Internet Engineering Task Force (IETF)
Request for Comments: 8572
Category: Standards Track
ISSN: 2070-1721

K. Watsen
Watsen Networks
I. Farrer
Deutsche Telekom AG
M. Abrahamsson
T-Systems

April 2019

Secure Zero Touch Provisioning (SZTP)

Защищённая автоматическая инициализация

Аннотация

В этом документе представлен метод защищённой подготовки устройства, загружаемого с заданными производителем настройками. Варианты решения позволяют применять его как в общественных, так и в частных сетях. На этапах подготовки можно обновить загрузочный образ, установить начальную конфигурацию и выполнить произвольные сценарии для решения дополнительных задач. После этого обновлённое устройство сможет устанавливать защищённые соединения с другими развёрнутыми системами управления сетью, например, NETCONF (RFC 6241) или RESTCONF (RFC 8040).

Статус документа

Документ содержит проект стандарта Internet (Standards Track).

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

Информацию о текущем статусе документа, ошибках и способах обратной связи можно найти по ссылке https://www.rfc-editor.org/info/rfc8572.

Оглавление

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

1. Введение

Фундаментальным требованием бизнеса для операторов сетей является снижение расходов, когда это возможно. Для операторов развёртывание устройств в разных местах может вести к значительным расходам, поскольку отправка квалифицированных специалистов на каждую площадку для установки непомерно дорога и требует много персонала.

Этот документ определяет защищённую автоматическую инициализацию (Secure Zero Touch Provisioning или SZTP) - стратегию начальной загрузки для безопасного получения данных начальной загрузки без каких-либо действий персонала сверх монтажа и подключения сетевых и питающих кабелей. За счёт этого SZTP позволяет запускать устройства на удалённых площадках персоналу без технических навыков, не требуя вмешательства оператора.

Решение SZTP включает обновление загрузочного образа, установку начальных параметров и выполнение произвольных сценариев для решения дополнительных задач. Обновлённое устройство способно организовать защищённые соединения с другими системами. Например, оно может создать соединения NETCONF [RFC6241] и/или RESTCONF [RFC8040] с развёрнутыми системами управления сетью.

Документ в первую очередь относится к физическим устройствам, где исходное состояние (5. Детали устройства) задаётся при производстве. Решение SZTP можно расширить для поддержки виртуальных машин и похожих логических конструкций, но детали этого оставлены для будущих работ.

1.1. Варианты применения

- Устройство подключено к удалённо управляемой сети. Этот вариант включает такие сценарии, как филиал или круглосуточный магазин, где устройство подключается к шлюзу доступа в сеть ISP. В предположении

невозможности настроить сеть ISP для обеспечения начальной загрузки и отсутствия вблизи другого устройства, которое могло бы обеспечить загрузку, у устройства остаётся лишь один способ - обратиться к серверу начальной загрузки через Internet.

- Устройство подключено к локально управляемой сети. Этот вариант отличается возможностью использовать близлежащие устройства, которые могут обеспечить начальную загрузку. Если таких устройств нет или они недоступны, можно воспользоваться предыдущим вариантом (сеть с удалённым управлением).

Концептуальные рабочие процессы развёртывания SZTP представлены а Приложении C.

1.2. Терминология

Artifact - артефакт
Этот термин служит в документе для представления трёх артефактов (объектов данных), определённых в разделе 3 (conveyed information, ownership voucher, owner certificate). Эти артефакты совместно обеспечивают все данные начальной загрузки, которые устройство может использовать.

Bootstrapping Data - данные начальной загрузки
Набор данных, которые устройство может получить в процессе начальной загрузки. В частности, к ним относятся три артефакта, описанные в разделе 3 (conveyed information, owner certificate, ownership voucher).

Bootstrap Server - сервер начальной загрузки
Любой сервер RESTCONF с реализацией модуля YANG, заданного в параграфе 7.3.

Conveyed Information - доставляемая (передаваемая) информация
Сведения о перенаправлении или вводные данные. Это один из трёх артефактов, описанных в разделе 3.

Device - устройство
Элемент сети, которому требуется начальная загрузка (5. Детали устройства).

Manufacturer - изготовитель
Изготовитель устройства или его полномочный представитель.

Network Management System (NMS) - система управления сетью
Система управления в конкретном развёртывании, за введение устройства в которую отвечает процесс начальной загрузки. С точки зрения устройства по завершении начальной загрузки в роли NMS выступает клиент NETCONF или RESTCONF.

Onboarding Information - вводные сведения
Это один из двух типов передаваемой информации, определённых в этом документе, а другим являются сведения о перенаправлении. Вводные сведения формально определены в контейнере onboarding-information структуры yang-data в параграфе 6.3.

Onboarding Server - сервер вводных сведений
Сервер начальной загрузки, лишь возвращающий вводные сведения.

Owner - владелец
Человек или организация, купившие или иным способом получившие устройство во владение.

Owner Certificate - сертификат владельца
Сертификат X.509, связывающий отождествление владельца с открытым ключом, который устройство применяет для проверки подписи через передаваемые сведения. Сертификат владельца может передаваться вместе с цепочкой промежуточных сертификатов, ведущей к известной точке доверия. Сертификат владельца является одним из трёх артефактов начальной загрузки, описанных в разделе 3.

Ownership Voucher - ваучер владения
Артефакт ваучера, определённый в [RFC8366] и являющийся одним из трёх артефактов начальной загрузки, описанных в разделе 3.

Redirect Information - сведения о перенаправлении
Это один из двух типов передаваемой информации, определённых в этом документе, а другим являются вводные сведения. Сведения о перенаправлении формально определены в контейнере conveyed-information структуры yang-data в параграфе 6.3.

Redirect Server - сервер перенаправления
Сервер начальной загрузки, возвращающий лишь сведения о перенаправлении. Такой сервер особенно полезен при размещении у изготовителя, как общеизвестный ресурс (например, в Internet) для перенаправления устройств на серверы начальной загрузки конкретного развёртывания.

Signed Data - подписанные данные
Передаваемые сведения, которые были подписаны, в частности, с применением секретного ключа владельца устройства.

Unsigned Data
Передаваемые сведения, которые не подписаны.

1.3. Уровни требований

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

1.4. Диаграммы деревьев

В диаграммах деревьев применяется нотация, заданная в [RFC8340].

2. Типы передаваемых сведений

В этом документе определяется два типа передаваемой информации, к которой устройства могут иметь доступ в процессе начальной загрузки. Эти сведения описаны в данном разделе, а примеры даны в параграфе 6.2.

2.1. Сведения о перенаправлении

Сведения о перенаправлении отправляют устройство на другой сервер начальной загрузки. Эти сведения представляются списком серверов начальной загрузки, с указанием имён (адресов IP), а также могут указывать порт и включать сертификат точки доверия, который устройство может использовать для проверки подлинности сервера.

Сведения о перенаправлении в модели данных YANG формально определены в контейнере redirect-information модуля YANG, представленного в параграфе 6.3. Дерево этого контейнера представлено ниже.

```
+--:(redirect-information)
   +-- redirect-information
      +-- bootstrap-server* [address]
         +-- address          inet:host
         +-- port?            inet:port-number
         +-- trust-anchor?    cms
```

Сведения о перенаправлении могут быть доверенными или недоверенными. Доверенными они считаются при получении через защищённое соединение с доверенным сервером начальной загрузки или при наличии подписи владельца устройства. В иных случаях сведения о перенаправлении считаются недоваренными.

Доверенные данные о перенаправлении полезны для обеспечения устройству возможности создания защищённого канала к указанному серверу начальной загрузки, что возможно при наличии в этих данных сертификата точки доверия для сервера начальной загрузки.

Недоверенные сведения о перенаправлении полезны для отправки устройства к серверу загрузки с подписанными данными для него. Если данные о перенаправлении не являются доверенными, устройство отбрасывает любые сертификаты точек доверия, которые могут быть включены.

Обработка сведений о перенаправлении описана в параграфе 5.5.

2.2. Вводные сведения

Вводные сведения обеспечивают данные, требуемые устройству для начальной загрузки и организации защищённых соединений с другими системами. В соответствии с определением в этом документе вводные сведения могут подробно указывать загрузочный образ для устройства, начальную конфигурацию и сценарии, которые устройство должно выполнить.

Вводные сведения в модели данных YANG формально заданы в контейнере onboarding-information модуля YANG, приведённого в параграфе 6.3. Дерево контейнера представлено ниже.

```
+--:(onboarding-information)
   +-- onboarding-information
      +-- boot-image
      |  +-- os-name?              string
      |  +-- os-version?           string
      |  +-- download-uri*         inet:uri
      |  +-- image-verification* [hash-algorithm]
      |     +-- hash-algorithm    identityref
      |     +-- hash-value        yang:hex-string
      +-- configuration-handling?      enumeration
      +-- pre-configuration-script?    script
      +-- configuration?               binary
      +-- post-configuration-script?   script
```

Вводные сведения должны быть доверенными, чтобы они были пригодны для устройства. Обработка устройством недоверенных вводных сведений не предусмотрена. Сведения считаются доверенными при получении через защищённое соединение с доверенным сервером начальной загрузки или при наличии подписи владельца устройства. В иных случаях вводные сведения считаются недоваренными.

Обработка вводных сведений описана в параграфе 5.6.

3. Артефакты

В этом документе определяются три артефакта, которые могут быть доступны устройствам при начальной загрузке. Каждый источник данных начальной загрузки указывает, как он представляет описанные в этом разделе артефакты (4. Источники данных для начальной загрузки).

3.1. Передаваемые сведения

Артефакт передаваемых сведений (conveyed information) содержит важные данные для начальной загрузки устройства. Этот артефакт служит для кодирования типов данных о перенаправлении и вводных сведений, описанных в разделе 2.

Артефакт имеет структуру криптографического сообщения (Cryptographic Message Syntax или CMS), как описано в [RFC5652], представленную с использованием правил ASN.1 DER[1], заданных в ITU-T X.690 [ITU.X690.2015]. Содержимое структуры CMS **должно** соответствовать модулю YANG, заданному в параграфе 6.3.

Структура CMS с передаваемой информацией может содержать подписанные или неподписанные данные начальной загрузки. Подписанные данные могут также шифроваться, но здесь они по-прежнему называются подписанными данными (см. параграф 1.2. Терминология).

Когда артефакт передаваемых сведений не подписан и не зашифрован, как это может быть при передаче по защищённым каналам, верхним типом содержимого структуры CMS **должен** быть один из идентификаторов OID, описанных в параграфе 10.3 (id-ct-sztpConveyedInfoXML или id-ct-sztpConveyedInfoJSON), или OID id-data (1.2.840.113549.1.7.1). При использовании OID id-data кодирование (JSON, XML и т. п.) **следует** передавать внешними средствами. В любом случае связанное содержимое является строкой октетов данных conveyed-information с ожидаемым кодированием.

[1]Distinguished encoding rules.

Когда артефакт передаваемой информации не подписан и не зашифрован, как это может быть при передаче по защищённому каналу, но оператор хочет убедиться, что содержимое может видеть лишь устройство, верхним типом содержимого структуры CMS **должен** быть OID id-envelopedData (1.2.840.113549.1.7.3). Кроме того, типом содержимого encryptedContentInfo **должен** быть один из OID, описанных в параграфе 10.3 (id-ct-sztpConveyedInfoXML или id-ct-sztpConveyedInfoJSON), или OID id-data (1.2.840.113549.1.7.1). При использовании OID id-data кодирование (JSON, XML и т. п.) **следует** передавать внешними средствами. В любом случае связанное содержимое является строкой октетов данных conveyed-information с ожидаемым кодированием.

Когда артефакт передаваемой информации подписан и не зашифрован, как это может быть при передаче по недоверенному каналу, верхним типом содержимого структуры CMS **должен** быть OID id-signedData (1.2.840.113549.1.7.2). Кроме того, внутренним eContentType **должен** быть один из OID, описанных в параграфе 10.3 (id-ct-sztpConveyedInfoXML или id-ct-sztpConveyedInfoJSON), или OID id-data (1.2.840.113549.1.7.1). При использовании OID id-data кодирование (JSON, XML и т. п.) **следует** передавать внешними средствами. В любом случае связанное содержимое является строкой октетов данных conveyed-information с ожидаемым кодированием.

Когда артефакт передаваемой информации подписан и зашифрован, как это может быть при передаче по недоверенному каналу с обеспечением конфиденциальности, верхним типом содержимого структуры CMS **должен** быть OID id-envelopedData (1.2.840.113549.1.7.3). Кроме того, типом encryptedContentInfo **должен** быть OID id-signedData (1.2.840.113549.1.7.2), для которого eContentType **должен** быть одним из OID, описанных в параграфе 10.3 (id-ct-sztpConveyedInfoXML или id-ct-sztpConveyedInfoJSON), или OID id-data (1.2.840.113549.1.7.1). При использовании OID id-data кодирование (JSON, XML и т. п.) **следует** передавать внешними средствами. В любом случае связанное содержимое является строкой октетов данных conveyed-information с ожидаемым кодированием.

3.2. Сертификат владельца

Артефакт сертификата владельца является сертификатом X.509 [RFC5280], применяемым для идентификации «владельца» (например, организации). Сертификат владельца может быть подписан удостоверяющим центром (certificate authority или CA). Сертификат владельца **должен** не задавать Key Usage или для Key Usage, как минимум, **должен** быть установлен бит digitalSignature. Этот документ не ограничивает значения subject и subjectAltName в сертификате владельца.

Сертификат владельца устройство использует для проверки подписи артефакта передаваемых сведений (3.1. Передаваемые сведения), который устройство должно получить, как указано в параграфе 3.5. В частности, устройство проверяет подпись с использованием открытого ключа в сертификате владельца для содержимого артефакта передаваемых сведений.

Артефакт сертификата владельца формально является структурой CMS, как задано в [RFC5652], представленной ASN.1 DER в соответствии с ITU-T X.690 [ITU.X690.2015].

Структура CMS сертификата владельца **должна** содержать сам сертификат, а также все промежуточные сертификаты, ведущие к закреплённому сертификату домена (pinned-domain-cert), указанному в ваучере владения. Артефакт сертификата владельца **может** включать и pinned-domain-cert.

Для поддержки устройств, развёрнутых в частных сетях, структура CMS сертификата владельца **может** включать достаточно свежие в соответствии с локальной политикой объекты отзыва (например, списки отзыва - Certificate Revocation List или CRL [RFC5280] и отклики OCSP [RFC6960]). Наличие объектов отзыва, прикреплённых к сертификату владельца, может избавить устройство от необходимости получать их динамически с использованием точки распространения CRL или ответчика протокола состояния сертификатов (Online Certificate Status Protocol или OCSP), указанного в связанных сертификатах.

В незашифрованном артефакте верхним типом содержимого структуры CMS сертификата владельца **должен** быть OID id-signedData (1.2.840.113549.1.7.2). Внутренняя структура SignedData является вырожденной и не подписывается, как при распространении сертификатов и объектов отзыва.

В зашифрованном артефакте верхним типом содержимого структуры CMS сертификата владельца **должен** быть OID id-envelopedData (1.2.840.113549.1.7.3), а типом содержимого encryptedContentInfo **должен** быть OID id-signedData (1.2.840.113549.1.7.2), в результате чего внутренняя SignedData является вырожденной и не подписывается, как при распространении сертификатов и объектов отзыва.

3.3. Ваучер владения

Артефакт ваучера владения служит для защищённой идентификации владельца устройства, известного изготовителю. Ваучер владения подписывает изготовитель устройства. Ваучер владения служит для проверки сертификата владельца (3.2. Сертификат владельца), который устройству также следует получить, как указано в параграфе 3.5. В частности, устройство проверяет наличие в сертификате владельца цепочки доверия, ведущей к доверенному сертификату, включённому в ваучер владения (pinned-domain-cert). Отметим, что эта связь сохраняется даже для самоподписанного сертификата владельца, являющегося также pinned-domain-cert.

Незашифрованный артефакт ваучера владения определён в [RFC8366]. Как указано, он является структурой CMS, в которой верхним типом содержимого **должен** быть OID id-signedData (1.2.840.113549.1.7.2), где eContentType **должен** быть OID id-ct-animaJSONVoucher (1.2.840.113549.1.9.16.1.40[1]) или OID id-data (1.2.840.113549.1.7.1). При использовании OID id-data кодирование (JSON, XML и т. п.) **следует** передавать внешними средствами. В любом случае связанное содержимое является строкой октетов данных ietf-voucher с ожидаемым кодированием.

В зашифрованном артефакте ваучера владения верхним типом содержимого **должен** быть OID id-envelopedData (1.2.840.113549.1.7.3), а типом содержимого encryptedContentInfo's **должен** быть OID id-signedData (1.2.840.113549.1.7.2), где eContentType **должен** быть OID id-ct-animaJSONVoucher (1.2.840.113549.1.9.16.1.40) или OID id-data (1.2.840.113549.1.7.1). При использовании OID id-data кодирование (JSON, XML и т. п.) **следует** передавать внешними средствами. В любом случае связанное содержимое является строкой октетов данных conveyed-information с ожидаемым кодированием.

[1]Этот идентификатор в данном и следующем абзаце был указан с ошибкой. См. https://www.rfc-editor.org/errata/eid6807. *Прим. перев.*

3.4. Шифрование артефактов

Каждый из трёх артефактов можно шифровать индивидуально. Шифрование может быть важным для сред, где содержимое считается конфиденциальным. Каждый из трёх артефактов шифруется одинаково - незашифрованная форма помещается внутрь типа CMS EnvelopedData.

В результате артефакты передаваемых сведений и ваучеров владения подписываются, затем шифруются, но не наоборот. Такая последовательность имеет несколько преимуществ - скрывается сертификат подписавшего и обеспечивается гарантия того, что владелец знает подписываемое содержимое. Эта последовательность также позволяет владельцу проверить незашифрованный ваучер, полученный от изготовителя, затем самостоятельно зашифровать его, возможно, вместе с текущими объектами отзыва, когда владелец готов поместить сертификат в незащищённое место.

При шифровании CMS **должен** применяться защищённый сертификат отождествления для устройства. Это может быть тот же сертификат, который применяется клиентом уровня TLS, используемым устройством при соединении с серверами начальной загрузки. Способ получения владельцем сертификата отождествления устройства для этой цели выходит за рамки этого документа.

3.5. Группировки артефактов

В предыдущих параграфах рассмотрены артефакты начальной загрузки, но лишь некоторые группировки этих артефактов имеет смысл возвращать в различных ситуациях начальной загрузки, описанных в этом документе.

Неподписанные данные
: Эта группа артефактов полезна, когда можно применить защиту транспортного уровня для доставки доверия (например, HTTPS) или передаваемая информация может быть обработана заранее (например, неподписанные сведения о перенаправлении).

Подписанные данные без отзывов
: Эта группа артефактов полезна, когда нужны подписанные данные (данные получены из недоверенного источника и не могут быть заранее обработаны), а отзывы не нужны или могут быть получены динамически.

Подписанные данные с отзывами
: Эта группа артефактов полезна, когда нужны подписанные данные (данные получены из недоверенного источника и не могут быть заранее обработаны), а отзывы нужны, но не могут быть получены динамически.

Наличие каждого артефакта и отличительные характеристики указаны для каждой группы в таблице (да и нет указывает наличие артефакта в группе).

Группировка	*Передаваемые сведения*	*Ваучер владения*	*Сертификат владельца*
Неподписанные данные	Да, без подписи	Нет	Нет
Подписанные данные без отзывов	Да, с подписью	Да, без отзывов	Да, без отзывов
Подписанные данные с отзывами	Да, с подписью	Да, с отзывами	Да, с отзывами

4. Источники данных для начальной загрузки

В этом разделе описаны несколько источников данных для начальной загрузки. Список не является исчерпывающим и может быть дополнен будущими документами. Для каждого источника указаны детали представления трёх артефактов, указанных в разделе 3.

4.1. Сменный носитель

Подключаемое напрямую съёмное устройство хранения (например, USB-носитель) **может** служить источником данных для начальной загрузки SZTP.

Отказаться от съёмных носителей сложно, поскольку они не требуют для работы внешней инфраструктуры. Необработанные (raw) загрузочные образы тоже можно размещать на съёмных носителях, позволяя им обеспечивать полное решение для автономной начальной загрузки.

Для использования съёмного носителя в качестве источника данных начальной загрузки устройству нужно лишь определить, что носитель подключён к устройству и примонтировать его файловую систему.

Съёмные устройства являются недоверенными источниками данных начальной загрузки. Это значит, что хранящиеся на них данные **должны** быть подписаны или **должны** содержать сведения, предназначенные для предварительной обработки (например, неподписанные сведения о перенаправлении).

С точки зрения артефактов съёмный носитель представляет себя как файловую систему и артефакты начальной загрузки должны быть представлены в виде файлов. Описанные в разделе 3 артефакты отображаются в файловую систему, как показано ниже.

Передаваемые сведения
: Отображаются в файл с двоичным артефактом, описанным в параграфе 3.1 (например, conveyed-information.cms).

Сертификат владельца
: Отображаются в файл с двоичным артефактом, описанным в параграфе 3.2 (например, owner-certificate.cms).

Ваучер владения
: Отображается в файл с двоичным артефактом, описанным в параграфе 3.3 (например, ownership-voucher.cms или ownership-voucher.vcj).

Формат файловой системы съёмного носителя и именование файлов выходят за рамки этого документа. Однако для улучшения совместимости устройствам **рекомендуется** поддерживать открытые и/или стандартизованные файловые системы. Устройствам также **рекомендуется** предполагать именование файлов, позволяющее создать не один экземпляр данных начальной загрузки (например, для разных устройств) на одном съёмном носителе. Именовать файлы **следует** уникально для изготовителя, чтобы на одном носителе можно было разместить данные начальной загрузки для устройств разных производителей.

4.2. Сервер DNS

Источником данных для начальной загрузки SZTP **может** служить сервер DNS. Это может быть привлекательным решением для развёртываний с имеющейся инфраструктурой DNS, поскольку может обеспечивать автоматическую (touchless) начальную загрузку, не требуя использовать сторонние ресурсы Internet.

DNS является недоверенным источником данных начальной загрузки. Даже при использовании DNSSEC [RFC6698] для аутентификации записей о ресурсах DNS (например, A, AAAA, CERT, TXT, TLSA) устройство не может быть уверенным, что возвращённый (например, сервером DHCP) домен, принадлежит законному владельцу. Это значит, что сведения из записей DNS **должны** быть подписаны (в соответствии с этим документом, а не DNSSEC) или **должны** быть данными для предварительной обработки (например, неподписанные сведения о перенаправлении).

4.2.1. Запросы DNS

Устройства, заявляющие поддержку DNS как источника данных начальной загрузки, **должны** сначала запрашивать записи DNS для устройства и лишь после неудачной загрузки **должны** запрашивать независимые от устройства записи DNS. Для каждого зависимого или независимого от устройства запроса устройство **должно** сначала передать multicast-запрос DNS [RFC6762], а затем, если не произошло успешной загрузки, **должен** передаваться unicast-запрос DNS [RFC1035] [RFC7766]. Это предполагает известность адреса сервера DNS, поэтому можно применять методы, подобные описанным в разделе 11 [RFC6763].

Для извлечения зависимых от устройства записей DNS устройство **должно** запрашивать записи TXT [RFC1035] из раздела <serial-number>._sztp, где <serial-number> - серийный номер устройства (то же значение, которое указано в сертификате отождествления защищённого устройства), _sztp - глобальный атрибут DNS, зарегистрированный этим документом (10.7. Реестр Underscored and Globally Scoped DNS Node Names). Примеры записей DNS для устройства приведены ниже.

```
TXT in <serial-number>._sztp.local.  (multicast)
TXT in <serial-number>._sztp.<domain>.  (unicast)
```

Для извлечения независимых от устройства записей DNS устройство **должно** запрашивать записи SRV [RFC2782] из раздела _sztp._tcp, где _sztp - имя службы, зарегистрированное этим документом (10.6. Реестр Service Name and Transport Protocol Port Number), а _tcp - глобальный атрибут DNS, зарегистрированный [RFC8552].

Отметим, что независимый от устройства отклик может содержать лишь неподписанные данные, поскольку для подписанных требуется использование ваучера владения для устройства. Применение записей SRV максимально привязано к имеющимся стандартам DNS. Отклик с несколькими записями SRV сравним с неподписанным списком сведений о перенаправлении на серверы начальной загрузки. Примеры зависимых от устройства записей DNS приведены ниже.

```
SRV in _sztp._tcp.local.  (multicast)
SRV in _sztp._tcp.<domain>.  (unicast)
```

4.2.2. Отклик DNS на зависимый от устройства запрос

Для зависимых от устройства запросов три артефакта начальной загрузки, определённых в разделе 3, кодируются в записи TXT, использующие пары ключ-значение, подобно методу, описанному в параграфе 6.3 [RFC6763].

Передаваемые сведения
: Отображаются в запись TXT с ключом ci и значением в виде двоичного артефакта, описанного в параграфе 3.1.

Сертификат владельца
: Отображаются в запись TXT с ключом oc и значением в виде двоичного артефакта, описанного в параграфе 3.2.

Ваучер владения
: Отображаются в запись TXT с ключом ov и значением в виде двоичного артефакта, описанного в параграфе 3.3.

Устройство **должно** игнорировать все прочие ключи, которые могут быть возвращены.

Отметим, что, несмотря на тип TXT, в записях **следует** (параграф 6.5 в [RFC6763]) кодировать двоичные данные.

Ниже приведён пример отклика для конкретного устройства с подписанными данными, как он может быть представлен пользовательским агентом. В примере предполагается, что серийный номер устройства - это <serial-number>, домен - example.com, а <binary data> - двоичное представление артефакта.

```
<serial-number>._sztp.example.com. 3600 IN TXT "ci=<binary data>"
<serial-number>._sztp.example.com. 3600 IN TXT "oc=<binary data>"
<serial-number>._sztp.example.com. 3600 IN TXT "ov=<binary data>"
```

Отметим, что в случае, когда ci представляет неподписанные данные, ключей oc и ov не будет в отклике.

4.2.3. Отклик DNS на независимый от устройства запрос

Для независимых от устройства запросов три артефакта начальной загрузки, определённых в разделе 3, кодируются в записи SRV.

Передаваемые сведения
: Этот артефакт не поддерживается напрямую и вместо этого суть сведений о перенаправлении отображается в записи SRV в соответствии с [RFC2782].

Сертификат владельца
: Не поддерживается. Независимые от устройства отклики не включают подписанных данных, поэтому не нужен артефакт сертификата владельца.

Ваучер владения
: Не поддерживается. Независимые от устройства отклики не включают подписанных данных, поэтому не нужен артефакт ваучера владения.

Ниже представлен пример независимого от устройства отклика, содержащего (фактически) неподписанные сведения о перенаправлении на 4 сервера начальной загрузки, как он может быть представлен пользовательским агентом. В примере предполагается домен example.com и наличие четырёх серверов начальной загрузки sztp[1-4].

```
_sztp._tcp.example.com. 1800 IN SRV 0 0 443 sztp1.example.com.
```

```
_sztp._tcp.example.com. 1800 IN SRV 1 0 443 sztp2.example.com.
_sztp._tcp.example.com. 1800 IN SRV 2 0 443 sztp3.example.com.
_sztp._tcp.example.com. 1800 IN SRV 2 0 443 sztp4.example.com.
```

Отметим, что в этом примере sztp3 и sztp4 имеют одинаковый приоритет, следовательно, эффективно представляют кластеризованную пару серверов начальной загрузки. Хотя у sztp1 и sztp2 лишь по одной записи SRV, может случиться так, что записи указывают на балансировщик, размещённый перед кластером серверов начальной загрузки.

Хотя в этом документе не применяется DNS-SD [RFC6763], в соответствии с параграфом 12.2 этого RFC, откликам Multicast DNS (mDNS) **следует** включать также все адресные записи (A и AAAA), указанные в SRV rdata.

4.2.4. Размер подписанных данных

Подписанные артефакты данных имеют большой размер в DNS. В варианте с наименьшим объёмом памяти каждый из них занимает несколько килобайт. Однако вводные сведения могут иметь больший размер. Все записи о ресурсах, включая TXT, ограничены размером 65535 байт, поскольку поле RDLENGTH имеет размер 16 битов (параграф 3.2.1 в [RFC1035]). Если желательно закодировать вводные сведения, превышающие это ограничение, в возвращаемых записях DNS следует вместо этого кодировать сведения о перенаправлении, указывающие устройству сервер начальной загрузки, с которого можно получить вводные сведения.

С учётом ожидаемого размера записей TXT неочевидно, что подписанные данные поместятся в пакет DNS протокола UDP даже при включённых механизмах расширения (Extension Mechanisms for DNS или EDNS(0)) [RFC6891]. В зависимости от содержимого подписанные данные могут не поместиться и в групповой (multicast) пакет DNS, который в соответствии с разделом 17 в [RFC6762] может достигать 9000 байтов. Поэтому для возврата подписанных данных предполагается потребность в доставке пакетов DNS по протоколу TCP [RFC7766].

4.3. Сервер DHCP

Сервер DHCP **может** служить источником данных начальной загрузки SZTP. Это может быть привлекательно для развёртываний с инфраструктурой DHCP, поскольку обеспечивает возможность автоматической (touchless) начальной загрузки без применения ресурсов Internet, поддерживаемых сторонними организациями.

Сервер DHCP является недоверенным источником данных начальной загрузки, поэтому хранящиеся там данные **должны** быть подписаны или это **должна** быть информация, которая может быть обработана предварительно (например, неподписанные сведения о перенаправлении).

Однако, в отличие от других источников данных начальной загрузки, описанных в этом документе, протокол DHCP (особенно DHCP для IPv4) сильно ограничен по размеру передаваемой информации и подписанные данные передать не удастся. Это означает, что через DHCP можно получить лишь неподписанные сведения о перенаправлении. Поскольку эти данные не подписаны, в них **не следует** включать необязательный сертификат привязки доверия, так как он занимает место в сообщении DHCP, а устройству все равно придётся его отбросить. По этой причине определённые в разделе 8 опции DHCP не позволяют кодировать сертификат привязки доверия.

Ниже приведено отображение артефактов, заданных в разделе 3, на поля DHCP, указанные в разделе 8.

Передаваемые сведения
: Этот артефакт не поддерживается напрямую и взамен суть сведений о перенаправлении отображается в опции DHCP, описанные в разделе 8.

Сертификат владельца
: Не поддерживается, поскольку в пакете DHCP недостаточно места для артефакта сертификата владельца.

Ваучер владения
: Не поддерживается, поскольку в пакете DHCP недостаточно места для артефакта ваучера владения.

4.4. Сервер начальной загрузки

Сервер начальной загрузки **может** служить источником данных для начальной загрузки SZTP. Сервер начальной загрузки определён как сервер RESTCONF [RFC8040], реализующий модуль YANG из раздела 7.

Применение сервера начальной загрузки в качестве источника данных для начальной загрузки привлекательно, поскольку он **может** использовать защиту транспортного уровня, исключая необходимость подписывать данные, что в некоторых случаях может быть проще в развёртывании.

В отличие от других источников данных начальной загрузки, описанных в этом документе, сервер начальной загрузки является не только источником данных, но и может получать данные от устройств, используя вызов YANG RPC report-progress из модуля YANG, определённого в параграфе 7.3. RPC report-progress обеспечивает видимость процесса загрузки (например, предупреждения и ошибки) и предоставляет потенциально полезные сведения по завершении (например ключи SSH для хоста и/или сертификаты привязок доверия TLS).

Сервер начальной загрузки может быть доверенным или недоверенным источником данных в зависимости от получения устройством привязки доверия для сервера из доверенного источника. Если сервер начальной загрузки является доверенным, передаваемые им сведения **могут** быть подписаны. При недоверенном сервере начальной загрузки передаваемые сведения **должны** быть подписаны или **должны** быть информацией, которая может предварительно обрабатываться (например, неподписанные сведения о перенаправлении).

Поскольку сервер начальной загрузки предоставляет данные, соответствующие модели YANG, артефакты должны сопоставляться с узлами YANG. Три артефакта, заданные в разделе 3, отображаются на узлы output в RPC get-bootstrapping-data из модуля, определённого в параграфе 7.3.

Передаваемые сведения
: Сопоставляется с листом conveyed-information в выводе RPC get-bootstrapping-data.

Сертификат владельца
: Сопоставляется с листом owner-certificate в выводе RPC get-bootstrapping-data.

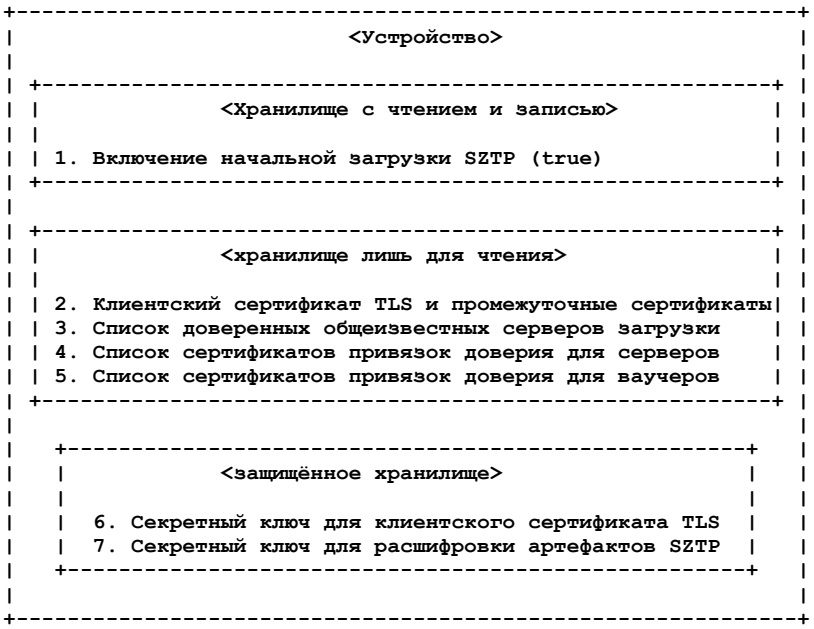
Ваучер владения
: Сопоставляется с листом ownership-voucher в выводе RPC get-bootstrapping-data.

Серверы SZTP имеют лишь две конечных точки - RPC get-bootstrapping-data и один из вызовов RPC report-progress. Эти RPC используют аутентифицированное имя пользователя RESTCONF для изоляции выполнения RPC от других устройств.

5. Детали устройства

Поддерживающие описанную в этом документе начальную загрузку устройства **должны** иметь настроенное заранее состояние и логику загрузки, описанные в последующих параграфах.

5.1. Исходное состояние



Нумерация в приведённом ниже списке соответствует нумерации на рисунке.

1. Устройства **должны** иметь настраиваемую переменную для включения и отключения начальной загрузки SZTP. Эта переменная по умолчанию **должна** разрешать начальную загрузку SZTP при первом включении питания устройства. Поскольку конфигурация, задаваемая процессом начальной загрузки, отключает начальную загрузку SZTP и эта конфигурация может быть объединена с имеющейся, **не рекомендуется** использовать узел конфигурации, зависящий от присутствия, поскольку его нельзя удалить при слиянии.

2. Устройствам, поддерживающим загрузку данных начальной загрузки с bootstrap-серверов (4.4. Сервер начальной загрузки), **следует** иметь клиентский сертификат TLS и все промежуточные сертификаты, ведущие к общеизвестной привязке доверия для сертификатов. Сертификатом общеизвестной привязки доверия может быть промежуточный сертификат или самоподписанный корневой сертификат. Для поддержки устройств без клиентского сертификата устройства **могут** идентифицировать и аутентифицировать себя серверу начальной загрузки с помощью схемы аутентификации HTTP в соответствии с параграфом 2.5 в [RFC8040], однако этот документ не задаёт механизма ввода данных оператором (например, пароля).

3. Устройства, поддерживающие загрузку начальных данных с общеизвестных bootstrap-серверов, **должны** иметь список таких серверов. В соответствии со сведениями о перенаправлении (параграф 2.1) каждый сервер начальной загрузки может быть указан именем или IP-адресом хоста, а также необязательным номером порта.

4. Устройства, поддерживающие загрузку начальных данных с общеизвестных bootstrap-серверов, **должны** иметь список сертификатов привязок доверия, которые могут служить для проверки подлинности общеизвестных серверов начальной загрузки. Для каждого сертификата привязки доверия, не являющегося самоподписанным корневым сертификатом, устройству **следует** иметь также цепочку промежуточных сертификатов, ведущих к самоподписанному корневому сертификату, включая его.

5. Устройства, поддерживающие загрузку подписанных данных (1.2. Терминология), **должны** иметь сертификаты привязок доверия для проверки ваучеров владения. Для каждого сертификата привязки доверия, не являющегося самоподписанным корневым сертификатом, устройству **следует** иметь также цепочку промежуточных сертификатов, ведущих к самоподписанному корневому сертификату, включая его.

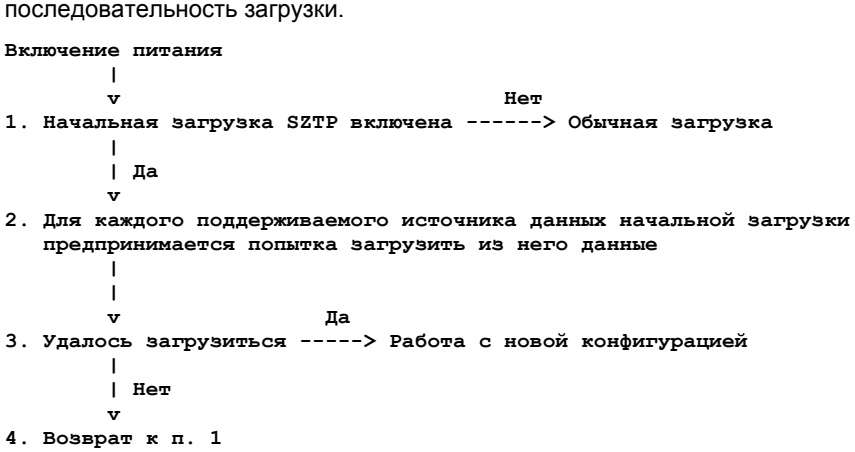
6. Устройства, поддерживающие клиентский сертификат уровня TLS для идентификации и аутентификации себя на сервере начальной загрузки, **должны** иметь секретный ключ, соответствующий открытому ключу в сертификате TLS. Этот ключ **следует** держать в защищённом хранилище, в идеале на криптографическом устройстве, таком как микросхема модуля доверенной платформы (trusted platform module или TPM).

7. Устройства, поддерживающие расшифровку артефактов SZTP, **должны** иметь секретный ключ, соответствующий открытому ключу для шифрования артефактов. Этот ключ **следует** держать в защищённом хранилище, в идеале на криптографическом устройстве, таком как микросхема модуля доверенной платформы (TPM). Это **может** быть секретный ключ, связанный с клиентским сертификатом TLS, используемым для соединения с серверами начальной загрузки.

Модуль YANG, представляющий эти данные, приведён в Приложении A.

5.2. Последовательность загрузки

Устройство, заявляющее поддержку описанной в документе стратегии, **должно** обеспечивать указанную ниже последовательность загрузки.



Примечание. В любой момент устройство можно настроить с помощью иного механизма, например через командный интерфейс (command-line interface или CLI).

Нумерация в приведённом ниже списке соответствует нумерации на диаграмме процесса.

1. При включении устройства оно сначала проверяет, включена ли начальная загрузка SZTP, как ожидается для заданного исходного состояния. При отключённой начальной загрузке SZTP устройство запускается как обычно.
2. Устройство выполняет итерацию по списку источников данных для начальной загрузки (раздел 4), как описано в параграфе 5.3.
3. Если устройство смогло загрузиться из источника данных начальной загрузки, оно работает с новой загруженной конфигурацией.
4. В противном случае устройство **должно** вернуться к следующему элементу списка источников загрузки.

Этот документ не запрещает одновременное использование альтернативных механизмов обеспечения, которые могут включать, например, CLI или web-интерфейс для пользователя и даже другой протокол начальной загрузки. Независимо от способа настройки в конфигурации **следует** сбросить флаг начальной загрузки SZTP, как указано в параграфе 5.1.

5.3. Работа с источниками данных для начальной загрузки

В этом параграфе рассматривается рекурсивный алгоритм, который устройства могут использовать для получения вводных сведений. Алгоритм является рекурсивным, поскольку источники могут возвращать сведения о перенаправлении, которые ведут к запуску алгоритма для другого источника данных начальной загрузки. Выбор источника считается успешным, когда получены вводные сведения, которым может предшествовать одно или несколько перенаправлений.

Важным аспектом алгоритма является понимание, когда данные должны быть подписаны.

Тип данных начальной загрузки	*От недоверенного источника*	*От доверенного источника*
Неподписанные сведения о перенаправлении	Да[1]	Да
Подписанные сведения о перенаправлении	Да	Да[2]
Неподписанные вводные сведения	Нет	Да
Подписанные вводные сведения	Да	Да[2]

В рекурсивном алгоритме применяется глобальная переменная trust-state с исходным значением FALSE. Конечной целью алгоритма является обработка вводных сведений (параграф 2.2) пока не будет достигнуто trust-state = TRUE.

Если источником данных для начальной загрузки (раздел 4) является bootstrap-сервер (параграф 4.4) и устройство способно аутентифицировать себя на сервере с использованием проверки пути сертификации X.509 (раздел 6 в [RFC6125]) к одной из заданных на устройстве или узнанных на предыдущем этапе привязок доверия, устройство **должно** установить trust-state = TRUE.

При организации соединения с доверенным или недоверенным сервером начальной загрузки устройство **должно** идентифицировать и аутентифицировать себя на сервере с использованием сертификата клиента уровня TLS и/или схемы аутентификации HTTP (параграф 2.5 в [RFC8040]). При использовании обоих механизмов они **должны** указывать один серийный номер. При отправке сертификата клиента устройство **должно** также передать все промежуточные сертификаты, ведущие к сертификату общеизвестной привязки доверия для сертификата клиента, возможно включая и её.

Для любого источника данных начальной загрузки (4. Источники данных для начальной загрузки) при получении зашифрованного артефакта устройство **должно** сначала расшифровать его, используя секретный ключ, связанный с сертификатом устройства, применяемым для шифрования артефакта.

Если артефакт передаваемых сведений подписан и устройство способно проверить подписанные данные по алгоритму из параграфа 5.4, оно **должно** установить trust-state = TRUE, при неспособности проверить подписанные данные устройство **должно** установить trust-state = FALSE. Отметим, что это относится к особому случаю, когда подписанные данные возвращаются даже из доверенного источника данных начальной загрузки.

[1]Источник, на который указывает перенаправление, **должен** вернуть подписанные данные или другое перенаправление без подписи.
[2]При обработке обычно не требуется, чтобы доверенный источник возвращал подписанные данные.

Если артефакт передаваемых сведений содержит данные о перенаправлении, устройство **должно** обработать сведения о перенаправлении в соответствии с параграфом 5.5, учитывая разрешённую глубину рекурсии. Реализации **должны** ограничивать глубину рекурсии и **следует** разрешать не более 10 рекурсивных перенаправлений. Рекурсия ведёт к повторению алгоритма, но на этот раз источником данных определённо будет сервер начальной загрузки, поскольку сведения о перенаправлении могут отправлять устройство лишь на bootstrap-серверы.

Если артефакт передаваемых сведений содержит вводные данные и trust-state = FALSE, устройство **должно** выйти из рекурсии (см. диаграмму процесса выше), вернувшись к последовательности, как описано в параграфе 5.2. В иных случаях устройство **должно** попытаться обработать вводные сведения, как описано в параграфе 5.6. Независимо от результата обработки вводных сведений, устройство **должно** выйти из рекурсии, возвращаясь к последовательности начальной загрузки, как описано в параграфе 5.2. Единственным отличием является ответ на вопрос о возможности начальной загрузки из какого-либо источника, показанный на диаграмме выше.

5.4. Проверка подписанных данных

При получении устройством подписанных данных оно должно проверить их, как описано здесь, включая случаи предоставления подписанных данных доверенным источником. Устройству вместе с подписанными данными должен предоставляться ваучер владения и сертификат владельца, как описано для разных источников данных начальной загрузки в разделе 4. Источники данных для начальной загрузки.

Для проверки подписанных данных устройство сначала **должно** убедиться в подлинности ваучера владения путём проверки его подписи через одну из настроенных заранее привязок доверия (5.1. Исходное состояние), что может повлечь использование промежуточных сертификатов, присоединённых к ваучеру владения. Если устройство имеет точные часы, оно **должно** убедиться, что ваучер владения создан в прошлом (created-on раньше текущего времени), а при наличии листа expires-on устройство должно проверить, что срок действия ваучера не истёк (expires-on после текущего времени). Устройство **должно** убедиться в приемлемости значения assertion (некоторые устройства воспринимают лишь значение verified). Устройство **должно** проверить свой серийный номер в листе serial-number. При наличии листа idevid-issuer устройство **должно** проверить корректность его значения. При успешной аутентификации ваучера владения устройство извлекает узел pinned-domain-cert - сертификат X.509, требуемый для проверки сертификата владельца на следующем этапе.

Затем устройство **должно** проверить подлинность сертификата владельца путём проверки пути сертификации X.509 к доверенному сертификату, извлечённому из узла pinned-domain-cert в ваучере владения. Эта проверка может повлечь использование дополнительных промежуточных сертификатов, присоединенных к артефакту сертификата владельца. Если узел domain-cert-revocation-checks в ваучере владения имеет значение true, устройство **должно** проверить статус отзыва цепочки сертификатов, использованной для подписи сертификата владельца - при недоступности свежего статуса отзыва или обнаружении отзыва устройству **недопустимо** считать сертификат владельца действительным.

В заключение устройство **должно** убедиться, что артефакт передаваемых сведений подписан с проверенным сертификатом владельца.

При отказе на любом из указанных выше шагов устройство **должно** счесть подписанные данные недействительными и отказаться от последующих шагов.

5.5. Обработка сведений о перенаправлении

Для обработки сведений о перенаправлении (параграф 2.1) устройство **должно** выполнять представленные в этом параграфе действия. Обработка проста - устройство последовательно выполняет действия по списку предложенных bootstrap-серверов, пока не найдёт тот, с которого сможет загрузиться.

Если сервер представлен именем хоста и распознавание DNS даёт не один адрес IP, устройство **должно** предпринять хотя бы одну попытку соединения с каждым адресом, прежде чем перейти к следующему. Если устройство способно получить данные начальной загрузки с любого из полученных от DNS адресов, оно **должно** сразу же обработать эти данные без попытки соединения с другими адресами, полученными от DNS.

Если сведения о перенаправлении являются доверенными (например, trust-state = TRUE) и запись для bootstrap-сервера содержит сертификат привязки доверия, устройство **должно** аутентифицировать сертификат TLS у сервера, используя проверку пути сертификации X.509 (раздел 6 в [RFC6125]) к заданной привязке доверия. Если запись для bootstrap-сервера не включает сертификат привязки доверия, устройство **должно** организовать условное (provisional) соединение с сервером начальной загрузки (вслепую принять сертификат сервера) и установить trust-state = FALSE.

Если сведения о перенаправлении являются недоверенными (например, trust-state = FALSE), устройство **должно** отбросить все привязки доверия, представленные в сведениях о перенаправлении, и организовать условное (provisional) соединение с сервером начальной загрузки (вслепую принять серверный сертификат TLS).

5.6. Обработка вводных сведений

Для обработки вводных сведений (параграф 2.2) устройство **должно** выполнить представленные в этом параграфе действия. Сначала устройство **должно** обработать загрузочный образ (при наличии), затем сценарий предварительной настройки (при наличии). После этого задаётся исходная конфигурация (при наличии) и выполняются сценарии после установки конфигурации (при наличии).

При получении вводных данных от доверенного сервера начальной загрузки устройство **должно** передать отчёт об исполнении bootstrap-initiated и завершающий отчёт boot-image-installed-rebooting, bootstrap-complete или отчёт об ошибке. Если узел reporting-level в RPC-отклике get-bootstrapping-data от сервера начальной загрузки имеет значение verbose, устройство дополнительно **должно** передать все промежуточные (non-terminating) отчёты (например, инициирование, предупреждения, завершение и пр.). Независимо от уровня отчётности, запрошенного сервером начальной загрузки, устройство **может** передавать дополнительные отчёты о ходе выполнения.

При получении вводных данных от недоверенного сервера начальной загрузки устройству **недопустимо** передавать серверу какие-либо отчёты о ходе работ, даже когда вводные сведения подписаны и аутентифицированы. Следует

помнить, что серверам начальной загрузки рекомендуется переводить недоверенные соединения в доверенные (последний абзац параграфа 9.6), чтобы иметь возможность собирать отчёты от устройств.

При возникновении ошибки на любом этапе устройство **должно** остановить последовательность начальной загрузки, описанную в параграфе 5.2. В контексте рекурсивного алгоритма устройство **должно** вернуться на предыдущий уровень, а не в самое начало. В процесс начальной загрузки **может** сохраниться то или иное состояние (например, обновленный загрузочный образ, log-файлы, остатки сценариев и т. п.). Однако сохранённое состояние **недопустимо** активировать каким-либо способом (например, новая конфигурация или запуск программы) и **недопустимо** препятствовать продолжению устройством последовательности начальной загрузки (т. е. обработки вводных сведений от другого bootstrap-сервера).

Далее описывается упорядоченная последовательность действий, которые устройство **должно** выполнить.

Если вводные сведения получены от доверенного сервера начальной загрузки, устройство **должно** передать отчет bootstrap-initiated. Отсутствие возврата строки состояния HTTP "204 No Content" является ошибкой и при её возникновении устройство **должно** попытаться отправить отчёт bootstrap-error перед выходом. Устройство **должно** проанализировать предоставленный документ с вводными сведениями для извлечения значений, применяемых на следующих этапах. Независимо от использования потокового синтаксического анализатора при возникновении ошибки в процессе анализа в случае подключения к доверенному серверу начальной загрузки устройство **должно** попытаться передать отчёт parsing-error перед выходом.

Если заданы критерии для загрузочного образа, устройство **должно** сначала проверить, соответствует ли им запущенный образ загрузки. Если на устройстве уже запущен указанный загрузочный образ, оставшаяся часть этого этапа пропускается. Если указанный образ ещё не запущен, устройство **должно** его загрузить, проверить и установить (в указанном порядке) и перезагрузиться. При подключении к доверенному серверу начальной загрузки устройство **может** попытаться отправить отчёт boot-image-mismatch. Для загрузки (download) образа устройство **должно** применять лишь URI, указанные вводными данными. Для проверки загрузочного образа устройство **должно** использовать проверочные отпечатки, представленные вводными данными, или криптографическую подпись, встроенную в сам образ загрузки, используя механизм, не задаваемый данным документом. До перезагрузки (при подключении к доверенному bootstrap-серверу) устройство должно попытаться передать отчёт boot-image-installed-rebooting. После перезапуска процесс начальной загрузки запускается снова, что в конечном итоге приведёт опять к этому шагу, но затем устройство запустит указанный образ загрузки и перейдёт к следующему шагу. Если при подключении к доверенному bootstrap-серверу происходит ошибка на любом этапе (до перезагрузки), устройство **должно** попытаться передать отчёт boot-image-error перед выходом.

Если задан сценарий предварительной настройки, устройство **должно** выполнить его, фиксируя весь вывод, а затем проверить наличие ошибок и предупреждений. Если возникает ошибка при соединении с доверенным сервером начальной загрузки, устройство **должно** попытаться передать отчёт pre-script-error перед выходом.

Если задана исходная конфигурация, устройство **должно** неделимо (atomically) представить её устройству, используя подход, заданный листом configuration-handling. Если возникает ошибка при соединении с доверенным сервером начальной загрузки, устройство **должно** попытаться передать отчёт config-error перед выходом.

Если задан сценарий для выполнения после настройки конфигурации, устройство **должно** выполнить его, фиксируя весь вывод, а затем проверить наличие ошибок и предупреждений. Если возникает ошибка при соединении с доверенным сервером начальной загрузки, устройство **должно** попытаться передать отчёт post-script-error перед выходом.

Если вводные данные были получены от доверенного bootstrap-сервера и в результате начальной загрузки не был сброшен флаг начальной загрузки SZTP, описанный в параграфе 5.1. Исходное состояние, устройству **следует** передать отчёт bootstrap-warning.

Если вводные данные были получены от доверенного bootstrap-сервера, устройство **должно** передать отчёт bootstrap-complete. Отсутствие возврата строки состояния HTTP "204 No Content" является ошибкой и при её возникновении устройство **должно** попытаться отправить отчёт bootstrap-error перед выходом.

На этом процесс обработки данных начальной загрузки завершается. Устройство работает с исходной конфигурацией. Если настроен звонок NETCONF Call Home или RESTCONF Call Home [RFC8071], устройство пытается связаться по телефону с указанным номером.

Примечание для разработчиков. Реализации могут отличаться способом предотвращения сохранения нежелательных состояний при ошибках в процесс загрузки. Если реализация возвращается (undo) к предыдущему шагу, применимы указанные ниже рекомендации.

- При возникновении ошибки устройство может отказаться от текущего и любых предшествующих шагов.
- Большинство шагов неделимы, например, обработка конфигурации (см. выше), обработка сценариев (как задано в модуле YANG ietf-sztp-conveyed-info).
- В случае ошибки после установки начальной конфигурации устройство должно восстановить прежнюю конфигурацию.
- В случае ошибки после успешного выполнения сценария для реализации может быть полезно иметь сценарий, способный принимать на входе параметр, указывающий необходимость удалить созданное предыдущим сценарием состояние.

6. Модель данных для передаваемых сведений

В этом разделе задан модуль YANG 1.1 [RFC7950], служащий для определения модели данных для артефакта передаваемых сведений, описанного в параграфе 3.1. Модель использует оператор расширения yang-data, заданный в [RFC8040]. Примеры, иллюстрирующие модель данных, представлены в параграфе 6.2.

6.1. Обзор модели данных

Ниже представлено дерево, иллюстрирующее модель данных для артефакта передаваемых сведений.

```
module: ietf-sztp-conveyed-info

  yang-data conveyed-information:
    +-- (information-type)
       +--:(redirect-information)
       |  +-- redirect-information
       |     +-- bootstrap-server* [address]
       |        +-- address         inet:host
       |        +-- port?           inet:port-number
       |        +-- trust-anchor?   cms
       +--:(onboarding-information)
          +-- onboarding-information
             +-- boot-image
             |  +-- os-name?              string
             |  +-- os-version?           string
             |  +-- download-uri*         inet:uri
             |  +-- image-verification* [hash-algorithm]
             |     +-- hash-algorithm    identityref
             |     +-- hash-value        yang:hex-string
             +-- configuration-handling?      enumeration
             +-- pre-configuration-script?    script
             +-- configuration?               binary
             +-- post-configuration-script?   script
```

6.2. Пример использования

Приведённый ниже пример иллюстрирует кодирование сведений о перенаправлении (параграф 2.1) с использованием JSON [RFC8259].

```
{
  "ietf-sztp-conveyed-info:redirect-information" : {
    "bootstrap-server" : [
      {
        "address" : "sztp1.example.com",
        "port" : 8443,
        "trust-anchor" : "base64encodedvalue=="
      },
      {
        "address" : "sztp2.example.com",
        "port" : 8443,
        "trust-anchor" : "base64encodedvalue=="
      },
      {
        "address" : "sztp3.example.com",
        "port" : 8443,
        "trust-anchor" : "base64encodedvalue=="
      }
    ]
  }
}
```

Следующий пример показывает кодирование вводных данных (параграф 2.2) с использованием JSON [RFC8259][1].

```
{
  "ietf-sztp-conveyed-info:onboarding-information" : {
    "boot-image" : {
      "os-name" : "VendorOS",
      "os-version" : "17.2R1.6",
      "download-uri" : [ "https://example.com/path/to/image/file" ],
      "image-verification" : [
        {
          "hash-algorithm" : "ietf-sztp-conveyed-info:sha-256",
          "hash-value" : "ba:ec:cf:a5:67:82:b4:10:77:c6:67:a6:22:ab:\
7d:50:04:a7:8b:8f:0e:db:02:8b:f4:75:55:fb:c1:13:b2:33"
        }
      ]
    },
    "configuration-handling" : "merge",
    "pre-configuration-script" : "base64encodedvalue==",
    "configuration" : "base64encodedvalue==",
    "post-configuration-script" : "base64encodedvalue=="
  }
}
```

6.3. Модуль YANG

В этом параграфе представлен модуль YANG для передаваемых сведений. В этом модуле применяются типы данных из [RFC5280], [RFC5652], [RFC6234] и [RFC6991], оператор расширения из [RFC8040] и кодирование, определённое в [ITU.X690.2015].

```
<CODE BEGINS> file "ietf-sztp-conveyed-info@2019-04-30.yang"
module ietf-sztp-conveyed-info {
  yang-version 1.1;
  namespace "urn:ietf:params:xml:ns:yang:ietf-sztp-conveyed-info";
```

[1] Здесь и далее символ \ в конце строки служит для переноса слишком длинных строк.

```
  prefix sztp-info;

  import ietf-yang-types {
    prefix yang;
    reference
      "RFC 6991: Common YANG Data Types";
  }
  import ietf-inet-types {
    prefix inet;
    reference
      "RFC 6991: Common YANG Data Types";
  }
  import ietf-restconf {
    prefix rc;
    reference
      "RFC 8040: RESTCONF Protocol";
  }

  organization
    "IETF NETCONF (Network Configuration) Working Group";
  contact
    "WG Web:   <https://datatracker.ietf.org/wg/netconf/>
     WG List:  <mailto:netconf@ietf.org>
     Author:   Kent Watsen <mailto:kent+ietf@watsen.net>";
  description
    "Этот модуль задаёт модель данных для артефакта передаваемых
     сведений, определённого в RFC 8572 (Secure Zero Touch
     Provisioning (SZTP)).

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

     Авторские права (Copyright (c) 2019) принадлежат IETF Trust
     и лицам, указанным как авторы кода. Все права защищены.

     Распространение и использование модуля в исходном или двоичном
     формате, с изменениями или без таковых разрешено в соответствии
     с условиями лицензии Simplified BSD License, изложенными в
     разделе 4.c документа IETF Trust's Legal Provisions применительно
     к документам IETF (http://trustee.ietf.org/license-info).

     Эта версия модуля YANG является частью RFC 8572, где правовые
     аспекты изложены более полно.";

  revision 2019-04-30 {
    description
      "Исходная версия";
    reference
      "RFC 8572: Secure Zero Touch Provisioning (SZTP)";
  }

  // Отождествления
  identity hash-algorithm {
    description
      "Базовое отождествление для проверки алгоритма хэширования.";
  }

  identity sha-256 {
    base hash-algorithm;
    description
      "Алгоритм SHA-256.";
    reference
      "RFC 6234: US Secure Hash Algorithms";
  }

  // Определения типов
  typedef cms {
    type binary;
    description
      "Структура ContentInfo, заданная в RFC 5652, кодируется
       с применением правил ASN.1 (DER), заданных в ITU-T X.690.";
    reference
      "RFC 5652:
         Cryptographic Message Syntax (CMS)

       ITU-T X.690:
         Information technology - ASN.1 encoding rules:
         Specification of Basic Encoding Rules (BER),
         Canonical Encoding Rules (CER) and Distinguished
         Encoding Rules (DER)";
  }

  // yang-data
```

```
    rc:yang-data conveyed-information {
      choice information-type {
        mandatory true;
        description
          "Этот выбор обеспечивает наличие в отклике
           redirect-information или onboarding-information.";
        container redirect-information {
          description
            "Сведения о перенаправлении в соответствии с параграфом 2.1
             в RFC 8572. Предназначены для перенаправления устройства
             на другой сервер начальной загрузки.";
          reference
            "RFC 8572: Secure Zero Touch Provisioning (SZTP)";
          list bootstrap-server {
            key "address";
            min-elements 1;
            description
              "Сервер начальной загрузки (bootstrap).";
            leaf address {
              type inet:host;
              mandatory true;
              description
                "Адрес IP или имя хоста bootstrap-сервера, на который
                 следует перенаправить устройство.";
            }
            leaf port {
              type inet:port-number;
              default "443";
              description
                "Номер порта на bootstrap-сервере. По умолчанию
                 применяется выделенный IANA порт https (443).";
            }
            leaf trust-anchor {
              type cms;
              description
                "Структура CMS, которая ДОЛЖНА содержать цепочку
                 сертификатов X.509, требуемых для аутентификации
                 сертификата TLS, представленного bootstrap-сервером.

                 В CMS ДОЛЖНА содержаться лишь 1 цепочка сертификатов.
                 Bootstrap-сервер ДОЛЖЕН аутентифицировать лишь
                 последний промежуточный сертификат CA в цепочке.

                 Во всех случаях цепочка ДОЛЖНА включать самоподписанный
                 корневой сертификат. Если корневой сертификат связан с
                 эмитентом сертификата TLS bootstrap-сервера, в цепочке
                 будет присутствовать лишь этот сертификат.

                 Если устройству нужно, структура CMS МОЖЕТ также
                 включать подходящие свежие объекты отзыва, по которым
                 устройство может проверить статус отзыва сертификатов.

                 CMS кодируется вырожденной формой структуры SignedData,
                 обычно применяемой для распространения сертификатов
                 X.509 и объектов отзыва (RFC 5280).";
              reference
                "RFC 5280:
                   Internet X.509 Public Key Infrastructure Certificate
                   and Certificate Revocation List (CRL) Profile";
            }
          }
        }
        container onboarding-information {
          description
            "Вводные сведения, описанные в параграфе 2.2 RFC 8572.
             Предоставляют устройству всё, что нужно для начальной
             загрузки.";
          reference
            "RFC 8572: Secure Zero Touch Provisioning (SZTP)";
          container boot-image {
            description
              "Критерии для загрузочного образа, который ДОЛЖЕН
               запускаться на устройстве, а также сведения,
               позволяющие устройству установить требуемый образ.";
            leaf os-name {
              type string;
              description
                "Название операционной системы, которая ДОЛЖНА быть
                 загружена на устройстве, чтобы не требовалось обновлять
                 программный образ (например, VendorOS).";
            }
            leaf os-version {
              type string;
              description
                "Версия операционной системы, которая ДОЛЖНА быть
                 загружена на устройстве, чтобы не требовалось обновлять
```

```
            программный образ (например, 17.3R2.1).";
        }
        leaf-list download-uri {
          type inet:uri;
          ordered-by user;
          description
            "Упорядоченный список URI, откуда можно получить один и
             тот же файл образа загрузки. Поддерживаемые устройством
             схемы URI (http, ftp, и пр.) зависят от производителя.
             Если применяется защищённая схема (например, https),
             устройство МОЖЕТ организовать недоверенное соединение с
             удаленным сервером, вслепую принимая его сертификат для
             получения загрузочного образа.";
        }
        list image-verification {
          must '../download-uri' {
            description
              "URI для загрузки должны предоставляться, если
               требуется проверка образа.";
          }
          key "hash-algorithm";
          description
            "Список хэш-значений, которые устройство может
             использовать для проверки файла образа загрузки.";
          leaf hash-algorithm {
            type identityref {
              base hash-algorithm;
            }
            description
              "Указывает используемый алгоритм хэширования.";
          }
          leaf hash-value {
            type yang:hex-string;
            mandatory true;
            description
              "Шестнадцатеричное значение заданного алгоритма
               хэширования, применённого к содержимому образа.";
          }
        }
      }
      leaf configuration-handling {
        type enumeration {
          enum merge {
            description
              "Слияние конфигурации с рабочим хранилищем.";
          }
          enum replace {
            description
              "Замена имеющегося содержимого рабочего хранилища
               переданной конфигурацией.";
          }
        }
        must '../configuration';
        description
          "Указывает, как серверу следует обрабатывать
           представленную конфигурацию.";
      }
      leaf pre-configuration-script {
        type script;
        description
          "Сценарий, который (при наличии) выполняется перед
           обработкой конфигурации.";
      }
      leaf configuration {
        type binary;
        must '../configuration-handling';
        description
          "Любая конфигурация, известная устройству. Применение типа
           binary позволяет встраивать содержимое (например, XML) в
           документ JSON. Кодирование содержимого, как и для
           сценария, зависит от производителя.";
      }
      leaf post-configuration-script {
        type script;
        description
          "Сценарий, который (при наличии) выполняется после
           обработки конфигурации.";
      }
    }
  }
}

typedef script {
  type binary;
  description
    "Сценарий для конкретного устройства, позволяющий выполнять
```

```
        команды для действий, недоступных через настройку конфигурации.

        Не предпринимается попыток стандартизации содержимого,
        контекста работы, языка программирования сценариев, кроме
        возможности указания предупреждений и ошибок, а также выдачи
        вывода. Содержимое сценария считается зависящим от
        производителя, линейки продукции и/или модели устройства.

        Если при выполнении сценария выдавались предупреждения,
        устройство ДОЛЖНО считать, что в сценарии имеются «мягкие»
        ошибки, не влияющие на управляемость по мнению сценария.

        Если при выполнении сценария возникли ошибки, устройство
        ДОЛЖНО считать, что в сценарии имеются серьёзные ошибки,
        влияющие на управляемость. В этом случае сценарий должен
        аккуратно завершить работу, удалив все состояния, которые
        могут мешать устройству продолжить последовательность
        начальной загрузки (например, обработку вводных сведений
        от другого сервера начальной загрузки).";
    }
  }
  <CODE ENDS>
```

7. API сервера начальной загрузки SZTP

В этом разделе определяется интерфейс API для серверов начальной загрузки. API определяется как созданный сервером RESTCONF [RFC8040], который поддерживает определённый в этом разделе модуль YANG 1.1 [RFC7950].

7.1. Обзор API

Ниже представлено дерево для RESTCONF API сервера начальной загрузки.

```
module: ietf-sztp-bootstrap-server

  rpcs:
    +---x get-bootstrapping-data
    |  +---w input
    |  |  +---w signed-data-preferred?   empty
    |  |  +---w hw-model?                string
    |  |  +---w os-name?                 string
    |  |  +---w os-version?              string
    |  |  +---w nonce?                   binary
    |  +--ro output
    |     +--ro reporting-level?    enumeration {onboarding-server}?
    |     +--ro conveyed-information    cms
    |     +--ro owner-certificate?      cms
    |     +--ro ownership-voucher?      cms
    +---x report-progress {onboarding-server}?
       +---w input
          +---w progress-type         enumeration
          +---w message?              string
          +---w ssh-host-keys
          |  +---w ssh-host-key* []
          |     +---w algorithm    string
          |     +---w key-data     binary
          +---w trust-anchor-certs
             +---w trust-anchor-cert*   cms
```

7.2. Пример использования

В этом параграфе приведены три примера, иллюстрирующие API сервера начальной загрузки. Два примера относятся к RPC get-bootstrapping-data (для недоверенного и доверенного bootstrap-сервера), один - к RPC report-progress.

В приведённом ниже примере показано устройство, применяющее API для получения данных начальной загрузки от недоверенного bootstrap-сервера. Устройство передаёт входной параметр signed-data-preferred и получает в ответ подписанные данные.

Запрос

```
POST /restconf/operations/ietf-sztp-bootstrap-server:get-bootstrappi\
ng-data HTTP/1.1
HOST: example.com
Content-Type: application/yang-data+xml[1]

<input
  xmlns="urn:ietf:params:xml:ns:yang:ietf-sztp-bootstrap-server">
  <signed-data-preferred/>
</input>
```

Отклик

```
HTTP/1.1 200 OK
Date: Sat, 31 Oct 2015 17:02:40 GMT
Server: example-server
Content-Type: application/yang-data+xml

<output
```

[1]Здесь и ниже в оригинале было ошибочно указано yang.data. См. https://www.rfc-editor.org/errata/eid6933. *Прим. перев.*

```
    xmlns="urn:ietf:params:xml:ns:yang:ietf-sztp-bootstrap-server">
    <conveyed-information>base64encodedvalue==</conveyed-information>
    <owner-certificate>base64encodedvalue==</owner-certificate>
    <ownership-voucher>base64encodedvalue==</ownership-voucher>
  </output>
```

В приведённом ниже примере показано устройство, применяющее API для получения данных начальной загрузки от доверенного bootstrap-сервера. Устройство передаёт серверу начальной загрузки дополнительные входные параметры, которые тот может использовать для подготовки отклика устройству.

Запрос

```
  POST /restconf/operations/ietf-sztp-bootstrap-server:get-bootstrappi\
  ng-data HTTP/1.1
  HOST: example.com
  Content-Type: application/yang-data+xml

  <input
    xmlns="urn:ietf:params:xml:ns:yang:ietf-sztp-bootstrap-server">
    <hw-model>model-x</hw-model>
    <os-name>vendor-os</os-name>
    <os-version>17.3R2.1</os-version>
    <nonce>extralongbase64encodedvalue=</nonce>
  </input>
```

Отклик

```
  HTTP/1.1 200 OK
  Date: Sat, 31 Oct 2015 17:02:40 GMT
  Server: example-server
  Content-Type: application/yang-data+xml

  <output
    xmlns="urn:ietf:params:xml:ns:yang:ietf-sztp-bootstrap-server">
    <reporting-level>verbose</reporting-level>
    <conveyed-information>base64encodedvalue==</conveyed-information>
  </output>
```

Следующий пример показывает устройство, использующее API для отправки отчёта о выполнении серверу начальной загрузки. Приведено сообщение bootstrap-complete, но устройство может передавать серверу другие отчёты. В примере устройство отправляет SSH-ключи хоста и сертификат TLS, которые bootstrap-сервер может, например, передать в NMS, как описано в приложении C.3. Включение устройства.

Запрос

```
  POST /restconf/operations/ietf-sztp-bootstrap-server:report-progress\
   HTTP/1.1
  HOST: example.com
  Content-Type: application/yang-data+xml

  <input
    xmlns="urn:ietf:params:xml:ns:yang:ietf-sztp-bootstrap-server">
    <progress-type>bootstrap-complete</progress-type>
    <message>example message</message>
    <ssh-host-keys>
      <ssh-host-key>
        <algorithm>ssh-rsa</algorithm>
        <key-data>base64encodedvalue==</key-data>
      </ssh-host-key>
      <ssh-host-key>
        <algorithm>rsa-sha2-256</algorithm>
        <key-data>base64encodedvalue==</key-data>
      </ssh-host-key>
    </ssh-host-keys>
    <trust-anchor-certs>
      <trust-anchor-cert>base64encodedvalue==</trust-anchor-cert>
    </trust-anchor-certs>
  </input>
```

Отклик

```
  HTTP/1.1 204 No Content
  Date: Sat, 31 Oct 2015 17:02:40 GMT
  Server: example-server
```

7.3. Модуль YANG

Приведённый ниже модуль YANG формально определяет интерфейс API сервера начальной загрузки, обращённый в сторону устройства. В модуле используются типы данных, определённые в [RFC4253], [RFC5652], [RFC5280] и [RFC8366], кодирование, определённое в [ITU.X690.2015], и ссылки на [RFC4250], [RFC6187], [Std-802.1AR].

```
  <CODE BEGINS> file "ietf-sztp-bootstrap-server@2019-04-30.yang"
  module ietf-sztp-bootstrap-server {
    yang-version 1.1;
    namespace "urn:ietf:params:xml:ns:yang:ietf-sztp-bootstrap-server";
    prefix sztp-svr;

    organization
      "IETF NETCONF (Network Configuration) Working Group";
    contact
      "WG Web:   <https://datatracker.ietf.org/wg/netconf/>
```

```
     WG List:  <mailto:netconf@ietf.org>
     Author:   Kent Watsen <mailto:kent+ietf@watsen.net>";
  description
    "This module defines an interface for bootstrap servers, as
     defined by RFC 8572 ('Secure Zero Touch Provisioning (SZTP)').

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

     Авторские права (Copyright (c) 2019) принадлежат IETF Trust
     и лицам, указанным как авторы кода. Все права защищены.

     Распространение и использование модуля в исходном или двоичном
     формате, с изменениями или без таковых разрешено в соответствии
     с условиями лицензии Simplified BSD License, изложенными в
     разделе 4.c документа IETF Trust's Legal Provisions применительно
     к документам IETF (http://trustee.ietf.org/license-info).

     Эта версия модуля YANG является частью RFC 8572, где правовые
     аспекты изложены более полно.";

  revision 2019-04-30 {
    description
      "Initial version";
    reference
      "RFC 8572: Secure Zero Touch Provisioning (SZTP)";
  }

  // Свойства
  feature redirect-server {
    description
      "Поддерживается функция сервера перенаправления.";
  }

  feature onboarding-server {
    description
      "Поддерживается функция сервера вводных сведений.";
  }

  // Определения типов
  typedef cms {
    type binary;
    description
      "Структура CMS в соответствии с RFC 5652, закодированная по
       правилам ASN.1 DER, заданным в ITU-T X.690.";
    reference
      "RFC 5652: Cryptographic Message Syntax (CMS)
       ITU-T X.690:
         Information technology - ASN.1 encoding rules:
         Specification of Basic Encoding Rules (BER),
         Canonical Encoding Rules (CER) and Distinguished
         Encoding Rules (DER)";
  }

  // RPC
  rpc get-bootstrapping-data {
    description
      "Этот вызов RPC позволяет устройству, указанному именем
       пользователя RESTCONF, получить данные начальной загрузки,
       доступные для него.";
    input {
      leaf signed-data-preferred {
        type empty;
        description
          "Этот необязательный входной параметр позволяет устройству
           взаимодействовать с bootstrap-сервером, который он
           предпочитает для получения подписанных данных. Устройствам
           СЛЕДУЕТ всегда передавать этот параметр недоверенному
           bootstrap-серверу, который при получении параметра ДОЛЖЕН
           возвратить подписанные данные или неподписанные сведения о
           перенаправлении. Серверу НЕДОПУСТИМО возвращать
           неподписанные вводные сведения.";
      }
      leaf hw-model {
        type string;
        description
          "Этот необязательный входной параметр позволяет устройству
           передать bootstrap-серверу свой аппаратный номер модели,
           заданный производителем. Этот параметр может требоваться,
           например, при отсутствии значения hardwareModelName в поле
           subjectAltName его сертификата IDevID, что разрешено
           802.1AR.";
        reference
```

```
            "IEEE 802.1AR: IEEE Standard for Local and
               metropolitan area networks - Secure
               Device Identity";
        }
        leaf os-name {
          type string;
          description
            "Этот необязательный входной параметр позволяет устройству
             передать bootstrap-серверу имя своей операционной системы.
             Этот параметр может быть полезен, если на устройстве,
             указанном серийным номером, могут работать разные ОС.";
        }
        leaf os-version {
          type string;
          description
            "Этот необязательный входной параметр позволяет устройству
             передать bootstrap-серверу версию своей операционной
             системы. Этот параметр может применяться bootstrap-
             сервером для возврата устройству зависящего от ОС отклика
             без необходимости потенциально «дорогого» обновления
             загрузочного образа.";
        }
        leaf nonce {
          type binary {
            length "16..32";
          }
          description
            "Этот необязательный входной параметр позволяет устройству
             передать bootstrap-серверу значение nonce. Это может быть
             особенно полезно для устройств без точных часов, так как
             bootstrap-сервер может динамически получать от
             изготовителя ваучер со значением nonce, как описано в
             RFC 8366.";
          reference
            "RFC 8366:
               A Voucher Artifact for Bootstrapping Protocols";
        }
      }
      output {
        leaf reporting-level {
          if-feature "onboarding-server";
          type enumeration {
            enum minimal {
              description
                "Передать отчёт о выполнении, требуемый RFC 8572.";
              reference
                "RFC 8572: Secure Zero Touch Provisioning (SZTP)";
            }
            enum verbose {
              description
                "Передача дополнительных отчётов о выполнении,
                 которые могут помочь в диагностике проблем SZTP.";
            }
          }
          default "minimal";
          description
            "Задаёт уровень отчётов о выполнении, которые bootstrap-
             сервер хотел бы получать при обработке вводных данных.
             Отчёты не передаются при обработке сведений о
             перенаправлении и недоверенным серверам начальной
             загрузки (например, устройство передало входной
             параметр <signed-data-preferred>).";
        }
        leaf conveyed-information {
          type cms;
          mandatory true;
          description
            "Артефакт передаваемых сведений SZTP, как описано в
             параграфе 3.1 RFC 8572.";
          reference
            "RFC 8572: Secure Zero Touch Provisioning (SZTP)";
        }
        leaf owner-certificate {
          type cms;
          must '../ownership-voucher' {
            description
              "Ваучер владения должен предоставляться при представлении
               сертификата владельца.";
          }
          description
            "Артефакт сертификата владельца, описанный в параграфе 3.2
             RFC 8572. Этот лист не обязателен, поскольку он нужен лишь
             в случае подписанного артефакта передаваемых сведений.";
          reference
            "RFC 8572: Secure Zero Touch Provisioning (SZTP)";
        }
```

```
        leaf ownership-voucher {
          type cms;
          must '../owner-certificate' {
            description
              "Сертификат владельца должен быть предоставлен при
               представлении ваучера владения.";
          }
          description
            "Артефакт ваучера владения, описанный в параграфе 3.3
             RFC 8572. Этот лист не обязателен, поскольку он нужен лишь
             в случае подписанного артефакта передаваемых сведений.";
          reference
            "RFC 8572: Secure Zero Touch Provisioning (SZTP)";
        }
      }
    }

    rpc report-progress {
      if-feature "onboarding-server";
      description
        "Этот вызов RPC позволяет устройству, указанному именем
         пользователя RESTCONF, сообщить bootstrap-серверу о своём
         процессе загрузки. Предполагается применение RPC при
         получении устройством вводных сведений от доверенного
         сервера начальной загрузки.";
      input {
        leaf progress-type {
          type enumeration {
            enum bootstrap-initiated {
              description
                "Указывает, что устройство применяет RPC
                 get-bootstrapping-data. Узел message ниже МОЖЕТ
                 включать любые дополнительные сведения, которые
                 изготовитель может считать полезными.";
            }
            enum parsing-initiated {
              description
                "Указывает, что устройство собирается начать анализ
                 вводных сведений. Это предназначено лишь для случаев,
                 когда анализ реализован в виде отдельного этапа.";
            }
            enum parsing-warning {
              description
                "Указывает, что устройство столкнулось с некритической
                 ошибкой при анализе отклика от сервера. В узле message
                 ниже СЛЕДУЕТ указать полученное предупреждение.";
            }
            enum parsing-error {
              description
                "Указывает, что устройство столкнулось с критической
                 ошибкой при анализе отклика от сервера. Например, это
                 может быть результат ошибочного кодирования, получения
                 устройством неподписанных данных вместо подписанных,
                 отсутствия в ваучере владения серийного номера
                 устройства, несоответствия подписи. В узле message
                 ниже СЛЕДУЕТ указать конкретную ошибку. Этот тип также
                 указывает, что устройство отказалось от попытки
                 начальной загрузки с этого bootstrap-сервера.";
            }
            enum parsing-complete {
              description
                "Указывает, что устройство успешно завершило анализ
                 вводных сведений. Этот тип применяется лишь при
                 реализации анализа на отдельном этапе.";
            }
            enum boot-image-initiated {
              description
                "Указывает, что устройство начинает обработку данных
                 загрузочного образа.";
            }
            enum boot-image-warning {
              description
                "Указывает, что устройство столкнулось с некритической
                 ошибкой при попытке установить загрузочный образ.
                 Возможные причины могут включать необходимость
                 форматирования раздела с потерей данных. В узле message
                 ниже СЛЕДУЕТ указать полученные предупреждения.";
            }
            enum boot-image-error {
              description
                "Указывает, что устройство столкнулось с ошибкой
                 при попытке установить загрузочный образ. Это
                 может быть связано с недоступностью файлового сервера,
                 отсутствием файла, несоответствием подписи и пр. В
                 узле message ниже СЛЕДУЕТ указать конкретную ошибку.
                 Этот тип также  указывает, что устройство отказалось
```

```
      от попытки начальной загрузки с этого сервера.";
  }
  enum boot-image-mismatch {
    description
      "Указывает, что устройство определило, что на нем
       работает некорректный загрузочный образ. Этому
       сообщению СЛЕДУЕТ вызывать попытку загрузить
       (download) образ.";
  }
  enum boot-image-installed-rebooting {
    description
      "Указывает, что устройство установило новый загрузочный
       образ и начинает перезапуск. После отправки этого
       сообщения от устройства не ожидается нового обращения
       к bootstrap-серверу для попытки начальной загрузки.";
  }
  enum boot-image-complete {
    description
      "Указывает, что устройство считает, что на нем работает
       корректный загрузочный образ.";
  }
  enum pre-script-initiated {
    description
      "Указывает, что устройство начинает выполнение сценария
       pre-configuration-script.";
  }
  enum pre-script-warning {
    description
      "Указывает, что устройство получило предупреждение при
       выполнении сценария pre-configuration-script. В узел
       message ниже СЛЕДУЕТ включить весь вывод сценария.";
  }
  enum pre-script-error {
    description
      "Указывает, что устройство столкнулось с ошибкой при
       выполнении сценария pre-configuration-script. В узел
       message ниже СЛЕДУЕТ включить весь вывод сценария.
       Этот тип также  указывает, что устройство отказалось
       от попытки начальной загрузки с этого сервера.";
  }
  enum pre-script-complete {
    description
      "Указывает, что устройство успешно выполнило сценарий
       pre-configuration-script.";
  }
  enum config-initiated {
    description
      "Указывает, что устройство начинает применять исходную
       конфигурацию.";
  }
  enum config-warning {
    description
      "Указывает, что устройство получило предупреждение при
       представлении исходной конфигурации. В узел message
       ниже СЛЕДУЕТ включить все выданные предупреждения.";
  }
  enum config-error {
    description
      "Указывает, что устройство столкнулось с ошибкой при
       представлении исходной конфигурации. В узел message
       ниже СЛЕДУЕТ включить все сообщения об ошибках.
       Этот тип также  указывает, что устройство отказалось
       от попытки начальной загрузки с этого сервера.";
  }
  enum config-complete {
    description
      "Указывает, что устройство успешно установило исходную
       конфигурацию.";
  }
  enum post-script-initiated {
    description
      "Указывает, что устройство начинает выполнение сценария
       post-configuration-script.";
  }
  enum post-script-warning {
    description
       "Указывает, что устройство получило предупреждение при
       выполнении сценария post-configuration-script. В узел
       message ниже СЛЕДУЕТ включить весь вывод сценария.";
  }
  enum post-script-error {
    description
      "Указывает, что устройство столкнулось с ошибкой при
       выполнении сценария post-configuration-script. В узел
       message ниже СЛЕДУЕТ включить весь вывод сценария.
       Этот тип также  указывает, что устройство отказалось
```

```
            от попытки начальной загрузки с этого сервера.";
        }
        enum post-script-complete {
          description
            "Указывает, что устройство успешно выполнило сценарий
             post-configuration-script.";
        }
        enum bootstrap-warning {
          description
            "Указывает предупреждение, для которого не подходит ни
             одно из перечисляемых progress-type. В узле message
             СЛЕДУЕТ описать предупреждение.";
        }
        enum bootstrap-error {
          description
            "Указывает ошибку, для которой не подходит ни одно
             из перечисляемых progress-type. В узле message СЛЕДУЕТ
             описать ошибку. Этот тип также  указывает, что
             устройство отказалось от попытки начальной загрузки
             с этого сервера.";
        }
        enum bootstrap-complete {
          description
            "Указывает, что устройство успешно обработало целиком
             onboarding-information и готово к управлению. Узел
             message ниже МОЖЕТ включать любые дополнительные
             сведения, которые изготовитель может счесть полезными.
             После отправки этого сообщения от устройства не
             ожидается нового обращения к bootstrap-серверу для
             попытки начальной загрузки.";
        }
        enum informational {
          description
            "Указывает дополнительную информацию, не включённую в
             другие типы, например, сообщение указывающее, что
             устройство начинает перезапуск после установки
             предоставленного образа загрузки. В узел message ниже
             СЛЕДУЕТ включит сведения, которые изготовитель может
             счесть полезными.
        }
      }
      mandatory true;
      description
        "Тип предоставляемого отчёта о выполнении.";
    }
    leaf message {
      type string;
      description
        "Необязательное произвольное значение.";
    }
    container ssh-host-keys {
      when "../progress-type = 'bootstrap-complete'" {
        description
          "SSH-ключи хоста передаются лишь для типа
           bootstrap-complete'.";
      }
      description
        "Список SSH-ключей хоста, которые NMS может использовать
         для аутентификации последующих соединений SSH с этим
         устройством (например, netconf-ssh, netconf-ch-ssh).";
      list ssh-host-key {
        description
          "SSH-ключ хоста, который NMS может использовать для
           аутентификации последующих соединений SSH с этим
           устройством (например, netconf-ssh, netconf-ch-ssh).";
        reference
          "RFC 4253: The Secure Shell (SSH) Transport Layer
                     Protocol";
        leaf algorithm {
          type string;
          mandatory true;
          description
            "Имя алгоритма с открытым ключом для этого ключа SSH.

             Пригодные значения указаны в субреесте Public Key
             Algorithm Names реестра Secure Shell (SSH) Protocol
             Parameters, поддерживаемого IANA.";
          reference
            "RFC 4250: The Secure Shell (SSH) Protocol Assigned
                       Numbers
             IANA URL: <https://www.iana.org/assignments/ssh-parame\\
                        ters>
                       (\\ добавлено для форматирования)";
        }
        leaf key-data {
          type binary;
```

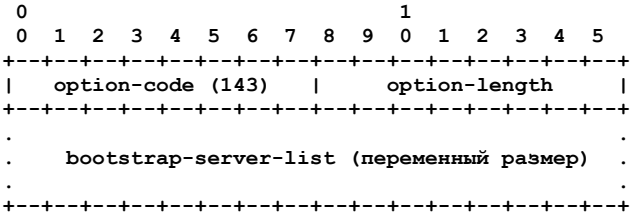
```
            mandatory true;
            description
              "Двоичные данные открытого ключа для этого ключа SSH,
               как указано в параграфе 6.6 RFC 4253, т. е.:
                 string    идентификатор формата сертификата или
                           открытого ключа
                   byte[n]   данные ключа/сертификата.";
            reference
              "RFC 4253: The Secure Shell (SSH) Transport Layer
                         Protocol";
          }
        }
      }
      container trust-anchor-certs {
        when "../progress-type = 'bootstrap-complete'" {
          description
            "Привязки доверия передаются лишь для отчётов типа
             bootstrap-complete.";
        }
        description
          "Список сертификатов привязок доверия, которые NMS может
           использовать для аутентификации последующих соединений с
           этим устройством по сертификату (например., restconf-tls,
           netconf-tls и даже netconf-ssh с поддержкой X.509 из
           RFC 6187). На практике привязки доверия для сертификатов
           IDevID не нужно передавать с помощью этого механизма.";
        reference
          "RFC 6187: X.509v3 Certificates for Secure Shell
                     Authentication";
        leaf-list trust-anchor-cert {
          type cms;
          description
            "Структура CMS, где тип содержимого верхнего уровня
             ДОЛЖЕН быть signed-data, как указано в разделе 5
             RFC 5652.

             В CMS ДОЛЖНА содержаться цепочка сертификатов X.509,
             требуемых для аутентификации представленного устройством
             сертификата.

             В CMS  ДОЛЖНА содержаться лишь 1 цепочка сертификатов.
             Последним в цепочке ДОЛЖЕН быть эмитент сертификата
             устройства.

             Во всех случаях цепочка ДОЛЖНА включать самоподписанный
             корневой сертификат. Если эмитентом сертификата
             устройства является корневой CA, присутствует лишь его
             сертификат.

             CMS представляется вырожденной формой структуры
             SignedData, которая обычно служит для распространения
             сертификатов X.509 и объектов отзыва (RFC 5280).";
          reference
            "RFC 5280: Internet X.509 Public Key Infrastructure
                       Certificate and Certificate Revocation List
                       (CRL) Profile
             RFC 5652: Cryptographic Message Syntax (CMS)";
        }
      }
    }
  }
}
<CODE ENDS>
```

8. Опции DHCP

В этом разделе определены две опции DHCP - по одной для DHCPv4 и DHCPv6. Опции имеют одинаковую семантику, но разный синтаксис.

8.1. Опция DHCPv4 SZTP Redirect

Опция DHCPv4 SZTP Redirect служит для предоставления клиенту одного или нескольких URI, указывающих серверы начальной загрузки, с которых можно пытаться получить дополнительную конфигурацию.



option-code

OPTION_V4_SZTP_REDIRECT (143).

option-length
Размер опции в октетах.
bootstrap-server-list
Список серверов, с которыми клиент может пытаться контактировать для получения дополнительных данных начальной загрузки в формате, показанном в параграфе 8.3. Базовое кодирование полей.

Поведение клиента DHCPv4

Клиент **может** запросить опцию OPTION_V4_SZTP_REDIRECT включением её кода в Parameter Request List (55) запроса DHCP.

При получении сообщения DHCPv4 Reply с опцией OPTION_V4_SZTP_REDIRECT клиент обрабатывает отклик в соответствии с параграфом 5.5, считая что значения address и port закодированы в URI.

Недействительные записи URI в поле uri-data клиент игнорирует. Если принятая опция OPTION_V4_SZTP_REDIRECT не включает хотя бы одной действительной записи URI в поле uri-data, клиент **должен** отбросить опцию.

Поскольку размер URI может превышать разрешённую длину опции DHCPv4 (255 октетов), клиент **должен** реализовать поведение агента декодирования, описанное в [RFC3396], для корректной обработки списка URI, разнесённого по нескольким принятым экземплярам опции OPTION_V4_SZTP_REDIRECT.

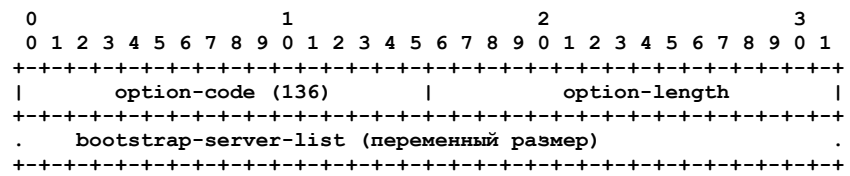
Поведение сервера DHCPv4

Сервер DHCPv4 **может** включать опцию OPTION_V4_SZTP_REDIRECT в передаваемые сообщения DHCP.[1]

Сообщение от сервера DHCP **должно** включать лишь одно поле bootstrap-server-list в опцию OPTION_V4_SZTP_REDIRECT. Однако список URI в этом поле может выходить за пределы разрешённого размера опций DHCPv4 (в соответствии с [RFC3396]). Если поле bootstrap-server-list помещается в один экземпляр OPTION_V4_SZTP_REDIRECT, серверу **недопустимо** передавать более 1 экземпляра опции. Если размер поля bootstrap-server-list слишком велик для одной опции, опция OPTION_V4_SZTP_REDIRECT **должна** расщепляться на несколько экземпляров в соответствии с процессом, описанным в [RFC3396].

8.2. Опция DHCPv6 SZTP Redirect

Опция DHCPv6 SZTP Redirect служит для предоставления клиенту одного или нескольких URI, указывающих серверы начальной загрузки, с которых можно пытаться получить дополнительную конфигурацию.



option-code
OPTION_V6_SZTP_REDIRECT (136).
option-length
Размер опции в октетах.
bootstrap-server-list
Список серверов, с которыми клиент может пытаться контактировать для получения дополнительных данных начальной загрузки в формате, показанном в параграфе 8.3. Базовое кодирование полей.

Поведение клиента DHCPv6

Клиент **может** запросить опцию OPTION_V6_SZTP_REDIRECT с помощью процесса, заданного в параграфах 18.2.1, 18.2.2, 18.2.4, 18.2.5, 18.2.6 и 21.7 [RFC8415]. Клиент включает коды запрашиваемых опций в опцию Option Request.

При получении сообщения DHCPv6 Reply с опцией OPTION_V6_SZTP_REDIRECT клиент обрабатывает отклик в соответствии с параграфом 5.5, считая что значения address и port закодированы в URI.

Недействительные записи URI в поле uri-data клиент игнорирует. Если принятая опция OPTION_V6_SZTP_REDIRECT не включает хотя бы одной действительной записи URI в поле uri-data, клиент **должен** отбросить опцию.

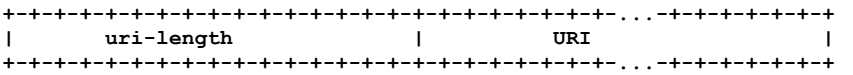
Поведение сервера DHCPv6

В параграфе 18.3 [RFC8415] описана работа сервера в части назначения опций. Сервер будет передавать конкретный код опции лишь в том случае, когда для этой опции заданы значения кодов и клиент запросил опцию.

Опция OPTION_V6_SZTP_REDIRECT является одиночной (singleton) и серверу **недопустимо** передавать более одного экземпляра этой опции.

8.3. Базовое кодирование полей

Опции DHCPv4 и DHCPv6, заданные здесь, кодируют список URI серверов начальной загрузки. Структура URI является опцией DHCP, которая может содержать несколько URI (см. параграф 5.7 в [RFC7227]). Формат записи URI в поле bootstrap-server-list показан ниже.



uri-length
2-октетное поле размера данных URI.
URI
URI для сервера начальной загрузки SZTP.

В URI сервера начальной загрузки SZTP **должна** использоваться схема URI https, определённая в параграфе 2.7.2 [RFC7230], и форма **должна** иметь вид https://<ip-address-or-hostname>[:<port>].

[1] В оригинале этот абзац содержал ошибку. См. https://www.rfc-editor.org/errata/eid6684. *Прим. перев.*

9. Вопросы безопасности

9.1. Точность часов

Описанное в документе решение полагается на сертификаты TLS и владельцев, а также ваучеры владения, для корректной **работы** (например, проверка срока действия или статуса отзыва) с которыми требуются точные часы. Реализациям следует обеспечивать на устройствах точность часов при отгрузке с производства и предпринимать шаги по предотвращению искажений работы часов.

При отсутствии возможности обеспечить точность часов реализации **рекомендуется** отключить все аспекты, связанные с проверкой времени. В частности, таким реализациям следует считать, что сертификаты TLS, ваучеры владения и другие сертификаты действительны всегда и не могут быть отозваны. Для ваучеров владения изготовителям **следует** задавать один ваучер на весь срок службы устройства.

Реализациям **не следует** полагаться на протокол NTP, поскольку в настоящее время он не обеспечивает защиты. Отметим документ IETF, посвящённый защите протокола NTP [NTS-NTP].

9.2. Использование сертификатов IDevID

Сертификаты IDevID, определённые в [Std-802.1AR], **рекомендуются** как для клиентов TLS при подключении к серверам начальной загрузки, так и для идентификации устройств при шифровании артефактов данных начальной загрузки SZTP.

9.3. Неизменяемое хранилище для привязок доверия

Устройства **должны** гарантировать, что все их сертификаты привязок доверия, включая используемые для соединений с серверами начальной загрузки и проверки ваучеров владения, защищены от постороннего вмешательства.

Со временем может потребоваться обновление этих сертификатов (например, изготовитель может добавить CA). Поэтому предполагается, что устройства **могут** обновить эти привязки доверия при возникновении необходимости с помощью проверяемого процесса, такого как обновление программ с использованием подписанных образов.

9.4. Защищённое хранилище для долгосрочных секретных ключей

Созданные изготовителем идентификаторы устройства могут иметь очень долгий срок действия. Например, в [Std-802.1AR] рекомендуется использовать notAfter = 99991231235959Z в сертификатах IDevID. С учётом длительного использования этих секретных ключей крайне важно хранить их так, что они были недоступны, например, в защищённом криптографическом процессоре (микросхема TPM).

9.5. Аутентификация сервера начальной загрузки вслепую

Этот документ разрешает устройству вслепую аутентифицировать сертификат TLS сервера начальной загрузки. Это делается для случаев, когда сведения о перенаправлении получены без защиты, что иногда следует разрешать.

Для компенсации этого документ требует, чтобы при соединении с недоверенным сервером начальной загрузки устройства проверяли подписи для загружаемых с сервера данных.

9.6. Раскрытие информации недоверенным серверам

Этот документ разрешает устройствам подключение к недоверенным серверам начальной загрузки. Однако такой сервер может контролироваться злоумышленниками, поэтому устройствам **следует** соблюдать осторожность при передаче данных на сервер начальной загрузки.

Устройства передают bootstrap-серверам разные данные на каждом уровне протоколов - TCP, TLS, HTTP, RESTCONF.

На уровне TCP устройства передают свой адрес IP, который может транслироваться в сети, и риска безопасности здесь не возникает.

На уровне TLS устройства могут использовать клиентские сертификаты для идентификации и аутентификации себя на недоверенном сервере начальной загрузки. Сертификат должен указывать по меньшей мере серийный номер устройства и может раскрывать дополнительные сведения, такие как изготовитель, аппаратная версия (модель), открытый ключ и т. п. Наличие таких сведений может позволить злоумышленнику организовать атаку. **Рекомендуется** не полагаться на защиту сети в таких случаях.

На уровне HTTP устройства могут применять схему аутентификации HTTP для идентификации и подтверждения своей подлинности недоверенным серверам начальной загрузки. Схема аутентификации раскрывает, по меньшей мере, серийный номер устройства, а в зависимости от применяемого механизма аутентификации, может раскрывать секрет, который предполагается известным лишь устройству (например, пароль). Устройствам **не следует** применять схемы аутентификации HTTP (например, HTTP Basic) с недоверенными bootstrap-серверами, раскрывающими секрет, который предполагается известным лишь устройству.

На уровне RESTCONF устройства применяют RPC get-bootstrapping-data (но не report-progress) при подключении к недоверенному серверу начальной загрузки. RPC get-bootstrapping-data позволяет передавать bootstrap-серверу дополнительные входные параметры (например, os-name, os-version, hw-model). Устройствам **рекомендуется** передавать недоверенным серверам лишь параметр signed-data-preferred. Хотя для bootstrap-сервера нормально сразу же возвращать подписанные вводные сведения, **рекомендуется** вместо этого переводить недоверенное соединение в доверенное, как описано в Приложении B, что позволяет устройству использовать RPC report-progress при обработке вводных сведений.

9.7. Упорядочение источников данных начальной загрузки

Для устройств, поддерживающих более одного источника данных для начальной загрузки, не требуется соблюдать какой-то конкретный порядок источников, поскольку все они считаются одинаково защищёнными. Однако с точки зрения приватности **рекомендуется** предпочитать источники с локальным доступом.

9.8. Безопасность секретных ключей, применяемых для доверия

Описанное в документе решение позволяет обеспечивать доверие к данным начальной загрузки двумя способами - защитой на транспортном уровне или подписыванием артефактов.

При использовании транспортной защиты (доверенный сервер начальной загрузки) секретный ключ для сертификата конечного объекта должен быть доступен для организации соединения TLS.

При подписывании артефактов ключ подписи должен быть доступен лишь при возврате bootstrap-сервером созданных динамически подписанных откликов с данными. Например, сервер начальной загрузки при получении входного параметра signed-data-preferred для RPC get-bootstrapping-data может динамически генерировать отклик с подписью.

Администраторам серверов начальной загрузки **рекомендуется** следовать проверенным (best practice) методам защиты секретных ключей, применяемых в online-операциях. Например, **рекомендуется** применять модули аппаратной защиты (hardware security module или HSM). Если HSM не используется, **рекомендуется** чаще обновлять секретный ключ при наличии на всех устройствах с удалённой начальной загрузкой точных часов (см. параграф 9.1).

Для лучшей защиты владельцам **рекомендуется** предоставлять лишь подписанные (с использованием защищённого секретного ключа) и зашифрованные (с использованием открытого ключа устройства из его сертификата отождествления защищённого устройства) данные начальной загрузки.

9.9. Уверенность в изготовителях

Представленный в этом документе протокол начальной загрузки SZTP передаёт некоторый контроль над начальной конфигурацией от владельца устройства изготовителю и его представителям. Изготовитель поддерживает список общеизвестных bootstrap-серверов, которым доверяют его устройства. По замыслу, если сначала не найдено данных начальной загрузки иными методами, устройство пытается связаться с общеизвестными серверами начальной загрузки. Нет никакого способа предотвратить это, кроме использования внешнего межсетевого экрана для блокировки таких соединений. Проблемы, связанные с доверенными bootstrap-серверами, рассмотрены в параграфе 9.10.

Изготовитель также ведёт список органов, подписывающих ваучеры, которым его устройства доверяют. Такие органы выпускают ваучеры, позволяющие владельцу доверять сертификату домена владельца. Крайне важно, чтобы изготовители обеспечивали целостность органов, подписывающих ваучеры, чтобы избежать некорректных назначений.

Операторам следует знать, что система предполагает их доверие всем заданным заранее серверам начальной загрузки, назначенным изготовителем. Хотя операторы могут использовать в сети средства блокировки доступа к общеизвестным bootstrap-серверам, они не могут запретить подписывающим ваучеры органам создавать ваучеры для их устройств.

9.10. Проблемы с доверенными серверами начальной загрузки

С общеизвестными и обнаруженными доверенными серверами могут быть связаны некоторые проблемы.

- Скомпрометированный доверенный сервер можно изменить для возврата неподписанных данных любого типа. Например, bootstrap-сервер, от которого предполагается получение лишь сведений о перенаправлении, можно настроить на возврат вводных данных, а сервер, от которого ожидаются подписанные данные, - на возврат неподписанных. В обоих случаях устройство будет воспринимать отклик, не зная, что тот должен быть иным. Сопровождающим доверенные серверы начальной загрузки **рекомендуется** обеспечивать защиту серверов от компрометации, а при возникновении таковой иметь механизмы для быстрого обнаружения и устранения последствий.
- Доверенный сервер начальной загрузки, на котором размещаются неподписанные или подписанные, но не зашифрованные данные, может раскрывать информацию нежелательным сторонам (например, своему администратору). Это проблема приватности, но она может раскрыть сведения, пригодные для организации атаки. Раскрытие сведений о перенаправлении имеет ограниченное влияние (это просто список bootstrap-серверов), тогда как раскрытие вводных данных может давать много информации (например, топологию сети, правила межсетевых экранов и т. п.). Операторам **рекомендуется** шифровать данные начальной загрузки, когда их содержимое считается важным, вплоть до сокрытия таких данных от администраторов bootstrap-сервера, который может поддерживаться сторонними лицами.

9.11. Срок действия переносимых сведений

Артефакт переносимых сведений не задаёт срок действия. Например, в сведениях о перенаправлении или вводных данных нет временных ограничений и ни один из этих артефактов нельзя отозвать.

Для неподписанных данных из недоверенного источника данных начальной загрузки нет смысла обсуждать срок действия сведений, коль скоро они не аутентифицируются и могут быть получены откуда угодно. Для неподписанных данных из доверенного источника (bootstrap-сервер) доступность данных является единственным мерилом их актуальности. Поскольку недоверенные данные приходят из доверенного источника, их текущая доступность имеет значение, а поскольку bootstrap-серверы используют TLS, содержимое обмена нельзя изменить или воспроизвести. Для подписанных данных из доверенного или недоверенного источника пригодность ограничена пригодностью ваучера владения и сертификата владельца, применяемых для аутентификации данных.

Пригодность ваучера владения в первую очередь ограничивается узлами created-on и expires-on. Хотя [RFC8366] рекомендует краткосрочные ваучеры (параграф 6.1), узел expires-on может указывать любое время в будущем и даже отсутствовать, указывая неограниченный срок действия. Пригодность ваучера ограничивается также PKI изготовителя, использованной для подписи ваучера. Хотя ваучер нельзя отозвать напрямую, для использованной при подписи PKI отзыв возможен.

Срок действия сертификата владельца в первую очередь ограничивается полем пригодности X.509 - значения notBefore и notAfter задаются удостоверяющим центром при подписании сертификата. Действие сертификата владельца ограничивается также пригодностью PKI, использованной при подписании ваучера. Сертификат владельца можно отозвать напрямую.

Владельцам, желающим иметь максимальную гибкость управления пригодностью подписанных данных, **рекомендуется** создавать уникальный сертификат владельца для каждого артефакта. Это не только позволяет указать срок действия артефакта, но и позволяет отозвать пригодность каждого артефакта.

9.12. Каскадирование доверия через перенаправление

Сведения о перенаправлении (параграф 2.1) предписывают загружающемуся устройству инициировать соединение HTTPS с указанными серверами начальной загрузки. Доверенные сведения о перенаправлении могут включать сертификат привязки доверия, применяемый устройством для аутентификации сертификата TLS конечного объекта, представленного bootstrap-сервером. В результате компрометация взаимодействия, обеспеченного сведениями о перенаправлении, ведёт к компрометации всех последующих взаимодействий.

9.13. Возможность повторного применения секретных ключей

В этом документе описано два применения сертификатов идентификации защищённых устройств. В основном это служит для аутентификации устройством себя на сервере начальной загрузки с использованием секретного ключа для проверки подлинности по клиентскому сертификату уровня TLS. Другим применением является расшифровка представленных артефактов начальной загрузки, точнее, в соответствии с разделом 6 в [RFC5652], расшифровка симметричного ключа, служащего для расшифровки данных.

Параграф 3.4. Шифрование артефактов допускает возможность применения одного сертификата идентификации защищённого устройства в обоих случаях, поскольку в [Std-802.1AR] указано, что сертификат DevID **может** иметь установленный бит KeyUsage keyEncipherment в дополнение к биту KeyUsage digitalSignature.

Хотя понятно, что повторное использование секретных ключей обычно не одобряется, этот документ считает такое использование приемлемым, поскольку нет каких-либо известных способов для подписи, созданной в одном контексте, вызвать (неверную) интерпретацию в другом.

9.14. Отсутствие проблем с шифрованием подписанных артефактов

Этот документ задаёт шифрование подписанных объектов, а не подписывание шифрованных, как можно было бы ожидать с учётом широко разрекламированных атак оракула (например, padding oracle attack). В документе такие атаки не считаются осуществимыми в контексте решения, поскольку расшифрованный текст никогда не покидает устройство.

9.15. Модуль YANG ietf-sztp-conveyed-info

Заданный в документе модуль ietf-sztp-conveyed-info определяет структуру данных, которая всегда оборачивается в структуру CMS. При доступе защищённого механизма (например, TLS) структура CMS может не подписываться. Однако при доступе незащищённых механизмов (например, съёмное хранилище) структура CMS должна быть подписана, чтобы устройство доверяло ей.

Реализациям следует учитывать, что подписанные данные начальной загрузки защищены лишь от изменения, содержимое остаётся видимым для других. Это не так влияет на безопасность, как на приватность. Содержимое могут читать посторонние при использовании незащищённых механизмов. В модуле ietf-sztp-conveyed-info задан оператор верхнего уровня choice, объявляющий, что содержимое имеет тип redirect-information или onboarding-information. Далее рассмотрены оба случая.

Когда структура CMS содержит redirect-information, наблюдатель может узнать bootstrap-серверы, куда направляется устройство, их адреса IP или имена, порты и сертификаты привязок доверия. Эти сведения могут дать некоторое представление о внутренней структуре сети.

Когда структура CMS содержит onboarding-information, наблюдатель может получить важные сведения о подготовке устройства. Эти сведения включают версию операционной системы, исходную конфигурацию и содержимое сценариев. Такая информация может быть конфиденциальной и следует принять меры для её защиты (например, шифровать артефакт с использованием открытого ключа устройства).

9.16. Модуль YANG ietf-sztp-bootstrap-server

Определённый в документе модуль ietf-sztp-bootstrap-server задаёт API для RESTCONF [RFC8040]. Нижним уровнем RESTCONF является HTTPS с обязательной реализацией защищённого транспорта TLS [RFC8446]. Модель доступа NACM (NETCONF Access Control Model) [RFC8341] обеспечивает предоставление доступа к предопределённому подмножеству протокольных операций и содержимого лишь указанным пользователям. Этот модуль не содержит узлов данных (только RPC), поэтому нет необходимости обсуждать конфиденциальность узлов данных.

Модуль определяет два операции RPC которые могут быть чувствительными в некоторых сетевых средах.

get-bootstrapping-data

Этот вызов RPC применяется устройствами для получения данных начальной загрузки. По замыслу каждое устройство, указанное свидетельствами аутентификации (например, сертификат клиента), может получить лишь свои данные. Для дополнительного ограничения доступа с этим вызовом RPC применение NACM не требуется.

report-progress

Этот вызов RPC применяется устройствами для отчётов о выполнении начальной загрузки. По замыслу каждое устройство, указанное свидетельствами аутентификации (например, сертификат клиента), может получить лишь свои данные. Для дополнительного ограничения доступа с этим вызовом RPC применение NACM не требуется.

10. Взаимодействие с IANA

10.1. Реестр IETF XML

Агентство IANA зарегистрировало два URI в субреестре ns реестра IETF XML Registry [RFC3688], доступного по ссылке <https://www.iana.org/assignments/xml-registry>. Ниже приведены регистрации в соответствии с форматом [RFC3688].

```
URI: urn:ietf:params:xml:ns:yang:ietf-sztp-conveyed-info
```

```
Registrant Contact: Рабочая группа NETCONF в IETF.
XML: N/A, URI относится к пространству имён XML.

URI: urn:ietf:params:xml:ns:yang:ietf-sztp-bootstrap-server
Registrant Contact: Рабочая группа NETCONF в IETF.
XML: N/A, URI относится к пространству имён XML.
```

10.2. Реестр YANG Module Names

Агентство IANA зарегистрировало два модуля YANG в реестре YANG Module Names [RFC6020], доступном по ссылке <https://www.iana.org/assignments/yang-parameters>. Ниже приведены регистрации в формате [RFC6020].

```
name:      ietf-sztp-conveyed-info
namespace: urn:ietf:params:xml:ns:yang:ietf-sztp-conveyed-info
prefix:    sztp-info
reference: RFC 8572

name:      ietf-sztp-bootstrap-server
namespace: urn:ietf:params:xml:ns:yang:ietf-sztp-bootstrap-server
prefix:    sztp-svr
reference: RFC 8572
```

10.3. Реестр SMI Security for S/MIME CMS Content

Агентство IANA зарегистрировало два идентификатора подчинённых объектов в реестре SMI Security for S/MIME CMS Content Type (1.2.840.113549.1.9.16.1), доступном по ссылке <https://www.iana.org/assignments/smi-numbers>. Ниже приведена регистрация в формате, заданном в параграфе 3.4 [RFC7107].

Десятичное значение	*Описание*	*Документ*
42	id-ct-sztpConveyedInfoXML	RFC 8572
43	id-ct-sztpConveyedInfoJSON	RFC 8572

id-ct-sztpConveyedInfoXML указывает, что conveyed-information кодируется с использованием XML, id-ct-sztpConveyedInfoJSON указывает кодирование conveyed-information с использованием JSON.

10.4. Реестр BOOTP Vendor Extensions and DHCP Options

Агентство IANA зарегистрировало код DHCP в реестре BOOTP Vendor Extensions and DHCP Options, доступном по ссылке <https://www.iana.org/assignments/bootp-dhcp-parameters>.

```
Tag:         143
Name:        OPTION_V4_SZTP_REDIRECT
Data Length: N
Meaning:     Список URI для bootstrap-серверов SZTP
Reference:   RFC 8572
```

10.5. Реестр Dynamic Host Configuration Protocol for IPv6 (DHCPv6)

Агентство IANA зарегистрировало код DHCP в субреестре Option Codes реестра Dynamic Host Configuration Protocol for IPv6 (DHCPv6), доступного по ссылке <https://www.iana.org/assignments/dhcpv6-parameters>.

```
Value:            136
Description:      OPTION_V6_SZTP_REDIRECT
Client ORO:       Yes
Singleton Option: Yes
Reference:        RFC 8572
```

10.6. Реестр Service Name and Transport Protocol Port Number

Агентство IANA зарегистрировало имя службы в реестре Service Name and Transport Protocol Port Number [RFC6335], доступном по ссылке <https://www.iana.org/assignments/service-names-port-numbers>. Ниже приведена регистрация в формате, заданном в параграфе 8.1.1 [RFC6335].

```
Service Name:            sztp
Transport Protocol(s):   TCP
Assignee:                IESG <iesg@ietf.org>
Contact:                 IETF Chair <chair@ietf.org>
Description:             Это имя службы применяется при создании метки
                         сервиса SRV _sztp для обнаружения серверов
                         начальной загрузки SZTP.
Reference:               RFC 8572
Port Number:             N/A
Service Code:            N/A
Known Unauthorized Uses: N/A
Assignment Notes:        Этот протокол применяет HTTPS как подложку.
```

10.7. Реестр Underscored and Globally Scoped DNS Node Names

Агентство IANA зарегистрировало имя службы в субреестре Underscored and Globally Scoped DNS Node Names [RFC8552] реестра Domain Name System (DNS) Parameters, доступного по ссылке <https://www.iana.org/assignments/dns-parameters>. Ниже приведена регистрация в формате раздела 3 [RFC8552].

```
RR Type:            TXT
_NODE NAME:         _sztp
Reference:          RFC 8572
```

11. Литература

11.1. Нормативные документы

[ITU.X690.2015] International Telecommunication Union, "Information Technology - ASN.1 encoding rules: Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)", ITU-T Recommendation X.690, ISO/IEC 8825-1, August 2015, <https://www.itu.int/rec/T-REC-X.690/>.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <https://www.rfc-editor.org/info/rfc1035>.

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

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, DOI 10.17487/RFC2782, February 2000, <https://www.rfc-editor.org/info/rfc2782>.

[RFC3396] Lemon, T. and S. Cheshire, "Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4)", RFC 3396, DOI 10.17487/RFC3396, November 2002, <https://www.rfc-editor.org/info/rfc3396>.

[RFC4253] Ylonen, T. and C. Lonvick, Ed., "The Secure Shell (SSH) Transport Layer Protocol", RFC 4253, DOI 10.17487/RFC4253, January 2006, <https://www.rfc-editor.org/info/rfc4253>.

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

[RFC5652] Housley, R., "Cryptographic Message Syntax (CMS)", STD 70, RFC 5652, DOI 10.17487/RFC5652, September 2009, <https://www.rfc-editor.org/info/rfc5652>.

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

[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, DOI 10.17487/RFC6125, March 2011, <https://www.rfc-editor.org/info/rfc6125>.

[RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <https://www.rfc-editor.org/info/rfc6762>.

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

[RFC7227] Hankins, D., Mrugalski, T., Siodelski, M., Jiang, S., and S. Krishnan, "Guidelines for Creating New DHCPv6 Options", BCP 187, RFC 7227, DOI 10.17487/RFC7227, May 2014, <https://www.rfc-editor.org/info/rfc7227>.

[RFC7230] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and Routing", RFC 7230, DOI 10.17487/RFC7230, June 2014, <https://www.rfc-editor.org/info/rfc7230>.

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

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

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

[RFC8366] Watsen, K., Richardson, M., Pritikin, M., and T. Eckert, "A Voucher Artifact for Bootstrapping Protocols", RFC 8366, DOI 10.17487/RFC8366, May 2018, <https://www.rfc-editor.org/info/rfc8366>.

[RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 8415, DOI 10.17487/RFC8415, November 2018, <https://www.rfc-editor.org/info/rfc8415>.

[RFC8552] Crocker, D., "Scoped Interpretation of DNS Resource Records through "Underscored" Naming of Attribute Leaves", BCP 222, RFC 8552, DOI 10.17487/RFC8552, March 2019, <https://www.rfc-editor.org/info/rfc8552>.

[Std-802.1AR] IEEE, "IEEE Standard for Local and metropolitan area networks - Secure Device Identity", IEEE 802.1AR.

11.2. Дополнительная литература

[NTS-NTP] Franke, D., Sibold, D., Teichel, K., Dansarie, M., and R. Sundblad, "Network Time Security for the Network Time Protocol", Work in Progress[1], draft-ietf-ntp-using-nts-for-ntp-18, April 2019.

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

[RFC4250] Lehtinen, S. and C. Lonvick, Ed., "The Secure Shell (SSH) Protocol Assigned Numbers", RFC 4250, DOI 10.17487/RFC4250, January 2006, <https://www.rfc-editor.org/info/rfc4250>.

[RFC6187] Igoe, K. and D. Stebila, "X.509v3 Certificates for Secure Shell Authentication", RFC 6187, DOI 10.17487/RFC6187, March 2011, <https://www.rfc-editor.org/info/rfc6187>.

[1]Опубликовано в RFC 8915. *Прим. перев.*

[RFC6234] Eastlake 3rd, D. and T. Hansen, "US Secure Hash Algorithms (SHA and SHA-based HMAC and HKDF)", RFC 6234, DOI 10.17487/RFC6234, May 2011, <https://www.rfc-editor.org/info/rfc6234>.

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

[RFC6335] Cotton, M., Eggert, L., Touch, J., Westerlund, M., and S. Cheshire, "Internet Assigned Numbers Authority (IANA) Procedures for the Management of the Service Name and Transport Protocol Port Number Registry", BCP 165, RFC 6335, DOI 10.17487/RFC6335, August 2011, <https://www.rfc-editor.org/info/rfc6335>.

[RFC6698] Hoffman, P. and J. Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC 6698, DOI 10.17487/RFC6698, August 2012, <https://www.rfc-editor.org/info/rfc6698>.

[RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <https://www.rfc-editor.org/info/rfc6763>.

[RFC6891] Damas, J., Graff, M., and P. Vixie, "Extension Mechanisms for DNS (EDNS(0))", STD 75, RFC 6891, DOI 10.17487/RFC6891, April 2013, <https://www.rfc-editor.org/info/rfc6891>.

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

[RFC7107] Housley, R., "Object Identifier Registry for the S/MIME Mail Security Working Group", RFC 7107, DOI 10.17487/RFC7107, January 2014, <https://www.rfc-editor.org/info/rfc7107>.

[RFC7766] Dickinson, J., Dickinson, S., Bellis, R., Mankin, A., and D. Wessels, "DNS Transport over TCP — Implementation Requirements", RFC 7766, DOI 10.17487/RFC7766, March 2016, <https://www.rfc-editor.org/info/rfc7766>.

[RFC8071] Watsen, K., "NETCONF Call Home and RESTCONF Call Home", RFC 8071, DOI 10.17487/RFC8071, February 2017, <https://www.rfc-editor.org/info/rfc8071>.

[RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, RFC 8259, DOI 10.17487/RFC8259, December 2017, <https://www.rfc-editor.org/info/rfc8259>.

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

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

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

[YANG-CRYPTO-TYPES] Watsen, K. and H. Wang, "Common YANG Data Types for Cryptography", Work in Progress, draft-ietf-netconf-crypto-types-05, March 2019.

[YANG-TRUST-ANCHORS] Watsen, K., "YANG Data Model for Global Trust Anchors", Work in Progress, draft-ietf-netconf-trust-anchors-03, March 2019.

Приложение A. Пример модели данных устройства

В этом приложении представлена ненормативная модель данных, позволяющая настроить начальную загрузку SZTP и определить, какие параметры применять в логике начальной загрузки устройства.

A.1. Обзор модели данных

Приведённое ниже дерево иллюстрирует модель данных устройства в SZTP.

```
module: example-device-data-model
  +--rw sztp
     +--rw enabled?                          boolean
     +--ro idevid-certificate?               ct:end-entity-cert-cms
     |       {bootstrap-servers}?
     +--ro bootstrap-servers {bootstrap-servers}?
     |  +--ro bootstrap-server* [address]
     |     +--ro address    inet:host
     |     +--ro port?      inet:port-number
     +--ro bootstrap-server-trust-anchors {bootstrap-servers}?
     |  +--ro reference*   ta:pinned-certificates-ref
     +--ro voucher-trust-anchors {signed-data}?
        +--ro reference*   ta:pinned-certificates-ref
```

Отметим, что на диаграмме показан лишь один настраиваемый узел - enabled. Предполагается, что для этого узла будет установлено значение true в заданной на производстве конфигурации, принятой по умолчанию, а потом будет установлено значение false или узел будет удалён, когда начальная загрузка SZTP больше не требуется.

A.2. Пример использования

Ниже представлен пример экземпляра для этой модели данных.

```
<sztp xmlns="https://example.com/sztp-device-data-model">
  <enabled>true</enabled>
  <idevid-certificate>base64encodedvalue==</idevid-certificate>
```

```
    <bootstrap-servers>
      <bootstrap-server>
        <address>sztp1.example.com</address>
        <port>8443</port>
      </bootstrap-server>
      <bootstrap-server>
        <address>sztp2.example.com</address>
        <port>8443</port>
      </bootstrap-server>
      <bootstrap-server>
        <address>sztp3.example.com</address>
        <port>8443</port>
      </bootstrap-server>
    </bootstrap-servers>
    <bootstrap-server-trust-anchors>
      <reference>manufacturers-root-ca-certs</reference>
    </bootstrap-server-trust-anchors>
    <voucher-trust-anchors>
      <reference>manufacturers-root-ca-certs</reference>
    </voucher-trust-anchors>
  </sztp>
```

A.3. Модуль YANG

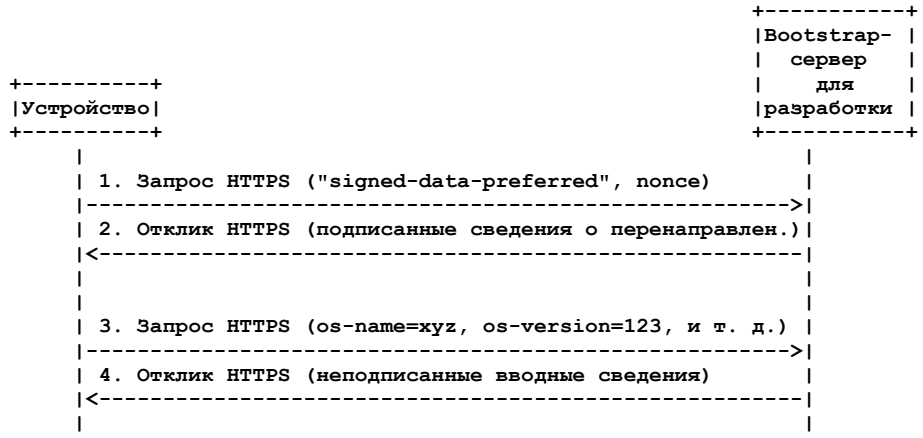
Модель устройства определена модулем YANG, заданным ниже. Модуль ссылается на [Std-802.1AR] и использует типы данных, определённые в [RFC6991], [YANG-CRYPTO-TYPES] и [YANG-TRUST-ANCHORS].

```
  module example-device-data-model {
    yang-version 1.1;
    namespace "https://example.com/sztp-device-data-model";
    prefix sztp-ddm;

    import ietf-inet-types {
      prefix inet;
      reference "RFC 6991: Common YANG Data Types";
    }
    import ietf-crypto-types {
      prefix ct;
      revision-date 2019-03-09;
      description
       "ietf-crypto-types определёно в
        draft-ietf-netconf-crypto-types";
      reference
       "draft-ietf-netconf-crypto-types-05:
          Common YANG Data Types for Cryptography";
    }
    import ietf-trust-anchors {
      prefix ta;
      revision-date 2019-03-09;
      description
       "ietf-trust-anchors определено в
        draft-ietf-netconf-trust-anchors.";
      reference
       "draft-ietf-netconf-trust-anchors-03:
          YANG Data Model for Global Trust Anchors";
    }

    organization
      "Название компании";

    contact
      "Author: Bootstrap Admin <mailto:admin@example.com>";
    description
      "Этот модуль определяет модель данных для поддержки начальной
       загрузки SZTP и обнаружения применяемых параметров. Модуль
       предполагает применение сертификата IDevID, а не иных
       сертификатов клиента, или использование схемы аутентификации
       клиента на основе HTTP.";

    revision 2019-04-30 {
      description
        "Initial version";
      reference
        "RFC 8572: Secure Zero Touch Provisioning (SZTP)";
    }

    // Свойства
    feature bootstrap-servers {
      description
        "Устройство поддерживает начальную загрузку
         с bootstrap-серверов.";
    }

    feature signed-data {
      description
        "Устройство поддерживает начальную загрузку
         подписанных данных.";
```

```
    }

    // Доступные протоколу узлы
    container sztp {
      description
        "Контейнер верхнего уровня для модели данных SZTP.";
      leaf enabled {
        type boolean;
        default false;
        description
          "Лист enabled контролирует возможность начальной загрузки
           SZTP. По умолчанию задано false, поэтому без включения
           (true) конфигурация не требуется.";
      }
      leaf idevid-certificate {
        if-feature bootstrap-servers;
        type ct:end-entity-cert-cms;
        config false;
        description
          "Эта структура CMS содержит сертификат IEEE 802.1AR IDevID
           и все промежуточные сертификаты, ведущие (с возможным
           включением) к сертификатам общеизвестных привязок доверия от
           изготовителя для сертификатов IDevID. Общеизвестная привязка
           доверия не имеет самоподписанного сертификата.";
        reference
          "IEEE 802.1AR:
             IEEE Standard for Local and metropolitan area
             networks - Secure Device Identity";
      }
      container bootstrap-servers {
        if-feature bootstrap-servers;
        config false;
        description
          "Список bootstrap-серверов, которых устройство будет пытаться
           достичь для начальной загрузки.";
        list bootstrap-server {
          key "address";
          description
            "Запись для bootstrap-сервера.";
          leaf address {
            type inet:host;
            mandatory true;
            description
              "IP-адрес или имя хоста для сервера начальной загрузки,
               куда устройство следует перенаправить.";
          }
          leaf port {
            type inet:port-number;
            default "443";
            description
              "Номер порта на сервере начальной загрузки. По умолчанию
               принят выделенный IANA порт https (443).";
          }
        }
      }
      container bootstrap-server-trust-anchors {
        if-feature bootstrap-servers;
        config false;
        description "Контейнер для списка ссылок на привязки доверия.";
        leaf-list reference {
          type ta:pinned-certificates-ref;
          description
            "Ссылка на список сертификатов закреплённых (CA), которые
             устройство применяет для проверки bootstrap-серверов.";
        }
      }
      container voucher-trust-anchors {
        if-feature signed-data;
        config false;
        description "Контейнер для списка ссылок на привязки доверия.";
        leaf-list reference {
          type ta:pinned-certificates-ref;
          description
            "Ссылка на список  сертификатов закреплённых (CA), которые
             устройство применяет для проверки ваучеров владения.";
        }
      }
    }
  }
```

Приложение B. Перевод недоверенного соединения в доверенное

На приведённом ниже рисунке представлена последовательность действий при начальной загрузке, которые переводят недоверенное соединение с bootstrap-сервером в доверенное с тем же сервером. Это позволяет устройству ограничить объем информации, раскрываемой злоумышленнику, у которого размещён недоверенный сервер начальной загрузки.



Взаимодействия на рисунке описаны ниже.

1. Устройство инициирует недоверенное соединение с сервером начальной загрузки, на что указывает "HTTPS" в двойных кавычках. Это по-прежнему соединение HTTPS, но устройство не может аутентифицировать сертификат TLS bootstrap-сервера. Поскольку устройство не может доверять серверу начальной загрузки, оно передаёт входной параметр signed-data-preferred и может добавить параметр nonce в RPC get-bootstrapping-data. Параметр signed-data-preferred информирует bootstrap-сервер о недоверии устройства, в результате чего устройство может скрывать от сервера некоторые дополнительные входные параметры (например, другие входные параметры, отчёты о выполнении и т. п.). Параметр nonce позволяет bootstrap-серверу динамически получить ваучер владения от MASA, что может быть важно для устройств без надёжных часов.

2. Сервер начальной загрузки видит входной параметр signed-data-preferred и знает, что он может передать неподписанные сведения о перенаправлении или подписанные данные любого типа. В данном случае bootstrap-сервер имеет возможность подписывать данные и выбирает отклик с подписанными сведениями о перенаправлении для защищённого перенаправления устройства обратно к себе, а не подписанными вводными данными, как можно было ожидать. На рисунке это не показано, но если был передан входной параметр nonce, сервер начальной загрузки может динамически связаться с MASA и загрузить ваучер со значением nonce. Детали протокола, обеспечивающего такую интеграцию, выходят за рамки документа.

3. После проверки подписанных сведений о перенаправлении устройство организует защищённое соединение с bootstrap-сервером. Устройство не знает, тот ли это bootstrap-сервер, с которым оно соединялось до этого, но поскольку оно может на этот раз аутентифицировать сервер, оно передаёт свой обычный запрос get-bootstrapping-data (с дополнительными входными параметрами), а также отчёт о выполнении (не показан).

4. На этот раз, поскольку параметр signed-data-preferred не был передан, имея доступ ко всем входным параметрам, bootstrap-сервер возвращает в этом примере неподписанные вводные сведения для устройства. Отметим также, что сервер начальной загрузки стал доверенным и устройство может передавать ему отчёты о выполнении.

Приложение C. Обзор рабочего процесса

Представленное в документе решение концептуально состоит из ненормативных рабочих процессов, описанных ниже. Предполагается, что детали реализаций будут различаться. Каждый из рисунков сопровождается подробным описанием представленных на рисунке действий с комментариями по части различий в реализациях.

C.1. Зачисление и упорядочение устройств

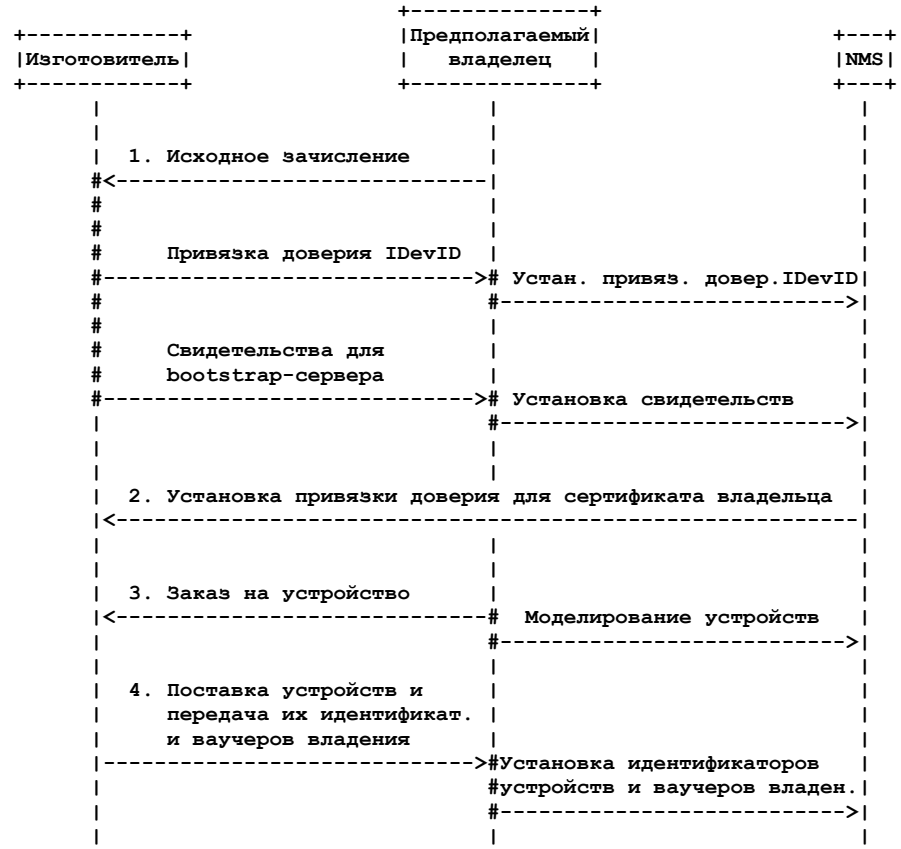
На рисунке представлены основные взаимодействия, которые могут происходить с момента, когда потенциальный владелец регистрируется в программе SZTP изготовителя, до момента, когда изготовитель поставляет устройства по заказу потенциального владельца.

1. Потенциальный владелец устройства инициирует процесс регистрации у изготовителя.

   - Независимо от намерений потенциального владельца в части начальной загрузки его устройств, он всегда будет получать от изготовителя сертификат привязки доверия для сертификатов IDevID. Этот сертификат устанавливается в NMS потенциального владельца, чтобы NMS могла проверить подлинность сертификатов IDevID на последующих этапах.

   - Если изготовитель поддерживает свой сервер начальной загрузки через Internet (например, сервер перенаправления), как описано в параграфе 4.4. Сервер начальной загрузки, свидетельства, требуемые для настройки bootstrap-сервера будут предоставлены потенциальному владельцу. Если bootstrap-сервер настраивается через API (не рассматривается в этом документе), свидетельства могут быть установлены в системе NMS потенциального владельца, чтобы NMS впоследствии могла настроить размещённый у производителя bootstrap-сервер.

2. Если устройства изготовителя способны проверять подписанные данные (5.4. Проверка подписанных данных) и в предположении, что NMS потенциального владельца способна самостоятельно подготовить и подписать данные начальной загрузки, эта NMS может установить сертификат привязки доверия для bootstrap-сервера изготовителя, используя свидетельства, предоставленные на предыдущем этапе. Этот сертификат является сертификатом привязки доверия, которых потенциальный владелец хотел бы получить от производителя в ваучерах владения, которые тот создаст, чтобы устройства могли доверять сертификату владельца. Способ применения сертификата привязки доверия для того, чтобы разрешить устройствам проверять подписанные данные начальной загрузки, описан в параграфе 5.4. Проверка подписанных данных.



3. Через некоторое время потенциальный владелец размещает заказ у изготовителя, возможно, со специальным флагом для обработки SZTP. В этот момент или перед размещением заказа владелец может смоделировать устройства в своей NMS, создавая виртуальные объекты для устройств без привязки к реальным устройствам. Например, модель можно использовать для имитации размещения устройств в сети и конфигурации, которую следует задать для полной работоспособности.

4. Когда изготовитель выполнит заказ, поставив устройства в предусмотренные места, он может сообщить владельцу серийные номера и точки доставки, которые владелец может использовать для подготовки сети к включению устройств. В дополнение к этому изготовитель может передать один или несколько ваучеров владения, криптографически передав владельцу права собственности на устройства. Владелец может поместить эти сведения в свою NMS, возможно, связав смоделированные устройства с серийными номерами и ваучерами владения.

C.2. Этапы подготовки сети владельцем для начальной загрузки

На рисунке показано, как владелец может подготовить сеть для устройств с начальной загрузкой.

1. Имея ранее смоделированные устройства, включая установку для них полностью работоспособных конфигураций и связанных с устройствами серийных номеров, а также (необязательно) ваучеров владения, владелец может «активировать» одно или несколько смоделированных устройств. Т. е. владелец предписывает NMS выполнить этапы, требуемые для подготовки реальных устройств к включению питания и запуску процесса начальной загрузки. Отметим, что в некоторых развёртываниях этот этап может объединяться с последним этапом предыдущего рабочего процесса. Здесь отмечено, что NMS выполняет эти этапы, но они могут быть выполнены вручную или с помощью иного механизма.

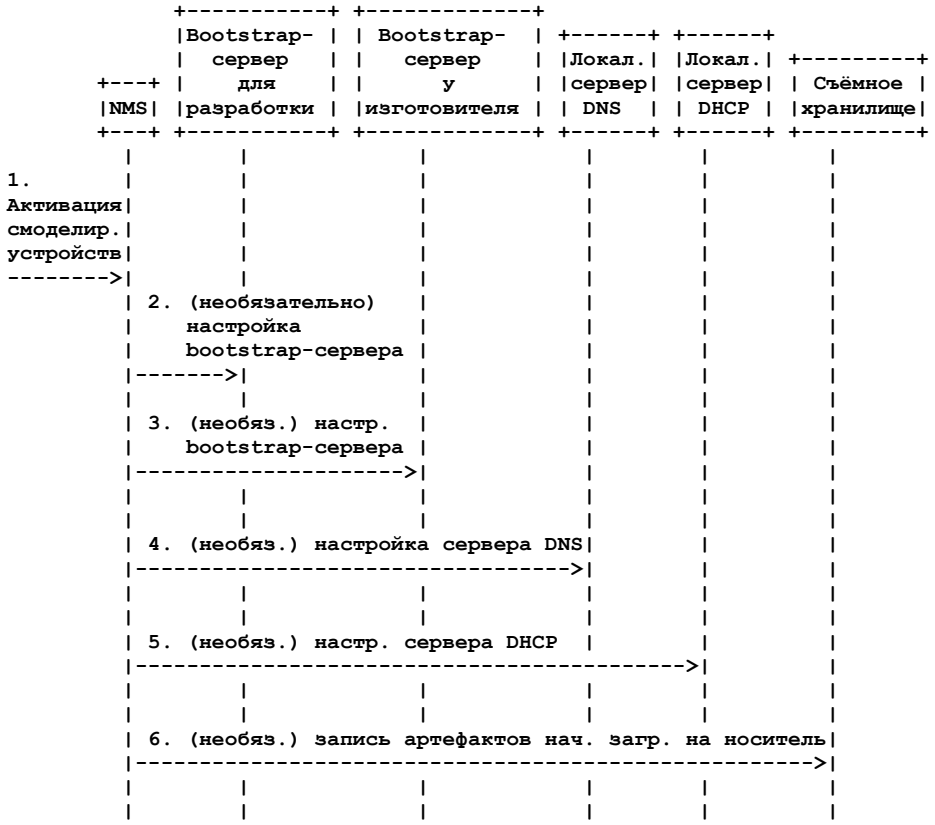
2. Если желательно использовать bootstrap-сервер конкретного развёртывания, он должен быть настроен на предоставление начальной загрузки конкретным устройствам. Настройка bootstrap-сервера может выполняться через программный интерфейс API, не определённый в этом документе. Показанный на рисунке как внешний узел, сервер начальной загрузки может быть реализован как внутренний компонент NMS.

3. Если желательно использовать bootstrap-сервер изготовителя, он должен быть настроен на предоставление начальной загрузки конкретным устройствам. Это должны быть данные перенаправления или вводные сведения, т. е. сервер начальной загрузки у производителя будет перенаправлять устройство на другой bootstrap-сервер или сам предоставлять вводные данные. Типы данных начальной загрузки, поддерживаемые bootstrap-сервером у производителя, могут зависеть от реализации и некоторые реализации могут поддерживать только сведения о перенаправлении или вводные данные. Настройка bootstrap-сервера может выполняться через программный интерфейс API, не определённый в этом документе.

4. Если желательно использовать сервер DNS для предоставления данных начальной загрузки, этот сервер должен быть настроен. Если желательно использовать multicast DNS, сервер DNS должен размещаться в локальной сети, в иных случаях можно пользоваться внешним сервером DNS. Настройка серверов DNS рассмотрена в параграфе 4.2. Сервер DNS. Настройка сервера DNS может выполняться через программный интерфейс API, не определённый в этом документе.

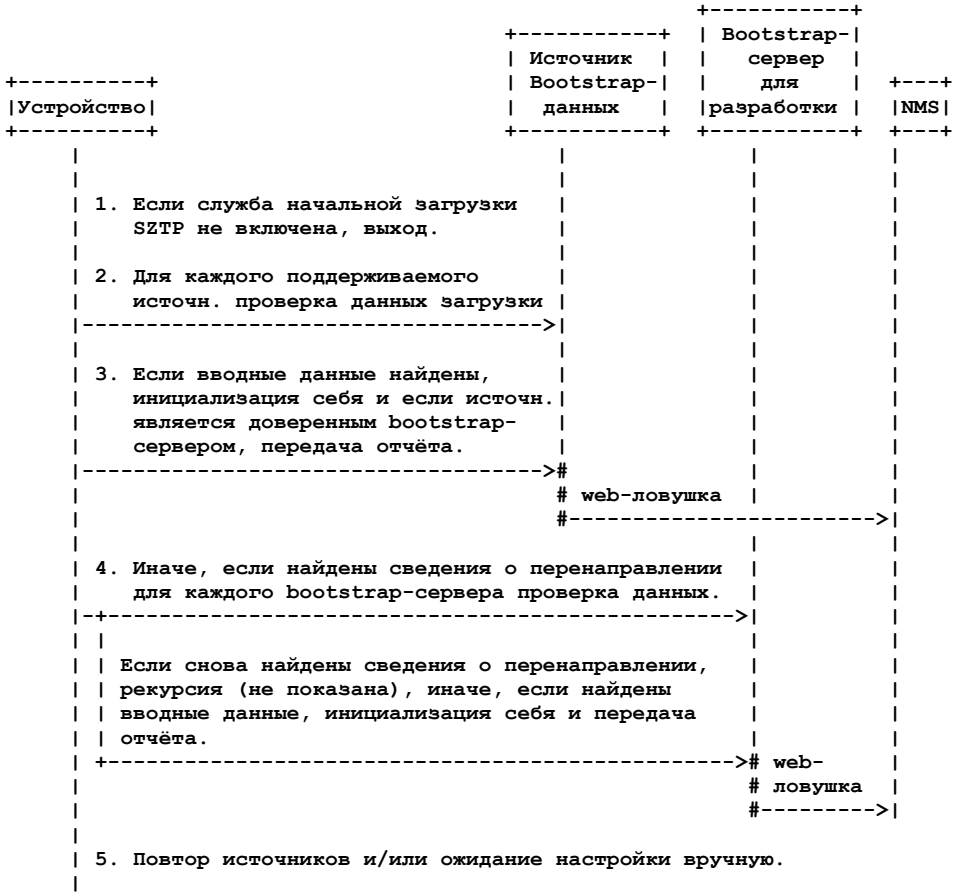
5. Если желательно использовать сервер DHCP для предоставления данных начальной загрузки, этот сервер должен быть настроен. Доступ к серверу DHCP возможен напрямую или через ретранслятор (DHCP relay). Настройка серверов DHCP рассмотрена в параграфе 4.3. Сервер DHCP. Настройка сервера DHCP может выполняться через программный интерфейс API, не определённый в этом документе.

6. Если желательно использовать съёмное устройство хранения (например, USB flash) для предоставления данных начальной загрузки, это устройство должно быть подключено. Настройка для съёмных устройств рассмотрена в параграфе 4.1. Сменный носитель.



C.3. Включение устройства

На рисунке показана последовательность действий при включении питания устройства.



1. При включении питания устройство проверяет, настроена ли начальная загрузка SZTP, как это должно быть в принятой по умолчанию заводской конфигурации. Если начальная загрузка SZTP не настроена, bootstrap-логика завершается без выполнения следующих этапов.

2. Для каждого источника данных начальной загрузки, поддерживаемых устройством в порядке предпочтений (например, сначала съёмный носитель, потом серверы Internet), устройство проверяет наличие данных начальной загрузки для него.

3. Если найдены вводные данные, устройство должным образом инициализирует себя (например, устанавливает загрузочный образ или применяет исходную конфигурацию). Если источником является bootstrap-сервер и ему можно доверять (аутентификация TLS), устройство также передаёт отчёт о выполнении bootstrap-серверу.

   - В исходной конфигурации следует настроить учётную запись администратора на устройстве (например, имя пользователя, открытый ключ SSH и т. п.), задать прослушивание соединений NETCONF или RESTCONF или инициирование исходящих звонков [RFC8071], а также отключить службу начальной загрузки SZTP (например, лист enabled в модели данных из Приложения A).

   - Если сервер начальной загрузки поддерживает пересылку отчётов о выполнении от устройства (например, через web-ловушку), отчёт bootstrap-complete (7.3. Модуль YANG) информирует внешнюю систему о том, когда она сможет, например, инициировать соединение с устройством. Для дальнейшей поддержки такого сценария отчёт bootstrap-complete может также ретранслировать SSH-ключи хоста и/или сертификаты TLS, которые внешняя система может использовать для аутентификации последующих соединений с устройством.

   - Если устройство успешно завершает процесс начальной загрузки, оно выходит из bootstrap-логики без рассмотрения других источников данных начальной загрузки.

4. В иных случаях, если найдены сведения о перенаправлении, устройство проходит по списку заданных bootstrap-серверов, проверяя у них наличие данных для начальной загрузки устройства. Если это снова данные перенаправления, рекурсия продолжается. В иных случаях, если bootstrap-сервер возвращает вводные данные, устройство выполняет действия, описанные в п. 3 выше.

5. Проверив все поддерживаемые источники данных начальной загрузки, устройство может повторить попытку для всех источников и/или предоставить интерфейс управления для настройки вручную (например, CLI, HTTP, NETCONF и т. п.). Если разрешена ручная настройка и конфигурация представлена, эта конфигурация должна отключить службу начальной загрузки SZTP, поскольку та больше не требуется.

Благодарности

Авторы благодарны за оживлённые дискуссии по почте и на встречах (в алфавитном порядке) Michael Behringer, Martin Bjorklund, Dean Bogdanovic, Joe Clarke, Dave Crocker, Toerless Eckert, Stephen Farrell, Stephen Hanna, Wes Hardaker, David Harrington, Benjamin Kaduk, Radek Krejci, Suresh Krishnan, Mirja Kuehlewind, David Mandelberg, Alexey Melnikov, Russ Mundy, Reinaldo Penno, Randy Presuhn, Max Pritikin, Michael Richardson, Adam Roach, Juergen Schoenwaelder, Phil Shafer.

Особая благодарность Steve Hanna, Russ Mundy и Wes за мозговой штурм исходного решения во время IETF 87 в Берлине.

Адреса авторов

Kent Watsen
Watsen Networks
Email: kent+ietf@watsen.net

Ian Farrer
Deutsche Telekom AG
Email: ian.farrer@telekom.de

Mikael Abrahamsson
T-Systems
Email: mikael.abrahamsson@t-systems.se

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru