Internet Engineering Task Force (IETF)
Request for Comments: 7011
STD: 77
Obsoletes: 5101
Category: Standards Track
ISSN: 2070-1721

B. Claise, Ed.
Cisco Systems, Inc.
B. Trammell, Ed.
ETH Zurich
P. Aitken
Cisco Systems, Inc.

September 2013

Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information

Спецификация протокола IPFIX для обмена информацией о потоках

Аннотация

Этот документ задаёт протокол экспорта сведений о потоках IP (IP Flow Information Export илиIPFIX), служащий для передачи информации о потоках трафика через сеть. Для передачи сведений Traffic Flow от экспортирующего процесса сборщику (Collecting Process) требуется общее представление потоков данных и стандарт для их передачи. Этот документ описывает как данные и шаблоны IPFIX передаются по разным транспортным протоколам IPFIX Exporting Process к IPFIX Collecting Process. Этот документ заменяет RFC 5101.

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 5741.

Информацию о текущем статусе документа, ошибках и способах обратной связи можно найти по ссылке http://www.rfc-editor.org/info/rfc7011.

Авторские права

Авторские права (Copyright (c) 2013) принадлежат IETF Trust и лицам, указанным в качестве авторов документа. Все права защищены.

К документу применимы права и ограничения, указанные в BCP 78 и IETF Trust Legal Provisions и относящиеся к документам IETF (http://trustee.ietf.org/license-info), на момент публикации данного документа. Прочтите упомянутые документы внимательно. Фрагменты программного кода, включённые в этот документ, распространяются в соответствии с упрощённой лицензией BSD, как указано в параграфе 4.e документа IETF Trust Legal Provisions, без каких-либо гарантий (как указано в Simplified BSD License).

Оглавление

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

1. Введение

Трафик в сети передачи данных можно считать состоящим из потоков, проходящих через элементы сети. Для администрирования и иных целей часто бывает интересно, полезно или даже необходимо иметь доступ к сведениям о потоках, проходящих через элементы сети. Процесс сбора (Collecting Process) должен быть способен получать данные

о потоках, передаваемые через множество элементов сети. Это требует единообразного метода для представления информации и её передачи от сетевых элементов в точку сбора (коллектор). Данный документ задаёт протокол для выполнения этих требований. Детально описано представление различных потоков, а также дополнительные данные, требуемые для интерпретации, форматы пакетов, используемые транспортные механизмы, вопросы безопасности и др.

1.1. Отличия от RFC 5101

Этот документ отменяет выпуск Proposed Standard спецификации протокола IPFIX [RFC5101]. Заданный здесь протокол способен взаимодействовать с протоколом [RFC5101]. Ниже указаны отличия данного документа от предшественника.

- Исправлены все технические и редакционные ошибки в [RFC5101].
- Поскольку реестр [IANA-IPFIX] сейчас является нормативным для всех определений информационных элементов (см. [RFC7012]), определения информационных элементов в разделе 4 заменены ссылками на этот реестр.
- Уточнено кодирование типов данных dateTimeSeconds, dateTimeMilliseconds, dateTimeMicroseconds и dateTimeNanoseconds, а также связанно с ним кодирование поля IPFIX Message Header Export Time, особенно в части эпохи, на которой основаны типы данных для временных меток.
- Добавлен параграф 5.2. Поддержка перехода временных меток через максимум, описывающий достижению максимума в полях временных меток.
- Уточнено кодирование, особенно в разделе 6. Связь с информационной моделью, указанием использования сетевого порядка байтов для всех значений IPFIX.
- Упрощено управление шаблонами (раздел 8. Управление шаблонами). Спецификация смягчена по сравнению с [RFC5101], особенно в части отказов при повторном использовании Template ID. Исключены ненужные крайние случаи при управлении шаблонами. Новый язык управления шаблонами совместим с языком [RFC5101], насколько поведение было определено в прежней спецификации.
- Параграф 11.3. Взаимная проверка подлинности был улучшен со ссылками на современную практику взаимной аутентификации в TLS, исключены ссылки на Punycode, поскольку это описано в [RFC6125].
- Внесены редакторские правки, включая изменение структуры разделов 8 - 10 для удобочитаемости. Общее для всех протоколов поведение описано отдельно с указанием исключений для каждого транспорта. Языки управления шаблонами унифицированы в разделе 8. Управление шаблонами.
- Добавлен раздел 12. Вопросы управления.

1.2. Обзор документов IPFIX

Протокол IPFIX обеспечивает сетевым администраторам доступ к информации IP Flow. Архитектура экспорта сведений IP Flow из процесса экспорта (Exporting Process) в процесс-коллектор (Collecting Process) определена в [RFC5470] в соответствии с требованиями [RFC3917]. Этот документ определяет как записи (Record) и шаблоны (Template) IPFIX передаются по разным транспортным протоколам из процессов экспорта в коллекторы.

В настоящее время определены 4 оптимизации и расширения IPFIX: метод экономии полосы для протокола IPFIX [RFC5473], эффективный метод экспорта двухсторонних потоков [RFC5103], метод определения и экспорта комплексных структур данных [RFC6313] и спецификация протокола для посредников (IPFIX Mediator) [IPFIX-MED-PROTO] на основе IPFIX Mediation Framework [RFC6183].

Основанный на файлах транспорт для IPFIX, определяющий, как сообщения IPFIX можно записать в файлы для рабочих процессов на основе документов и архивирования, рассматривается в [RFC5655].

Формальное описание информационных элементов IPFIX (IPFIX Information Element) - их имён, типов данных и дополнительной семантики - дано в [RFC7012]. Реестр элементов поддерживает IANA [IANA-IPFIX]. Встроенный (inline) экспорт сведений о типах информационных элементов задан в [RFC5610].

Модель выбора пакетов и отчётов [RFC5474] позволяет элементам сети выбирать подмножество пакетов статистически или иным методом и экспортировать поток отчётов по выбранным пакетам в коллектор. Набор методов отбора пакетов, стандартизованных протоколом выборки (Packet Sampling или PSAMP), описан в [RFC5475]. Протокол PSAMP [RFC5476], использующий IPFIX как протокол экспорта, задаёт экспорт сведений о пакетах из процесса экспорта PSAMP (Exporting Process) в коллектор PSAMP. Вместо экспорта отчётов о пакетах (PSAMP Packet Report) входными данными для генерации записей о потоках IPFIX (Flow Record) может служить поток выбранных пакетов. Как и IPFIX, протокол PSAMP имеет формальное описание информационных элементов (имена, типы, дополнительная семантика). Информационная модель PSAMP определена в [RFC5477].

В [RFC6615] задан модуль MIB для мониторинга, а в [RFC6728] - модель данных для настройки и мониторинга совместимых с IPFIX и PSAMP сетевых устройств по протоколу настройки сети (Network Configuration Protocol или NETCONF). В [RFC6727] задан модуль PSAMP MIB как расширение модуля IPFIX SELECTOR MIB из [RFC6615].

В части разработки [RFC5153] содержит рекомендации по реализации и применению протокола IPFIX, а [RFC5471] - рекомендации по тестированию. В [RFC5472] описано, какие типы приложений могут использовать протокол IPFIX и предоставляемую информацию, а также связь модели IPFIX с другими моделями и архитектурой.

2. Терминология

Ключевые слова **необходимо** (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 RFC 2119 [RFC2119].

Определения базовых терминов, таких как Traffic Flow, Exporting Process, Collecting Process, Observation Points и т. п., семантически идентичны применяемым в документе с требованиями к IPFIX [RFC3917]. Некоторые термины описаны более подробно для более чёткого определения протокола. Определены дополнительные термины, нужные для протокола. Определения данного документа и [RFC5470] эквивалентны, определения, связанные лишь с протоколом IPFIX, представлены только здесь. Сводка терминов в параграфе 2.1 даёт краткий обзор связей между терминами.

Observation Point - точка наблюдения

Место в сети, где можно наблюдать пакеты, например, линия, к которой подключён датчик, общая среда (такая как ЛВС Ethernet), порт маршрутизатора или набор интерфейсов маршрутизатора (физических или логических).

Отметим, что каждая точка наблюдения связана с доменом наблюдения (см. ниже) и одна точка наблюдения может быть множеством из нескольких других Observation Point. Например, точкой наблюдения может быть линейная плата, включающая набор других точек наблюдения на своих интерфейсах.

Observation Domain - домен наблюдения

Наибольший набор (множество) точек наблюдения, для которых сведения о потоках могут агрегироваться измерительным процессом. Например, доменом наблюдения может быть линейная плата маршрутизатора, если она включает несколько интерфейсов, каждый из которых является точкой наблюдения. В генерируемые доменом сообщения IPFIX включается идентификатор домена (Observation Domain ID), который уникален в масштабе процесса экспорта (Exporting Process). Таким образом, процесс сбора может идентифицировать домен наблюдения от экспортёра, передающего сообщения IPFIX. Каждая точка наблюдения связана с Observation Domain. **Рекомендуется** обеспечивать уникальность идентификаторов домена также на уровне устройства IPFIX.

Packet Treatment - обработка (трактовка) пакетов

Действия с пакетом, выполняемые устройством пересылки или иным промежуточным устройством, включая пересылку, отбрасывание, задержку в целях управления трафиком и т. п.

Traffic Flow или Flow - поток трафика или поток

В сообществе Internet используется несколько определений термина «поток». В контексте IPFIX здесь используется приведённое ниже определение.

Потоком считается набор пакетов или кадров, проходящих через точку наблюдения в сети за некий интервал времени. Все пакеты одного потока имеют общий набор свойств и каждое свойство определяется результатом применения функции к значениям:

1. одного или нескольких полей пакета (например, IP-адрес получателя), полей транспортного заголовка (например, номер целевого порта) или полей прикладного заголовка (например, RTP [RFC3550]);
2. одного или нескольких параметров самого пакета (например, число меток MPLS);
3. одного или нескольких полей, выведенных при обработке (Treatment) пакета (например, next-hop IP, выходной интерфейс и т. п.).

Пакет считается принадлежащим потоку, если он имеет все свойства, заданные для потока.

Отметим, что набор пакетов потока может быть пустым, т. е. поток может не включать пакетов. Поскольку выборка является обработкой пакетов, это определение включает пакеты, выбранные механизмом отбора.

Flow Key - ключ потока

Каждое из полей, которое соответствует любому из критериев:

1. относится к заголовку пакета (например, IP-адрес получателя);
2. является свойством самого пакета (например, размер пакета);
3. выводится при обработке пакета (например, номер автономной системы AS)

и применяется для определения потока (т. е. является общим свойством всех пакетов потока), называется ключом потока. Например, традиционный квинтет (5-tuple) из IP-адресов отправителя и получателя, номеров портов отправителя и получателя, а также транспортного протокола, группирует все пакеты, относящиеся к одному из направлений передачи на одном сокете.

Flow Record - запись о потоке

Запись о потоке содержит сведения о конкретном потоке, наблюдаемом в Observation Point. Запись содержит измеренные свойства потока (например, общее число байтов во всех пакетах потока) и обычно включает характеристические свойства потока (например, IP-адрес источника).

Metering Process - процесс измерения

Процесс измерения генерирует записи о потоках (Flow Record) на основе заголовков пакетов, характеристик и обработки (Packet Treatment) в одной или нескольких точках наблюдения.

Metering Process состоит из набора функций, включающих извлечения заголовков пакетов, временные метки, выборку, классификацию и поддержку записей о потоках. Поддержка записей о потоках может включать создание новых и обновление имеющихся записей, расчёт статистики потока, обнаружение завершения срока действия потока, передачу записей процессу экспорта и удаление записей.

Exporting Process - процесс экспорта

Процесс экспорта передаёт сообщения IPFIX одному или нескольким процессам сбора (Collecting Process). Записи о потоках в сообщении генерируются одним или несколькими процессами измерения.

Exporter - экспортёр

Устройство, на котором размещён один или несколько процессов экспорта.

IPFIX Device - устройство IPFIX

Устройство IPFIX содержит в себе по меньшей мере один процесс экспорта и может включать дополнительные процессы экспорта, а также произвольное число точек наблюдения и измерительных процессов.

Collecting Process - процесс сбора

Collecting Process получает сообщения IPFIX от одного или нескольких процессов экспорта. Процесс сбора может сохранять или обрабатывать записи о потоках из этих сообщений, но эти действия выходят за рамки документа.

Collector - коллектор (сборщик)

Устройство, на котором размещён по меньшей мере процесс сбора.

Template - шаблон

Упорядоченная последовательность пар <тип, размер>, служащая для полного задания структуры и семантики конкретного набора сведений, который нужно передать от устройства IPFIX коллектору. Каждый шаблон однозначно указывается идентификатором Template ID.

IPFIX Message - сообщение IPFIX

Сообщение, исходящее от процесса экспорта и содержащее записи IPFIX от этого процесса. Получателем сообщения является процесс сбора. Сообщения IPFIX инкапсулируются на транспортном уровне.

Message Header - заголовок сообщения
Первая часть сообщения IPFIX, обеспечивающая базовые сведения о сообщении, такие как версия IPFIX, размер и порядковый номер сообщения и т. п.

Template Record - шаблонная запись
Шаблонная запись определяет структуру и интерпретацию полей Data Record.

Data Record - запись с данными
Запись, содержащая значения параметров, соответствующих шаблону (Template Record).

Options Template Record - запись с шаблоном опций
Шаблонная запись, определяющая структуру и интерпретацию полей Data Record, включая определение области применения Data Record.

Set - набор
Набор записей, имеющих похожую структуру, с заголовком-префиксом. В сообщении IPFIX после заголовка может (но не обязано) присутствовать несколько наборов (Set). Имеется три типа наборов: Template Set, Options Template Set, Data Set.

Template Set - набор шаблонов
Набор из одной или нескольких шаблонных записей, собранных вместе в сообщении IPFIX.

Options Template Set - набор шаблонов опций
Набор из одной или нескольких записей Options Template, собранных вместе в сообщении IPFIX.

Data Set - набор данных
Одна или несколько записей Data Record одного типа, собранных вместе в сообщении IPFIX. Каждая запись ранее определена через Template Record или Options Template Record.

Information Element - информационный элемент
Независимое от протокола и кодирования описание атрибута, которое может присутствовать в записи IPFIX. Информационные элементы заданы в реестре IANA IPFIX Information Elements [IANA-IPFIX]. Связанный с информационным элементом тип задаёт ограничения на содержимое элемента и определяет допустимые механизмы кодирования при использовании в IPFIX.

Transport Session - транспортная сессия
В протоколе управления потоковой передачей (Stream Control Transmission Protocol или SCTP) транспортные сессии называют ассоциациями SCTP. Ассоциацию однозначно указывают конечные точки SCTP [RFC4960]. В TCP транспортные сессии называют соединения TCP и они однозначно указываются комбинацией адресов IP и номеров портов TCP. В UDP транспортные сессии называют сеансами (сессиями) UDP и они однозначно указываются комбинацией адресов IP и номеров портов UDP.

2.1. Сводка терминов

На рисунке A показана сводка терминов IPFIX и связей между ними.

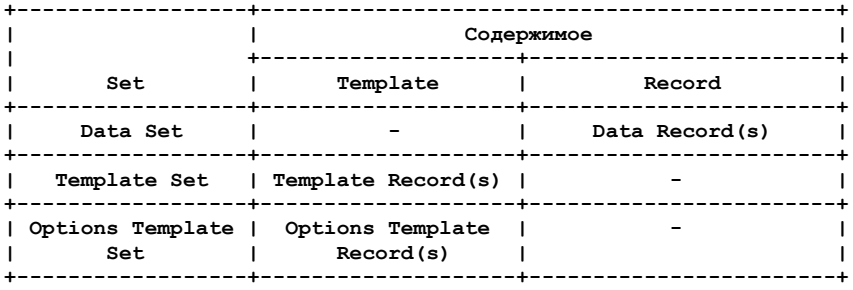


Рисунок A. Сводная таблица терминов.

Data Set состоит из записей Data Record без включения шаблонов (Template Record). Записи Data Record определяются Template Record или Options Template Record.

Набор шаблонов (Template Set) содержит лишь записи Template Record.

Набор шаблонов опций (Options Template Set) содержит только записи Options Template Record.

3. Формат сообщений IPFIX

Сообщение IPFIX состоит из заголовка Message Header, за которым могут следовать наборы Set, любого из 3 возможных типов Data Set, Template Set, Options Template Set. Формат сообщения IPFIX показан на рисунке B.

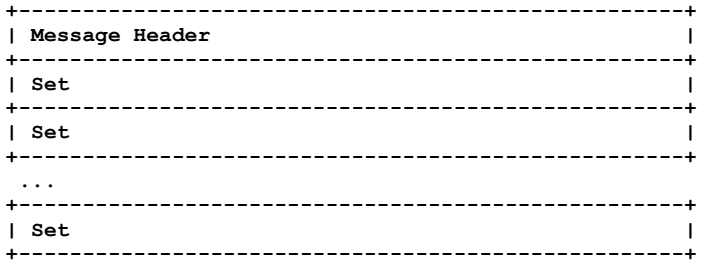


Рисунок B. Формат сообщения IPFIX.

Рассмотрим примеры сообщений IPFIX.

1. IPFIX Message с чередующимися Template, Data, Options Template Sets, как показано на рисунке C. Наборы Template и Options Template передаются «по запросу» перед первым Data Set, структуру которого они задают.

2. Сообщение, содержащее лишь наборы данных, передаётся после указания и передачи процессу сборки подходящих Template Record, как показано на рисунке D.

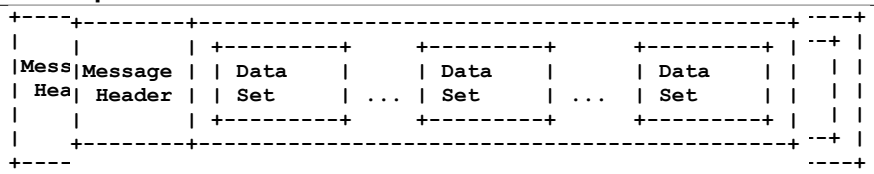


Рисунок D. Сообщение IPFIX, пример 2.
Рисунок C. Сообщение IPFIX, пример 1.

3. Сообщение, содержащее только наборы Template и Options Template, как показано на рисунке E. Такие сообщения служат для массового определения и переопределения Template и Options Template.

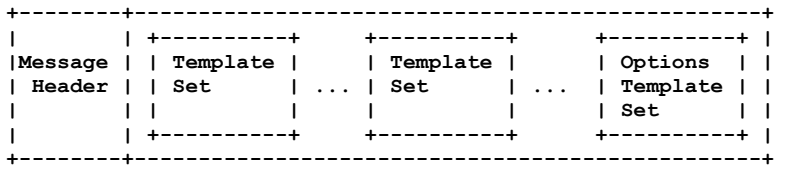


Рисунок E. Сообщение IPFIX, пример 3.

3.1. Формат заголовка сообщения

Формат заголовка сообщений IPFIX показан на рисунке F.

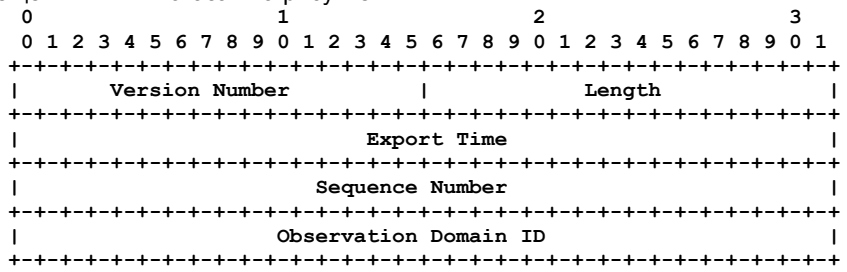


Рисунок F. Формат заголовка сообщения IPFIX.

Каждое поле Message Header экспортируется с сетевым порядком байтов. Поля заголовка описаны ниже.

Version
Версия IPFIX, которой соответствует это сообщение. Поле имеет значение 0x000a для текущей версии, будучи на 1 больше номера версии в службе экспорта NetFlow версии 9 [RFC3954].

Length
Общий размер заголовка IPFIX Message в октетах с учётом самого заголовка и включённых наборов (Set).

Export Time
Время, когда заголовок сообщения IPFIX покинул экспортёра, указанное в секундах с начала эпохи UNIX (1 января 1970, 00:00 UTC) 32-битовым целым числом без знака.

Sequence Number
Увеличивающийся порядковый номер с модулем 2^{32}, учитывающий все IPFIX Data Record, переданные в текущем потоке из текущего домена наблюдения процессом экспорта до получения этого сообщения IPFIX. Каждый поток SCTP учитывает порядковые номера раздельно, а номера в соединениях TCP и сессиях UDP относятся к одному потоку. Это значение может применяться процессом сбора для обнаружения пропуска записей IPFIX Data Record. Записи Template и Options Template не увеличивают Sequence Number.

Observation Domain ID
32-битовый идентификатор домена наблюдения, уникальный в масштабе локального процесса экспорта. Этот процесс использует Observation Domain ID для однозначного указания процессу сбора домена наблюдений, измеряющего потоки. **Рекомендуется** обеспечивать уникальность идентификаторов в масштабе устройства IPFIX. Процессам сбора **следует** использовать Transport Session и поле Observation Domain ID для идентификации разных потоков от одного экспортёра. **Следует** устанавливать Observation Domain ID = 0, если для всего сообщения IPFIX нет конкретного Observation Domain ID, например, при экспорте Exporting Process Statistics или в случае иерархии коллекторов при экспорте агрегированных Data Record.

3.2. Формат спецификатора поля

Производителям нужна возможность определять фирменные (proprietary) информационные элементы, потому, что они, например, поставляют ещё не стандартизованную продукцию или информационный элемент является коммерчески важным. В этом параграфе определён формат спецификаторов полей (Field Specifier) для зарегистрированных IANA [IANA-IPFIX] и фирменных информационных элементов.

Информационные элементы указываются идентификаторами. Значение бита Enterprise = 0 указывает Information Element из реестра [IANA-IPFIX] и полю Enterprise Number присутствовать **недопустимо**. При Enterprise = 1 соответствующий информационный элемент считается фирменным (enterprise-specific) и поле Enterprise Number **должно** присутствовать. Примеры этого представлены в приложении A.2.2.

Формат Field Specifier представлен на рисунке G.

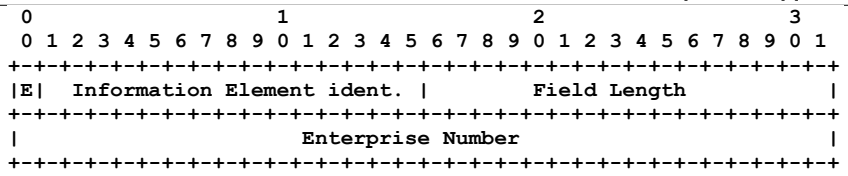


Рисунок G. Формат спецификатора поля.

E

Бит Enterprise является первым в Field Specifier. Если бит сброшен (0) идентификатор Information Element указывает информационный элемент из реестра [IANA-IPFIX] и 4-октетное поле Enterprise Number включать **недопустимо**. Если бит установлен (1) идентификатор указывает фирменный информационный элемент и поле Enterprise Number **должно** присутствовать.

Information Element identifier

Численное значение, представляющее Information Element (см. [IANA-IPFIX]).

Field Length

Размер соответствующего кодированного информационного элемента в октетах (см. [IANA-IPFIX]). Значение Field Length может быть меньше указанного в [IANA-IPFIX], если применяется кодирование с сокращением размера (6.2. Кодирование с сокращённым размером). Значение 65535 зарезервировано для информационных элементов переменного размера (7. Информационные элементы переменного размера).

Enterprise Number

Выделенный IANA номер [IANA-PEN] организации, определившей Information Element в Template Record.

3.3. Формат Set и Set Header

Set - базовый термин для набора записей с похожей структурой. Имеется три типа наборов: Template Set, Options Template Set, Data Set, каждый из которых включает Set Header и хотя бы одну запись. Определения Set Format и Set Header Format приведены в следующих параграфах.

3.3.1. Формат Set

Формат Set показан на рисунке H. Записи могут иметь тип Template Record, Options Template Record, Data Record. **недопустимо** смешивать типы записей в Set.

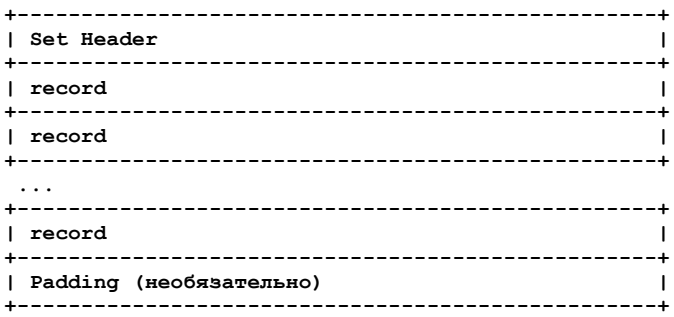


Рисунок H. Формат Set.

Set Header

Заголовок Set в формате, указанном в параграфе 3.3.2. Формат Set Header.

Record

Запись в формате Template Record, Options Template Record или Data Record.

Padding

Exporting Process **может** добавлять октеты заполнения, чтобы последующие наборы Set начинались с принятой границы. Из соображений безопасности октеты заполнения **должны** иметь значение 0. Размер заполнения **должен** быть меньше размера любой допустимой записи в этом Set. Если дополнение сообщений IPFIX желательно в сочетании с очень короткими записями, можно применять информационный элемент paddingOctets для дополнения записей, чтобы их размер возрастал до значения, кратного 4 или 8 октетам. Поскольку Template Set по определению выравниваются на 4 октета, заполнение требуется лишь при задании иного выравнивания (например, 8 октетов).

3.3.2. Формат Set Header

Каждый набор Set включает базовый заголовок, формат которого показан на рисунке I.

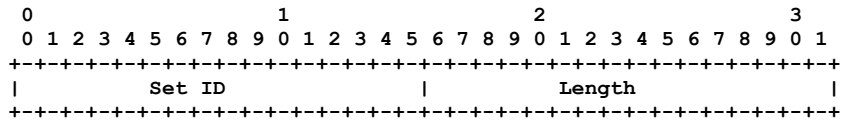


Рисунок I. Формат заголовка Set.

Поля заголовка экспортируются в сетевом порядке байтов.

Set ID

Идентификатор набора. Значение 2 зарезервировано для Template Set, 3 - для Options Template Set, 4 - 255 - резерв на будущее. Значения от 256 служат для Data Set. Значение 0 и 1 не применяются [RFC3954].

Length

Общий размер Set в октетах, включая Set Header, все записи и заполнение (при наличии). Поскольку Set **может** включать несколько записей, значение Length **должно** служить для определения позиции следующего Set.

3.4. Форматы записей

IPFIX определяет 3 формата записей, описанных в последующих параграфах: Template Record, Options Template Record, Data Record.

3.4.1. Формат Template Record

Одним из важных элементов формата записей IPFIX является Template Record. Шаблоны значительно повышают гибкость формата записей, поскольку позволяют процессам сбора обрабатывать сообщения IPFIX без обязанности знать интерпретацию всех Data Record. Template Record содержит комбинацию идентификаторов информационных элементов, выделенных IANA или созданных предприятием (enterprise-specific).

Формат Template Record показан на рисунке J и включает Template Record Header, а также хотя бы 1 спецификатор. Формат спецификатора показан на рисунке G.

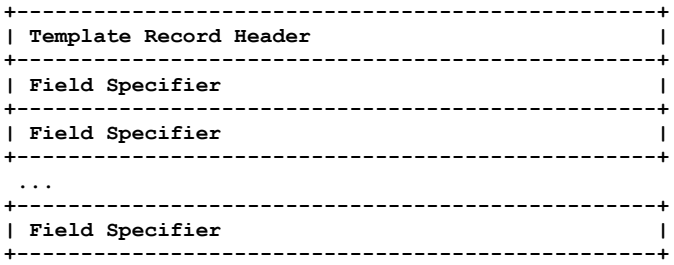


Рисунок J. Формат Template Record.

Формат Template Record Header показан на рисунке K.

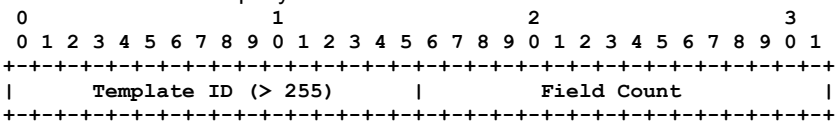


Рисунок K. Формат заголовка Template Record.

Template ID

Каждая запись Template Record задаётся уникальным идентификатором Template ID из диапазона от 256 до 65535. Уникальность идентификаторов требуется в масштабе Transport Session и Observation Domain, где создается Template ID. Поскольку Template ID применяются как Set ID в описываемых ими Set (3.4.3. Формат Data Record), значения 0-255 зарезервированы для особых типов Set (например, сами Template Set), а Template и Options Template (3.4.2. Формат Options Template Record) не могут иметь общие Template ID в рамках Transport Session и Observation Domain. Для порядка выделения значений Template ID ограничения не задаются. Процесс экспорта может выделять Template ID по своему усмотрению, а процессу сбора **недопустимо** предполагать рост Template ID или допущения о содержимом Template на основе лишь Template ID.

Field Count

Число полей в данной записи Template Record.

На рисунке L дан пример Template Set с выделенными IANA и фирменными информационными элементами. Они содержат Set Header, Template Header и несколько Field Specifier. Информационные элементы с id.s 1.2 и 2.1 присутствуют в [IANA-IPFIX] (бит Enterprise = 0) и для них не указывается Enterprise Number.

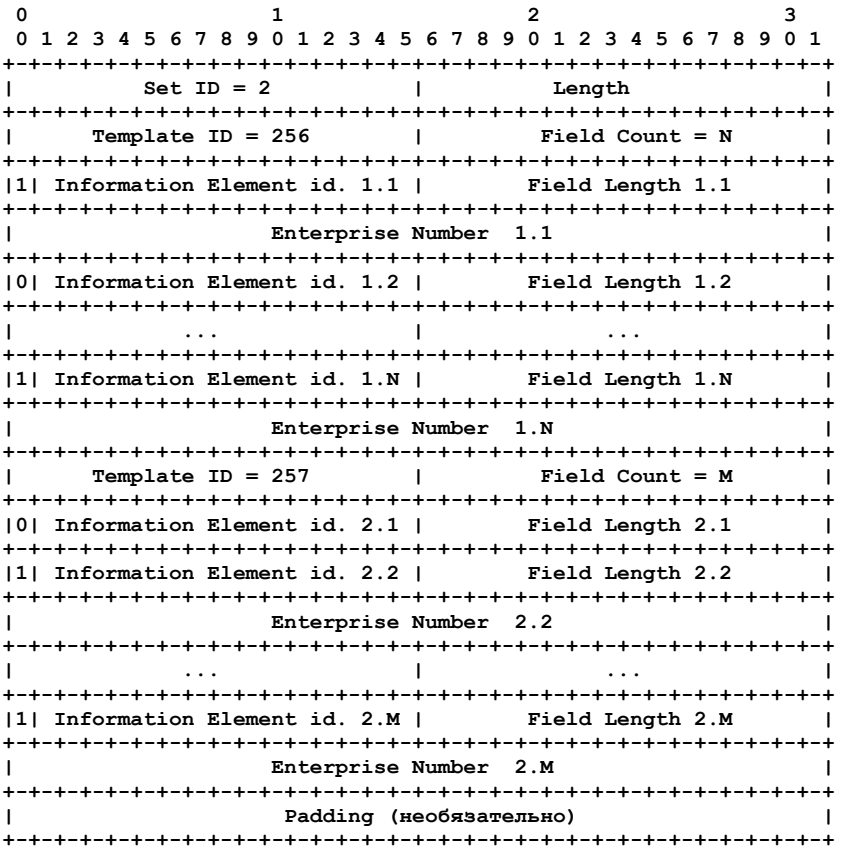


Рисунок L. Пример Template Set.

3.4.2. Формат Options Template Record

Благодаря понятию области действия (scope), Options Template Record даёт экспортёру возможность предоставить сборщику дополнительные сведения, что было бы невозможно при использовании лишь Flow Record.

Описания Options Template для метаданных отчётов о процессах измерения и экспорта IPFIX приведены в разделе 4. Требования к отчётам.

3.4.2.1. Область действия

Область действия (scope), доступная лишь в Options Template Set, даёт контекст сообщённых Information Element в Data Record. Областью действия является один или несколько информационных элементов, заданных в Options Template Record. Процессам сбора **следует** поддерживать в качестве области действия по меньшей мере информационные элементы observationDomainId, exportingProcessId, meteringProcessId, templateId, lineCardId, exporterIPv4Address, exporterIPv6Address и ingressInterface. Протокол IPFIX не препятствует использованию для области действия любых информационных элементов, однако некоторые элементы (например, информационные элементы счётчиков) не имеют смысла в таком качестве.

Заголовок сообщения IPFIX Message Header уже содержит идентификатор домена наблюдения. Отличный от 0 Observation Domain ID можно считать неявным указанием области действия для Data Records в сообщении IPFIX.

В Options Template Record **можно** указать несколько полей Scope, комбинация которых будет задавать область действия. Например, при указании meteringProcessId и templateId областью действия будет данный шаблон для данного процесса измерения. Если разный порядок полей Scope может менять семантику записи, процесс экспорта **должен** сохранять порядок полей Scope. Например, в контексте PSAMP [RFC5476] при определении первым полем Scope функции фильтрации, а вторым - функции выборки порядок полей будет иметь значение. Применение сначала функции отбора, а затем функции фильтра может давать иные Data Record, нежели в случае исходного порядка.

3.4.2.2. Формат Options Template Record

Записи Options Template Record могут содержать произвольные комбинации идентификаторов элементов (выделенных IANA и фирменных). Формат Options Template Record показан на рисунке M и включает Template Record Header, а также 1 или несколько Field Specifier (см. Рисунок G).

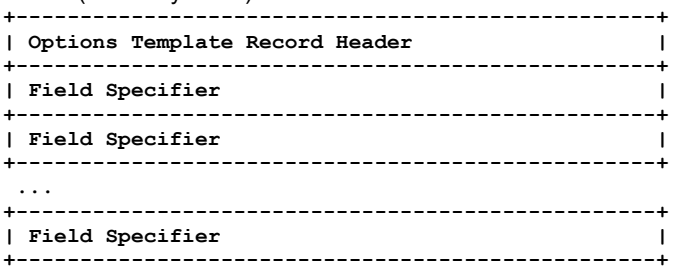


Рисунок M. Формат записи Options Template.

Формат заголовка Options Template Record приведён на рисунке N.

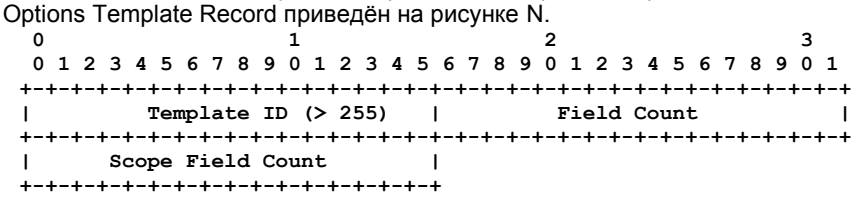


Рисунок N. Формат заголовка Options Template Record.

Template ID

Каждая запись Options Template имеет уникальный идентификатор Template ID из диапазона 256 - 65535. Уникальность идентификаторов требуется в масштабе Transport Session и Observation Domain, где создается Template ID. Поскольку Template ID применяются как Set ID в описываемых ими Set (3.4.3. Формат Data Record), значения 0-255 зарезервированы для особых типов Set (например, сами Template Set), а Template и Options Template (3.4.2. Формат Options Template Record) не могут иметь общие Template ID в рамках Transport Session и Observation Domain. Для порядка выделения значений Template ID ограничения не задаются. Процесс экспорта может выделять Template ID по своему усмотрению, а процессу сбора **недопустимо** предполагать рост Template ID или допущения о содержимом Template на основе лишь Template ID.

Field Count

Число полей в Options Template Record, включая поля Scope.

Scope Field Count

Число полей Scope в данной Options Template Record. Поля Scope являются обычными полями за исключением того, что сборщик интерпретирует их как область действия. Значение счётчика полей Scope N указывает, что первые N Field Specifier в Template Record являются полями Scope. **Недопустимо** задавать Scope Field Count = 0.

Рисунок O показывает пример Options Template Set с информационными элементами IANA и enterprise-specific. Он включает Set Header, Options Template Header и несколько Field Specifier.

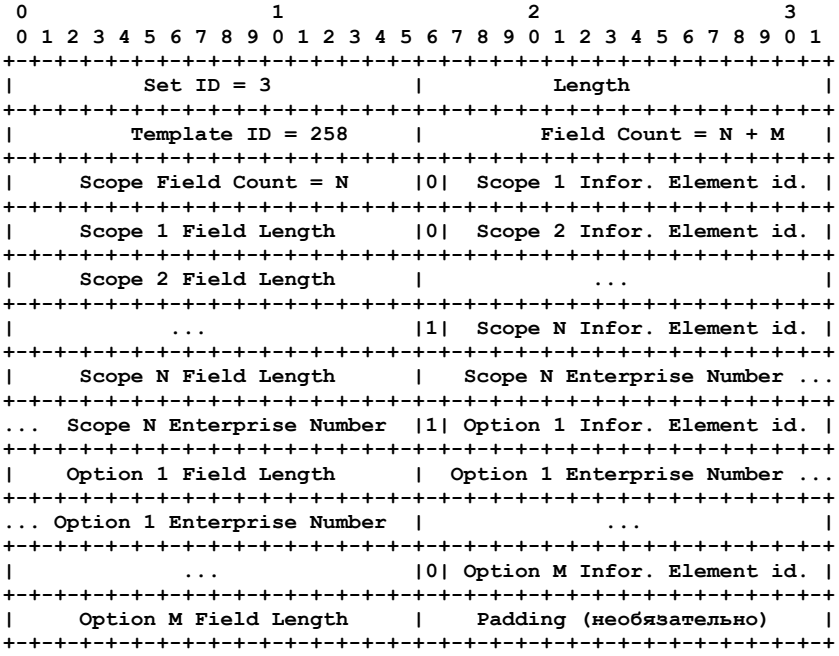


Рисунок O. Пример Options Template Set.

3.4.3. Формат Data Record

Записи Data Record передаются в Data Set. Формат Data Record показан на рисунке P и включает хотя бы одно поле Field Value. Template ID, к которому относятся Field Value, указывается в поле заголовка Set ID, например, Set ID = Template ID.

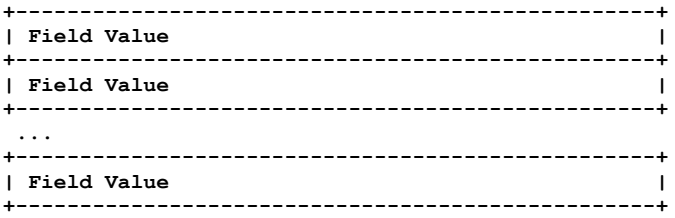


Рисунок P. Формат Data Record.

Отметим, что Field Value не обязаны иметь размер 16 битов и кодируются в соответствии с типом данных, как указано в [RFC7012].

Интерпретация формата Data Record возможна лишь в случае доступности у сборщика записи Template Record, соответствующей Template ID.

На рисунке Q приведён пример Data Set с заголовком Set Header и несколькими полями Field Value.

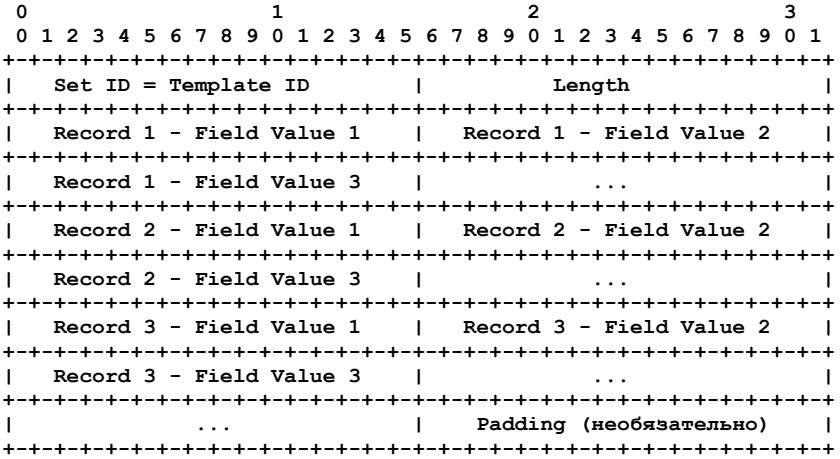


Рисунок Q. Data Set с записями Data Record.

4. Требования к отчётам

Некоторые конкретные Options Template и Options Template Record требуются для предоставления дополнительных сведений о Flow Record и Metering Process.

Могут быть реализованы Options Template и Options Template Record определённые в параграфах, которые налагают некоторые ограничения на реализации процессов измерения и экспорта. В том случае Options Template следует реализовать с соответствии с этими параграфами. В этих конкретных IPFIX Options Template всегда задаётся минимальный набор информационных элементов, но с конкретных шаблонах могут применяться дополнительные информационные элементы.

Процесс сбора **должен** проверять возможные комбинации информационных элементов в Options Template Record для корректной интерпретации следующий Options Template.

4.1. Шаблон опций статистики измерений

Metering Process Statistics Options Template задаёт структуру Data Record для статистических отчётов процесса измерения. В него **следует** включать указанные ниже информационные элементы, как задано в [IANA-IPFIX].

(scope) observationDomainId
Этот информационный элемент **должен** быть определён как поле Scope и **должен** присутствовать, если в содержащем элемент сообщении не задано Observation Domain ID = 0.

(scope) meteringProcessId
При наличии этого элемента он **должен** быть определён как поле Scope.

exportedMessageTotalCount
exportedFlowRecordTotalCount
exportedOctetTotalCount
Процессу экспорта **следует** экспортировать Data Record, указанную в Metering Process Statistics Options Template на регулярной основе или в соответствии с некими правилами экспорта, для чего **следует** обеспечивать возможность настройки.

Отметим, что при доступности нескольких процессов измерения в домене наблюдения экспортёра, информационный элемент meteringProcessId **должен** указываться как дополнительное поле Scope.

4.2. Шаблон опций статистики надёжности процесса измерения

Metering Process Reliability Statistics Options Template задаёт структуру Data Record для отчёта о недостаточной надёжности процесса измерения. В него **следует** включать указанные ниже информационные элементы, как задано в [IANA-IPFIX].

(scope) observationDomainId
Этот информационный элемент **должен** быть определён как поле Scope и **должен** присутствовать, если в содержащем элемент сообщении не задано Observation Domain ID = 0.

(scope) meteringProcessId
При наличии этого элемента он **должен** быть определён как поле Scope.

ignoredPacketTotalCount
ignoredOctetTotalCount
время, когда был проигнорирован первый пакет
Временная метка первого пакета, проигнорированного процессом измерения, заданная любым из элементов observationTimeSeconds, observationTimeMilliseconds, observationTimeMicroseconds, observationTimeNanoseconds.

время, когда был проигнорирован последний пакет
Временная метка последнего пакета, проигнорированного процессом измерения, заданная любым из элементов observationTimeSeconds, observationTimeMilliseconds, observationTimeMicroseconds, observationTimeNanoseconds.

Процессу экспорта **следует** экспортировать Data Record, заданную Metering Process Reliability Statistics Options Template на регулярной основе или в соответствии с некими правилами экспорта, для чего **следует** обеспечивать возможность настройки.

Отметим, что при доступности нескольких процессов измерения в домене наблюдения экспортёра, информационный элемент meteringProcessId **должен** указываться как дополнительное поле Scope.

Поскольку Metering Process Reliability Statistics Options Template содержит два информационных элемента с идентичными временными метками, а порядок информационных элементов в Template Record не гарантируется, процесс сбора интерпретирует интервал игнорирования пакетов как диапазон между двумя значениями. Учёт перехода через максимум рассмотрен в параграфе 5.2. Поддержка перехода временных меток через максимум.

4.3. Шаблон опций статистики надёжности процесса экспорта

Exporting Process Reliability Statistics Options Template задаёт структуру Data Record для отчёта о недостаточной надёжности процесса экспорта. В него **следует** включать указанные ниже информационные элементы, как задано в [IANA-IPFIX].

(scope) Exporting Process Identifier
Идентификатор процесса экспорта, для которого сообщается о надёжности. Поле может содержать любой из информационных элементов exporterIPv4Address, exporterIPv6Address, exportingProcessId, который **должен** быть указан как поле Scope.

notSentFlowTotalCount
notSentPacketTotalCount
notSentOctetTotalCount
время, когда был отброшен первый поток
Временная метка первой Flow Record, отброшенной процессом экспорта, заданная любым из элементов observationTimeSeconds, observationTimeMilliseconds, observationTimeMicroseconds, observationTimeNanoseconds.

время, когда был отброшен последний поток
Временная метка последней Flow Record, отброшенной процессом экспорта, заданная любым из элементов observationTimeSeconds, observationTimeMilliseconds, observationTimeMicroseconds, observationTimeNanoseconds.

Процессу экспорта **следует** экспортировать Data Record, заданную Exporting Process Reliability Statistics Options Template на регулярной основе или в соответствии с некими правилами экспорта, для чего **следует** обеспечивать возможность настройки.

Поскольку Exporting Process Reliability Statistics Options Template содержит два информационных элемента с идентичными временными метками, а порядок информационных элементов в Template Record не гарантируется, процесс сбора интерпретирует интервал игнорирования пакетов как диапазон между двумя значениями. Учёт перехода через максимум рассмотрен в параграфе 5.2. Поддержка перехода временных меток через максимум.

4.4. Шаблон опций ключей потока

Flow Keys Options Template задаёт структуру Data Record для Flow Key в отчётах о потоках. Flow Keys Data Record расширяет конкретную Template Record, указанную templateId. Запись Template расширяется путём указания информационных элементов, содержащихся в соответствующих Data Record, описывающих свойства потока, которые служат ключами (Flow Key) для потока в отчёте.

В Flow Keys Options Template **следует** включать указанные ниже информационные элементы, как задано в [IANA-IPFIX]

(scope) templateId
Этот информационный элемент **должен** быть задан как поле Scope.
flowKeyIndicator

5. Вопросы синхронизации

5.1. Export Time и Flow Record Time в сообщении IPFIX

Поле Export Time Message Header в сообщении IPFIX указывает время, когда IPFIX Message Header покидает Exporter, с использованием того же формата кодирования, как в абстрактном типе данных dateTimeSeconds [RFC7012], т. е. в секундах с начала эпохи UNIX (1 января 1970 г., 00:00 UTC) в форме 32-битового целого числа без знака.

Некоторые информационные элементы, связанные со временем, могут указываться смещением от Export Time. Например, Data Record, требующие микросекундной точности, могут экспортировать время начала и завершения потока в информационных элементах flowStartMicroseconds и flowEndMicroseconds, которые указывают время эпохи NTP (1 января 1900 г., 00:00 UTC) в 64-битовом поле. Другим решением является экспорт flowStartDeltaMicroseconds и flowEndDeltaMicroseconds в Data Record с указанием времени начала и завершения отрицательным смещением от Export Time в форме 32-битового целого числа без знака. Это снижает требования к пропускной способности для экспорта за счёт уменьшения временных меток на 4 байта, но повышает нагрузку на экспортёра, поскольку Exporting Process требуется рассчитывать flowStartDeltaMicroseconds и flowEndDeltaMicroseconds для каждой Data Record в сообщении IPFIX.

Нужно отметить, что временные метки на основе Export Time вносят некоторые ограничения для Data Record в сообщении IPFIX. В примере с информационными элементами flowStartDeltaMicroseconds и flowEndDeltaMicroseconds запись Data Record может включать лишь метки из интервала 71 минуту до Export Time, поскольку иначе размер метки начала выйдет за пределы 32-битового значения смещения.

5.2. Поддержка перехода временных меток через максимум

Абстрактный тип данных dateTimeSeconds [RFC7012] и поле Export Time Message Header (5.1. Export Time и Flow Record Time в сообщении IPFIX) кодируются 32-битовыми целыми числами без знака в секундах с начала эпохи UNIX (1 января 1970 г., 00:00 UTC), как задано в [POSIX.1]. Эти поля достигнут максимума 7 февраля 2106 в 06:28:16 UTC.

Для поддержки использования протокола IPFIX после этой даты процессу экспорта **следует** экспортировать значения dateTimeSeconds и поле Export Time Message Header как число секунд с начала эпохи UNIX (1 января 1970 г., 00:00 UTC) по модулю 232. Процессам сбора **следует** использовать текущую дату или иной контекст для подобающей интерпретации значений dateTimeSeconds и поля Export Time Message Header.

Аналогичные соображения применимы и к основанным на эпохе NTP абстрактным типам dateTimeMicroseconds и dateTimeNanoseconds [RFC7012]. Процессу экспорта Exporting Process **следует** экспортировать значения dateTimeMicroseconds и dateTimeNanoseconds, как будто эра NTP [RFC5905] задана неявно, а процессам сбора **следует** использовать текущую дату или иной контекст для определения эры NTP, чтобы должным образом интерпретировать значения dateTimeMicroseconds и dateTimeNanoseconds в полученных Data Record.

Абстрактный тип dateTimeMilliseconds будет достигать максимума приблизительно через 500 миллиардов лет и спецификация поведения этого типа в более позднее время оставлена для будущей версии этой спецификации.

На долгосрочное хранение файлов [RFC5655] для архивирования влияет переход временных меток через максимум. Поскольку применение текущей даты для интерпретации сохранённых в файлах значений может через несколько десятков лет привести к ошибкам, **рекомендуется** сохранять такие файлы с данными контекста, обеспечивающими корректную интерпретацию временных меток.

6. Связь с информационной моделью

Как и в IPFIX Message Header и Set Header, значения информационных элементов [RFC7012], кроме типов string и octetArray, кодируются в каноническом формате с сетевым порядком байтов (big-endian).

6.1. Кодирование типов данных IPFIX

В последующих параграфах определено кодирование типов данных из [RFC7012].

6.1.1. Интегральные типы

Интегральные типы unsigned8, unsigned16, unsigned32, unsigned64, signed8, signed16, signed32, signed64 **должны** кодироваться с использованием канонического формата с каноническим порядком байтов. Интегральные типы со знаком представляются как дополнение до 2.

6.1.2. Адресные типы

Адресные типы macAddress, ipv4Address, ipv6Address **должны** кодироваться так же, как интегральные с размером 6, 4 и 16 октетов, соответственно, и сетевым порядком байтов.

6.1.3. float32

Данные типа float32 **должны** кодироваться как числа с плавающей точкой IEEE binary32 в соответствии с [IEEE.754.2008] с сетевым порядком байтов, как указано в параграфе 3.6 [RFC1014]. Отметим, что на машинах с прямым порядком байтов (little-endian) требуется соответствующая перестановка байтов перед экспортом. Метод перестановки может зависеть от платформы.

6.1.4. float64

Данные типа float64 **должны** кодироваться как числа с плавающей точкой IEEE binary64 в соответствии с [IEEE.754.2008] с сетевым порядком байтов, как указано в параграфе 3.6 [RFC1014]. Отметим, что на машинах с прямым порядком байтов (little-endian) требуется соответствующая перестановка байтов перед экспортом. Метод перестановки может зависеть от платформы.

6.1.5. boolean

Логический тип данных (boolean) задан в соответствии с TruthValue в [RFC2579] и кодируется 1-октетным целым числом в соответствии с параграфом 6.1.1. Значение 1 указывает истину (true), 2 - ложь (false). Все прочие значения остаются неопределёнными.

6.1.6. string u octetArray

Тип данных string представляет строки конечной длины из разрешённых символов Unicode. Тип string **должен** кодироваться в формате UTF-8 [RFC3629]. Строки передаются как массив (возможно пустой) октетов с использованием информационных элементов переменного размера. Процессу экспорта IPFIX **недопустимо** передавать сообщения IPFIX, содержащие некорректные строки UTF-8 для информационных элементов типа string, процессам сбора **следует** обнаруживать и игнрировать такие значения. Дополнительная информация представлена в [UTF8-EXPLOIT].

Для типа octetArray не задано правил кодирования, он представляет необработанный (raw) массив (возможно пустой) октетов, интерпретация которых задаётся определением информационного элемента.

6.1.7. dateTimeSeconds

Тип dateTimeSeconds выражается 32-битовым целым числом без знака с сетевым порядком байтов, указывает число секунд с начала эпохи UNIX (1 января 1970 г., 00:00 UTC), как указано в [POSIX.1]. dateTimeSeconds кодируется аналогично полю IPFIX Message Header Export Time и может представлять даты с 1 января 1970 г. до 7 февраля 2106 без перехода через максимум (см. параграф 5.2. Поддержка перехода временных меток через максимум).

6.1.8. dateTimeMilliseconds

Тип dateTimeMilliseconds выражается 64-битовым целым числом без знака с сетевым порядком байтов, указывает число секунд с начала эпохи UNIX (1 января 1970 г., 00:00 UTC), как указано в [POSIX.1] и может представлять даты с 1 января 1970 г. приблизительно на 500 миллиардов лет без перехода через максимум.

6.1.9. dateTimeMicroseconds

Тип dateTimeMicroseconds выражается 64-битовым целым числом без знака, кодируемым в формате NTP Timestamp, как указано в разделе 6 [RFC5905]. Это поле состоит из двух 32-битовых целых чисел без знака - Seconds (секунды) и Fraction (доли секунды). Поле Seconds указывает число секунд с начала эпохи NTP (1 января 1900 г., 00:00 UTC). Поле Fraction указывает доли секуды в единицах $1/(2^{32})$ секунды (приблизительно 233 пикосекунды). Значение может представлять даты от 1 января 1900 г. до 8 февраля 2036 г. в текущей эре NTP (см. параграф 5.2. Поддержка перехода временных меток через максимум).

Отметим, что типы dateTimeMicroseconds и dateTimeNanoseconds используют идентичное кодирование. Тип dateTimeMicroseconds предназначен лишь для представления временных меток с микросекундным разрешением, поэтому младшим 11 битам (2^{11} x 233 пксек = 0,477 мксек) поля Fraction **следует** иметь значение 0 и они **должны** игнорироваться во всех информационных элементах этого типа.

6.1.10. dateTimeNanoseconds

Тип dateTimeNanoseconds выражается 64-битовым полем, кодируемым в соответствии с форматом NTP Timestamp, как указано в разделе 6 [RFC5905]. Это поле состоит из двух 32-битовых целых чисел без знака - Seconds (секунды) и Fraction (доли секунды). Поле Seconds указывает число секунд с начала эпохи NTP (1 января 1900 г., 00:00 UTC). Поле Fraction указывает доли секуды в единицах $1/(2^{32})$ секунды (приблизительно 233 пикосекунды). Значение может представлять даты от 1 января 1900 г. до 8 февраля 2036 г. в текущей эре NTP (см. параграф 5.2. Поддержка перехода временных меток через максимум).

Отметим, что типы dateTimeMicroseconds и dateTimeNanoseconds. Для интерпретации поля Fraction в типе dateTimeNanoseconds ограничений не задано.

6.2. Кодирование с сокращённым размером

Информационные элементы, кодируемые как числа со знаком, без знака или с плавающей точкой, **можно** представить меньшим числом октетов, нежели предполагает определение типа в модели информации, исходя из допущения, что для любого типа, который нужно передавать экспортёру, достаточно меньшего размера. Это сокращает расход пропускной способности сети между экспортером и коллектором. Отметим, что определения информационных элементов [IANA-IPFIX] всегда задают максимальный размер.

Например, информационная модель задаёт octetDeltaCount как тип unsigned64, для которого нужно 64 бита, однако экспортёр, который никогда не передаёт значения больше 4294967295, может выбрать представление unsigned32.

Такое поведение экспортёр указывает заданием в Template размера, который меньше, чем заданный для типа информационного элемента. В приведённом примере экспортёр будет указывать в шаблоне размер 4 вместо 8.

Кодирование с сокращённым размером **можно** применять для типов unsigned64, signed64, unsigned32, signed32, unsigned16, signed16, при этом **должно** сохраняться наличие или отсутствие знака. Сокращать размер можно на любое число октетов по сравнению с исходным, пока результат устраивает, т. е. отбрасываются лишь нули в старших битах. Например, unsigned64 можно сократить до 7, 6, 5, 4, 3, 2 или 1 октета.

Кодирование с сокращённым размером **можно** применять для преобразования float64 в float32. Тип float32 не только имеет меньший размер за счёт сокращения размера мантиссы, но и точность его ниже. После преобразования float64 размер сокращается до 4 октетов.

Сокращение размера **недопустимо** для других типов из [RFC7012], которые предполагают фиксированный размер, поскольку эти типы имеют внутреннюю структуру (как ipv4Address или dateTimeMicroseconds) или ограниченные диапазоны, которые не подходят для сокращенного размера (например, dateTimeMilliseconds).

Информационные элементы типов octetArray и string можно экспортировать с любым размером, учитывая ограничения на размеры соответствующего информационного элемента в его описании.

7. Информационные элементы переменного размера

Механизм IPFIX Template оптимизирован для информационных элементов фиксированного размера [RFC7012]. Для элементов с переменным размером **должен** использоваться описанный здесь механизм передачи размера для выделенных IANA и фирменных информационных элементов.

В Template Set для Information Element Field Length указывается зарезервированное значение 65535, указывающее процессу сборки, что размер информационного элемента будет указан в самом элементе.

В большинстве случаев размер информационного элемента не превышает 255 октетов и для этогоприменяется показанный ниже механизм с указанием размера перед информационным элементом (Рисунок R).

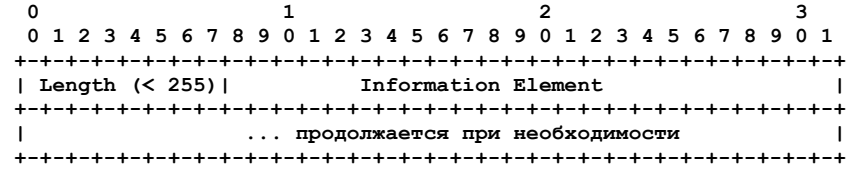


Рисунок R. Информационный элемент (IE) переменного размера до 255 октетов.

Размер можно указать также 3 октетами перед информационным элементом, что позволяет указывать размеры больше 255 октетов. В этом случае первый октет поля Length **должен** иметь значение 255, а фактический размер в октетах указывается вторым и третьим октетом, как показано на рисунке S.

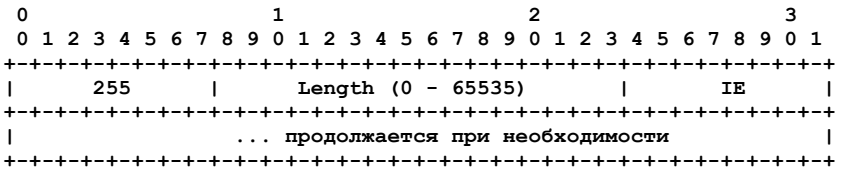


Рисунок S. Информационный элемент (IE) переменного размера до 65535 октетов.

Октеты размера (1 или 3 октета в начале) **недопустимо** учитывать в поле размера информационного элемента.

8. Управление шаблонами

В этом разделе описано управление шаблонами Template и Options Template в процессах экспорта и сбора. Цель управления шаблонами состоит в обеспечении, насколько это возможно, наличия у процессов экспорта и сбора согласованного представления о шаблонах Template и Options Template, применяемых для кодирования идекодирования записей в процессах экспорта и сбора. Достижение этой цели сложняется двумя факторами - 1) необходимость поддержки неоднократного использования Template ID в транспортной сессии и 2) необходимость поддержки передачи шаблонов без гарантии доставки при использовании UDP для доставки сообщений IPFIX.

Заданные здесь механизмы управления шаблонами применимы для экспорта сообщений IPFIX в SCTP, TCP или UDP. Дополнительное рассмотрение для протоколов SCTP и UDP приведено в параграфах 8.3 и 8.4, соответственно.

Процесс экспорта назначает и поддерживает значения Template ID на уровне транспортной сессии и домена наблюдений. Созданной заново Template Record процесс экспорта назначает свободное значение Template ID. Процесс сбора **должен** сохранять всю принятую информацию о Template Record на протяжении каждой транспортной сессии, пока она не будет использована снова или отозвана, как описано в парагафе 8.1, или завершится срок действия при использовании UDP, как указано в параграфе 8.4, чтобы иметь возможность интерпретировать соответствующие записи Data Record.

Процессу сбора **недопустимо** предполагать, что Template ID от данного процесса экспорта указывают на те же шаблоны, которые были в предыдущей транспортной сессии с тем же процессом экспорта. Процессу сбора **недопустимо** использовать шаблоны из одной транспортной сессии для декодирования Data Set в другой сессии.

Если конкретный информационный элемент требуется шаблоном, но отсутствует в наблюдаемых пакетах, процесс экспорта **может** экспортировать записи Flow Record без этого информационного элемента в Data Record, описанной в новом шаблоне.

Если информационный элемент требуется шаблоном более одного раза, разным экземплярам этого элемента **следует** размещаться в логическом порядке их обработки в процессе измерения. Например, если выбранные пакеты проходят через 2 хэш-функции и эти хэш-значения передаются в одном шаблоне, первому хэш-значению следует относиться к первой в процессе измерения хэш-функции. При экспорте двух IP-адресов отправителя пакета IPv4-in-IPv4 в первый элемент sourceIPv4Address следует включать адрес IPv4 из внешнего заголовка, во второй - из внутреннего. Процесс сборки **должен** корректно обрабатывать шаблоны с несколькими идентичными информационными элементами.

Процессу экспорта **следует** передавать Template Set и Options Template Set до любого Data Sets, использующего этот (Options) Template ID, чтобы обеспечить сборщику наличие Template Record до получения первой Data Record. Записи Data Record, соответствующие Template Record **могут** помещаться в те же и/или последующие сообщения IPFIX. Однако процессу сбора **недопустимо** предполагать, что Data Set и сответствующий Template Set (Options Template Set) экспортируется в том же сообщении IPFIX.

Хотя Collecting Process обычно получает записи Template Record от процесса экспорта до получения Data Record, это бывает не всегда, например, в результате нарушения порядка или при перезапуске Collecting Process по протоколу UDP. В таких случаях Collecting Process **может** буферизовать записи Data Record, для которых у него нет шаблонов, в ожидании описывающих их записей Template Record. Однако следует отметить, что при наличии отзыва или переопределения шаблона (8.1. Отзыв и переопределение шаблонов) это может приводить к некорректной интерпретации записей Data Record.

Разные домены наблюдение в одной транспортной сессии **могут** использовать одно значение Template ID для разных шаблонов и процесс сборки **должен** корректно обрабатывать такие случаи.

Шаблоны Options Template и Template, относящиеся к взаимозависимым записям (например, с общими свойствами, как описано в [RFC5473]), **следует** передавать в одном сообщении IPFIX.

8.1. Отзыв и переопределение шаблонов

Шаблоны, которые Exporting Process больше не будет применять, **можно** отозвать с помощью Template Withdrawal. После получения Template Withdrawal процесс сбора **должен** прекратить использование шаблона при интерпретации последующих Data Set. Отметим, что этот механизм не применим при использовании для сообщений IPFIX транспорта UDP (см. 8.4. Дополнительные вопросы управления шаблонами для UDP).

Template Withdrawal включает Template Record для отзываемого Template ID с Field Count = 0 (Рисунок T).

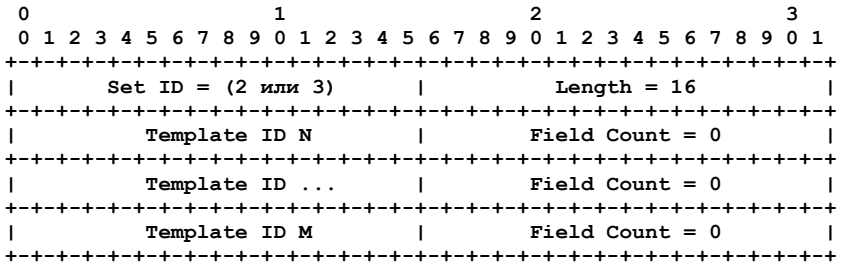


Рисунок T. Формат отзыва шаблона.

Поле Set ID **должно** иметь значение 2 для Template Set Withdrawal или 3 для Options Template Set Withdrawal. **Можно** отозвать несколько Template ID в одном Template Withdrawal, в этом случае **можно** применять заполнение.

Template Withdrawal **можно** чередовать с Template Set, Options Template Set и Data Set в сообщении IPFIX. В этом случае Template и Template Withdrawal нужно интерпретировать в порядке их размещения в сообщении IPFIX. Процессу экспорта **не следует** передавать Template Withdrawal, пока не пройдёт время, достаточное для получения и обработки Data Record, описываемых отзываемым шаблоном (см. 8.2. Действия по управлению последовательными шаблонами).

Завершение транспортной сессии неявно отзывает все использованные в этой сессии шаблоны и нужно передать шаблоны снова в последующих сессиях между Exporting Process и Collecting Process. Это применимо только для SCTP и TCP, случай для протокола UDP описан в параграфах 8.4. Дополнительные вопросы управления шаблонами для UDP и 10.3.4. Организация и разрыв сессий.

Для данного домена наблюдения **можно** отозвать все шаблоны с помощью All Templates Withdrawal, как показано на рисунке U, а Options Template **можно** отозвать с помощью All Options Templates Withdrawal (Рисунок V).

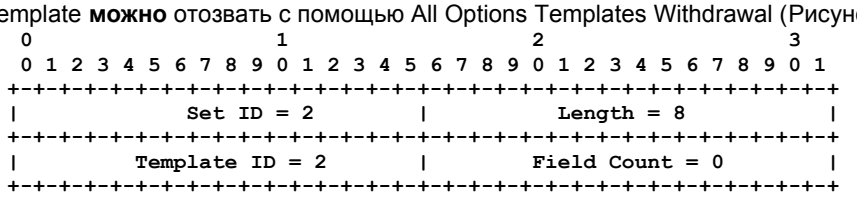


Рисунок U. Формат All Templates Withdrawal Set.

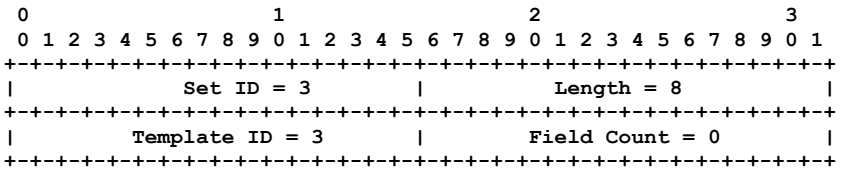


Рисунок V. Формат Options All Templates Withdrawal Set.

Template ID **можно** применять снова для новых шаблонов, передавая Template Record или Options Template Record для данного Template ID после отзыва шаблона.

Если Collecting Process получает Template Withdrawal для Template или Options Template, которые на были сохранены, это указывает на ошибку или некорректную реализацию Exporting Process. Получение и обработка Data Record остаются возможными, но процесс сбора **должен** игнорировать Template Withdrawal, а ошибку **следует** записать.

Если Collecting Process получает новую запись Template Record или Options Template для уже выделенного Template ID и этот Template или Options Template идентичен уже полученному, **следует** записать в журнал повторную передачу - это не ошибка и не влияет на интерпретацию записей Data Record.

Если Collecting Process получает новую запись Template или Options Template для уже выделенного Template ID и Template или Options Template отличается от уже имеющегося, это указывает на ошибку или некорректную реализацию Exporting Process. Продолжение приёма и однозначной интерпретации Data Record для этого Template ID больше не возможны и процессу сбора следует записать ошибку. Дальнейшие действия Collecting выходят за рамки допумента.

8.2. Действия по управлению последовательными шаблонами

Поскольку нет гарантий упорядоченной доставки сообщений IPFIX через потоки SCTP или по протоколу UDP, процесс экспорта **должен** упорядочивать все действия по управлению шаблонами (т. е. обработку Template для новых шаблонов и Template Withdrawal для отзыва) с использованием поля Export Time в заголовке сообщения IPFIX.

Процессу экспорта **недопустимо** экспортировать Data Set, описанный новым шаблоном, в сообщении IPFIX с Export Time до Export Time в заголовке сообщения IPFIX с этим шаблоном. Если новый шаблон или описываемый им Data Set присутствует в том же сообщении IPFIX, Template Set с шаблоном **должен** размещаться в сообщении до Data Set.

Процессу экспорта **недопустимо** экспортировать Data Set, описываемые отозванным шаблоном в сообщениях IPFIX с Export Time после Export Time сообщения IPFIX с Template Withdrawal для этого шаблона.

Иными словами, шаблон описывает записи Data Record, содержащиеся в сообщениях IPFIX, когда Export Time в этих сообщениях находится в интервале между конкретным временем начала и завершения (включительно). Время старта - это Export Time в сообщении IPFIX Message с Template Record, а время завершения может быть одним из двух - если шаблон отозван во время сессии, - это Export Time сообщения IPFIX с Template Withdrawal для шаблона, в иных случаях - это завершение транспортной сессии.

Даже при упорядоченной передаче сообщения IPFIX с действиями по управлению шаблонами могут приходить в процесс сбора с нарушением порядка, например, при передаче по UDP или в разных потоках SCTP. С учётом этого Template Withdrawal и последующее повтороное использование Template ID могут значительно усложнить проблему определения срока действия шаблона в Collecting Process. Процесс сбора **может** реализовать буфер и использовать Export Time для устранения неоднозначности порядка действий по управлению шаблонами. Прireализации такого буфера его **следует** делать настраиваемым для обеспечения задержки порядка максимальной задержки из-за переупорядочения, возникающей в процессе сбора. Отметим для этого случая, что время (часы) Collecting Process не имеет значения, поскольку сравниваются лишь значения Export Time в сообщениях.

8.3. Дополнительные вопросы управления шаблонами для SCTP

Этот параграф применим только к SCTP и при возникновении противоречий с разделом 8 или параграфом 8.1, сведения, приведённые здесь, имеют более высокий приоритет.

Template Set и Options Template Set **можно** передавать в любом потоке SCTP. Data Set, переданные в данном потоке SCTP Stream **могут** быть представлены записями Template Record экспортированными в любом потоке SCTP.

Template Set и Options Template Set **должны** передаваться надёжно с использованием упорядоченной доставки SCTP.

Template Withdrawal **можно** передавать в любом потоке SCTP и они **должны** передаваться надёжно с использованием упорядоченной доставки SCTP. Template ID **можно** применять снова путём передачи Template Withdrawal и/или новой Template Record в потоке SCTP, отличном от того, где был передан исходный шаблон.

Дополнительные вопросы управления шаблонами рассмотрены в [RFC6526], где задано расширения для явных шаблонов каналов с потоками SCTP. В обмен на более строгие правила назначения Template Record потокам SCTP это расширение позволяет быстро и надёжно применять повторные Template ID и определять потерю Data Record на уровне шаблона.

8.4. Дополнительные вопросы управления шаблонами для UDP

Этот параграф применим только к UDP и при возникновении противоречий с разделом 8 или параграфом 8.1, сведения, приведённые здесь, имеют более высокий приоритет.

Поскольку UDP не имеет метода надёжной передачи шаблонов, процессы экспорта, применяющие транспорт UDP, **должны** повторять передачу каждого активного шаблона с регулярными интервалами. Интервал повтора **должен** быть настраиваемым, например, через параметры templateRefreshTimeout и optionsTemplateRefreshTimeout, как указано в [RFC6728]. Принятые по умолчанию значения этих параметров зависят от реализации и развёртывания.

Перед экспортом любой записи Data Record, описываемой данной Template Record или Options Template Record, особенно в случае повтрного использования Template ID, как описано в параграфе 8.1, процессу экспорта **следует** передать несколько копий Template Record в отдельных сообщениях IPFIX, чтобы помочь в их гарантированном получении процессом сбора.

Для снижения расхода ресурсов на шаблоны, которые Exporting Process уже не использует, процесс сбора **может** задать срок действия каждого шаблона, полученного в транспортной сессии. Если Exporting Process не обновит шаблон в течение этого срока, процесс сбора может отбросить шаблон. Срок действия шаблона в Collecting Process **можно** задать параметром конфигурации или вывести из интервала периодического повтора передачи шаблонов от процесса экспорта. Во втором случае срок действия по умолчанию **следует** делать не меньше 3-кратного интервала повтора.

Процессу экспорта **недопустимо** передавать Template Withdrawal (параграф 8.1) по UDP, а процесс сбора **должен** игнорировать такой отзыв, полученный по UDP. Template ID могут повторно использоваться процессом экспорта путём экспорта нового шаблона с Template ID не раньше завершение 3-кратного интервала повтора передачи, иначе повторное использование Template ID может вести к некорректной интерпретации Data Record.

При получении процессом сбора новой Template Record или Options Template Record по UDP для уже выделенного Template ID с Template или Options Template идентичным уже полученному, **не следует** записывать повтор в системный журнал, поскольку это обычная операция обновления шаблона по протоколу UDP.

При получении процессом сбора новой Template Record или Options Template Record по UDP для уже выделенного Template ID с Template или Options Template, отличающимся от уже полученного, процесс сбора **должен** заменить

Template или Options Template для этого Template ID полученным вновь шаблоном. Это нормальная операция повторного использования Template ID по протоколу UDP.

Поскольку Template ID в каждый момент уникальны для сессии UDP и домена наблюдения, процессу сбора **следует** поддерживать для всех текущих Template Record и Options Template Record набор сведений <устройство IPFIX, порт отправителя UDP у экспортера, IP-адрес сборщика, порт получателя UDP у сборщика, Observation Domain ID, Template ID, определение шаблона, время последнего получения>.

9. Коллектор

Здесь описана обработка протокола IPFIX в процессе сбора, общая для всех транспортных протоколов. Дополнительное рассмотрение для SCTP и UDP приведено в параграфах 9.2 и 9.3, соответственно. Управление шаблонами в процессе сбора рассмотрено в разделе 8.

Collecting Process **должен** слушать запросы от Exporting Process на организацию соединений (ассоциаций) для запуска новой транспортной сессии.

Процесс сбора **должен** отмечать идентификатор любого непонятного ему информационного элемента и **может** отбрасывать такие элементы из полученных Data Record.

Процесс сбора **должен** воспринимать заполнение в Data Record и Template Record. Размер заполнения определяет разность Set Length и Set Header (4 октета для Set ID и Set Length) по модулю минимального размера записи, выеденному из Template Record.

Протокол IPFIX имеет поле Sequence Number в заголовке экспорта, которое увеличивается на число IPFIX Data Record в сообщении IPFIX. Сборщик может обнаруживать потерю, нарушение порядка и дубликаты сообщений IPFIX по этим номерам. Сборщику **следует** поддерживать механизм записи для фактов нарушения порядка сообщений IPFIX. Такое нарушение может возникать при нехватке у экспортёра ресурсов, когда тот не может передать сообщение в момент его создания, перегрузке в сети между экспортером и сборщиком, нехватке ресурсов у сборщика, когда тот не может обработать сообщения IPFIX в момент прибытия, нарушении порядка приема, дублировании пакетов или атаке с внедрением ложных сообщений.

9.1. Обработка процессом сборки сообщений IPFIX с ошибками формата

Если Collecting Process получает сообщение IPFIX с ошибкой формата, он **должен** отбросить сообщение и **следует** записать ошибку. Неверно сформированными считаются сообщения IPFIX, которые невозможно интерпретировать из-за бессмысленных значений размера (например, информационный элемент переменного размера превышает размер Set, размер Set больше сообщения IPFIX или сообщение IPFIX короче IPFIX Message Header) или резервного значения Version (это может указывать применение будущей версии IPFIX, но на практике чаще всего связано с передачей процессу сбора данных, не относящихся к IPFIX). Заполнение в Set не нарушает формат сообщения IPFIX.

Поскольку наиболее вероятной причиной ошибки в формате сообщений IPFIX является плохая реализация процесса экспорта или передача не связанных с IPFIX данных в IPFIX Collecting Process, для решения проблемы вероятно потребуется участие человека. Collecting Process **может** попытаться исправить ситуцию своими силами, включая:

- разрыв соединения TCP или SCTP;
- использование окна получателя для снижения нагрузки на сеть от плохо работающего процесса экспорта;
- буферизацию и сохранение некорректных сообщений IPFIX для последующей диагностики;
- попытки ресинхронизировать поток, например, как описано в параграфе 10.3 [RFC5655].

Ресинхронизацию следует применять лишь в случае, когда у процесса сбора есть основания считать ошибку временной. Процессу сбора **следует** прекращать обработку сообщений IPFIX от явно неисправных процессов экспорта (например, передавших подряд несколько сообщений IPFIX с ошибками формата).

9.2. Процесс сбора по протоколу SCTP

Exporting Process может запрашивать и поддерживать несколько потоков в ассоциации SCTP, а процесс сбора **должен** поддерживать создание множества потоков SCTP.

9.3. Процесс сбора по протоколу UDP

Транспортная сессия для сообщений IPFIX по протоколу UDP определяется с точки зрения процесса экспорта и примерно соответствует времени, в течение которого данный Exporting Process передаёт сообщения IPFIX через UDP данному процессу сбора. Поскольку это сложно обнаружить в процессе сбора, тот **может** отбросить всё состояние транспортной сессии при отсутствии сообщений IPFIX от данного Exporting Process в данной транспортной сессии в течение настраиваемого интервала бездействия.

Процессу сбора **следует** воспринимать Data Record без связанной Template Record (или других определений, таких как общие свойства), требуемой для декодирования Data Record. Если записи Template Record или иные определения не были получены к моменту приёма Data Record, процесс сбора **может** на короткое время сохранить Data Record и декодировать их по получении Template Record или иных определений, сравнивая Export Time в сообщениях IPFIX, содержащих Template Record, с временем в сообщениях с Data Record, как указано в параграфе 8.2. Отметим, что этот механизм может вести к некорректной интерпретации записей в случае повторного использования Template ID или иных идентификаторов с ограниченным сроком действия.

10. Транспортный протокол

Спецификация протокола IPFIX разработана с учётом независимости от транспортного протокола. Отметим, что экспортёр может передавать данные процессам сбора, использующим независимые транспортные протоколы.

16-битовое поле Length в IPFIX Message Header ограничивает размер сообщений IPFIX до 65535 октетов, включая заголовок. Процесс сбора **должен** поддерживать обработку сообщений IPFIX размером до 65535 октетов.

Хотя процессы экспорта и сбора могут поддерживать несколько транспортных протоколов, транспортные сессии привязаны к одному протоколу. Процессу экспорта или сбора **недопустимо** переносить состояние транспортной сессии между сессиями, использующими разный транспорт для одной пары Exporting Process и Collecting Process. Иными словами, Exporting Process (или Collecting Process) с поддержкой нескольких транспортных протоколов концептуально является несколькими процессами экспорта (сбора), по одному на протокол.

10.1. Соответствие транспорту и применение транспорта

Для соответствия спецификации **должна** быть реализована поддержка SCTP [RFC4960] с расширением Partially Reliable SCTP (PR-SCTP), заданным в [RFC3758]. **Можно** также реализовать поддержку UDP [UDP] и/или TCP [TCP].

SCTP следует применять в системах, где экспортёры и коллекторы взаимодействуют по каналам, устойчивым к перегрузке. Протокол SCTP может обеспечивать требуемый уровень надёжности при использовании PR-SCTP.

TCP можно применять в системах, где экспортёры и коллекторы взаимодействуют по каналам, устойчивым к перегрузке, но SCTP предпочтительней из-за способности ограничивать «обратное давление» на экспортёра и ориентации на потоки, а не сообщения.

UDP можно использовать, хотя этот протокол не обеспечивает контроля перегрузок. В этом случае трафик IPFIX между экспортёром и сборщиком должен поддерживаться отдельно для минимизации риска потерь из-за перегрузок.

По умолчанию Collecting Process слушает соединения SCTP, TCP, UDP на порту 4739 и защищённые соединения SCTP, TCP, UDP - на порту 4740 (см. 11. Вопросы безопасности). Exporting Process по умолчанию пытается соединиться с одним из этих портов. **Должна** обеспечиваться возможность настройки экспортёра и сборщика на работу через другой порт.

10.2. SCTP

Ниже описана работа IPFIX по протоколу SCTP [RFC4960] с расширением PR-SCTP [RFC3758].

10.2.1. Предотвращение перегрузок

SCTP обеспечивает требуемый уровень предотвращения перегрузок по своему устройству. SCTP детектирует перегрузку на сквозном пути между IPFIX Exporting Process и IPFIX Collecting Process, ограничивая соответственно скорость передачи. Когда у IPFIX Exporting Process есть записи для экспорта, но обнаружена временная невозможность передачи по SCTP, экспортёр может подождать некоторое время до повтора передачи или отбросить запись. В последнем случае отброшенные данные экспорта **следует** учитывать, чтобы можно было указать объем отброшенных данных с использованием механизма, описанного в параграфе 4.3.

10.2.2. Надёжность

Транспортный протокол SCTP надёжен по определению, но может доставлять сообщения с частичной надёжностью [RFC3758].

Применение надёжной передачи SCTP для экспорта IPFIX само по себе не гарантирует доставки всех записей Data Record. Если на канале между процессами экспорта и сбора возникает перегрузка или требуется значительное число повторов передачи, выходные очереди в процессе экспорта могут заполниться и Exporting Process **может** приостановить, экспортировать или отбросить сообщения IPFIX. Если записи Data Record отбрасываются, потеря данных **должна** быть отражена в IPFIX Sequence Number.

10.2.3. MTU

SCTP обеспечивает требуемую для сообщений IPFIX фрагментацию на основе определения Path MTU (PMTU).

10.2.4. Организация и разрыв ассоциаций

Процесс экспорта IPFIX инициирует ассоциацию SCTP с IPFIX Collecting Process. Exporting Process **может** организовать более одной ассоциации (bundle в терминах SCTP) с процессом сбора.

Exporting Process **может** поддерживать более 1 ассоциации с разными процессами сбора (возможно, на одном хосте).

При отключении процесса экспорта (shut down), ему **следует** прервать (shut down) ассоциации SCTP.

Если Collecting Process больше не ждёт сообщений IPFIX, ему **следует** отключить (shut down) свою сторону ассоциации. Процессу сбора **следует** принимать и обрабатывать сообщения IPFIX, пока Exporting Process не закроет ассоциацию на своей стороне.

Если Collecting Process обнаруживает аномально прерванную ассоциацию SCTP, он **должен** продолжать прослушивание на предмет создания новой ассоциации.

Если Exporting Process обнаруживает аномально прерванную ассоциацию SCTP с процессом сбора, ему **следует** попытаться восстановить её.

Тайм-ауты для ассоциаций **следует** делать настраиваемыми.

10.2.5. Восстановление при отказах

Если Collecting Process не подтверждает попытку процесса экспорта создать ассоциацию, SCTP будет пытаться автоматически создать её, экспоненциально увеличивая интервал повтора. Экспортёр **может** сделать запись в журнале при возникновении тайм-аута SCTP. Значение тайм-аута следует делать настраиваемым для экспортёра.

Exporting Process **может** создать резервную (backup) ассоциацию SCTP с процессом сбора, если на том поддерживается восстановление при отказах.

10.2.6. Потоки

Exporting Process **может** запросить более 1 SCTP Stream на ассоциацию и каждый из этих потоков может служить для передачи сообщений IPFIX, содержащих Data Set, Template Set, Options Template Set. В зависимости от требований приложения Exporting Process может передавать Data Set с полной или частичной надёжностью, используя упорядоченную или неупорядоченную доставку в любом потоке SCTP, организованном при создании ассоциации SCTP.

IPFIX Exporting Process **может** использовать определение сервиса PR-SCTP в соответствии с разделом 4 [RFC3758] для поддержки частичной надёжности передачи сообщений IPFIX, содержащих лишь Data Set. Однако процессу экспорта **следует** помечать такие сообщения для повтора передачи, пока позволяют ресурсы и иные ограничения.

10.3. UDP

Ниже описана работа IPFIX по протоколу UDP [UDP].

10.3.1. Предотвращение перегрузок

В UDP нет встроенного механизма предотвращения перегрузок, поэтому использование протокола на путях с возможными перегрузками не рекомендуется. UDP **можно** применять в средах, где экспортёры и сборщики всегда взаимодействуют по выделенным каналам, где нет перегрузки, т. е. по каналам пропускная способность которых выше максимальной скорости передачи данных экспортёрами.

10.3.2. Надёжность

UDP не является надёжным транспортным протоколом и не может гарантировать доставку. Сообщения IPFIX от процесса экспорта в процесс сбора при использовании UDP могут теряться. UDP **недопустимо** применять с приложениями, не обеспечивающими устойчивости к некоторому уровню потерь сообщений IPFIX.

Процессу сбора **следует** выявлять потери и переупорядочение записей IPFIX Data Record по разрывам в порядковых номерах IPFIX. В случае UDP порядковый номер IPFIX содержит общее число (по модуля 2^{32}) IPFIX Data Record, переданных в транспортной сессии до получения этого сообщения IPFIX. Сборщику **следует** контролировать нарушение порядка, потерю и дублирование сообщений IPFIX, отслеживая значения Sequence Number.

Процессы экспорта, передающие сообщения IPFIX через UDP, **должны** включать корректную контрольную сумму UDP [UDP] в дейтаграммы UDP с сообщениями IPFIX.

10.3.3. MTU

Максимальный размер экспортируемых сообщений **должен** быть настраиваемым, чтобы общий размер пакетов не превышал PMTU. Если значение PMTU не известно, **следует** задавать максимальный размер пакета в 512 октетов.

10.3.4. Организация и разрыв сессий

UDP не использует явных соединений, поэтому при работе IPFIX по UDP нет реальной организации и разрыва сессий. Exporting Process начинает передавать сообщения IPFIX процессу сбора в один момент и прекращает в другой. Это может приводить к некоторому усложнению управления шаблонами, как описано выше в параграфе 8.4.

10.3.5. Восстановление при отказах и дублирование сессий

Поскольку UDP не использует явных соединений, процесс экспорта не может узнать у транспортного протокола о том, что Collecting Process больше не может принимать сообщения IPFIX, поэтому невозможно вызвать механизм восстановления. Однако Exporting Process **может** дублировать сообщение IPFIX нескольким процессам сбора.

10.4. TCP

Ниже описана работа IPFIX по протоколу TCP [TCP].

10.4.1. Предотвращение перегрузок

TCP контролирует скорость, с которой данные могут передаваться от Exporting Process в Collecting Process, применяя механизм, учитывающий перегрузки в сети и возможности получателя. Поэтому процесс экспорта может быть не способен передавать сообщения IPFIX со скоростью их генерации процессом измерения по причине перегрузки в сети или неспособности процесса сбора обрабатывать сообщения IPFIX достаточно быстро. При временной перегрузке Exporting Process может буферизовать сообщения IPFIX для передачи, но такая буферизация в любом случае ограничена по ресурсам и требованиям своевременной доставки, поэтому долгая или сильная перегрузка может приводить к блокировке процесса экспорта.

Когда у Exporting Process имеются записи Data Record для экспорта, но буфер передачи заполнен и экспортёр хочет избежать блокировки, он может принять решение об отбрасывании некоторых Data Record. Отброшенные записи **должны** учитываться, чтобы их номера можно было потом указать в отчёте, как описано в параграфе 4.3.

10.4.2. Надёжность

TCP обеспечивает надёжную доставку от процесса экспорта в процесс сбора.

10.4.3. MTU

TCP предоставляет потоковые услуги, а не дейтаграммы или пакетный сервис. Сообщения IPFIX при передаче по TCP разделяются по значению поля Length в заголовке сообщения IPFIX. Exporting Process может выбрать любой дозволенный размер для сообщения IPFIX, поскольку TCP выполняет сегментацию.

Exporting Process может выбирать размер сообщений IPFIX меньше разрешённого максимум для своевременного экспорта записей Data Record.

10.4.4. Организация и разрыв соединений

IPFIX Exporting Process инициирует соединение TCP с процессом сбора. Exporting Process может поддерживать более 1 активного соединения с разными процессами сбора (возможно, на одном хосте). Exporting Process может поддерживать более 1 активного соединения с одним процессом сбора для предотвращения блокировки head-of-line в доменах наблюдения.

Экспортёр **может** записывать в системный журнал случаи тайм-аута при организации соединения TCP. Значение тайм-аута следует делать настраиваемым для экспортёра.

При выключении Exporting Process (shut down), процессу экспорта **следует** разорвать (shut down) соединение TCP.

Когда Collecting Process больше не ждёт сообщения IPFIX, ему **следует** завершить соединение. Процессу сбора **следует** продолжать чтение сообщений IPFIX Message, пока Exporting Process не закроет свою сторону.

Если Collecting Process обнаруживает, что соединение TCP с Exporting Process разорвано аварийно, он **должен** продолжать прослушивание новых соединений.

Если Exporting Process обнаруживает, что соединение TCP с Collecting Process прервано аварийно, ему **следует** пытаться восстановить соединение. Число попыток и тайм-аут для восстановления **следует** делать настраиваемыми. В принятой по умолчанию конфигурации процессу экспорта **недопустимо** пытаться восстанавливать соединение чаще 1 раза в минуту.

10.4.5. Восстановление при отказах

Если Collecting Process не подтверждает попытку процесса экспорта создать соединение, TCP будет пытаться автоматически организовать его, экспоненциально увеличивая интервал повтора. Экспортёр **может** сделать запись в журнале при возникновении тайм-аута TCP. Значение тайм-аута следует делать настраиваемым для экспортёра.

Exporting Process **может** создать резервное (backup) соединение TCP с процессом сбора, если на том поддерживается восстановление при отказах.

11. Вопросы безопасности

Соображения по безопасности для протокола IPFIX были получены в результате анализ потенциальных угроз, отмеченных в разделе «Вопросы безопасности» документа с требованиями к IPFIX [RFC3917]. Требования безопасности для IPFIX указаны ниже.

1. Должен обеспечиваться механизм защиты конфиденциальности данных IPFIX, передаваемых процессом экспорта процессу сбора, для предотвращения раскрытия записей Flow Record, передаваемых через IPFIX.
2. Должен обеспечиваться механизм защиты конфиденциальности данных IPFIX, передаваемых процессом экспорта процессу сбора, для предотвращения вставки некорректных данных или управляющей информации (например, Template) или дублирования сообщений в потоке IPFIX Message.
3. Должен обеспечиваться механизм проверки подлинности процессов экспорта и сбора для предотвращения экспорта данных из неуполномоченного экспортёра или неуполномоченному сборщику.

Поскольку IPFIX может применяться для сбора информации в целях экспертизы и выставления счетов, атаки, нацеленные на искажение, отключение или перехват сведений из системы сбора IPFIX могут быть основной целью изощрённых сетевых атак.

Злоумышленник, имеющий возможность внедрять ложные сообщения в поток IPFIX Message, может влиять на приложения, использующие IPFIX (подмена данных) или сам процесс сбора IPFIX (изменение или отзыв шаблонов, смена опций), поэтому целостность сообщений IPFIX имеет важное значение.

Сообщения IPFIX могут содержать сведения, представляющие ценность для злоумышленника, включая данные конфигурации сети, а также трафик конечных пользователей и содержимое пакетов, поэтому должны приниматься меры по ограничению доступа. Когда информационный элемент включает пользовательские данные (payload), его **следует** передавать процессу сбора с применением защиты от перехвата. Подходящие для этого меры включают прямые соединения «точка-точка», недоступные для атакующих, или использование шифрования. Collecting Process отвечает за обеспечение удовлетворительного уровня защиты собираемых данных, включая при необходимости шифрование и/или обезличивание данных в отчётах (см. 11.8. Вопросы приватности собранных данных).

11.1. Применимость TLS и DTLS

Протоколы защиты на транспортном уровне TLS (Transport Layer Security) [RFC5246] и DTLS (Datagram Transport Layer Security) [RFC6347] разработаны для защиты конфиденциальности и целостности, а также проверки подлинности, требуемых протоколу IPFIX, без необходимости использования заранее распределенных ключей.

Процессы экспорта и сбора IPFIX, использующие TCP, **должны** поддерживать TLS версии 1.1 и **следует** поддерживать TLS версии 1.2 [RFC5246], включая обязательные для каждой версии шифры. Процессы экспорта и сбора IPFIX, использующие UDP или SCTP, **должны** поддерживать DTLS версии 1.0 и **следует** поддерживать DTLS версии 1.2 [RFC6347], включая обязательные для каждой версии шифры[1].

Отметим, что DTLS выбран механизмом защиты для SCTP. Хотя привязка TLS для SCTP определена в [RFC3436], она требует, чтобы все взаимодействия выполнялись через надёжные двухсторонние потоки, а также требует одно соединение TLS на поток, что несовместимо с обоснованием выбора SCTP в качестве транспортного протокола IPFIX.

Отметим, что при использовании DTLS имеется уязвимость MITM (man-in-the-middle), которой нет в TLS, позволяющая незаметно для отправителя и получателя удалять сообщения из потока. Кроме того, при использовании DTLS с SCTP атакующий может внедрять данные управления SCTP и отключать (shut down) ассоциации SCTP, вызывая потерю сообщений IPFIX, если они буферизованы вне ассоциации SCTP. Методы преодоления этих уязвимостей описаны в [RFC6083].

[1] Протоколы TLS версии 1.1 и DTLS версии 1.0 формально отменены RFC 8996 в марте 2021 г. *Прим. перев.*

При использовании DTLS с SCTP процесс экспорта **должен** гарантировать передачу каждого сообщения IPFIX в том же потоке SCTP, который бы применялся для передачи IPFIX Message напрямую через SCTP. Отметим, что DTLS может передавать свои управляющие сообщения в потоке 0 с полной гарантией, однако это не повлияет на обработку сообщений IPFIX в процессе сбора для потока 0, поскольку DTLS извлекает свои управляющие сообщения до передачи IPFIX Message на уровень приложения.

При использовании DTLS с SCTP или UDP **следует** применять расширение Heartbeat [RFC6520], особенно в долгосрочных транспортных сессиях, для сохранения статуса активности сессии.

Процессам экспорта и сбора **недопустимо** запрашивать, предлагать или применять любую версию уровня защищённого сокета (Secure Socket Layer или SSL), а также версии TLS до 1.1 по причине уязвимостей в ранних версиях TLS (см. Приложение E в [RFC5246]).

11.2. Применение

IPFIX Exporting Process инициирует связь с IPFIX Collecting Process и действует как клиент TLS или DTLS в соответствии с [RFC5246] и [RFC6347], а IPFIX Collecting Process выступает как сервер TLS или DTLS. Клиент DTLS создаёт защищённое соединение SCTP с портом 4740 сервера DTLS при использовании транспорта SCTP, с портом TCP 4740 сервера TLS при использовании транспорта TCP и с портом UDP 4740 сервера DTLS для транспорта UDP.

11.3. Взаимная проверка подлинности

При использовании TLS или DTLS процессы экспорта и сбора IPFIX следует идентифицировать по сертификатам с DNS-ID, как описано в параграфе 6.4 [RFC6125], включение Common Name (CN-ID) в сертификаты, указывающие процессы экспорта и сбора IPFIX, **не рекомендуется**.

Для предотвращения MITM-атак со стороны самозванных процессов экспорта или сбора (восприятие или экспорт данных неуполномоченной стороны) **должна** применяться взаимная аутентификация при использовании TLS и DTLS. Процессы экспорта **должны** сверять ссылочные идентификаторы процессов сбора, которым они экспортируют сообщения IPFIX, а процессы сбора **должны** сверять ссылочные идентификаторы процессов экспорта, от которых они получают сообщения IPFIX, с сохранёнными сертификатами. Процессу экспорта **недопустимо** экспортировать данные в непроверенные Collecting Process, а процессам сбора **недопустимо** воспринимать сообщения IPFIX от непроверенных Exporting Process.

Процессы экспорта и сбора **должны** поддерживать проверку сертификатов по явно уполномоченному списку партнёров, указанных Common Name, и **следует** поддерживать проверку ссылочных идентификаторов на соответствие DNS-ID или CN-ID с поиском партнёра через DNS (lookup).

Процессы экспорта и сбора **должны** применять отличные от NULL шифры для аутентификации, защиты целостности и конфиденциальности.

11.4. Защита от DoS-атак

Злоумышленник может организовать атаку на службы (denial-of-service или DoS) системы IPFIX напрямую, отправляя большой объём трафика процессам сбора или опосредованно, создавая большой объем измеряемого трафика.

Прямые DoS-атаки могут также включать истощение состояний на транспортном уровне (например, создание множества ожидающих соединений) или в процессе сбора IPFIX (например, отправка Flow Record, ожидающих шаблонов, большое число Option и т. п.).

SCTP поддерживает механизм обмена cookie, предназначенный для защиты от истощения состояний SCTP при DoS-атаках. В TCP имеется механизм SYN cookie для смягчения таких атак, который **следует** применять всем процессам сбора, воспринимающим соединения TCP. В DTLS также имеется механизм cookie для защиты от истощения состояний серверов DTLS.

Читателю следует отметить, что невозможно предотвратить обработку поддельных сообщений IPFIX (и создание состояний) при работе по UDP и SCTP. Применение TLS и DTLS обычно может предотвратить создание ложных состояний, но эти протоколы сами подвержены атакам на истощение состояний. Поэтому сборщикам **следует** применять ограничение скорости для TLS и DTLS (например, ограничение числа новых сессий TLS и DTLS по времени).

Атаки с истощением состояний IPFIX можно ослабить, ограничивая частоту создания новых соединений (ассоциаций) в процессах сбора, ограничения скорости восприятия сообщений IPFIX сборщиками и адаптивного ограничения числа хранимых состояний, особенно для записей, ожидающих шаблон. Эти ограничения скорости и состояний **могут** обеспечивать процессы сбора, при этом ограничения **следует** делать настраиваемыми. Процесс сбора IPFIX может устранить риск истощения состояний от недоверенных узлов, требуя взаимной аутентификации TLS или DTLS, позволяющей процессу сбора воспринимать сообщения IPFIX только из доверенных источников.

В части косвенных атак на службы поведение IPFIX при перегрузке зависит от транспортного протокола. При работе по TCP контроль перегрузок приведёт к замедлению потока сообщений IPFIX и в конечном итоге остановит его, блокируя систему IPFIX. В SCTP ситуация несколько лучше, поскольку некоторые сообщения IPFIX будут получены процессом сбора в результате предотвращения блокировки head-of-line за счёт множества потоков SCTP и функций частичной гарантии, что может позволить увидеть атаку. Похожая ситуация возникает и для UDP, поскольку некоторые дейтаграммы будут поступать в процесс сбора за счёт эффективного применения выборки к потоку сообщений IPFIX, что может сделать атаку видимой.

Для минимизации потери сообщений IPFIX при перегрузке можно использовать некоторые механизмы дифференциации услуг для приоритизации трафика IPFIX по отношению к другому трафику в канале. Кроме того, можно доставлять сообщения IPFIX через выделенную сеть. В этом случае требуется обеспечить в такой сети возможность обработки пиковых потоков сообщений IPFIX.

11.5. Когда DTLS или TLS не подходит

Применение DTLS или TLS в некоторых случаях может оказаться невозможным из-за проблем с производительностью или иных ограничений. Без взаимной аутентификации TLS или DTLS процессы экспорта IPFIX могут использовать IP-адрес отправителя для проверки подлинности партнёра. Правила выделения адресов IPпроцессам экспорта и сбора из указанных диапазонов и применение фильтрации на входе для предотвращения подмены могут усилить полезность такого подхода. Полный вынос трафика IPFIX в отдельную сеть, когда это возможно, может повысить защищенность ещё больше. В любом случае использование открытых процессов сбора (принимающих сообщения IPFIX от любых Exporting Process, независимо от адреса IP или отождествления) настоятельно не рекомендуется.

Современные реализации TCP и SCTP устойчивы к атакам со вставкой вслепую (см. [RFC4960] и [RFC6528]), однако в UDP такой защиты нет. По этой причине доставка сообщений IPFIX по протоколу UDP без DTLS не защищена и такой трафик **следует** выносить в отдельную сеть.

11.6. Запись атак на IPFIX

Процессы сбора IPFIX **должны** обнаруживать возможную потерю и вставку сообщений IPFIX, отслеживая IPFIX, и **следует** поддерживать механизм записи для указания в отчёте нарушивших порядок сообщений. Отметим, что атакующий может воспользоваться обработкой переупорядоченных соединений в процессе сбора, поэтому следует соблюдать осторожность при такой обработке. Например, Collecting Process, просто сбрасывающий ожидаемый порядковый номер при получении большего Sequence Number, может быть временно «ослеплен» преднамеренной вставкой больших порядковых номеров.

Процессам экспорта и сбора IPFIX **следует** записывать любые попытки соединения с отказом при аутентификации, будь то представление неуполномоченного или несоответствующего сертификата при взаимной аутентификации TLS/DTLS или попытка соединения с неразрешенного адреса IP при работе без TLS или DTLS.

Процессам экспорта и сбора IPFIX **следует** обнаруживать и записывать любые попытки сброса ассоциации SCTP или соединения TCP.

11.7. Защита коллектора

Безопасность сборщика и его реализации важна для общей защиты, однако полный набор рекомендация по защите сборщиков выходит за рамки этого документа.

Поскольку IPFIX использует кодирование с префиксом размера, разработчикам сборщиков следует принять меры для обнаружения несогласованных значений, которые могут влиять на декодирование сообщений IPFIX и корректность работы при наличии таких несогласованных значений. В частности, должна проверяться согласованность размера IPFIX Message, Set и информационных элементов с переменным размером для предотвращения проблем с буферами.

Разработчикам сборщиков следует уделять особое внимание кодировке UTF-8 в данных типа string, поскольку интерпретация UTF-8 с некорректным форматом может вести к уязвимости (см. 6.1.6. string и octetArray).

11.8. Вопросы приватности собранных данных

Данные о потоках, которые экспортируют Exporting Process и собирают Collecting Process, обычно содержат сведения о трафике в наблюдаемой сети. Эта информация может быть приватной и конфиденциальной. Хранилище таких данных должно быть защищено техническими и административными мерами для сохранения приватности пользователей наблюдаемой сети. Полная спецификация таких мер выходит за рамки документа и зависит от приложения и используемой технологии хранения.

12. Вопросы управления

В [RFC6615] задан модуль MIB, определяющий управляемые объекты для мониторинга устройств IPFIX, включая базовую конфигурацию. Эта база MIB может служить для измерения влияния экспорта IPFIX на контролируемую сеть и включает таблицы, охватывающие транспортные сессии, определение кэша, точки наблюдения, Template и Options Template, свойства экспорта (восстановление, балансировка, дубликаты) и статистику экспорта по процессам, сессиям и шаблонам.

С точки зрения эксплуатации важная функция этого модуля обеспечивается таблицей Transport Session Statistical, указывающей скорость (байт/сек), с которой коллектор получает сообщения IPFIX, переданные экспортером. Особый интерес представляет таблица Transport Session Statistical из параграфа 5.8.1 этого модуля MIB, раскрывающая скорость сбора или экспорта сообщений IPFIX, которая позволяет измерить полосу для экспорта IPFIX.

В [RFC6727] описаны расширения модуля IPFIX-SELECTOR-MIB, заданного в [RFC6615], и указаны управляемые объекты для предоставления сведений о применяемых функциях отбора пакетов и их параметрах (фильтрация и выборка).

Поскольку IPFIX-SELECTOR-MIB [RFC6615] и PSAMP-MIB [RFC6727] содержат объекты, доступные лишь для чтения, они не подходят для настройки устройств IPFIX. В [RFC6728] задана модель данных конфигурации для протоколов IPFIX и PSAMP, использующая протокол настройки сети (Network Configuration Protocol или NETCONF). Эта модель охватывает процессы выбора, кэширование, а также процессы экспорта и сбора в устройствах IPFIX и PSAMP и описана диаграммами классов унифицированного языка моделирования UML (Unified Modeling Language), а также задана формально с помощью YANG. Данные конфигурации представлены на расширяемом языке разметки (Extensible Markup Language или XML).

Несколько механизмов, заданных с протоколом IPFIX могут помочь в отслеживании и сокращении полосу, используемой для экспорта IPFIX:

- метод экономии пропускной способности при экспорте избыточной информации в IPFIX [RFC5473];
- эффективный метод для экспорта двухсторонних потоков [RFC5103];
- метод определения и экспорта комплексных структур данных [RFC6313].

Кроме того, можно применять PSAMP [RFC5474] для экспорта пакетов, собранных статистическими и иными методами, которые могут быть применимы для многих областей мониторинга, где подходит и IPFIX. PSAMP также обеспечивает контроль влияния на измеряемую сеть через скорость выборки. Набор методов отбора пакетов (выборка, фильтрация, обработка), стандартизованных в PSAMP, описан в [RFC5475]. PSAMP также задаёт явно определяемый предел скорости экспорта в параграфе 8.4 [RFC5474].

13. Взаимодействие с IANA

Агентство IANA обновило реестр IPFIX Information Elements [IANA-IPFIX], заменив все ссылки на RFC 5101 ссылками на данный документ.

Сообщения IPFIX включают два поля с выделяемыми значениями. Это IPFIX Version Number с номером версии протокола IPFIX, который применяется при экспорте сообщений IPFIX, и поле IPFIX Set ID, указывающее тип каждого набора информации в сообщении IPFIX.

Информационные элементы, применяемые в IPFIX и субреестры значений информационных элементов управляются IANA [IANA-IPFIX], как и Private Enterprise Number, используемые в фирменных информационных элементах [IANA-PEN]. Данный документ не меняет эти реестры.

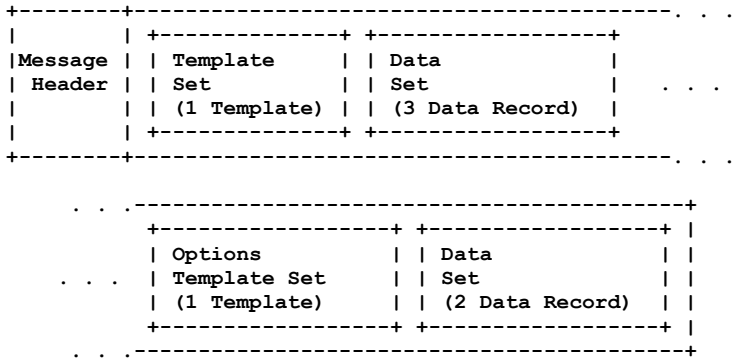
Значение IPFIX Version Number = 0x000a (10) зарезервировано для протокола IPFIX, заданного этим документом. Значения 0 и 1 для Set ID не используются по историческим причинам [RFC3954]. Значение Set ID = 2 зарезервировано для Template Set, Set ID = 3 - для Options Template Set. Остальные значения Set ID (4 - 255) зарезервированы на будущее. Значения Set ID > 255 применяются для Data Set.

Новые значения в реестрах IPFIX Version Number и IPFIX Set IDs выделяются по процедуре Standards Action [RFC5226], т. е. требуется документ RFC категории Standards Track, одобренный IESG.

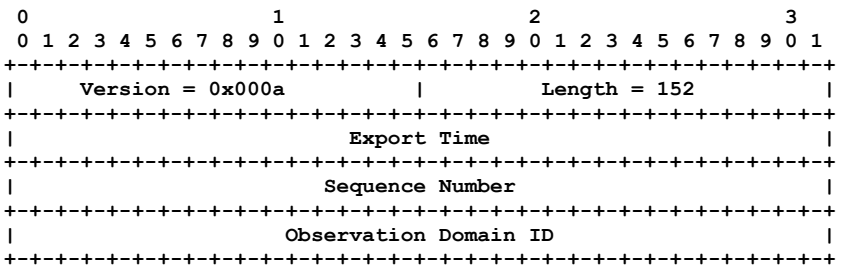
Приложение A. Примеры кодирования IPFIX

В этом ненормативном приложении представлены примеры кодирования IPFIX.

Рассмотрим пример сообщения IPFIX, содержащего Template Set, Data Set (3 Data Record), Options Template Set и ещё Data Set (2 Data Record, связанные с предшествующим Options Template Record). Сообщение IPFIX будет иметь вид



A.1. Пример заголовка сообщения



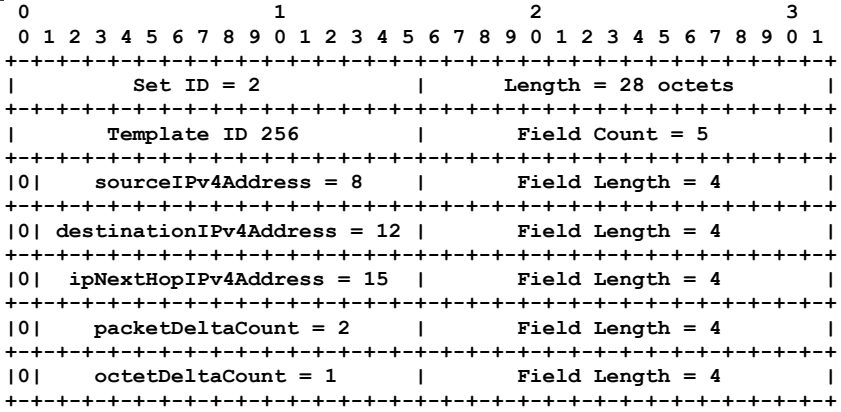
A.2. Примеры Template Set

A.2.1. Template Set с информационными элементами IANA

Нужно передать в отчёте перечисленные ниже информационные элементы.

- Адрес отправителя IPv4: sourceIPv4Address [IANA-IPFIX] размером 4 октета.
- Адрес получателя IPv4: sourceIPv4Address [IANA-IPFIX] размером 4 октета.
- Адрес next-hop (IPv4): ipNextHopIPv4Address [IANA-IPFIX] размером 4 октета.
- Число пакетов в потоке: packetDeltaCount [IANA-IPFIX] размером 4 октета.
- Число октетов в потоке: octetDeltaCount [IANA-IPFIX] размером 4 октета.

Template Set будет иметь вид

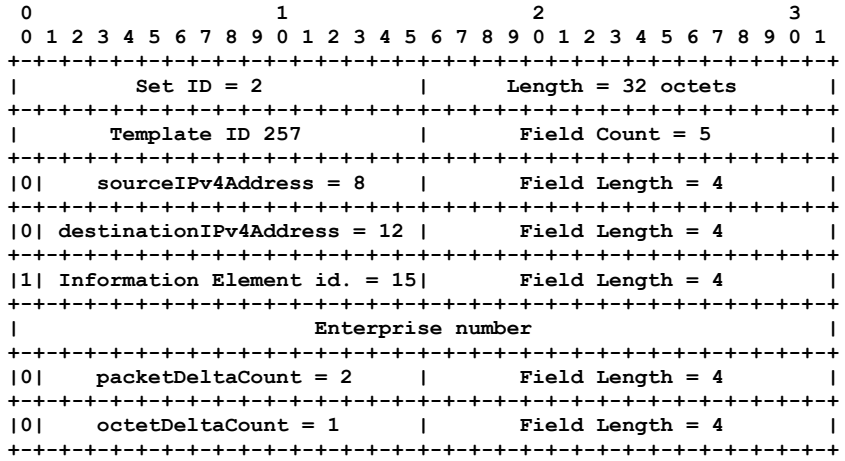


A.2.2. Template Set с фирменными информационными элементами

Нужно передать в отчёте перечисленные ниже информационные элементы.

- Адрес отправителя IPv4: sourceIPv4Address [IANA-IPFIX] размером 4 октета.
- Адрес получателя IPv4: sourceIPv4Address [IANA-IPFIX] размером 4 октета.
- Фирменный Information Element с фирменными сведениями типа 15 и размером 4 октета
- Число пакетов в потоке: packetDeltaCount [IANA-IPFIX] размером 4 октета.
- Число октетов в потоке: octetDeltaCount [IANA-IPFIX] размером 4 октета.

Template Set будет иметь вид



A.3. Пример Data Set

В этом примере отчёт содержит три записи Flow Record, как показано в таблице. Дополнение здесь не требуется.

IP-адрес источника	*IP-адрес получателя*	*Адрес Next-Hop*	*Число пакетов*	*Число октетов*
192.0.2.12	192.0.2.254	192.0.2.1	5009	5344385
192.0.2.27	192.0.2.23	192.0.2.2	748	388934
192.0.2.56	192.0.2.65	192.0.2.3	5	6534

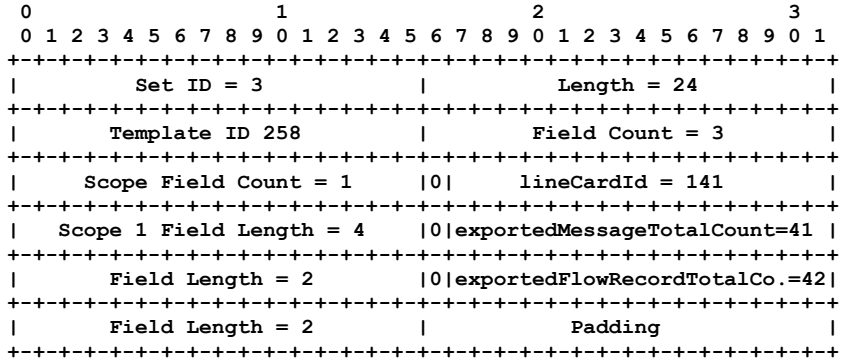
A.4. Примеры Options Template Set

A.4.1. Пример Options Template Set с информационными элементами IANA

На уровне линейной платы (в маршрутизаторе с 2 платами) передаются указанные ниже информационные элементы.

- Общее число сообщений IPFIX: exportedMessageTotalCount [IANA-IPFIX] размером 2 октета.
- Общее число экспортируемых потоков: exportedFlowRecordTotalCount [IANA-IPFIX] размером 2 октета.

Линейная плата представлена элементом lineCardId [IANA-IPFIX] в поле Scope. Options Template Set имеет вид

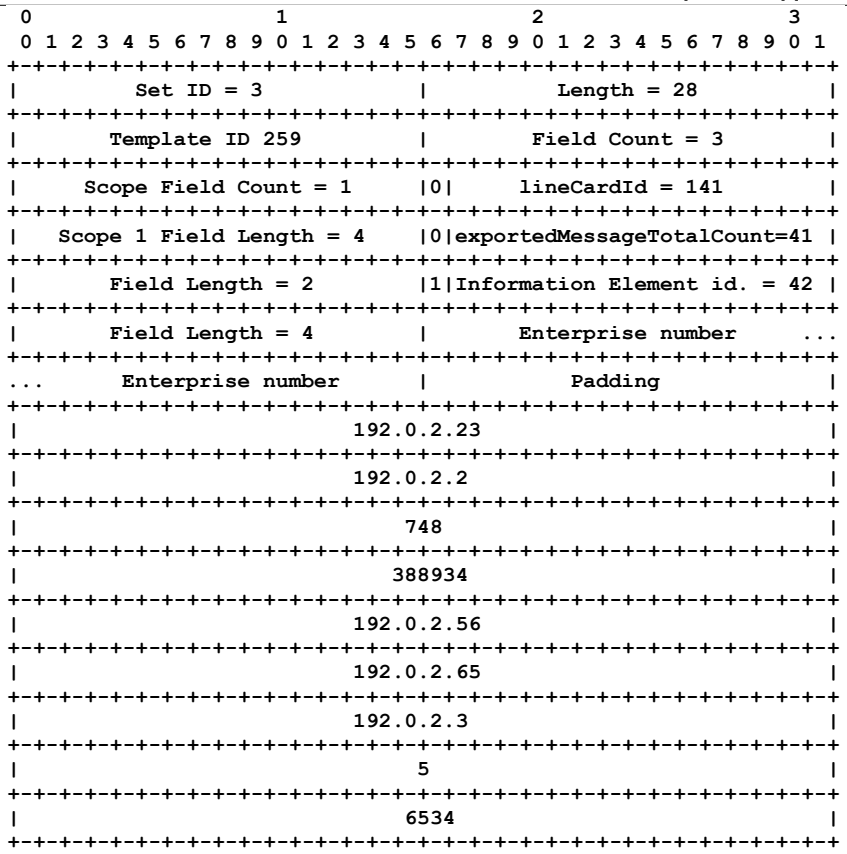


A.4.2. Options Template Set с фирменными информационными элементами

На уровне линейной платы (в маршрутизаторе с 2 платами) передаются указанные ниже информационные элементы.

- Общее число сообщений IPFIX: exportedMessageTotalCount [IANA-IPFIX] размером 2 октета.
- Зависящее от предприятия число экспортируемых потоков с типом 42 и размером 4 октета.

Линейная плата представлена элементом lineCardId [IANA-IPFIX] в поле Scope. Options Template Set имеет вид

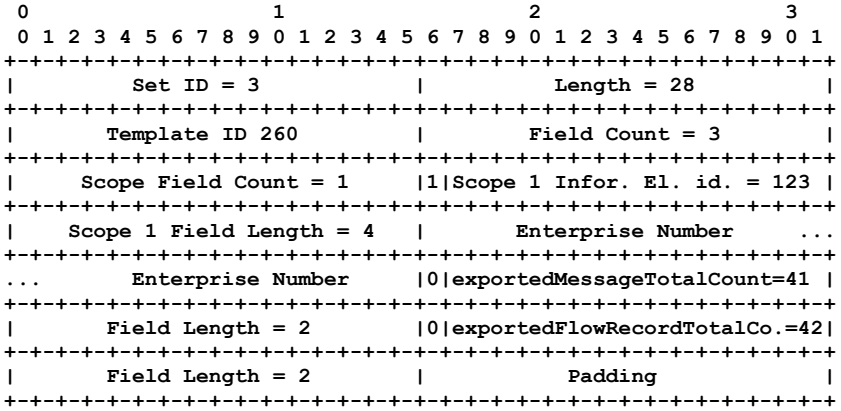


A.4.3. Options Template Set с фирменным полем Scope

В этом примере нужно экспортировать те же сведения, что и в примере приложения A.4.1

- Общее число сообщений IPFIX: exportedMessageTotalCount [IANA-IPFIX] размером 2 октета.
- Общее число экспортируемых потоков: exportedFlowRecordTotalCount [IANA-IPFIX] размером 2 октета.

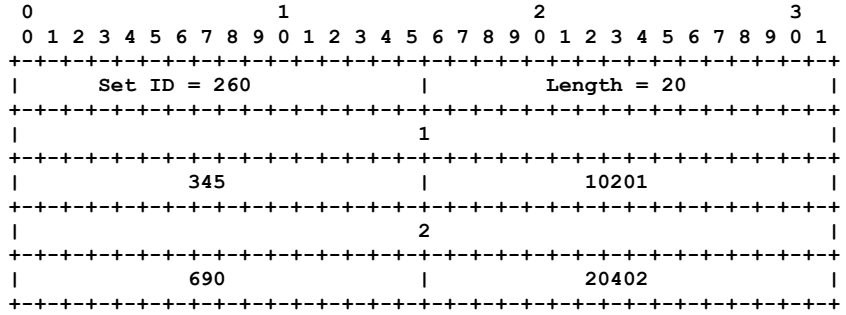
На этот раз данные относятся к фирменной области действия, указанной фирменным информационным элементом с номером 123. Options Template Set имеет вид



A.4.4. Data Set с фирменным полем Scope

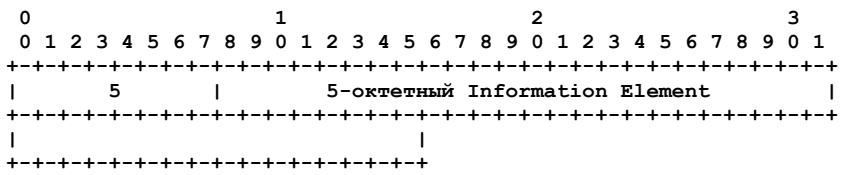
В этом примере передаются две записи Data Record.

Поле Enterprise 123	*Сообщение IPFIX*	*Экспортируемые Flow Record*
1	345	10201
2	690	20402

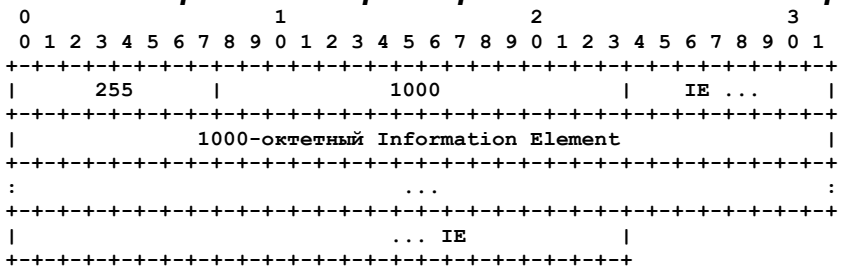


A.5. Примеры информационных элементов переменного размера

A.5.1. Пример информационного элемента переменного размера (< 255 октетов)



A.5.2. Пример элемента переменного размера с 3-октетным кодированием



Нормативные документы

[IANA-IPFIX] IANA, "IP Flow Information Export (IPFIX) Entities", <http://www.iana.org/assignments/ipfix/>.

[RFC1014] Sun Microsystems, Inc., "XDR: External Data Representation Standard", RFC 1014, June 1987.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3436] Jungmaier, A., Rescorla, E., and M. Tuexen, "Transport Layer Security over Stream Control Transmission Protocol", RFC 3436, December 2002.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[RFC3758] Stewart, R., Ramalho, M., Xie, Q., Tuexen, M., and P. Conrad, "Stream Control Transmission Protocol (SCTP) Partial Reliability Extension", RFC 3758, May 2004.

[RFC4960] Stewart, R., Ed., "Stream Control Transmission Protocol", RFC 4960, September 2007.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[RFC5905] Mills, D., Martin, J., Ed., Burbank, J., and W. Kasch, "Network Time Protocol Version 4: Protocol and Algorithms Specification", RFC 5905, June 2010.

[RFC6125] Saint-Andre, P. and J. Hodges, "Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS)", RFC 6125, March 2011.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012.

[RFC6520] Seggelmann, R., Tuexen, M., and M. Williams, "Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS) Heartbeat Extension", RFC 6520, February 2012.

[RFC7012] Claise, B., Ed., and B. Trammell, Ed., "Information Model for IP Flow Information Export (IPFIX)", RFC 7012, September 2013.

[TCP] Postel, J., "Transmission Control Protocol", STD 7, RFC 793, September 1981.

[UDP] Postel, J., "User Datagram Protocol", STD 6, RFC 768, August 1980.

Дополнительная литература

[IEEE.754.2008] Institute of Electrical and Electronics Engineers, "IEEE Standard for Floating-Point Arithmetic", IEEE Standard 754, August 2008.

[IPFIX-MED-PROTO] Claise, B., Kobayashi, A., and B. Trammell, "Operation of the IP Flow Information Export (IPFIX) Protocol on IPFIX Mediators", Work in Progress, July 2013.

[IANA-PEN] IANA, "Private Enterprise Numbers", <http://www.iana.org/assignments/enterprise-numbers/>.

[POSIX.1] IEEE 1003.1-2008, "IEEE Standard for Information Technology - Portable Operating System Interface (POSIX(R))", 2008.

[RFC2579] McCloghrie, K., Ed., Perkins, D., Ed., and J. Schoenwaelder, Ed., "Textual Conventions for SMIv2", STD 58, RFC 2579, April 1999.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

[RFC3917] Quittek, J., Zseby, T., Claise, B., and S. Zander, "Requirements for IP Flow Information Export (IPFIX)", RFC 3917, October 2004.

[RFC3954] Claise, B., Ed., "Cisco Systems NetFlow Services Export Version 9", RFC 3954, October 2004.

[RFC5101] Claise, B., Ed., "Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of IP Traffic Flow Information", RFC 5101, January 2008.

[RFC5103] Trammell, B. and E. Boschi, "Bidirectional Flow Export Using IP Flow Information Export (IPFIX)", RFC 5103, January 2008.

[RFC5153] Boschi, E., Mark, L., Quittek, J., Stiemerling, M., and P. Aitken, "IP Flow Information Export (IPFIX) Implementation Guidelines", RFC 5153, April 2008.

[RFC5470] Sadasivan, G., Brownlee, N., Claise, B., and J. Quittek, "Architecture for IP Flow Information Export", RFC 5470, March 2009.

[RFC5471] Schmoll, C., Aitken, P., and B. Claise, "Guidelines for IP Flow Information Export (IPFIX) Testing", RFC 5471, March 2009.

[RFC5472] Zseby, T., Boschi, E., Brownlee, N., and B. Claise, "IP Flow Information Export (IPFIX) Applicability", RFC 5472, March 2009.

[RFC5473] Boschi, E., Mark, L., and B. Claise, "Reducing Redundancy in IP Flow Information Export (IPFIX) and Packet Sampling (PSAMP) Reports", RFC 5473, March 2009.

[RFC5474] Duffield, N., Ed., Chiou, D., Claise, B., Greenberg, A., Grossglauser, M., and J. Rexford, "A Framework for Packet Selection and Reporting", RFC 5474, March 2009.

[RFC5475] Zseby, T., Molina, M., Duffield, N., Niccolini, S., and F. Raspall, "Sampling and Filtering Techniques for IP Packet Selection", RFC 5475, March 2009.

[RFC5476] Claise, B., Ed., Johnson, A., and J. Quittek, "Packet Sampling (PSAMP) Protocol Specifications", RFC 5476, March 2009.

[RFC5477] Dietz, T., Claise, B., Aitken, P., Dressler, F., and G. Carle, "Information Model for Packet Sampling Exports", RFC 5477, March 2009.

[RFC5610] Boschi, E., Trammell, B., Mark, L., and T. Zseby, "Exporting Type Information for IP Flow Information Export (IPFIX) Information Elements", RFC 5610, July 2009.

[RFC5655] Trammell, B., Boschi, E., Mark, L., Zseby, T., and A. Wagner, "Specification of the IP Flow Information Export (IPFIX) File Format", RFC 5655, October 2009.

[RFC6083] Tuexen, M., Seggelmann, R., and E. Rescorla, "Datagram Transport Layer Security (DTLS) for Stream Control Transmission Protocol (SCTP)", RFC 6083, January 2011.

[RFC6183] Kobayashi, A., Claise, B., Muenz, G., and K. Ishibashi, "IP Flow Information Export (IPFIX) Mediation: Framework", RFC 6183, April 2011.

[RFC6313] Claise, B., Dhandapani, G., Aitken, P., and S. Yates, "Export of Structured Data in IP Flow Information Export (IPFIX)", RFC 6313, July 2011.

[RFC6526] Claise, B., Aitken, P., Johnson, A., and G. Muenz, "IP Flow Information Export (IPFIX) Per Stream Control Transmission Protocol (SCTP) Stream", RFC 6526, March 2012.

[RFC6528] Gont, F. and S. Bellovin, "Defending against Sequence Number Attacks", RFC 6528, February 2012.

[RFC6615] Dietz, T., Ed., Kobayashi, A., Claise, B., and G. Muenz, "Definitions of Managed Objects for IP Flow Information Export", RFC 6615, June 2012.

[RFC6727] Dietz, T., Ed., Claise, B., and J. Quittek, "Definitions of Managed Objects for Packet Sampling", RFC 6727, October 2012.

[RFC6728] Muenz, G., Claise, B., and P. Aitken, "Configuration Data Model for the IP Flow Information Export (IPFIX) and Packet Sampling (PSAMP) Protocols", RFC 6728, October 2012.

[UTF8-EXPLOIT] Davis, M. and M. Suignard, "Unicode Technical Report #36: Unicode Security Considerations", The Unicode Consortium, July 2012.

Благодарности

Спасибо Ganesh Sadasivan за существенный вклад на ранних этапах спецификации протокола, Juergen Quittek за координацию между IPFIX и PSAMP, Nevil Brownlee, Dave Plonka и Andrew Johnson за подробные рецензии, Randall Stewart и Peter Lei за их опыт и вклад для SCTP, Martin Djernaes за первый отзыв по разделу SCTP, Michael Behringer и Eric Vyncke за советы и знания в части безопасности, Michael Tuexen за помощь в разделе DTLS, Elisa Boschi за вклад в улучшение раздела SCTP, Mark Fullmer, Sebastian Zander, Jeff Meyer, Maurizio Molina, Carter Bullard, Tal Givoly, Lutz Mark, David Moore, Robert Lowe, Paul Calato, Andrew Feren, Gerhard Muenz, Sue Hares и многим другим за технические обзоры и отклики. Отдельная благодарность Adrian Farrel за внимание к аспектам управления и эксплуатации.

Участники работы

Stewart Bryant
Cisco Systems
10 New Square, Bedfont Lakes
Feltham, Middlesex TW18 8HA
United Kingdom
EMail: stbryant@cisco.com

Simon Leinen
SWITCH
Werdstrasse 2
P.O. Box 8021
Zurich
Switzerland
Phone: +41 44 268 1536
EMail: simon.leinen@switch.ch

Thomas Dietz
NEC Europe Ltd.
NEC Laboratories Europe
Network Research Division
Kurfuersten-Anlage 36
69115 Heidelberg
Germany
Phone: +49 6221 4342-128

EMail: Thomas.Dietz@nw.neclab.eu

Адреса авторов

Benoit Claise (editor)
Cisco Systems, Inc.
De Kleetlaan 6a b1
1831 Diegem
Belgium
Phone: +32 2 704 5622
EMail: bclaise@cisco.com

Brian Trammell (editor)
Swiss Federal Institute of Technology Zurich
Gloriastrasse 35
8092 Zurich
Switzerland
Phone: +41 44 632 70 13
EMail: trammell@tik.ee.ethz.ch

Paul Aitken
Cisco Systems, Inc.
96 Commercial Quay
Commercial Street, Edinburgh EH6 6LX
United Kingdom
Phone: +44 131 561 3616
EMail: paitken@cisco.com

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru