"	Не удалось сохранить внутренние настройки в файл {1}
3.12	Address pool "{1}" not found.	Не найден пул адресов “{1}”.

Описание обработки интерфейсов

В результате обработки интерфейсов в LSP могут появиться записи двух структур: `NetworkInterface` и `FilterChain`.

Структура `NetworkInterface` используется для описания сетевого интерфейса и тех действий, которые должны быть выполнены с пакетом при его прохождении через этот интерфейс – фильтрация и классификация.

Структура `FilterChain` формирует условие срабатывания конкретного правила пакетной фильтрации для партнеров по взаимодействию (для crypto-maps).

1. Формирование `NetworkInterface`:
   - Пишется параметр `LogicalName`=”<native-interface-name>”, где <native-interface-name> – внутреннее имя интерфейса, зарегистрированное в агенте
   - Описание остальных параметров (если источник для параметра отсутствует в Cisco-like конфигурации – параметр не пишется):

Параметр	Источник в Cisco-like конфигурации (подкоманда команды interface)
InputFilter – задает правила как stateless (пакетной) так и statefull (контекстной) фильтрации для входящих пакетов на этот интерфейс	`ip access-group in` при ее отсутствии - ip inspect inspection-name in
OutputFilter – задает правила как stateless (пакетной) так и statefull (контекстной) фильтрации для исходящих пакетов с этого интерфейса	`ip access-group out` при ее отсутствии - `ip inspect inspection-name out`
InputClassification – задает правила классификации входящих пакетов на этот интерфейс и выставления значения поля TOS в IP-заголовке пакетов. Классификация и маркирование входящего защищаемого трафика производится после его успешной декапсуляции..	`service-policy input`
OutputClassification – задает правила классификации исходящих пакетов с этого интерфейса и выставления значения поля TOS в IP-заголовке пакетов. Классификация и маркирование исходящего защищаемого трафика производится до его IPsec инкапсуляции. В случае туннелирования результирующее значение TOS-байта копируется из внутреннего IP-заголовка во внешний.	`service-policy output`

53. Формирование `FilterChain` для фильтрации трафика:
   - Для каждой записи листа доступа формируется отдельный фильтр.
   - Запись `deny` транслируется в `DROP`.
   - Запись `permit` транслируется:
     - в ссылку на `inspection chain label`, если присутствует `inspection chain`, и в записи листа доступа указан протокол TCP или IP
     - в `PASS`, если `inspection chain` отсутствует.
   - Формирование параметров LogEventID ()

- Последней записью перед inspect chain label или последней записью в FilterChain при отсутствии inspect chain пишется фильтр drop all:

```
Filter (
     Action = DROP
)
```

54. Формирование FilterChain для инспектирования трафика:

- Если присутствует фильтрация трафика, то первая запись `inspect chain` помечается `label` со значением имени `inspect chain` (из команды `ip inspect name` …)
- Записи формируются в порядке следования команд `ip inspect name` в Cisco-like конфигурации. Исключение: если в команде `ip inspect name` указан протокол tcp, то это всегда порождает предпоследний фильтр перед “pass all” (независимо от расположения самой команды в конфигурации):

```
Filter (
     ProtocolID *= 6
     Action = PASS
     ExtendedAction = inspect_tcp< … >
)
```

- Ссылка на `port map` порождает последовательность фильтров, основанную на командах `ip port map`:
  - Параметр `ProtocolID` всегда равен 6.
  - Параметр `Action` – всегда `PASS`.
  - Порт или диапазон портов (из команды `ip port map`) транслируется в параметр `DestinationPort`.
  - Если в `port map` присутствует ссылка на список доступа, то каждая запись из этого списка порождает отдельную запись Filter:
    - `DestinationIP` (единичный адрес или подсеть) берется из записи списка доступа
    - В случае `permit` – пишется запись с параметром `ExtendedAction` (формирование смотрите ниже)
    - В случае `deny` – параметр `ExtendedAction` отсутствует
    - При наличии спецификатора `log` пишется параметр (): `Log = TRUE`
  - В параметр `ExtendedAction` прописывается значение `inspect_tcp<…>`. Исключение: если в `port map` указывается ftp, то прописывается `inspect_ftp <…>`.
  - В качестве параметров `ExtendedAction` может прописываться `timeout` и флаги `NOALERT` и `AUDIT`.
  - Последней записью пишется фильтр pass all:

```
Filter (
      Action = PASS:
)
```

  - Формирование параметров LogEventID и Log .

55. Пример формирования FilterChain для фильтрации и инспектирования трафика

- Фрагмент Cisco-like конфигурации:

```
ip inspect alert-off
...
access-list 1 deny 10.20.30.40
access-list 1 permit 10.20.30.0 0.0.0.255
!
```

```
ip port-map user-1 port tcp 2000
!
ip port-map user-2 port tcp 2001 list 1
!
ip inspect name inspect1 user-1 timeout 1008
ip inspect name inspect1 ftp audit-trail on
ip inspect name inspect1 tcp alert on audit-trail off timeout 1234
ip inspect name inspect1 user-2 alert off
...
ip access-list extended acl1
 deny udp 3.4.0.0 0.0.255.255 any
 deny icmp host 3.4.5.6 9.8.7.0 0.0.0.255
 permit tcp host 1.2.3.4 any established
 permit tcp any any
...
interface FastEthernet0/0
 ip access-group acl1 in
 ip inspect inspect1 in
```

- Фрагмент полученной LSP конфигурации:

```
FilterChain FilterChain:acl1 (
    Filters *= Filter (
        SourceIP *= 3.4.0.0/16
        ProtocolID *= 17
        Action = DROP
        LogEventID = "acl1"
    ),
    Filter (
        SourceIP *= 3.4.5.6
        DestinationIP *= 9.8.7.0/24
        ProtocolID *= 1
        Action = DROP
        LogEventID = "acl1"
    ),
    Filter (
        SourceIP *= 1.2.3.4
        ProtocolID *= 6
        Action = "inspect1"
        ExtendedAction = tcp_flags< any_set *= RST, ACK >
        LogEventID = "acl1"
    ),
    Filter (
        ProtocolID *= 6
        Action = "inspect1"
        LogEventID = "acl1"
```

```
    ),
    Filter (
        Action = DROP
    ),
    Filter (
        Label = "inspect1"
        ProtocolID *= 6
        DestinationPort *= 2000
        Action = PASS
        ExtendedAction = inspect_tcp< timeout = 1008  flags *= NOALERT >
        LogEventID = "Inspect:inspect1:user-1"
    ),
    Filter (
        ProtocolID *= 6
        DestinationPort *= 21
        Action = PASS
        ExtendedAction = inspect_ftp< flags *= NOALERT, AUDIT >
        LogEventID = "Inspect:inspect1:ftp"
    ),
    Filter (
        DestinationIP *= 10.20.30.40
        ProtocolID *= 6
        DestinationPort *= 2001
        Action = PASS
        LogEventID = "Inspect:inspect1:user-2:1"
    ),
    Filter (
        DestinationIP *= 10.20.30.0/24
        ProtocolID *= 6
        DestinationPort *= 2001
        Action = PASS
        ExtendedAction = inspect_tcp< flags *= NOALERT >
        LogEventID = "Inspect:inspect1:user-2:1"
    ),
    Filter (
        ProtocolID *= 6
        Action = PASS
        ExtendedAction = inspect_tcp< timeout = 1234 >
        LogEventID = "Inspect:inspect1:tcp"
    ),
    Filter (
        Action = PASS
    )
)
```

56. Формирование `FilterChain` для QoS классификации трафика.

- **Берется описание policy map** и последовательно раскрывается по порядку следования class map, за одним исключением: при наличии default class map, она порождает предпоследнюю по счету запись в FilterChain.
  - При переборе class maps делается просмотр на один шаг вперед: запоминается имя следующей class map для того, чтобы использовать его как ссылку на label.
- **Первые по счету фильтры всех class maps**, кроме первой, помечаются меткой (label) с именем class map в качестве значения.
  - Для default class map используется метка “class-default”.
- **Далее описывается конвертирование очередной class map**.
- **Если class map не содержит критериев классификации трафика**, для нее пишется вырожденный фильтр вида:

```
Filter (
   Label = "<class-map-i>"
   Action = "<class-map-i+1>"
)
```

  - Варианты таких `class maps`:
    - пустая `class map`
    - `class map` типа `match-all`, которая содержит хотя бы одну ссылку на несуществующий или пустой список доступа
    - `class map` типа `match-any`, которая содержит только ссылки на несуществующие или пустые листы доступа.
- **Если class map применима ко всему трафику**, пишется `Filter` следующего вида:

```
Filter (
   Label = "<class-map>"
   Action = PASS
   ExtendedAction = classify_mark< tos_set=  tos_set_mask=... >
   LogEventID = "Classification:<policy-map>:<class-map>"
)
```

Примечание: здесь и далее предполагается, что присутствуют команды set precedence и/или set dscp. Если такие команды отсутствуют, то в результирующих фильтрах отсутствуют параметры tos_set и tos_set_mask. Подробнее см. далее.

  - Варианты таких `class maps`:
    - `class map` по умолчанию (задается командой `class class-default`);
    - `class map` типа `match-all`, которая содержит единственный критерий классификации трафика – команду `match-any` ;
    - `class map` типа `match-any`, в которой один из критериев классификации трафика - команда `match-any`.

- **Правила конвертирования команд set (маркировка трафика):**
  - Если команды set precedence и set dscp отсутствуют, параметры tos_set и tos_set_mask в результирующих фильтрах не пишутся.
  - Команда `set precedence` транслируется в

```
ExtendedAction = classify_mark< tos_set=<num_precedence>*32 tos_set_mask=224 >
```

где

<num_precedence>*32 – числовое значение команды `set precedence`, умноженное на 32 (битовый сдвиг на 5 бит влево, т.е.

результирующее значение TOS-байта для указанного `precedence` без учета остальных битов)

маска 224 (11100000) указывает на `precedence`.

- Команда `set dscp` транслируется в

```
ExtendedAction = classify_mark< tos_set=<num_dscp>*4 tos_set_mask=252 >
```

где

<num_dscp>*4 – числовое значение команды `set dscp`, умноженное на 4 (битовый сдвиг на 2 бита влево, т.е. результирующее значение TOS-байта для указанного `DSCP` без учета остальных битов)

маска 252 (11111100) указывает на `DSCP`.

- В некоторых случаях в одном фильтре могут смешиваться указания как для классификации, так и для маркировки трафика. Например (в оригинале ExtendedAction пишется в одну строку):

```
Filter (
   Action = PASS
   ExtendedAction = classify_mark<  tos_match *= 160, 192
tos_match_mask=224 tos_set=96 tos_set_mask=224 >
)
```

Данный фильтр задает правило: трафик с `precedence=internet` или `precedence=critical` маркировать как `precedence=flash`.

- **Далее будут расматриваться невырожденные варианты построения class maps**. Они могут приводить к разветвленной структуре ссылок на различные labels. Следует отметить, что в общем случае для конкретного class map могут быть ссылки на две labels:
  - Фильтры, порожденные следующим критерием классификации трафика в рамках текущей class map. Далее такие ссылки будут указываться, как “<Curr_class>”.
  - Корневой фильтр, описывающий начало следующего class map в policy map. Далее будет указываться, как “<Next_class>”.
  - Пример записи FilterChain:

```
...
    Filters *= Filter (
        Action = "<Curr_class>"
        ExtendedAction = classify_mark<  tos_match *= 40,
192 tos_match_mask=252 >
        LogEventID = "Classification:..."
    ),
    Filter (
        Action = "<Next_class>"
    ),
...
```

Данная запись обозначает:

если TOS-байт трафика содержит значение `DSCP=af11` или `DSCP=cs6`, то рассмотреть следующий критерий классификации трафика из текущей `class map` (соответствует `class map` типа `match-all`),

в противном случае - перейти к рассмотрению следующей `class map`.

- **Конвертирование class map типа match-all:**

- Команда `match-any`, если она не является единственным критерием классификации, игнорируется.
- Команды `match dscp` и `match precedence` транслируются (каждая по отдельности) в следующие конструкции:
  - При наличии <Curr_class>:
    ```
        Filter (
            Action = "<Curr_class>"
            ExtendedAction = classify_mark<  tos_match *= ...
    tos_match_mask=... >
            LogEventID = "Classification:..."
        ),
        Filter (
            Action = "<Next_class>"
        )
    ```
  - При отсутствии <Curr_class> (последний критерий классификации для данного class map):
    ```
        Filter (
            Action = PASS
            ExtendedAction = classify_mark<  tos_match *= ...
    tos_match_mask=... tos_set=... tos_set_mask=... >
            LogEventID = "Classification:..."
        )
    ```
  - Пример:
    Фрагмент Cisco-like конфигурации:
    ```
    ...
    class-map match-all cl-map-1
     match dscp af11 cs6
     match precedence internet critical
    !
    policy-map pol-map-1
    class cl-map-1
      set precedence flash
    !
    ...
    ```

    Фрагмент LSP:
    ```
    ...
    FilterChain ClassificationChain:pol-map-1 (
        Filters *= Filter (
            Action = "cl-map-1:match_precedence"
            ExtendedAction = classify_mark<  tos_match *= 40,
    192 tos_match_mask=252 >
            LogEventID = "Classification:pol-map-1:cl-map-
    1:match_dscp"
        ),
        Filter (
            Action = PASS
        ),
        Filter (
            Label = "cl-map-1:match_precedence"
            Action = PASS
            ExtendedAction = classify_mark<  tos_match *= 160,
    192 tos_match_mask=224 tos_set=96 tos_set_mask=224 >
            LogEventID = "Classification:pol-map-1:cl-map-
    1:match_precedence"
        ),
        Filter (
            Action = PASS
        )
    )
    ...
    ```

- Для каждого критерия классификации, задаваемого командой `match access-group`, пишется отдельный `FilterChain`. Для каждой записи листа доступа формируется отдельный фильтр:
  - deny транслируется в ссылку на <Next_class>. При отсутствии последней – в PASS-правило.
  - permit транслируется в ссылку на <Curr_class>. При отсутствии последней – в PASS-правило, в котором в качестве параметра ExtendedAction задается маркировка трафика (tos_set/tos_set_mask).
  - При наличии спецификатора log или log-input, пишется параметр Log=TRUE (подробнее см.).
  - Следует отметить, что указания TCP-флагов в листах доступа (включая спецификатор established) игнорируются.
  - Если присутствует ссылка <Curr_class> (данный критерий классификации не последний для данного class map), то в конце пишется фильтр:

    При наличии <Next_class>:

    ```
    Filter (
        Action = "<Next_class>"
    )
    ```

    При отсутствии <Next_class>:

    ```
    Filter (
        Action = PASS
    )
    ```

  - Пример:

    Фрагмент Cisco-like конфигурации:

    ```
    ip access-list extended acl1
     deny tcp host 10.20.30.40 host 10.10.10.10
     permit ip host 10.20.30.40 10.10.10.0 0.0.0.255
    !
    ip access-list extended acl2
    ...
    class-map match-all cl-map-1
     ...
     match access-group name acl1
     match access-group name acl2
    !
    class-map cl-map-2
    ...
    policy-map pol-map-1
     class cl-map-1
      set precedence flash
     class cl-map-2
      set precedence critical
    ...
    ```

    Фрагмент LSP:

    ```
    ...
        Filter (
            SourceIP *= 10.20.30.40
            DestinationIP *= 10.10.10.10
            ProtocolID *= 6
            Action = "cl-map-2"
            LogEventID = "Classification:pol-map-1:cl-map-
    1:acl1"
        ),
        Filter (
            SourceIP *= 10.20.30.40
            DestinationIP *= 10.10.10.0/24
            Action = "cl-map-1:acl2"
    ```

```
        LogEventID = "Classification:pol-map-1:cl-map-
1:acl1"
    ),
    Filter (
        Action = "cl-map-2"
    ),
    Filter (
        Label = "cl-map-1:acl2"
...
    Filter (
        Label = "cl-map-2"
...
    Filter (
        Action = PASS
    )
...
```

- **Конвертирование class map типа match-any:**
  - Присутствие команды `match-any`, приводит к тому, что данная `class map` применима ко всему трафику. Этот случай рассмотрен выше.
  - Команды match dscp и match precedence транслируются (каждая по отдельности) в фильтр вида:

```
Filter (
   Action = PASS
   ExtendedAction = classify_mark<  tos_match *= ...
tos_match_mask=... tos_set=... tos_set_mask=... >
   LogEventID = "Classification:..."
)
```

  - Команды match access-group, ссылающиеся на несуществующие или пустые ACLs, игнорируются.
  - Для каждого критерия классификации, задаваемого командой match access-group, пишется отдельный набор фильтров. Для каждой записи листа доступа формируется отдельный фильтр:
    - deny транслируется в ссылку на <Curr_class>. При ее отсутствии – в ссылку на <Next_class>. При отсутствии последней – в PASS-правило.
    - permit транслируется в PASS-правило, в котором в качестве параметра ExtendedAction задается маркировка трафика (tos_set/tos_set_mask).
    - Следует отметить, что указания TCP-флагов в записи листа доступа (включая спецификатор established) игнорируются.
  - Пример:

    Фрагмент Cisco-like конфигурации:

```
ip access-list extended acl1
 deny tcp host 10.20.30.40 host 10.10.10.10
 permit ip host 10.20.30.40 10.10.10.0 0.0.0.255
!
ip access-list extended acl2
 permit udp any any
!
class-map match-any cl-map-1
 match precedence priority
match access-group name acl1
 match access-group name acl2
!
class-map cl-map-2
 match precedence internet critical
 match dscp af11 cs6
!
policy-map pol-map-1
 class class-default
```

```
 class cl-map-1
  set precedence flash
class cl-map-2
  set precedence network
...
```

Фрагмент LSP:

```
...
FilterChain ClassificationChain:pol-map-1 (
    Filters *= Filter (
        Action = PASS
        ExtendedAction = classify_mark<  tos_match *= 32
tos_match_mask=224 tos_set=96 tos_set_mask=224 >
        LogEventID = "Classification:pol-map-1:cl-map-
1:match_precedence"
    ),
    Filter (
        SourceIP *= 10.20.30.40
        DestinationIP *= 10.10.10.10
        ProtocolID *= 6
        Action = "cl-map-1:acl2"
        LogEventID = "Classification:pol-map-1:cl-map-
1:acl1"
    ),
    Filter (
        SourceIP *= 10.20.30.40
        DestinationIP *= 10.10.10.0/24
        Action = PASS
        ExtendedAction = classify_mark< tos_set=96
tos_set_mask=224 >
        LogEventID = "Classification:pol-map-1:cl-map-
1:acl1"
    ),
    Filter (
        Label = "cl-map-1:acl2"
        ProtocolID *= 17
        Action = PASS
        ExtendedAction = classify_mark< tos_set=96
tos_set_mask=224 >
        LogEventID = "Classification:pol-map-1:cl-map-
1:acl2"
    ),
    Filter (
        Label = "cl-map-2"
        Action = "cl-map-2:match_precedence"
        ExtendedAction = classify_mark<  tos_match *= 40,
192 tos_match_mask=252 >
        LogEventID = "Classification:pol-map-1:cl-map-
2:match_dscp"
    ),
    Filter (
        Action = "class-default"
    ),
    Filter (
        Label = "cl-map-2:match_precedence"
        Action = PASS
        ExtendedAction = classify_mark<  tos_match *= 160,
192 tos_match_mask=224 tos_set=224 tos_set_mask=224 >
        LogEventID = "Classification:pol-map-1:cl-map-
2:match_precedence"
    ),
    Filter (
        Label = "class-default"
```

```
            Action = PASS
            LogEventID = "Classification:pol-map-1:class-
default"
        )
)
...
```

57. Из интерфейса последовательно читаются crypto maps из crypto map set, прописанного в команде crypto map <crypto_map> (режим конфигурирования интерфейса).
    - Из описания crypto map читается access list, прописанный в команде match address <access_list> (режим конфигурирования crypto map).
    - Далее для простоты такой access list будет указываться как crypto-map-acl.
    - Если в файле `cs_conv.ini` параметр `ike_autopass` выставлен в значение `"on"` (значение по умолчанию, см.), то в начале FilterChain пишется следующий фильтр:

```
Filter (
    ProtocolID = 17
    SourcePort = 500, 4500
    Action = PASS
    PacketType = LOCAL_UNICAST, LOCAL_MISDIRECTED
)
```

        - Следует учесть, что данная настройка помешает созданию конфигурации с вложенным IPsec.
            - Если необходимо составить конфигурацию с вложенным IPsec, то следует параметр `ike_autopass` выставить в значение `"off"` и самостоятельно прописать соответствующий PASS-фильтр для IKE пакетов, в котором указан адрес партнера по защищенному соединению.
    - Происходит трансляция из crypto-map-acl в структуры Filter следующим образом:
        - deny транслируется в:
            - если присутствует следующая crypto map, то в ссылку (по label) на первый фильтр, сгенерированный из crypto-map-acl следующей crypto map.
            - правило PASS для последней crypto map в списке.
        - permit транслируется в правило APPLY (IPSec). При этом пишутся параметры из данного crypto map.
        - Указания TCP-флагов (включая спецификатор established), а также спецификаторы log и log-input игнорируются.
58. В случае, если в crypto map set присутствует ссылка на dynamic template set (задается командами crypto dynamic map), в котором есть несколько dynamic crypto maps, в crypto-map-acls которых существуют пересечения по адресам, в фильтре происходит объединение правил.
    - **Объединение правил для статических crypto maps не производится** (ни между разными статическими crypto maps, ни между статическими и динамическими crypto maps).
    - **В случае если статическая crypto map имеет приоритет ниже, чем динамическая**, могут возникать логические неувязки. Настоятельно рекомендуется давать статическим crypto maps приоритет выше, чем динамическим. Следует отметить, что в документации Cisco также присутствует эта рекомендация.
        - Если данная рекомендация не выполнена – выдается предупреждение [].
    - **Не производится объединение правил для динамических crypto maps**, которые входят в разные dynamic template sets, которые в свою очередь входят в один crypto map set.

- **В случае если в dynamic template set существует пересечение по адресам** правил, в которых для одних dynamic templates прописаны правила permit, а для других – deny; в фильтре прописывается правило вида (PASS), (Action1), …, (ActionN).
    - Логика формирования данных фильтров может существенно отличаться от логики Cisco.
    - В данном примере продемонстрирован особый прием: специально для прописывания PASS-правила сделан crypto dynamic-map dmap 2 (на самом деле приоритет этого dynamic map в данном конкретном случае не важен), в котором нет ничего, кроме связи с ACL, состоящим из deny-правила (правил): отсутствуют transform sets и т.п. Следует отметить, что данный способ может использоваться только с агентом, и неприменим на реальных устройствах Cisco.
    - Данная логика действует только на явно прописанные deny-правила. Для неявных правил deny ip any any, которые предполагаются в конце каждого access list, никаких объединений правил не делается.
- **В случае, если для данной crypto map задан IKECFG пул** (любым способом: или с помощью команд crypto isakmp client configuration address-pool local / crypto map … client configuration address initiate/respond; или с помощью команды set pool), то в структуре IKERule прописывается соответствующий AddressPool.
    - Следует учитывать, что если используется crypto map с crypto-map-acl, то в данном crypto-map-acl надо вписывать правила, в которые попадают адреса из пула. Например:

```
! ...
ip local pool pool1 192.168.211.10 192.168.211.30
! ...
crypto map cmap 10 ipsec-isakmp
 set peer 10.1.1.10
 ! ...
 match address cr-acl
 set pool pool1
!
ip access-list extended cr-acl
 permit ip <...> host 10.1.1.10
 permit ip <...> 192.168.211.0 0.0.0.255
! ...
```

    - Примечание: приведенная конфигурация корректна, но используется редко. На практике IKECFG пул чаще всего задается для динамических crypto maps, к которым не привязан ни один crypto-map-acl.
    - Требование к прописыванию фильтра с адресами из пула примерно соответствует поведению Cisco IOS.
    - Однако данное поведение отличается от версии 3.1: там можно было написать укороченный crypto-map-acl (фильтр с адресами из пула создавался неявно):

```
ip access-list extended cr-acl
 permit ip <...> host 10.1.1.10
```

59. Происходит проверка нужно ли прописывать данный фильтр. Если этот фильтр совпадает или полностью включается в один из предыдущих фильтров, прописанных для данного интерфейса, тогда этот фильтр не прописывается в LSP.

60. Формирование параметров LogEventID и Log структуры FilterChain:

- Примечание 1: LogEventID формируется всегда.
- Примечание 2: если структура Filter формируется не из записи ACL, параметр Log выставляется в FALSE.
- Примечание 3: параметр Log выставляется в TRUE в случае, если структура Filter формируется из записи ACL и в этой записи присутствует спецификатор log или log-input.
- Объекты, кроме IPsec policy (см. ниже):

Тип исходного объекта	Ситуация	Параметр LogEventID	Допустимость параметра Log=TRUE
Фильтрующий ACL		“<ACL-name>”	+
Inspection	Запись в port map, не имеющая ссылок на ACL	“Inspect:<inspect-name>:<port-map-name>”	
	Запись в port map, ссылающаяся на ACL	“Inspect:<inspect-name>:<port-map-name>”	+
	Ссылка на протокол TCP	“Inspect:<inspect-name>:tcp”	
Classification	match acl…	“Classification:<policy-map-name>:<class-map-name>:<acl-name>”	+
	match dscp…	“Classification:<policy-map-name>:<class-map-name>:match_dscp”	
	match precedence…	“Classification:<policy-map-name>:<class-map-name>:match_precedence	
	match any	“Classification:<policy-map-name>:<class-map-name>”	
Clear-Text фильтрация внутри защищенного соединения		“ClearText:<acl-name>”	+

- IPsec policy:
  - Вначале идет префикс “IPsec”.
  - Далее, если допускается открытый трафик, пишется “:Bypass”.
  - Далее, если присутствует защищенный трафик, пишется “:Protect“.
    - Для динамических crypto maps возможны сочетания “IPsec:Bypass:Protect”.
  - Далее, после двоеточия, пишется набор из следующих параметров:
    - Имя и порядковый номер crypto map, разделенные двоеточием.
    - Если используется dynamic map: имя и порядковый номер dynamic map, разделенные двоеточием.
    - Если присутствует ACL (может отсутствовать для dynamic map): после двоеточия пишется имя этого ACL.
  - Если источником являются несколько crypto maps, то блоки описания для каждой crypto map разделяются символом “$” (доллар).
  - Параметр Log=TRUE допускается: пишется в том случае, если в текущей записи ACL присутствует модификатор log или log-input.

- Если фильтр формируется на основе нескольких записей ACL, достаточно присутствие одного из этих модификаторов хотя бы в одной из задействованных записей.

- Пример исходной конфигурации:

```
...
ip access-list extended crypto-acl-1
 permit ip any any
!
ip access-list extended crypto-acl-2
 deny ip host 192.168.101.102 any log
 permit tcp 192.168.3.0 0.0.0.255 any eq 80 log
!
crypto dynamic-map dmap 10
 match address crypto-acl-1
 set transform-set tr1
!
crypto dynamic-map dmap 20
 match address crypto-acl-2
 set transform-set tr2
!
crypto map cmap 10 ipsec-isakmp dynamic dmap
...
```

Native LSP конфигурация

```
...
FilterChain IPsecPolicy:cmap (
    Filters *= Filter (
        SourceIP *= 192.168.101.102
        Action = PASS
        ExtendedAction = ipsec< sa *= IPsecAction:dmap:10
fallback_action = PASS >
        LogEventID =
"IPsec:Bypass:Protect:cmap:10:dmap:10:crypto-acl-
1$cmap:10:dmap:20:crypto-acl-2"
        Log = TRUE
    ),
    Filter (
        SourceIP *= 192.168.3.0/24
        ProtocolID *= 6
        DestinationPort *= 80
        Action = PASS
        ExtendedAction = ipsec< sa *= IPsecAction:dmap:10,
IPsecAction:dmap:20 >
        LogEventID = "IPsec:Protect:cmap:10:dmap:10:crypto-
acl-1$cmap:10:dmap:20:crypto-acl-2"
        Log = TRUE
    ),
    Filter (
        Action = PASS
        ExtendedAction = ipsec< sa *= IPsecAction:dmap:10 >
        LogEventID = "IPsec:Protect:cmap:10:dmap:10:crypto-
acl-1"
    )
)
...
```

Формирование имен структур LSP при конвертировании

При конвертировании Cisco-like конфигурации в LSP конфигурацию имена структур LSP формируются из имен и индексов объектов Cisco-like конфигурации. При этом следует учитывать ряд ограничений:

- В объектах Cisco-like конфигурации разных типов могут использоваться одинаковые имена. В LSP имя объекта должно быть уникальным.
- Могут использоваться цифровые индексы. В LSP требуется задавать идентификаторы, начинающиеся с буквы.
- Как правило, синтаксис Cisco-like имен более свободный (например, допускаются символы, которые нельзя использовать в идентификаторах LSP).
- В некоторых случаях требуется формировать имя структуры LSP из группы объектов Cisco-like конфигурации.
- Один объект Cisco-like конфигурации (или группа объектов) может порождать несколько LSP объектов (каждый из которых должен обладать уникальным именем).

Общие сведения по формированию имен:

- **Сначала готовится прототип имени объекта**. Для этого прототипа нет каких-то специальных требований: например это может быть имя объекта Cisco-like конфигурации, константная строка, сочетание префикса и имен нескольких объектов и т.п.
- **Основные административные принципы формирования имени**:
  - В имени структуры LSP используются имена и/или числовые идентификаторы объектов Cisco-like конфигурации (далее для простоты “Cisco-объект”), порождающих данную структуру.
  - Если существует однозначная связь между Cisco-объектом и структурой LSP:
    - Если Cisco-объект идентифицируется по имени, то напрямую используется данное имя.
    - Если Cisco-объект идентифицируется числом или набором разрозненных параметров, то в качестве имени структуры LSP используется команда IOS, создающая данный Cisco-объект, в которой пробелы заменены на символ "**:**" (двоеточие).
  - Если Cisco-объект может породить несколько структур LSP строго одного типа в зависимости от дополнительных объектов, начало имени порождается аналогично предыдущему пункту, а затем через двоеточие перечисляются идентификаторых дополнительных объектов.
  - Если структура LSP порождается из нескольких Cisco-объектов: Cisco-объект порождает несколько структур LSP или если существуют какие-то иные неоднозначности (например, один и тот же ACL может порождать разные структуры LSP одного и того же типа FilterChain), то в начало имени структуры добавляется префикс, описывающий ролевую принадлежность данной структуры и заканчивающийся на двоеточие.
    - В некоторых случаях в качестве префикса может использоваться тип структуры LSP.
  - Если Cisco-объект идентифицируется несколькими параметрами (например, имя и индекс), то они в имени LSP-структуры разделяются символом "**:**" (двоеточие).
  - Если LSP-структура порождается несколькими разнотипными объектами, образующими вместе некоторую законченную группу, то их идентификаторы также разделяются двоеточием.

- Если LSP-структура порождается несколькими однотипными объектами или несколькими замкнутыми группами разнотипных объектов, то группы идентификаторов разделяются символом “$” (доллар).
  - Например, пусть в Cisco-like конфигурации присутствуют некоторые объекты:
    ```
    obj_type_1 abc 15
    obj_type_2 1000
    obj_type_1 def 20
    obj_type_2 2000
    ```
  - Возможный вариант порождаемого имени:
    ```
    SomePrefix:abc:15:1000$def:20:2000
    ```
- Важный частный случай: когда объект порождается из конкретной crypto map, действуют следующие правила для идентификации crypto map:
  - Если объект порождается из статической crypto map, то в имени объекта присутствуют имя и индекс данной crypto map, разделенные двоеточием. Например:
    ```
    cmap:10
    ```
  - Если объект порождается из конкретной записи динамического crypto map template, то в имени объекта присутствуют следующие части: имя и индекс динамической crypto map, а также имя динамического crypto map template и индекс конкретной записи в нем. Например, фрагмент конфигурации:
    ```
    crypto dynamic-map dmap 20
    ...
    crypto map cmap 5 ipsec-isakmp dynamic dmap
    ```

    порождает объект, в имени которого присутствует:

    ```
    cmap:5:dmap:20
    ```
  - Если имя получается слишком длинным, часть объектов, порождающих данную структуру, может отсутствовать в сформированном имени. В этом случае к имени добавляется суффикс $$etc. Например:
    ```
    IKERule:cmap-1:10:dmap-1:10$cmap-1:20:dmap-2:10$$etc
    ```

- **Далее производится нормализация имени:**
  - Все символы, кроме букв латинского алфавита, цифр и символов “`-`” (минус), “`$`” (доллар) и “`:`” (двоеточие), преобразуются к символу подчеркивания.
  - Если имя начинается с цифры или иного символа, отличного от букв латинского алфавита и подчеркивания, перед ним ставится буква `n`.
- **Далее производится поиск полученного имени** среди уже сформированных (для обеспечения уникальности):
  - Если имя не найдено, считаем его окончательно сформированным.
  - Если имя найдено, добавляем к нему последовательно суффиксы $$1, $$2 и т.д. до тех пор, пока не будет найдено имя, которое еще не использовалось.
- **Полученное имя записывается в конфигурацию** и запоминается для того, чтобы оно не было использовано для другого объекта.

Далее описываются конкретные правила формирования прототипов имен объектов:

Имя структуры Native LSP	Вариант использования	Правило формирования	Примеры
`IKETransform`		Конкатенация префикса “crypto:isakmp:policy:” и индекса ISAKMP policy (команда crypto isakmp policy)	`crypto:isakmp:policy:10`

Имя структуры Native LSP	Вариант использования	Правило формирования	Примеры
AHProposal		Конкатенация имени transform-set и суффикса ":AH" (команда crypto ipsec transform-set)	trset1:AH
ESPProposal		Конкатенация имени transform-set и суффикса ":ESP" (команда crypto ipsec transform-set)	trset1:ESP
IKERule	Статическая crypto map	Конкатенация "IKERule:" и обозначения crypto map (см.). Если объект порожден из нескольких crypto maps, то их обозначения перечисляются через знак доллара $	IKERule:cmap:1 IKERule:cmap:2:dmap:10 IKERule:cmap-1:1:dmap:1$cmap-2:1:dmap:10
AuthMethodRSASign		RSA:Sign	RSA:Sign
AuthMethodDSSSign		DSS:Sign	DSS:Sign
AuthMethodGOSTSign		GOST:Sign	GOST:Sign
AuthMethodPreshared		Конкатенация "Preshared:" и идентификатора ключа – в зависимости от типа: IP-адрес или hostname (точки заменяются на знак подчеркивания). Если значение данного ключа подходит для нескольких идентификаторов, то эти идентификаторы перечисляются через двоеточие (действует правило по отсечению слишком длинных имен (см.)	Preshared:192_168_1_2 Preshared:host1_example_com Preshared:host1_example_com:192_168_1_2:192_168_1_3
CertDescription		CA:RSA CA:DSS CA:GOST	CA:GOST
IPsecAction		Конкатенация префикса "IPsecAction:" и обозначения crypto map (см.). Если объект порожден из нескольких crypto maps, то их обозначения перечисляются через знак	IPsecAction:cmap:1 IPsecAction:cmap:5:dmap:20 IPsecAction:cmap-1:1:dmap-1:1$cmap-2:1:dmap-1:1

Имя структуры Native LSP	Вариант использования	Правило формирования	Примеры
		доллара $.	
AddressPool	Используется команда “crypto isakmp client configuration address-pool local…” или команда “set pool…”, в которой не задана маска подсети.	Имя pool (команда ip local pool)	pool1
FilterChain	Фильтр	Конкатенация “FilterChain:” и имени ACL	FilterChain:101 FilterChain:acl1
	Inspection (только при отсутствии фильтрующего ACL; в противном случае – часть фильтрующего ACL)	Конкатенация “InspectChain:” и имени inspection, заданного командой ip inspect name	InspectChain:inspect1
	Классификация трафика (QoS):	Конкатенация “ClassificationChain:” и имени policy map (команда policy-map).	ClassificationChain:pmap1
	Clear-Text фильтрация внутри защищенного соединения.	Конкатенация “ClearText:” и имени ACL, заданного командой set ip access-group (режим конфигурирования crypto map)	ClearText:clear-text-acl ClearText:15
	Правила защиты (IPsec policy)	Конкатенация “IPsecPolicy:” и имени crypto map set	IPsecPolicy:cmap
IdentityEntry		Имя identity (команда crypto identity)	id1

Расширения сертификата (Certificate Extensions)

Имеются некоторые ограничения при работе с расширениями сертификата (Extensions), которые помечены как критичные. В таблице приведен список расширений сертификата, которые будут распознаваться и обрабатываться Продуктом, если у них установлен признак критичности TRUE. Если в сертификате будут присутствовать другие расширения, не указанные в таблице и заданные как критичные, то такой сертификат не может быть использован. Если же расширение отсутствует в таблице, но является некритичным, то оно игнорируется, и сертификат используется.

Name	OID value
Subject Key Identifier	2.5.29.14
Key Usage	2.5.29.15
Subject Alternative Name	2.5.29.17
Issuer Alternative Name	2.5.29.18
Basic Constraints	2.5.29.19
Name Constraints	2.5.29.30
CRL Distribution Points	2.5.29.31
Authority Key Identifier	2.5.29.35

Описания значений и полный список Certificate Extensions можно посмотреть в документе RFC 5280 (http://tools.ietf.org/html/rfc5280#section-4.2).

Можно изменить реакцию Продукта на отдельные расширения сертификата, помеченные как критичные и отсутствующие в вышеприведенной таблице. Администратор может настроить список расширений сертификата, который будут игнорироваться Продуктом, как если бы эти расширения являлись некритичными. Эти расширения надо описать в файле `x509opts.ini`, который расположен в каталоге `/opt/VPNagent/etc`. Расширения описываются в секции `IgnoringUnsupportedCriticalExtentions`.

Игнорируемое `Critical Extention` задается в формате `<KEY>=<OID>`, где:

`<KEY>` – имя расширения, состоящее из букв и цифр и не содержащее разделителей, должно быть уникальным в передах секции;

`<OID>` – OID игнорируемого расширения, состоящий из десятичных чисел, разделенных точками. Распознавание расширения происходит по OID.

Пример файла x509opts.ini:

```
[IgnoringUnsupportedCriticalExtentions]
!!
! Key name is any Alpha-Numerical well-known name of OID
! Key names of different OIDs cannot match
!!
subjectDirectoryAttributes=2.5.29.9
```

```
CertificatePolicies=2.5.29.32
QcStatements=1.3.6.1.5.5.7.1.3
HcRole=1.0.21091.2.0.5
```

Примечание 1: следует подчеркнуть, что таким образом нельзя проигнорировать распознаваемые Продуктом `Critical Extention`s, например `BasicConstraints`.

Примечание 2: секция `IgnoringUnsupportedCriticalExtentions`, даже пустая, обязательно должна прсутствовать в файле `x509opts.ini`.