Маршрутизаторы NSG

Программное обеспечение NSG Linux 2.0

Руководство пользователя

Часть 5

Туннелирование и виртуальные частные сети (VPN)

Версия программного обеспечения 2.0 build 7

Обновлено 10.11.2016

Москва 2016

АННОТАЦИЯ

Данный документ содержит руководство по настройке и применению маршрутизаторов NSG, оснащенных программным обеспечением NSG Linux 2.0. Документ имеет следующую структуру:

Руководства по применению маршрутизаторов под управлением NSG Linux 1.0 и базового программного обеспечения NSG, а также других продуктов NSG (модемов, мостов и т.п.), содержатся в отдельных документах.

ВНИМАНИЕ

Данное Руководство пользователя предназначено для лучшего понимания процедуры настройки в целом и описывает сущность выполняемых действий — а именно, *что* необходимо настраивать. Рассматриваемые вопросы относятся, как правило, к сущности используемой технологии и являются общими для любых её реализаций, независимо от конкретного производителя и устройства. (Исключением являются вопросы, специфичные для оборудования NSG — таких, как организация пользовательского интерфейса или настройка системы бесперебойных соединений ***ui*TCP**.)

Основной документацией по NSG Linux 2.0 является встроенная справка на борту устройства. Она описывает конкретные команды и параметры настройки — т.е. *как* настраивать функции и возможности, описанные в данном Руководстве. Для просмотра справки по каждому из параметров следует использовать кнопку в Web-интерфейсе или команду _manual (_m) в консольном интерфейсе. Если справка на вашем языке отсутствует, следует установить на устройстве русскую локаль.

ВНИМАНИЕ Продукция компании непрерывно совершенствуется, в связи с чем возможны изменения отдельных аппаратных и программных характеристик по сравнению с настоящим описанием. Сведения о последних изменениях приведены в файлах README.TXT, CHANGES, а также в документации на отдельные устройства.

Замечания и комментарии по документации NSG принимаются по адресу: doc@nsg.net.ru.

ООО «Эн–Эс–Джи»
Россия 105187 Москва
ул. Вольная, д.35
Тел./факс: (+7–495) 727–19–59 (многоканальный)

http://www.nsg.ru/
mailto:info@nsg.net.ru
mailto:sales@nsg.net.ru
mailto:support@nsg.net.ru

§ СОДЕРЖАНИЕ §

Часть 5. Туннелирование и виртуальные частные сети (VPN)

§5.1. Туннелирование протоколов через сети IP

Программное обеспечение NSG Linux 2.0 позволяет организовывать туннели (как простые, так и безопасные) для передачи пакетов различных протоколов через сеть IP. Как правило, в прикладных задачах трафик корпоративной сети туннелируется через сеть общего пользования. Данная версия NSG Linux 2.0 поддерживает следующие типы туннелей и виртуальных частных сетей (VPN):

— IPv4/v6-over-IPv4/v6 (GRE)
— Ethernet bridge-over-IP (GRE)
— PPTP (клиент и сервер, с поддержкой MPPE)
— PPP-over-Ethernet (клиент и сервер, с поддержкой MPPE)
— IPSec
— STunnel
— ***ui*TCP** — фирменную VPN NSG 4 уровня, ориентированную на гарантированную доставку данных и поддержание непрерывных сеансов работы прикладного ПО при многократных переключениях между основным и резервными каналами связи.

Настройка всех типов туннелей производится в узле меню .tunnel .

§5.2. Туннели GRE

§5.2.1. Общие параметры туннеля

Управление туннелями GRE производится в узле .tunnel.gre . Для создания нового туннеля следует нажать кнопку + или ввести команду _new. Туннели GRE в NSG Linux 2.0 обозначаются gre1, gre2 и т.д. Имя создаваемого туннеля можно ввести полностью в этом формате, либо только целое число (оно будет преобразовано к стандартному имени автоматически).

ПРИМЕЧАНИЕ Протокол GRE является самостоятельным протоколом транспортного уровня (в терминах модели OSI), работающим не поверх общеизвестных протоколов TCP или UDP, а параллельно с ними. Идентификатор протокола GRE, указываемый в заголовке IP-пакетов — 47. Для нормальной работы GRE-туннелей в системах с брандмауэрами и фильтрами необходимо разрешить прохождение пакетов IP с данным идентификатором.

Общими параметрами для туннелей GRE всех типов являются публичные адреса удалённого и локального шлюзов, ключ и выходной интерфейс. Не все из них являются обязательными, но в любом случае требуется указать хотя бы часть из них.

Адрес назначения (destination) — публичный адрес удалённого устройства, работающего в паре с данным. Как правило, в большинстве задач этот адрес указывается явно. Параметр обязательный по существу. Если адрес назначения не указан, то туннель может работать только на приём пакетов (при этом входящие пакеты фильтруются также по остальным параметрам — ключу и т.п.); передача пакетов невозможна, однако механизм *keepalive* при этом может отвечать на пакеты от удалённой стороны.

Адрес источника (source) — локальный адрес устройства NSG, который будет указываться в качестве источника в пакетах, отправляемых в сеть общего пользования. Этот же адрес должен быть указан в качестве назначения в пакетах, получаемых из сети общего пользования и относящихся к данному туннелю. Если адрес источника не указан (значение 0.0.0.0), то в исходящих пакетах в качестве источника указывается адрес того IP-интерфейса, с которого отправляются пакеты.

Указание выходного интерфейса (device) выполняет двоякую роль. Во-первых, оно жёстко привязывает туннель к IP-интерфейсу с указанным именем; работа туннеля через какие-либо другие интерфейсы (например, с использованием маршрута по умолчанию) запрещается. Во-вторых, этот параметр синхронизирует состояние туннеля с состоянием указанного интерфейса: устанавливает, разрывает и переустанавливает туннель в зависимости от наличия транспорта для него. В частности, явное назначение device необходимо при создании туннелей через динамические интерфейсы, такие как сотовые соединения.

Если выходной интерфейс не указан, то он выбирается согласно текущей таблице маршрутизации.

ПРИМЕЧАНИЕ В качестве транспорта для туннеля может использоваться любой IP-интерфейс, в т.ч. другой туннельный интерфейс.

Ключ туннеля (key)— число длиной 32 бита. Для удобства ввода ключ может быть задан как в виде обычного десятичного числа, так и в десятично-точечной нотации. При помощи ключа реализуется слабый метод защиты туннеля, а также выбор туннеля, если их в данном устройстве несколько с совпадающими (или не определенными) IP-адресами.

Как можно видеть, в общем случае для туннеля должно быть установлено хотя бы одно из двух значений destination или device. Если туннелей более одного, то должно быть также установлено хотя бы одно из двух значений source или key.

Для туннелей с инкапсуляцией eth (см. ниже), а также для работы механизма keepalive (см. ниже) обязательно указывать source или device.

ПРИМЕЧАНИЕ В ряде программных и аппаратных продуктов, в т.ч. в некоторых продуктах Cisco Systems, возможна работа только одного туннеля с одинаковыми IP-адресами сторон. В NSG Linux такая ситуация допускается, а туннели в этом случае различаются по ключу.

Опционально в туннеле могут быть включены или настроены следующие дополнительные параметры:

1. Проверка контрольной суммы во входящих пакетах (checksum). Если проверка включена, то все поврежденные пакеты уничтожаются.
2. Установка поля Type of Service для пакетов GRE-туннеля в сети общего пользования (tos). По умолчанию для инкапсуляции IP-over-GRE установлено специальное значение inherit — поле ToS копируется из туннелируемых пакетов корпоративной сети; для инкапсуляции Ethernet-over-GRE устанавливается 0.
3. Установка поля TTL для пакетов GRE-туннеля в сети общего пользования (ttl). Поскольку туннель сокращает количество шагов маршрутизации (*hops*) для этих пакетов, рекомендуется использовать небольшие значения. По умолчанию, устанавливается значение 64.

4. Контроль последовательности пакетов (sequence-datagrams). Данный механизм предназначен для уничтожения всех "запоздавших" пакетов. Если он включен, то в исходящие GRE пакеты добавляются порядковые номера, а во входящих — проверяются их номер. Контроль должен быть включен либо выключен одновременно на обеих сторонах туннеля.

   Далее, если включён контроль последовательности пакетов, то для её восстановления можно задействовать отдельный буфер. Глубина буфера устанавливается параметром reoder-buffer . Если значение данного параметра равно 0, то буфер не работает, т.е. пакеты, выпадающие из последовательности, уничтожаются немедленно. Например, если пакеты получены в такой последовательности:

   1 2 3 5 6 4 7 8

   то пакет 4 будет удален.

   Максимальное время ожидания пакетов в буфере устанавливается параметром reoder-timeout (в миллисекундах). По истечении этого времени, или по заполнению буфера, полученный пакет передаётся на дальнейшую обработку в системе, даже если перед ним имеются пропущенные.

Помимо этого механизма, в NSG Linux 2.0 реализованная дополнительная функция, не предусмотренная стандартом GRE: буфер для восстановления синхронизации туннеля при перегрузке или рестарте удалённой стороны. Максимальное значение счётчика пакетов равно 2^{32}–1, т.е. более 4 млрд. В стандартной реализации GRE, если туннель существует длительное время и прошёл более половины счётчика, а затем удалённый конец туннеля рестартовал и начал нумерацию пакетов заново, то для восстановления синхронизации могут использоваться, в лучшем случае, пакеты из последних 2 млрд. Пакеты с ещё меньшими номерами молча уничтожаются. Таким образом, если туннель перед рестартом почти приблизился к пределу в 4 млрд., то число уничтоженных пакетов до восстановления синхронизации близко к 2 млрд., и может пройти немалое время, прежде чем туннель сможет хотя бы попытаться восстановить синхронизацию.

Чтобы сократить это время, такие пакеты не уничтожаются, а накапливаются в специальном буфере размером sync-buffer. Степень упорядоченности этих пакетов оценивается параметром *качества*: это процент пакетов, имеющих номера в диапазоне от номера первого пакета до номера первого + размер буфера. Если произошло полное заполнение буфера в пределах интервала sync-timeout с качеством не ниже sync-integrity, то считается, что счетчик противоположного конца туннеля сбросился и необходимо восстановить синхронизацию. Тогда нумерация пакетов устанавливается на первый пакет в этом буфере, и вся накопленная очередь передаётся на дальнейшую обработку. По умолчанию, система ожидает 20 пакетов в пределах временного интервала 10 секунд с качеством 95% (т.е. не более 1 случайного пакета с номером вне интервала N_1 … N_1+20).

Данный механизм является фирменной разработкой NSG. Однако он функционирует только в одном направлении (пассивно синхронизирует устройство NSG с удалённой стороной), поэтому не препятствует работе с любыми другими реализациями GRE и продуктами сторонних производителей.

§5.2.2. Параметры полезной нагрузки (ip, ethernet)

Вторая часть настройки туннеля GRE — это настройка полезной нагрузки, т.е. пакетов, передаваемых внутри туннеля. В данной версии NSG Linux 2.0 поддерживаются два типа полезной нагрузки: пакеты IPv4 и пакеты Ethernet. Выбор между ними производится параметром encapsulation . В зависимости от выбранной инкапсуляции, в данном узле появляются дополнительные параметры.

Туннель с инкапсуляцией ip функционирует аналогично физическому интерфейсу "точка-точка". Конфигурация такого туннеля содержит узлы ifAddress и link, назначение которых одинаково для всех IP-интерфейсов. Важнейшим из параметров является ifAddress.prefix (IP-адрес и длина маски интерфейса); как правило, адрес интерфейсу GRE назначается статически. Туннель участвует в IP-маршрутизации наравне с остальными IP-интерфейсами.

Помимо этого, в туннеле с инкапсуляцией ip может использоваться специфический механизм *keepalive* для контроля его работоспособности (см. следующий параграф).

Туннель с инкапсуляцией eth аналогичен физическому порту Ethernet, в частности, на нём могут создаваться VLAN. Однако над ним не находится никакой IP-интерфейс и, следовательно, он не может использоваться напрямую для передачи IP-пакетов. Единственный способ использовать этот туннель — это включить его целиком, либо какие-то из созданных на нём VLAN, в состав программного моста 2 уровня (*bridge group*) или агрегированного канала (*bond group*).

Подробно об IP-адресах см. Часть 3, о *bridge groups* и *bond groups* — Часть 2.

§5.2.3. Механизм *keepalive* для туннелей GRE

Протокол GRE не предусматривает встроенного механизма *keepalive*, однако у различных производителей имеются собственные реализации этого механизма. В программном обеспечении NSG Linux используется механизм, предложенный компанией Cisco Systems; подробное описание этого алгоритма приведено в документе Cisco Systems: *GRE Tunnel Keepalives* (Document ID: 64565) и доступно по адресу: http://www.cisco.com/en/US/tech/tk827/tk369/technologies_tech_note09186a008048cffc.shtml.

Суть данного механизма состоит в том, что на удаленную сторону посылается специально сформированный пакет GRE с инкапсулируемым протоколом IP. Внутри него, однако, находится не просто IP-пакет, а еще один GRE-пакет, имеющий адресом назначения IP-адрес системы-инициатора запроса. Такая конструкция не противоречит спецификации GRE, поскольку пакет GRE является частным случаем IP-пакета. В качестве идентификатора протокола в этом пакете указан 0, что позволяет отличить его от остальных пакетов GRE-туннеля.

Инициатор посылает запросы GRE *keepalive* через установленные промежутки времени. Удаленная сторона туннеля разбирает внешний пакет GRE, извлекает из него вложенный пакет и обрабатывает его в соответствии со своей таблицей маршрутизации. Поскольку этот пакет представляет собой готовый пакет GRE-туннеля, он маршрутизируется обратно инициатору. Тот, получив пакет, разбирает его заголовок, по идентификатору протокола определяет, что это не пакет с полезными данными, а ответ на *keepalive*, и считает запрос отвеченным.

При отключенном механизме *keepalive* интерфейс посылает данные в туннель "наугад", не имея никакой информации о доступности и работоспособности удаленной стороны. В этом случае реализация GRE совместима с продуктами любых других сторонних производителей. Однако, чтобы туннель не превратился в "черную дыру", для контроля целостности данных следует использовать механизмы вложенных протоколов.

ВНИМАНИЕ Для работы механизма *keepalive* обязательно должен быть указан source или device.

ВНИМАНИЕ Для работы механизма *keepalive* создаются служебные правила в цепочках .ip.mangle.GRE_KA и ip.mangle.INPUT . После включения/выключения *keepalive* необходимо отдельно применить изменения в этом узле. Настройка данных правил пользователем не предусмотрена.

ПРИМЕЧАНИЕ Механизм *keepalive* реализован только для туннелей типа IP-over-IP (GRE).

Пример конфигурации.

```
tunnel
: gre
: : gre1
: : : destination          = "10.0.52.34"
: : : encapsulation        = "ip"
: : : source               = "10.0.52.33"
: : : keepalive            = "yes"
: : : keepalive-interval   = 5
: : : keepalive-retry      = 3
```

В данном случае запрос посылается каждые 5 секунд. В случае 3 неудачных запросов интерфейс gre1 переходит в состояние DOWN. При этом удаляются маршруты через этот интерфейс и др. Пакеты с данными, поступающие от удаленной стороны туннеля, сбрасываются с сообщением "proto unreachable" (как если бы туннеля не было вовсе).

Независимо от состояния интерфейса запросы продолжают посылаться; при получении первого ответа, т.е. при восстановлении работоспособности туннеля, интерфейс переходит в состояние UP, для него восстанавливаются все маршруты и дополнительные службы.

Из сути данного механизма следует, что запрос *keepalive* формируется в рамках туннеля и ответить на него может только вторая сторона туннеля, в рамках которого он создан. При этом наличие потока данных (в любую сторону) никак не влияет на алгоритм поднятия и опускания туннеля, т.е. если ответы на *keepalive* не приходят, то интерфейс перейдет в состояние DOWN независимо от того, что данные вроде как идут.

В частности, для того, чтобы система отвечала на запросы удаленной стороны, в ней должен быть создан туннельный интерфейс, для него указано административное состояние UP и указан адрес удаленной стороны. На входящие запросы туннель отвечает всегда, даже если сам он находится в состоянии DOWN по причине неполучения ответов *keepalive* от удаленной стороны, или переведён в состояние DOWN административно. Кроме того, прохождение пакетов GRE keepalive должно быть не запрещено фильтрами на обеих сторонах.

ВНИМАНИЕ Обычные Linux-системы не поддерживают *keepalive* и не отвечают на него, несмотря на то, что туннели поддерживаются. Это следует из общего правила, гласящего, что система не может принимать пакеты, которые якобы исходят от неё самой (а именно такой прием используется в данном случае).
Маршрутизаторы Cisco Systems поддерживают ответ на *keepalive* даже в том случае, если сами не умеют его посылать, но умеют создавать туннели.
Для продуктов других производителей возможна одна из двух вышеописанных ситуаций, в также иные фирменные реализации GRE *keepalive*.

Следующее фирменное расширение механизма GRE *keepalive* в продуктах NSG относится к случаю, когда туннель имеет ключ (key). Ключ выполняет двоякую функцию: во-первых, он служит для защиты туннеля (относительно слабой), а во-вторых — для идентификации пакетов, относящихся к тому или иному туннелю, в ситуации, когда между двумя устройствами установлено несколько туннелей с одинаковыми адресами источника и назначения. Реализация Cisco этого не предусматривает, а именно, не включает ключ в посылаемые пакеты *keepalive* и не отвечает на чужие пакеты, содержащие ключ. Таким образом, она не позволяет использовать *keepalive* при наличии нескольких параллельных туннелей.

Реализация NSG в этом отношении более гибкая и позволяет выбирать режим использования ключа в пакетах *keepalive* в зависимости от удалённой стороны. Переменная keepalive-without-key может принимать следующие значения:

yes-strict Не указывать ключ туннеля в исходящих пакетах *keepalive* и отвечать только на входящие пакеты, в которых также не указан ключ. Соответствует реализации *keepalive* в Cisco.

yes Не указывать ключ туннеля в исходящих пакетах и отвечать на любые входящие пакеты от удалённой стороны — как с ключом, так и без него. Совместимо с Cisco.

no-strict Указывать ключ туннеля в исходящих пакетах и отвечать только на пакеты, в которых также указан ключ. Абсолютно строгий режим, применим только при наличии устройств NSG на обеих сторонах туннеля.

no Указывать ключ туннеля в исходящих пакетах, но отвечать на любые входящие пакеты.

ВНИМАНИЕ Следует проявлять осторожность при использовании *keepalive*, если где-либо на маршруте через сеть общего пользования выполняется (или может выполняться) NAT. Значительная часть реализаций NAT не поддерживает GRE *keepalive*, даже если корректно обрабатывает пакеты GRE в общем случае. Для поддержки *keepalive* в NAT необходима особая дополнительная опция, поскольку замену адресов необходимо выполнять не только во внешнем заголовке запроса, но и в заголовке вложенного пакета-ответа.
При настройке GRE через сети, использующие NAT, следует сначала настроить его без *keepalive*, убедиться в работоспособности полученного туннеля (с помощью *ping* или любым другим способом), а затем включить *keepalive* и исследовать возможность работы с ним. Достаточно информативную картину работы данного механизма можно получить с помощью утилиты *tcpdump* на публичном интерфейсе.

§5.3. Сети VPDN второго уровня

§5.3.1. Общие замечания о VPDN

Виртуальные частные сети второго уровня в данной версии NSG Linux представлены соединениями PPP-over-Ethernet (PPPoE) и Point-to-Point Tunneling Protocol (PPTP). Оба эти протокола представляют собой расширения PPP, позволяющие устанавливать соединения "точка-точка", вместо физической среды, через сети Ethernet и IP, соответственно. Оба протокола имеют ряд свойств, унаследованных от PPP:

— Соединения имеют сеансовый, а не статический, характер.
— При установке PPP-соединения может быть выполнена аутентификация и авторизация (односторонняя или взаимная), а по мере работы — учёт работы пользователей. Для этих целей могут быть использованы локальная таблица пользователей или централизованные сервера TACACS+ и RADIUS.
— При установке IP-соединения могут быть согласованы параметры IP.
— При передаче трафика может использоваться сжатие с использованием различных методов, а также защита по протоколу MPPE (Microsoft Point-to-Point Encryption).

Второй и четвёртый пункт из этого списка полностью решают задачи, определяющие сущность VPN: аутентификацию сторон, обеспечение целостности трафика и защиту от несанкционированного доступа к данным. С учетом сеансового характера соединений, такие сети называются *Virtual Private Dial-up Networks* — VPDN.

ПРИМЕЧАНИЕ На сегодняшний день алгоритм MPPE имеет известные принципиальные уязвимости, позволяющие конструктивно взломать его. Поэтому он представляет собой весьма слабую защиту и не рекомендуется для передачи критически важных данных.

PPPoE и PPTP широко применяются в сетях доступа для аутентификации, авторизации и учёта работы клиентов городских и домовых сетей Ethernet и *x*DSL. PPTP также удобен для решения задачи обратного доступа из головного офиса в филиалы, подключаемые к сетям общего пользования (как проводным, так и беспроводным) посредством базовой услуги — с динамическими приватными IP-адресами. После подключения к Интернет клиентское устройство доступа инициирует со своей стороны установление туннеля до центрального маршрутизатора корпоративной сети. Через этот туннель хосты, расположенные в головном офисе, уже могут инициировать взаимодействие с филиалом, минуя при этом NAT поставщика услуг. (Задачу можно было бы решить по той же схеме и с помощью других типов туннелей, но PPTP из них наиболее прост и удобен для применений, не требующих высокой степени защищённости.)

Существенное отличие PPPoE и PPTP от PPP заключается в том, что эти два протокола — асимметричные, т.е. основаны на чётком разделении ролей клиента и сервера. Клиент всегда инициирует соединение со своей стороны, сервер находится в режиме пассивного ожидания соединений. Как правило, сервер динамически назначает клиенту параметры IP.

§5.3.2. Общие настройки клиентов и серверов VPDN

Настройка туннелей PPPoE и PPTP производится в узлах .tunnel.pppoe и .tunnel.pptp , соответственно. Для создания нового туннеля следует нажать кнопку + или ввести команду _new. Имя туннеля формируется по шаблону pppoe*N* либо pptp*N*, соответственно. (При создании нового туннеля достаточно указать номер, префикс добавляется автоматически.) Для клиента это имя является синонимом его сетевого интерфейса (ppp*M*). Сами интерфейсы PPP/PPTP/PPPoE нумеруются динамически, и если их в системе несколько, то предсказать их номера заранее невозможно; по этой причине для маршрутизации, для описания соединений *ui*TCP и т.п. следует указывать имя туннеля, например, pppoe1.

Режим работы устройства в рамках данной VPDN устанавливается параметром mode = "client" или "server", соответственно. В зависимости от значения этого параметра, в конфигурации туннеля появляются дополнительные узлы.

Во всех случаях настройки туннеля содержат узел ppp.main, в котором находятся все параметры туннелируемого PPP. При настройке этого узла следует руководствоваться здравым смыслом и Частью 2 данного Руководства. Часть параметров PPP в типовых конфигурациях используется на стороне клиента, часть — на стороне сервера, но бывают и "перевёрнутые" конфигурации. Некоторые параметры или их значения в данном узле могут быть рудиментарными и сохранены только для единообразия с PPP на физических портах; применительно к туннелям они не имеют смысла.

ВНИМАНИЕ Параметр authentication определяет протокол, по которому устройство NSG будет *требовать* аутентификацию *от удалённой стороны*, и используется, как правило, на стороне сервера. Список протоколов, по которым устройство NSG — как правило, в качестве клиента — согласно (или, строго говоря, *не* согласно) аутентифицировать *себя на удалённой стороне*, определяется параметром refuse-auth . По умолчанию, принимаются любые протоколы аутентификации, запрошенные удалённой стороной, за исключением EAP.

ВНИМАНИЕ Если устройство NSG (или иная *NIX-система) служит одновременно и в качестве клиента, и в качестве сервера по PPP или производным от него протоколам, настоятельно рекомендуется задавать имена и пароли раздельно: для клиентов — непосредственно в меню соответствующих портов, для серверов — в узле system.ppp-secrets .
Возможно также использовать секрет, состоящий не из 2, а из 3 компонент: имени клиента, имени сервера и пароля. Имя сервера в данном случае позволяет использовать каждый пароль только для той или иной задачи.
В частности, данные меры предосторожности абсолютно необходимы, если устройство подключается в качестве клиента к сотовым сетям GPRS, 3G или CDMA с общеизвестными именем и паролем, и одновременно служит сервером для доступа по PPP или PPTP.

ПРИМЕЧАНИЕ Протокол PPPoE добавляет заголовок длиной 8 байт, PPTP — 33–37 байт, поэтому размер MTU уменьшается на эту величину. Максимальное значение MTU для PPPoE — 1492 байта, для PPTP — 1460 байт. Дополнительно оно может быть уменьшено с помощью параметра ppp.main.mtu .
В частности, при работе через суб-интерфейс VLAN MTU рекомендуется уменьшить на 4 байта. Несоблюдение этой рекомендации, в принципе, не препятствует работе, но может приводить к излишней фрагментации пакетов.

Вне узла ppp находятся параметры и команды, специфические для того или иного протокола или режима работы.

§5.3.3. Клиент PPPoE

Клиент PPPoE всегда привязывается к определённому интерфейсу с инкапсуляцией Ethernet (физическому или виртуальному). При старте клиент рассылает по сети широковещательный запрос для поиска доступных серверов. После того, как сервер (или несколько серверов) откликнулся со своим MAC-адресом, клиент начинает процедуру установления соединения с ним.

Для большей однозначности, если в сети откликаются несколько серверов, можно использовать в качестве критерия имя сервера, с которым он должен работать.

Итак, специфическими параметрами для клиента PPPoE являются:

— Имя интерфейса, на котором он будет работать (iface). Параметр обязательный. На этом же родительском интерфейсе, параллельно с пакетами PPPoE, могут отправляться и приниматься обычные пакеты Ethernet или VLAN, соответственно. Формат, в котором будет отправлен пакет ("обычный" или PPPoE), целиком определяется маршрутизацией.

— Имя сервера (name). Предназначен для выбора нужного сервера, если их в сети обнаруживается несколько. Если имя не указано, то клиент будет работать с первым откликнувшимся. Если в сети заведомо имеется только один сервер, то данный параметр необязателен.

Для отладочных целей предназначены разовые команды show.log и show.discovery . Операция discovery является специфической для процедуры PPPoE и состоит в поиске доступных серверов. Пример вывода:

```
Access-Concentrator: LinuxRH9
Got a cookie: 43 19 cf a8 b3 0b 07 2f ce ea 76 00 e1 14 d5 1e db 20 00 00
AC-Ethernet-Address: 00:0c:6e:41:51:b5
Access-Concentrator: NSGbasicSW
Got a cookie: e8 49 c1 49 5b 94 0a d3 8c 21 d8 ef 99 5c b4 95
AC-Ethernet-Address: 00:09:56:10:05:97
Access-Concentrator: CISCO
Got a cookie: cb 13 b7 11 c0 6a 92 ff fb bf 42 ea 3f 8f 5f 03
AC-Ethernet-Address: 00:02:16:66:7b:40
Access-Concentrator: NSGLinux
Got a cookie: 96 79 58 ff 45 3e 8f 1b f8 af 75 3e 5b 20 90 b6 00 00 00 76
AC-Ethernet-Address: 00:09:56:12:00:fe
```

Здесь в первой строке указаны имена серверов (Access-Concentrator), в третьей строке — их MAC-адреса. В случае, если ни один из серверов не отозвался на процедуру *discovery*, будет выведено сообщение:

```
pppoe: Timeout waiting for PADO packets
```

ПРИМЕЧАНИЕ При работе с PPPoE-серверами компании Cisco Systems рекомендуется использовать на них следующие настройки для корректного согласования размера MTU:

```
vpdn-group <номер>
    ip mtu adjust
```

а также, в некоторых случаях, для корректной обработки фрагментированных пакетов

```
no ip cef
```

§5.3.4. Сервер PPPoE

Специфическими параметрами для сервера PPPoE являются:

— Имя интерфейса, на котором он будет работать (iface). Параметр обязательный. На этом же родительском интерфейсе, параллельно с пакетами PPPoE, могут отправляться и приниматься обычные пакеты Ethernet или VLAN, соответственно. Формат, в котором будет отправлен пакет ("обычный" или PPPoE), целиком определяется маршрутизацией.

— Имя сервера (name), с которым он откликается на запросы клиентов. Имя позволяет клиенту выбрать нужный сервер, если их в сети несколько. Если имя не установлено явно, то вместо него используется значение hostname данного устройства.
Для некоторых клиентов PPPoE (в частности, Microsoft) поле name в ответе сервера является обязательным. В данной версии NSG Linux оно всегда является непустым, и этого достаточно для нормальной работы.

— Максимальное число одновременных сессий, т.е. одновременно работающих пользователей (limit).

— IP-адреса — собственный адрес сервера (ip-address) и начальный адрес из диапазона, выделяемого клиентам (peer-ip-address). Последующим клиентам назначаются адреса, увеличивающиеся на 1. Размер этого диапазона (или максимальный адрес) определяется параметром limit.
Адрес сервера используется один и тот же во всех соединениях и, в общем случае, никак не связан с адресами, которые выдаются клиентам. По умолчанию сервер устанавливает адрес для себя 10.0.0.1 и начальный адрес клиентского пула 10.67.15.1.

ПРИМЕЧАНИЕ Если используется централизованный сервер RADIUS/TACACS+, то в данной версии NSG Linux 2.0 воспринимается только положительный/отрицательный ответ сервера на запрос авторизации. Если сервер присылает другие параметры, в частности, IP-адреса, они игнорируются.

— Максимальное время неактивности сессии на уровне Ethernet (idle). В отличие от аналогичного параметра ppp.main.idle-time, это время относится не только к полезному трафику, передаваемому поверх PPP, а ко всему трафику PPP, передаваемому поверх Ethernet (включая служебные пакеты и т.п.). Рекомендуется использовать этот параметр для защиты сервера от "зависших" сессий. Рекомендуемое значение параметра — в 3–5 раз больше ppp.main.idle-time .

§5.3.5. Клиент PPTP

Протокол PPTP предназначен для передачи пакетов PPP через сеть IP при помощи общего механизма GRE. Для работы этого протокола, помимо потока датаграмм, содержащих полезную нагрузку PPP-over-GRE, организуется управляющее соединение, устанавливаемое от клиента к серверу на порт TCP 1723. Для работы PPTP необходимо, чтобы на стороне сервера было разрешено принимать входящие TCP-пакеты и запросы на установление соединений по данному порту.

Специфическим для клиента PPTP является адрес сервера PPTP (destination). Параметр обязательный.

ВНИМАНИЕ При настройке сервера PPTP необходимо обратить внимание на случай, когда туннельный интерфейс сервера является ненумерованным (*unnumbered*). При такой конфигурации он не должен использовать IP-адрес от того внешнего интерфейса, через который входит туннель.

Дополнительно может быть явным образом указан адрес, который будет подставляться в пакеты PPTP в качестве источника (source-ip). Адрес должен принадлежать одному из интерфейсов устройства. Если адрес не указан, то по умолчанию подставляется адрес того интерфейса, через который уходит пакет. Для контроля доступности удалённой стороны может использоваться механизм *keepalive* в управляющем соединении, имеющий обычные два параметра: интервал оправки запросов (keepalive) и максимальное число неотвеченных подряд (retry). Если keepalive не указан, этот механизм не используется.

Особо стоит остановиться на контроле целостности соединения PPTP, поскольку он может осуществляться в трех местах: на уровне несущего соединения PPP в сети общего пользования, на уровне соединения PPTP и в управляющем соединении PPTP. Рекомендуется использовать контроль только на одном объекте, представляющем собой наиболее слабое звено стека, а именно:

— Для соединений через сотовые сети, коммутируемые модемные линии и другие типы подключений, которые могут быть потенциально ненадежны и неустойчивы — в несущем соединении PPP.

— Для соединений через сети Ethernet и другие надежные среды — в управляющем соединении PPTP.

ПРИМЕЧАНИЕ При работе с PPTP-серверами компании Cisco Systems рекомендуется использовать на них следующие настройки для корректного согласования размера MTU и фрагментации пакетов:

```
no ip cef
vpdn-group <номер>
    ip mtu adjust
```

§5.3.6. Особенности маршрутизации по умолчанию при использовании PPTP

Особая ситуация возникает, если в системе должно быть назначено два маршрута по умолчанию для разных целей. Один маршрут назначается через публичную сеть и нужен, по существу, для того, чтобы пакеты PPTP (как данные, так и управляющее соединение) доходили до сервера. Другой маршрут, через туннель PPTP, предназначен для всех пакетов приватной сети.

Ситуация была бы значительно проще, если бы маршрут через публичную сеть на сервер PPTP был прописан в таблице маршрутизации явным образом. В этом случае маршрут в туннель был бы единственным, используемым по умолчанию, и проблема не возникала бы. Однако весьма часто встречается ситуация, когда настройки публичного интерфейса Ethernet принимаются от поставщика услуг по DHCP, поэтому IP-адрес следующего шлюза в публичной сети заранее неизвестен. Поставщик же может назначить клиенту только маршрут по умолчанию; по этому маршруту нужно прокопать туннель до сервера PPTP и дальше работать через туннель. Такая схема часто используется, например, в домовых сетях.

Чтобы разрешить эту проблему, клиент PPTP производит следующие манипуляции с маршрутами:

— Если перед стартом PPTP существует маршрут по умолчанию, то вычисляется маршрут на сервер PPTP и для него создаётся явная запись с маской /32. По общим правилам маршрутизации, такая запись имеет приоритет перед любой более общей. Далее клиент PPTP стартует, устанавливает соединение с сервером и создаёт свой маршрут по умолчанию. Метрика этого маршрута (в узле pptp.*имя*.ppp) должна быть установлена *меньшей*, чем метрика исходного публичного маршрута (в узле .ip.route, или в узле .port.*имя*.ppp, или в узле .port.*имя*.ifAddress.dhcp-options, в зависимости от того, как он создаётся). При разрыве туннеля PPTP явный маршрут на сервер, созданный для него, удаляется.
  Если метрика маршрута через туннель PPTP равна или больше метрики уже существующего маршрута, то никакие дополнительные маршруты не создаются и не удаляются.

  ПРИМЕЧАНИЕ Для сравнения, в продуктах корпорации Майкрософт в этой ситуации также выполняется аналогичная процедура, но только метрика маршрута в туннель устанавливается минимальной, а метрика старого (публичного) маршрута по умолчанию увеличивается до фиксированного значения 20. В NSG Linux 2.0 метрики обоих маршрутов назначаются вручную, что позволяет более гибко адаптировать систему к требованиям разнообразных корпоративных задач. Однако при этом администратор устройства должен сам следить за правильным соотношением между метриками того и другого маршрутов.

— Если в исходном состоянии вместо маршрута по умолчанию через публичную сеть задан маршрут на некоторую сеть, содержащую искомый сервер PPTP, то производятся те же операции.

— Если в исходном состоянии маршрут на сервер PPTP задан явным образом с маской /32, то никакие дополнительные маршруты не создаются и не удаляются.

§5.3.7. Сервер PPTP

Специфическими параметрами для сервера PPTP являются:

— IP-адрес, на котором он принимает запросы от удалённых клиентов (listen). Параметр обязательный. Адрес должен принадлежать одному из IP-интерфейсов данного устройства (но не может быть адресом локального интерфейса 127.0.0.1).

— Максимальное число одновременных сессий, т.е. одновременно работающих пользователей (limit).

— IP-адреса — собственный адрес сервера (ip-address) и начальный адрес из диапазона, выделяемого клиентам (peer-ip-address). Последующим клиентам назначаются адреса, увеличивающиеся на 1. Размер этого диапазона (или максимальный адрес) ограничивается параметром limit.
Интерфейсы сервера в такой конфигурации являются де-факто ненумерованными, и в качестве ip-address следует использовать адрес, принадлежащий одному из интерфейсов устройства NSG. Для совместимости с некоторыми клиентами PPTP (в т.ч. встроенным в NSG Linux), не следует использовать для сервера PPTP адрес от того интерфейса, через который входят туннели. Рекомендуется использовать для этой цели специальный адрес, назначаемый интерфейсу lo.

Кроме того, сервер PPTP в NSG Linux 2.0 имеет следующие особенности в части настройки и использования параметров PPP (узел ppp.main):

— Режим соединения следует указывать как connection = "permanent". При режиме passive или on-demand соединение может не устанавливаться, поскольку некоторые реализации клиентов после установления управляющего соединения ждут LCP запросы от сервера и не посылают свои.

— Если клиенты запрашивают IP-адреса, то PPTP-сервер NSG последовательно назначает им свободные адреса, начиная от указанного параметром peer-ip-address = "a.b.c.d" . Но при этом изменяется только последний байт адреса, т.е. в любом случае адреса увеличиваются не далее чем до a.b.c.254 — даже если значение limit ещё не достигнуто.
Таким образом, фактическое число клиентов, которые могут быть одновременно подключены к серверу, ограничено параметром limit или адресным диапазоном, в зависимости от того, какое из этих ограничений будет достигнуто раньше.

— Если адреса клиентов не заданы параметром peer-ip-address, тогда по умолчанию им последовательно назначаются адреса 192.168.1.1, 192.168.1.2, etc.

— Если адрес сервера не задан параметром ip-address, тогда по умолчанию он последовательно назначает себе в каждом соединении адреса 192.168.0.1, 192.168.0.2, etc.

— Механизм *keepalive* на управляющем TCP-соединении не настраивается, но используется и имеет фиксированный интервал посылки пакетов — 60 сек. и фиксированное число попыток — 1. Для более точного контроля можно использовать *keepalive* в PPP-соединении.

§5.4. Туннели IPsec — общие сведения

§5.4.1. IPsec и его реализация в NSG Linux 2.0

IPsec — протокол туннелирования 3 уровня, предназначенный для безопасной передачи пакетов IP через сеть общего пользования. Протокол IPsec предусматривает два существенно различных режима работы:

— *Туннельный режим* предназначен для соединения двух приватных IP-подсетей через сеть общего пользования. Пакеты приватных сетей инкапсулируются в новые пакеты IP целиком, вместе с заголовком (включая IP-адреса приватных сетей). Данный режим предназначен для сетевых устройств (шлюзов IPsec), на которых не исполняются, как правило, никакие прикладные программы. Прикладные хосты располагаются в приватных сетях, защищённых этими шлюзами.

— *Транспортный режим* предназначен для соединения двух прикладных хостов, непосредственно подключённых к сети общего пользования. В этом режиме скрывается только содержимое IP-пакета, заголовок пакета остаётся единственным и не изменяется.

Устройства NSG, как сетевое оборудование, предназначены для работы в туннельном режиме. В общем случае, поддерживается также и транспортный режим, который может быть настроен с помощью дополнительных опций.

Реализация IPsec в NSG Linux 2.0 основана на пакете OpenSWAN и включает в себя как встроенный стек IPsec в ядре Linux 2.6 и выше, так и его более раннюю реализацию KLIPS из пакета FreeS/WAN. Оба варианта соответствуют всем стандартам (IETF RFC), относящимся к IPsec, и совместимы как с открытыми Linux-системами, так и с проприетарными продуктами различных производителей.

Процедура установления безопасных туннелей IPsec состоит из двух этапов:

1. Создание *безопасной ассоциации* (Securty Association, SA) — Main Mode. На этом этапе два хоста, пытающиеся установить туннель, должны идентифицировать и аутентифицировать друг друга. Ассоциация характеризуется списком хостов, входящих в неё, способом аутентификации (PSK, RSA-секрет, или сертификаты X.509) и, соответственно, реквизитами для этой аутентификации. Кроме того, для SA может быть установлено ограниченное время существования, по истечении которого она должна быть создана заново.
2. Установление *туннеля* в рамках существующей ассоциации — Quick Mode. Выполняется после того, как два VPN-шлюза признали друг друга членами одной безопасной ассоциации. Туннель характеризуется следующими основными настройками:
   — Описание трафика, направляемого в туннель (де-факто при этом также создаётся маршрут в удалённую приватную сеть).
   — Алгоритм(ы) защиты и аутентификации (контроля целостности) передаваемого трафика.
   — Публичные адреса обеих сторон, маршруты и другие реквизиты туннеля в публичной сети.

   Туннель идентифицируется, в рамках данной SA, уникальным номером — SPI (Security Parameter Index). Для работы туннеля необходимы, в общем случае, два ключа: для аутентификации и для защиты трафика. Эти ключи могут назначаться двумя способами, которые взаимосвязаны с фактом существования туннеля:

   — Постоянно существующий туннель. Создаётся немедленно после образования SA и существует до тех пор, пока не будет удален из конфигурации устройства. Все параметры туннеля — ключи, и алгоритм защиты трафика — назначаются администратором устройства вручную, статически, и должны быть установлены одинаковыми на обеих сторонах. Никакие способы автоматического разрыва туннеля или изменения его параметров в процессе его работы не предусмотрены.

   — Динамически создаваемый туннель. Для создания такого туннеля используется процедура автоматического согласования ключей, определенная протоколом IKE (Internet Key Exchange). В результате образования SA подготавливаются условия для создания туннеля, но непосредственно создание туннеля происходит, как правило, тогда, когда на одной из сторон появляются данные для передачи по туннелю. (Создание туннеля немедленно после образования SA также возможно, но указывается дополнительными опциями.) Ключи, индекс туннеля (SPI) и конкретный алгоритм защиты трафика (из перечня разрешённых на одной и на другой стороне) согласовываются и пересогласовываются автоматически по мере необходимости.
     Созданный туннель может быть разорван любой из участвующих сторон в произвольный момент времени. В частности, это может произойти по следующим причинам:
     — Истечение установленного времени неактивности.
     — Истечение установленного срока жизни туннеля (по времени или объему переданного трафика).
     — Выход из строя удалённого шлюза, или потеря связи с ним, обнаруживаемые при помощи механизма DPD (Dead Peer Detection).
     — Вручную по инициативе администратора одного из VPN-шлюзов, при помощи соответствующей разовой команды.
     После разрыва туннель остается в состоянии готовности и может быть автоматически создан снова по инициативе одной из сторон. При этом его параметры согласовываются заново.

Постоянные туннели IPsec в настоящее время используются крайне редко. Одна из причин состоит в необходимости вручную вводить длинные ключи. Другая, более существенная — в том, что ключи задаются статически, без ограничения срока действия. Они могут быть подобраны за конечное время (по крайней мере, теоретически), причём достаточно небольшое — при современных вычислительных мощностях и, тем более, при распределённой атаке с помощью сети (ботнета) из зомби-компьютеров. По этой причине NSG Linux 2.0 ориентирован на использование только динамических туннелей. Постоянные туннели, тем не менее, также могут быть реализованы с помощью дополнительных опций.

Технология IPsec по сути своей симметрична, т.е. не подразумевает разделения на клиент и сервер. Процедуры аутентификации, согласования параметров и установления туннеля могут выполняться с равным успехом в обе стороны, независимо от того, какая из сторон является инициатором. Однако в практических решениях полная симметрия имеет место достаточно редко. Чаще встречается ситуация, когда VPN-шлюзы функционально различны: мощный высокопроизводительный *сервер* в центральном офисе ожидает соединения от многих удалённых клиентов, а каждый из *клиентов* инициирует эти соединения по мере необходимости и обслуживает значительно меньшую часть приватной сети (удалённый офис, банкомат и т.п.) Как следствие, условия функционирования сервера и клиента также могут различаться по существу: сервер находится на высоконадёжном (Ethernet, Fiber Ethernet) соединении со статическим IP-адресом, в то время как клиенты могут использовать самые разные технологии доступа в Интернет, динамические адреса, с большой вероятностью — NAT из сети поставщика услуг в Интернет, и т.п.

§5.4.2. IPsec и адресный план сети

Технология IPsec, по определению, функционирует на 3 уровне протокольного стека (уровень IP) и обеспечивает взаимодействие IP-подсетей. Говоря более точно о туннельном режиме IPsec, это означает, что:

— Адресные пространства приватных сетей, находящихся по одну и по другую сторону туннеля, не пересекаются друг с другом.

— Адресные пространства приватных сетей не пересекаются с адресным пространством публичной сети, к которой подключён VPN-шлюз.

Это постулаты, принятые безусловно по существу технологии IPsec, и только их выполнение гарантирует, что IPsec будет функционировать должным образом. Нарушение этих правил может приводить к непредсказуемым результатам, в зависимости от конкретной конфигурации и от особенностей конкретной реализации IPsec.

§5.4.3. IPsec и NAT на транзитных устройствах

Технология IPsec в своей изначальной реализации конфликтует с NAT, поскольку NAT модифицирует поля в заголовке публичного IP-пакета, а IPsec контролирует их. Это может быть как Source NAT на стороне клиента, так и Destination NAT на стороне сервера. Проблема имеет две стороны:

— Если устройство находится за NAT, то адрес, под которым оно известно самому себе, не совпадает с адресом, под которым она известна удалённой стороне.

— Поскольку адрес, под которым устройство видно удалённой стороне, может быть заранее неизвестен, то она должна либо разрешать построение SA с любыми партнёрами, либо идентифицировать данное устройство по имени, а не по IP-адресу. Первый вариант проще, но является значительно менее безопасным, поэтому требует использования более сильных средств аутентификации.

Чтобы устанавливать туннели IPsec через промежуточные узлы, осуществляющие NAT, используется стандартный механизм NAT Traversal (NAT–T), использующий дополнительную инкапсуляцию IPsec-over-UDP. При создании *security association* два устройства определяют, производится ли NAT на промежуточных узлах, и при необходимости согласовывают использование NAT–T.

§5.4.4. IPsec и фильтрация пакетов

Технология IPsec использует специфические протоколы 4 уровня (в рамках модели OSI), передаваемые поверх IP (т.е. параллельно с TCP, UDP, ICMP и GRE): ESP (номер протокола — 50) или AH (51), в зависимости от выбранного алгоритма защиты трафика. Для нормальной работы IPsec необходимо, чтобы данный протокол не был запрещён фильтрами на входе самих VPN-шлюзов или где-либо на промежуточных узлах сети. Помимо этого, должны быть разрешены пакеты UDP с портами назначения и источника 500.

При использовании NAT Traversal дополнительно должен быть открыт порт UDP 4500.

Пример настройки разрешающего фильтра строго для входящих пакетов IPsec с сервера 123.45.67.89. Устройство NSG подключено к публичной сети несколькими интерфейсами PPP (например, двумя сотовыми модемами через разных операторов одновременно):

```
ip
: filter
: : INPUT
: : : default-target        = "DROP"
: : : 1
: : : : protocol-num        = 50
: : : : in-interface        = "ppp+"
: : : : source              = "123.45.67.89"
: : : : target              = "ACCEPT"
: : : 2
: : : : protocol-num        = 51
: : : : in-interface        = "ppp+"
: : : : source              = "123.45.67.89"
: : : : target              = "ACCEPT"
: : : 3
: : : : protocol            = "udp"
: : : : in-interface        = "ppp+"
: : : : source              = "123.45.67.89"
: : : : source-port         = "500"
: : : : destination-port    = "500"
: : : : target              = "ACCEPT"
: : : 4
: : : : protocol            = "udp"
: : : : in-interface        = "ppp+"
: : : : source              = "123.45.67.89"
: : : : source-port         = "4500"
: : : : destination-port    = "4500"
: : : : target              = "ACCEPT"
```

ПРИМЕЧАНИЕ Настройку фильтров следует производить в последнюю очередь, предварительно убедившись в нормальной работе остальных компонент системы.

§5.4.5. Отладка IPsec

Отладочные сообщения IPsec удобно выводить в Syslog и просматривать в нём. Включение и просмотр Syslog производятся в узле .system.syslog (см. Часть 1).

Текущее состояние IPsec можно просмотреть командой ipsec.show.status .

§5.5. Туннели IPsec — настройка

§5.5.1. Общие замечания о настройке IPsec в NSG Linux 2.0

Реализация IPsec в NSG Linux 2.0 основана на проекте OpenSWAN и следует методике, принятой в нём. Параметры, содержащиеся внутри узла .tunnel.ipsec, соответствуют наиболее употребительным записям конфигурационных файлов ipsec.conf и ipsec.secrets. В случае необходимости, можно использовать также любые другие настройки, предусмотренные в этих файлах, создавая их в узлах extra-options в формате параметр = "значение" . Подробнее обо всех возможных параметрах и особенностях их использования см. первоисточник — *man pages* по двум вышеуказанным файлам. (Дополнительные параметры могут быть добавлены в меню явным образом по мере получения соответствующих запросов от пользователей.)

Большинство настроек для каждого туннеля находится в файле ipsec.conf и, соответственно, в узлах .tunnel.ipsec.setup и .tunnel.ipsec.connections . В файл ipsec.secrets (узел .tunnel.ipsec.secrets) вынесена секретная информация для взаимной аутентификации сторон, не передаваемая по сети. Эти параметры требуют особых мер безопасности при конфигурировании системы.

Формат настроек IPsec в OpenSWAN имеет одну особенность по сравнению с другими реализациями. Он ориентирован преимущественно на симметричные соединения "точка-точка". При этом, чтобы облегчить работу администратора, было предложено составить по возможности универсальные файлы конфигурации, пригодные для обеих сторон туннеля, и просто копировать их с одного устройства на другое. Конфигурационные файлы составляются, по возможности, симметричными по отношению к обеим сторонам, и две стороны соединения в них именуются не "локальной" и "удалённой", а "левой" и "правой". При этом конфигурация заведомо содержит информацию, избыточную либо для одного, либо для другого устройства; каждое устройство самостоятельно выбирает, какая часть настроек — левая или правая — относится к нему, исходя из параметров других своих компонент (IP-адресов интерфейсов и т.п.).

Хотя выгода такого подхода представляется неочевидной, а в некоторых случаях он принципиально невозможен, и хотя большинство практических инсталляций, как уже сказано выше, имеет функционально асимметричный характер "клиент-сервер", в NSG Linux 2.0 он сохранён для единообразия с другими Linux-решениями. Для большей ясности администратору рекомендуется принять для своей системы некоторое постоянное распределение ролей, например, всегда считать, что **Л**евое устройство — к**Л**иент, п**Р**авое — се**Р**ве**Р** (либо всегда наоборот). Это поможет уменьшить число человеческих ошибок. Кроме того, этот принцип следует держать в уме при чтении документации по настройке OpenSWAN, поскольку некоторые рекомендации, приводимые в *man pages* и других инструкциях, следуют исключительно из него.

В любом случае, принципиальную роль для настройки IPsec (равно как и любой другой системы) играет факт существования того или иного параметра, а не его название в конкретной реализации. Для удобства пользователя, в §5.5.7 приведена таблица соответствия основных параметров IPsec в NSG Linux 2.0 и в Cisco-подобном командном языке.

§5.5.2. Общая настройка IPsec

Узел .tunnel.ipsec.setup содержит глобальные настройки, относящиеся к функционированию подсистемы IPsec в целом. Это узел соответствует секции setup файла ipsec.conf . Наиболее важные настройки в данном разделе:

— Используемая реализация IPsec — параметр protostack. По умолчанию, используется NETKEY — встроенная "родная" реализация IPsec в ядре Linux 2.6 и выше. В некоторых случаях, для совместимости с отдельными устаревшими Linux-системами, может быть предпочтительно использование KLIPS (реализации IPsec из пакета FreeS/WAN) или его модификации MAST.

— Поддержка NAT Traversal (по умолчанию, включена).

Другие настройки, в случае необходимости, могут быть добавлены в узле extra-options. Узел содержит именованный список дополнительных опций, где именем элемента должно являться имя опции, а значением — значение этой опции.

§5.5.3. Настройка Security Association на основе PSK

Узел .tunnel.ipsec.secrets содержит описания безопасных ассоциаций (SA), в которых может участвовать данное устройство (В той части, которая является единой для всех туннелей в рамках одной SA. Отдельные параметры могут устанавливаться индивидуально для каждого туннеля в узле ipsec.connections.имя... .) Каждая SA характеризуется способом аутентификации сторон, реквизитами для этой аутентификации и списком узлов, которые могут быть членами данной ассоциации. Каждая запись в списке psk или rsa определяет одну безопасную ассоциацию и соответствует одной записи в файле ipsec.secrets . Аналогом в Cisco-подобном синтаксисе является команда crypto isakmp ... .

Для аутентификации с помощью *разделяемого ключа* (Pre-Shared Key, PSK) необходимо создать запись в списке psk. Каждая запись содержит собственно ключ (secret) и список узлов (indices). Ключ является общим для всех членов ассоциации.

Список хостов перечисляет узлы, которые могут быть членами данной ассоциации. Каждый хост в этом списке может быть указан одним из способов:

— IP-адресом в явном виде.
— Доменным именем хоста, которое разрешается в IP-адрес с помощью DNS (не рекомендуется, поскольку это не только усложняет систему, но также снижает её надёжность и безопасность: работа IPsec ставится в зависимость от доступности сервера DNS и от его собственной защищённости).
— Именем в формате user@host.domain (поля user и domain — необязательные, символ @ и завершающая точка обязательны). При использовании PSK есть особенность: такой формат пригоден только для идентификации отвечающей стороны (сервера) на устройстве-инициаторе соединения (клиенте). Клиент же может идентифицироваться на сервере только IP-адресом.
— Перечнем основных полей сертификата X.509, если таковые используются; подробно о данном формате см. §5.5.5.
— Записью %any, которой соответствует любой удалённый IP-адрес.

Идентификатор, который *данное* устройство использует при создании SA, указывается индивидуально для каждого туннеля и вводится в поля ipsec.connections.имя.leftid и rightid . (В действительности всегда используется только одно из них).

Список может содержать любое число элементов или быть пустым. В частности:

— Если список пустой, то ему удовлетворяет любая пара хостов.
— Если список содержит ровно одну запись, то она относится к локальному устройству. Идентификатор удалённой стороны при этом игнорируется.
— Если список содержит две или более записи, то для аутентификации с помощью PSK необходимо найти в нём соответствия для обеих сторон. Для аутентификации с помощью асимметричных ключей (RSA-секретов или сертификатов X.509) достаточно, чтобы в списке нашлось соответствие для локального хоста.

PSK является общим для всех членов одной SA. Поэтому удобно внести в одну запись всех членов данной SA (обоих или более), и эту запись можно копировать целиком на все эти хосты.

ВНИМАНИЕ PSK на сегодняшний день считается недостаточно надёжной защитой. Использовать его рекомендуется только при глобальных статических IP-адресах обеих сторон — в этом случае хост де-факто аутентифицируется *сочетанием* адреса и PSK. В противном случае PSK может быть взломан за более или менее разумное время методом подбора или, тем более, распределённого подбора с помощью достаточно большого ботнета.
Если хотя бы одна из сторон имеет динамический адрес и/или находится за NAT, то аутентифицировать её рекомендуется с помощью RSA-секретов или сертификатов.

§5.5.4. Настройка Security Association на основе асимметричных ключей (RSA, X.509)

Для аутентификации с помощью асимметричных RSA-секретов или сертификатов X.509 необходимо создать запись в списке rsa. В обоих случаях используется пара, состоящая из публичного и приватного ключей. Приватный ключ хранится только на данном устройстве, неизвестен никаким другим устройствам, и никогда не передаётся по сети. Публичный ключ должен быть перенесён на удалённое устройство. При попытке установить соединение оно передаёт этот ключ, и если он соответствует приватному, то соединение принимается.

Различие состоит в том, что при использовании RSA-секретов публичный ключ переносится на удалённое устройство вручную заранее, и его аутентичность никак не проверяется. Сущность алгоритма X.509 состоит в том, что аутентичность ключей RSA подтверждается сертификатами, полученными от доверенного центра. Публичный ключ передаётся с локального устройства на удалённое в теле сертификата непосредственно в процессе образования SA. Это существенно более надёжный метод, чем просто пара ключей. Поскольку оба метода примерно одинаково сложны в настройке, то при прочих равных условиях целесообразно использовать, как правило, именно сертификаты. Несертифицированные RSA-секреты на практике используются редко.

Назначение полей каждой записи в узле secrets.rsa зависит от способа задания ключей:

Чтобы сгенерировать **RSA-секрет**, необходимо указать длину ключа (keylength) и выполнить разовую команду generate. Процедура генерации секрета ресурсоёмкая и длительная (до нескольких минут на младших устройствах), поэтому для её выполнения требуется подтверждение. После выполнения (в Web-интерфейсе может быть необходимо обновить экран кнопкой в верхней части экрана) секрет записывается полностью в поле secret, публичный ключ от него можно вывести разовой командой show-public-key .

Если данный секрет предназначен для другого устройства, то его следует скопировать из поля secret и перенести безопасным образом (на Flash, внутри защищённой корпоративной сети и т.п. — но не передавать по сетям общего пользования) на это устройство. Публичный ключ от этого секрета также необходимо извлечь, чтобы установить его на устройстве NSG. Последним генерируется секрет для самого устройства NSG, а публичный ключ от него переносится на удалённое устройство.

Если, наоборот, секрет для данного устройства NSG генерируется сторонним удостоверяющим центром, то его необходимо перенести на устройство и вести в поле secret .

Публичные ключи для обеих сторон указываются индивидуально для каждого туннеля и вводятся в поля ipsec.connections.*имя*.leftrsasigkey и rightrsasigkey . В действительности каждому устройству требуется знать только свой приватный ключ и публичный ключ удалённой стороны.

Поле file в данном случае не используется.

Приватный ключ X.509 хранится в отдельном файле, путь к которому указывается в поле file. Иначе говоря, если это поле не пустое, то аутентификация производится с помощью сертификатов, если пустое — то с помощью RSA-секретов.

Сертификаты для обеих сторон указываются индивидуально для каждого туннеля и вводятся в поля ipsec.connections.*имя*.leftcert и rightcert . В действительности каждому устройству требуется знать только свой приватный ключ, свой сертификат (публичный ключ содержится в сертификате) и сертификат общего удостоверяющего центра (корневой сертификат). Подробнее об этой части настройки см. следующий параграф.

Если для ключей и сертификатов указываются относительные пути, то они отсчитываются от директорий /etc/ipsec.d/private и /etc/ipsec.d/certs , соответственно. Если они хранятся в ином месте, то необходимо указать абсолютный путь, например, /mnt/usbstorage/my_keys или /mnt/usbstorage/certs . В обоих случаях файл может быть записан в формате PEM или DER (при этом суффикс имени файла не имеет значения, анализируется содержимое файла по существу). Если файл ключа защищён паролем, то пароль следует ввести в поле secret .

Сертификат удостоверяющего центра в данной версии NSG Linux 2.0 должен храниться в директории /etc/ipsec.d/cacerts , и его местоположение не настраивается. Автоматизированная процедура управления сертификатами X.509 для IPsec планируется к реализации в последующих версиях NSG Linux 2.0.

Список хостов, входящих в состав SA, в данном случае трактуется иначе, чем для SA на основе PSK (см. пред. параграф). По существу, асимметричные методы предназначены для ситуаций, в которых адрес одной или даже обеих сторон априори неизвестен, или искажён при прохождении через NAT. Для аутентификации с помощью асимметричных ключей достаточно, чтобы в списке нашлось соответствие для локального хоста; идентификатор удалённой стороны игнорируется, а другие элементы списка, если они есть, рассматриваются как другие возможные идентификаторы данного хоста (например, другие IP-адреса, принадлежащие ему). Поэтому список хостов в данном случае обычно содержит единственный элемент — локальный хост, или не используется вовсе.

ПРИМЕЧАНИЕ Асимметричные секреты предполагают, что каждая из сторон не знает приватного ключа другой. Таким образом, записи secrets.rsa объективно уникальны для каждого хоста, вопреки общей тенденции для OpenSWAN — иметь единый файл конфигурации.

§5.5.5. Настройка туннелей

Для описания каждого нового туннеля необходимо создать новый элемент в списке .tunnel.ipsec.connections . Элементы этого списка соответствуют секциям conn в файле ipsec.conf . Бóльшая часть настроек каждого туннеля содержится в этом узле. Если на устройстве задаётся несколько туннелей (например, для сервера, к которому будут подключаться несколько клиентов), то общую часть настроек для всех туннелей можно вынести в элемент со специальным именем %default . При этом любой параметр, содержащийся в этой записи, можно переопределить впоследствии; значение, установленное для конкретного туннеля, всегда имеет приоритет.

ВНИМАНИЕ Особым является параметр auto (см. ниже). Для него значением по умолчанию является ignore, поэтому его необходимо указывать в конфигурации каждого туннеля явным образом.

Как отмечено выше, файл ipsec.conf является избыточным и содержит настройки для обоих сторон туннеля (левой и правой), чтобы его можно было просто копировать с одного устройства на другое; таким же образом в NSG Linux 2.0 можно копировать ветви setup и connections (но ветвь secrets — только при использовании PSK). Все параметры можно разделить на несколько функциональных групп:

Характеристики SA. Первая часть параметров определяет SA, в рамках которой будет создаваться туннель. К ним относятся:

— Способ аутентификации (authby): PSK (значение secret), асимметричный ключ (rsasig), любой из двух или без аутентификации (имеет смысл только для прямых соединений между двумя шлюзами).

— Реквизиты для аутентификации (при использовании асимметричных ключей). По существу, каждая из сторон должна иметь либо заранее известный публичный ключ партнёра (в случае RSA-секретов без сертификатов), либо свой собственный сертификат (при обмене сертификатами передаются, как их составная часть, публичные ключи). Свой собственный приватный ключ, либо его местоположение и пароль к нему, задаётся в узле secrets.

  В случае RSA-секретов это идентификаторы устройств (left/rightid) и публичные ключи (left/rightrsasigkey). Идентификатор служит указателем, по которому подбирается нужная запись в узле secrets. Используется только свой идентификатор. Подробно о других форматах идентификаторов см. §5.5.3.

  ВНИМАНИЕ Если на устройстве создаётся два или более туннелей с разными RSA-секретами, то они должны иметь различные идентификаторы.

  В случае X.509 настраивается следующий набор параметров для обеих сторон:

  1. Файлы сертификатов (left/rightcert). По существу, каждая из сторон использует только свой собственный сертификат, а сертификат удалённой стороны получает непосредственно в процессе образования SA.
  2. Идентификаторы сторон (left/rightid) должны строго соответствовать полю unstructuredName их сертификатов и записываются в формате:
     "/C=*страна*/ST=*регион*/L=*город*/O=*организация*/OU=*подразделение*/CN=*имя*/emailAddress=*имя@сервер*"
     Оба идентификатора используются по существу: свой передаётся удалённой стороне, чужой — идентифицирует туннель среди многих, которые могут быть определены на данном устройстве. В некоторых ситуациях свой идентификатор может быть опущен, тогда устройство сначала передаёт удалённой стороне в качестве идентификатора свой IP-адрес, а когда эта попытка завершается неудачей, автоматически генерирует нужную строку из полей сертификата и повторяет попытку. Но это работает только в том случае, если реализация IPsec на обеих сторонах предусматривает и допускает такую последовательность попыток. Во избежание ошибок рекомендуется *всегда указывать оба* идентификатора на обеих сторонах.
  3. В полях left/rightrsasigkey указывается специальное значение:
     leftsigkey = "%cert"
     rightsigkey = "%cert"
     По существу используется только запись, относящаяся к удалённой стороне. Она означает, что её публичный ключ извлекается из её сертификата.

  ПРИМЕЧАНИЕ Реализация IPsec в данной версии NSG Linux 2.0 имеет следующие особенности:
  1. Приватный ключ обязательно должен быть защищён паролем.
  2. Вышеописанные идентификаторы сторон (left/rightid) не должны содержать пробелы и другие спецсимволы.
  3. Если удалённая сторона (left/right) задана явным образом (по публичному IP-адресу, имени и др.), то поле *cert для этой стороны должно быть пустым — вопреки общей идее OpenSWAN о тождественных конфигурациях на обеих сторонах.

— Время жизни созданной безопасной ассоциации (ikelifetime), в терминах стандартов IPsec — *ISAKMP lifetime*. Аналогичная настройка в Cisco-подобных языках
  crypto isakmp policy ...
    lifetime ...

ВНИМАНИЕ В асимметричных конфигурациях "клиент-сервер" значение ikelifetime на отвечающей стороне (сервере) следует устанавливать меньше, чем на вызывающей стороне (клиенте). Это обеспечивает более устойчивую процедуру пересогласования SA.

ПРИМЕЧАНИЕ Время жизни SA не тождественно времени жизни туннеля в рамках SA (*IPsec lifetime*, см. ниже). Это две настройки, различные по существу, и они устанавливаются разными параметрами.
В частности, время жизни SA может быть даже меньше, чем время жизни туннеля; в этом случае по истечении *ISAKMP lifetime* SA удаляется, но туннель продолжает работать столько, сколько ему положено. По истечении *IPsec lifetime* туннель начинает пересогласовываться, что, в свою очередь, приводит к переустановлению сначала SA, а затем — собственно туннеля.

Описание приватных сетей и трафика между ними. Параметры leftsubnet и rightsubnet в совокупности характеризуют трафик, который должен быть направлен в туннель с одной и с другой стороны (по IP-адресам источника и назначения). Обе сети записываются в формате адрес сети/длина маски. В каждом из параметров можно указать несколько сетей, через пробел. (Фактически в конфигурации *openswan* это будет параметр left-/rightsubnets , а не left-/rightsubnet, однако в командных оболочках NSG синтаксис сохранён для совместимости с конфигурациями от предыдущих версий.)

ВНИМАНИЕ По существу технологии IPsec предполагается, что левая и правая (или локальная и удалённая) сети есть независимые подсети IP и не пересекаются по IP-адресам ни друг с другом, ни с публичной сетью. В противном случае результаты могут быть непредсказуемыми, в зависимости от особенностей реализации IPsec на обеих сторонах.

Дополнительно к отбору пакетов по IP-адресам, реализация IPsec в NSG Linux 2.0 допускает проверку по протоколам 4 уровня. Параметры для настройки протоколов в данной версии NSG Linux 2.0 не предусмотрены явным образом, но могут быть заданы, в случае необходимости, через поле extra-options :

```
connections
: имя
: : extra-options
: : : leftprotoport      = "протокол[/порт]"
: : : rightprotoport     = "протокол[/порт]"
```

Протокол здесь может быть указан как номером, так и алфавитным именем. Для TCP и UDP можно указать специфический номер порта, как по номеру, так и по стандартному имени прикладного протокола, а также в виде записи %any. Значения протоколов и портов должны быть настроены согласованным образом на обеих сторонах.

В терминах настройки IPsec в Cisco-подобных языках, эти параметры соответствуют расширенному access-list, который используется для отбора трафика в туннель. Отличие состоит в том, что вместо шаблонов используются только сплошные маски, шаблоны с чередованием нулей и единиц не допускаются. Для практических целей это несущественно, поскольку все интересующие подсети можно перечислить по отдельности.

Дополнительно к описанию обеих сетей как таковых, реализация IPsec в Linux требует явно указывать IP-адреса приватных интерфейсов обоих устройств (left/rightsourceip, по существу используется только свой параметр). Данные параметры необходимы для автоматической настройки маршрутизации при создании туннеля. Прямого аналога в конфигурации Cisco они не имеют.

Описание допустимых **алгоритмов** защиты и аутентификации данных — узел esp. Аналог в Cisco-подобных языках — crypto transform-set .

Узел содержит список наиболее употребительных алгоритмов (другие алгоритмы могут быть добавлены в поле extra, через запятую). При установлении туннеля (на этапе ISAKMP Phase2) устройство NSG передаёт удалённой стороне весь список алгоритмов, для которых установлено значение true . Выбор алгоритма зависит от аналогичных настроек на удалённой стороне. Если соединение, наоборот, инициировано удалённой стороной, то из предложений, присланных ею, будет выбрано первое, входящее в данный список.

ПРИМЕЧАНИЕ Некоторые редко используемые алгоритмы не только не перечислены в списке, но и не включены в данную версию NSG Linux 2.0 фактически. Попытка их указания и использования будет завершена с ошибкой. При необходимости включения таких алгоритмов в следующие версии следует обратиться в службу технической поддержки NSG.

Для построения туннелей без шифрования (с одной только аутентификацией данных) следует добавить опцию auth="ah" в узле extra-options данного туннеля и соответствующий список алгоритмов аутентификации в поле esp.extra .

ПРИМЕЧАНИЕ При согласовании устройство NSG посылает удалённой стороне сразу весь набор разрешённых алгоритмов (в отличие от реализации IPsec в Cisco и Microsoft, посылающей предложения поочерёдно в соответствии с установленными политиками).

Характеристики собственно туннеля. Данная группа параметров управляет поведением туннеля и его представлением в публичной сети.

IP-адреса публичных интерфейсов обоих устройств (left/right). Помимо явного указания адресов, допустимы также следующие специальные значения:

%defaultroute — Адрес публичного интерфейса, а также left/rightnexthop , определяются автоматически как *default gateway* из текущей таблицы маршрутизации на момент старта IPsec. Из двух сторон соединения данное значение допускается только на одной. Данная конфигурация необходима, в данной версии NSG Linux, для всех клиентов, работающих с динамических IP-адресов и/или из приватных сетей через NAT.

%any — Любое значение, предлагаемое в ходе установления соединения. Используется, как правило, для сервера, если клиенты работают с динамических адресов.

%opportunistic — IP-адреса left и leftnexthop определяются с помощью DNS (клиентом, работающим на данной машине).

В терминах настройки IPsec в Cisco-подобных языках, один из параметров left и right определяет интерфейс, на котором строится туннель, а другой соответствует параметру

```
crypto map ...
    set peer ...
```

Адреса ближайших шлюзов для обеих машин (left/rightnexthop). Помимо явного указания адресов, допускаются также следующие специальные значения:

%defaultroute — Адрес шлюза определяется автоматически как *default gateway* из текущей таблицы маршрутизации на момент старта IPsec. Данная конфигурация необходима, в данной версии NSG Linux, для всех клиентов, работающих с динамических IP-адресов.

%direct — Прямое соединение без промежуточных шлюзов. Вместо шлюза подставляется адрес другой стороны соединения. (Установка по умолчанию.)

ПРИМЕЧАНИЕ Если установлено left/right="%defaultroute" , то значение left/rightnexthop , соответственно, не должно быть указано явным образом. Фактически в этом случае вообще не следует использовать данный параметр. Допускается запись left/rightnexthop="%defaultroute" .

Из двух параметров leftnexthop и rightnexthop каждая сторона соединения использует только один. В терминах настройки IPsec в Cisco-подобных языках, он соответствует параметру

```
crypto map ...
    set nexthop ...
```

Режим PFS (Perfect Forward Security) — повторного согласования длины ключа на этапе Quick Mode. В этом случае успешный подбор текущего ключа не компрометирует все предыдущие ключи. Поддерживаются группы Диффи-Хеллмана 2 либо 5 (1024 и 1536 бит, соответственно). Группа 1 (768 бит) в NSG Linux не поддерживается ввиду её недостаточной криптостойкости при современных вычислительных ресурсах.

ПРИМЕЧАНИЕ Длина ключа безопасной пересылки (PFS) может согласовываться как на стадии Main Mode, так и на стадии Quick Mode. На стадии Main Mode согласование длины ключа на устройствах NSG включено безусловно.

Аналогичная настройка для Cisco-подобных языков — set pfs group ... / no set pfs .

Совместная работа возможна только в следующих случаях:

— PFS выключено на обеих сторонах туннеля.
— На устройстве NSG согласование PFS включено, а на устройстве Cisco выбрана группа 2 или 5.

При всех других возможных сочетаниях настроек туннель установлен не будет.

Срок жизни ключа туннеля (keylife), в терминах стандартов IPsec — *IPsec lifetime*. По истечении данного времени туннель разрывается и переустанавливается заново с новым ключом. Аналогичная настройка в Cisco-подобных языках

```
crypto map ...
    set security-association lifetime ...
```

Действия с данным туннелем при старте IPsec (auto). Возможные варианты:

add — Добавить туннель в список известных системе. Туннель будет записан в конфигурацию, но не будет использоваться. Чтобы использовать его, нужно изменить данный параметр на route или start.

route — Добавить туннель в список и создать защищенный маршрут в сеть, расположенную на другой стороне туннеля. Фактическое установление туннеля произойдёт по требованию, при наличии данных на передачу.

start Добавить туннель в список, создать маршрут и фактически установить туннель.

ignore Ничего не делать.

Параметр относится только к данному устройству; на удалённой стороне, в принципе, он может быть установлен иначе. Как правило, для симметричных конфигураций целесообразно использовать одинаковое значение на обеих сторонах туннеля. В частности, для туннеля, который должен создаваться при включении устройств и поддерживаться постоянно, следует установить start на обеих сторонах, с тем, чтобы перезагрузка любой из сторон приводила к немедленному переустановлению туннеля.

При использовании общей части %default данный параметр необходимо указывать отдельно явным образом в конфигурации каждого туннеля.

Настройки механизма Dead Peer Detection — интервал между посылками пакетов *DPD keepalive* и максимальный интервал неактивности партнёра (dpddelay, dpdtimeout). В терминах Cisco-подобной конфигурации

```
crypto map ...
    keepalive ... waiting ...
```

dpddelay соответствует первому параметру, а dpdtimeout — произведению интервала и максимального числа попыток. Настройки должны быть согласованы на обеих сторонах.

Если пакеты *DPD keepalive* не проходят в течение времени dpdtimeout, то будет выполнено действие, указанное параметром dpdaction (по умолчанию — clear, в отличие от принятого в OpenSWAN). В асимметричных конфигурациях "клиент-сервер" рекомендуется устанавливать для него значение clear на отвечающей стороне (сервере) и restart на вызывающей стороне (клиенте).

§5.5.6. IPsec и локальный NAT

В данной реализации IPsec реализован на более низком подуровне, чем NAT и фильтры, т.е. "ближе к периферии устройства". Преобразование пакетов производится следующим образом:

Для исходящих пакетов в публичную сеть — пакет сначала подвергается преобразованию Source NAT или Masquerading. После этого выполняется проверка на предмет того, не подлежит ли он передаче в защищённом виде — в первую очередь, по IP-адресам источника и назначения (left/rightsubnet). Но к этому моменту адрес источника в нём уже изменён на публичный адрес интерфейса! В результате пакет не попадает в туннель и уходит в публичную сеть в открытом виде. (Или он вообще уничтожается запрещающим фильтром.)

Такая ситуация возникает в самой типичной задаче, когда требуется весь трафик в удалённый офис отправить через туннель, а весь остальной трафик — в публичную сеть через этот же самый интерфейс. Чтобы устранить проблему, необходимо добавить в цепочки NAT исключающее правило для пакетов между приватными подсетями. Это правило должно иметь более высокий приоритет. Пример соответствующих фрагментов конфигурации (left — локальное устройство, right — удалённое):

```
ip
: nat
: : POSTROUTING
: : : 1
: : : : source          = "172.16.0.0/16"
: : : : destination     = "172.17.0.0/16"
: : : : target          = "ACCEPT"
: : : 2
: : : : out-interface   = "eth1"
: : : : target          = "MASQUERADE"
tunnel
: ipsec
: : connections
: : : ToTheOtherOffice
: : : : leftsubnet      = "172.16.0.0/16"
: : : : rightsubnet     = "172.17.0.0/16"
```

Для входящих пакетов из публичной сети — сначала выполняется распаковка пакетов IPsec, и из них извлекаются приватные пакеты в том виде, в каком они существовали в удалённом сегменте приватной сети. Эти пакеты заново поступают на вход IP-стека и проходят все процедуры NAT, маршрутизации и т.п. в обычном порядке, с самого начала. В частности, если смотреть трафик на публичном интерфейсе с помощью утилиты tcpdump, то в трассе каждый входящий пакет будет фигурировать дважды: сначала в защищённом виде, потом в открытом.

§5.5.7. Соответствие настроек Linux и Cisco

Для пользователей, знакомых с настройкой IPsec в реализации Cisco Systems и сходных с ней по языку конфигурирования (в т.ч. NSG Linux 1.0), приведём суммарную таблицу эквивалентов для основных команд.

Linux	Cisco
secrets.psk	crypto isakmp key …
secrets.rsa	crypto isakmp policy … authentication rsa-sig
connections.*имя*.left/rightsubnet	access-list
connections.*имя*.left/rightsourceip	Аналог отсутствует. Маршрутизация в удалённые защищаемые сети настраивается вручную.
connections.*имя*.esp	crypto transform-set
connections.*имя*.left/right	Один из параметров определяет интерфейс, на котором включается crypto map, другой соответствует crypto map … set peer
connections.*имя*.left/rightnexthop	crypto map … set nexthop Другой параметр не используется.
connections.*имя*.pfs	set pfs group … / no set pfs
connections.*имя*.ikelifetime	crypto isakmp policy … lifetime
connections.*имя*.keylife	crypto map … set security-association lifetime
connections.*имя*.dpddelay connections.*имя*.dpdtimeout	crypto map … keepalive …

§5.5.8. Особенности настройки IPsec в NSG Linux 1.0

Одно существенное различие между NSG Linux 2.0 и 1.0 состоит в том, что в последнем не предусмотрен отбор пакетов в туннель с учётом протоколов 4 уровня. Хотя формально в NSG Linux 1.0 *access-list* может содержать указание специфического протокола (например, tcp), оно не имеет никакой силы и в туннель отбирается весь IP-трафик между указанными сетями, независимо от протокола 4 уровня. Если на другой стороне туннеля используется NSG Linux 2.0 или оборудование Cisco, то на нём в аналогичной настройке должен быть обязательно указан протокол IP в целом; в противном случае соединение установлено не будет.

§5.5.9. Особенности настройки IPsec в продуктах Майкрософт

Если на другой стороне туннеля IPsec используется программный шлюз под управлением одной из ОС семейства Windows, то при его настройке необходимо обратить внимание на следующие моменты:

— Перед началом настройки IPsec следует настроить обычную маршрутизацию на обеих сторонах и убедиться в нормальном прохождении пакетов из одного сегмента защищаемой сети в другой. Настройку маршрутизации в продуктах Майкрософт удобнее всего производить из командной строки при помощи команды route -p add … (для справки см. route help), но можно сделать это и путём кликания мышкой в окошках.

— Настройка производится в оснастке "Локальная политика безопасности". Для работы IPsec необходимо создать *политику*, содержащую два *правила безопасности IP* — для трафика в одну и в другую сторону.

— Флаг PFS в окне "Параметры обмена ключами" необходимо согласовать со значением на устройстве NSG.

Пример настройки см. в Приложении А, §5–А.7.

Поскольку реализация IPsec в продуктах Майкрософт основана на решении Cisco, то логично ожидать, что для них также имеет место проблема зацикливания пакетов с адресами источника и назначения, равными адресами локального и удалённого шлюзов (см. пункт "г" в п.5.5.10).

Реализация IPsec в продуктах Майкрософт сама по себе не предусматривает работу в транспортном режиме IPsec, например, на одиночном ПК, подключённом через сети общего пользования. Однако существует ряд программных продуктов других производителей, позволяющих решить эту задачу (де-факто — путём организации программного шлюза IPsec в рамках ОС Windows). Примеры настройки некоторых программных клиентов см. в Приложении А, §5–А.8.

§5.5.10. Особенности реализации IPsec в устройствах Cisco Systems

Стандарты и спецификации IPsec допускают неоднозначное толкование некоторых деталей. Кроме того, они не определяют некоторые смежные вопросы функционирования устройства. Различные производители могут по-разному интерпретировать эти моменты, и возникающие отличия следует взаимно учитывать при установлении туннелей между их устройствами. В частности, при настройке оборудования Cisco Systems совместно с NSG Linux 2.0 необходимо обратить внимание на следующие моменты.

а) Маршрутизация при использовании туннелей

В маршрутизаторах Cisco наличие туннеля само по себе не оказывает никакого влияния на таблицу маршрутизации. Иначе говоря, помимо создания туннеля, необходимо вручную сконфигурировать маршрут в удаленную сеть, находящуюся на другой стороне туннеля (обычный статический маршрут). Пример конфигурации (фрагмент, непосредственно связанный с маршрутизацией):

```
!
interface FastEthernet0/0
ip address 10.0.0.31 255.0.0.0
!
crypto map test1 1 ipsec-manual
set peer 10.0.2.11
.....
!
ip route 192.168.1.0 255.255.255.0 10.0.2.11
!
```

Здесь 10.0.0.31 — IP-адрес интерфейса Cisco, 10.0.2.11 — IP-адрес удаленного маршрутизатора. Последняя строка означает, что неизвестная для данного устройства сеть 192.168.1.0 с маской 255.255.255.0 находится за точкой 10.0.2.11.

При удалении туннеля следует удалить и статические маршруты, проходящие через этот туннель.

В NSG Linux 2.0 для создания маршрутов в удаленные сегменты приватной сети необходимо и достаточно указать значения left/rightsourceip (де-факто каждому устройству требуется только адрес его собственного приватного интерфейса). Тогда при установлении и удалении туннелей автоматически будут создаться/удаляться и соответствующие записи в таблице маршрутизации.

б) Организация туннеля — стадия MAIN MODE

1. При инициализации туннеля со стороны NSG предлагается сразу весь (!) пакет предложений, содержащихся в узле esp .
2. При получении запроса на установление туннеля список предложений, поступивший от удаленной стороны, поочередно сравнивается в приведенным выше списком. Первый совпавший вариант отсылается в качестве подтверждения (выбора).
3. В Cisco все варианты туннелей ISAKMP образуют приоритетное множество предложений (*policies*), которые при посылке отсылаются в порядке, определяемом приоритетом *policy*, а при приеме предложений начинают проверяться в соответствии с приоритетом.

в) Организация туннеля — стадия QUICK MODE

1. При инициализации туннеля со стороны NSG предлагается (или выбирается из предложенных) пакет из всех предложений, перечисленных в узле esp .
2. В маршрутизаторах Cisco конкретное множество правил преобразования и их приоритет определяются в самом описании crypto-map . Это устанавливается перечислением в параметре

   (config-crypto-map)# transform-set <предложение_1> <предложение_2> <предложение_3> ...

   Если инициатором соединения был удаленный маршрутизатор, то устройство NSG выберет из присланных ему алгоритмов первый, который будет найден в списке esp .

г) Использование протоколов динамической маршрутизации и рекурсивное попадание пакетов в туннель

При использовании протоколов динамической маршрутизации (RIP, OSPF и др.) внутри туннелей IPsec необходимо учитывать, что пакеты этих протоколов должны были бы иметь IP-адреса источника и назначения, совпадающие с адресами туннельных интерфейсов. Реализация IPsec в продуктах Cisco не допускает такую ситуацию, поскольку в этом случае зашифрованный пакет снова подпадает под критерии отбора для шифрования, и процесс зацикливается. Пакет бесконечно возвращается на этап шифрования и никогда не будет отправлен.

Для устранения данной проблемы в программном обеспечении Cisco реализован специальный механизм Virtual Tunnel Interface (VTI) и специальный тип объектов VirtualAccess. Возможно также использовать в *access-list* фильтрацию по типу протокола 4 уровня.

§5.5.11. Принудительный рестарт IPsec при изменении или рестарте канала связи

Если IPsec используется поверх какого-либо неустойчивого канала связи (сотовым, комумтируемым телефонным), или имеется выбор между несколькими разными каналами (например, проводным и сотовым, или двумя сотовыми), то при переустановлении соединения или при переходе на другой канал IP-адрес устройства меняется. Следовательно, для удалённой стороны она становится уже другим партнёром, с ним нужно устанавливать туннель заново, а старый туннель разорвать.

Поскольку технология IPsec предназначена для систем общего типа, она не привязана ни к каким другим аппаратным или программным компонентом и взаимодействует только с IP-маршрутизацией. Если для описания данной стороны соединения используется директива %defaultroute, то при изменении маршрута по умолчанию IPsec рестартует немедленно. Во всех остальных случаях IPsec не получает никакой информации об изменениях обстановки, а задача слежения за ней решается с помощью механизма встроенного механизма IPsec — DPD. Пакеты, отсылаемые по старому туннелю, не будут доходить до адресата, через какое-то время DPD сработает и инициирует рестарт туннеля; новый туннель будет построен уже по новому маршруту и от нового адреса. Достоинство этого механизма в его универсальности: он, как и IPsec в целом, разработан без какой-либо привязки к событиям на нижележащих уровнях. Но оборотная сторона этого достоинства в том, что он не может работать никак иначе, кроме как по непрохождению пакетов в течение определённого времени. На практике рекомендуемое время срабатывания DPD составляет 90–180 сек, иногда больше.

В устройствах NSG процесс переустановления туннеля (кроме вышеуказанного случая с маршрутом по умолчанию) можно ускорить, используя обработчик событий. Тогда IPsec будет рестартовать немедленно, как только изменилось состояние публичных интерфейсов или маршрутов. Например, если m1 — единственный сотовый порт:

```
services
: event-handler
: : 1
: : : virt-sensor    = ifstate.m1
: : : state          = up
: : : prev-state     = other
: : : script         = "ipsec setup restart"
```

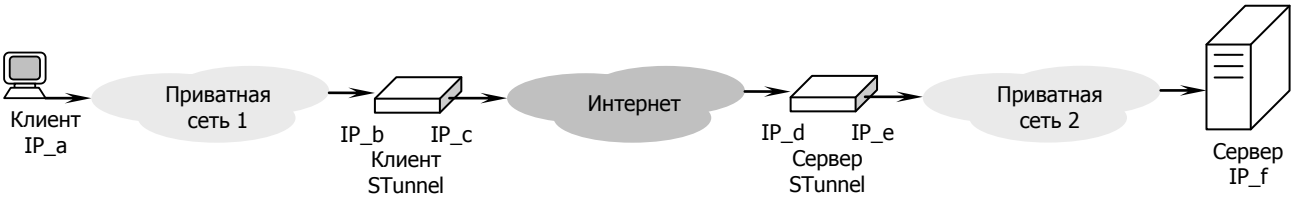
Если сотовых портов два и они ранжированы по метрикам маршрутов (основной и резервный), то рестартовать IPsec следует при переходе основного порта и в UP, и в DOWN. Если контроль основного канала Ethernet (или другого Ethernet-подобного) осуществляется с помощью netping, то команду ipsec setup restart; следует добавить в оба скрипта netping (см. Часть 4).

ВНИМАНИЕ Если данное устройство описано в left/right как %defaultroute, а маршрут на удалённый шлюз указан как маршрут по умолчанию и изменяется при изменении каналов связи, то принудительный рестарт IPsec использовать не следует.

§5.6. Туннели STunnel

§5.6.1. Общее описание технологии Stunnel

STunnel — технология туннелирования 4 уровня (в терминах модели OSI), предназначенная для защиты индивидуальных TCP-соединений. Как и технологии туннелирования 2 уровня, она принципиально асимметрична и подразумевает постоянное разделение ролей между вызывающим хостом (прикладным клиентом) и отвечающим хостом (прикладным сервером). Связка из клиента и сервера STunnel, находящихся на границе одной и другой корпоративной подсетей, соответственно, с сетью общего пользования, всегда выполняет роль TCP-прокси между ними. Схема функционирования STunnel показана на рисунке.



Клиенту, который имеет IP-адрес IP_a, необходимо обратиться к серверу, расположенному по адресу IP_f в другой части корпоративной сети. Процедура разбивается на 3 последовательных соединения:

1. Вместо сервера IP_f клиент обращается к локальному клиенту STunnel по адресу IP_b и некоторому номеру порта TCP назначения (например, 80 — HTTP).
2. Клиент STunnel принимает соединение на этом порту и устанавливает новое TCP соединение с сервером STunnel через публичную сеть, с адреса IP_c на адрес IP_d, по тому же или другому номеру порта TCP (например, 443 — HTTPS).
3. Сервер STunnel принимает это соединение и устанавливает третье соединение со своего внутреннего адреса IP_e на адрес IP_f. Номер порта TCP назначения в этом соединении может, в общем случае, как совпадать с портом, использованным в первом (что наиболее логично) или втором соединении, так и отличаться от них.

При этом на втором этапе задействуется SSL — универсальный механизм, лежащий в основе многих безопасных технологий: HTTPS, OpenVPN и др. Для защиты данных при передаче через публичную сеть используется асимметричная пара ключей RSA, подтвержденная сертификатом X.509. Для сервера STunnel наличие ключа и сертификата является обязательным, поскольку в противном случае соединение не может быть установлено. Для клиента STunnel они опциональны, в зависимости от того, с какими настройками (требовать ли аутентификации от клиента или нет) работает сервер.

Поддержка STunnel в NSG Linux 2.0 реализована на основе стандартного пакета STunnel и настраивается в терминах файла stunnel.conf. Подробно обо всех возможных опциях см. *man pages* по stunnel.conf.

§5.6.2. Настройка пользовательских туннелей Stunnel

Настройка туннелей STunnel производится в узле .tunnel.stunnel.*имя_туннеля* и включает в себя:

1. Выбор режима работы данного устройства в туннеле STunnel: клиент либо сервер.
2. Назначение IP-адресов и портов. Устанавливаются параметрами accept и connect в формате host:port . Точный смысл этих параметров зависит от выбранного режима:
   - — В режиме сервера параметр accept относится к входящим защищённым соединениям, а connect — к исходящим открытым соединениям с прикладным сервером.
   - — В режиме клиента параметр accept относится к входящим открытым соединениям от прикладного клиента, а connect — к исходящим защищённым соединениям.

   В обоих случаях IP-адрес, указанный в параметре accept, должен принадлежать данному устройству NSG. По умолчанию, если этот адрес не указан явно, то принимаются соединения с любым адресом назначения, принадлежащим этому устройству. Явное указание позволяет выбрать только один из этих адресов.

   В параметре connect, если адрес удалённого хоста не указан явно, то подразумевается localhost или 127.0.0.1, т.е. само устройство NSG.

   Дополнительно может использоваться параметр local — явное указание IP-адреса, который будет использоваться в исходящих соединениях в качестве адреса источника. Адрес должен принадлежать одному из интерфейсов данного устройства. Если адрес не указан явно, то по умолчанию подставляется IP-адрес того интерфейса, через который отправляются пакеты.

3. Настройка безопасности. В первую очередь необходимо указать используемую версию протокола SSL (sslVersion) и режим аутентификации удалённой стороны (verify). Сервер должен аутентифицироваться на стороне клиента всегда, клиент на стороне сервера — опционально. Для того, чтобы аутентифицировать удалённую сторону, устройство должно иметь файл с корневым сертификатом. Для сертификации самого себя на удалённой стороне устройство должно иметь также собственный закрытый ключ и сертификат к нему.

   Соответственно, далее необходимо указать, как минимум, директорию и имя файла с корневым сертификатом (CApath, CAfile). Если устройство работает в качестве клиента и аутентификация двусторонняя, то необходимы также имена файлов с сертификатом (cert) и закрытым ключом (key) данного устройства. Оба файла должны быть в формате PEM. Сертификат и ключ могут храниться в одном файле, в этом случае файл с ключом можно отдельно не указывать. Если директория не указана или не содержит требуемых файлов, туннель работать не будет.

   Проверка действительности сертификатов, присланных удалённой стороной, производится последовательно тремя способами:

   — По локальному списку сертификатов, если установлена опция verify = 3. Для этого сертификаты всех возможных партнёров, которые могут работать с данной стороной, должны быть помещены в директорию CApath и на них сгенерированы ссылки именами вида xxxxxxxxx.0, xxxxxxxxx.1 и т.п., где xxxxxxxxxx — хэш от поля Subject сертификата, а счётчик позволяет различать сертификаты с одинаковым хэшем. Сформировать их можно с помощью утилиты openssl или на централизованном сервере сертификатов.
   Если для данного сертификата *не* находится копии в этой директории, то сертификат считается недействительным, даже если он подписан действительным корневым сертификатом. Эту опцию целесообразно использовать на сервере, обслуживающем большое число априори известных клиентов; если клиентский сертификат скомпрометирован, то достаточно просто удалить его копию с сервера.

   — По локальному списку отозванных сертификатов (параметры CRLpath, CRLfile), если этот список задан в конфигурации. Если указанный список не существует, то все сертификаты считаются действительными.

   — По централизованному серверу OCSP (Online Certificate Status Protocol), если URL сервера указан в параметре OCSP. Если этот сервер не существует или недоступен, то все сертификаты считаются недействительными.

   Наконец, необходимо задать список алгоритмов защиты трафика, используемых данным устройством (параметр ciphers). Список может содержать несколько позиций, разделённых двоеточиями — хотя в корпоративных решениях, как правило, априори известно, что используется только 1–2 алгоритма. Обозначения алгоритмов см. в документе:

   http://www.openssl.org/docs/apps/ciphers.html

   По умолчанию, разрешены все алгоритмы, соответствующие выбранной версии протокола SSL.

Другие переменные, предусмотренные в файле stunnel.conf, могут быть установлены с помощью списка extra-options в формате опция = "значение" . Опции безопасности, соответствующие полю options в stunnel.conf, включаются в узле options . Рекомендуется включить следующий минимальный набор опций для блокировки известных уязвимостей и устаревших версий протокола: ALL; NO_SSLv2; NO_SSLv3 .

Для управления отладкой STunnel используется общий параметр .tunnel.stunnel.debug . Возможные значения — от 0 (минимальный уровень, только критические сообщения) до 7 (наиболее подробный уровень). Журнал STunnel и текущее содержимое файла stunnel.conf можно просмотреть в узле .tunnel.stunnel.show .

ВНИМАНИЕ Сертификаты имеют конечный срок действия. Во избежание потери связи с клиентами необходимо заблаговременно заменять сертификаты, срок действия которых истекает.

ВНИМАНИЕ Для работы SSL необходимо, чтобы на всех устройствах было корректно установлено системное время. Настоятельно рекомендуется синхронизировать все устройства от единого сервера NTP, доступного по открытому каналу или, как минимум, без использования сертификатов.

Пример. Ручное создание туннеля для управления устройством по HTTPS с двусторонней аутентификацией:

```
stunnel
: my_bi-auth_https
: : CApath = "/etc/uitcp/certs/"
: : CAfile = "root.pem"
: : key=/etc/certs/serverkey.pem
: : cert=/etc/certs/server.pem
: : options = "ALL;NO_SSLv2"
: : verify = 3
: : accept=443
: : connect=127.0.0.1:80
```

§5.6.3. Автоматически создаваемые туннели

Помимо туннелей, создаваемых пользователем, в узле .tunnel.stunnel могут автоматически создаваться служебные туннели, используемые теми или иными встроенными службами устройства NSG. В данной версии к ним относится туннель с именем ~HTTPS~, который автоматически создаётся и удаляется с помощью опции .services.http.https . (По существу, сервис HTTPS есть сочетание HTTP и STunnel.) Для него устанавливается фиксированный набор параметров по умолчанию, а также генерируется минимальный набор сертификатов и ключей, также управляемых опциями HTTPS.

ПРИМЕЧАНИЕ Сертификат устройства NSG, генерируемый в ходе данной процедуры, является самоподписанным. При первом подключении к устройству по HTTPS будет выдано предупреждение броузера. Для продолжения работы необходимо вручную добавить данный сертификат в исключения (в список доверенных сертификатов).

Настройка данных туннелей пользователем не предусмотрена. Если значения, устанавливаемые по умолчанию, не устраивают пользователя, то необходимо выключить службу HTTPS, а затем вручную создать туннель STunnel с обязательными параметрами accept = 443 и connect = 80 и желаемым набором других параметров.

§5.6.4. Уникальная идентификация клиентов SSL

В некоторых прикладных решениях имеется необходимость в том, чтобы прикладной сервер идентифицировал своего клиента, например, по заранее известному IP-адресу. Если же такое соединение пропускается через туннель STunnel, то уникальность клиента теряется: STunnel работает как прокси и все соединения с прикладным сервером исходят с адреса сервера STunnel (на рисунке на стр.27 — IP_e). Это принципиальная особенность данного типа туннелей, поскольку они работают на 4 уровне протокольной иерархии и могут произвольно манипулировать нижележащими уровнями. Соединения от сервера STunnel к прикладному серверу различаются только номером порта TCP источника, который выбирается системой случайным образом и никакой полезной информации о том, какой именно клиент работает по данному соединению, не содержит.

Для сохранения уникальности клиентов возможны следующие варианты:

1. Использовать уникальные идентификаторы прикладного уровня — такие, как POS ID в банковских системах, имя/пароль клиента и т.п.
2. Настроить для каждого клиента STunnel уникальный номер порта TCP назначения на сервере STunnel (connect = "IP_d:порт_d"), а на сервере STunnel — соответствующий входящий порт и уникальный порт назначения на прикладном сервере (accept = "порт_d", connect = "IP_f:порт_f"). Таким образом, каждый прикладной клиент будет попадать в строго определённый порт на сервере. Отметим, что настройка портов назначения на прикладных клиентах и входящих портов на клиентах STunnel (в точке IP_b) при этом может быть единообразной для всех клиентских площадок.

   Недостаток такого варианта — его трудоёмкость и, соответственно, вероятность человеческих ошибок.
3. Теоретически, для сервера STunnel предусмотрена специальная опция transparent = yes , позволяющая передать в соединение с прикладным сервером, в качестве IP-адреса источника, публичный IP-адрес клиента STunnel (IP_c на стр.27). Таким образом, прикладной сервер видел бы каждого клиента по уникальному IP-адресу. Однако данная функциональность требует включения специальных опций при генерации ядра Linux и не является стандартной, поскольку противоречит фундаментальным принципам построения сетей IP (по этим же причинам она отсутствует в реализациях STunnel для Windows). В данной версии NSG Linux она не реализована.
4. Использовать более развитые варианты VPN на основе SSL, предусматривающие передачу пакетов IP третьего уровня поверх туннеля 4 уровня. В ним относятся, в частности, OpenVPN или ***ui*TCP** (фирменная технология NSG, подробно см. Часть 6) в режиме raw IP или виртуальных интерфейсов.

§5.7. Туннели OpenVPN

§5.7.1. Общие сведения об OpenVPN

OpenVPN — развитая кросс-платформенная система туннелирования 4 уровня (в терминах модели OSI). Как и STunnel, для аутентификации сторон и защиты трафика она опирается на TLS/SSL и сертификаты X.509. (Возможна также аутентификация на основе симметричного ключа, но это менее безопасно, особенно при неизвестном заранее IP-адресе одной из сторон.) В отличие от STunnel, работающего как TCP-прокси, OpenVPN создаёт полноценные виртуальные интерфейсы, которые могут использоваться либо как аналог порта Ethernet (2 уровня), либо как IP-интерфейс 3 уровня. OpenVPN может работать в режиме "точка-точка" между двумя устройствами или в режиме "точка-многоточка", где один сервер обслуживает многих клиентов. В качестве протокола нижележащего уровня для пакетов туннеля может использоваться UDP либо TCP.

Реализация OpenVPN в NSG Linux основана на стандартном демоне openvpn и поддерживает все его возможности. Подробную документацию собственно по OpenVPN см. в *man pages*, а также на сайте разработчика http://www.openvpn.net/ .

Большинство параметров OpenVPN, представленных в интерфейсе, совпадают с одноимёнными ключами командной строки или файла конфигурации openvpn. Любые другие параметры могут быть добавлены пользователем в поле extra; эти параметры добавляются к временному файлу конфигурации, создаваемому для запуске openvpn, как есть. Поле может содержать несколько опций, разделённых \n; в этом случае при редактировании в Web-интерфейсе оно автоматически преобразуется в двумерное текстовое окно.

ПРИМЕЧАНИЕ. В числе дополнительных параметров не допускается использование dev и writepid, чтобы избежать конфликтов. Эти параметры устанавливаются средствами конфигурации NSG в любом случае, исходя из имени туннеля (ovpnNUM).
Во избежание конфликтов, рекомендуется настраивать все параметры, предусмотренные в дереве конфигурации явным образом, именно в этом дереве, а в параметре extra использовать только параметры, не присутствующие явно.

В частности, если у пользователя имеется готовый файл конфигурации, отлаженный в аналогичном решении на базе обычного ПК под управлением ОС Linux, то для замены ПК на устройство NSG достаточно указать его в поле config-file. В этом случае, однако, необходимо удалить или закомментировать в нём параметры dev и writepid. Эти параметры устанавливаются средствами конфигурации NSG в любом случае, исходя из имени туннеля (ovpnNUM).

§5.7.2. Режимы и протоколы работы туннеля

Настройка туннелей производится в узле .tunnel.openvpn . Каждому одиночному туннелю или серверу OpenVPN соответствует один подузел в нём. Туннели автоматически получают имена вида ovpn*N* ; для создания туннеля достаточно указать его номер.

Шлюз OpenVPN может работать в одном из двух режимов (параметр mode). Одноранговый режим позволяет установить соединение "точка-точка" между двумя устройствами, или подключиться в качестве клиента в системе "точка-многоточка". Режим сервера допускает одновременное подключение многих удалённых клиентов к данному устройству. Каждый режим имеет свои особенности в выборе нижележащего протокола.

Протокол UDP является предпочтительным для работы OpenVPN и предполагается по умолчанию. Он равно приемлем как для одноранговго режима, так и для сервера. В этом случае два шлюза обмениваются датаграммами через сеть общего пользования; контроль доставки возлагается на механизм собственно OpenVPN и, если внутри него передаётся трафик TCP, то и на TCP тоже.

Протокол TCP предполагает установление соединений через сеть общего пользования и контроль доставки пакетов по этому соединению. Для туннеля в режиме "точка-точка" необходимо явно указать, кто из двух устройств является TCP-клиентом, а кто — TCP-сервером. Различие между ними относится к процедуре установления туннеля: клиент инициирует соединение с сервером по указанному IP-адресу и порту, а сервер пассивно открывает порт на прослушивание и ждёт входящих соединений от клиента. Кроме этого, в настройках сервера указываются параметры Диффи-Хеллмана для защиты трафика в данном туннеле, а клиент обязан следовать предложениям сервера.

Для системы "точка-многоточка", работающей по TCP, клиенты должны быть настроены в режиме tcp-client, а для сервера установленный протокол tcp подразумевает пассивный режим работы.

ПРИМЕЧАНИЕ Переключение основного режима p2p/server приводит к установке поля протокола в значение по умолчанию (UDP).

§5.7.3. Описание концов туннеля в сети общего пользования

Для установления туннеля необходимо описать локальное и удалённое устройства. Для локального устройства можно указать явным образом IP-адрес и номер порта TCP или UDP (в зависимости от выбранного протокола). В этом случае OpenVPN будет работать только на публичном интерфейсе, которому принадлежит указанный адрес. По существу, на этом адресе OpenVPN принимает входящие пакеты, и этот адрес ставит в качестве адреса источника в исходящих пакетах.

Если адрес не указан, то OpenVPN работает на всех публичных интерфейсах, т.е. принимает входящие пакеты по всем IP-адресам, присвоенным данному устройству, а в исходящих пакетах адрес источника указывается динамически в зависимости от того, через какой интерфейс отправляется пакет согласно действующей таблице маршрутизации.

Описание удалённой стороны туннеля предусмотрено только в одноранговом режиме. Однако здесь оно более сложно, поскольку для клиента допускается работа с несколькими серверами (для резервирования), а для сервера — поддержка клиента с непостоянным IP-адресом. Настройка производится в отдельном подузле remote .

Во-первых, в данном узле составляется список IP-адресов удалённой стороны. Список может содержать более одного адреса.

Во-вторых, данный узел содержит три параметра, регламентирующих работу с этими и иными адресами:

rport — Номер порта TCP или UDP, соответственно, используемого OpenVPN на удалённой системе.

remote-random — Порядок выбора очередного сервера из списка: по кругу или случайным образом. Во втором случае нагрузка от многих клиентов распределяется между несколькими серверами более равномерно.

float — Разрешение динамической смены адреса. Если она запрещена, то удалённая сторона обязана всегда работать только со статическими адресами, указанными в списке явным образом. Список в этом случае обязательно должен содержать хотя бы один адрес.
Если разрешены "плавающие" адреса, то удалённая сторона обязана работать с одного из адресов, указанных в списке, в момент установления туннеля. После этого она может перейти на иной адрес, и если приходящие пакеты успешно аутентифицируются, то туннель продолжит работу. В частном случае, если список адресов не задан, то клиенту разрешается работать с любых адресов. При этом он может как устанавливать новые соединения, так и динамически менять свой адрес в процессе работы.

В случае сервера для топологии "точка-многоточка" предполагается, что адреса клиентов априори неизвестны и могут быть любыми, равно как и номера портов на них. По этой причине описание удалённой стороны не требуется. При этом для сервера можно ограничить число клиентов, которые могут подключаться к нему одновременно. Кроме того, для сервера всегда предполагается режим безопасности TLS; дополнительно можно разрешить или запретить одновременное подключение нескольких клиентов с одним и тем же именем в сертификате (подробнее см. следующий параграф).

§5.7.4. Защита туннеля

Для туннеля OpenVPN, по постановке задачи, предполагается взаимная аутентификация сторон, аутентификация и защита всего полезного трафика в них. Для этой цели может использоваться широкий выбор алгоритмов аутентификации и алгоритмов защиты, включая отсутствие таковых.

Первоначальная аутентификация сторон для одноранговых соединений устанавливается параметром security . В частности, она может не производиться, или производиться простейшем образом, с помощью симметричного общего секретного ключа. В качестве дополнительно параметра необходимо указать файл этого ключа.

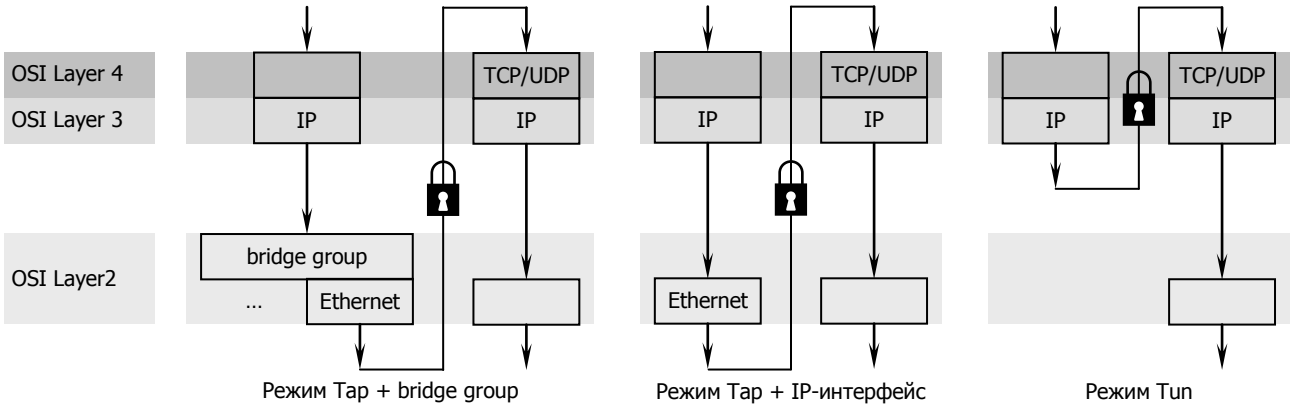
Если аутентификация сторон производится по полной программе, т.е. на основе сертификатов X.509 (или, строго говоря, на основе асимметричных ключей RSA, подтверждённых этими сертификатами), то на одной из сторон необходимо установить режим безопасности tls-client, а на другой — tls-server . На обеих сторонах необходимо указать файлы с корневым сертификатом, собственным сертификатом и приватным (закрытым) ключом данного устройства. Все сертификаты и ключи должны быть записаны в формате PEM по существу; расширение имени файла никакой роли не играет.

На сервере, дополнительно к этому, указывается файл с настройками Диффи-Хеллмана.

Если устройство работает в режиме сервера в топологии "точка-многоточка", то для него однозначно устанавливается режим аутентификации tls-server . Дополнительно к перечисленным выше параметрам безопасности, сервер имеет ещё один: разрешение устанавливать туннели одновременно с несколькими клиентами, имеющими одинаковые сертификаты (строго говоря, одинаковые поля CommonName). Если это запрещено, то при попытке подключения второго такого клиента сеанс работы с предыдущим клиентом будет разорван и установлен туннель (если удастся) ко второму клиенту.

§5.7.5. Настройка полезной нагрузки в туннеле

Для передачи полезного трафика туннель OpenVPN может эмулировать один из двух типов устройств (dev-type): порт Ethernet (tap) либо виртуальный IP-интерфейс (tun). В соответствии с этим, в конфигурации туннеля появляются все параметры, специфичные для устройств 2 либо 3 уровня: IP- и MAC-адреса, параметры QoS и т.п. В частности, туннель типа tap может быть включен в состав программного моста (*bridge group*), агрегированного потока (*bond group*), или же использоваться для передачи IP-трафика, инкапсулируя его в Ethernet. Туннель типа tun используется для передачи IP-пакетов без дополнительной инкапсуляции.



На обеих сторонах туннеля данный параметр должен быть установлен одинаково. Для него предусмотрено также третье значение — none ; оно означает, что туннель не используется для передачи какого-либо трафика, и может использоваться для временного прекращения работы по нему или же на этапе отладки туннеля для проверки процедур его установления.

Если устройство работает как сервер в топологии "точка-многоточка", то при установленном типе tap оно служит программным коммутатором, объединяющим всех клиентов и локальный Ethernet-подобный интерфейс на втором уровне. При установленном типе tun сервер автоматически выделяет подсеть с маской /30 на каждый из туннелей и создаёт маршруты через эти подсети в сети, описанные IP-префиксами клиентов. Диапазон адресов этих внутренних сетей, а также иные дисциплины назначения IP-адресов в туннелях, можно при необходимости регулировать дополнительными параметрами OpenVPN (--topology, --ifconfig-pool и др.) в поле extra .

Трафик туннеля может динамически сжиматься с помощью алгоритма LZO. Сжатие может быть принудительно включено, всегда выключено, или же включаться адаптивно в зависимости от его эффективности. В последнем случае шлюз OpenVPN периодически анализирует передаваемый им трафик; если это оказываются уже сжатые данные и эффективность последующего сжатия невысока (или даже может оказаться отрицательной), то сжатие отключается.

В конфигурациях "клиент-сервер" параметры полезной нагрузки могут автоматически передаваться от сервера клиенту после установления туннеля, с использованием механизма *push*. Данная функция относится к расширенным возможностям OpenVPN и может быть настроена в параметре extra . Её возможности, однако, ограничены параметрами полезной нагрузки, поскольку она сама работает по защищённому туннелю уже *после* его создания; очевидно, что с её помощью нельзя обмениваться параметрами, относящимися к созданию самого туннеля.

§5.7.6. Другие параметры туннеля

Среди прочих параметров туннеля OpenVPN следует выделить ping — посылку специальных контрольных пакетов в случае, если полезный трафик отсутствует в течение указанного времени. Данные пакеты позволяют решить две задачи:

— Если на пути к удалённой стороне находится брандмауэр, следящий за статусом соединений TCP и потоков UDP, соответственно, то *ping* позволяет поддерживать на нём данное соединение/поток в качестве активного.

— Удостовериться в доступности и работоспособности удалённой стороны. Если при этом установлен ненулевой параметр ping-exit, то при отсутствии пакетов от удалённой стороны в течение этого времени туннель будет разорван.

Если данный параметр не указан, *ping* автоматически не посылается. Более тонкая настройка механизма ping с помощью ряда дополнительных опций возможна в поле extra .

Название *ping* в данном случае является условным; оно означает специальный тип пакета OpenVPN и не имеет ничего общего с ICMP Echo. Ответы на данные пакеты не посылаются, поэтому, чтобы обеспечить их посылку в обоих направлениях, необходимо включить их на обеих сторонах туннеля.

Например, при следующих настройках на обеих сторонах:

```
ping = 10
ping-exit = 60
```

пакеты *ping* будут посылаться с той и с другой стороны после 10 секунд отсутствия входящего трафика от партнёра. Если пакеты *ping* от партнёра не поступают в течение 60 сек, т.е. 6 пакетов подряд, то туннель разрывается.

Другой механизм разрыва туннелей по неактивности основан исключительно на статистике полезного трафика, без посылки контрольных пакетов:

inactive Тайм-аут неактивности, в секундах. Если в течение данного времени отсутствует полезный трафик (входящий и исходящий одновременно), то туннель разрывается.

inactive-bytes Пороговый объём полезного трафика для разрыва туннеля. Используется в качестве дополнительного параметра: туннель считается активным только в случае, если за время inactive передано и принято (в совокупности) не менее указанного числа байт. При помощи данного параметра можно исключить, например, ситуации, когда приложения на одной и другой стороне в периоды неактивности регулярно обмениваются своими собственными пакетами *keepalive* (которые для туннеля формально являются полезной нагрузкой). Если значение данного параметра равно нулю, то разрыв по неактивности производится обычным образом, строго по времени: после каждого принятого или полученного байта таймер запускается заново.

Например, предположим, что приложения обмениваются пакетами *keepalive* 1 раз в минуту, туда и обратно, размер пакета 64 байта. Тогда при следующих установках:

```
inactive = 600
inactive-bytes = 1281
```

туннель будет разорван через 10 минут при условии, что за всё это время ничего, кроме прикладных *keepalive*, по нему не передавалось.

Критерии разрыва туннеля по потере *ping* и по отсутствию полезного трафика могут использоваться совместно.

ПРИМЕЧАНИЕ При подсчёте полезного трафика и детектировании периодов неактивности игнорируются пакеты *ping*, управляющие пакеты TLS и другие служебные пакеты OpenVPN.

Параметр verb управляет уровнем детализации отладочных сообщений.

Параметр description позволяет задать текстовое описание туннеля для удобства администрирования и непосредственно на его работу не влияет.

Просмотреть текущий файл конфигурации OpenVPN, построенный динамически из введённых выше параметров, можно командой actual-config .

§5.8. Технология бесперебойных соединений *ui*TCP

Фирменная разработка NSG ***ui*TCP** предназначена для построения бесперебойных TCP-соединений между узлами сети в критически ответственных приложениях, например, при подключении банкоматов. С точки зрения протокольной архитектуры, она представляет собой VPN 4 уровня, аналогичную OpenVPN, KerioNet или, в минимальной реализации STunnel. Для защиты трафика, как и в других аналогичных решениях, используется SSL и механизм ключей и сертификатов X.509.

Отличительными особенностями ***ui*TCP** являются:

— Механизмы прозрачного переключения между многими разнотипными каналами связи и поставщиками сетевых услуг.

— Гарантированная доставка данных.

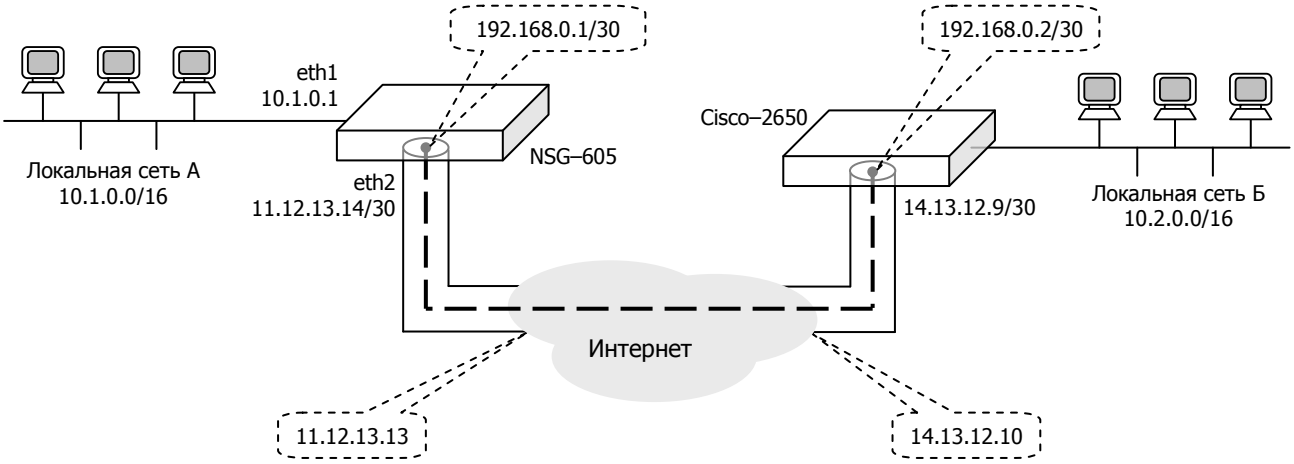
— Поддержание непрерывных TCP-сессий и других сеансов работы прикладного ПО при многократных переключениях между каналами связи.

Ввиду того, что данная система не является общеизвестной и требует детального описания, полное изложение её возможностей и настроек вынесено в отдельную Часть 6 данного Руководства.

Приложение 5–А. Примеры настройки туннелей и VPN

§5–А.1. Настройка туннеля GRE между NSG и Cisco

Схема стенда показана на рисунке. Стенд состоит из двух пограничных маршрутизаторов, соединенных через сеть общего пользования. Для наглядности на одной стороне используется устройство NSG–605, на другой — Cisco–2650.



Через маршрутизаторы связаны две приватные сети 10.1.0.0/16 и 10.2.0.0/16. Трафик этих сетей передается в туннеле между интерфейсами пограничных маршрутизаторов NSG–605 (11.12.13.14) и Cisco–2650 (14.13.12.9). При этом весь IP-пакет приватной сети, включая заголовок, передается как данные в новом пакете между двумя маршрутизаторами. На противоположной стороне пакеты корпоративной сети извлекаются из пакетов туннеля и передаются в приватную сеть. Для простоты приведены минимальные настройки, необходимые для работы туннеля. (Параметры, не относящиеся непосредственно к туннелю, опущены.)

Настройка NSG–605

```
ip
: route
: : 1
: : : device          = "gre1"
: : : network         = "10.2.0.0/16"
: : 2
: : : gateway         = "11.12.13.13"
: : : network         = "14.13.12.9/32"
port
: eth1
: : ifAddress
: : : prefix          = "10.1.0.1/16"
tunnel
: gre
: : gre1
: : : description     = "tunnel NSG - CISCO"
: : : adm-state       = "up"
: : : destination     = "14.13.12.9"
: : : encapsulation   = "ip"
: : : source          = "11.12.13.14"
: : : ifAddress
: : : : prefix        = "192.168.0.1/30"
```

Настройка Cisco–2650

```
!
interface FastEthernet0/0
    ip address 10.2.0.1 255.255.0.0
!
interface tunnel 0
    description "tunnel CISCO - NSG"
    tunnel mode gre ip
    tunnel destination 11.12.13.14
    tunnel source 14.13.12.9
    ip address 192.168.0.2/30
!
ip route 10.1.0.0/16  192.168.0.1
ip route 11.12.13.14  255.255.255.255  14.13.12.10
!
```

§5–A.2. Подключение устройства NSG к серверам PPPoE

Настройка устройства NSG в качестве клиента PPPoE. IP-адрес назначается динамически. Устройство обеспечивает доступ в Интернет для локальной сети, расположенной за ним. Курсивом показаны существенные настройки, установленные по умолчанию.

```
ip
: nat
: : POSTROUITNG
: : : 1
: : : : out-interface      = "client_demo"
: : : : target             = "MASQUERADE"
tunnel
: pppoe
: : client_demo
: : : adm-state            = "up"
: : : iface                = "eth4"
: : : mode                 = "client"
: : : name                 = "имя сервера"
: : : ppp
: : : : main
: : : : : default-route    = true
: : : : : ipcp
: : : : : : accept-address = true
: : : : : : accept-dns     = true
: : : : : sent-username    = basile
: : : : : sent-password    = pOuPKiNe
```

Для быстрой настройки можно использовать команду add-nat .

Настройка устройства Cisco в качестве сервера PPPoE:

```
!
aaa new-model
aaa authentication ppp default local
vpdn enable
!
vpdn-group 1
    accept-dialin
    protocol pppoe
    virtual-template 2
    ip mtu adjust
    pppoe limit per-mac 20
    local name CISCO
!
no ip cef
!
username basile  password 0 P0uPkiNe
!
interface FastEthernet0/0
    pppoe enable
!
interface Virtual-Template 2
    ip address 16.0.0.1 255.0.0.0
    peer default ip address pool TEST
    ppp authenticate chap
!
ip local pool TEST 16.0.0.2 16.0.0.20
```

Настройка устройства NSG с ПО NSG Linux 1.0 в качестве сервера PPPoE:

```
!
nsg
    virtual-template 1
        ppp authentication chap local
        peer ip address 19.0.0.2
        ip address 19.0.0.1
        exit
    vpdn-group 1
        pppoe limit per-mac 20
        protocol pppoe
        virtual-template 1
        local name NSGLinux
    port eth0
        vpdn-group 1
        exit
    users user-name basile open pOuPKiNe
    exit
```

Настройка Linux Red Hat 9 (пакет Roaring Penguin PPPoE Version 3.5) в качестве сервера PPPoE:

```
pppoe-server -I eth0 -C Linux -L 18.0.0.1 -R 18.0.0.2
```

Содержимое обязательного файла /etc/ppp/pppoe-server-options

```
require-chap
lcp-echo-interval 10
lcp-echo-failure 2
```

Содержимое файла /etc/ppp/chap-secrets:

```
basile * pOuPKiNew
```

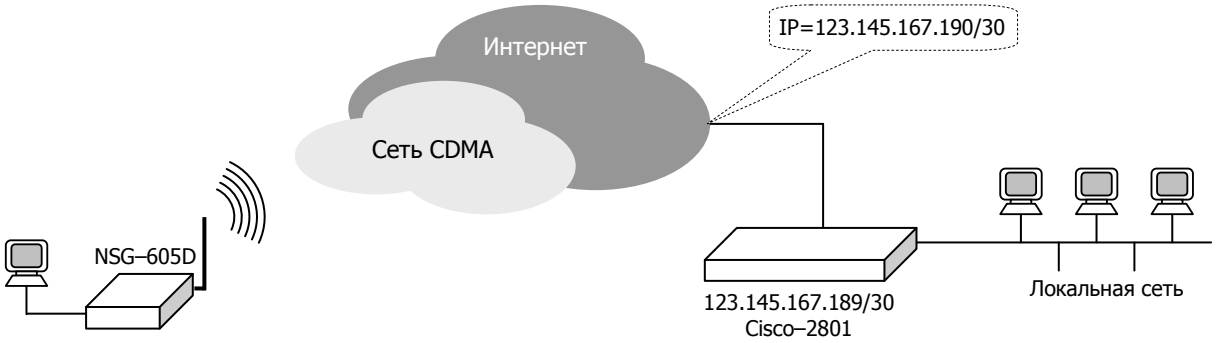
§5–A.3. Подключение клиентов PPPoE к устройству NSG

Устройство под управлением NSG Linux 2.0 используется в качестве сервера PPPoE. Аутентификация выполняется локально по протоколу PAP. В качестве дополнительной опции клиентам передаются адреса двух серверов DNS. Курсивом показаны существенные настройки, установленные по умолчанию.

```
system
: ppp-secrets
: : pap
: : 1
: : : client                    = "nif-nif"
: : : secret                    = "straw"
: : 2
: : : client                    = "nuf-nuf"
: : : secret                    = "wood"
: : 3
: : : client                    = "naf-naf"
: : : secret                    = "brick"
.....................................................................
tunnel
: pppoe
: : server_demo
: : : adm-state                 = "up"
: : : iface                     = "eth1"
: : : limit                     = 100
: : : mode                      = "server"
: : : name                      = "nsg1800"
: : : ppp
: : : : main
: : : : : authentication        = "pap"
: : : : : authentication-scheme = "local"
: : : : : idle                  = 300
: : : : : ip-address            = "172.16.0.1"
: : : : : ipcp
: : : : : : accept-address      = false
: : : : : : accept-dns          = false
: : : : : : accept-peer-address = false
: : : : : : dns1                = "123.45.67.89"
: : : : : : dns2                = "123.45.67.90"
: : : : : peer-ip-address       = "172.16.0.2"
```

§5–A.4. Подключение устройства NSG к серверу PPTP

Пример настройки соединения PPTP через сотовую сеть CDMA (SkyLink). Используется устройство NSG–605D. На центральном узле корпоративной VPN в качестве сервера PPTP используется устройство Cisco. Туннель защищён с помощью MPPE (в результате согласования будет выбрано 128 бит) и MS–CHAP v2. Курсивом показаны существенные настройки, установленные по умолчанию.



Настройка NSG–605D

```
ip
: route
: : 1
: : device                    = "cdma"
: : network                   = "123.145.167.189/32"
port
: cdma
: : adm-state                 = "up"
: : ppp
: : : main
: : : : chat
: : : : : timeout             = 30
: : : : debug-level           = 1
: : : : default-route         = false
: : : : ipcp
: : : : : accept-address      = true
: : : : : accept-peer-address = true
: : : : lcp-echo-failure      = 3
: : : : lcp-echo-interval     = 10
: : : : sent-password         = "internet"
: : : : sent-username         = "mobile"
tunnel
: pptp
: : client_demo
: : : adm-state               = "up"
: : : destination             = "123.145.167.189"
: : : mode                    = "client"
: : : ppp
: : : : main
: : : : : default-route       = true
: : : : : encrypt-mppe        = "auto"
: : : : : encrypt-mppe-mode   = "stateless"
: : : : : ipcp
: : : : : : accept-address    = true
: : : : : : accept-peer-address = true
: : : : : lcp-echo-failure    = 0
: : : : : lcp-echo-interval   = 0
: : : : : sent-password       = "P0uPkiNe"
: : : : : sent-username       = "basile"
```

Настройка PPTP для Cisco 2801:

```
!
aaa new-model
aaa authentication ppp default local
no ip cef
vpdn enable
vpdn-group 1
    accept-dialin
        protocol pptp
        virtual-template 1
    ip mtu adjust
!
username basile  password 0 P0uPkiNe
!
interface FastEthernet0/0
    ip address 123.145.167.189 255.255.255.252
!
interface Loopback0
    ip address 172.16.0.1 255.255.255.0
!
interface Virtual-Template1
    ip unnumbered Loopback0
    ip virtual-reassembly
    peer default ip address pool APOOL
    keepalive 11 3
    ppp encrypt mppe auto
    ppp authentication ms-chap-v2
!
ip local pool APOOL 172.16.0.2 172.16.0.20
ip route 0.0.0.0 0.0.0.0 FastEthernet0/0 123.145.167.190
!
```

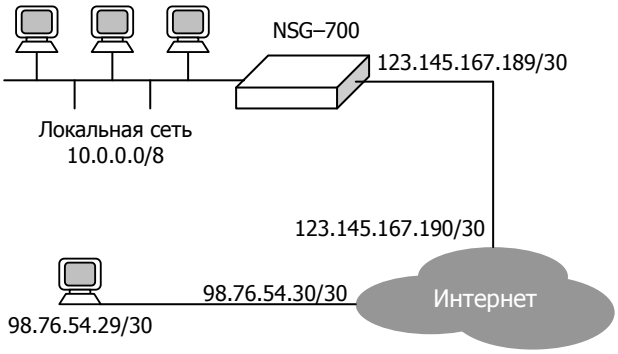
Здесь mobile и internet — имя и пароль для доступа к услуге CDMA, 123.145.167.189 — адрес удаленного сервера в Интернет, basile P0uPkine — имя и пароль для PPTP-соединения с этим сервером. Для простоты, пароли клиентов прописаны непосредственно в соответствующих узлах ppp . В случае некорректного отсоединения от сети CDMA (пропадание сигнала и т.п.) отказ будет детектирован через 30 сек (3 попытки по 10 сек). После этого PPP-интерфейс рестартует и попытается снова установить соединение с сетью. Если соединение будет восстановлено успешно и с прежним IP-адресом, а по туннелю в течение всего этого времени никакие данные не посылались, то туннель PPTP продолжит работу, так что переустановление физического соединения и PPP-соединения останется незамеченным для пользователей сети. Если же соединение не восстановлено, а PPTP-интерфейс попытается передать данные по туннелю, то управляющее соединение обнаружит, что интерфейс сети общего пользования находится в состоянии DOWN (или снова в UP, но с иным IP-адресом), и туннель будет разорван.

§5–A.5. Подключение клиента PPTP к устройству NSG

Пример настройки соединения PPTP для удалённого клиента, которому нужно предоставить доступ в корпоративную сеть. В качестве сервера используется устройство NSG–700/4AU, подключённое к поставщику услуг через порт Fast Ethernet. Туннель защищён с помощью MPPE (в результате согласования будет выбрано 128 бит) и MS–CHAP v2. Аутентификация производится по централизованному серверу RADIUS. Курсивом показаны существенные настройки, установленные по умолчанию.

Настройка NSG–700 в качестве сервера

```
ip
: route
: : 1
: : : gateway                  = "123.145.167.190"
: : : network                  = "0.0.0.0/0"
ethernet-switch
: phy0
: : vlan-groups
: : : 1                        = 101
: : : 2                        = 102
: : vlan-tagged                = true
: phy1
: : vlan-group                 = 101
: phy2
: : vlan-group                 = 102
port
: eth0
: : vlan
: : : eth0.101
: : : : ifAddress
: : : : : prefix               = "123.145.167.189/30"
: : : eth0.102
: : : : ifAddress
: : : : : prefix               = "10.0.0.1/8"
system
: aaa
: : ppp
: : : 1
: : : : type                   = "radius"
: : : : port                   = 1812
: : : : secret                 = "qwerty"
: : : : server                 = "10.0.0.123"
: : : : timeout                = 3
tunnel
: pptp
: : server_demo
: : : adm-state                = "up"
: : : listen                   = "123.145.167.189"
: : : limit                    = 10
: : : mode                     = "server"
: : : ppp
: : : : main
: : : : : authentication       = "ms-chap-v2"
: : : : : authentication-scheme = "system-aaa"
: : : : : encrypt-mppe         = "auto"
: : : : : encrypt-mppe-mode    = "stateless"
: : : : : idle                 = 300
: : : : : ip-address           = "172.16.0.1"
: : : : : ipcp
: : : : : : accept-address     = false
: : : : : : accept-dns         = false
: : : : : : accept-peer-address = false
: : : : : peer-ip-address      = "172.16.0.2"
```



В качестве клиента может использоваться, например, ПК под управлением ОС Windows, Linux, второе устройство NSG и т.п.

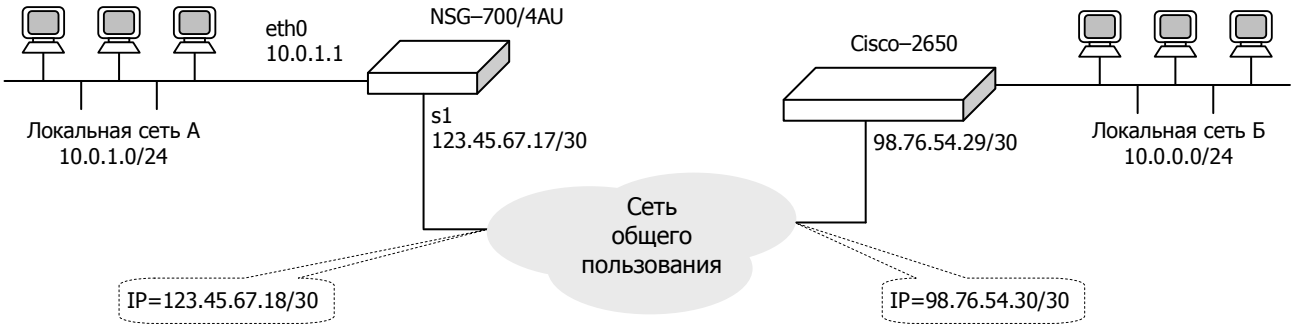
Настройка удалённого NSG–700 в качестве клиента:

```
ip
: route
: : 1
: : device                     = "client_demo"
: : network                    = "10.0.0.0/8"
: : 2
: : gateway                    = "98.76.54.30"
: : network                    = "0.0.0.0/0"
port
: eth0
: ifAddress
: : prefix  = "98.76.54.29/30"
tunnel
: pptp
: : client_demo
: : : adm-state                = "up"
: : : destination              = "123.145.167.189"
: : : mode                     = "client"
: : : ppp
: : : : main
: : : : : default-route        = true
: : : : : encrypt-mppe         = "auto"
: : : : : encrypt-mppe-mode = "stateless"
: : : : : ipcp
: : : : : : accept-address     = true
: : : : : : accept-peer-address = true
: : : : : lcp-echo-failure     = 3
: : : : : lcp-echo-interval    = 10
: : : : : sent-password        = "P0uPkiNe"
: : : : : sent-username        = "basile"
```

В результате согласования будет установлено MPPE 128 бит и режим stateless. Дополнительно в данном примере необходимо настроить сервер RADIUS.

§5–A.6. Настройка туннеля IPsec (IKE) между NSG и Cisco

Схема стенда показана на рисунке. Стенд состоит из двух пограничных маршрутизаторов, соединенных через сеть общего пользования. Интерфейсы соседних маршрутизаторов в этой сети имеют IP-адреса 123.45.67.89/30 и 98.76.54.30/30. Для наглядности на одной стороне используется устройство NSG–700 с дополнительным модулем UM–ET100, на другой — Cisco–2650. Курсивом показаны существенные настройки, установленные по умолчанию.



Через маршрутизаторы связаны две приватные сети 10.0.1.0/24 и 10.0.0.0/24. Трафик этих сетей передается в безопасном туннеле между интерфейсами пограничных маршрутизаторов NSG–700 (123.45.67.17) и Cisco–2650 (98.76.54.29). При этом весь пакет, включая заголовок, защищается по алгоритму 3DES (длина ключа 168 бит) и передается как данные в IP-пакете между двумя маршрутизаторами. Дополнительно передается аутентификационный заголовок SHA–1, обеспечивающий аутентичность и целостность данных. На противоположной стороне туннеля данные пакета расшифровываются и передаются в приватную сеть. Весь остальной трафик принимается и отсылается указанными интерфейсами без какой-либо обработки.

Настройка NSG–700

Общие настройки:

```
port
: eth0
: : ifAddress
: : : prefix              = "10.0.1.1/8"
: s1
: : type                  = "eth"
: : ifAddress
: : : prefix              = "123.45.67.17/30"
```

Включение IPsec, задание PSK — аналог *crypto isakmp key*. Порядок перечисления идентификаторов не имеет значения.

```
tunnel
: ipsec
: : enable                = true
: : secrets
: : : psk
: : : : 1
: : : : : indices
: : : : : : 1             = "98.76.54.29"
: : : : : : 2             = "123.45.67.17"
: : : : : secret          = "aa"
```

Создание туннеля и его общие настройки:

```
: : connections
: : : nsg2cisco
: : : : authby            = "secret"
: : : : auto              = "start"
: : : : dpddelay          = 30
: : : : dpdtimeout        = 120
: : : : ikelifetime       = 3600
```

Описание защищаемых сетей (аналог *access-list*):

```
: : : : leftsubnet        = "10.0.1.0/8"
: : : : rightsubnet       = "10.0.0.0/8"
```

Выбор алгоритмов защиты и аутентификации трафика (аналог *transform-set*):

```
: : : : esp
: : : : : 3des-sha1        = true
```

Другие параметры (аналог *crypto map* самого по себе и в меню интерфейса). rightnexthop — в данном случае, параметр формальный, де-факто не требуется.

```
: : : : left               = "123.45.67.17"
: : : : leftid             = "123.45.67.17"
: : : : leftnexthop        = "123.45.67.18"
: : : : right              = "98.76.54.29"
: : : : rightid            = "98.76.54.29"
: : : : rightnexthop       = "98.76.54.30"
```

Указание защищаемых интерфейсов. Собственный интерфейс требуется для маршрутизации, интерфейс удалённой стороны в данном случае — формальный параметр.

```
: : : : leftsourceip       = "10.0.1.1"
: : : : rightsourceip      = "10.0.0.1"
```

Настройка Cisco–2650

```
access-list 153 permit ip 10.0.0.0 0.0.0.255 10.0.1.0 0.0.0.255
crypto ipsec transform-set ts3 esp-3des esp-sha-hmac
!
crypto map M1 3 ipsec-isakmp
    set peer 123.45.67.17
    set transform-set ts3
    match address 153
!
crypto isakmp key aa address 123.45.67.17 98.76.54.29
!
```

Дополнительно требуется определить *policy* с указанием использования механизма PreShared Key (поскольку *default policy* использует RSA):

```
!
crypto isakmp policy 1
    encr 3des
    authentication pre-share
    group 2
    lifetime 3600
!
```

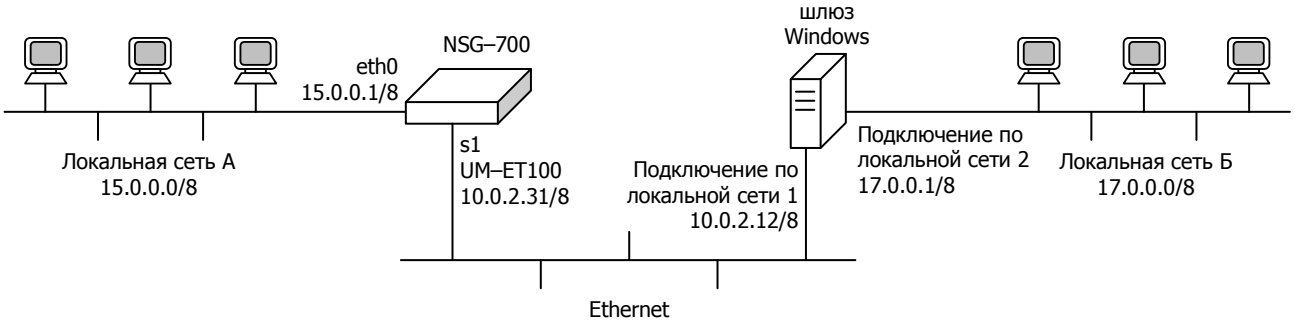
В заключение конфигурируется IP-интерфейс и маршрутизация:

```
!
interface FastEthernet0/0
    ip address 98.76.54.29 255.255.255.252
    crypto map M1
!
ip  route  10.0.1.0  255.0.0.0  123.45.67.17
ip  route  123.45.67.17  255.255.255.255  98.76.54.30
!
```

§5–A.7. Настройка туннеля IPsec (IKE) между NSG и Windows

Имеются две локальные сети, одна из которых подключена к устройству NSG–700, другая — к некоторому программному шлюзу, на котором установлена операционная система Windows 2000 или старше корпорации Майкрософт. Требуется организовать безопасный туннель IPsec между этими сетями. Для большей ясности задачи будем предполагать, что два шлюза соединены между собой просто сетью Ethernet. Курсивом показаны существенные настройки, установленные по умолчанию.

ВНИМАНИЕ Предполагается, что до начала настройки IPsec на обоих шлюзах настроена маршрутизация, так что хосты из одной сети успешно обмениваются IP-пакетами с хостами из другой сети, и наоборот.



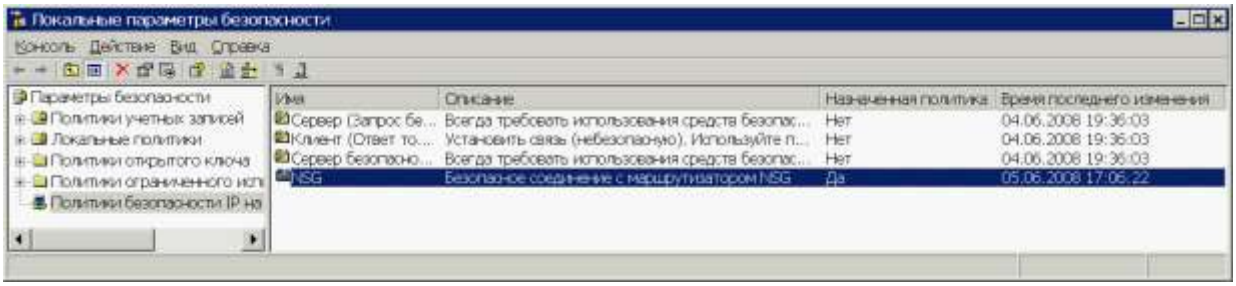
Конфигурация NSG–700:

```
ip
: route
: : 1
: : : gateway          = "10.0.2.12"
: : : network          = "17.0.0.0/8"
port
: eth0
: : ifAddress
: : : prefix           = "15.0.0.1/8"
: s1
: : type               = "eth"
: : ifAddress
: : : prefix           = "10.0.2.31/8"
tunnel
: ipsec
: : enable             = true
: : secrets
: : : psk
: : : : 1
: : : : : indices
: : : : : : 1          = "10.0.2.31"
: : : : : : 2          = "10.0.2.12"
: : : : : secret       = "12345678"
: : connections
: : : nsg2win
: : : : authby         = "secret"
: : : : auto           = "start"
: : : : esp
: : : : : 3des-sha1    = true
: : : : left           = "10.0.2.31"
: : : : leftnexthop    = "%direct "
: : : : leftsourceip   = "15.0.0.1"
: : : : leftsubnet     = "15.0.0.0/8"
: : : : pfs            = "no"
: : : : right          = "10.0.2.12"
: : : : rightnexthop   = "%direct"
: : : : rightsubnet    = "17.0.0.0/8"
```

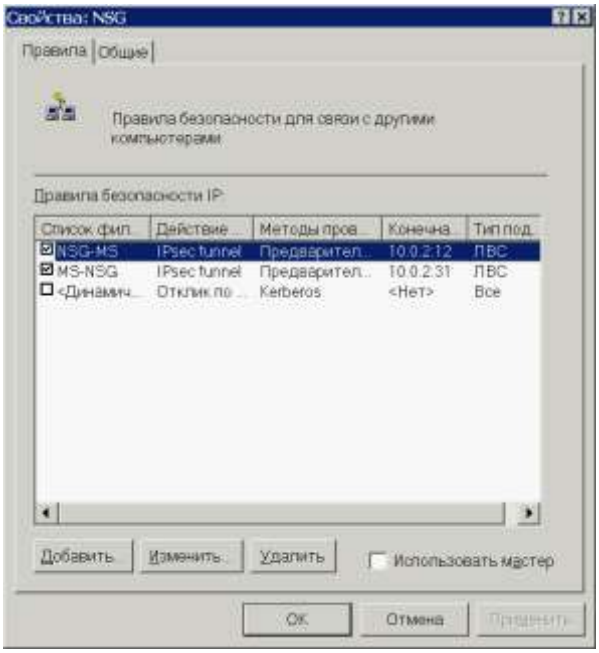
только для проверки маршрутизации; после настройки VPN удалить

Настройка Microsoft Windows:

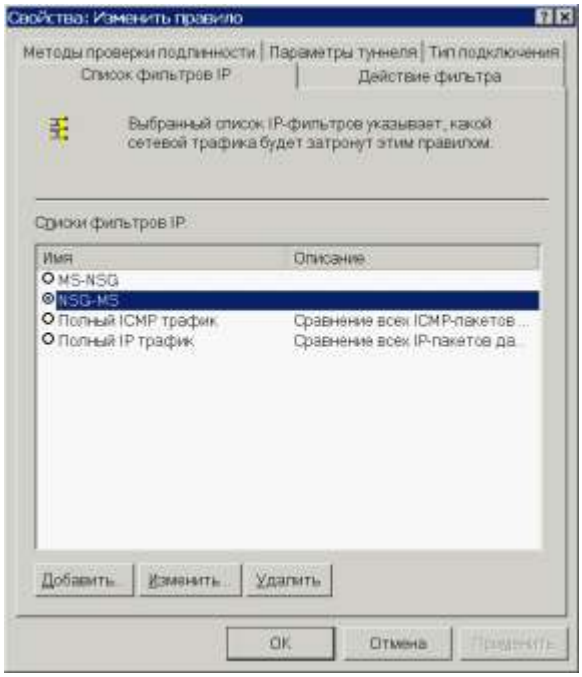
1. Запустить консоль локальной политики безопасности: "Пуск" — "Настройка" — "Панель управления" — "Администрирование" — "Локальная политика безопасности" (или "Пуск" — "Программы" — "Администрирование" — …). В левой части консоли выбрать "Политики безопасности IP…". В правой части консоли создать новую политику, например, под названием NSG.

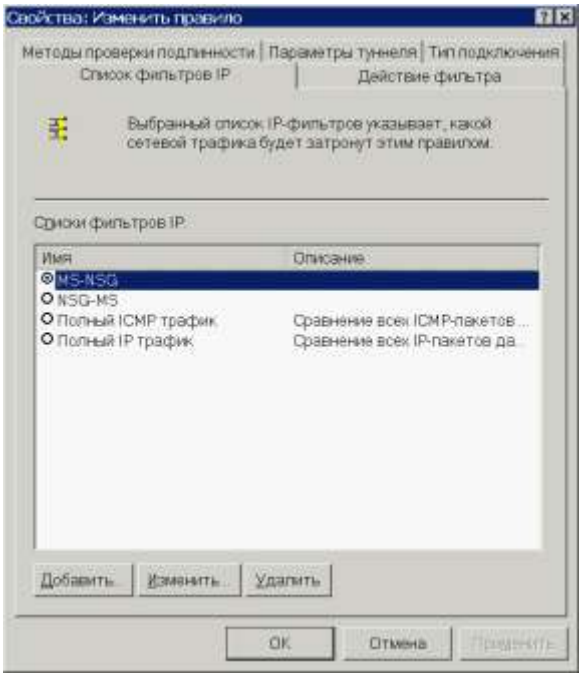


2. Открыть свойства созданной политики. Для облегчения работы рекомендуется создать политику с минимальными настройками, которые требует Мастер Создания Политик, а затем отключить на всех вкладках опцию "Использовать Мастер" и настроить нужные параметры самостоятельно. Необходимо создать и активировать (поставить отметку слева) два правила, описывающие, насколько можно понять новояз корпорации Майкрософт, передачу трафика между двумя шлюзами во встречных направлениях. Имя и описание правил существенной роли не играют; будем называть их NSG–MS и MS–NSG, соответственно. Процедура создания правил описана ниже. Остальные правила, существующие по умолчанию, следует удалить или отключить. Для создания и редактирования объектов в данном окне используются кнопки "Добавить" и "Изменить".

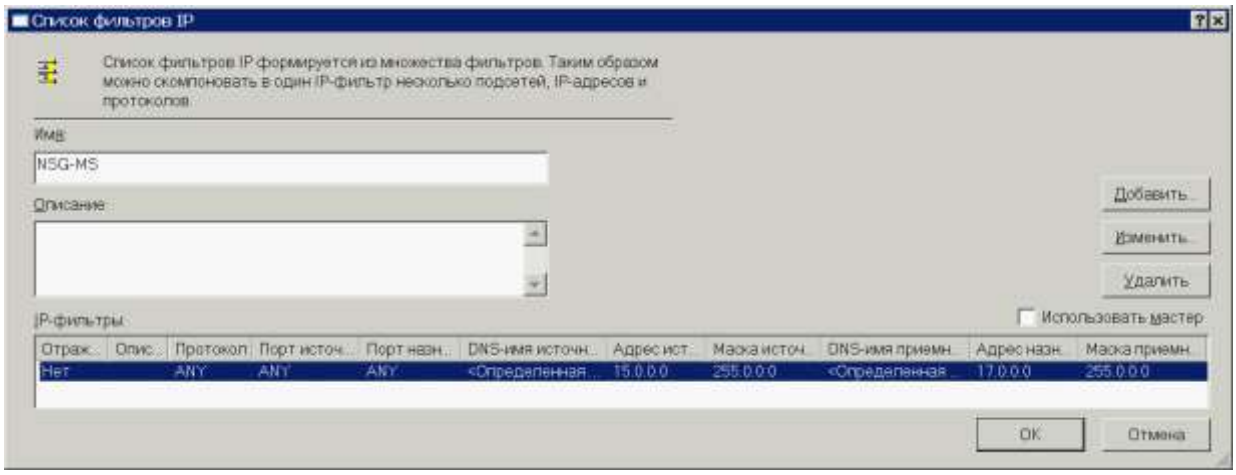


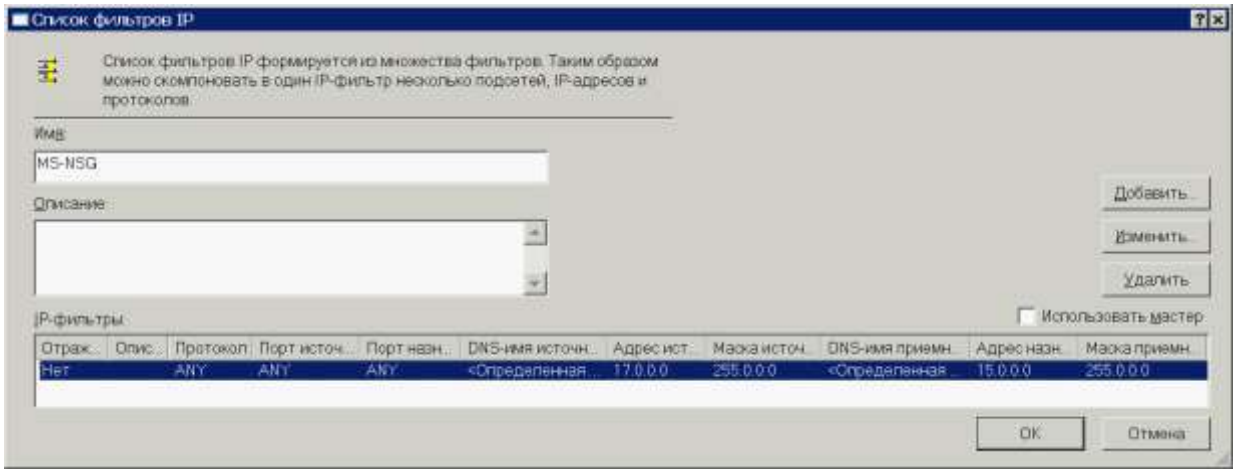
3. Для создания нового правила нажать кнопку "Добавить". Откроется окно с 5 вкладками. На вкладке "Список фильтров IP" нужно создать два фильтра и для каждого из правил выбрать соответствующий фильтр. Создаваемое таким образом правило получает имя от используемого в нём фильтра. На рисунках показан вид окна после создания фильтров. Фильтры, существующие по умолчанию, следует отключить или удалить.



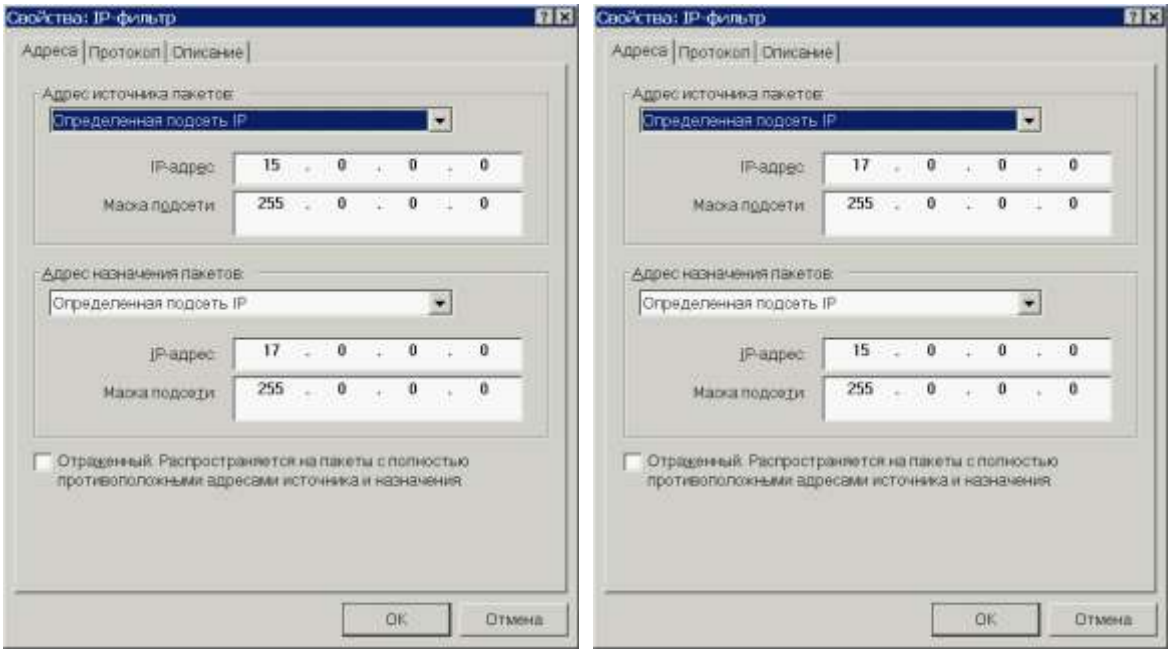


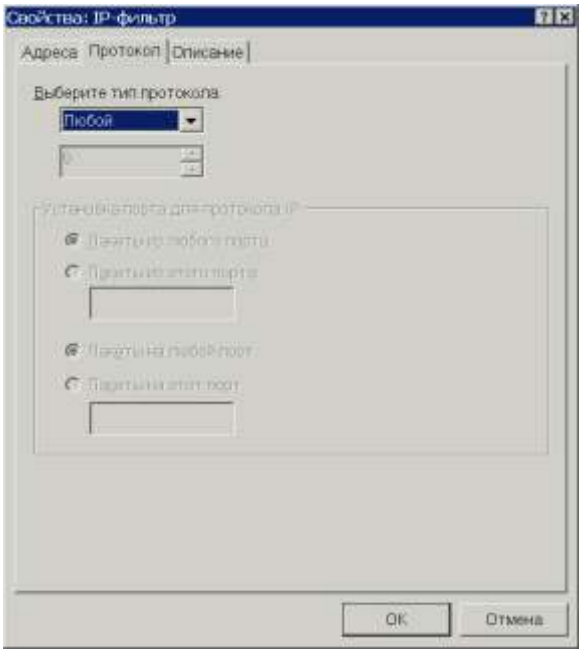
3.1. В окне редактирования свойств фильтра следует ввести имя, которое будет присвоено фильтру и правилу, и нажать кнопку "Добавить". На рисунке показан вид данного окна для того и для другого фильтров после добавления записей.



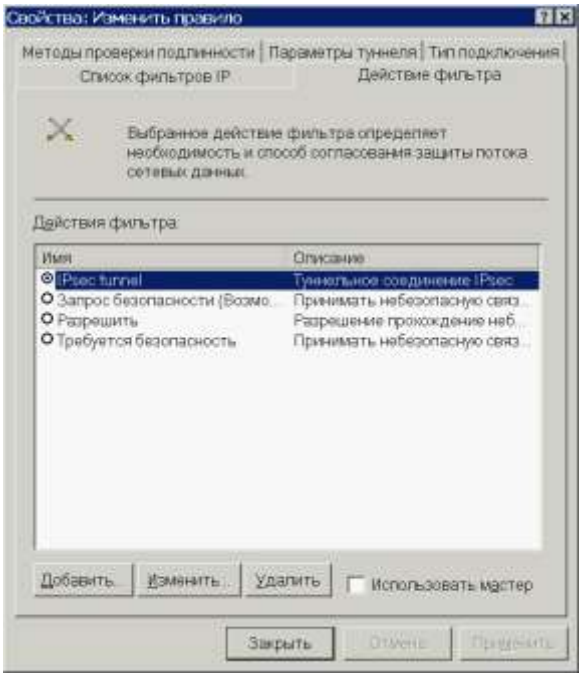


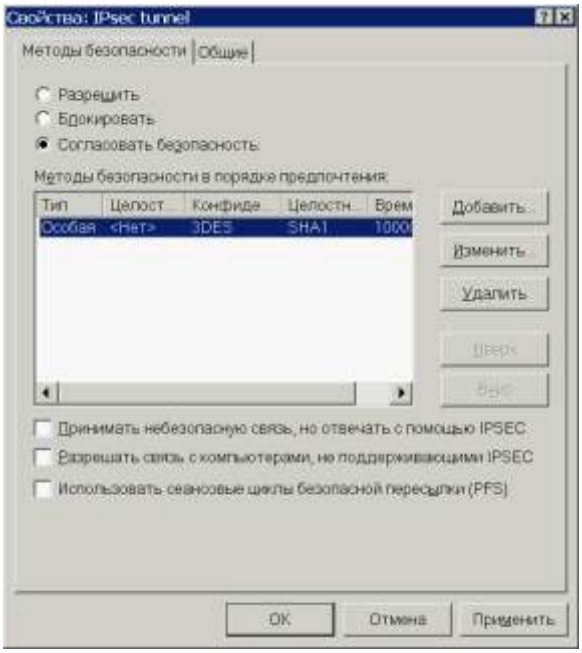
3.1.1. В окне добавления/изменения фильтра на вкладке "Адреса" выбрать для источника и назначения тип "Определённая подсеть IP" и указать адреса и маски обеих сетей (в противоположном порядке для двух фильтров). Опция "Отражённый" никакого влияния на работу системы не оказывает, её назначение неясно. При необходимости можно указать конкретные типы протоколов и номера портов TCP и UDP источника и назначения, чтобы направлять в защищённый туннель только специфические пакеты. По завершении настроек нажать кнопку "OK", убедиться, что окно "Список фильтров IP" приняло вид, изображённый выше, и закрыть это окно.

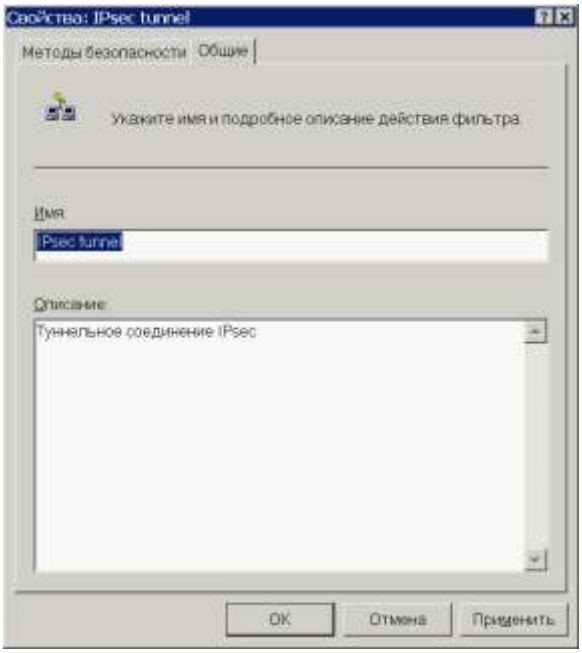




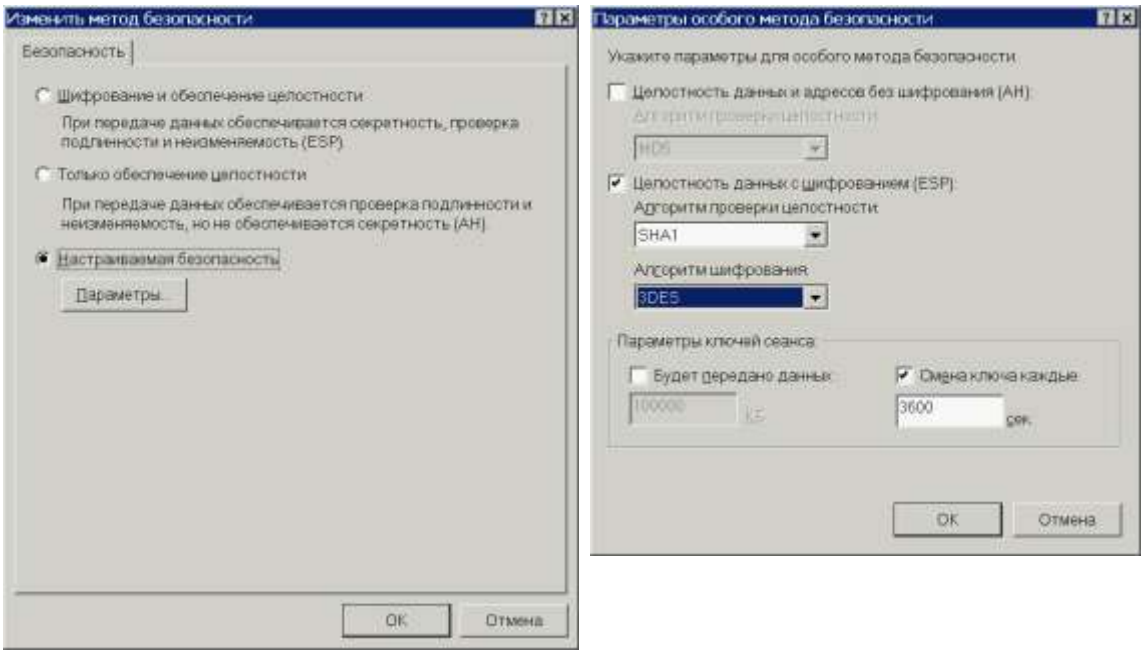
3.2. На вкладке "Действия фильтра" создать и выбрать действие "IPsec tunnel". В окне редактирования действия задаются его имя и описание (вкладка "Общие"). На вкладке "Методы безопасности" выбрать опцию "Согласовывать безопасность" и нажать кнопку "Добавить". На рисунке показан вид данного окна после добавления записей. Действие создаётся одинаковым для обоих фильтров.



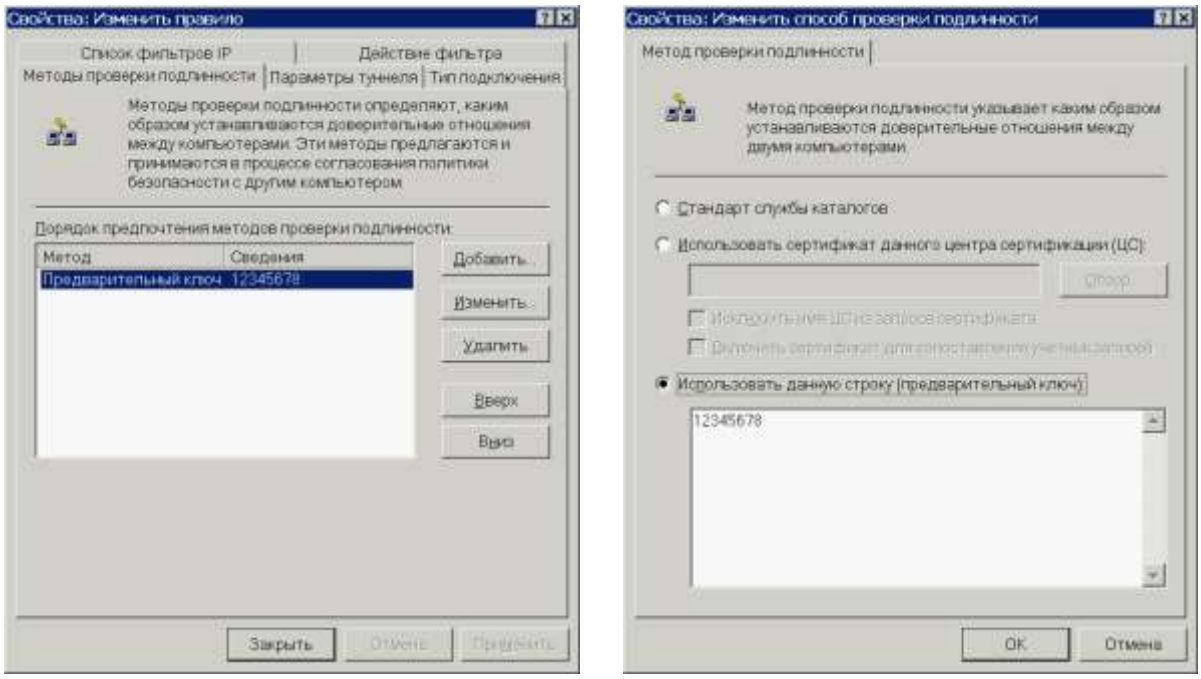




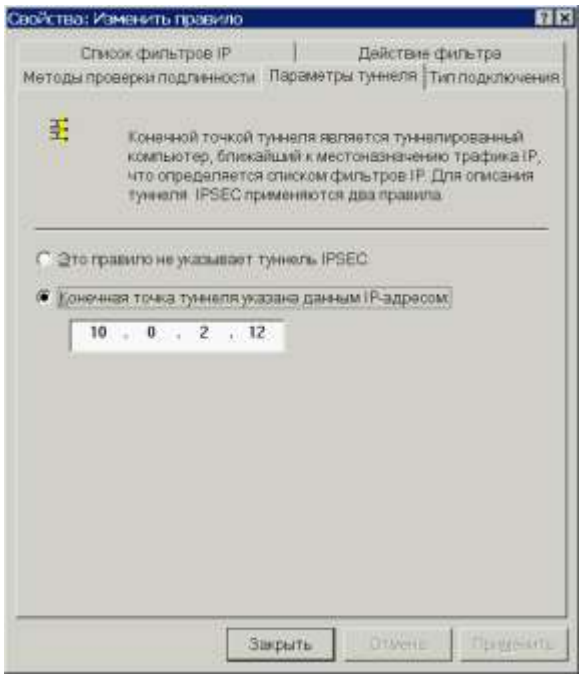
3.2.1. В окне "Изменить метод безопасности" выбрать опцию "Настраиваемая безопасность" и нажать кнопку "Параметры". В окне "Параметры особого метода безопасности" выбрать опцию "Целостность данных с шифрованием (ESP)", алгоритм проверки целостности SHA1, алгоритм шифрования 3DES. Рекомендуется включить опцию регулярной смены ключей. По завершении настроек нажать кнопку "OK", убедиться, что окно "Свойства: IPsec tunnel" приняло вид, изображённый выше, и закрыть это окно.

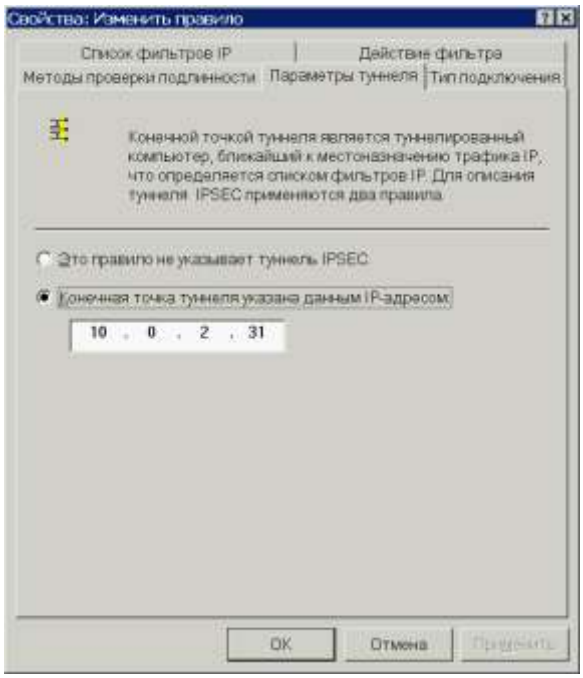


3.3. На вкладке "Методы проверки подлинности" создать или изменить единственный метод — с использованием предварительного ключа. В окне свойств метода выбрать опцию "Использовать данную строку (предварительный ключ) и ввести ключ, установленный на устройстве NSG в узле ipsec.secrets . Методы, существующие по умолчанию, удалить или передвинуть в конец списка. Данная настройка одинакова для обоих фильтров.

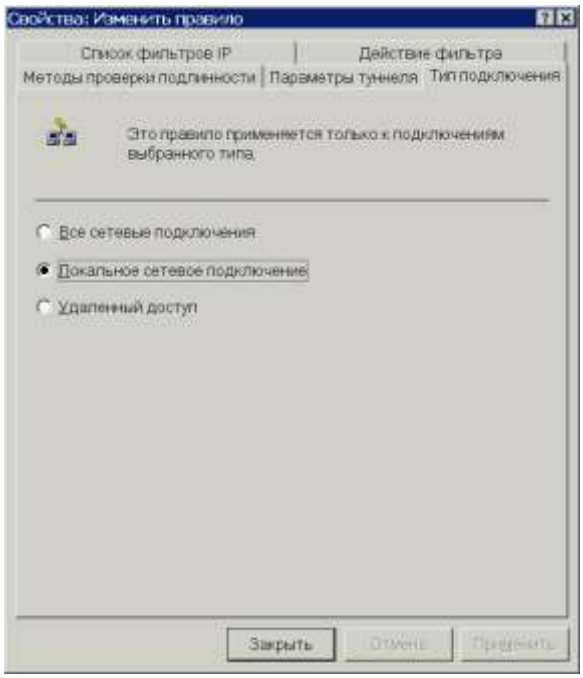


3.4. На вкладке "Параметры туннеля" для каждого из фильтров выбрать опцию "Конечная точка туннеля указана данным IP-адресом" и указать адрес внешнего (открытого) IP-интерфейса удалённой стороны. На рисунке слева фильтр NSG–MS, справа MS–NSG.

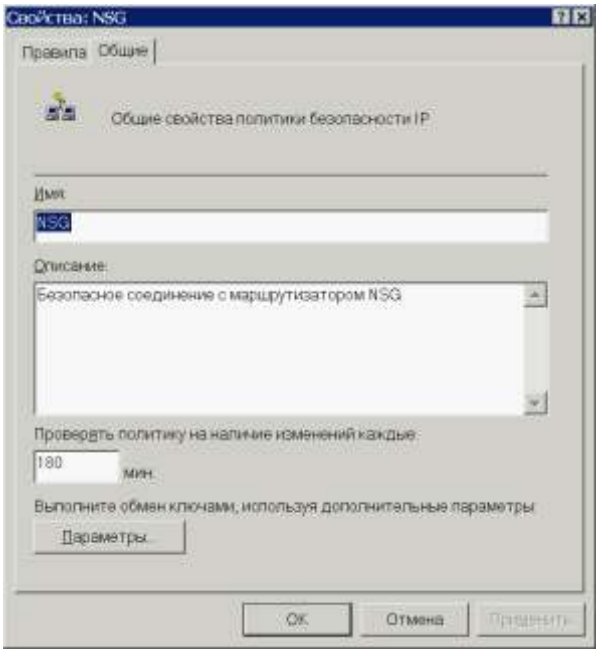




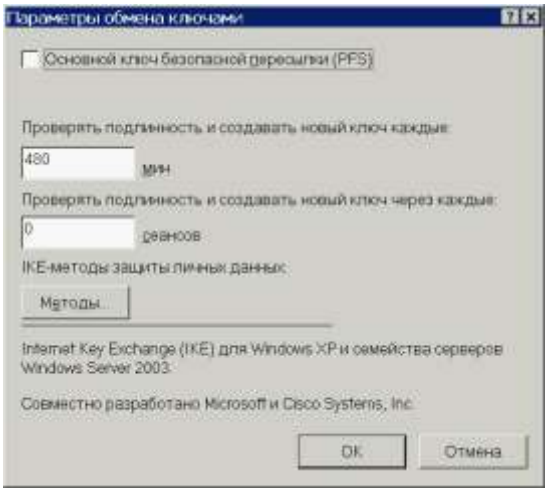
3.5. На вкладке "Тип подключения" выбрать опцию "Локальное сетевое подключение" для обоих фильтров. Закрыть окно свойств фильтра.



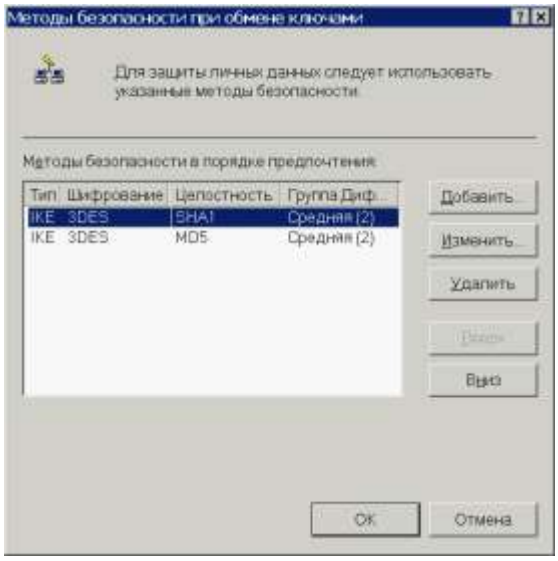
4. На вкладке "Общие" окна "Свойства политики" установить имя политики и её описание для административных целей. То и другое может быть произвольным, удобным администратору. Нажать кнопку "Параметры".



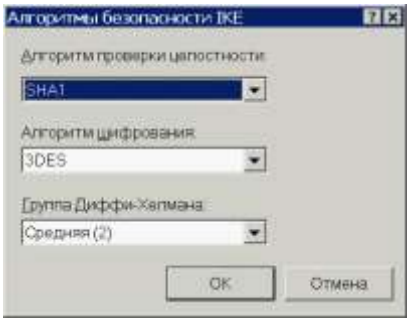
4.1. В открывшемся окне "Параметры обмена ключами" отключить опцию "Основной ключ безопасной пересылки (PFS)". Нажать кнопку "Методы".



4.1.1. В открывшемся окне "Методы безопасности при обмене ключами" оставить только один или два метода IKE: шифрование — 3DES, целостность — SHA1 или MD5, группа Диффи-Хеллмана — средняя (2).



4.1.1.1. Для создания и редактирования методов используются кнопки "Добавить" и "Изменить", которые открывают окно "Алгоритмы безопасности". Здесь следует выбрать необходимые параметры, а затем последовательно закрыть все окна свойств.



5. В консоли локальной политики безопасности выбрать политику "NSG" и совершить над ней действие "Назначить". Убедиться, что в колонке "Назначенная политика" для данной политике стоит значение "Да" (см. рис. выше в п.1).
6. Убедиться, что хосты из обеих сетей доступны друг для друга.
7. Убедиться, что трафик действительно передаётся по защищённому туннелю, с помощью команд tunnel.ipsec.show.* на устройстве NSG, либо с помощью *ping* между приватными интерфейсами обоих устройств (15.0.0.1 и 17.0.0.1)

ПРИМЕЧАНИЕ Как утверждает корпорация Майкрософт (см. статью 252735 Базы Знаний), её продукты не поддерживают подключение в транспортном режиме IPsec (т.е. в качестве конечного хоста) через удалённый шлюз к сети, находящейся за этим шлюзом. При необходимости подключения удалённых пользователей по такой схеме используется инкапсуляция IPsec-over-L2TP. Поскольку данная версия NSG Linux не поддерживает L2TP, то включение по такой схеме невозможно.

Как альтернативу, и альтернативу предпочтительную, корпорация Майкрософт рекомендует использовать туннели PPTP в сочетании с MPPE, реализующие схожую схему инкапсуляции IP (через защиту данных и протокол второго уровня снова в IP). Следует отметить, однако, что на сегодняшний день алгоритм MPPE не является полностью безопасным.

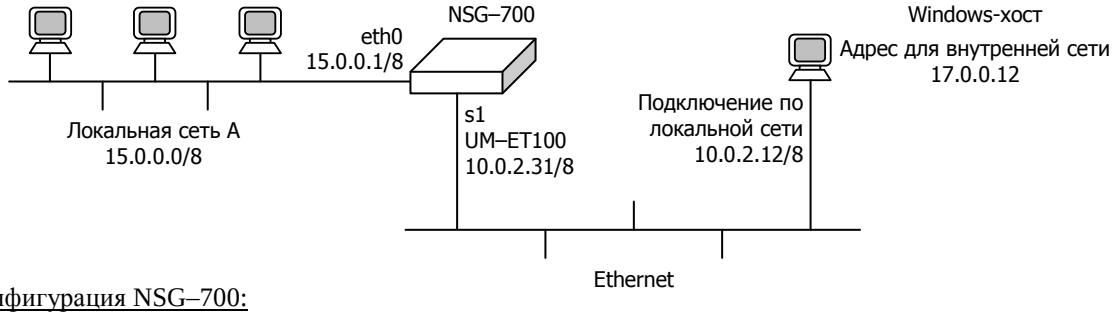
Другое решение задачи состоит в использовании программных VPN-клиентов сторонних разработчиков (см. след. параграф).

§5–A.8. Настройка туннеля IPsec (IKE) между NSG и IPsec-клиентами для Windows

Собственно операционные системы Windows поддерживают работу только в туннельном режиме в качестве промежуточного шлюза (см. §5–A.7). По этой причине соединить одиночный Windows-хост с устройством NSG при помощи туннеля IPsec собственными средствами Windows невозможно. Однако существует значительное число IPsec клиентов сторонних разработчиков, позволяющих решить эту задачу. Все они, по существу, организуют в системе внутренний шлюз IPsec, за которым оказывается основной стек IP, доступный приложениям и имеющий отдельный IP-адрес. Между этим шлюзом и устройством NSG организуется туннель, внутри которого передаётся трафик между приложениями Windows и сетью, расположенной за устройством NSG.

Схема стенда показана на рисунке. Для большей ясности задачи будем предполагать, что два шлюза соединены между собой просто сетью Ethernet. Курсивом показаны существенные настройки, установленные по умолчанию.

ВНИМАНИЕ Предполагается, что до начала настройки IPsec на обоих шлюзах настроена маршрутизация, так что хосты из защищённой сети успешно обмениваются IP-пакетами с Windows-хостом, и наоборот.



Конфигурация NSG–700:

```
ip
: route
: : 1
: : : gateway          = "10.0.2.12"
: : : network          = "17.0.0.0/8"
port
: eth0
: : ifAddress
: : : prefix           = "15.0.0.1/8"
: s1
: : type               = "eth"
: : ifAddress
: : : prefix           = "10.0.2.31/8"
tunnel
: ipsec
: : enable             = true
: : secrets
: : : psk
: : : : 1
: : : : : indices
: : : : : : 1          = "10.0.2.31"
: : : : : : 2          = "10.0.2.12"
: : : : : secret       = "12345678"
: : connections
: : : nsg2greenbow
: : : : authby         = "secret"
: : : : auto           = "start"
: : : : esp
: : : : : 3des-sha1    = true
: : : : left           = "10.0.2.31"
: : : : leftid         = "10.0.2.31"
: : : : leftnexthop    = "%direct "
: : : : leftsourceip   = "15.0.0.1"
: : : : leftsubnet     = "15.0.0.0/8"
: : : : pfs            = "no"
: : : : right          = "10.0.2.12"
: : : : rightid        = "10.0.2.12"
: : : : rightnexthop   = "%direct"
: : : : rightsubnet    = "17.0.0.12/32"
```

только для проверки маршрутизации; после настройки VPN удалить

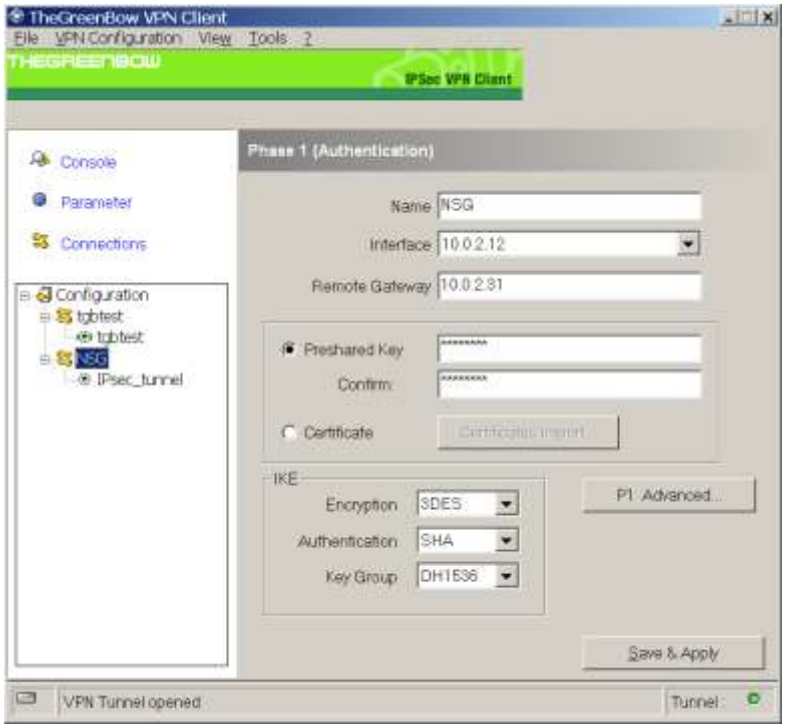
Настройка TheGreenBow VPN Client (http://www.thegreenbow.com):

Вся конфигурация укладывается в двух компактных окнах. В окне Phase 1 описываются имя и параметры туннеля, в окне Phase 2 — параметры закрываемого трафика. Оба набора параметров создаются при помощи щелчка правой клавишей мышью и пункта Add ... в выпадающем меню. Настройка показана на рисунках, существенные параметры перечислены ниже.

Encryption — 3DES

Authentication — SHA или MD5

Key Group — DH1024 или DH1536



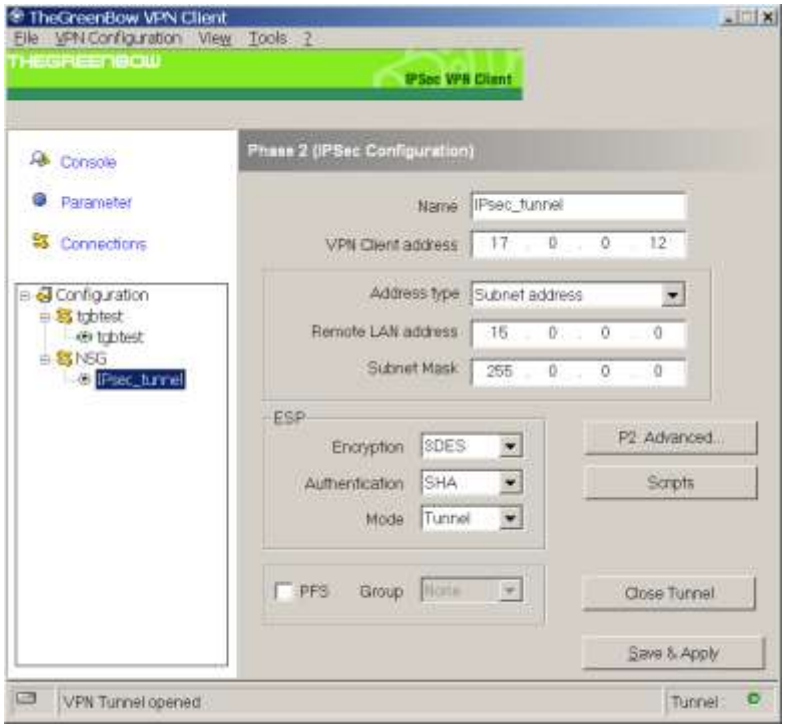
Encryption — 3DES

Authentication — SHA или MD5

Mode — Tunnel

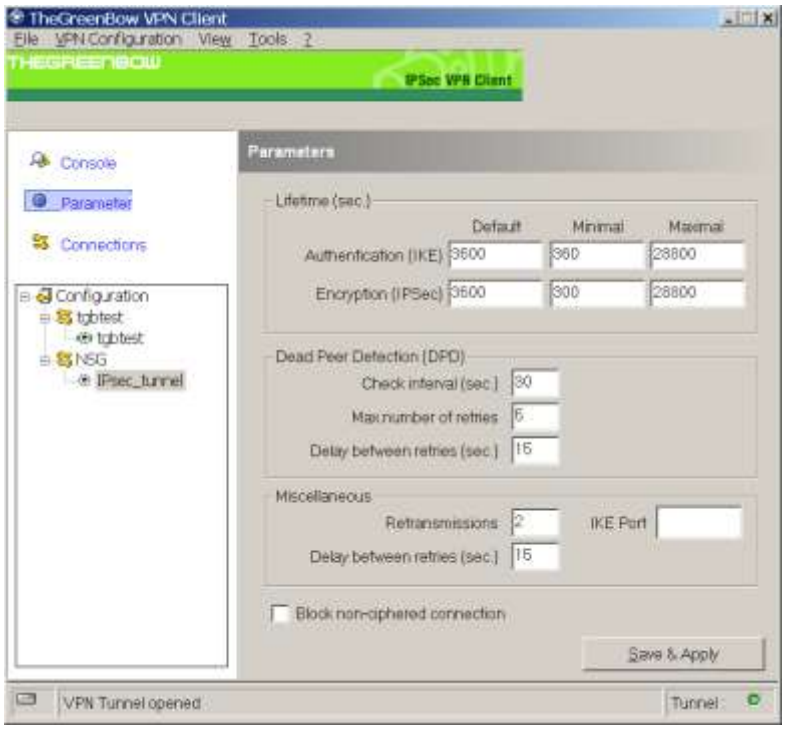
PFS — отключено

ПРИМЕЧАНИЕ
Внутренний адрес IP-клиента может как совпадать с внешним (10.0.2.12), так и отличаться от него (17.0.0.12), при соответствующих настройках left/rightsubnet на устройстве NSG.

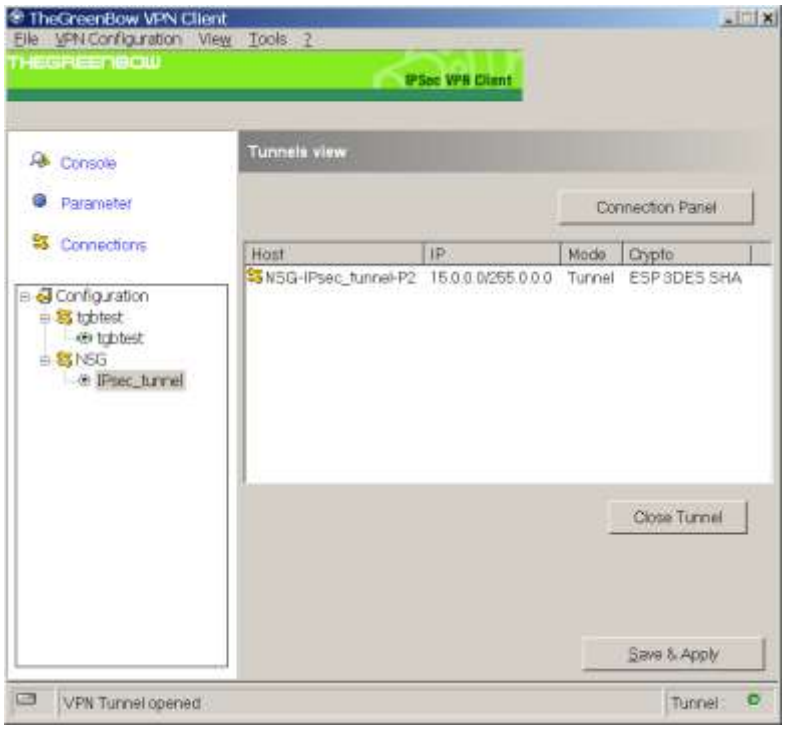


Каждый туннель устанавливается и разрывается вручную по нажатию кнопки Open/Close Tunnel в окне Phase 2.

Дополнительные параметры для всех туннелей, такие как время жизни ключей, можно установить в окне Parameter.



Установленные туннели можно просмотреть в окне Connections.



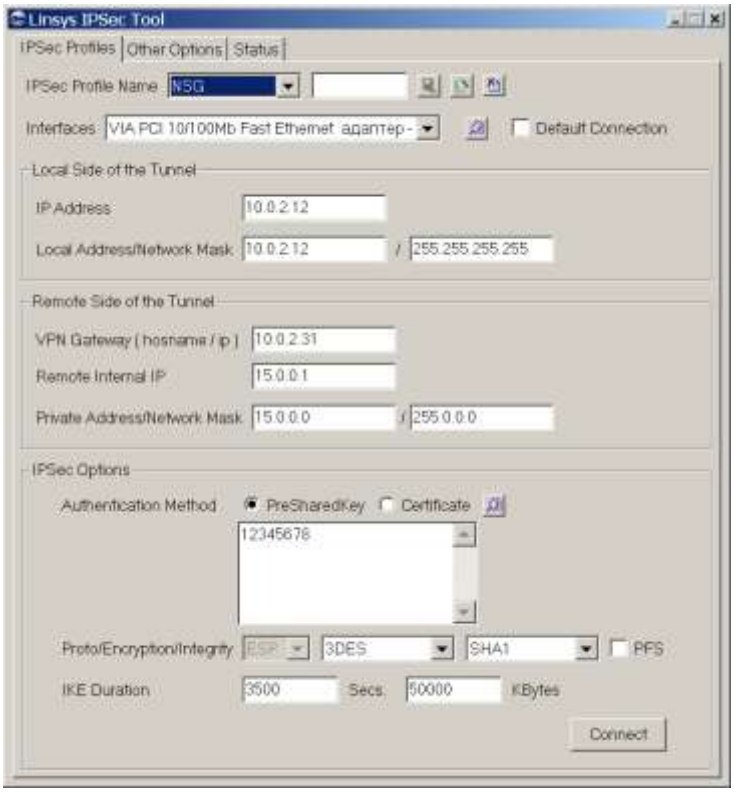
Настройка Linsys IPsec Tool (с открытым кодом, http://sourceforge.net/project/showfiles.php?group_id=139613)

Данный клиент может работать как в режиме одиночного хоста, так и в режиме шлюза, за которым расположен закрываемая подсеть (при соответствующих настройках left/rightsubnet на устройстве NSG). При этом в режиме одиночного хоста внутренний IP-адрес должен совпадать с внешним (см. рисунок).

Конфигурация NSG–700:

```
port
: eth0
: : ifAddress
: : : prefix              = "15.0.0.1/8"
: s1
: : type                  = "eth"
: : ifAddress
: : : prefix              = "10.0.2.31/8"
tunnel
: ipsec
: : enable                = true
: : secrets
: : : psk
: : : : 1
: : : : : indices
: : : : : : 1             = "10.0.2.31"
: : : : : : 2             = "10.0.2.12"
: : : : : secret          = "12345678"
: : connections
: : : nsg2linsys
: : : : authby            = "secret"
: : : : auto              = "start"
: : : : esp
: : : : : 3des-sha1       = true
: : : : left              = "10.0.2.31"
: : : : leftnexthop       = "%direct "
: : : : leftsourceip      = "15.0.0.1"
: : : : leftsubnet        = "15.0.0.0/8"
: : : : pfs               = "no"
: : : : right             = "10.0.2.12"
: : : : rightnexthop      = "%direct"
: : : : rightsubnet       = "10.0.0.12/32"
```

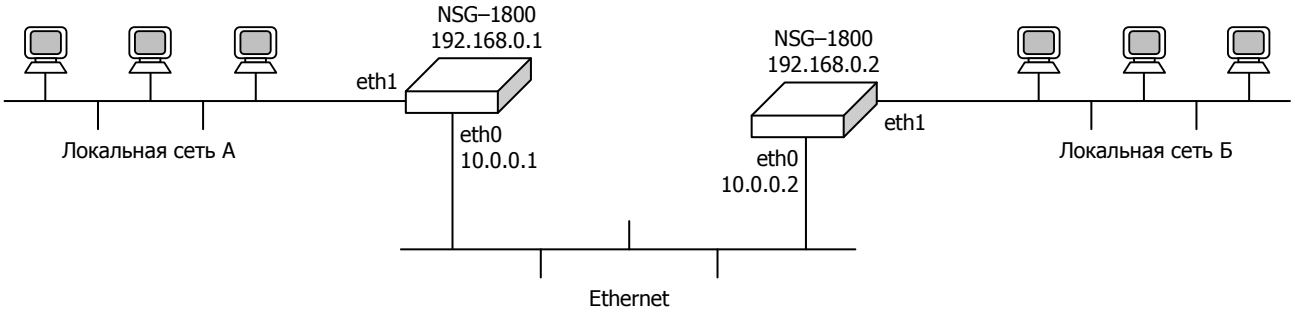
Настройка Linsys:



Если в поле Local Address/Network Mask указана какая-либо другая подсеть (подключённая к другому интерфейсу хоста), то пакеты из этой сети будут направляться в туннель. Однако пакеты, отправляемые с самого Windows-хоста, будут иметь адрес источника, равный внешнему IP-адресу (10.0.2.12) и, соответственно, передаваться вне туннеля. Является ли такое поведение багом или фичей — рекомендуется уточнить у разработчиков данного клиента.

§5–A.9. Объединение сетей Ethernet через GRE и IPsec

Имеются две локальные сети офисов, между которыми имеется высокоскоростное соединение. Требуется прозрачно объединить их на втором уровне (в режиме моста) и защитить передаваемые данные с помощью IPsec. Схема стенда представлена на рисунке. Используются два устройства NSG–1800.



Конфигурация публичных интерфейсов и маршрутов. Маршрут в сеть 10.0.0.0/8 в данном случае создаётся по умолчанию, но для работы IPsec в нижеприведённой конфигурации всё равно необходимо иметь маршрут по умолчанию.

```
ip                                          ip
: route                                     : route
: : 1                                       : : 1
: : : network        = "0.0.0.0/0"          : : : network        = "0.0.0.0/0"
: : : gateway        = "10.0.0.2"           : : : gateway        = "10.0.0.1"
: : : _keep          = true                 : : : _keep          = true
port                                        port
: eth0                                      : eth0
: : ifAddress                               : : ifAddress
: : : prefix = "10.0.0.1/8"                 : : : prefix = "10.0.0.2/8"
```

Между устройствами образован туннель Ethernet-over-GRE, концы которого соединены с локальными сетями при помощи *bridge groups*. Сами устройства доступны в объединённой локальной сети по адресам 192.168.0.1 и 192.168.0.2, соответственно.

```
bridge                                      bridge
: br1                                       : br1
: : ifAddress                               : : ifAddress
: : : prefix = "192.168.0.1/24"             : : : prefix = "192.168.0.2/24"
port                                        port
: eth1                                      : eth1
: : bridge-group= "br1"                     : : bridge-group= "br1"
tunnel                                      tunnel
: gre                                       : gre
: : gre1                                    : : gre1
: : : encapsulation= "eth"                  : : : encapsulation= "eth"
: : : source= "10.0.0.1"                    : : : source= "10.0.0.2"
: : : destination= "10.0.0.2"               : : : destination= "10.0.0.1"
: : : bridge-group= "br1"                   : : : bridge-group= "br1"
```

Пакеты этого туннеля с адресами источника и назначения 10.0.0.1 и 10.0.0.2 направляются в туннель IPSec, при этом конфигурация получается идентичной для обеих сторон:

```
tunnel
: ipsec
: : enable                  = true
: : secrets
: : : psk
: : : : 1
: : : : : indices
: : : : : : 1               = "10.0.0.1"
: : : : : : 2               = "10.0.0.2"
: : : : : secret            = "XAXA"
: : connections
: : : secure_bridge
: : : : authby              = "secret"
: : : : auto                = "start"
: : : : esp
```

```
: : : : : 3des-md5          = true
: : : : left                = "10.0.0.1"
: : : : leftsubnet          = "10.0.0.1/32"
: : : : right               = "10.0.0.2"
: : : : rightsubnet         = "10.0.0.2/32"
```

В данном примере использована особенность реализации IPsec в Linux, допускающая направлять в туннель любые IPsec пакеты с адресами, равными публичным адресам шлюзов. Именно это позволяет использовать для концов туннеля GRE те же адреса публичных интерфейсов 10.0.0.1 и 10.0.0.2, как это было бы в случае без IPsec.

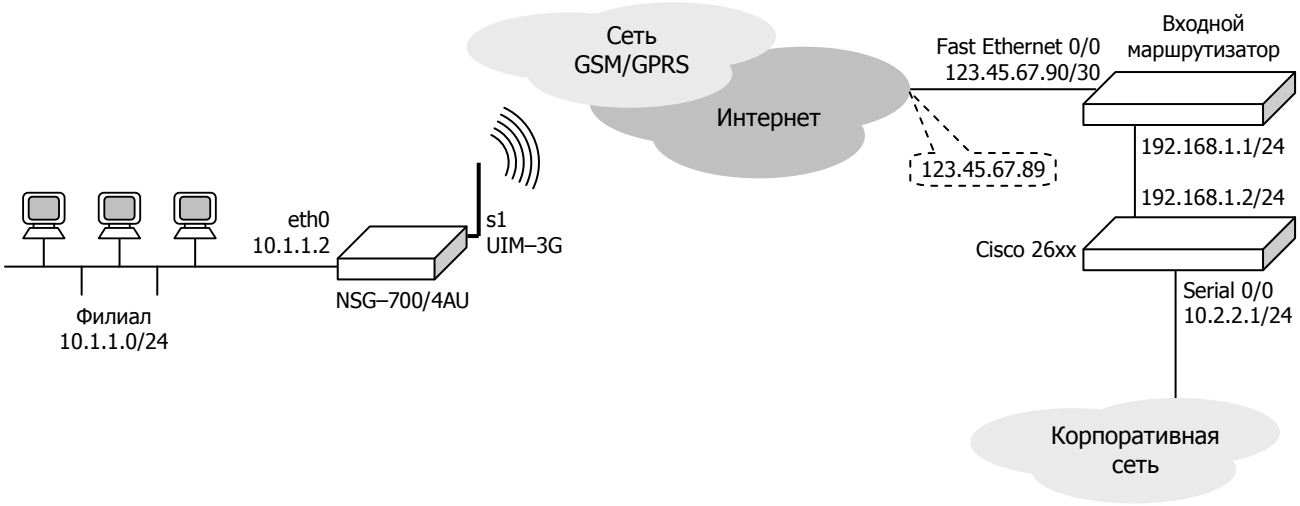
С другой стороны, чтобы избежать путаницы с адресами, можно терминировать туннель GRE на адресах, присвоенных каким-либо другим интерфейсам или псевдоинтерфейсам. В нижеприведённом примере адреса 11.0.0.1 и 11.0.0.2 назначены локальным псевдоинтерфейсам dummy0 соответствующих устройств.

```
bridge                                          bridge
: br1                                           : br1
: : ifAddress                                   : : ifAddress
: : : prefix            = "192.168.0.1/24"      : : : prefix            = "192.168.0.2/24"
ip                                              ip
: route                                         : route
: : 1                                           : : 1
: : : network           = "0.0.0.0/0"           : : : network           = "0.0.0.0/0"
: : : gateway           = "10.0.0.2"            : : : gateway           = "10.0.0.1"
: : : _keep             = true                  : : : _keep             = true
port                                            port
: eth0                                          : eth0
: : ifAddress                                   : : ifAddress
: : : prefix            = "10.0.0.1/8"          : : : prefix            = "10.0.0.2/8"
: eth1                                          : eth1
: : bridge-group        = "br1"                 : : bridge-group        = "br1"
pseudo-interface                                pseud-interface
: dummy0                                        : dummy0
: : ifAddress                                   : : ifAddress
: : : prefix            = "11.0.0.1/32"         : : : prefix            = "11.0.0.2/32"
tunnel                                          tunnel
: gre                                           : gre
: : gre1                                        : : gre1
: : : encapsulation     = "eth"                 : : : encapsulation     = "eth"
: : : source            = "11.0.0.1"            : : : source            = "11.0.0.2"
: : : destination       = "11.0.0.2"            : : : destination       = "11.0.0.1"
: : : bridge-group      = "br1"                 : : : bridge-group      = "br1"
: ipsec                                         : ipsec
: : enable              = true                  : : enable              = true
: : secrets                                     : : secrets
: : : psk                                       : : : psk
: : : : 1                                       : : : : 1
: : : : : indices                               : : : : : indices
: : : : : : 1           = "10.0.0.1"            : : : : : : 1           = "10.0.0.1"
: : : : : : 2           = "10.0.0.2"            : : : : : : 2           = "10.0.0.2"
: : : : : secret         = "XAXA"                : : : : : secret         = "XAXA"
: : connections                                 : : connections
: : : secure_bridge                             : : : secure_bridge
: : : : authby          = "secret"              : : : : authby          = "secret"
: : : : auto            = "start"               : : : : auto            = "start"
: : : : esp                                     : : : : esp
: : : : : 3des-md5      = true                  : : : : : 3des-md5      = true
: : : : left            = "10.0.0.1"            : : : : left            = "10.0.0.1"
: : : : leftsubnet      = "11.0.0.1/32"         : : : : leftsubnet      = "11.0.0.1/32"
: : : : right           = "10.0.0.2"            : : : : right           = "10.0.0.2"
: : : : rightsubnet     = "11.0.0.2/32"         : : : : rightsubnet     = "11.0.0.2/32"
```

§5–A.10. Соединение NSG–Cisco с использованием *Destination NAT* и *dynamic map*

Требуется использовать устройство NSG в качестве клиентского в схеме, описанной в документе Cisco:

PIX/ASA 7.x and later : Dynamic IPsec Between a Statically addressed PIX and a Dynamically addressed IOS Router with NAT Configuration Example



Основные особенности рассматриваемого решения:

— Сервер расположен во внутренней сети головного офиса, скрытой за Destination NAT на входном маршрутизаторе.

— Клиент(ы) получает динамический приватный адрес для доступа в сеть поставщика услуг. Выход из сети поставщика услуг в Интернет осуществляется с использованием Source NAT.

— Трафик из локальной сети филиала (10.1.1.0/24) в сеть головного офиса (10.2.2.0/24) направляется в безопасный туннель IPsec.

— Весь остальной трафик из локальной сети филиала во внешний мир NAT-ируется и отправляется с Source IP публичного интерфейса NSG.

— Используется механизм контроля целостности туннеля (DPD), а также поддержание и восстановление IPSec-туннеля в случае переустановки соединения с сетью общего пользования.

Используется устройство NSG–700/4AU с модулем UIM–3G. Поставщик услуг GPRS — МТС. Курсивом показаны существенные настройки, установленные по умолчанию.

Конфигурация устройства NSG–700:

Подключение к сотовой сети:

```
port
: s1
: type                      = "3g"
: : adm-state               = "up"
: : ppp
: : : main
: : : : chat
: : : : : apn               = "internet.mts.ru"
: : : : default-route       = true
: : : : ipcp
: : : : : accept-address    = true
: : : : : accept-peer-address = true
: : : : lcp-echo-failure    = 3
: : : : lcp-echo-interval   = 10
: : : : sent-username       = "mts"
system
: ppp-secrets
: : chap
: : : 1
: : : : client              = "mts"
: : : : secret              = "mts"
```

Настройка IPsec. В данном примере для аутентификации на стороне NSG устройство Cisco идентифицирует себя по имени, причём это возможно только потому, что инициатором создания туннеля является NSG. В обратном направлении такая аутентификация невозможна, поэтому на обеих сторонах устройство NSG формально идентифицирует себя по IP-адресу, но этот адрес может быть произвольным. Для аутентификации на стороне Cisco само устройство Cisco также идентифицирует себя произвольным IP-адресом.

```
tunnel
: ipsec
: : enable                = true
: : secrets
: : : psk
: : : : 1
: : : : : indices
: : : : : : 1             = "@Router."
: : : : : : 2             = "%any"
: : : : : secret          = "12345678"
: : setup
: : : nat-traversal       = "yes"
: : connections
: : : nsg2cisco
: : : : authby            = "secret"
: : : : dpddelay          = 15
: : : : dpdtimeout        = 30
: : : : esp
: : : : : 3des-md5        = true
: : : : left              = "%defaultroute"
: : : : leftsourceip      = "10.1.1.2"
: : : : leftsubnet        = "10.1.1.0/24"
: : : : right             = "123.45.67.90"
: : : : rightsubnet       = "10.2.2.0/24"
```

Конфигурация устройства Cisco–26xx:

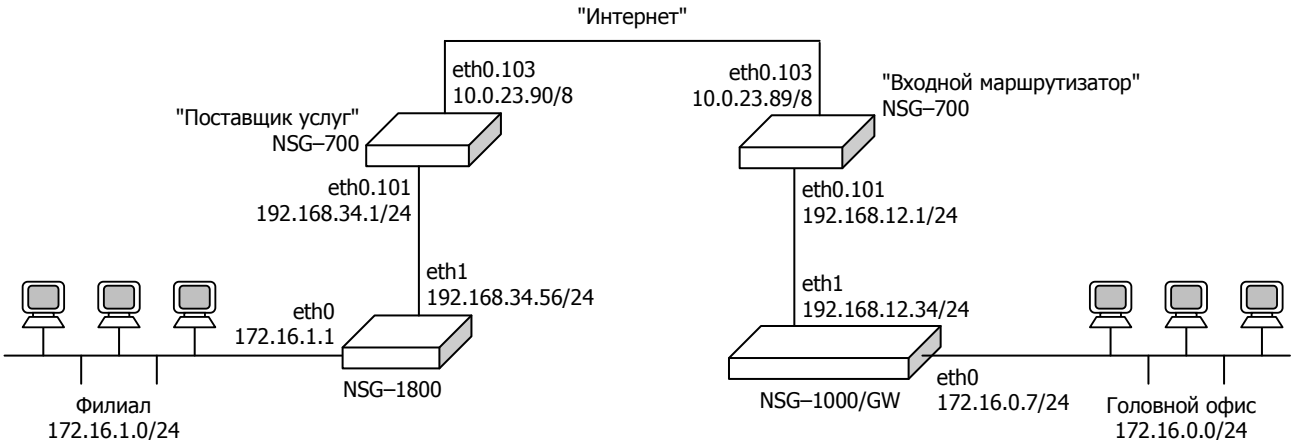
```
!
version 12.3
hostname Router
!
crypto isakmp policy 1
    encr 3des
    hash md5
    authentication pre-share
    group 2
crypto isakmp key 12345678 address 0.0.0.0 0.0.0.0
crypto isakmp identity hostname
crypto isakmp keepalive 10 periodic
crypto isakmp nat keepalive 10
!
crypto ipsec transform-set router-set esp-3des esp-md5-hmac
!
crypto dynamic-map cisco 1
    set transform-set router-set
    match address 101
!
crypto map dyn-map 10 ipsec-isakmp dynamic cisco
!
interface FastEthernet0/0
    ip address 192.168.1.2 255.255.255.0
    crypto map dyn-map
!
interface Serial0/0
    ip address 10.2.2.1 255.255.255.0
!
ip classless
ip route 0.0.0.0 0.0.0.0 192.168.1.1
!
access-list 101 permit ip 10.2.2.0 0.0.0.255 10.1.1.0 0.0.0.255
!
```

§5–A.11. Подключение клиентов IPsec с двойным NAT–T и сертификатами X.509

Устройство NSG–1000/GW используется в центральном офисе в качестве шлюза VPN, к которому подключаются удалённые филиалы по IPsec. При этом само оно находится внутри сети офиса, за входным маршрутизатором, на котором настроен Destination NAT для портов UDP 500 и 4500. Филиалы могут иметь динамические адреса, а также приватные адреса в сети поставщиков услуг, и выходить в Интернет через NAT. Таким образом, IP-адреса клиентов, под которыми они видны в Интернет, априори неизвестны, неинформативны и также могут не соответствовать их фактическим адресам, под которыми они известны сами себе.

Традиционный механизм аутентификации сторон и защиты данных с помощью PSK на сегодняшний день является достаточно надёжным только при условии статических IP-адресов обеих сторон (т.е. когда каждая сторона фактически аутентифицируется по совокупности реквизитов IP-адрес + PSK); применять его для произвольных адресов (0.0.0.0) не рекомендуется. По этой причине используется механизм аутентификации на основе асимметричной пары RSA-секретов, подтверждённых сертификатами X.509.

Схема стенда приведена на рисунке. В качестве клиента в удалённом офисе используется устройство NSG–1800. Два устройства NSG–700 моделируют Source NAT на выходе из сети поставщика услуг в Интернет и Destination NAT на входе в сеть головного офиса, а соединение между ними — собственно Интернет.



В соответствии с рекомендуемой мнемоникой для идентификации сторон в NSG Linux 2.0, "левой" стороной туннеля считается к**Л**иент, "правой" стороной — се**Р**ве**Р** (для удобства запоминания). Для большей ясности, вопреки традиционной для Linux избыточной и симметричной конфигурации, приведены только параметры, используемые по существу. NAT Traversal, по умолчанию, включёно.

Приватный ключ на устройстве NSG–1000/GW хранится в файле /etc/ipsec.d/private/serverkey.pem, а пароль от него записан в параметр secret. Сертификат хранится в файле /etc/ipsec.d/certs/server.pem. Помимо этих двух файлов, для работы системы X.509 нужен, как минимум, ещё один файл — корневой сертификат, который хранится в файле /etc/ipsec.d/cacert/root.pem . Аналогичным образом разложены файлы сертификатов и ключей на клиентских устройствах, различаются только имена файлов.

Строка rightid должна строго соответствовать всем перечисленным полям сертификата сервера, а все строки leftid — сертификата соответствующего клиента.

Конфигурация сервера:

```
ip
: route
: : 1
: : : gateway          = "192.168.12.1"
: : : network          = "0.0.0.0/0"
port
: eth0
: : ifAddress
: : : prefix           = "172.16.0.7/24"
: eth1
: : ifAddress
: : : prefix           = "192.168.12.34/24"
system
: ntp
: : enable             = true
: : host               = "194.149.67.129"
tunnel
: ipsec
: : enable             = true
: : secrets
: : : rsa
: : : : 1
: : : : : file         = "serverkey.pem"
: : : : : secret       = "PASS_SERVER"
: : connections
: : : %default
: : : : authby         = "rsasig"
: : : : esp
: : : : : 3des-md5     = true
: : : : left           = "%any"
: : : : leftrsasigkey  = "%cert"
: : : : right          = "192.168.12.34"
: : : : rightcert      = "server.pem"
: : : : rightid        = "/C=RU/ST=MO/L=Moscow/O=NSG/OU=programmers/CN=radius_server/email ↵
Address=support@nsg.net.ru"
: : : : rightnexthop   = "192.168.12.1"
: : : : rightsourceip  = "172.16.0.7"
: : : : rightsubnet    = "172.16.0.0/24"
: : : branch1
: : : : auto           = "add"
: : : : leftid         = "/C=RU/ST=MO/L=Moscow/O=NSG/OU=programmers/CN=user1/email ↵
Address=support@nsg.net.ru"
: : : : leftsubnet     = "172.16.1.0/24"
: : : branch2
.....
```

Конфигурация клиента:

```
ip
: route
: : 1
: : : gateway          = "192.168.34.1"
: : : network          = "0.0.0.0/0"
port
: eth0
: : ifAddress
: : : prefix           = "172.16.1.1/24"
: eth1
: : ifAddress
: : : prefix           = "192.168.34.56/24"
system
: ntp
: : enable             = true
: : host               = "194.149.67.129"
tunnel
: ipsec
: : enable             = true
: : secrets
: : : rsa
: : : : 1
: : : : : file         = "clientkey.pem"
: : : : : secret       = "PASS_USER1"
: : connections
: : : branch
: : : : authby         = "rsasig"
: : : : auto           = "start"
: : : : esp
: : : : : 3des-md5     = true
: : : : left           = "%defaultroute"
: : : : leftcert       = "client.pem"
: : : : leftid         = "/C=RU/ST=MO/L=Moscow/O=NSG/OU=programmers/CN=user1/emailAddress=
support@nsg.net.ru"
: : : : leftnexthop    = "%defaultroute"
: : : : leftsourceip   = "172.16.1.1"
: : : : leftsubnet     = "172.16.1.0/24"
: : : : right          = "10.0.23.89"
: : : : rightid        = "/C=RU/ST=MO/L=Moscow/O=NSG/OU=programmers/CN=radius_server/email
Address=support@nsg.net.ru"
: : : : rightrsasigkey = "%cert"
: : : : rightsubnet    = "172.16.0.0/24"
```

ПРИМЕЧАНИЕ В данной реализации IPsec идентификатор клиента (leftid) на самом клиенте не требуется по существу, однако на стороне сервера они обязательны оба. Во избежание ошибок рекомендуется всегда указывать оба идентификатора на обеих сторонах.

Конфигурация промежуточных устройств NSG–700:

"Поставщик услуг":

```
ethernet
: switch
: : phy0
: : : vlan-groups
: : : : 1                = 101
: : : : 2                = 102
: : : : 3                = 103
: : : vlan-tagged        = true
: : phy1
: : : vlan-group         = 101
: : phy2
: : : vlan-group         = 102
: : phy3
: : : vlan-group         = 103
: : vlan-mode            = true
ip
: nat
: : POSTROUTING
: : : 1
: : : : out-interface    = "eth0.103"
: : : : target           = "MASQUERADE"
port
: eth0
: : vlan
: : : eth0.101
: : : : ifAddress
: : : : : prefix         = "192.168.34.1/24"
: : : eth0.103
: : : : ifAddress
: : : : : prefix         = "10.0.23.90/8"
```

"Входной маршрутизатор":

```
ethernet
: switch
: : phy0
: : : vlan-groups
: : : : 1                = 101
: : : : 2                = 102
: : : : 3                = 103
: : : vlan-tagged        = true
: : phy1
: : : vlan-group         = 101
: : phy2
: : : vlan-group         = 102
: : phy3
: : : vlan-group         = 103
: : vlan-mode            = true
ip
: nat
: : PREROUTING
: : : 1
: : : : protocol         = "udp"
: : : : in-interface     = "eth0.103"
: : : : destination      = "10.0.23.89"
: : : : source-port      = "500"
: : : : destination-port = "500"
: : : : target           = "DNAT"
: : : : to-destination   = "192.168.12.34"
: : : 2
: : : : protocol         = "udp"
: : : : in-interface     = "eth0.103"
: : : : destination      = "10.0.23.89"
: : : : source-port      = "4500"
: : : : destination-port = "4500"
: : : : target           = "DNAT"
: : : : to-destination   = "192.168.12.34"
port
: eth0
: : vlan
: : : eth0.101
: : : : ifAddress
: : : : : prefix         = "192.168.12.1/24"
: : : eth0.103
: : : : ifAddress
: : : : : prefix         = "10.0.23.89/8"
```

Эквивалентная конфигурация NSG–700 "поставщика услуг" с NSG Linux 1.0:

```
!
nsg
    access-list  ext-ip 101
        add 1 permit ip any any
        exit
    ethernet-switch
        mode vlan
        exit
    port eth0
        vlan 101
            ip address 192.168.34.1/24
            exit
        vlan 103
            ip address 10.0.23.90/8
            nat source prio 1 access-list 101 masquerade
            exit
        exit
!
```