Internet Engineering Task Force (IETF)
Request for Comments: 9061
Category: Standards Track
ISSN: 2070-1721

R. Marin-Lopez
G. Lopez-Millan
University of Murcia
F. Pereniguez-Garcia
University Defense Center

July 2021

A YANG Data Model for IPsec Flow Protection Based on Software-Defined Networking (SDN)

Модель данных YANG для защиты потоков IPsec на основе SDN

Аннотация

Этот документ описывает защиту потоков IPsec (целостность и конфиденциальность) через интерфейс с контроллером функций сетевой защиты (Interface to Network Security Function или I2NSF). Рассматриваются два базовых общеизвестных сценария IPsec - взаимодействие между шлюзами (gateway-to-gateway) и взаимодействие между хостами (host-to-host). Описанная в документе служба позволяет настраивать и отслеживать защищённые связи IPsec (Security Association или SA) от контроллера I2NSF к одной или нескольким основанным на потоках функциям защиты сети (Network Security Function или NSF), применяющих IPsec для защиты трафика данных.

Документ сосредоточен на интерфейсе I2NSF в сторону NSF и обеспечивает модели данных YANG для настройки баз данных IPsec, а именно, базы правил защиты (Security Policy Database или SPD), базы защищённых связей (Security Association Database или SAD), базы полномочий партнёров (Peer Authorization Database или PAD) и обмена ключами в Internet версии 2 (Internet Key Exchange Version 2 или IKEv2). Это позволяет создавать IPsec SA с минимальным вовлечением сетевого администратора. Документ определяет 3 модуля YANG, но не задаёт новых протоколов.

Статус документа

Документ относится к категории Internet Standards Track.

Документ является результатом работы IETF[1] и представляет согласованный взгляд сообщества IETF. Документ прошёл открытое обсуждение и был одобрен для публикации IESG[2]. Дополнительную информацию о стандартах Internet можно найти в разделе 2 в RFC 7841.

Информация о текущем статусе документа, найденных ошибках и способах обратной связи доступна по ссылке https://www.rfc-editor.org/info/rfc9061.

Оглавление

[1]Internet Engineering Task Force - комиссия по решению инженерных задач Internet.
[2]Internet Engineering Steering Group - комиссия по инженерным разработкам Internet.

1. Введение

Архитектура программно-определяемых сетей (Software-Defined Networking или SDN) позволяет администраторам напрямую программировать, организовывать и управлять сетевыми ресурсами черех программы. Парадигма SDN переносит управления сетевыми ресурсами централизованному объекту - контроллеру SDN. Контроллер SDN настраивает и управляет сетевыми ресурсами, а также обеспечивает абстрактное представление сетевых ресурсов приложениям SDN. Эти приложения могут настраивать и автоматизировать операции (включая управление) абстрактных сетевых ресурсов программным способом через интерфейс [RFC7149] [ITU-T.Y.3300] [ONF-SDN-Architecture] [ONF-OpenFlow].

В последнее время некоторые сетевые сценарии требуют централизованного управления различными аспектами безопасности, например, программно-управляемые сети WAN (Software-Defined WAN или SD-WAN). SD-WAN представляют собой расширения SDN, обеспечивающие программные абстракции для создания защищённых наложенных сетей на основе традиционных WAN и сетей филиалов. В SD-WAN применяется IPsec [RFC4301] как базовый протокол защиты. Цель SD-WAN состоит в предоставлении гибкого и автоматизируемого развёртывания из центральной точки для поддержки по запросам услуг защиты, таких как управление защищёнными связями IPsec (IPsec Security Association или IPsec SA). В параграфе 4.3.3 (Client-Specific Security Policy in Cloud VPNs) [RFC8192] описан другой пример использования для облачного центра обработки данных (ЦОД). В примере [RFC8192] сказано, что: «динамическое управление ключами очень важно для защиты VPN и распространения правил». Такие VPN могут создаваться с использованием IPsec. Управление IPsec SA у ЦОД с использованием централизованного объекта является вариантом, где может применяться эта спецификация.

Поэтому с расширением использования решений на основе SDN, где сетевые ресурсы разворачиваются автономно, механизм управления IPsec SA из центрального объекта становится все более актуальным для отрасли. В ответ на эту потребность устав рабочей группы I2NSF[1] ставит задачу: «определить набор программных интерфейсов и моделей данных для управления и мониторинга аспектов физических и виртуальных NSF». Как определено в [RFC8192], функцией сетевой безопасности (Network Security Function или NSF) является: «функция, применяемая для обеспечения целостности, конфиденциальности и доступности сетевых коммуникаций, обнаружение нежелательной активности в сети, блокировка или смягчение последствий нежелательных действий». Этот документ обращает особое внимание на обеспечение целостности и конфиденциальности средствами IPsec. В параграфе 3.1.9 [RFC8192] сказано: «требуется контроллер для создания, управления и распространения различных ключей в распределенные NSF», однако «отсутствует стандартный интерфейс для предоставления защищённых связей и управления ими». На основе парадигмы SDN модель I2NSF [RFC8329] определяет централизованный объект (контроллер I2NSF), управляющий одной или множеством NSF через интерфейс I2NSF NSF-Facing. В этом документе задана архитектура, позволяющая контроллеру I2NSF выполнять процедуры управления ключами. В частности, определены три модели данных YANG для интерфейса I2NSF NSF-Facing, позволяющие контроллеру I2NSF настраивать и отслеживать основанные на потоках NSF с поддержкой IPsec.

Архитектура IPsec [RFC4301] чётко разделяет обработку для обеспечения услуг по защите пакетов IP и процедуры управления ключами для создания IPsec SA, что позволяет централизовать управления ключами в контроллере I2NSF. Этот документ рассматривает два типовых варианта автономного управления IPsec SA - между шлюзами (gateway-to-gateway) и между хостами (host-to-host) [RFC6071]. В этих случаях роль NSF могут играть хосты, шлюзы или те и другие сразу. По причине сложности вариант взаимодействия между хостом и шлюзам в документе не рассматривается. Причины этой сложности состоят в том, что в этом варианте хост может не управляться контроллером I2NSF и не может быть настроен. Тем не менее, заданные здесь интерфейсы можно рассматривать как отправную точку для анализа и представления решений для взаимодействия между хостом и шлюзом.

При определении моделей данных YANG для интерфейса I2NSF NSF-Facing в документе рассмотрены 2 общих случая, указанных ниже.

1. Вариант с IKE. NSF реализует протокол обмена ключами Internet версии 2 (Internet Key Exchange Version 2 или IKEv2) и базы данных IPsec для правил безопасности (Security Policy Database или SPD), защищённых связей (Security Association Database или SAD) и предоставления полномочий партнёрам (Peer Authorization Database или PAD). Контроллер I2NSF отвечает за предоставление NSF требуемых сведений в SPD и PAD (например, свидетельства IKE) и самом протоколе IKE (например, параметры для согласования IKE_SA_INIT).

[1] Interface to Network Security Functions - интерфейс с функциями сетевой безопасности.

2. Вариант без IKE. NSF реализует лишь базы данных IPsec (без IKE). Контроллер I2NSF предоставит требуемые параметры для создания действительных записей SPD и SAD NSF. Таким образом, NSF будет поддерживать лишь IPsec, а управление ключами будет передано контроллеру I2NSF.

В обоих случаях нужна модель данных YANG для интерфейса I2NSF NSF-Facing, чтобы безопасно обеспечить это предоставление между контроллером I2NSF и NSF. Используя язык моделирования YANG версии 1.1 [RFC7950], модели данных YANG из [netconf-vpn] и [TRAN-IPSECME-YANG], а также структуры данных из [RFC4301] и [RFC7296], этот документ определяет требуемые интерфейсы с моделью данных YANG для конфигурации и состояния IKE, PAD, SPD, SAD (см. параграфы 5.1 - 5.3). Предложенная модель данных YANG соответствует архитектуре хранилищ данных управления сетью (Network Management Datastore Architecture или NMDA) [RFC8342]. Примеры использования этих моделей данных представлены в приложениях A - C.

Ниже указаны основные цели этого документа.

- Описание архитектуры управления IPsec на основе I2NSF, позволяющего создавать и поддерживать IPsec SA с контроллера I2NSF для защиты конкретных потоков данных между парой основанных на потоках NSF, реализующих IPsec.
- Отображение этой архитектуры на модель I2NSF.
- Определение интерфейсов, требуемых для управления и мониторинга IPsec SA в NSF с контроллера I2NSF. Определены модели данных YANG для конфигурации и состояния IPsec и IKEv2 через интерфейс I2NSF NSF-Facing. Модели данных YANG можно применять по имеющимся протоколам, таким как протокол настройки сети (Network Configuration Protocol или NETCONF) [RFC6241] или RESTCONF [RFC8040]. Таким образом, этот документ задаёт три модуля YANG (см. раздел 5), но не определяет новых протоколов.

2. Терминология

Документ использует термины из [ITU-T.Y.3300], [RFC8192], [RFC4301], [RFC6437], [RFC7296], [RFC6241], [RFC8329].

Термин из [ITU-T.Y.3300]:

- Software-Defined Networking (SDN) - программно-определяемая сеть.

Термины из [RFC8192]:

- Network Security Function (NSF) - функция защиты сети;
- flow-based NSF - NSF на основе потоков.

Термины из [RFC4301]:

- Peer Authorization Database (PAD) - база данных о полномочиях партнёров;
- Security Association Database (SAD) - база данных о защищённых связях;
- Security Policy Database (SPD) - база правил безопасности.

Термины из [RFC6437]:

- flow - поток;
- traffic flow - поток трафика.

Термин из [RFC7296]:

- Internet Key Exchange Version 2 - (IKEv2) обмен ключами Internet версии 2.

Термины из [RFC6241]:

- configuration data - данные конфигурации;
- configuration datastore - хранилище данных конфигурации;
- state data - данные состояния;
- startup configuration datastore - хранилище данных стартовой конфигурации;
- running configuration datastore - хранилище данных рабочей конфигурации.

2.1. Уровни требований

Ключевые слова **должно** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не следует** (SHALL NOT), **следует** (SHOULD), **не нужно** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **не рекомендуется** (NOT RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе интерпретируются в соответствии с BCP 14 [RFC2119] [RFC8174] тогда и только тогда, когда они выделены шрифтом, как показано здесь.

3. Описание управления IPsec на основе SDN

Как отмечено в разделе 1, рассматриваются два варианта NSF - с реализацией IKEv2 и без IKE.

3.1. Вариант с IKE - IKEv2/IPsec в NSF

В этом случае NSF реализует IPsec с поддержкой IKEv2. Контроллер I2NSF отвечает за поддержку и применение сведений о соединениях IPsec (определение узлов, где нужно запустить сессии IKEv2/IPsec, идентификация типа защищаемого трафика, выведение и доставка свидетельств IKEv2, таких как заранее распределенные ключи, сертификаты и т. п.) и применение других параметров конфигурации IKEv2 (например, криптографических алгоритмов для организации IKEv2 SA) к NSF для согласования IKEv2.

Реализация IKEv2 может работать с этими записями для организации IPsec SA. Пользователь I2NSF устанавливает требования IPsec и сведения о конечных точках (через интерфейс I2NSF Consumer-Facing [RFC8329]), а контроллер I2NSF транслирует эти требования в записи IKEv2, SPD и PAD, которые будут установлены в NSF (через интерфейс I2NSF NSF-Facing). Имея эти сведения, NSF может просто запустить IKEv2 для организации требуемой IPsec SA (когда поток трафика нужно защищать). На рисунке 1 показаны уровни и соответствующая функциональность.

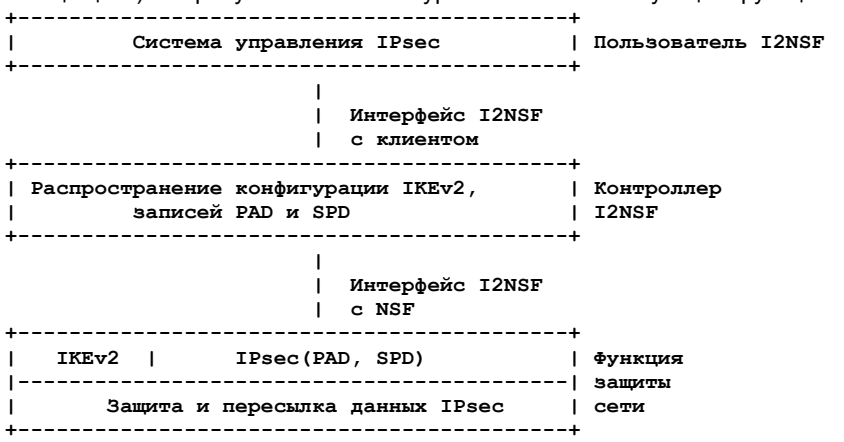

Рисунок 1.IKE/IPsec в NSF.

Службы защиты потоков IPsec на основе I2NSF обеспечивают гибкое динамическое управление IPsec SA в NSF на основе потоков. Для поддержки этого в варианте с IKE определяется модель YANG для данных конфигурации IKEv2, SPD, PAD и данных состояния IKEv2, требуемых для определения интерфейса I2NSF в сторону NSF (NSF-Facing), см. 5. Модели YANG для данных конфигурации.

3.2. Вариант без IKE - IPsec (без IKEv2) в NSF

В этом случае NSF не развёртывает IKEv2, поэтому контроллер I2NSF выполняет функции IKEv2 для защиты и управления IPsec SA, заполняя и поддерживая SPD и SAD. Как показано на рисунке 2, при применении пользователем I2NSF основанной на потоках политики защиты через интерфейс в сторону клиента (Consumer-Facing Interface) контроллер I2NSF транслирует требования правил в записи SPD и SAD, устанавливаемые в NSF. Записи PAD не требуются, поскольку в этом случае IKEv2 для NSF не применяется.

Рисунок 2. IPsec в NSF без IKEv2.

Для поддержки работы без IKE **должна** быть определена модель YANG для данных конфигурации SPD и SAD, а также данных состояния SAD для интерфейса NSF-Facing (см. 5. Модели YANG для данных конфигурации). В частности, вариант без IKE предполагает, что контроллер I2NSF выполняет некоторые функции защиты, которые обычно реализует IKEv2, а именно (неполный список):

- создание вектора инициализации (Initialization Vector или IV);
- предотвращение сброса счётчика для одного и того же ключа;
- генерация псевдослучайных криптографических ключей для IPsec SA;
- создание IPsec SA, когда этого требует уведомление (например, sadb-acquire) от NSF;
- смена ключей IPsec SA на основе уведомлений от NSF (завершение срока действия);
- обнаружение и поддержка работы через NAT.

В дополнение к этим функциям контроллер I2NSF **должен** выполнять другой набор задач (неполный список):

- генерация случайного индекса параметров защиты IPsec SA (Security Parameter Index или SPI);
- выбор криптографического алгоритма;
- использование расширенных порядковых номеров;
- организация подходящих селекторов трафика (Traffic Selector).

4. Сравнение вариантов с IKE и без IKE

В принципе, вариант с IKE проще развернуть, поскольку современные NSF (хосты или шлюзы) на основе потоков имеют доступ к реализациям IKEv2. Хотя реализация IKEv2/IPsec обычно развёртывается на шлюзах, это легко сделать и на хосте. Недостатком является потребность в большем объёме ресурсов для NSF при использовании IKEv2, например, памяти для реализации и расчётов IKEv2, поскольку каждая смена ключей IPsec SA может включать обмен Диффи-Хеллмана (Diffie-Hellman или DH).

Вариант без IKE даёт выигрыш при развёртывании на NSF с ограниченными ресурсами. Кроме того, IKEv2 не требуется при взаимодействии «шлюз-шлюз» или «хост-хост», когда оба работают с одним контроллером I2NSF (см. D.1. Пример организации IPsec SA). Сложность создания и поддержки IPsec SA переносится на контроллер I2NSF, поскольку IKEv2 не входит в NSF. В результате это может усложнять реализацию контроллера по сравнению с вариантом IKE. Например, контроллер I2NSF должен иметь дело с задержкой на пути между контроллером I2NSF и NSF (для решения таких задач, как смена ключа) или с созданием и установкой новых IPsec SA. Однако это относится не только к данному решению, но и к любой сети на основе SDN. Этот подход может осложнять расширение и создавать проблемы производительности при большом числе NSF.

Тем не менее, из литературы по управлению сетями на основе SDN с использованием централизованного контроллера (например, I2NSF Controller) известно о проблемах расширяемости и производительности и решения уже найдены и рассмотрены (например, иерархические контроллеры, включающие множество реплик, выделенные высокоскоростные сети управления и т. п.) В контексте управления IPsec на основе I2NSF одним из способов снижения задержки и устранения некоторых проблем производительности можно установить правила IPsec и IPsec SA одновременно (упреждающий режим, как описано в D.1. Пример организации IPsec SA) вместо ожидания уведомлений (например, sadb-acquire от NSF, требующего создания IPsec SA) для установки IPsec SA (реактивный режим). Другим способом сокращения издержек, а также решения вопросов расширяемости и производительности в контроллере I2NSF является применение варианта с IKE, описанного в этом документе, поскольку в этом случае IPsec SA поддерживаются между NSF без участия контроллера I2NSF, а вместо этого контроллер I2NSF просто обеспечивает начальную настройку (записи IKEv2, PAD, SPD). Предложены иные решения, такие как Controller-IKE [IPSECME-CONTROLLER-IKE], где NSF предоставляют свои открытые ключи DH контроллеру I2NSF, а тот распространяет их всем партнёрам. Партнёры могут вычислить парные секреты для каждого из своих партнёров и сообщения между NSF не нужны. Механизм смены ключей дополнительно описан в [IPSECME-CONTROLLER-IKE].

С точки зрения безопасности вариант с IKE обеспечивает лучшую защиту, нежели вариант без (см. 7. Вопросы безопасности). Основная причина этого заключается в том, что сеансовые ключи создают NSF, а не контроллер I2NSF.

4.1. Процесс смены ключей

Смена ключей IPsec SA является важной частью управления IPsec SA. С моделями данных YANG, заданными здесь, контроллер I2NSF может настраивать параметры процесса смены ключей (вариант с IKE) или выполнять этот процесс (вариант без IKE).

Для варианта с IKE процесс смены ключей выполняет IKEv2, следуя данным из SPD и SAD (например, на основе срока действия IPsec SA, заданного контроллером I2NSF с использованием модели YANG из этого документа). Поэтому соединения IPsec будут активны, пока пользователь I2NSF не потребует иного или контроллер I2NSF не обнаружит что-либо неверное.

В варианте без IKE контроллер I2NSF **должен** заботиться о смене ключей. Когда срок действия IPsec SA истекает, контроллер **должен** создать новую IPsec SA и **может** удалить старую (например, если срок действия старой IPsec SA не задан). Процесс смены ключей начинается с получением контроллером I2NSF уведомления sadb-expire или по инициативе самого контроллера I2NSF на основе сведений о сроке действия данных состояния, полученных от NSF. Способ реализации контроллером I2NSF алгоритма смены ключей выходит за рамки этого документа. Тем не менее, пример возможной смены ключей представлен в D.2. Пример смены ключей в варианте без IKE.

4.2. Потеря состояния NSF

При перезапуске NSF теряется состояние IPsec (затрагиваемое NSF). По умолчанию контроллер I2NSF может считать, что было потеряно все состояние, поэтому он будет передавать NSF сведения IKEv2, SPD и PAD в варианте с IKE, SPD и SAD в варианте без IKE. В обоих случаях контроллер I2NSF знает о затронутом NSF (например, разорвано соединение NETCONF/TCP с NSF, контроллер I2NSF получил уведомление sadb-bad-spi от NSF и т. п.). Кроме того, контроллер I2NSF хранит список NSF, имеющих IPsec SA с затронутым NSF, поэтому он знает затронутые IPsec SA.

В варианте с IKE контроллеру I2NSF может потребоваться настройка для затронутого NSF новых сведений IKEv2, SPD и PAD. Как вариант, конфигурацию IKEv2 **можно** сделать неизменной при перезапуске NSF без ущерба для безопасности путём записи в хранилище стартовой конфигурации NSF. В этом случае при каждой перезагрузке NSF будет применяться одна и та же конфигурация, что позволит избежать обращения к контроллеру I2NSF. Контроллеру может потребоваться передача новых (например, свежий ключ PSK для аутентификации) элементам NSF, у которых были IKEv2 SA и IPsec SA с затронутым NSF.

В варианте без IKE контроллеру I2NSF **следует** удалить старые IPsec SA в неотказавших узлах, связанных с затронутым NSF. После перезапуска узла контроллер I2NSF **должен** выполнить действия по восстановления защищённой IPsec связи между отказавшим узлом и другими узлами, имеющими IPsec SA с затронутым NSF. Способ реализации контроллером I2NSF Controller алгоритма обработки возможной потери состояния NSF выходит за рамки документа. Тем не менее, пример обработки дан в D.3. Пример контроля потери состояния NSF в варианте без IKE.

4.3. Работа через NAT

В случае с IKE уже предоставляется механизм IKEv2 для обнаружения размещения одного или обоих партнёров за системой NAT. В таких случаях нужна инкапсуляция UDP или TCP для пакетов инкапсуляции защищённых данных (Encapsulating Security Payload или ESP) [RFC3948] [RFC8229]. Отметим, что транспортный режим IPsec **недопустимо** применять в этой спецификации, если требуется NAT.

В случае без IKE у NSF нет поддержки от реализации IKEv2 для обнаружения работы через NAT. Если у NSF нет иного механизма обнаружения таких ситуаций, контроллеру I2NSF **следует** реализовать свой механизм. Парадигма SDN обычно предполагает, что у контроллера I2NSF имеется представление управляемой сети. Это представление создаётся путём запроса информации от управляемых NSF или использования сведений, выталкиваемых NSF контроллеру I2NSF. На основе этих данных контроллер I2NSF **может** предсказать наличие NAT между двумя хостами и применить требуемые к обоим NSF правила в дополнение к инкапсуляции UDP или TCP для пакетов ESP [RFC3948] [RFC8229]. Интерфейс обнаружения NSF, расположенных за NAT, выходит за рамки этого документа.

Если у контроллера I2NSF нет механизма обнаружения работы хоста через NAT, **должен** применяться вариант с IKE.

4.4. Регистрация и обнаружение NSF

Регистрацией NSF называют процесс предоставления контроллеру I2NSF сведений о действительной функции NSF, таких как сертификат, адрес IP и т. п. Эта информация включается в список NSF, находящихся под контролем. В этом документе предполагается, что в обоих вариантах, для работы NSF в системе эта функция **должна** быть зарегистрирована в контроллере I2NSF. Таким образом при запуске NSF и организации соединения с контроллером I2NSF уже известно о пригодности NSF для присоединения к системе.

В процессе регистрации или при соединении NSF с контроллером I2NSF контроллер **должен** обнаружить некоторые возможности NSF, такие как поддерживаемые криптонаборы, метод аутентификации, работа с IKE или без IKE и т. п. Процессы регистрации и обнаружения выходят за рамки этого документа.

5. Модели YANG для данных конфигурации

Для поддержки вариантов с IKE и без IKE представлены модели с различными параметрами и значениями, которые должны быть настроены для управления IPsec SA. В частности для варианта с IKE требуется моделировать параметры конфигурации IKEv2, SPD и PAD, а для варианта без IKE требуются модели данных YANG для SPD и SAD. Ниже определены три модуля: ietf-i2nsf-ikec (параграф 5.1, общий для обоих вариантов), ietf-i2nsf-ike (параграф 5.2 для варианта с IKE), ietf-i2nsf-ikeless (параграф 5.3 для варианта без IKE). Поскольку модуль ietf-i2nsf-ikec включает лишь общие с другими модулями определения типов и группировки, модули ietf-i2nsf-ike и ietf-i2nsf-ikeless даны упрощенно.

5.1. Модуль ietf-i2nsf-ikec

5.1.1. Обзор модели данных

Модуль ietf-i2nsf-ikec определяет лишь типы данных (typedef) и группировки, применяемые другими модулями.

5.1.2. Модуль YANG

Этот модуль включает нормативные ссылки на [RFC3947], [RFC4301], [RFC4303], [RFC8174], [RFC8221], [RFC3948], [RFC8229], [RFC6991], [IANA-Protocols-Number], [IKEv2-Parameters], [IKEv2-Transform-Type-1] и [IKEv2-Transform-Type-3].

```
<CODE BEGINS> file "ietf-i2nsf-ikec@2021-07-14.yang"
module ietf-i2nsf-ikec {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-i2nsf-ikec";
  prefix nsfikec;

  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Data Types.";
  }

  organization
    "IETF I2NSF Working Group";
  contact
    "WG Web:  <https://datatracker.ietf.org/wg/i2nsf/>
     WG List: <mailto:i2nsf@ietf.org>

     Author: Rafael Marin-Lopez
               <mailto:rafa@um.es>

     Author: Gabriel Lopez-Millan
               <mailto:gabilm@um.es>

     Author: Fernando Pereniguez-Garcia
               <mailto:fernando.pereniguez@cud.upct.es>";
  description
    "Базовая модель данных для вариантов с IKE и без IKE, заданных
     службой защиты потоков IPsec на основе SDN.

     Ключевые слова ДОЛЖНО, НЕДОПУСТИМО, ТРЕБУЕТСЯ, НУЖНО, НЕ НУЖНО,
     СЛЕДУЕТ, НЕ СЛЕДУЕТ, РЕКОМЕНДУЕТСЯ, НЕ РЕКОМЕНДУЕТСЯ, МОЖНО,
     НЕОБЯЗАТЕЛЬНО в этом документе трактуются в соответствии с
     BCP 14 (RFC 2119) (RFC 8174) тогда и только тогда, когда они
     указаны заглавными буквами, как показано здесь.

     Авторские права (Copyright (c) 2021) принадлежат IETF Trust и
     лицам, указанным как авторы. Все права защищены.

     Распространение и применение модуля в исходной или двоичной
     форме с изменениями или без таковых разрешено в соответствии с
     лицензией Simplified BSD License, изложенной в параграфе 4.c
```

```
      IETF Trust's Legal Provisions Relating to IETF Documents
      (https://trustee.ietf.org/license-info).

      Эта версия модуля YANG является частью RFC 9061, где правовые
      аспекты приведены более полно.";

    revision 2021-07-14 {
      description
        "Исходный выпуск.";
      reference
        "RFC 9061: A YANG Data Model for IPsec Flow Protection
                   Based on Software-Defined Networking (SDN).";
    }

    typedef encr-alg-t {
      type uint16;
      description
        "Алгоритм шифрования указывается 16-битовым номером из реестра
         IANA. Приемлемые значения ДОЛЖНЫ соответствовать уровню
         требований для алгоритмов шифрования ESP и IKEv2.";
      reference
        "IANA: Internet Key Exchange Version 2 (IKEv2) Parameters,
               IKEv2 Transform Attribute Types, Transform Type 1 -
               Encryption Algorithm Transform IDs
         RFC 8221: Cryptographic Algorithm Implementation Requirements
                   and Usage Guidance for Encapsulating Security
                   Payload (ESP) and Authentication Header (AH)
         RFC 8247: Algorithm Implementation Requirements and Usage
                   Guidance for the Internet Key Exchange Protocol
                   Version 2 (IKEv2).";
    }

    typedef intr-alg-t {
      type uint16;
      description
        "Алгоритм защиты целостности указывается 16-битовым номером из
         реестра IANA. Приемлемые значения ДОЛЖНЫ соответствовать
         уровню требований защиты целостности для ESP и IKEv2.";
      reference
        "IANA: Internet Key Exchange Version 2 (IKEv2) Parameters,
               IKEv2 Transform Attribute Types, Transform Type 3 -
               Integrity Algorithm Transform IDs
         RFC 8221: Cryptographic Algorithm Implementation
                   Requirements and Usage Guidance for Encapsulating
                   Security Payload (ESP) and Authentication Header
                   (AH)
         RFC 8247: Algorithm Implementation Requirements and Usage
                   Guidance for the Internet Key Exchange Protocol
                   Version 2 (IKEv2).";
    }

    typedef ipsec-mode {
      type enumeration {
        enum transport {
          description
            "Транспортный режим IPsec. NAT не поддерживается.";
        }
        enum tunnel {
          description
            "Туннельный режим IPsec.";
        }
      }
      description
        "Определение типа для режима IPsec - transport или tunnel.";
      reference
        "RFC 4301: Security Architecture for the Internet Protocol,
                   Section 3.2.";
    }

    typedef esp-encap {
      type enumeration {
        enum espintcp {
          description
            "ESP с инкапсуляцией TCP.";
          reference
            "RFC 8229: TCP Encapsulation of IKE and IPsec Packets.";
        }
        enum espinudp {
          description
            "ESP с инкапсуляцией UDP.";
          reference
            "RFC 3948: UDP Encapsulation of IPsec ESP Packets.";
        }
        enum none {
          description
            "Без инкапсуляции ESP.";
```

```
      }
    }
    description
      "Типы инкапсуляции ESP при возможном наличии трансляции NAT
       между двумя NSF.";
    reference
      "RFC 8229: TCP Encapsulation of IKE and IPsec Packets
       RFC 3948: UDP Encapsulation of IPsec ESP Packets.";
  }

  typedef ipsec-protocol-params {
    type enumeration {
      enum esp {
        description
          "Протокол IPsec ESP.";
      }
    }
    description
      "Поддерживается только ESP, но потом возможно расширение.";
    reference
      "RFC 4303: IP Encapsulating Security Payload (ESP).";
  }

  typedef lifetime-action {
    type enumeration {
      enum terminate-clear {
        description
          "Завершает все IPsec SA и пропускает пакеты.";
      }
      enum terminate-hold {
        description
          "Завершает IPsec SA и отбрасывает пакеты.";
      }
      enum replace {
        description
          "Заменяет IPsec SA со сменой ключей.";
      }
    }
    description
      "При завершении срока действия IPsec SA нужно принимать меры.
       Имеется 3 варианта: terminate-clear, terminate-hold, replace";
    reference
      "RFC 4301: Security Architecture for the Internet Protocol,
                 Section 4.5.";
  }

  typedef ipsec-traffic-direction {
    type enumeration {
      enum inbound {
        description
          "Входящий трафик.";
      }
      enum outbound {
        description
          "Исходящий трафик.";
      }
    }
    description
      "Направление трафика IPsec определяется как inbound или
       outbound. С точки зрения NSF входной и выходной трафик
       определяются в соответствии с параграфом 3.1 в RFC 4301.";
    reference
      "RFC 4301: Security Architecture for the Internet Protocol,
                 Section 3.1.";
  }

  typedef ipsec-spd-action {
    type enumeration {
      enum protect {
        description
          "Защита трафика с применением IPsec.";
      }
      enum bypass {
        description
          "Передача трафика в обход без защиты IPsec.";
      }
      enum discard {
        description
          "Отбрасывание трафика пакетов IP.";
      }
    }
    description
      "Действие при соответствии трафика политике IPsec. В RFC 4301
       задано 3 варианта BYPASS, PROTECT, DISCARD.";
    reference
      "RFC 4301: Security Architecture for the Internet Protocol,
```

```
                Section 4.4.1.";
    }

    typedef ipsec-inner-protocol {
      type union {
        type uint8;
        type enumeration {
          enum any {
            value 256;
            description
              "Любой номер протокола IP.";
          }
        }
      }
      default "any";
      description
        "защита IPsec может быть применена к конкретному трафику IP и
         L4 (TCP, UDP, SCTP и т. п.) или ЛЮБОМУ протоколу в данных
         (payload) пакета IP. Номер протокола указывается типом uint8
         или ЛЮБЫМ определением перечисляемого типа со значением 256
         для указания номера протокола. Отметим, что для IPv6 протокол
         в данных пакета IP указывает поле Next Header заголовка IPv6";
      reference
        "RFC 4301: Security Architecture for the Internet Protocol,
                Section 4.4.1.1
         IANA: Protocol Numbers.";
    }

    grouping encap {
      description
        "Эта группа узлов позволяет определить тип инкапсуляции при
         необходимости работать через NAT и указании порта.";
      leaf espencap {
        type esp-encap;
        default "none";
        description
          "ESP в TCP, ESP с UDP, ESP в TLS.";
      }
      leaf sport {
        type inet:port-number;
        default "4500";
        description
          "Номер порта-источника при инкапсуляции.";
      }
      leaf dport {
        type inet:port-number;
        default "4500";
        description
          "Номер порта-получателя при инкапсуляции.";
      }
      leaf-list oaddr {
        type inet:ip-address;
        description
          "Если нужно, этот лист-список содержит исходный адрес,
           использованный до трансляции пакета в NAT.";
      }
      reference
        "RFC 3947: Negotiation of NAT-Traversal in the IKE
         RFC 8229: TCP Encapsulation of IKE and IPsec Packets.";
    }

    grouping lifetime {
      description
        "Значения срока действия ограничены для IPsec SA.";
      leaf time {
        type uint32;
        units "seconds";
        default "0";
        description
          "Число секунд с момент добавления IPsec SA. Например, 180
           означает, что прошло 180 секунд с момента добавления
           IPsec SA. Значение 0 подразумевает бесконечность.";
      }
      leaf bytes {
        type uint64;
        default "0";
        description
          "Срок действия IPsec SA завершается после обработки IPsec SA
           числа байтов, заданного этим листом, и СЛЕДУЕТ поменять
           ключи. Значение 0 подразумевает бесконечность.";
      }
      leaf packets {
        type uint32;
        default "0";
        description
          "Срок действия IPsec SA завершается после обработки IPsec SA
```

```
          числа пакетов, заданного этим листом, и СЛЕДУЕТ поменять
          ключи. Значение 0 подразумевает бесконечность.";
      }
      leaf idle {
        type uint32;
        units "seconds";
        default "0";
        description
          "NSF расходует ресурсы системы на сохранение IPsec SA. Для
           бездействующей IPsec SA это будет напрасный расход. Если
           IPsec SA бездействует в течение указанного числа секунд, её
           СЛЕДУЕТ удалить. Значение 0 подразумевает бесконечность.";
      }
      reference
        "RFC 4301: Security Architecture for the Internet Protocol,
                   Section 4.4.2.1.";
    }

    grouping port-range {
      description
        "Группировка задаёт диапазон портов, как указано в RFC 4301,
         например, 1500 (начальный номер)-1600 (конечный номер).
         Диапазон портов применяется в селекторе трафика.";
      leaf start {
        type inet:port-number;
        description
          "Начальный номер порта.";
      }
      leaf end {
        type inet:port-number;
        must '. >= ../start' {
          error-message
            "Конечный номер порта ДОЛЖЕН быть не меньше начального.";
        }
        description
          "Конечный номер порта. Для указания одного порта указываются
           одинаковое начальное и конечное значения.";
      }
      reference
        "RFC 4301: Security Architecture for the Internet Protocol,
                   Section 4.4.1.2.";
    }

    grouping tunnel-grouping {
      description
        "Параметры, требуемые для определения конечных точек туннеля
         IP, когда IPsec SA требует туннельный режим. Туннель задают
         адреса IP локальной и удалённой конечных точек.";
      leaf local {
        type inet:ip-address;
        mandatory true;
        description
          "Адрес IP локальной конечной точки туннеля.";
      }
      leaf remote {
        type inet:ip-address;
        mandatory true;
        description
          "Адрес IP удалённой конечной точки туннеля.";
      }
      leaf df-bit {
        type enumeration {
          enum clear {
            description
              "Сбрасывать флаг DF во внешнем заголовке. Принято по
               умолчанию.";
          }
          enum set {
            description
              "Устанавливать флаг DF во внешнем заголовке.";
          }
          enum copy {
            description
              "Копировать флаг DF во внешний заголовок.";
          }
        }
        default "clear";
        description
          "Разрешает задавать бит DF при инкапсуляции трафика в
           туннельном режиме IPsec. RFC 4301 описывает 3 варианта
           обработки флага DF при туннельной инкапсуляции: сброс,
           установка, копирование из внутреннего заголовка IP. Лист
           ДОЛЖЕН игнорироваться, если локальный и удалённый адреса
           IP являются адресами IPv6.";
        reference
          "RFC 4301: Security Architecture for the Internet Protocol,
```

```
                Section 8.1.";
      }
      leaf bypass-dscp {
        type boolean;
        default "true";
        description
          "Значение true задаёт копирование кода DSCP из внутреннего
           заголовка во внешний. Значение false указывает отображение
           значений DSCP в соответствии с ../dscp-mapping.";
        reference
          "RFC 4301: Security Architecture for the Internet Protocol,
                     Section 4.4.1.2.";
      }
      list dscp-mapping {
        must '../bypass-dscp = "false"';
        key "id";
        ordered-by user;
        leaf id {
          type uint8;
          description
            "Индекс списка отображений.";
        }
        leaf inner-dscp {
          type inet:dscp;
          description
            "Значение DSCP во внутреннем пакете IP. Если лист не задан,
             предполагается ЛЮБОЕ внутреннее значение DSCP.";
        }
        leaf outer-dscp {
          type inet:dscp;
          default "0";
          description
            "Значение DSCP во внешнем заголовке IP.";
        }
        description
          "Список, представляющий массив отображений внутреннего кода
           DSCP во внешний при установке bypass-dscp false. Для задания
           принятого по умолчанию отображения, когда внутреннее
           значение не соответствует ни одному узлу списка, в конец
           списка включается новый элемент, где лист inner-dscp не
           задан (ANY), а outer-dscp содержит отображаемое значение.
           Если outer-dscp, не указывает значения, предполагается 0.";
        reference
          "RFC 4301: Security Architecture for the Internet Protocol,
                     Section 4.4.1.2 and Appendix C.";
      }
    }

    grouping selector-grouping {
      description
        "Группировка определения селектора трафика, используемого в
         правилах IPsec policies и IPsec SA.";
      leaf local-prefix {
        type inet:ip-prefix;
        mandatory true;
        description
          "Префикс локального адреса IP.";
      }
      leaf remote-prefix {
        type inet:ip-prefix;
        mandatory true;
        description
          "Префикс удалённого адреса IP.";
      }
      leaf inner-protocol {
        type ipsec-inner-protocol;
        default "any";
        description
          "Внутренний протокол, который будет защищаться IPsec.";
      }
      list local-ports {
        key "start end";
        uses port-range;
        description
          "Список локальных портов. Для внутреннего протокола ICMP это
           16-битовое значение представляет код и тип. Если этот список
           не задан, предполагается, что start и end имеют значения 0
           (любой порт).";
      }
      list remote-ports {
        key "start end";
        uses port-range;
        description
          "Список удалённых портов. Когда вышележащим протоколом
           является ICMP, это 16-битовое значение представляет код и
           тип. Если этот список не задан, предполагается, что start и
```

```
          end имеют значения 0(любой порт).";
      }
      reference
        "RFC 4301: Security Architecture for the Internet Protocol,
                   Section 4.4.1.2.";
    }

    grouping ipsec-policy-grouping {
      description
        "данные конфигурации для записи IPsec SPD.";
      leaf anti-replay-window-size {
        type uint32;
        default "64";
        description
          "Для установки размера окна anti-replay. По умолчанию 64,
           в соответствии с RFC 4303.";
        reference
          "RFC 4303: IP Encapsulating Security Payload (ESP),
                     Section 3.4.3.";
      }
      container traffic-selector {
        description
          "Пакеты выбираются для обработки по селекторам трафика,
           которые указывают IP и данные из внутреннего заголовка.";
        uses selector-grouping;
        reference
          "RFC 4301: Security Architecture for the Internet Protocol,
                     Section 4.4.4.1.";
      }
      container processing-info {
        description
          "Обработка SPD. Если требуемым действием является защита,
           здесь содержатся сведения, требуемые для обработки.";
        leaf action {
          type ipsec-spd-action;
          default "discard";
          description
            "При обходе или отбрасывании контейнер ipsec-sa-cfg пуст.";
        }
        container ipsec-sa-cfg {
          when "../action = 'protect'";
          description
            "Конфигурация IPsec SA, включаемая в запись SPD.";
          leaf pfp-flag {
            type boolean;
            default "false";
            description
              "Каждый селектор имеет флаг заполнения из пакета
               (Populate From Packet или PFP). Установленный для
               селектора X флаг указывает, что создаваемой IPsec SA
               следует брать своё значение (локальный и удалённый
               адреса IP, Next Layer Protocol и т. п.) для X из
               значений в пакете. В иных случаях IPsec SA следует
               брать значения для X из записи в SPD.";
          }
          leaf ext-seq-num {
            type boolean;
            default "false";
            description
              "Значение true указывает, что данные IPsec SA применяет
               расширенные порядковые номера (64 бита). Значение false
               указывает применение обычных 32-битовых номеров.";
          }
          leaf seq-overflow {
            type boolean;
            default "false";
            description
              "Флаг, указывающий, должно ли переполнение счётчика
               порядковых номеров препятствовать дальнейшей передаче
               пакетов в IPsec SA (false) и, следовательно, нужна смена
               ключей, или разрешён переход счётчика через максимум
               (true). При аутентифицированном шифровании со связанными
               данными (Authenticated Encryption with Associated Data
               или AEAD) (лист esp-algorithms/encryption/algorithm-type)
               этот флаг ДОЛЖЕН иметь значение false. Установка значения
               true настоятельно не рекомендуется.";
          }
          leaf stateful-frag-check {
            type boolean;
            default "false";
            description
              "Указывает применяется (true) или нет (false) проверка
               фрагментов с учётом состояния для создаваемой IPsec SA";
          }
          leaf mode {
            type ipsec-mode;
```

```
            default "transport";
            description
              "Туннельный или транспортный режим IPsec SA.";
          }
          leaf protocol-parameters {
            type ipsec-protocol-params;
            default "esp";
            description
              "Протокол защиты IPsec SA. Сейчас поддерживается лишь
               ESP, но возможно расширение в будущем.";
          }
          container esp-algorithms {
            when "../protocol-parameters = 'esp'";
            description
              "Конфигурация параметров и алгоритмов ESP.";
            leaf-list integrity {
              type intr-alg-t;
              default "0";
              ordered-by user;
              description
                "Конфигурация аутентификации ESP на основе заданного
                 алгоритма защиты целостности. При шифровании AEAD узел
                 integrity не используется.";
              reference
                "RFC 4303: IP Encapsulating Security Payload (ESP),
                           Section 3.2.";
            }
            list encryption {
              key "id";
              ordered-by user;
              leaf id {
                type uint16;
                description
                  "Идентификатор, однозначно указывающий каждую запись
                   списка, т. е. алгоритм шифрования и размер ключа
                   (если размер нужен).";
              }
              leaf algorithm-type {
                type encr-alg-t;
                default "20";
                description
                  "По умолчанию 20 (ENCR_AES_GCM_16).";
              }
              leaf key-length {
                type uint16;
                default "128";
                description
                  "По умолчанию размер ключа составляет 128 битов.";
              }
              description
                "Алгоритмы шифрования или AEAD для IPsec SA. Список
                 упорядочивается по снижению приоритета. При пустом
                 списке шифрование не применяется (алгоритм NULL).";
              reference
                "RFC 4303: IP Encapsulating Security Payload (ESP),
                           Section 3.2.";
            }
            leaf tfc-pad {
              type boolean;
              default "false";
              description
                "Значение true разрешает заполнение для
                 конфиденциальности трафика потока (Traffic Flow
                 Confidentiality или TFC) при шифровании ESP, false
                 запрещает заполнение.";
              reference
                "RFC 4303: IP Encapsulating Security Payload (ESP),
                           Section 2.7.";
            }
            reference
              "RFC 4303: IP Encapsulating Security Payload (ESP).";
          }
          container tunnel {
            when "../mode = 'tunnel'";
            uses tunnel-grouping;
            description
              "Определение конечных точек туннеля IPsec.";
          }
        }
        reference
          "RFC 4301: Security Architecture for the Internet Protocol,
                     Section 4.4.1.2.";
      }
    }
  }
  <CODE ENDS>
```

5.2. Модуль ietf-i2nsf-ike

В этом параграфе описан модуль YANG для варианта с IKE.

5.2.1. Обзор модели данных

Модель, связанная с IKEv2, была создана на основе стандарта IKEv2 [RFC7296] и наблюдения за некоторыми реализациями с открытым кодом, такие как strongSwan [strongswan] и Libreswan [libreswan].

Модель PAD создана на основе параграфа 4.4.3 в [RFC4301]. Отметим, что многие реализации встраивают настройку PAD как часть настройки IKEv2.

Модель SPD в основном получена на основе параграфа 4.4.1 и Приложения D к [RFC4301].

Модель данных YANG для варианта с IKE задана в модуле ietf-i2nsf-ike, структура которого показана ниже с использованием нотации деревьев YANG из [RFC8340].

```
module: ietf-i2nsf-ike
  +--rw ipsec-ike
    +--rw pad
    |  +--rw pad-entry* [name]
    |     +--rw name                            string
    |     +--rw (identity)
    |     |  +--:(ipv4-address)
    |     |  |  +--rw ipv4-address?            inet:ipv4-address
    |     |  +--:(ipv6-address)
    |     |  |  +--rw ipv6-address?            inet:ipv6-address
    |     |  +--:(fqdn-string)
    |     |  |  +--rw fqdn-string?             inet:domain-name
    |     |  +--:(rfc822-address-string)
    |     |  |  +--rw rfc822-address-string?   string
    |     |  +--:(dnx509)
    |     |  |  +--rw dnx509?                  binary
    |     |  +--:(gnx509)
    |     |  |  +--rw gnx509?                  binary
    |     |  +--:(id-key)
    |     |  |  +--rw id-key?                  binary
    |     |  +--:(id-null)
    |     |     +--rw id-null?                empty
    |     +--rw auth-protocol?                auth-protocol-type
    |     +--rw peer-authentication
    |        +--rw auth-method?         auth-method-type
    |        +--rw eap-method
    |        |  +--rw eap-type    uint64
    |        +--rw pre-shared
    |        |  +--rw secret?   yang:hex-string
    |        +--rw digital-signature
    |           +--rw ds-algorithm?           uint8
    |           +--rw (public-key)?
    |           |  +--:(raw-public-key)
    |           |  |  +--rw raw-public-key?   binary
    |           |  +--:(cert-data)
    |           |     +--rw cert-data?        binary
    |           +--rw private-key?            binary
    |           +--rw ca-data*                binary
    |           +--rw crl-data?               binary
    |           +--rw crl-uri?                inet:uri
    |           +--rw ocsp-uri?               inet:uri
    +--rw conn-entry* [name]
    |  +--rw name                             string
    |  +--rw autostartup?                     autostartup-type
    |  +--rw initial-contact?                 boolean
    |  +--rw version?                         auth-protocol-type
    |  +--rw fragmentation
    |  |  +--rw enabled?   boolean
    |  |  +--rw mtu?      uint16
    |  +--rw ike-sa-lifetime-soft
    |  |  +--rw rekey-time?    uint32
    |  |  +--rw reauth-time?   uint32
    |  +--rw ike-sa-lifetime-hard
    |  |  +--rw over-time?   uint32
    |  +--rw ike-sa-intr-alg*  nsfikec:intr-alg-t
    |  +--rw ike-sa-encr-alg* [id]
    |  |  +--rw id                uint16
    |  |  +--rw algorithm-type?   nsfikec:encr-alg-t
    |  |  +--rw key-length?       uint16
    |  +--rw dh-group?                              fs-group
    |  +--rw half-open-ike-sa-timer?                uint32
    |  +--rw half-open-ike-sa-cookie-threshold?     uint32
    |  +--rw local
    |  |  +--rw local-pad-entry-name    string
    |  +--rw remote
    |  |  +--rw remote-pad-entry-name    string
    |  +--rw encapsulation-type
    |  |  +--rw espencap?   esp-encap
    |  |  +--rw sport?      inet:port-number
    |  |  +--rw dport?      inet:port-number
```

```
|  |  +--rw oaddr*      inet:ip-address
|  +--rw spd
|  |  +--rw spd-entry* [name]
|  |    +--rw name                    string
|  |    +--rw ipsec-policy-config
|  |      +--rw anti-replay-window-size?   uint32
|  |      +--rw traffic-selector
|  |      |  +--rw local-prefix      inet:ip-prefix
|  |      |  +--rw remote-prefix     inet:ip-prefix
|  |      |  +--rw inner-protocol?   ipsec-inner-protocol
|  |      |  +--rw local-ports* [start end]
|  |      |  |  +--rw start    inet:port-number
|  |      |  |  +--rw end      inet:port-number
|  |      |  +--rw remote-ports* [start end]
|  |      |     +--rw start    inet:port-number
|  |      |     +--rw end      inet:port-number
|  |      +--rw processing-info
|  |        +--rw action?         ipsec-spd-action
|  |        +--rw ipsec-sa-cfg
|  |         +--rw pfp-flag?              boolean
|  |         +--rw ext-seq-num?           boolean
|  |         +--rw seq-overflow?          boolean
|  |         +--rw stateful-frag-check?   boolean
|  |         +--rw mode?                  ipsec-mode
|  |         +--rw protocol-parameters? ipsec-protocol-params
|  |              +--rw esp-algorithms
|  |              |  +--rw integrity*    intr-alg-t
|  |              |  +--rw encryption* [id]
|  |              |  |  +--rw id                 uint16
|  |              |  |  +--rw algorithm-type?   encr-alg-t
|  |              |  |  +--rw key-length?       uint16
|  |              |  +--rw tfc-pad?      boolean
|  |              +--rw tunnel
|  |                 +--rw local           inet:ip-address
|  |                 +--rw remote          inet:ip-address
|  |                 +--rw df-bit?         enumeration
|  |                 +--rw bypass-dscp?    boolean
|  |                 +--rw dscp-mapping* [id]
|  |                    +--rw id            uint8
|  |                    +--rw inner-dscp?   inet:dscp
|  |                    +--rw outer-dscp?   inet:dscp
|  +--rw child-sa-info
|  |  +--rw fs-groups*                fs-group
|  |  +--rw child-sa-lifetime-soft
|  |  |  +--rw time?      uint32
|  |  |  +--rw bytes?     yang:counter64
|  |  |  +--rw packets?   uint32
|  |  |  +--rw idle?      uint32
|  |  |  +--rw action?    nsfikec:lifetime-action
|  |  +--rw child-sa-lifetime-hard
|  |     +--rw time?      uint32
|  |     +--rw bytes?     yang:counter64
|  |     +--rw packets?   uint32
|  |     +--rw idle?      uint32
|  +--ro state
|     +--ro initiator?             boolean
|     +--ro initiator-ikesa-spi?   ike-spi
|     +--ro responder-ikesa-spi?   ike-spi
|     +--ro nat-local?             boolean
|     +--ro nat-remote?            boolean
|     +--ro encapsulation-type
|     |  +--ro espencap?   esp-encap
|     |  +--ro sport?      inet:port-number
|     |  +--ro dport?      inet:port-number
|     |  +--ro oaddr*      inet:ip-address
|     +--ro established?           uint64
|     +--ro current-rekey-time?    uint64
|     +--ro current-reauth-time?   uint64
+--ro number-ike-sas
    +--ro total?               yang:gauge64
    +--ro half-open?           yang:gauge64
    +--ro half-open-cookies?   yang:gauge64
```

Модель данных YANG содержит уникальный контейнер ipsec-ike, определённый ниже. Он включает контейнер pad, служащий для настройки базы данных PAD с данными аутентификации для локальных и удалённых партнёров (NSF). Точнее, контейнер содержит список записей, каждая из которых указывает отождествление, метод аутентификации и свидетельства, которые партнёр (локальный или удалённый) будет применять. Таким образом, каждая запись содержит указанные сведения локального или удалённого элемента NSF. В результате контроллер I2NF может сохранять эти сведения.

Список conn-entry со сведениями о различных соединениях IKE, которые узел может поддерживать с другими партнёрами. Каждая запись для соединения состоит из большого числа параметров для настройки различных аспектов определённого соединения IKE между двумя партнёрами, аутентификационные данные локального и удалённого партнёра, конфигурацию IKE SA (мягкий и строгий срок действия, криптоалгоритмы и т. п.), список правил IPsec, описывающих тип защищаемого трафика (локальные и удалённые подсети и порты), способ защиты (ESP, туннельный

или транспортный режим, криптоалгоритмы и т. п.), конфигурация Child SA (мягкий и строгий срок действия), сведения о состоянии соединения IKE (SPI, использование NAT, текущее время завершения срока действия и т. п.).

Контейнер ipsec-ike объявляет контейнер number-ike-sas для задания сообщаемых программами IKE сведений о состоянии, относящихся к числу организованных соединений IKE.

5.2.2. Пример использования

В Приложении A приведён пример конфигурации варианта NSF с IKE в туннельном режиме (шлюз-шлюз) с аутентификацией NSF на основе сертификатов X.509.

5.2.3. Модуль YANG

Этот модуль YANG включает нормативные ссылки на [RFC5280], [RFC4301], [RFC5915], [RFC6991], [RFC7296], [RFC7383], [RFC7427], [RFC7619], [RFC8017], [ITU-T.X.690], [RFC5322], [RFC8229], [RFC8174], [RFC6960], [IKEv2-Auth-Method], [IKEv2-Transform-Type-4], [IKEv2-Parameters], [IANA-Method-Type].

```
<CODE BEGINS> file "ietf-i2nsf-ike@2021-07-14.yang"
module ietf-i2nsf-ike {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-i2nsf-ike";
  prefix nsfike;

  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Data Types.";
  }
  import ietf-yang-types {
    prefix yang;
    reference
      "RFC 6991: Common YANG Data Types.";
  }
  import ietf-i2nsf-ikec {
    prefix nsfikec;
    reference
      "RFC 9061: A YANG Data Model for IPsec Flow Protection
                 Based on Software-Defined Networking (SDN).";
  }
  import ietf-netconf-acm {
    prefix nacm;
    reference
      "RFC 8341: Network Configuration Access Control
                 Model.";
  }

  organization
    "IETF I2NSF Working Group";
  contact
    "WG Web:  <https://datatracker.ietf.org/wg/i2nsf/>
     WG List: <mailto:i2nsf@ietf.org>

     Author: Rafael Marin-Lopez
               <mailto:rafa@um.es>

     Author: Gabriel Lopez-Millan
               <mailto:gabilm@um.es>

     Author: Fernando Pereniguez-Garcia
               <mailto:fernando.pereniguez@cud.upct.es>
    ";
  description
    "Этот модуль содержит модель для варианта IPsec IKE основанной на
     SDN службы защиты потоков IPsec.

     Ключевые слова ДОЛЖНО, НЕДОПУСТИМО, ТРЕБУЕТСЯ, НУЖНО, НЕ НУЖНО,
     СЛЕДУЕТ, НЕ СЛЕДУЕТ, РЕКОМЕНДУЕТСЯ, НЕ РЕКОМЕНДУЕТСЯ, МОЖНО,
     НЕОБЯЗАТЕЛЬНО в этом документе трактуются в соответствии с
     BCP 14 (RFC 2119) (RFC 8174) тогда и только тогда, когда они
     указаны заглавными буквами, как показано здесь.

     Авторские права (Copyright (c) 2021) принадлежат IETF Trust и
     лицам, указанным как авторы. Все права защищены.

     Распространение и применение модуля в исходной или двоичной
     форме с изменениями или без таковых разрешено в соответствии с
     лицензией Simplified BSD License, изложенной в параграфе 4.c
     IETF Trust's Legal Provisions Relating to IETF Documents
     (https://trustee.ietf.org/license-info).

     Эта версия модуля YANG является частью RFC 9061, где правовые
     аспекты приведены более полно.";

  revision 2021-07-14 {
    description
      "Исходный выпуск.";
```

```
    reference
      "RFC 9061: A YANG Data Model for IPsec Flow Protection
                 Based on Software-Defined Networking (SDN).";
  }

  typedef ike-spi {
    type uint64 {
      range "0..max";
    }
    description
      "Индекс параметров защиты (SPI) IKE SA.";
    reference
      "RFC 7296: Internet Key Exchange Protocol Version 2
                 (IKEv2), Section 2.6.";
  }

  typedef autostartup-type {
    type enumeration {
      enum add {
        description
          "Конфигурация IKE/IPsec загружается в реализацию IKE, но
           IKE/IPsec SA не запускается.";
      }
      enum on-demand {
        description
          "Конфигурация IKE/IPsec загружается в реализацию IKE.
           Павила IPsec переносятся в NSF, но IPsec SA не создаются
           сразу же. Реализация IKE будет согласовывать IPsec SA по
           мере необходимости (через уведомление ACQUIRE).";
      }
      enum start {
        description
          "Конфигурация IKE/IPsec загружается, переносится в ядро
           NSF и основанные на IKEv2 связи IPsec SA создаются сразу
           же (без ожидания).";
      }
    }
    description
      "Правила установки конфигурации IPsec SA в ядро NSF при запуске
       реализации IKEv2.";
  }

  typedef fs-group {
    type uint16;
    description
      "Группы DH для IKE и смены ключей IPsec SA.";
    reference
      "IANA: Internet Key Exchange Version 2 (IKEv2) Parameters,
             IKEv2 Transform Attribute Types, Transform Type 4 -
             Diffie-Hellman Group Transform IDs
       RFC 7296: Internet Key Exchange Protocol Version 2
                 (IKEv2), Section 3.3.2.";
  }

  typedef auth-protocol-type {
    type enumeration {
      enum ikev2 {
        value 2;
        description
          "Протокол аутентификации IKEv2. Сейчас определён лишь 1
           протокол, но в будущем возможно расширение набора.";
      }
    }
    description
      "Версия протокола аутентификации IKE, заданная в базе(PAD).
       Она определена как enum для поддержки будущих версий IKE.";
    reference
      "RFC 7296: Internet Key Exchange Protocol Version 2
                 (IKEv2).";
  }

  typedef auth-method-type {
    type enumeration {
      enum pre-shared {
        description
          "Выбор аутентификации по заранее распространенным ключам.";
        reference
          "RFC 7296: Internet Key Exchange Protocol Version 2
                     (IKEv2).";
      }
      enum eap {
        description
          "Выбор протокола (EAP) в качестве метода аутентификации.";
        reference
          "RFC 7296: Internet Key Exchange Protocol Version 2
                     (IKEv2).";
```

```
        }
        enum digital-signature {
          description
            "Выбор цифровой подписи как метода аутентификации.";
          reference
            "RFC 7296: Internet Key Exchange Protocol Version 2 (IKEv2)
             RFC 7427: Signature Authentication in the Internet Key
                       Exchange Version 2 (IKEv2).";
        }
        enum null {
          description
            "без аутентификации (Null).";
          reference
            "RFC 7619: The NULL Authentication Method in the Internet
                       Key Exchange Protocol Version 2 (IKEv2).";
        }
      }
      description
        "Метод аутентификации партнёра, указанный в PAD.";
    }

    container ipsec-ike {
      description
        "Конфигурация IKE для NSF, включающая параметры PAD, данные
         соединения IKE и состояния.";
      container pad {
        description
          "Конфигурация базы данных (PAD), каждая запись которой
           содержит данные конфигурации локального или удалённого
           партнёра. Поэтому контроллер I2NSF сохраняет данные
           аутентификации (и свидетельства) не только для локального,
           но и для удалённого элемента NSF. Локальный элемент NSF
           МОЖЕТ применять одно отождествление для разных типов
           аутентификации и свидетельств. Можно указать записи для
           локального (например, A) и удалённого (например, B) элемента
           NSF, чтобы задать все требуемые данные для аутентификации
           между A и B (../conn-entry/local и ../conn-entry/remote).";
        list pad-entry {
          key "name";
          ordered-by user;
          description
            "Список записей базы данных PAD, упорядоченных контроллером
             I2NSF. Каждая запись однозначно указывается именем.";
          leaf name {
            type string;
            description
              "Уникальное в PAD имя данной записи.";
          }
          choice identity {
            mandatory true;
            description
              "Конкретный партнёр (локальный или удалённый) IKE будет
               идентифицироваться по одному из этих отождествлений.";
            reference
              "RFC 4301: Security Architecture for the Internet
                         Protocol, Section 4.4.3.1.";
            case ipv4-address {
              leaf ipv4-address {
                type inet:ipv4-address;
                description
                  "Отождествление в виде 4-октетного адреса IPv4.";
              }
            }
            case ipv6-address {
              leaf ipv6-address {
                type inet:ipv6-address;
                description
                  "Отождествление в виде 16-октетного адреса IPv6,
                   например,2001:db8::8:800:200c:417a.";
              }
            }
            case fqdn-string {
              leaf fqdn-string {
                type inet:domain-name;
                description
                  "Отождествление в виде строки полного доменного имени
                   (FQDN), например, example.com. В строку НЕДОПУСТИМО
                   включать какие-либо терминаторы (NULL, CR и т. п.)";
              }
            }
            case rfc822-address-string {
              leaf rfc822-address-string {
                type string;
                description
                  "Отождествление в виде строки полного адреса
                   электронной почты (RFC 5322), например,
```

```
            jsmith@example.com. В строку НЕДОПУСТИМО
            включать какие-либо терминаторы (NULL, CR и т. п.)";
          reference
            "RFC 5322: Internet Message Format.";
        }
      }
      case dnx509 {
        leaf dnx509 {
          type binary;
          description
            "Двоичное представление DER отличительного имени
             ASN.1 X.500, как задано в IKEv2.";
          reference
            "RFC 5280: Internet X.509 Public Key Infrastructure
                       Certificate and Certificate Revocation
                       List (CRL) Profile
             RFC 7296: Internet Key Exchange Protocol Version 2
                       (IKEv2), Section 3.5.";
        }
      }
      case gnx509 {
        leaf gnx509 {
          type binary;
          description
            "Структура ASN.1 X.509 GeneralName, как задано в
             RFC 5280, закодированная с использованием правил
             ASN.1 DER в соответствии с ITU-T X.690.";
          reference
            "RFC 5280: Internet X.509 Public Key Infrastructure
                       Certificate and Certificate Revocation
                       List (CRL) Profile.";
        }
      }
      case id-key {
        leaf id-key {
          type binary;
          description
            "Необрабатываемый (opaque) поток октетов, который
             может служить для передачи фирменных сведений при
             использовании фирменных типов аутентификации.";
          reference
            "RFC 7296: Internet Key Exchange Protocol Version 2
                       (IKEv2), Section 3.5.";
        }
      }
      case id-null {
        leaf id-null {
          type empty;
          description
            "Применяется аутентификация ID_NULL, если не
             используются аутентификационные данные IKE.";
          reference
            "RFC 7619: The NULL Authentication Method in the
                       Internet Key Exchange Protocol Version 2
                       (IKEv2).";
        }
      }
    }
    leaf auth-protocol {
      type auth-protocol-type;
      default "ikev2";
      description
        "Сейчас поддерживается только IKEv2, но в будущем
         возможны иные протоколы аутентификации.";
    }
    container peer-authentication {
      description
        "Позволяет контроллеру безопасности настроить метод
         аутентификации (заранее распространённый ключ, eap,
         цифровая подпись, null), который будет применяться
         для конкретного партнёра, и используемые свидетельства
         в зависимости от выбранного метода аутентификации.";
      leaf auth-method {
        type auth-method-type;
        default "pre-shared";
        description
          "Тип метода аутентификации (заранее распространенный
           ключ, eap, цифровая подпись, null).";
        reference
          "RFC 7296: Internet Key Exchange Protocol Version 2
                     (IKEv2), Section 2.15.";
      }
      container eap-method {
        when "../auth-method = 'eap'";
        leaf eap-type {
          type uint32 {
```

```
          range "1 .. 4294967295";
        }
        mandatory true;
        description
          "Тип метода EAP, заданный значением из реестра IANA.
           В зависимости от метода EAP может применяться ключ
           PSK или сертификаты.";
      }
      description
        "Описание метода EAP, используемого при значении eap.";
      reference
        "IANA: Extensible Authentication Protocol (EAP)
               Registry, Method Types
         RFC 7296: Internet Key Exchange Protocol Version 2
                   (IKEv2), Section 2.16.";
    }
    container pre-shared {
      when "../auth-method[.='pre-shared' or
            .='eap']";
      leaf secret {
        nacm:default-deny-all;
        type yang:hex-string;
        description
          "Значение ключа PSK. NSF не разрешает считывание
           этого значения из соображений безопасности. Значение
           ДОЛЖНО устанавливаться, если метод EAP использует
           заранее распространённый ключ или выбран метод
           аутентификации pre-shared.";
      }
      description
        "Значение секретного ключа для аутентификации PSK или
         EAP на основе PSK.";
    }
    container digital-signature {
      when "../auth-method[.='digital-signature'
            or .='eap']";
      leaf ds-algorithm {
        type uint8;
        default "14";
        description
          "Алгоритм цифровой подписи, заданный значением из
           реестра IANA. По умолчанию применяется базовый метод
           цифровой подписи. В зависимости от алгоритма ДОЛЖНЫ
           указываться последующие листья. Например, если метод
           EAP или цифровая подпись включает сертификат, листья
           cert-data и private-key будут содержать информацию";
        reference
          "IANA: Internet Key Exchange Version 2 (IKEv2)
                 Parameters, IKEv2 Authentication Method.";
      }
      choice public-key {
        leaf raw-public-key {
          type binary;
          description
            "Двоичное значение открытого ключа, интерпретация
             которого определяется алгоритмом цифровой подписи.
             Например, ключ RSA представляет RSAPublicKey в
             соответствии с RFC 8017 а криптографию с
             эллиптической кривой (ECC) — publicKey, как
             указано в RFC 5915.";
          reference
            "RFC 5915: Elliptic Curve Private Key Structure
             RFC 8017: PKCS #1: RSA Cryptography
                       Specifications Version 2.2.";
        }
        leaf cert-data {
          type binary;
          description
            "Данные сертификата X.509 в формате DER. Если задан
             raw-public-key, этот лист будет пустым.";
          reference
            "RFC 5280: Internet X.509 Public Key
                       Infrastructure Certificate
                       and Certificate Revocation
                       List (CRL) Profile.";
        }
        description
          "Если контроллер I2NSF знает, что NSF уже владеет
           секретным ключом, связанным с данным открытым ключом
           (например, NSF создаёт пару ключей по отдельному
           каналу), будет настраиваться лишь один из листьев
           данного choice, но не лист private-key. По открытому
           ключу NSF может определить секретный ключ для
           использования.";
      }
      leaf private-key {
```

```
              nacm:default-deny-all;
              type binary;
              description
                "Двоичное значение секретного ключа, интерпретация
                 которого определяется алгоритмом цифровой подписи.
                 Например, ключ RSA представляет RSAPublicKey в
                 соответствии с RFC 8017 а криптографию с
                 эллиптической кривой (ECC) — publicKey, как указано
                 в RFC 5915. Это значение устанавливается, если
                 определён открытый ключ и контроллер I2NSF отвечает
                 за настройку секретного ключа. В иных случаях лист
                 не задаётся и значение хранится в секрете.";
              reference
                "RFC 5915: Elliptic Curve Private Key Structure
                 RFC 8017: PKCS #1: RSA Cryptography
                           Specifications Version 2.2.";
            }
            leaf-list ca-data {
              type binary;
              description
                "Список сертификатов доверенных удостоверяющих
                 центров (CA) в кодировке ASN.1 (DER). По умолчанию
                 пустое значение.";
            }
            leaf crl-data {
              type binary;
              description
                "Структура CertificateList, как задано в RFC 5280,
                 в представлении ASN.1 (DER)в соответствии с ITU-T
                 X.690. По умолчанию пустое значение.";
              reference
                "RFC 5280: Internet X.509 Public Key Infrastructure
                           Certificate and Certificate Revocation
                           List (CRL) Profile.";
            }
            leaf crl-uri {
              type inet:uri;
              description
                "URI списка отзыва сертификатов X.509 (CRL).
                 По умолчанию пустое значение.";
              reference
                "RFC 5280: Internet X.509 Public Key Infrastructure
                           Certificate and Certificate Revocation
                           List (CRL) Profile.";
            }
            leaf oscp-uri {
              type inet:uri;
              description
                "URI протокола статуса сертификатов OCSP.
                 По умолчанию пустое значение.";
              reference
                "RFC 6960: X.509 Internet Public Key Infrastructure
                           Online Certificate Status Protocol - OCSP
                 RFC 5280: Internet X.509 Public Key Infrastructure
                           Certificate and Certificate Revocation
                           List (CRL) Profile.";
            }
            description
              "Контейнер digital-signature.";
          } /*Контейнер digital-signature*/
        } /*Контейнер peer-authentication*/
      }
    }
    list conn-entry {
      key "name";
      description
        "Список сведений о соединениях IKE с партнёрами. Список
         поддерживается в реальном масштабе времени по мере создания
         IKE SA с этими узлами.";
      leaf name {
        type string;
        description
          "Идентификатор записи для соединения.";
      }
      leaf autostartup {
        type autostartup-type;
        default "add";
        description
          "По умолчанию лишь добавляется конфигурация без запуска
           защищённой связи.";
      }
      leaf initial-contact {
        type boolean;
        default "false";
        description
          "Значение true отключает использование уведомлений
```

```
        INITIAL_CONTACT. При значении false использование
        INITIAL_CONTACT зависит от реализации IKEv2.";
    }
    leaf version {
      type auth-protocol-type;
      default "ikev2";
      description
        "Версия IKE (поддерживается лишь версия 2).";
    }
    container fragmentation {
      leaf enabled {
        type boolean;
        default "false";
        description
          "Управляет фрагментацией IKEv2 (true или false).";
        reference
          "RFC 7383: Internet Key Exchange Protocol Version 2
                     (IKEv2) Message Fragmentation.";
      }
      leaf mtu {
        when "../enabled='true'";
        type uint16 {
          range "68..65535";
        }
        description
          "MTU для использования при фрагментации IKEv2.";
        reference
          "RFC 7383: Internet Key Exchange Protocol Version 2
                     (IKEv2) Message Fragmentation.";
      }
      description
        "Фрагментация IKEv2 в соответствии с RFC 7383. Если
         фрагментация IKEv2 разрешена, можно задать MTU.";
    }
    container ike-sa-lifetime-soft {
      description
        "Мягкий срок действия IKE SA. Можно задать два значения -
         время смены ключей IKE SA или время новой аутентификации
         IKE SA.";
      leaf rekey-time {
        type uint32;
        units "seconds";
        default "0";
        description
          "Число секунд между сменами ключей IKE SA. Значение 0
           указывает неограниченный срок действия ключей.";
      }
      leaf reauth-time {
        type uint32;
        units "seconds";
        default "0";
        description
          "Интервал повторной аутентификации IKE SA (в секундах)
           Значение 0 указывает неограниченный срок действия.";
      }
      reference
        "RFC 7296: Internet Key Exchange Protocol Version 2
                   (IKEv2), Section 2.8.";
    }
    container ike-sa-lifetime-hard {
      description
        "Жёсткий срок действия IKE SA, по истечении которого
         IKE SA удаляется.";
      leaf over-time {
        type uint32;
        units "seconds";
        default "0";
        description
          "Срок действия IKE SA в секундах. Значение 0
           указывает неограниченный срок действия.";
      }
      reference
        "RFC 7296: Internet Key Exchange Protocol Version 2
                   (IKEv2).";
    }
    leaf-list ike-sa-intr-alg {
      type nsfikec:intr-alg-t;
      default "12";
      ordered-by user;
      description
        "Алгоритм защиты целостности для организации IKE SA. Список
         упорядочен по снижению приоритета. По умолчанию задано
         значение 12 (AUTH_HMAC_SHA2_256_128).";
    }
    list ike-sa-encr-alg {
      key "id";
```

```
        min-elements 1;
        ordered-by user;
        leaf id {
          type uint16;
          description
            "Идентификатор, однозначно указывающий каждую запись
             списка, т. е. алгоритм шифрования и размер ключа
             (если размер нужен).";
        }
        leaf algorithm-type {
          type nsfikec:encr-alg-t;
          default "12";
          description
            "Принятое по умолчанию значение 12 (ENCR_AES_CBC).";
        }
        leaf key-length {
          type uint16;
          default "128";
          description
            "По умолчанию ключ имеет размер 128 битов.";
        }
        description
          "Алгоритм шифрования или AEAD для IKE SA. Список
           упорядочен по снижению приоритета.";
      }
      leaf dh-group {
        type fs-group;
        default "14";
        description
          "Номер группы показателя Диффи-Хеллмана для применения в
           IKE_SA_INIT при обмене ключами IKE SA.";
      }
      leaf half-open-ike-sa-timer {
        type uint32;
        units "seconds";
        default "0";
        description
          "Тайм-аут полуоткрытой IKE SA. Значение 0 указывает
           неограниченный срок.";
        reference
          "RFC 7296: Internet Key Exchange Protocol Version 2
                     (IKEv2), Section 2.";
      }
      leaf half-open-ike-sa-cookie-threshold {
        type uint32;
        default "0";
        description
          "Число полуоткрытых IKE SA, активирующее механизм cookie.
           Значение 0 предполагает неограниченное число.";
        reference
          "RFC 7296: Internet Key Exchange Protocol Version 2
                     (IKEv2), Section 2.6.";
      }
      container local {
        leaf local-pad-entry-name {
          type string;
          mandatory true;
          description
            "Имя записи PAD, в которой хранятся данные аутентификации
             конкретного локального партнёра. Значение ДОЛЖНО
             совпадать с pad-entry-name.";
        }
        description
          "Данные аутентификации локального партнёра.";
      }
      container remote {
        leaf remote-pad-entry-name {
          type string;
          mandatory true;
          description
            "Имя записи PAD, в которой хранятся данные аутентификации
             конкретного удалённого партнёра. Значение ДОЛЖНО
             совпадать с pad-entry-name.";
        }
        description
          "Данные аутентификации удаленного партнёра.";
      }
      container encapsulation-type {
        uses nsfikec:encap;
        description
          "Конфигурационные сведения о портах источника и получателя,
           которые следует использовать при инкапсуляции IKE, и тип
           инкапсуляции, который следует применять при работе через
           NAT. Это лишь рекомендация, поскольку реализация IKE
           может выбрать иную инкапсуляцию, как указано в RFC 8229.";
        reference
```

```
        "RFC 8229: TCP Encapsulation of IKE and IPsec Packets.";
    }
    container spd {
      description
        "Конфигурация базы правил безопасности (SPD). Эти основные
         помещаются в группировку ipsec-policy-grouping.";
      list spd-entry {
        key "name";
        ordered-by user;
        leaf name {
          type string;
          description
            "Уникальное имя записи SPD для задания политики IPsec";
        }
        container ipsec-policy-config {
          description
            "Контейнер конфигурации политики IPsec.";
          uses nsfikec:ipsec-policy-grouping;
        }
        description
          "Список записей, представляющих SPD. Поскольку в этом
           случае NSF реализует IKE, требуется лишь передать
           политику IPsec от локального элемента NSF к удалённому.
           Реализация IKE будет помещать правила IPsec в ядро NSF
           для обоих направления (входного и выходного) и создавать
           соответствующие IPsec SA на основе сведений из SPD.";
      }
      reference
        "RFC 7296: Internet Key Exchange Protocol Version 2
                   (IKEv2), Section 2.9.";
    }
    container child-sa-info {
      leaf-list fs-groups {
        type fs-group;
        default "0";
        ordered-by user;
        description
          "Ненулевое значение указывает необходимость полной защиты
           (forward secrecy) при создании IPsec SA и задаёт номер
           группы для процесса обмена ключами для достижения полной
           защиты. Список упорядочивается по снижению приоритета.";
      }
      container child-sa-lifetime-soft {
        description
          "Мягкий срок действия IPsec SA, по достижении которого
           выполняется действие, заданное листом action.";
        uses nsfikec:lifetime;
        leaf action {
          type nsfikec:lifetime-action;
          default "replace";
          description
            "По завершении срока IPsec SA нужно выполнить данное
             действие по отношению к IPsec SA. Возможны 3 варианта:
             terminate-clear, terminate-hold, replace.";
          reference
            "RFC 4301: Security Architecture for the Internet
                       Protocol, Section 4.5
             RFC 7296: Internet Key Exchange Protocol Version 2
                       (IKEv2), Section 2.8.";
        }
      }
      container child-sa-lifetime-hard {
        description
          "Жёсткий срок действия IPsec SA, по истечение которого
           IPsec SA прерывается.";
        uses nsfikec:lifetime;
        reference
          "RFC 7296: Internet Key Exchange Protocol Version 2
                     (IKEv2), Section 2.8.";
      }
      description
        "Сведения для IPsec SA, включающие группу полной защиты
         (PFS) и интервал замены ключей IPsec SA.";
    }
    container state {
      config false;
      leaf initiator {
        type boolean;
        description
          "Роль инициатора для данного соединения.";
      }
      leaf initiator-ikesa-spi {
        type ike-spi;
        description
          "IKE SA SPI инициатора.";
      }
```

```
          leaf responder-ikesa-spi {
            type ike-spi;
            description
              "IKE SA SPI ответчика.";
          }
          leaf nat-local {
            type boolean;
            description
              "Значение true указывает размещение локальной конечной
               точки за NAT.";
          }
          leaf nat-remote {
            type boolean;
            description
              "Значение true указывает размещение удалённой конечной
               точки за NAT.";
          }
          container encapsulation-type {
            uses nsfikec:encap;
            description
              "Сведения о портах отправителя и получателя, используемых
               при инкапсуляции IKE, и тип инкапсуляции при работе
               через NAT.";
            reference
              "RFC 8229: TCP Encapsulation of IKE and IPsec Packets.";
          }
          leaf established {
            type uint64;
            units "seconds";
            description
              "Число секунд с момента создания IKE SA.";
          }
          leaf current-rekey-time {
            type uint64;
            units "seconds";
            description
              "Число секунд до смены ключей IKE SA.";
          }
          leaf current-reauth-time {
            type uint64;
            units "seconds";
            description
              "Число секунд до новой аутентификации IKE SA.";
          }
          description
            "Данные состояния IKE для конкретного соединения.";
        } /* ike-sa-state */
      } /* ike-conn-entries */
      container number-ike-sas {
        config false;
        leaf total {
          type yang:gauge64;
          description
            "Общее число активных IKE SA.";
        }
        leaf half-open {
          type yang:gauge64;
          description
            "Число полуоткрытых активных IKE SA.";
        }
        leaf half-open-cookies {
          type yang:gauge64;
          description
            "Число полуоткрытых активных IKE SA с cookie.";
        }
        description
          "Общие сведения о IKE SA, в частности, текущее число IKE SA";
      }
    } /* Контейнер ipsec-ike */
  }
  <CODE ENDS>
```

5.3. Модуль ietf-i2nsf-ikeless

В этом параграфе описан модуль YANG для варианта без IKE.

5.3.1. Обзор модели данных

Для этого варианта определение модели SPD в основном взято из параграфа 4.4.1 и Приложения D в [RFC4301], но внесены указанные ниже изменения.

- Для простоты каждое правило IPsec (spd-entry) содержит один селектор трафика вместо их списка. Причина заключается в том, что фактические реализации ядра представляют лишь один селектор трафика на правило.
- Каждое правило IPsec включает идентификатор (reqid) для связывания с IPsec SA, как принято в Linux.
- Каждое правило IPsec имеет лишь одно имя, а не список имён.

- Удалены комбинированные алгоритмы, поскольку для шифрования **можно** применять алгоритмы AEAD.
- Туннельные данные были расширены отображением DSCP, поскольку некоторые реализации ядер разрешают настраивать эти значения.

Определение модели SAD в основном взято из параграфа 4.4.2 в [RFC4301], но внесены указанные ниже изменения.

- Для простоты каждое правило IPsec (spd-entry) содержит один селектор трафика вместо их списка. Причина заключается в том, что фактические реализации ядра представляют лишь один селектор трафика на правило.
- Каждая IPsec SA включает идентификатор (reqid) для связывания IPsec SA с правилом IPsec. Причиной послужила возможность включения этого значения во многих реализациях ядра.
- IPsec SA именуются так же, как правила IPsec.
- Модель поддерживает задание алгоритма шифрования, которым может быть AEAD или иной (не AEAD) алгоритм. При указании алгоритма AEAD не требуется задавать алгоритм защиты целостности, для иных алгоритмов он требуется [RFC8221].
- Туннельные данные были расширены отображением DSCP. Предполагается, что NSF, охватываемые этим документом, обеспечивают полную функциональность ECN для предотвращения отбрасывания уведомлений ECN о перегрузках [RFC6040].
- Срок действия IPsec SA может задаваться временем простоя или числом пакетов IP в качестве порога для триггера. Это обусловлено поддержкой фактическими реализациями ядер установки таких триггеров.
- Включены данные для настройки типа инкапсуляции (encapsulation-type) для пакетов IPsec ESP (UDP [RFC3948] или TCP [RFC8229]).

Модель уведомлений определена на основе спецификации PF_KEYv2 [RFC2367].

Модель данных YANG для варианта без IKE задана в модуле ietf-i2nsf-ikeless, структура которого показана ниже в форме дерева YANG с нотацией [RFC8340].

```
module: ietf-i2nsf-ikeless
  +--rw ipsec-ikeless
    +--rw spd
    |  +--rw spd-entry* [name]
    |     +--rw name  string
    |     +--rw direction nsfikec:ipsec-traffic-direction
    |     +--rw reqid? uint64
    |     +--rw ipsec-policy-config
    |        +--rw anti-replay-window-size?   uint32
    |        +--rw traffic-selector
    |        |  +--rw local-prefix      inet:ip-prefix
    |        |  +--rw remote-prefix     inet:ip-prefix
    |        |  +--rw inner-protocol?   ipsec-inner-protocol
    |        |  +--rw local-ports* [start end]
    |        |  |  +--rw start    inet:port-number
    |        |  |  +--rw end      inet:port-number
    |        |  +--rw remote-ports* [start end]
    |        |     +--rw start    inet:port-number
    |        |     +--rw end      inet:port-number
    |        +--rw processing-info
    |           +--rw action?         ipsec-spd-action
    |           +--rw ipsec-sa-cfg
    |             +--rw pfp-flag?              boolean
    |             +--rw ext-seq-num?           boolean
    |             +--rw seq-overflow?          boolean
    |             +--rw stateful-frag-check?   boolean
    |             +--rw mode?                  ipsec-mode
    |             +--rw protocol-parameters? ipsec-protocol-params
    |              +--rw esp-algorithms
    |              |  +--rw integrity*    intr-alg-t
    |              |  +--rw encryption* [id]
    |              |  |  +--rw id              uint16
    |              |  |  +--rw algorithm-type?   encr-alg-t
    |              |  |  +--rw key-length?       uint16
    |              |  +--rw tfc-pad?      boolean
    |              +--rw tunnel
    |                 +--rw local           inet:ip-address
    |                 +--rw remote          inet:ip-address
    |                 +--rw df-bit?         enumeration
    |                 +--rw bypass-dscp?    boolean
    |                 +--rw dscp-mapping* [id]
    |                    +--rw id            uint8
    |                    +--rw inner-dscp?   inet:dscp
    |                    +--rw outer-dscp?   inet:dscp
    +--rw sad
      +--rw sad-entry* [name]
       +--rw name               string
       +--rw reqid?             uint64
       +--rw ipsec-sa-config
       |  +--rw spi                        uint32
       |  +--rw ext-seq-num?               boolean
       |  +--rw seq-overflow?              boolean
       |  +--rw anti-replay-window-size?   uint32
```

```
        |  +--rw traffic-selector
        |  |  +--rw local-prefix      inet:ip-prefix
        |  |  +--rw remote-prefix     inet:ip-prefix
        |  |  +--rw inner-protocol?   ipsec-inner-protocol
        |  |  +--rw local-ports* [start end]
        |  |  |  +--rw start    inet:port-number
        |  |  |  +--rw end      inet:port-number
        |  |  +--rw remote-ports* [start end]
        |  |     +--rw start    inet:port-number
        |  |     +--rw end      inet:port-number
        |  +--rw protocol-parameters? nsfikec:ipsec-protocol-params
        |  +--rw mode?                    nsfikec:ipsec-mode
        |  +--rw esp-sa
        |  |  +--rw encryption
        |  |  |  +--rw encryption-algorithm?   nsfikec:encr-alg-t
        |  |  |  +--rw key?                    yang:hex-string
        |  |  |  +--rw iv?                     yang:hex-string
        |  |  +--rw integrity
        |  |     +--rw integrity-algorithm?   nsfikec:intr-alg-t
        |  |     +--rw key?                   yang:hex-string
        |  +--rw sa-lifetime-hard
        |  |  +--rw time?      uint32
        |  |  +--rw bytes?     yang:counter64
        |  |  +--rw packets?   uint32
        |  |  +--rw idle?      uint32
        |  +--rw sa-lifetime-soft
        |  |  +--rw time?      uint32
        |  |  +--rw bytes?     yang:counter64
        |  |  +--rw packets?   uint32
        |  |  +--rw idle?      uint32
        |  |  +--rw action?    nsfikec:lifetime-action
        |  +--rw tunnel
        |  |  +--rw local           inet:ip-address
        |  |  +--rw remote          inet:ip-address
        |  |  +--rw df-bit?         enumeration
        |  |  +--rw bypass-dscp?    boolean
        |  |  +--rw dscp-mapping* [id]
        |  |  |  +--rw id            uint8
        |  |  |  +--rw inner-dscp?   inet:dscp
        |  |  |  +--rw outer-dscp?   inet:dscp
        |  |  +--rw dscp-values*    inet:dscp
        |  +--rw encapsulation-type
        |     +--rw espencap?   esp-encap
        |     +--rw sport?      inet:port-number
        |     +--rw dport?      inet:port-number
        |     +--rw oaddr*      inet:ip-address
        +--ro ipsec-sa-state
           +--ro sa-lifetime-current
           |  +--ro time?      uint32
           |  +--ro bytes?     yang:counter64
           |  +--ro packets?   uint32
           |  +--ro idle?      uint32
           +--ro replay-stats
              +--ro replay-window
              |  +--ro w?   uint32
              |  +--ro t?   uint64
              |  +--ro b?   uint64
              +--ro packet-dropped?       yang:counter64
              +--ro failed?               yang:counter64
              +--ro seq-number-counter?   uint64

notifications:
  +---n sadb-acquire {ikeless-notification}?
  |  +--ro ipsec-policy-name    string
  |  +--ro traffic-selector
  |     +--ro local-prefix      inet:ip-prefix
  |     +--ro remote-prefix     inet:ip-prefix
  |     +--ro inner-protocol?   ipsec-inner-protocol
  |     +--ro local-ports* [start end]
  |     |  +--ro start    inet:port-number
  |     |  +--ro end      inet:port-number
  |     +--ro remote-ports* [start end]
  |        +--ro start    inet:port-number
  |        +--ro end      inet:port-number
  +---n sadb-expire {ikeless-notification}?
  |  +--ro ipsec-sa-name           string
  |  +--ro soft-lifetime-expire?   boolean
  |  +--ro lifetime-current
  |     +--ro time?      uint32
  |     +--ro bytes?     yang:counter64
  |     +--ro packets?   uint32
  |     +--ro idle?      uint32
  +---n sadb-seq-overflow {ikeless-notification}?
  |  +--ro ipsec-sa-name    string
  +---n sadb-bad-spi {ikeless-notification}?
     +--ro spi    uint32
```

Модель данных YANG содержит уникальный контейнер ipsec-ikeless, включающий контейнеры spd и sad. В контейнере spd содержится список записей базы правил безопасности SPD. По сравнению с моделью YANG для варианта с IKE эта часть задаёт несколько дополнительных параметров, требуемых из-за отсутствия программ IKE в NSF - направление трафика для применения правила IPsec и значение reqid для привязки правила IPsec к IPsec SA, поскольку иначе поиск будет затруднён. Контейнер sad содержит список записей базы защищённых связей SAD. Обычно каждая запись позволяет указать данные конфигурации (SPI, селекторы трафика, туннельный или транспортный режим, криптоалгоритмы и ключевой материал, мягкий и жёсткий срок действия и т. п.), а также данные состояния (оставшийся срок действия, статистика повторов и т. п) конкретной IPsec SA.

Кроме того, модуль определяет набор уведомлений, позволяющих элементу NSF информировать контроллер I2NSF о соответствующих событиях, таких как завершение срока действия IPsec SA, переполнение порядкового номера, недопустимый индекс SPI в полученном пакете.

5.3.2. Пример использования

В Приложении B приведён пример конфигурации NSF в транспортном режиме (хост-хост) без IKE, а в Приложении C - примеры уведомлений о получении и завершении срока действия IPsec SA, переполнении порядкового номера, некорректных SPI.

5.3.3. Модуль YANG

Этот модуль YANG включает нормативные ссылки на [RFC4301], [RFC4303], [RFC6991], [RFC8174] и [RFC8341].

```
<CODE BEGINS> file "ietf-i2nsf-ikeless@2021-07-14.yang"
module ietf-i2nsf-ikeless {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-i2nsf-ikeless";
  prefix nsfikels;

  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Data Types.";
  }
  import ietf-yang-types {
    prefix yang;
    reference
      "RFC 6991: Common YANG Data Types.";
  }
  import ietf-i2nsf-ikec {
    prefix nsfikec;
    reference
      "RFC 9061: A YANG Data Model for IPsec Flow Protection
                 Based on Software-Defined Networking (SDN).";
  }
  import ietf-netconf-acm {
    prefix nacm;
    reference
      "RFC 8341: Network Configuration Access Control Model.";
  }

  organization
    "IETF I2NSF Working Group";
  contact
    "WG Web:  <https://datatracker.ietf.org/wg/i2nsf/>
     WG List: <mailto:i2nsf@ietf.org>

     Author: Rafael Marin-Lopez
              <mailto:rafa@um.es>

     Author: Gabriel Lopez-Millan
              <mailto:gabilm@um.es>

     Author: Fernando Pereniguez-Garcia
              <mailto:fernando.pereniguez@cud.upct.es>
    ";
  description
    "Модель данных для основанной на SDN службы защиты потоков IPsec
     без IKE.

     Ключевые слова ДОЛЖНО, НЕДОПУСТИМО, ТРЕБУЕТСЯ, НУЖНО, НЕ НУЖНО,
     СЛЕДУЕТ, НЕ СЛЕДУЕТ, РЕКОМЕНДУЕТСЯ, НЕ РЕКОМЕНДУЕТСЯ, МОЖНО,
     НЕОБЯЗАТЕЛЬНО в этом документе трактуются в соответствии с
     BCP 14 (RFC 2119) (RFC 8174) тогда и только тогда, когда они
     указаны заглавными буквами, как показано здесь.

     Авторские права (Copyright (c) 2021) принадлежат IETF Trust и
     лицам, указанным как авторы. Все права защищены.

     Распространение и применение модуля в исходной или двоичной
     форме с изменениями или без таковых разрешено в соответствии с
     лицензией Simplified BSD License, изложенной в параграфе 4.c
     IETF Trust's Legal Provisions Relating to IETF Documents
     (https://trustee.ietf.org/license-info).
```

```
       Эта версия модуля YANG является частью RFC 9061, где правовые
       аспекты приведены более полно.";

     revision 2021-07-14 {
       description
         "Исходный выпуск.";
       reference
         "RFC 9061: A YANG Data Model for IPsec Flow Protection
                    Based on Software-Defined Networking (SDN).";
     }

     feature ikeless-notification {
       description
         "Указывает поддержку сервером генерации уведомлений в модуле
          ikeless. Для расширения применимости модуля уведомления
          заданы как свойства (feature). Для реализаций без IKE
          предполагается поддержка этих свойств в NSF.";
     }

     container ipsec-ikeless {
       description
         "Контейнер для настройки варианта без IKE, содержащий
          контейнеры spd и sad. Первый позволяет контроллеру I2NSF
          настраивать правила IPsec в базе данных SPD, а второй -
          настраивать IPsec SA в базе данных SAD.";
       reference
         "RFC 4301: Security Architecture for the Internet Protocol.";
       container spd {
         description
           "Конфигурация базы правил безопасности (SPD).";
         reference
           "RFC 4301: Security Architecture for the Internet Protocol,
                      Section 4.4.1.2.";
         list spd-entry {
           key "name";
           ordered-by user;
           leaf name {
             type string;
             description
               "Уникальное имя записи SPD.";
           }
           leaf direction {
             type nsfikec:ipsec-traffic-direction;
             mandatory true;
             description
               "Входящий или исходящий трафик. В варианте без IKE
                контроллеру I2NSF нужно задавать направления для правил,
                применяемых в NSF. В варианте с IKE это направление
                задавать не нужно, поскольку IKE определяет его.";
           }
           leaf reqid {
             type uint64;
             default "0";
             description
               "Связывает правило IPsec с IPsec SA с тем же reqid. Это
                нужно только в варианте без IKE поскольку в варианте с
                IKE это будет внутренняя привязка IKE.";
           }
           container ipsec-policy-config {
             description
               "Контейнер с конфигурацией политики IPsec.";
             uses nsfikec:ipsec-policy-grouping;
           }
           description
             "SPD представляется списком записей, каждая из которых
              задаёт правило IPsec.";
         } /*Список spd-entry*/
       } /*Контейнер spd*/
       container sad {
         description
           "Конфигурация базы данных IPsec SAD.";
         reference
           "RFC 4301: Security Architecture for the Internet Protocol,
                      Section 4.4.2.1.";
         list sad-entry {
           key "name";
           ordered-by user;
           leaf name {
             type string;
             description
               "Имя записи в базе SAD.";
           }
           leaf reqid {
             type uint64;
             default "0";
             description
```

```
        "Связывает IPsec SA с правилом IPsec с тем же reqid.";
    }
    container ipsec-sa-config {
      description
        "Контейнер для настройки деталей IPsec SA.";
      leaf spi {
        type uint32 {
          range "0..max";
        }
        mandatory true;
        description
          "Индекс параметров безопасности (SPI) для IPsec SA.";
      }
      leaf ext-seq-num {
        type boolean;
        default "true";
        description
          "Значение true указывает использование в этой IPsec SA
           расширенных порядковых номеров (64 бита), false
           указывает обычные порядковые номера (32 бита).";
      }
      leaf seq-overflow {
        type boolean;
        default "false";
        description
          "Флаг, управляющей дальнейшей передачей пакетов при
           переполнении счётчика порядковых номеров IPsec SA.
           Значение false запрещает дальнейшую передачу и требует
           замены ключей, true разрешает продолжать. При
           использовании AEAD (лист esp-algorithms/encryption/
           algorithm-type) этот флаг ДОЛЖЕН иметь значение false.
           Устанавливать true настоятельно не рекомендуется.";
      }
      leaf anti-replay-window-size {
        type uint32;
        default "64";
        description
          "Размер окна anti-replay, по умолчанию 64 в
           соответствии с рекомендациями RFC 4303.";
        reference
          "RFC 4303: IP Encapsulating Security Payload (ESP),
                     Section 3.4.3.";
      }
      container traffic-selector {
        uses nsfikec:selector-grouping;
        description
          "Селектор трафика IPsec SA.";
      }
      leaf protocol-parameters {
        type nsfikec:ipsec-protocol-params;
        default "esp";
        description
          "Протокол защиты для IPsec SA (пока лишь ESP).";
      }
      leaf mode {
        type nsfikec:ipsec-mode;
        default "transport";
        description
          "Туннельный или транспортный режим.";
      }
      container esp-sa {
        when "../protocol-parameters = 'esp'";
        description
          "Для IPsec SA с ESP требуется задать алгоритмы
           шифрования и защиты целостности, а также ключевой
           материал.";
        container encryption {
          description
            "Настройка алгоритма шифрования или AEAD для IPsec
             ESP.";
          leaf encryption-algorithm {
            type nsfikec:encr-alg-t;
            default "12";
            description
              "Настройка шифрования ESP. Для алгоритмов AEAD лист
               integrity-algorithm не используется.";
          }
          leaf key {
            nacm:default-deny-all;
            type yang:hex-string;
            description
              "Ключ шифрования ESP. Если лист не задан, ключ
               шифрования не определён (например, NULL). Размер
               ключа определяется заданным здесь значением, по
               умолчанию размер ключа составляет 128 битов.";
          }
```

```
        leaf iv {
          nacm:default-deny-all;
          type yang:hex-string;
          description
            "Значение вектора инициализации (IV) для шифрования
             ESP. Если этот лист не задан, значение IV не
             определено (например, шифрование NULL).";
        }
      }
      container integrity {
        description
          "Настройка защиты целостности для IPsec ESP. Этот
           контейнер позволяет настроить защиту целостности,
           если она нужна, а алгоритм AEAD не применяется.";
        leaf integrity-algorithm {
          type nsfikec:intr-alg-t;
          default "12";
          description
            "Алгоритм аутентификации сообщений (MAC) для защиты
             целостности в ESP (по умолчанию
             AUTH_HMAC_SHA2_256_128). С алгоритмами AEAD этот
             лист не используется.";
        }
        leaf key {
          nacm:default-deny-all;
          type yang:hex-string;
          description
            "Значение ключа защиты целостности ESP. Если этот
             лист не задан, ключ будет не определён (например,
             выбран алгоритм AEAD и алгоритм защиты целостности
             не нужен). Размер определяется выбранным ключом.";
        }
      }
    } /*Контейнер esp-sa*/
    container sa-lifetime-hard {
      description
        "Жёсткий срок действия IPsec SA, по завершении которого
         связь разрывается и удерживается (hold).";
      uses nsfikec:lifetime;
    }
    container sa-lifetime-soft {
      description
        "Мягкий срок действия IPsec SA.";
      uses nsfikec:lifetime;
      leaf action {
        type nsfikec:lifetime-action;
        description
          "Действие по окончании срока: terminate-clear,
           terminate-hold, replace.";
      }
    }
    container tunnel {
      when "../mode = 'tunnel'";
      uses nsfikec:tunnel-grouping;
      leaf-list dscp-values {
        type inet:dscp;
        description
          "Значения DSCP, разрешённые для внутренних пакетов,
           передаваемых через IPsec SA. Если значение не
           задано, применяется фильтрация DSCP. При
           ../bypass-dscp false и наличии dscp-mapping каждое
           значение здесь будет совпадать с внутренним кодом
           DSCP для отображения DSCP (список dscp-mapping).";
        reference
          "RFC 4301: Security Architecture for the Internet
                     Protocol, Section 4.4.2.1.";
      }
      description
        "Конечные точки туннеля IPsec.";
    }
    container encapsulation-type {
      uses nsfikec:encap;
      description
        "Контейнер с данными настройки для портов отправителя и
         получателя, которые будут применяться для инкапсуляции
         ESP и тип инкапсуляции при работе через NAT.";
    }
  } /*ipsec-sa-config*/
  container ipsec-sa-state {
    config false;
    description
      "Контейнер данных состояния IPsec SA.";
    container sa-lifetime-current {
      uses nsfikec:lifetime;
      description
        "Текущий срок действия SAD.";
```

```
          }
          container replay-stats {
            description
              "Данные состояния для окна anti-replay.";
            container replay-window {
              leaf w {
                type uint32;
                description
                  "Размер окна повторов.";
              }
              leaf t {
                type uint64;
                description
                  "Наибольший порядковый номер, аутентифицированный
                   на данный момент - верхняя граница окна.";
              }
              leaf b {
                type uint64;
                description
                  "Нижняя граница окна.";
              }
              description
                "Контейнер с параметрами, определяющими состояние
                 окна повтора - размер окна (w), наибольший
                 аутентифицированный порядковый номер (t), нижняя
                 граница окна. В соответствии с Приложением A2.1 к
                 RFC 4303 выполняется условие w = t - b + 1.";
              reference
                "RFC 4303: IP Encapsulating Security Payload (ESP),
                           Appendix A.";
            }
            leaf packet-dropped {
              type yang:counter64;
              description
                "Пакеты, отброшенные как повторы (replay).";
            }
            leaf failed {
              type yang:counter64;
              description
                "Число пакетов, обнаруженных вне окна повтора.";
            }
            leaf seq-number-counter {
              type uint64;
              description
                "Текущее значение 64-битового счётчика при
                 использовании в IPsec SA  расширенных порядковых
                 номеров или 32-битовое для обычных номеров.";
            }
          } /* Контейнер replay-stats*/
        } /*ipsec-sa-state*/
        description
          "Список записей базы данных SAD.";
      } /*Список sad-entry*/
    } /*Контейнер sad*/
  } /*Контейнер ipsec-ikeless*/

  /* Уведомления */

  notification sadb-acquire {
    if-feature "ikeless-notification";
    description
      "Элемент NSF обнаружил, что требуется IPsec SA для исходящих
       пакетов IP, соответствующих записи SPD. Контейнер
       traffic-selector в этом уведомлении содержит сведения о
       вызвавшем уведомление пакете IP.";
    leaf ipsec-policy-name {
      type string;
      mandatory true;
      description
        "Уникальное имя записи SPD, которая соответствует IPsec SA,
         требуемой для пакета IP. Предполагается, что контроллер
         I2NSF будет иметь копию информации об правилах, чтобы
         чтобы получить все сведения по этому идентификатору. Тип
         IPsec SA указывается в правиле, поэтому контроллер
         безопасности может также узнать тип IPsec SA, которую он
         ДОЛЖЕН создать.";
    }
    container traffic-selector {
      description
        "Пакет IP, требующий IPsec SA. В частности, контейнер будет
         включать IP-адреса/маски отправителя и получателя, протокол
         (udp, tcp и т. п.), номера портов отправителя и получателя";
      uses nsfikec:selector-grouping;
    }
  }
```

```
    notification sadb-expire {
      if-feature "ikeless-notification";
      description
        "завершение IPsec SA (мягкое или жесткое).";
      leaf ipsec-sa-name {
        type string;
        mandatory true;
        description
          "Уникальное имя записи SAD, для которой завершается срок
           действия IPsec SA. Предполагается, что у контроллера I2NSF
           будет копия сведений о IPsec SA (без криптографического
           материала и данных состояния) индексированная по именам
           (уникальный идентификатор), чтобы он мог знать всю
           информацию (криптоалгоритмы и т. п.) о завершающейся IPsec
           SA для замены ключей (мягкий срок действия) или удаления
           (жёсткий срок действия) по этому идентификатору.";
      }
      leaf soft-lifetime-expire {
        type boolean;
        default "true";
        description
          "Значение true указывает, что завершившийся срок действия был
           мягким, false указывает завершение жёсткого срока.";
      }
      container lifetime-current {
        description
          "Текущий срок действия IPsec SA. При soft-lifetime-expired
           true в этом контейнере содержатся сведения о текущем мягком
           сроке действия. Это может помочь контроллеру NSF узнать,
           какие (мягкие) ограничения были достигнуты (время, число
           байтов, число пакетов, срок бездействия).";
        uses nsfikec:lifetime;
      }
    }

    notification sadb-seq-overflow {
      if-feature "ikeless-notification";
      description
        "Уведомление о переполнении порядкового номера.";
      leaf ipsec-sa-name {
        type string;
        mandatory true;
        description
          "Уникальное имя записи SAD для IPsec SA, в которой возникло
           переполнение порядкового номера. Когда NSF выдаёт это
           событие заранее, переполнение зависит от реализации и
           выходит за рамки этой спецификации. Предполагается, что у
           контроллера I2NSF будет копия сведений о IPsec SA (без
           криптографического материала и данных состояния),
           индексированная по имени (уникальный идентификатор), чтобы
           можно было получить все сведения (криптоалгоритмы и пр.) о
           IPsec SA для замены ключей IPsec SA.";
      }
    }

    notification sadb-bad-spi {
      if-feature "ikeless-notification";
      description
        "Уведомление о получении NSF пакета с некорректным индексом
         SPI (т. е. не представленным в SAD).";
      leaf spi {
        type uint32 {
          range "0..max";
        }
        mandatory true;
        description
          "Значение SPI в ошибочном пакете IPsec.";
      }
    }
  }
  <CODE ENDS>
```

6. Взаимодействие с IANA

Агентство IANA внесло указанные ниже пространства имён в субреестр ns реестра IETF XML Registry [RFC3688]

```
URI:  urn:ietf:params:xml:ns:yang:ietf-i2nsf-ikec
Registrant Contact:  The IESG.
XML:  N/A, запрошенный URI является пространством имён XML.

URI:  urn:ietf:params:xml:ns:yang:ietf-i2nsf-ike
Registrant Contact:  The IESG.
XML:  N/A, запрошенный URI является пространством имён XML.

URI:  urn:ietf:params:xml:ns:yang:ietf-i2nsf-ikeless
Registrant Contact:  The IESG.
XML:  N/A, запрошенный URI является пространством имён XML.
```

Агентство IANA зарегистрировало указанные ниже модули YANG в реестре YANG Module Names [RFC6020]

```
Name:         ietf-i2nsf-ikec
Maintained by IANA:  N
Namespace:    urn:ietf:params:xml:ns:yang:ietf-i2nsf-ikec
Prefix:       nsfikec
Reference:    RFC 9061

Name:         ietf-i2nsf-ike
Maintained by IANA:  N
Namespace:    urn:ietf:params:xml:ns:yang:ietf-i2nsf-ike
Prefix:       nsfike
Reference:    RFC 9061

Name:         ietf-i2nsf-ikeless
Maintained by IANA:  N
Namespace:    urn:ietf:params:xml:ns:yang:ietf-i2nsf-ikeless
Prefix:       nsfikels
Reference:    RFC 9061
```

7. Вопросы безопасности

К этому документу применимы соображения безопасности SDN, отмеченные в соответствующих разделах [ITU-T.Y.3300] и [RFC7426].

С одной стороны, важно отметить, что **должна** существовать защищённая связь (security association) между контроллером I2NSF и NSF для защиты критических сведений (криптографические ключи, параметры настройки и пр.), передаваемых между этими объектами. Природа и средства организации этой связи выходят за рамки этого документа (т. е. являются частью подготовки или подключения устройства).

С другой стороны, при обязательности шифрования всего трафика NSF, принятой по умолчанию политикой **должно** быть отбрасывание пакетов (DISCARD) для предотвращения утечки нешифрованных пакетов. Эта принятая по умолчанию политика **должна** задаваться заранее в хранилище стартовой конфигурации NSF до взаимодействия NSF с контроллером I2NSF. Кроме того, в хранилище стартовой конфигурации **должны** быть заранее включены все разрешающие правила (ALLOW), чтобы элемент NSF мог взаимодействовать с контроллером I2NSF после развёртывания NSF. Этап предварительной настройки выполняется не контроллером I2NSF, а иными средствами до развёртывания NSF. Таким образом, при запуске или перезагрузке NSF всегда будет применяться стартовая конфигурация до взаимодействия с контроллером I2NSF.

Этот раздел содержит два отдельных параграфа с анализом вопросов безопасности в NSF с IKEv2 (вариант с IKE) и без IKEv2 (вариант без IKE). В общем случае контроллер I2NSF, как обычно в парадигме SDN, является целью для разных типов атак (см. [SDNSecServ] и [SDNSecurity]). Поэтому контроллер I2NSF является ключевым элементом инфраструктуры и **должен** надёжно защищаться. В частности, контроллер I2NSF будет работать с криптографическим материалом и злоумышленники могут пытаться получить доступ к этому материалу. Последствия этого зависят от выбранного варианта - с IKE или без IKE.

7.1. Вариант с IKE

В варианте с IKE контроллер I2NSF передаёт свидетельства IKEv2 (PSK, открытые и секретные ключи, сертификаты и пр.) элементам NSF, используя защищённую связь между контроллером I2NSF и NSF. Контроллеру I2NSF **недопустимо** хранить свидетельства IKEv2 после их распространения. Кроме того, NSF **недопустимо** разрешать чтение этих значений после того, как они применены контроллером I2NSF (операции write-only). Одним из вариантов является возврат при всех попытках чтения одного и того же значения (все 0).

Если злоумышленник имеет доступ к контроллеру I2NSF во время создания ключевого материала, он может получить доступ к этому материалу. Поскольку эти значения применяются при аутентификации NSF в IKEv2, он может представиться соответствующим NSF. Ниже приведены несколько важных рекомендаций.

- Конфигурациям IKEv2 **следует** соблюдать рекомендации [RFC8247].
- При использовании аутентификации PSK в IKEv2 контроллер I2NSF **должен** удалять PSK сразу после генерации и распространения.
- При использовании открытых и секретных ключей контроллер I2NSF **может** генерировать оба ключа пары. В таких случаях контроллер I2NSF **должен** удалить соответствующий секретный ключ сразу после его отправки NSF. Как вариант, NSF **может** создать секретный ключ и экспортировать в контроллер I2NSF только открытый ключ. Генерация криптографического материала (открытый и секретный ключ) в NSF и экспорт открытых ключей выходят за рамки этого документа.
- При использовании сертификатов NSF **может** создать секретный ключ и экспортировать открытый ключ контроллеру I2NSF для сертификации. Генерация криптографического материала (открытый и секретный ключ) в NSF и экспорт открытых ключей выходят за рамки этого документа.

7.2. Вариант без IKE

В варианте без IKE контроллер I2NSF передаёт в базу SAD элементов NSF сведения IPsec SA, включающие секретные сеансовые ключи, применяемые для шифрования и защиты целостности. Контроллеру I2NSF **недопустимо** хранить эти ключи после их распространения. Кроме того, элементам NSF, получающим секретный ключевой материал, недопустимо разрешать чтение этих значений каким-либо другим объектам (включая контроллер I2NSF) после их применения (операции write-only) в NSF. Тем не менее, атакующий с доступом к контроллеру I2NSF во время создания ключевого материала может эти значения получить и сможет наблюдать трафик IPsec и расшифровывать или даже изменять и заново шифровать трафик между партнёрами.

Защищённые связи между контроллером I2NSF и NSF **должны** обеспечивать уровень защиты не хуже, чем в IPsec SA, настроенных в NSF. В частности, защищённые связи между контроллером I2NSF и NSF **должны** обеспечивать полную

защиту (forward secrecy), если это обеспечивается в IPsec SA, которые контроллер I2NSF настраивает в NSF. Алгоритм шифрования, применяемый в защищённых связях между контроллером I2NSF и NSF, **должен** иметь стойкость не ниже (как минимум ключи размером 128 битов), чем у алгоритмов, применяемых в IPsec SA.

7.3. Модули YANG

Заданные этим документом модули YANG определяет схемы для данных, предназначенные для доступа через сеть с использованием протоколов управления, таких как NETCONF [RFC6241] или RESTCONF [RFC8040]. Нижним уровнем NETCONF служит защищённый транспорт с обязательной поддержкой SSH (Secure Shell) [RFC6242]. Нижним уровнем RESTCONF служит протокол HTTPS с обязательной поддержкой защиты на транспортном уровне (TLS) [RFC8446].

Модель доступа к конфигурации сети (NACM - Network Configuration Access Control Model) [RFC8341] обеспечивает возможность разрешить доступ лишь определённых пользователей NETCONF или RESTCONF к заранее заданному подмножеству операций NETCONF или RESTCONF и содержимого.

В этих модулях данных YANG определено множество узлов данных, которые разрешают запись, создание и удаление (т. е. config true, как принято по умолчанию). Эти узлы могут быть конфиденциальными или уязвимыми в некоторых сетевых средах. Запись в такие узлы (например, edit-config) без должной защиты может негативно влиять на работу сети. Ниже перечислены ветви и узлы, которые могут быть конфиденциальны или уязвимы.

Для варианта с IKE (ietf-i2nsf-ike)
/ipsec-ike

Весь контейнер в этом модуле чувствителен к операциям записи. Атакующий может добавить или изменить свидетельства, используемые для аутентификации (например, представиться NSF), корня доверия (например, поменять сертификаты доверенных CA), криптоалгоритмов (можно снизить уровень защиты), правил IPsec (например, создать утечку данных путём смены политики на разрешающую), а также обычно может изменить свидетельства и условия для IKE SA между любыми NSF.

Для варианта без IKE (ietf-i2nsf-ikeless)
/ipsec-ikeless

Весь контейнер в этом модуле чувствителен к операциям записи. Атакующий может добавить, удалить или изменить правила IPsec (например, создать утечку данных путём смены политики на разрешающую) в контейнере /ipsec-ikeless/spd, любую IPsec SA между NSF через контейнер /ipsec-ikeless/sad и обычно может изменить любую IPsec SA и правила IPsec между любыми NSF.

Некоторые из доступных для чтения узлов в модулях YANG могут быть конфиденциальны или уязвимы в той или иной сетевой среде. Важно контролировать доступ к таким объектам (например, get, get-config, notification). Ниже перечислены ветви и узлы, которые могут быть конфиденциальны или уязвимы.

Для варианта с IKE (ietf-i2nsf-ike)
/ipsec-ike/pad

Этот контейнер содержит чувствительные к операциям чтения сведения, которые **недопустимо** возвращать клиенту. Например, это может быть криптографический материал, настроенный в NSF (peer-authentication/pre-shared/secret и peer-authentication/digital-signature/private-key), уже защищённый расширением NACM default-deny-all в этом документе.

Для варианта без IKE (ietf-i2nsf-ikeless)
/ipsec-ikeless/sad/sad-entry/ipsec-sa-config/esp-sa

Этот контейнер включает симметричные ключи для IPsec SA. Например, encryption/key содержит ключ шифрования ESP, а encryption/iv - вектор инициализации (IV). В integrity/key содержится ключ защиты целостности ESP. Эти значения **недопустимо** считывать кому либо и они защищены расширением NACM default-deny-all в этом документе.

8. Литература

8.1. Нормативные документы

[IANA-Method-Type] IANA, "Method Type", <https://www.iana.org/assignments/eap-numbers/>.

[IANA-Protocols-Number] IANA, "Protocol Numbers", <https://www.iana.org/assignments/protocol-numbers/>.

[IKEv2-Auth-Method] IANA, "IKEv2 Authentication Method", <https://www.iana.org/assignments/ikev2-parameters/>.

[IKEv2-Parameters] IANA, "Internet Key Exchange Version 2 (IKEv2) Parameters", <https://www.iana.org/assignments/ikev2-parameters/>.

[IKEv2-Transform-Type-1] IANA, "Transform Type 1 - Encryption Algorithm Transform IDs", <https://www.iana.org/assignments/ikev2-parameters/>.

[IKEv2-Transform-Type-3] IANA, "Transform Type 3 - Integrity Algorithm Transform IDs", <https://www.iana.org/assignments/ikev2-parameters/>.

[IKEv2-Transform-Type-4] IANA, "Transform Type 4 - Diffie-Hellman Group Transform IDs", <https://www.iana.org/assignments/ikev2-parameters/>.

[ITU-T.X.690] International Telecommunication Union, "Information Technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", ITU-T Recommendation X.690, ISO/IEC 8825-1, February 2021.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC3947] Kivinen, T., Swander, B., Huttunen, A., and V. Volpe, "Negotiation of NAT-Traversal in the IKE", RFC 3947, DOI 10.17487/RFC3947, January 2005, <https://www.rfc-editor.org/info/rfc3947>.

[RFC3948] Huttunen, A., Swander, B., Volpe, V., DiBurro, L., and M. Stenberg, "UDP Encapsulation of IPsec ESP Packets", RFC 3948, DOI 10.17487/RFC3948, January 2005, <https://www.rfc-editor.org/info/rfc3948>.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, DOI 10.17487/RFC4301, December 2005, <https://www.rfc-editor.org/info/rfc4301>.

[RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <https://www.rfc-editor.org/info/rfc4303>.

[RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 5280, DOI 10.17487/RFC5280, May 2008, <https://www.rfc-editor.org/info/rfc5280>.

[RFC5322] Resnick, P., Ed., "Internet Message Format", RFC 5322, DOI 10.17487/RFC5322, October 2008, <https://www.rfc-editor.org/info/rfc5322>.

[RFC5915] Turner, S. and D. Brown, "Elliptic Curve Private Key Structure", RFC 5915, DOI 10.17487/RFC5915, June 2010, <https://www.rfc-editor.org/info/rfc5915>.

[RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <https://www.rfc-editor.org/info/rfc6020>.

[RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <https://www.rfc-editor.org/info/rfc6241>.

[RFC6242] Wasserman, M., "Using the NETCONF Protocol over Secure Shell (SSH)", RFC 6242, DOI 10.17487/RFC6242, June 2011, <https://www.rfc-editor.org/info/rfc6242>.

[RFC6960] Santesson, S., Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol — OCSP", RFC 6960, DOI 10.17487/RFC6960, June 2013, <https://www.rfc-editor.org/info/rfc6960>.

[RFC6991] Schoenwaelder, J., Ed., "Common YANG Data Types", RFC 6991, DOI 10.17487/RFC6991, July 2013, <https://www.rfc-editor.org/info/rfc6991>.

[RFC7296] Kaufman, C., Hoffman, P., Nir, Y., Eronen, P., and T. Kivinen, "Internet Key Exchange Protocol Version 2 (IKEv2)", STD 79, RFC 7296, DOI 10.17487/RFC7296, October 2014, <https://www.rfc-editor.org/info/rfc7296>.

[RFC7383] Smyslov, V., "Internet Key Exchange Protocol Version 2 (IKEv2) Message Fragmentation", RFC 7383, DOI 10.17487/RFC7383, November 2014, <https://www.rfc-editor.org/info/rfc7383>.

[RFC7427] Kivinen, T. and J. Snyder, "Signature Authentication in the Internet Key Exchange Version 2 (IKEv2)", RFC 7427, DOI 10.17487/RFC7427, January 2015, <https://www.rfc-editor.org/info/rfc7427>.

[RFC7619] Smyslov, V. and P. Wouters, "The NULL Authentication Method in the Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 7619, DOI 10.17487/RFC7619, August 2015, <https://www.rfc-editor.org/info/rfc7619>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[RFC8017] Moriarty, K., Ed., Kaliski, B., Jonsson, J., and A. Rusch, "PKCS #1: RSA Cryptography Specifications Version 2.2", RFC 8017, DOI 10.17487/RFC8017, November 2016, <https://www.rfc-editor.org/info/rfc8017>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8221] Wouters, P., Migault, D., Mattsson, J., Nir, Y., and T. Kivinen, "Cryptographic Algorithm Implementation Requirements and Usage Guidance for Encapsulating Security Payload (ESP) and Authentication Header (AH)", RFC 8221, DOI 10.17487/RFC8221, October 2017, <https://www.rfc-editor.org/info/rfc8221>.

[RFC8229] Pauly, T., Touati, S., and R. Mantha, "TCP Encapsulation of IKE and IPsec Packets", RFC 8229, DOI 10.17487/RFC8229, August 2017, <https://www.rfc-editor.org/info/rfc8229>.

[RFC8247] Nir, Y., Kivinen, T., Wouters, P., and D. Migault, "Algorithm Implementation Requirements and Usage Guidance for the Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 8247, DOI 10.17487/RFC8247, September 2017, <https://www.rfc-editor.org/info/rfc8247>.

[RFC8340] Bjorklund, M. and L. Berger, Ed., "YANG Tree Diagrams", BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018, <https://www.rfc-editor.org/info/rfc8340>.

[RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <https://www.rfc-editor.org/info/rfc8341>.

[RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <https://www.rfc-editor.org/info/rfc8342>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

8.2. Дополнительная литература

[IPSECME-CONTROLLER-IKE] Carrel, D. and B. Weis, "IPsec Key Exchange using a Controller", Work in Progress, Internet-Draft, draft-carrel-ipsecme-controller-ike-01, 10 March 2019, <https://datatracker.ietf.org/doc/html/draft-carrel-ipsecme-controller-ike-01>.

[ITU-T.Y.3300] International Telecommunications Union, "Y.3300: Framework of software-defined networking", June 2014, <https://www.itu.int/rec/T-REC-Y.3300/en>.

[libreswan] The Libreswan Project, "Libreswan VPN software", <https://libreswan.org/>.

[netconf-vpn] Stefan Wallin, "Tutorial: NETCONF and YANG", January 2014, <https://ripe68.ripe.net/presentations/181-NETCONF-YANG-tutorial-43.pdf>.

[ONF-OpenFlow] Open Networking Foundation, "OpenFlow Switch Specification", Version 1.4.0 (Wire Protocol 0x05), October 2013, <https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v1.4.0.pdf>.

[ONF-SDN-Architecture] Open Networking Foundation, "SDN architecture", Issue 1, June 2014, <https://www.opennetworking.org/wp-content/uploads/2013/02/TR_SDN_ARCH_1.0_06062014.pdf>.

[RFC2367] McDonald, D., Metz, C., and B. Phan, "PF_KEY Key Management API, Version 2", RFC 2367, DOI 10.17487/RFC2367, July 1998, <https://www.rfc-editor.org/info/rfc2367>.

[RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <https://www.rfc-editor.org/info/rfc3688>.

[RFC6040] Briscoe, B., "Tunnelling of Explicit Congestion Notification", RFC 6040, DOI 10.17487/RFC6040, November 2010, <https://www.rfc-editor.org/info/rfc6040>.

[RFC6071] Frankel, S. and S. Krishnan, "IP Security (IPsec) and Internet Key Exchange (IKE) Document Roadmap", RFC 6071, DOI 10.17487/RFC6071, February 2011, <https://www.rfc-editor.org/info/rfc6071>.

[RFC6437] Amante, S., Carpenter, B., Jiang, S., and J. Rajahalme, "IPv6 Flow Label Specification", RFC 6437, DOI 10.17487/RFC6437, November 2011, <https://www.rfc-editor.org/info/rfc6437>.

[RFC7149] Boucadair, M. and C. Jacquenet, "Software-Defined Networking: A Perspective from within a Service Provider Environment", RFC 7149, DOI 10.17487/RFC7149, March 2014, <https://www.rfc-editor.org/info/rfc7149>.

[RFC7426] Haleplidis, E., Ed., Pentikousis, K., Ed., Denazis, S., Hadi Salim, J., Meyer, D., and O. Koufopavlou, "Software-Defined Networking (SDN): Layers and Architecture Terminology", RFC 7426, DOI 10.17487/RFC7426, January 2015, <https://www.rfc-editor.org/info/rfc7426>.

[RFC8192] Hares, S., Lopez, D., Zarny, M., Jacquenet, C., Kumar, R., and J. Jeong, "Interface to Network Security Functions (I2NSF): Problem Statement and Use Cases", RFC 8192, DOI 10.17487/RFC8192, July 2017, <https://www.rfc-editor.org/info/rfc8192>.

[RFC8329] Lopez, D., Lopez, E., Dunbar, L., Strassner, J., and R. Kumar, "Framework for Interface to Network Security Functions", RFC 8329, DOI 10.17487/RFC8329, February 2018, <https://www.rfc-editor.org/info/rfc8329>.

[SDNSecServ] Scott-Hayward, S., O'Callaghan, G., and P. Sezer, "SDN Security: A Survey", 2013 IEEE SDN for Future Networks and Services (SDN4FNS), pp. 1-7, DOI 10.1109/SDN4FNS.2013.6702553, November 2013, <https://doi.org/10.1109/SDN4FNS.2013.6702553>.

[SDNSecurity] Kreutz, D., Ramos, F., and P. Verissimo, "Towards secure and dependable software-defined networks", Proceedings of the second ACM SIGCOMM workshop on Hot Topics in software defined networking, pp. 55-60, DOI 10.1145/2491185.2491199, August 2013, <https://doi.org/10.1145/2491185.2491199>.

[strongswan] CESNET, "strongSwan: the OpenSource IPsec-based VPN Solution", <https://www.strongswan.org/>.

[TRAN-IPSECME-YANG] Tran, K., Wang, H., Nagaraj, V. K., and X. Chen, "Yang Data Model for Internet Protocol Security (IPsec)", Work in Progress, Internet-Draft, draft-tran-ipsecme-yang-01, 18 March 2016, <https://datatracker.ietf.org/doc/html/draft-tran-ipsecme-yang-01>.

Приложение A. Пример конфигурации XML для варианта IKE

Этот пример показывает файл конфигурации XML, передаваемый контроллером I2NSF для организации IPsec SA между парой NSF (Рисунок 3) в туннельном режиме (шлюз-шлюз) с ESP, аутентификацией по сертификатам X.509 (для краткости base64encodedvalue==) и вариантом с IKE.

```
<ipsec-ike xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-ike"
```

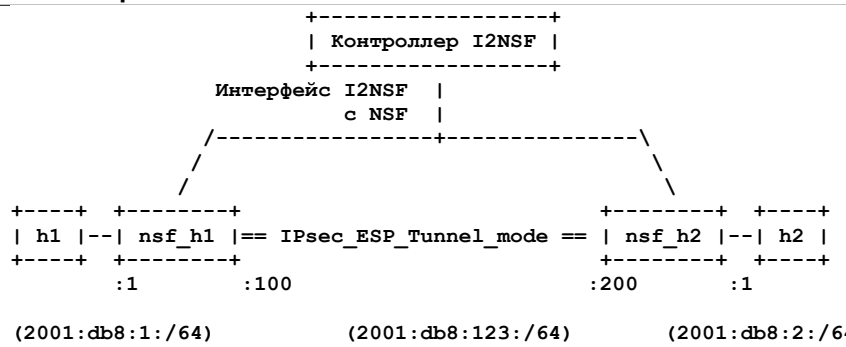

Рисунок 3. Вариант IKE, туннельный режим, аутентификация по сертификату X.509.

```
xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <pad>
    <pad-entry>
      <name>nsf_h1_pad</name>
      <ipv6-address>2001:db8:123::100</ipv6-address>
      <peer-authentication>
         <auth-method>digital-signature</auth-method>
         <digital-signature>
            <cert-data>base64encodedvalue==</cert-data>
            <private-key>base64encodedvalue==</private-key>
            <ca-data>base64encodedvalue==</ca-data>
         </digital-signature>
      </peer-authentication>
    </pad-entry>
    <pad-entry>
      <name>nsf_h2_pad</name>
      <ipv6-address>2001:db8:123::200</ipv6-address>
      <auth-protocol>ikev2</auth-protocol>
      <peer-authentication>
        <auth-method>digital-signature</auth-method>
        <digital-signature>
          <!-- Цифровая подпись RSA -->
          <ds-algorithm>1</ds-algorithm>
          <cert-data>base64encodedvalue==</cert-data>
          <ca-data>base64encodedvalue==</ca-data>
        </digital-signature>
      </peer-authentication>
    </pad-entry>
  </pad>
  <conn-entry>
     <name>nsf_h1-nsf_h2</name>
     <autostartup>start</autostartup>
     <version>ikev2</version>
     <initial-contact>false</initial-contact>
     <fragmentation><enabled>false</enabled></fragmentation>
     <ike-sa-lifetime-soft>
        <rekey-time>60</rekey-time>
        <reauth-time>120</reauth-time>
     </ike-sa-lifetime-soft>
     <ike-sa-lifetime-hard>
        <over-time>3600</over-time>
     </ike-sa-lifetime-hard>
     <!--AUTH_HMAC_SHA2_512_256-->
     <ike-sa-intr-alg>14</ike-sa-intr-alg>
     <!--ENCR_AES_CBC - 128 битов-->
     <ike-sa-encr-alg>
        <id>1</id>
     </ike-sa-encr-alg>
     <!-- 8192-битовая группа MODP-->
     <dh-group>18</dh-group>
     <half-open-ike-sa-timer>30</half-open-ike-sa-timer>
     <half-open-ike-sa-cookie-threshold>
        15
     </half-open-ike-sa-cookie-threshold>
     <local>
         <local-pad-entry-name>nsf_h1_pad</local-pad-entry-name>
     </local>
     <remote>
         <remote-pad-entry-name>nsf_h2_pad</remote-pad-entry-name>
     </remote>
     <spd>
       <spd-entry>
          <name>nsf_h1-nsf_h2</name>
          <ipsec-policy-config>
            <anti-replay-window-size>64</anti-replay-window-size>
            <traffic-selector>
               <local-prefix>2001:db8:1::0/64</local-prefix>
               <remote-prefix>2001:db8:2::0/64</remote-prefix>
               <inner-protocol>any</inner-protocol>
            </traffic-selector>
```

```
            <processing-info>
               <action>protect</action>
               <ipsec-sa-cfg>
                  <pfp-flag>false</pfp-flag>
                  <ext-seq-num>true</ext-seq-num>
                  <seq-overflow>false</seq-overflow>
                  <stateful-frag-check>false</stateful-frag-check>
                  <mode>tunnel</mode>
                  <protocol-parameters>esp</protocol-parameters>
                  <esp-algorithms>
                     <!-- AUTH_HMAC_SHA1_96 -->
                     <integrity>2</integrity>
                      <encryption>
                          <!-- ENCR_AES_CBC -->
                          <id>1</id>
                          <algorithm-type>12</algorithm-type>
                          <key-length>128</key-length>
                      </encryption>
                      <encryption>
                          <!-- ENCR_3DES-->
                          <id>2</id>
                          <algorithm-type>3</algorithm-type>
                      </encryption>
                     <tfc-pad>false</tfc-pad>
                  </esp-algorithms>
                  <tunnel>
                     <local>2001:db8:123::100</local>
                     <remote>2001:db8:123::200</remote>
                     <df-bit>clear</df-bit>
                     <bypass-dscp>true</bypass-dscp>
                  </tunnel>
               </ipsec-sa-cfg>
            </processing-info>
         </ipsec-policy-config>
      </spd-entry>
   </spd>
   <child-sa-info>
      <!-- 8192-битовая группа MODP -->
      <fs-groups>18</fs-groups>
      <child-sa-lifetime-soft>
         <bytes>1000000</bytes>
         <packets>1000</packets>
         <time>30</time>
         <idle>60</idle>
         <action>replace</action>
      </child-sa-lifetime-soft>
      <child-sa-lifetime-hard>
         <bytes>2000000</bytes>
         <packets>2000</packets>
         <time>60</time>
         <idle>120</idle>
      </child-sa-lifetime-hard>
   </child-sa-info>
  </conn-entry>
</ipsec-ike>
```

Приложение B. Пример конфигурации XML для варианта без IKE

Этот пример показывает файл конфигурации XML, передаваемый контроллером I2NSF для организации IPsec SA между парой NSF (Рисунок 4) в транспортном режиме (хост-хост) с ESP в варианте без IKE.

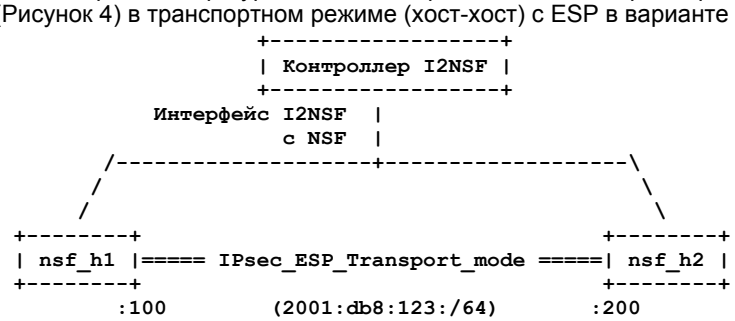

Рисунок 4. Вариант без IKE, транспортный режим.

```
<ipsec-ikeless
  xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-ikeless"
  xmlns:nc="urn:ietf:params:xml:ns:netconf:base:1.0">
  <spd>
    <spd-entry>
        <name>
           in/trans/2001:db8:123::200/2001:db8:123::100
        </name>
        <direction>inbound</direction>
        <reqid>1</reqid>
        <ipsec-policy-config>
           <traffic-selector>
             <local-prefix>2001:db8:123::200/128</local-prefix>
```

```
            <remote-prefix>2001:db8:123::100/128</remote-prefix>
            <inner-protocol>any</inner-protocol>
          </traffic-selector>
          <processing-info>
             <action>protect</action>
             <ipsec-sa-cfg>
               <ext-seq-num>true</ext-seq-num>
               <seq-overflow>false</seq-overflow>
               <mode>transport</mode>
               <protocol-parameters>esp</protocol-parameters>
               <esp-algorithms>
                  <!--AUTH_HMAC_SHA1_96-->
                  <integrity>2</integrity>
                  <!--ENCR_AES_CBC -->
                  <encryption>
                    <id>1</id>
                    <algorithm-type>12</algorithm-type>
                     <key-length>128</key-length>
                  </encryption>
                  <encryption>
                    <id>2</id>
                    <algorithm-type>3</algorithm-type>
                  </encryption>
               </esp-algorithms>
             </ipsec-sa-cfg>
          </processing-info>
        </ipsec-policy-config>
      </spd-entry>
      <spd-entry>
        <name>out/trans/2001:db8:123::100/2001:db8:123::200</name>
        <direction>outbound</direction>
        <reqid>1</reqid>
        <ipsec-policy-config>
          <traffic-selector>
            <local-prefix>2001:db8:123::100/128</local-prefix>
            <remote-prefix>2001:db8:123::200/128</remote-prefix>
            <inner-protocol>any</inner-protocol>
          </traffic-selector>
          <processing-info>
            <action>protect</action>
            <ipsec-sa-cfg>
              <ext-seq-num>true</ext-seq-num>
              <seq-overflow>false</seq-overflow>
              <mode>transport</mode>
              <protocol-parameters>esp</protocol-parameters>
              <esp-algorithms>
                <!-- AUTH_HMAC_SHA1_96 -->
                <integrity>2</integrity>
                <!-- ENCR_AES_CBC -->
                <encryption>
                   <id>1</id>
                   <algorithm-type>12</algorithm-type>
                   <key-length>128</key-length>
                </encryption>
                <encryption>
                   <id>2</id>
                   <algorithm-type>3</algorithm-type>
                </encryption>
              </esp-algorithms>
             </ipsec-sa-cfg>
          </processing-info>
        </ipsec-policy-config>
      </spd-entry>
    </spd>
    <sad>
      <sad-entry>
        <name>out/trans/2001:db8:123::100/2001:db8:123::200</name>
        <reqid>1</reqid>
        <ipsec-sa-config>
          <spi>34501</spi>
          <ext-seq-num>true</ext-seq-num>
          <seq-overflow>false</seq-overflow>
          <anti-replay-window-size>64</anti-replay-window-size>
          <traffic-selector>
            <local-prefix>2001:db8:123::100/128</local-prefix>
            <remote-prefix>2001:db8:123::200/128</remote-prefix>
               <inner-protocol>any</inner-protocol>
           </traffic-selector>
           <protocol-parameters>esp</protocol-parameters>
           <mode>transport</mode>
           <esp-sa>
             <encryption>
                <!-- //ENCR_AES_CBC -->
                <encryption-algorithm>12</encryption-algorithm>
                <key>01:23:45:67:89:AB:CE:DF</key>
                <iv>01:23:45:67:89:AB:CE:DF</iv>
```

```
                </encryption>
                <integrity>
                   <!-- //AUTH_HMAC_SHA1_96 -->
                   <integrity-algorithm>2</integrity-algorithm>
                   <key>01:23:45:67:89:AB:CE:DF</key>
                </integrity>
             </esp-sa>
          </ipsec-sa-config>
       </sad-entry>
       <sad-entry>
          <name>in/trans/2001:db8:123::200/2001:db8:123::100</name>
          <reqid>1</reqid>
          <ipsec-sa-config>
              <spi>34502</spi>
              <ext-seq-num>true</ext-seq-num>
              <seq-overflow>false</seq-overflow>
              <anti-replay-window-size>64</anti-replay-window-size>
              <traffic-selector>
                 <local-prefix>2001:db8:123::200/128</local-prefix>
                 <remote-prefix>2001:db8:123::100/128</remote-prefix>
                 <inner-protocol>any</inner-protocol>
              </traffic-selector>
              <protocol-parameters>esp</protocol-parameters>
              <mode>transport</mode>
              <esp-sa>
                 <encryption>
                    <!-- //ENCR_AES_CBC -->
                    <encryption-algorithm>12</encryption-algorithm>
                    <key>01:23:45:67:89:AB:CE:DF</key>
                    <iv>01:23:45:67:89:AB:CE:DF</iv>
                 </encryption>
                 <integrity>
                    <!-- //AUTH_HMAC_SHA1_96 -->
                    <integrity-algorithm>2</integrity-algorithm>
                    <key>01:23:45:67:89:AB:CE:DF</key>
                 </integrity>
              </esp-sa>
              <sa-lifetime-hard>
                 <bytes>2000000</bytes>
                 <packets>2000</packets>
                 <time>60</time>
                 <idle>120</idle>
              </sa-lifetime-hard>
              <sa-lifetime-soft>
                 <bytes>1000000</bytes>
                 <packets>1000</packets>
                 <time>30</time>
                 <idle>60</idle>
                 <action>replace</action>
              </sa-lifetime-soft>
          </ipsec-sa-config>
       </sad-entry>
    </sad>
</ipsec-ikeless>
```

Приложение C. Примеры уведомлений XML

Ниже показаны несколько файлов XML, иллюстрирующих разные типы уведомлений, определённые в модели данных YANG без IKE, которые передаются элементами NSF контроллеру I2NSF. Эти уведомления передаются в варианте без IKE.

```
<sadb-expire xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-ikeless">
<ipsec-sa-name>in/trans/2001:db8:123::200/2001:db8:123::100
</ipsec-sa-name>
    <soft-lifetime-expire>true</soft-lifetime-expire>
       <lifetime-current>
          <bytes>1000000</bytes>
          <packets>1000</packets>
          <time>30</time>
          <idle>60</idle>
       </lifetime-current>
</sadb-expire>
```

Рисунок 5. Пример уведомления sadb-expire.

```
<sadb-acquire xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-ikeless">
    <ipsec-policy-name>in/trans/2001:db8:123::200/2001:db8:123::100
    </ipsec-policy-name>
    <traffic-selector>
        <local-prefix>2001:db8:123::200/128</local-prefix>
        <remote-prefix>2001:db8:123::100/128</remote-prefix>
        <inner-protocol>any</inner-protocol>
         <local-ports>
              <start>0</start>
              <end>0</end>
         </local-ports>
         <remote-ports>
```

```
                <start>0</start>
                <end>0</end>
            </remote-ports>
        </traffic-selector>
    </sadb-acquire>
```

Рисунок 6. Пример уведомления sadb-acquire

```
    <sadb-seq-overflow
        xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-ikeless">
          <ipsec-sa-name>in/trans/2001:db8:123::200/2001:db8:123::100
          </ipsec-sa-name>
    </sadb-seq-overflow>
```

Рисунок 7. Пример уведомления sadb-seq-overflow.

```
    <sadb-bad-spi
            xmlns="urn:ietf:params:xml:ns:yang:ietf-i2nsf-ikeless">
           <spi>666</spi>
    </sadb-bad-spi>
```

Рисунок 8. Пример уведомления sadb-bad-spi.

Приложение D. Примеры использования

D.1. Пример организации IPsec SA

Это приложение иллюстрирует применимость вариантов с IKE и без IKE в традиционных конфигурациях IPsec хост-хост и шлюз-шлюз. В примерах предполагается наличие пары NSF, которым нужно организовать сквозную связь IPsec SA для защиты своего взаимодействия. Оба элемента NSF могут быть хостами, которые обмениваются трафиком, или шлюзами, например, для соединения двух филиалов.

Применимость этих конфигураций проявляется в текущих и новых сетевых сценариях. Например, технологии SD-WAN обеспечивают динамические соединения VPN по запросу между филиалами и облачными службами SaaS[1]. Кроме того, службы предоставления инфраструктуры как услуги (Infrastructure as a Service или IaaS) обеспечивают среды виртуализации и развёртывания, которые часто используют IPsec для организации защищённых каналов между виртуальными экземплярами (хост-хост) и предоставления услуг VPN для виртуализованных сетей (шлюз-шлюз).

Как показано ниже, основанная на I2NSF система управления IPsec (для вариантов с IKE и без IKE) обеспечивает ряд преимуществ.

1. Возможность создавать IPsec SA между NSF, основываясь лишь на применении общих правил защиты на основе потоков у пользователей I2NSF. Таким образом, администраторы могут управлять всеми защищёнными связями из центральной точки с абстрактным представлением сети.
2. NSF в системе не требуют ручной настройки, что позволяет автоматизировать развёртывание.

D.1.1. Вариант с IKE

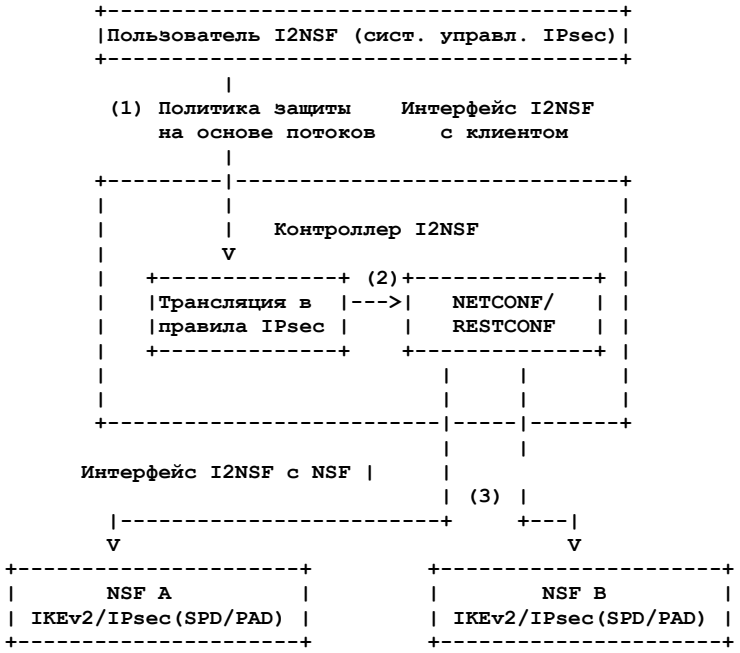

Рисунок 9. Взаимодействие между хостами и между шлюзами в варианте с IKE.

На рисунке 9 показано применение варианта с IKE для защиты данных на пути между NSF A и NSF B

1. Пользователь I2NSF задаёт общие правила защиты на основе потоков (например, защищать трафик данных между NSF A и B). Контроллер I2NSF находит вовлечённые Controller NSF (NSF A и NSF B).
2. Контроллер I2NSF генерирует свидетельства IKEv2 и транслирует правила в записи SPD и PAD.

[1]Software as a Service - программы как услуги.

3. Контроллер I2NSF устанавливает конфигурацию IKEv2, включающую записи SPD и PAD в оба NSF A и B. При отказе какой-либо операции в NSF A или NSF B контроллер останавливает процесс и выполняет откат, удаляя конфигурацию IKEv2, SPD и PAD, установленную в NSF A или B.

Если предыдущие этапы успешны, между NSF A и NSF B организуется поток, защищённый IPsec SA с IKEv2.

D.1.2. Вариант без IKE

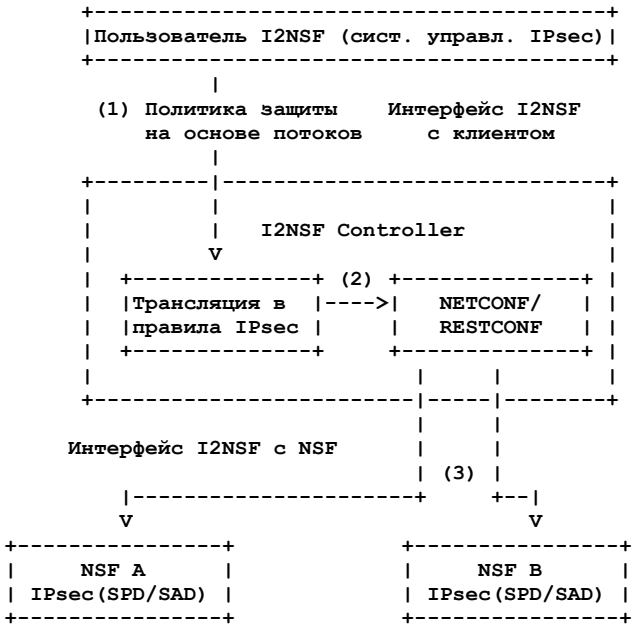

Рисунок 10. Взаимодействие между хостами и между шлюзами в варианте без IKE.

На рисунке 10 показано применение варианта без IKE для защиты данных на пути между NSF A и NSF B.

1. Пользователь I2NSF организует базовую политику защиты данных на основе потоков, а контроллер I2NSF находит вовлечённые NSF.

2. Контроллер I2NSF транслирует правила основанной на потоках защиты в записи IPsec SPD и SAD.

3. Контроллер помещает записи в базы данных IPsec NSF A и NSF B (SPD и SAD). Последующие действия указаны ниже.

   - Контроллер I2NSF выбирает два случайных значения индексов SPI, например, SPIa1 для входящей IPsec SA в NSF A и SPIb1 для входящей IPsec SA в NSF B. Значению SPIa1 **недопустимо** совпадать с каким-либо из входящих SPI в A, а SPIb1 **недопустимо** совпадать с каким-либо из входящих SPI в B. Кроме того, индекс SPIa1 **должен** применяться в B для исходящей IPsec SA к A, а SPIb1 **должен** применяться в A для исходящей IPsec SA к B. Генерируется также свежий криптографический материал для новых входящих и исходящих IPsec SA и их параметры.

   - После этого контроллер I2NSF одновременно передаёт новую входящую IPsec SA с SPIa1 и новую исходящую IPsec SA с SPIb1 элементу NSF A, а новую входящую IPsec SA с SPIb1 и новую исходящую IPsec SA с SPIa1 элементу B вместе с соответствующими правилами IPsec.

   - После получения контроллером I2NSF подтверждений от NSF A и NSF B он знает, что IPsec SA корректно установлены и готовы к работе.

   В другом варианте этой операции контроллер I2NSF сначала передаёт правила IPsec и новые входящие IPsec SA элементам A и B. После подтверждения успеха этих операций от NSF A и NSF B, контроллер устанавливает новые исходящие IPsec SA. Это может увеличивать длительность процесса, но позволяет передавать трафик через сеть до полного завершения организации IPsec SA. **Возможны** и другие варианты выполнения этапа 3.

4. При отказе какой-либо из предшествующих операций (например, NSF A указывает ошибку при попытке контроллера I2NSF установить запись SPD или создать новые IPsec SA) контроллер I2NSF **должен** выполнить откат операций, удаляя все новые IPsec SA (входящие и исходящие) и записи SPD, которые были установлены в каждом из NSF, и останавливает процесс. Отметим, что контроллер I2NSF **может** повторять попытки несколько раз.

5. Если этапы 1 - 3 успешны, поток между NSF A и NSF B будет защищён IPsec SA, организованными контроллером I2NSF. Следует отметить, что контроллер I2NSF связывает сроки действия новых IPsec SA. По истечении срока действия NSF передаёт уведомление sadb-expire контроллеру I2NSF для запуска процесса замены ключей.

Вместо установки правил IPsec (в SPD) и IPsec SA (в SAD) на этапе 3 (упреждающий режим) контроллер I2NSF может устанавливать на этапе 3 только записи SPD (реактивный режим). В этом случае при необходимости защитить пакет с помощью IPsec элемент NSF, который первым увидел пакет данных, передаёт уведомление sadb-acquire для информирования контроллера I2NSF о потребности в записях SAD с IPsec SA для обработки этого пакета. При отказе какой-либо операции установки входящей или исходящей IPsec SA контроллер I2NSF останавливает процесс и выполняет операции отката, удаляя все вновь установленные SA.

D.2. Пример смены ключей в варианте без IKE

Для пояснения процесса смены ключей между парой IPsec NSF A и B предположим, что SPIa1 указывает входящую IPsec SA в A, SPIb1 - входящую IPsec SA в B.

1. Контроллер I2NSF выбирает два случайных значения SPI для новых входящих IPsec SA, например, SPIa2 для IPsec SA в A и SPIb2 для IPsec SA в B. Значению SPIa1 **недопустимо** совпадать в каким-либо входящим SPI в A, а значению SPIb1 MUST NOT **недопустимо** совпадать в каким-либо входящим SPI в B. Затем контроллер I2NSF создаёт входящую IPsec SA с SPIa2 в A и входящую IPsec SA в B с SPIb2. Контроллер может передать информацию одновременно в A и B.
2. Когда контроллер I2NSF получит подтверждения от A и B, он будет знать, что входящие IPsec SA установлены корректно. Затем контроллер параллельно передаёт в A и B исходящие IPsec SA - исходящую IPsec SA в A с SPIb2 и исходящую IPsec SA в B с SPIa2. После этого новые IPsec SA готовы к работе.
3. Когда контроллер I2NSF получить подтверждения от A и B об установке исходящих IPsec SA, он параллельно удаляет старые IPsec SA в A (входящая SPIa1 и исходящая SPIb1) и B (исходящая SPIa1 и входящая SPIb1).

При отказе какой-либо операции на этапе 1 (например, NSF A сообщает об ошибке при попытке контроллера I2NSF установить новую входящую IPsec SA) контроллер I2NSF **должен** выполнить операции отката, удаляя все новые входящие SA, установленные на этапе 1.

Если этап 1 выполнен, но произошёл отказ на этапе 2 (например, NSF A сообщает об ошибке при попытке контроллера I2NSF установить новую исходящую IPsec SA) контроллер I2NSF **должен** выполнить операции отката, удаляя все новые исходящие SA, установленные на этапе 2 и входящие SA, установленные на этапе 1 (в указанном порядке).

Если этапы 1 и 2 выполнены, но произошёл отказ на этапе 3, контроллер I2NSF не будет выполнять откат для операций этапов 1 и 2, поскольку новые действительные IPsec SA были созданы и готовы к работе. Контроллер I2NSF **может** повторить попытку удаления старых входящих и исходящих IPsec SA в NSF A и NSF B несколько раз, пока не будет достигнут успех. В крайнем случае старые IPsec SA будут удалены по завершению их срока действия.

D.3. Пример контроля потери состояния NSF в варианте без IKE

В варианте без IKE при обнаружении контроллером I2NSF потери элементом NSF состояния IPsec он выполняет указанные ниже действия.

1. Контроллеру I2NSF **следует** удалить старые IPsec SA на узлах без отказов и организовать новые с отказавшим узлом. Это предотвратит утечку нешифрованных данных с узлов без отказов.
2. Если затронутый узел перезапускается, контроллер I2NSF настраивает новые входящие IPsec SA между этим узлом и всеми узлами, с которыми он взаимодействовал.
3. После организации входящих IPsec SA контроллер I2NSF настраивает исходящие IPsec SA в параллель.

Этапы 2 и 3 могут выполняться одновременно с возможной потерей пакетов. Если возможность потерь не критична, такая оптимизация сокращает число обменов между контроллером I2NSF и элементами NSF.

Благодарности

Авторы благодарят Paul Wouters, Valery Smyslov, Sowmini Varadhan, David Carrel, Yoav Nir, Tero Kivinen, Martin Bjorklund, Graham Bartlett, Sandeep Kampati, Linda Dunbar, Mohit Sethi, Martin Bjorklund, Tom Petch, Christian Hopps, Rob Wilton, Carlos J. Bernardos, Alejandro Perez-Mendez, Alejandro Abad-Carrascosa, Ignacio Martinez, Ruben Ricart и всех членов IESG, просматривавших документ, за их ценные замечания.

Адреса авторов

Rafa Marin-Lopez
University of Murcia
Faculty of Computer Science
Campus de Espinardo S/N
30100 Murcia
Spain
Phone: +34 868 88 85 01
Email: rafa@um.es

Gabriel Lopez-Millan
University of Murcia
Faculty of Computer Science
Campus de Espinardo S/N
30100 Murcia
Spain
Phone: +34 868 88 85 04
Email: gabilm@um.es

Fernando Pereniguez-Garcia
University Defense Center
Spanish Air Force Academy
MDE-UPCT
30720 San Javier Murcia
Spain
Phone: +34 968 18 99 46
Email: fernando.pereniguez@cud.upct.es

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru