Network Working Group
Request For Comments: 3080
Category: Standards Track

M. Rose
Invisible Worlds, Inc.

March 2001

Ядро расширяемого протокола обмена блоками (BEEP)

The Blocks Extensible Exchange Protocol Core

Статус документа

Данный документ содержит спецификацию протокола, предложенного сообществу Internet, и служит запросом к дискуссии в целях развития протокола. Информацию о статусе данного протокола можно найти в текущей редакции документа "Internet Official Protocol Standards" (STD 1). Документ может распространяться без ограничений.

Тезисы

Этот документ описывает ядро базового прикладного протокола для основанных на соединении асинхронных коммуникаций, названного ядром BEEP[1]. BEEP поддерживает одновременный и независимый обмен в контексте одного отождествления пользователя с использованием текстовых и двоичных сообщений.

Оглавление

[1]Blocks Extensible Exchange Protocol — расширяемый протокол обмена блоками.

1. Введение

Этот документ описывает ядро базового прикладного протокола для основанного на соединении асинхронного обмена сообщениями, названного BEEP.

Ядро BEEP включает механизм кадрирования, обеспечивающий одновременный и независимый обмен сообщениями между узлами-партнерами. Сообщения включают произвольное содержимое MIME [1], но обычно являются текстовыми (структурированными на основе XML [2]).

Весь обмен сообщениями происходит в контексте канала — привязки к четко определенному аспекту априложения типа транспортной безопасности, аутентификации пользователя или обмена данными.

С каждым каналом связан «профиль», определяющий синтаксис и семантику участвующих в обмене сообщений. В работе BEEP нотация управления каналом является неявной. В дополнение к определению профиля управления каналом BEEP данный документ включает:

- профиль транспортной защиты TLS [3];
- семейство профилей SASL [4].

Другие профили (например, используемые при обмене данными) определяются разработчиками прикладных протоколов.

2. Ядро BEEP

Сессия BEEP отображается на базовый (нижележащий) транспортный сервис. Отдельный набор документов описывает реализацию сеансов BEEP на основе конкретных транспортных протоколов. Например, в документе [5] описано отображение сессии BEEP в одно соединение TCP [6].

При организации сессии каждый узел BEEP анонсирует поддерживаемые им профили. Позднее, в процессе создания канала клиент предлагает для этого канала один или множество профилей. Если сервер создает канал, он выбирает одие из профилей и указывает его в отклике, в противной случае он может указать, что ни один из профилей не подходит и отвергнуть создание канала.

Использование канала делится на два этапа (категории).

Начальная настройка применяется профилями, которые выполняют инициализацию при организации сессии BEEP (например, согласование транспортной защиты). Хотя для выполнения инициализации могут потребоваться некоторые обмены, эти каналы становятся неактивными в начале сеанса BEEP и сохраняются в таком состоянии во время сеанса.

Постоянное использование профилями, которые поддерживают обмен данными. Эти каналы создаются после начальной настройки.

Отметим, что в силу самой его природы канал настройки в каждый момент времени может быть только один, а каналов передачи данных в протоколе BEEP может использоваться одновременно более одного.

2.1. Роли

Хотя протокол BEEP относится к числу одноранговых (peer-to-peer), удобно помечать каждого узла в контексте выполняемой этим узлом в данный момент роли.

- При организации сессии BEEP узел, ожидающий новых соединений играет роль «слушателя», а другой узел, который организует соединение со слушателем, — роль «инициатора». Ниже в примерах они обозначены L: и I:, соответственно.
- Узел BEEP, начинающий обмен, считается клиентом, другой узел — сервером. В последующих примерах они обозначаются C: и S:, соответственно.

Обычно узел BEEP в роли сервера также играет роль слушателя. Однако по причине одноранговой природы BEEP такого требования не задается.

2.1.1. Стили обмена

BEEP поддерживает три стиля обмена.

MSG/RPY — клиент передает сообщение MSG с запросом к серверу выполнить некую задачу. Сервер выполняет запрос и передает сообщение RPY (позитивный отклик).

MSG/ERR — клиент передает сообщение MSG с запросом к серверу выполнить некую задачу. Сервер не выполняет задачу и передает сообщение ERR (негативный отклик).

MSG/ANS — клиент передает серверу сообщение MSG, сервер в процессе выполнения задачи может (но не обязан) ответить сообщением ANS (возможно, несколькими), а по завершении отвечает сообщением NUL, которое говорит о выполнении запроса.

Первые два варианта называются обменами «один к одному», третий - «один ко многим».

2.2. Сообщения и кадры

Сообщения структурируются в соответствии с правилами MIME и каждое сообщение может начинаться с entity-headers (см. раздел 3 в [1]). Если entity-headers отсутствует или имеется только часть, по умолчанию используются:

- Content-Type - application/octet-stream;
- Content-Transfer-Encoding - binary.

Обычно сообщение передается в одном кадре, однако может потребоваться или оказаться удобным сегментирование сообщения на множество кадров (например, если к передаче готова только часть сообщения).

Каждый кадр состоит из заголовка, данных (payload) и трейлера. Заголовок и трейлер представляются печатными строками ASCII и завершаются парой символов CRLF. Между заголовком и трейлером помещается (возможно пустая) последовательность октетов данных.

Например, приведенное ниже сообщение содержится в одном кадре, где 120 октетов данных занимают 5 строк (каждая строка завершается парой символов CRLF).

```
C: MSG 0 1 . 52 120
C: Content-Type: application/beep+xml
C:
C: <start number='1'>
C:    <profile uri='http://iana.org/beep/SASL/OTP' />
C: </start>
C: END
```

Еще раз отметим, что в этом примере сообщение представлено в одном кадре.

2.2.1. Синтаксис кадра

ABNF [7] для кадра имеет вид:

```
frame      = data / mapping

data       = header payload trailer

header     = msg / rpy / err / ans / nul

msg        = "MSG" SP common           CR LF
rpy        = "RPY" SP common           CR LF
ans        = "ANS" SP common SP ansno CR LF
err        = "ERR" SP common           CR LF
nul        = "NUL" SP common           CR LF

common     = channel SP msgno SP more SP seqno SP size
channel    = 0..2147483647
msgno      = 0..2147483647
more       = "." / "*"
seqno      = 0..4294967295
size       = 0..2147483647
ansno      = 0..2147483647

payload    = *OCTET

trailer    = "END" CR LF

mapping    = ;; каждое транспортное отображение может определять свои кадры
```

2.2.1.1. Заголовок кадра

Заголовок кадра состоит из трехсимвольной последовательности (MSG, RPY, ERR, ANS или NUL), за которой могут следовать параметры. В качестве разделителей используются одиночные символы пробела (десятичный код 32, " "). Завершается заголовок парой символов CRLF.

Номер канала (channel) должен быть целым числом из диапазона 0 - 2147483647.

Номер сообщения (msgno) должен быть целым числом из диапазона 0 - 2147483647 и должен отличаться от номеров всех других сообщений MSG в том же канале, для которых отклик еще не получен полностью.

Индикатором продолжения (more) может содержать символ звездочки (десятичный код 42) или точки (десятичный код 46) и показывает является ли этот кадр последним для сообщения:

* - за этим кадром следует по крайней мере еще один кадр данного сообщения;

. - последний кадр сообщения.

Порядковый номер (seqno) должен быть целым числом из диапазона 0 - 4294967295 и указывает порядковый номер первого октета данных (payload) для соответствующего канала (2.2.1.2. Данные кадра).

Размер данных (size) должен указываться целым числом из диапазона 0 - 2147483647 и показывает точное число октетов данных (без учета заголовка и трейлера).

Отметим, что кадр может не включать данных, как показано ниже.

```
S: RPY 0 1 * 287 20
S:     ...
S:     ...
S: END
S: RPY 0 1 . 307 0
S: END
```

Номер ответа (ansno) должен быть целым числом из диапазона 0 - 2147483647[1] и должен отличаться от номеров других ответов на обрабатываемое сообщение.

Имеется два типа кадров — данные (data) и отображения (mapping). Каждое транспортное отображение (2.5. Транспортные отображения) может определять свои кадры. Например, в документе [5] определен кадр SEQ. В оставшейся части этого параграфа рассматриваются кадры данных.

Когда сообщение сегментируется и передается в нескольких кадрах, эти кадры должны передаваться последовательно без промежуточных кадров с другими сообщениями в том же канале. Однако есть два исключения — во-первых, не накладывается ограничений на чередование с кадрами для других каналов, во-вторых, при обмене one-to-many одновременно может обрабатываться множество запросов. Следовательно, кадры для сообщений ANS могут чередоваться в одном канале, а для их сортировки используются номера ответов, как показано ниже.

```
S: ANS 1 0 * 0 20 0
S:     ...
S:     ...
S: END
S: ANS 1 0 * 20 20 1
S:     ...
S:     ...
S: END
S: ANS 1 0 . 40 10 0
S:     ...
S: END
```

Здесь два сообщения ANS чередуются в канале 1, как часть отклика на сообщение с номером 0. Отметим, что порядковые номера увеличиваются в каждом кадре и это не зависит от передаваемых в этих кадрах сообщений.

Существует несколько признаков некорректных кадров:

- заголовок не начинается с MSG, RPY, ERR, ANS или NUL;
- параметры в заголовке не могут быть определены или являются непригодными (например, синтаксически некорректны);
- значение номера канала не указывает существующий канал;
- заголовок начинается с MSG, а номер сообщения указывает на сообщение MSG, которое было принято полностью, но отклик не был полностью передан;
- заголовок не начинается с MSG и указан номер сообщения, для которого отклик уже получен полностью;
- заголовок не начинается с MSG и указан номер сообщения, которое никогда не передавалось (за исключением организации сеанса, 2.3.1.1. Сообщение Greeting);
- заголовок начинается с MSG, RPY, ERR или ANS и указан номер сообщения, для которого был получен по меньшей мере один кадр, а трехсимвольные последовательности в начале кадра и полученного непосредственно перед этим сообщением кадра не совпадают;
- заголовок начинается с NUL и указан номер сообщения для которого был получен по меньшей мере еще один кадр, а в заголовке непосредственно предшествующего тип кадра указано значение, отличное от ANS;
- индикатор продолжения в предыдущем кадре этого канала имеет значение * (есть еще кадры), а номер сообщения в нем отличается от номера в текущем кадре;
- значение порядкового номера не соответствует ожидаемому для связанного канала (2.2.1.2. Данные кадра);
- заголовок начинается с NUL, а индикатор продолжения имеет значение * (есть еще кадры) или размер данных отличается от 0.

Если кадр имеет некорректную форму, сессия прерывается без генерации отклика. В целях диагностики рекомендуется делать запись об этом в системном журнале.

2.2.1.2. Данные кадра

Данные (payload) кадра представляют собой последовательность октетов (возможно, пустую).

С каждым октетом данных, передаваемым через канал в определенном направлении, связывается порядковый номер. Первый октет в каждом кадре имеет наименьшее значение номера, а для остальных номера возрастают последовательно. При создании канала для первого октета данных устанавливается номер 0.

[1] В оригинале ошибочно указано значение 4294967295. См. https://www.rfc-editor.org/errata_search.php?eid=992. *Прим. перев.*

Доступное пространство порядковых номеров конечно и представляет собой диапазон значений от 0 до 4294967295 (2^{32} - 1). По причине конечных размеров пространства работа с порядковыми номерами выполняется на основе модуля 2^{32} и при достижении максимального номера отсчет продолжается с 0. Арифметика порядковых номеров рассмотрена в разделах 2 - 5 работы [8].

При получении кадра сумма порядкового номера и размера данных по модулю 2^{32} дает ожидаемое значение порядкового номера первого октета данных в следующем кадре. Поэтому получение в следующем кадре иного порядкового номера говорит о потере синхронизации между партнерами BEEP, что ведет к разрыву сессии. В целях диагностики рекомендуется делать запись об этом в системном журнале.

2.2.1.3. Трейлер кадра

Трейлером кадра служит строка END, за которой следует пара символов CRLF.

При получении кадра, в котором после данных следуют символы, отличные от трейлера, сессия разрывается без генерации отклика. В целях диагностики рекомендуется делать запись об этом в системном журнале.

2.2.2. Семантика кадров

Семантика каждого сообщения зависит от канала, поэтому связанный с каналом профиль должен определять:

- инициализационные сообщения (если таковые используются) для обмена при создании канала;
- сообщения, которые могут передаваться в данных (payload) канала;
- семантику этих сообщений.

Организация этих данных описана в шаблоне регистрации профиля (5. Регистрационные шаблоны).

2.2.2.1. Некорректно сформированные сообщения

При определении поведения профиля шаблон должен задавать отклики на сообщения MSG некорректной формы. Например, профиль управления каналом передает негативный отклик, включающий сообщение об ошибке (2.3.1.5. Сообщение Error).

При получении некорректно сформированного отклика на канале 0 сессия разрывается без генерации отклика. В целях диагностики рекомендуется делать запись об этом в системном журнале.

Если некорректно сформированный отклик получен на другом канале, этот канал должен быть закрыт с использованием процедуры, описанной в параграфе 2.3.1.3. Сообщение Close.

2.3. Управление каналом

В момент организации сессии BEEP определен лишь канал 0, используемый для управления. В параграфе 6.1. Управление каналом BEEP приведена регистрация профиля для управления каналами BEEP.

Управление каналом позволяет каждому узлу BEEP анонсировать поддерживаемые им профили (2.3.1.1. Сообщение Greeting), привязать один из экземпляров этих профилей к каналу (2.3.1.2. Сообщение Start), а впоследствии закрыть все каналы и сессию BEEP (2.3.1.3. Сообщение Close).

Узлам BEEP следует поддерживать не менее 257 одновременных каналов.

2.3.1. Семантика сообщений

2.3.1.1. Сообщение Greeting

При организации сессии BEEP каждый из узлов указывает свои возможности, незамедлительно передавая позитивный отклик с номером сообщения 0, содержащий элемент greeting.

```
L: <ожидание входящего соединения>
I: <организация соединения>
L: RPY 0 0 . 0 110
L: Content-Type: application/beep+xml
L:
L: <greeting>
L:    <profile uri='http://iana.org/beep/TLS' />
L: </greeting>
L: END
I: RPY 0 0 . 0 52
I: Content-Type: application/beep+xml
I:
I: <greeting />
I: END
```

Из этого примера можно предположить, что узел BEEP в роли инициатора ждет от «слушателя» BEEP его «поздравления» (greeting), однако это лишь «артефакт» представления и на практике оба партнера могут передавать свои отклики одновременно.

Элемент greeting имеет два необязательных атрибута (features и localize) и может (не обязательно) включать элементы profile - по одному для каждого поддерживаемого партнером-сервером профиля.

- При наличии атрибута features он содержит один или множество маркеров функций (feature token), каждый из которых указывает дополнительную возможность профиля управления каналом, поддерживаемого узлом BEEP.
- При наличии атрибута localize он включает один или множество маркеров языка (language token), определенных в работе [9], каждый из которых указывает желаемый язык, который удаленный партнер будет

использовать при генерации текстовых сообщений для элементов close и error (маркеры упорядочиваются по снижению уровня предпочтения).

- Каждый элемент profile внутри элемента greeting указывает профиль и, в отличие от элементов profile внутри элемента start, может не включать инициализационного сообщения.

Регистрационные шаблоны для необязательных функций определены в параграфе 5.2. Шаблон регистрации функции

2.3.1.2. Сообщение Start

Когда узел BEEP желает организовать канал, он передает элемент start через канал 0.

```
C: MSG 0 1 . 52 120
C: Content-Type: application/beep+xml
C:
C: <start number='1'>
C:    <profile uri='http://iana.org/beep/SASL/OTP' />
C: </start>
C: END
```

Элемент start имеет атрибут number, необязательный атрибут serverName и один или множество элементов profile.

- Атрибут number указывает номер канала (от 1 до 2147483647), используемый для идентификации канала в будущих сообщениях.
- Атрибут serverName (произвольная строка) указывает желаемое имя сервера для этой сессии BEEP.
- Каждый элемент profile, содержащийся в элементе start, имеет атрибут uri, необязательный атрибут encoding и произвольные символы в качестве содержимого:
  - атрибут uri достоверно указывает профиль;
  - при наличии атрибута encoding он показывает, представлено ли содержимое элемента profile в форме строки base64;
  - при наличии содержимого элемента profile оно должно быть не больше 4K октетов и задает инициализационное сообщение при создании канала.

Для предотвращения конфликтов порядковых номеров каналов в запросах на создание канала узлы-инициаторы указывают только нечетные положительные целые числа, а узлы-слушатели — четные положительные значения.

Атрибут serverName для первого элемента start, успешно принятого узлом BEEP, сохраняет значимость на протяжении всей сессии BEEP. Узел BEEP решает, будет ли он работать в качестве указанного serverName и при отказе передает элемент error в негативном отклике.

При получении узлом BEEP элемента start по каналу 0, он проверяет каждый из предложенных профилей и принимает решение об использовании одного из них для создания канала. При положительном решении соответствующий элемент profile передается в позитивном отклике, в противном случае передается негативный отклик с элементом error.

При создании канала значение serverName из первого «успешного» элемента start просматривается на предмет наличия конфигурационных данных, например желаемого сертификата сервера при старте транспортного профиля TLS (3.1. Профиль транспортной защиты TLS).

Пример успешного создания канала показан ниже.

```
C: MSG 0 1 . 52 178
C: Content-Type: application/beep+xml
C:
C: <start number='1'>
C:    <profile uri='http://iana.org/beep/SASL/OTP' />
C:    <profile uri='http://iana.org/beep/SASL/ANONYMOUS' />
C: </start>
C: END
S: RPY 0 1 . 221 87
S: Content-Type: application/beep+xml
S:
S: <profile uri='http://iana.org/beep/SASL/OTP' />
S: END
```

Неудача при создании канала может выглядеть, как показано ниже.

```
C: MSG 0 1 . 52 120
C: Content-Type: application/beep+xml
C:
C: <start number='2'>
C:    <profile uri='http://iana.org/beep/SASL/OTP' />
C: </start>
C: END
S: ERR 0 1 . 221 127
S: Content-Type: application/beep+xml
S:
S: <error code='501'>number attribute
S: in &lt;start&gt; element must be odd-valued</error>
S: END
```

В заключение приведем пример с обменом инициализационными сообщениями в процессе создания канала.

```
C: MSG 0 1 . 52 158
C: Content-Type: application/beep+xml
C:
C: <start number='1'>
C:    <profile uri='http://iana.org/beep/TLS'>
C:        <![CDATA[<ready />]]>
C:    </profile>
C: </start>
C: END
S: RPY 0 1 . 110 121
S: Content-Type: application/beep+xml
S:
S: <profile uri='http://iana.org/beep/TLS'>
S:     <![CDATA[<proceed />]]>
S: </profile>
S: END
```

2.3.1.3. Сообщение Close

Узел BEEP, желающий закрыть канал, передает элемент close через канал 0.

```
C: MSG 0 2 . 235 71
C: Content-Type: application/beep+xml
C:
C: <close number='1' code='200' />
C: END
```

Элемент close имеет атрибуты number и code, необязательный атрибут xml:lang и необязательный текст для диагностики в качестве своего содержимого.

- Атрибут number указывает номер закрываемого канала.
- Атрибут code содержит трехзначный числовой код отклика, имеющий смысл для программ (8. Коды откликов).
- Атрибут xml:lang указывает язык содержимого элемента (значение предлагается, но не задается в качестве обязательного атрибутом localize в элементе greeting, переданном удаленным партнером BEEP).
- Текстовое сообщение (может включать много строк) имеет значимость для разработчиков, администраторов и, возможно, пользователей, но не для программ.

Отметим, что при наличии диагностического текста атрибут xml:lang не указывается лишь в том случае, когда используется язык, указанный первым удаленным партнером BEEP.

Если атрибут number имеет значение 0, это говорит о желании узла BEEP закрыть сессию (2.4. Организация и завершение сессии), а все другие значения number указывают на существующий канал, к которому применяется оставшаяся часть данного параграфа.

Узел BEEP может передать сообщение close для канала, когда все сообщения MSG, переданные им в этот канал, были подтверждены (подтверждением служит первый кадр, принятый узлом BEEP, который отправил сообщение MSG).

После передачи сообщения close этот узел BEEP должен прекратить передачу каких-либо сообщений MSG через закрываемый канал, пока не будет получен отклик на сообщение close (сообщение error с отказом от закрытия канала или сообщение ok, говорящее о начале процедуры закрытия).

Примечание. Пока не будет получен позитивный отклик на запрос закрытия канала, узел BEEP должен быть готов к обработке всех сообщений MSG, принимаемых по этому каналу.

Когда узел BEEP получает сообщение close для канала, он может в любой момент отвергнуть запрос на закрытие, передав сообщение error в негативном отклике.

В иных случаях до восприятия запроса на закрытие канала и передачи позитивного отклика ok этот узел должен:

- завершить передачу находящихся в очереди сообщений MSG по этому каналу;
- дождаться завершения откликов на остающиеся сообщения MSG, переданные им в этот канал;
- завершить передачу откликов на все остающиеся сообщения MSG, которые он получил по этому каналу и убедиться в успешной доставке финальных кадров этих откликов:
  - для транспортных отображений с гарантией упорядочения в канале отклики должны быть переданы до отправки сообщения ok в позитивном отклике;
  - в остальных случаях отклики должны быть переданы и подтверждены нижележащим транспортом до отправки сообщения ok в позитивном отклике.

Кратко успешное закрытие канала может выглядеть, как показано ниже.

```
C: MSG 0 2 . 235 71
C: Content-Type: application/beep+xml
C:
C: <close number='1' code='200' />
C: END
S: RPY 0 2 . 392 46
S: Content-Type: application/beep+xml
S:
S: <ok />
S: END
```

Отказ при закрытии канала может иметь вид:

```
C: MSG 0 2 . 235 71
C: Content-Type: application/beep+xml
C:
C: <close number='1' code='200' />
C: END
S: ERR 0 2 . 392 79
S: Content-Type: application/beep+xml
S:
S: <error code='550'>still working</error>
S: END
```

2.3.1.4. Сообщение OK

Когда узел BEEP согласен закрыть канал (или завершить сессию BEEP), он передает элемент ok в позитивном отклике.

Элемент ok не включает атрибутов или содержимого.

2.3.1.5. Сообщение Error

Когда узел BEEP отвергает создание канала, он передает элемент error в негативном отклике.

```
I: MSG 0 1 . 52 115
I: Content-Type: application/beep+xml
I:
I: <start number='2'>
I:    <profile uri='http://iana.org/beep/FOO' />
I: </start>
I: END
L: ERR 0 1 . 221 105
L: Content-Type: application/beep+xml
L:
L: <error code='550'>all requested profiles are
L: unsupported</error>
L: END
```

Элемент error имеет атрибут code, необязательный атрибуто xml:lang и необязательное текстовое поле в качестве содержимого.

- Атрибут code содержит трехзначный числовой код отклика, имеющий смысл для программ (8. Коды откликов).
- Атрибут xml:lang указывает язык содержимого элемента (значение предлагается, но не задается в качестве обязательного атрибутом localize в элементе greeting, переданном удаленным партнером BEEP).
- Текстовое сообщение (может включать много строк) имеет значимость для разработчиков, администраторов и, возможно, пользователей, но не для программ.

Отметим, что при наличии диагностического текста атрибут xml:lang не указывается лишь в том случае, когда используется язык, указанный первым удаленным партнером BEEP.

В дополнение к сказанному узел BEEP передает элемент error всякий раз, когда:

- он получает сообщение MSG с некорректно сформированным или неожиданным элементом;
- он получает сообщение MSG, запрашивающее закрытие канала (или сессии BEEP), но узел отвергает закрытие;
- сессия BEEP организована, а узел выступающий в роли слушателя BEEP, не доступен (слушатель BEEP не передал элемент greeting).

В последнем случае оба партнера BEEP разрывают сеанс и рекомендуется записывать это событие в системные журналы обоих партнеров BEEP.

2.4. Организация и завершение сессии

Когда сессия BEEP организована, каждый из партнеров BEEP обозначает свою доступность путем передачи позитивного отклика с номером сообщения 0 и элементом greeting, как показано ниже.

```
L: <ожидание входящего соединения>
I: <создание соединения>
L: RPY 0 0 . 0 110
L: Content-Type: application/beep+xml
L:
L: <greeting>
L:    <profile uri='http://iana.org/beep/TLS' />
L: </greeting>
L: END
I: RPY 0 0 . 0 52
I: Content-Type: application/beep+xml
I:
I: <greeting />
I: END
```

Если же узел, выступающий в роли слушателя, не доступен, он передает негативный отклик, как показано ниже.

```
L: <ожидание входящего соединения>
```

```
I: <создание соединения>
L: ERR 0 0 . 0 60
L: Content-Type: application/beep+xml
L:
L: <error code='421' />
L: END
I: RPY 0 0 . 0 52
I: Content-Type: application/beep+xml
I:
I: <greeting />
I: END
I: <закрытие соединения>
L: <закрытие соединения>
L: <ожидание следующего соединения>
```

Элемент greeting, переданный инициатором BEEP в этом случае игнорируется. Рекомендуется записать этот факт в системные журналы обоих партнеров BEEP.

Отметим, что в обоих примерах инициатор BEEP ждет, когда слушатель BEEP передаст greeting, но это лишь способ представления 0 - на практике оба партнера BEEP передают свои отклики одновременно.

Узел BEEP, желающий завершить сессию BEEP, передает элемент close с нулевым значением атрибута number в канал 0. Другой узел BEEP указывает свое согласие передачей элемента ok в позитивном отклике, как показано ниже.

```
C: MSG 0 1 . 52 60
C: Content-Type: application/beep+xml
C:
C: <close code='200' />
C: END
S: RPY 0 1 . 264 46
S: Content-Type: application/beep+xml
S:
S: <ok />
S: END
I: <закрытие соединения>
L: <закрытие соединения>
L: <ожидание следующего соединения>
```

Если партнер не хочет закрывать сессию BEEP, обмен может выглядеть иначе:

```
C: MSG 0 1 . 52 60
C: Content-Type: application/beep+xml
C:
C: <close code='200' />
C: END
S: ERR 0 1 . 264 79
S: Content-Type: application/beep+xml
S:
S: <error code='550'>still working</error>
S: END
```

Если завершение сеанса отвергнуто, сессию BEEP, по возможности, не следует прерывать.

2.5. Транспортные отображения

Все транспортные взаимодействия происходят в контексте сессии — отображения на конкретный транспортный сервис. По этой причине данный документ определяет требования, которые должны быть выполнены в любом документе, описывающем реализацию сессий BEEP с конкретным транспортным сервисом.

2.5.1. Управление сессией

Сессии BEEP работают на основе соединений (connection-oriented). Документ по отображению должен определять:

- как организуется сеанс BEEP;
- как узел BEEP указывает себя в роли слушателя;
- как узел BEEP указывает себя в роли инициатора;
- как сессия BEEP освобождается;
- как сессия BEEP разрывается (завершается).

2.5.2. Обмен сообщениями

Сессии BEEP работают на основе сообщений. Документ по отображению должен определять:

- как осуществляется гарантированная передачи и прием сообщений;
- как обеспечивается сохранение порядка сообщений в рамках одного канала;
- как сообщения разных каналов могут передаваться без ограничений по упорядочению.

2.6. Асинхронное взаимодействие

BEEP поддерживает асинхронные взаимодействия как в рамках одного канала, так и между каналами. Это свойство позволяет организовать конвейер (внутри канала) и параллельное взаимодействие (между каналами).

2.6.1. Внутри одного канала

Узел BEEP, выступающий в роли клиента, может передать множество сообщений MSG в один канал, не ожидая приема соответствующих откликов. Это позволяет создать конвейерную обработку сообщений в рамках одного канала.

Узел BEEP, играющий роль сервера, должен обрабатывать все сообщения MSG для данного канала в порядке их получения. В результате этого он должен генерировать отклики в том же порядке, в каком по данному каналу были получены соответствующие сообщения MSG.

Отметим, что при обмене «один со многими» (2.1.1. Стили обмена) откликом на сообщение MSG могут (не обязательно) служить сообщения ANS, за которыми следует сообщение NUL. При таком стиле обмена составляющие отклик сообщения ANS могут чередоваться. Когда сервер BEEP указывает завершение отклика путем генерации сообщения NUL, он может начать обработку следующего сообщения MSG, полученного по этому каналу.

2.6.2. Между разными каналами

Узел BEEP в роли клиента может передать множество сообщений MSG по разным каналам, не дожидаясь получения соответствующих откликов. Каналы работают независимо (в параллель).

Узел BEEP в роли сервера может обрабатывать сообщения MSG из разных каналов в любом порядке, выбранном им. В результате этого, хотя отклики для данного канала будут создаваться в порядке приема соответствующих сообщений MSG, на создание откликов в разных каналах ограничений по упорядочению откликов не накладывается.

2.6.3. Упреждающие отклики

Узел BEEP в роли сервера может передать негативный отклик до приема финального кадра сообщения MSG. В таких случаях узел BEEP должен игнорировать последующие кадры MSG для данного сообщения, вплоть до финального кадра MSG (включительно).

Если узел BEEP в роли клиента получает негативный отклик до отправки финального кадра сообщения MSG, он должен передать кадр MSG с индикацией завершения (.) и без данных.

2.6.4. Взаимовлияние сообщений

Если обработка конкретного сообщения оказывает влияние на другие сообщения (в том же или другом канале), соответствующий профиль должен определять поведение в таких случаях (например, сообщения которого меняют базовое транспортное отображение).

2.7. Поведение равноправных узлов

Протокол BEEP является одноранговым, поэтому оба партнера должны быть готовы к приему всех сообщений, определенных в этом документе. Поэтому инициатор BEEP, способный выступать лишь в качестве клиента, должен аккуратно вести себя при получении сообщений MSG. Все профили должны обеспечивать подобающие сообщения об ошибках для ответа на неожиданные сообщения MSG.

В силу одноранговой природы протокола BEEP нумерация для каждого направления передачи осуществляется независимо. Т. е. номера в сообщениях MSG, переданных инициатором BEEP, не связаны с номерами в сообщениях MSG, переданных узлом BEEP в роли слушателя.

Например, два приведенных ниже сообщения

```
I: MSG 0 1 . 52 120
I: Content-Type: application/beep+xml
I:
I: <start number='1'>
I:    <profile uri='http://iana.org/beep/SASL/OTP' />
I: </start>
I: END
L: MSG 0 1 . 221 116
L: Content-Type: application/beep+xml
L:
L: <start number='2'>
L:    <profile uri='http://iana.org/beep/APEX' />
L: </start>
L: END
```

указывают на разные сообщения в канале 0.

3. Транспортная безопасность

Когда сессия BEEP организована, данные передаются в открытом виде (plaintext) без защиты конфиденциальности. Соответственно, транспортная безопасность BEEP достигается использованием профиля начальной настройки.

Данный документ определяет один профиль:

- профиль транспортной защиты TLS на основе протокола TLS версии 1 [3].

На двухсторонней основе могут определяться и развертываться другие профили. Отметим, что по причине тесной связи с транспортным сервисом конкретный профиль транспортной защиты скорей всего будет относиться лишь к одному транспортному отображению (2.5. Транспортные отображения).

Когда канал с транспортной защитой начинает процесс согласования на уровне базового протокола, все каналы (включая канал 0) сессии BEEP будут закрыты. Поетому при завершении процесса согласования (с любым результатом) оба партнера будут передавать новые сообщения greeting. Если согласование завершится отказом, каждый из партнеров BEEP может разорвать сессию и об этом следует сделать запись в системном журнале.

Узел BEEP может выбирать разные варианты greeting с учетом используемой защиты конфиденциальности. Например,

```
L: <ожидание входящего соединения>
I: <организация соединения>
L: RPY 0 0 . 0 110
L: Content-Type: application/beep+xml
L:
L: <greeting>
L:    <profile uri='http://iana.org/beep/TLS' />
L: </greeting>
L: END
I: RPY 0 0 . 0 52
I: Content-Type: application/beep+xml
I:
I: <greeting />
I: END
I: MSG 0 1 . 52 158
I: Content-Type: application/beep+xml
I:
I: <start number='1'>
I:    <profile uri='http://iana.org/beep/TLS'>
I:        <![CDATA[<ready />]]>
I:    </profile>
I: </start>
I: END
L: RPY 0 1 . 110 121
L: Content-Type: application/beep+xml
L:
L: <profile uri='http://iana.org/beep/TLS'>
L:     <![CDATA[<proceed />]]>
L: </profile>
L: END
```

... успешное согласование транспортной защиты ...

```
L: RPY 0 0 . 0 221
L: Content-Type: application/beep+xml
L:
L: <greeting>
L:    <profile uri='http://iana.org/beep/SASL/ANONYMOUS' />
L:    <profile uri='http://iana.org/beep/SASL/OTP' />
L:    <profile uri='http://iana.org/beep/APEX' />
L: </greeting>
L: END
I: RPY 0 0 . 0 52
I: Content-Type: application/beep+xml
I:
I: <greeting />
I: END
```

Конечно, не все узлы BEEP должны быть такими целеустремленными:

```
L: <ожидание входящего соединения>
I: <организация соединения>
L: RPY 0 0 . 0 268
L: Content-Type: application/beep+xml
L:
L: <greeting>
L:    <profile uri='http://iana.org/beep/SASL/ANONYMOUS' />
L:    <profile uri='http://iana.org/beep/SASL/OTP' />
L:    <profile uri='http://iana.org/beep/APEX' />
L:    <profile uri='http://iana.org/beep/TLS' />
L: </greeting>
L: END
I: RPY 0 0 . 0 52
I: Content-Type: application/beep+xml
I:
I: <greeting />
I: END
I: MSG 0 1 . 52 158
I: Content-Type: application/beep+xml
I:
I: <start number='1'>
I:    <profile uri='http://iana.org/beep/TLS'>
I:        <![CDATA[<ready />]]>
```

```
I:    </profile>
I: </start>
I: END
L: RPY 0 1 . 268 121
L: Content-Type: application/beep+xml
L:
L: <profile uri='http://iana.org/beep/TLS'>
L:     <![CDATA[<proceed />]]>
L: </profile>
L: END
```

... отказ при согласовании транспортной защиты ...

```
L: RPY 0 0 . 0 268
L: Content-Type: application/beep+xml
L:
L: <greeting>
L:    <profile uri='http://iana.org/beep/SASL/ANONYMOUS' />
L:    <profile uri='http://iana.org/beep/SASL/OTP' />
L:    <profile uri='http://iana.org/beep/APEX' />
L:    <profile uri='http://iana.org/beep/TLS' />
L: </greeting>
L: END
I: RPY 0 0 . 0 52
I: Content-Type: application/beep+xml
I:
I: <greeting />
I: END
```

3.1. Профиль транспортной защиты TLS

Регистрация данного профиля описана в параграфе 6.2.

3.1.1. Идентификации и инициализация профиля

Профиль транспортной защиты TLS идентифицируется как

http://iana.org/beep/TLS

в элементе profile в процессе создания канала.

В процессе организации канала соответствующий элемент profile внутри элемента start может содержать элемент ready. Если канал создан, перед отправкой соответствующего отклика узел BEEP обрабатывает элемент ready и включает результат в отклик, как показано ниже.

```
C: MSG 0 1 . 52 158
C: Content-Type: application/beep+xml
C:
C: <start number='1'>
C:    <profile uri='http://iana.org/beep/TLS'>
C:        <![CDATA[<ready />]]>
C:    </profile>
C: </start>
C: END
S: RPY 0 1 . 110 121
S: Content-Type: application/beep+xml
S:
S: <profile uri='http://iana.org/beep/TLS'>
S:     <![CDATA[<proceed />]]>
S: </profile>
S: END
```

Отметим, что возможны ситуации, когда канал создан, но инкапсулированная операция привела к отказу.

```
C: MSG 0 1 . 52 173
C: Content-Type: application/beep+xml
C:
C: <start number='1'>
C:    <profile uri='http://iana.org/beep/TLS'>
C:        <![CDATA[<ready version="oops" />]]>
C:    </profile>
C: </start>
C: END
S: RPY 0 1 . 110 193
S: Content-Type: application/beep+xml
S:
S: <profile uri='http://iana.org/beep/TLS'>
S:     <![CDATA[<error code='501'>version attribute
S: poorly formed in &lt;ready&gt; element</error>]]>
S: </profile>
S: END
```

В таком случае передается позитивный отклик (поскольку канал создан), но инкапсулированный в него отклик указывает причину отказа операции.

3.1.2. Синтаксис сообщений

В параграфе 7.2 определены сообщения, используемые профилем транспортной защиты TLS.

3.1.3. Семантика сообщений

3.1.3.1. Сообщение Ready

Элемент ready имеет необязательный атрибут version и не включает содержимого:

- элемент version указывает наиболее старую версию TLS, принимаемую для использования.

Когда узел BEEP передал элемент ready, ему недопустимо передавать какой-либо трафик базовому транспорту до получения соответствующего отклика (proceed или error). Получивший такой элемент узел BEEP должен завершить отправку всех ожидающих откликов и только после этого обрабатывать элемент ready.

3.1.3.2. Сообщение Proceed

Элемент proceed не имеет атрибутов и содержимого, он передается в качестве отклика на элемент ready.

Когда узел BEEP получает элемент ready, ему недопустимо передавать какой-либо трафик базовому транспорту до генерации соответствующего отклика[1]. Если узел BEEP решил принять предложенную транспортную защиту, он неявно закрывает все каналы (включая канал 0), передает элемент proceed и ждет завершения процесса согласования защиты на транспортном уровне.

Когда узел BEEP получает элемент proceed в отклике на свой элемент ready, он неявно закрывает все каналы (включая канал 0) и незамедлительно начинает процесс согласования защиты на транспортом уровне.

4. Аутентификация пользователей

На момент организации сессии BEEP обеспечивается анонимный доступ без отслеживания пользователя. Поэтому проверка подлинности в BEEP обеспечивается с помощью профиля начальной настройки.

Этот документ определяет семейство профилей, основанных на механизмах SASL:

- каждый механизм из реестра IANA SASL [15] имеет соответствующий профиль.

На двухсторонней основе могут определяться и развертываться другие профили.

При успешной аутентификации на любом канале проверенное отождествление применяется во всех действующих и будущих каналах данной сессии BEEP и дополнительные проверки подлинности не разрешаются.

Отметим, что независимо от транспортной защиты и проверки подлинности пользователя вопрос предоставления полномочий решается на уровне каждого узла BEEP. Поэтому каждый узел может ограничивать набор доступных операций на основе представленных при аутентификации свидетельств (т. е., неуполномоченные операции могут быть отвергнуты с кодом ошибки 530).

4.1. Семейство профилей SASL

Регистрация для этого профиля представлена в параграфе 6.3.

Отметим, что SASL может обеспечивать аутентификацию пользователей и транспортную защиту. Если транспортная защита уже согласована для сессии BEEP, использование уровня защиты SASL не допускается, а после успешного согласования SASL профилю транспортной защиты недопустимо начинать свой процесс согласования.

В разделе 4 спецификации SASL [4] требуется предоставление в определении протокола указанной ниже информации:

Имя службы. beep

Инициализационная последовательность. Создание канала с использованием профиля BEEP, соответствующего механизму SASL начинает обмен сообщениями. Дополнительный параметр, соответствующий начальному отклику, отправленному клиентом, передается в элементе blob в процессе создания канала.

Последовательность обмена. Вызовы (запросы) и отклики передаются в элементах blob. Атрибут status элемента blob используется сервером для индикации успешного завершения обмена, а коиентом — для прерывания обмена. Сервер указывает отказ при обмене передачей элемента error.

Согласование уровня защиты. В начале согласования уровня защиты все каналы (включая канал 0) сессии BEEP закрываются. Следовательно, при завершении процесса согласования, независимо от результата, оба партнера BEEP передают новые сообщения greeting.

Если уровень защиты согласован, он вступает в силу сразу после сообщения с откликом об успешном завершении.

Использование отождествления для предоставления полномочий. Это отождествление доступно для всех каналов в течение всего срока действия сессии BEEP.

4.1.1. Идентификация и инициализация профиля

Каждый механизм SASL, зарегистрированный IANA, включается в список

`http://iana.org/beep/SASL/mechanism`

где mechanism указывает имя, присвоенное этому механизму агентством IANA.

Отметим, что в процессе создания канала узел BEEP может предлагать удаленному партнеру множество профилей, как показано ниже.

[1] С учетом последнего предложения параграфа 3.1.3.1 корректней было бы сказать: «Когда узел BEEP начал обработку элемента ready ...» *Прим. перев.*

```
C: MSG 0 1 . 52 178
C: Content-Type: application/beep+xml
C:
C: <start number='1'>
C:    <profile uri='http://iana.org/beep/SASL/ANONYMOUS' />
C:    <profile uri='http://iana.org/beep/SASL/OTP' />
C: </start>
C: END
S: RPY 0 1 . 221 87
S: Content-Type: application/beep+xml
S:
S: <profile uri='http://iana.org/beep/SASL/OTP' />
S: END
```

В процессе создания канала соответствующий элемент profile в элементе BEEP start может содержать элемент blob. Отметим, что возможно создание канала и отказ инкапсуляции. Например,

```
C: MSG 0 1 . 52 183
C: Content-Type: application/beep+xml
C:
C: <start number='1'>
C:    <profile uri='http://iana.org/beep/SASL/OTP'>
C:        <![CDATA[<blob>AGJsb2NrbWFzdGVy</blob>]]>
C:    </profile>
C: </start>
C: END
S: RPY 0 1 . 221 178
S: Content-Type: application/beep+xml
S:
S: <profile uri='http://iana.org/beep/SASL/OTP'>
S:     <![CDATA[<error code='534'>authentication mechanism is
S: too weak</error>]]>
S: </profile>
S: END
```

В этом случае передается позитивный отклик (о создании канала), но отклик инкапсуляции будет сообщать об отказе.

В остальных случаях сервер передает вызов (challenge) или подтверждает успех. Например,

```
C: MSG 0 1 . 52 183
C: Content-Type: application/beep+xml
C:
C: <start number='1'>
C:    <profile uri='http://iana.org/beep/SASL/OTP'>
C:        <![CDATA[<blob>AGJsb2NrbWFzdGVy</blob>]]>
C:    </profile>
C: </start>
C: END
S: RPY 0 1 . 221 171
S: Content-Type: application/beep+xml
S:
S: <profile uri='http://iana.org/beep/SASL/OTP'>
S:    <![CDATA[<blob>b3RwLXNoYTEgOTk5NyBwaXh5bWlzYXM4NTgwNSBleHQ=
                                                    </blob>]]>
S: </profile>
S: END
```

Отметим, что в этом примере предполагается, что элемент blob в отклике сервера имеет появляется в двух строках — это особенность представления. Фактически используется одна строка.

Если получен вызов, клиент отвечает на него и ждет другого отклика. Например,

```
C: MSG 1 0 . 0 97
C: Content-Type: application/beep+xml
C:
C: <blob>d29yZDpmZXJuIGhhbmcgYnJvdyBib25nIGhlcmQgdG9n</blob>
C: END
S: RPY 1 0 . 0 66
S: Content-Type: application/beep+xml
S:
S: <blob status='complete' />
S: END
```

Клиент может, естественно, прервать процесс проверки подлинности, передав <blob status='abort' />.

Кроме того, сервер может отвергнуть отклик клиента. Например,

```
C: MSG 1 0 . 0 97
C: Content-Type: application/beep+xml
C:
C: <blob>d29yZDpmZXJuIGhhbmcgYnJvdyBib25nIGhlcmQgdG9n</blob>
C: END
S: ERR 1 0 . 0 60
```

```
S: Content-Type: application/beep+xml
S:
S: <error code='535' />
S: END
```

В зависимости от механизма SASL элемент инициализации может передаваться в одном направлении при создании канала. Например,

```
C: MSG 0 1 . 52 125
C: Content-Type: application/beep+xml
C:
C: <start number='1'>
C:    <profile uri='http://iana.org/beep/SASL/CRAM-MD5' />
C: </start>
C: END
S: RPY 0 1 . 221 185
S: Content-Type: application/beep+xml
S:
S: <profile uri='http://iana.org/beep/SASL/CRAM-MD5'>
S: <![CDATA[<blob>PDE4OTYuNjk3MTcwOTUyQHBvc3RvZmZpY2UucmVzdG9uLm1
                                                  jaS5uZXQ+</blob>]]>
S: </profile>
S: END
```

Отметим, что в этом примере предполагается, что элемент blob в отклике сервера имеет появляется в двух строках — это особенность представления. Фактически используется одна строка.

4.1.2. Синтаксис сообщений

В параграфе 7.3 определены сообщения, используемые для каждого профиля семейства SASL.

Отметим, что в результате использования в SASL множества механизмов обмена двоичными данными, содержимое элемента blob всегда представляется в форме строк base64.

4.1.3. Семантика сообщений

Элемент blob имеет необязательный атрибут status, содержащий произвольные октеты:

- при наличии атрибута status они может принимать одно из трех значений:

  abort используется клиентом для индикации прерывания процесса проверки подлинности;

  complete и используется сервером для индикации успешного завершения обмена;

  continue используется любой из сторон в остальных случаях.

В заключение отметим, что механизм EXTERNAL в SASL работает с «внешними службами аутентификации», которые могут обеспечиваться одним из описанные ниже способов:

- в соединении профиль будет использоваться транспортной защиты, способный предоставлять данные аутентификации (см., например, параграф 3.1);
- базовое соединение обеспечивает сетевую службу, способную выполнить строгую проверку подлинности (например, IPSec [12]);
- локально определенные услуги защиты.

Для успешной аутентификации должны быть выполнены два условия:

- внешняя служба аутентификации должна быть активна№
- при наличии аутентификационного отождествления оно должно согласоваться со свидетельствами, представляемыми внешней службой аутентификации (если такое отождествление пусто, оно автоматически выводится из свидетельств, представленных внешней службой проверки подлинности).

5. Регистрационные шаблоны

5.1. Шаблон регистрации профиля

При регистрации профиля предоставляется перечисленная ниже информация.

Идентификация профиля: задает идентификатор URI [10], полномочно указывающий этот профиль.

Сообщения при организации канала: задает типы данных, которые могут передаваться при создании канала.

Сообщения а начале обмена «один к одному»: задает типы данных, которые могут присутствовать в начале обмена.

Сообщения в позитивных откликах: задает типы данных, которые могут присутствовать в позитивном отклике.

Сообщения в негативных откликах: задает типы данных, которые могут присутствовать в негативном отклике.

Сообщения в обмене «один со многими»: задает типы данных, которые могут присутствовать в обмене «один-к-одному».

Синтаксис сообщений: задает синтаксис типов данных для обмена в этом профиле.

Семантика сообщений: задает семантику типов данных для обмена в этом профиле.

Контактные данные: задает почтовые и электронные данные для связи с автором профиля.

5.2. Шаблон регистрации функции

При регистрации функции для профиля канала управления предоставляется следующая информация:

Идентификация функции: указывает строку идентификатора функции (если функция регистрируется в IANA, идентификатор должен начинаться с x-);

Семантика функции: указывает семантику функции;

Контактные данные: задает почтовые и электронные данные для связи с автором профиля.

6. Исходные регистрации

6.1. Управление каналом BEEP

Идентификация профиля: не применимо.

Сообщения при организации канала: не применимо.

Сообщения а начале обмена «один к одному»: "start" или "close".

Сообщения в позитивных откликах: "greeting", "profile" или "ok".

Сообщения в негативных откликах: "error".

Сообщения в обмене "один со многими": нет.

Синтаксис сообщений: см. параграф 7.1.

Семантика сообщений: см. параграф 2.3.1.

Контактные данные: см. раздел «Адрес автора» в этом документе.

6.2. Профиль транспортной защиты TLS

Идентификация профиля: http://iana.org/beep/TLS.

Сообщения при организации канала: "ready".

Сообщения а начале обмена «один к одному»: "ready".

Сообщения в позитивных откликах: "proceed".

Сообщения в негативных откликах: "error".

Сообщения в обмене «один со многими»: нет.

Синтаксис сообщений: см. параграф 7.2.

Семантика сообщений: см. параграф 3.1.3.

Контактные данные: см. раздел «Адрес автора» в этом документе.

6.3. Семейство профилей SASL

Идентификация профиля: http://iana.org/beep/SASL/mechanism, где mechanism указывает маркер, зарегистрированный IANA.

Сообщения при организации канала: "blob".

Сообщения а начале обмена «один к одному»: "blob".

Сообщения в позитивных откликах: "blob".

Сообщения в негативных откликах: "error".

Сообщения в обмене «один со многими»: нет.

Синтаксис сообщений: см. параграф 7.3.

Семантика сообщений: см. параграф 4.1.3.

Контактные данные: см. раздел «Адрес автора» в этом документе.

6.4. application/beep+xml

Тип MIME media: application.

Субтип MIME: beep+xml.

Требуемые параметры: нет.

Дополнительные параметры: кодировка (по умолчанию UTF-8 [13]).

Вопросы представления: Этот тип может содержать двоичные данные, поэтому при использовании транспорта, который не поддерживает передачу таких данных, должно применяться подходящее кодирование.

Вопросы безопасности: Нет, однако любой профиль BEEP, использующий этот тип среды (носителя), должен описывать связанные с ним проблемы безопасности.

Вопросы интероперабельности: не применимо.

Опубликованная спецификация: Этот тип является подмножеством спецификации XML 1.0 [2] с двумя исключениями. Во-первых, не может присутствовать ссылок, на элементы отличающихся от 5 предопределенных базовых ("&", "<", ">", "'", and """) и числовых. Во-вторых, не могут применяться декларации XML (например, <?xml version="1.0" ?>) или DOCTYPE (например, <!DOCTYPE ...>). Поэтому в случае использования кодировки, отличающейся от UTF-8, должен применяться параметр charset. Все остальные инструкции XML 1.0 (например, блоки CDATA, инструкции обработки и т. п.) разрешены.

Приложения, которые могут использовать этот тип: Любой профиль BEEP, применяющий подмножество XML 1.0.

Дополнительная информация: нет.

Контакты для получения дополнительной информации: см. раздел «Адрес автора» в этом документе.

Предусмотренное использование: ограниченное применение.

Автор или контролер изменений: IESG.

7. Определения типов документов (DTD)

7.1. Управление каналами BEEP

```
<!--
  DTD для управления каналами BEEP от 2000-10-29
  См. это DTD по ссылке:

    <!ENTITY % BEEP PUBLIC "-//IETF//DTD BEEP//EN"
              "http://xml.resource.org/profiles/BEEP/beep.dtd">
    %BEEP;
  -->

<!--
  Типы данных DTD:
        элемент       синтаксис/ссылка     пример
        =======       ================     =======
    номер канала
        CHAN          1..2147483647        1

    полномочная идентификация профиля
        URI           c.f., [RFC-2396]     http://invisible.net/

    один или несколько маркеров возможностей, разделенных пробелами
        FTRS          NMTOKENS             "magic"

    тег языка
        LANG          c.f., [RFC-1766]     "en", "en-US", etc.

    Необязательные теги языков
        LOCS          NMTOKENS             "en-US"

    3-значный код отклика
        XYZ            [1-5][0-9][0-9]     500
-->

<!ENTITY % CHAN       "CDATA">
<!ENTITY % URI        "CDATA">
<!ENTITY % FTRS       "NMTOKENS">
<!ENTITY % LANG       "NMTOKEN">
<!ENTITY % LOCS       "NMTOKEN">
<!ENTITY % XYZ        "CDATA">

<!--
  Сообщения BEEP, передаваемые как application/beep+xml

     роль       MSG         RPY         ERR
    =======     ===         ===         ===
    I и L                   greeting    error

    I или L     start       profile     error

    I или L     close       ok          error
  -->

<!ELEMENT greeting    (profile)*>
<!ATTLIST greeting
          features    %FTRS;            #IMPLIED
          localize    %LOCS;            "i-default">
```

```
<!ELEMENT start       (profile)+>
<!ATTLIST start
          number      %CHAN;              #REQUIRED
          serverName  CDATA               #IMPLIED>

<!-- элемент profile пуст, если содержится в greeting -->
<!ELEMENT profile     (#PCDATA)>
<!ATTLIST profile
          uri         %URI;               #REQUIRED
          encoding    (none|base64)       "none">

<!ELEMENT close       (#PCDATA)>
<!ATTLIST close
          number      %CHAN;              "0"
          code        %XYZ;               #REQUIRED
          xml:lang    %LANG;              #IMPLIED>

<!ELEMENT ok          EMPTY>

<!ELEMENT error       (#PCDATA)>
<!ATTLIST error
          code        %XYZ;               #REQUIRED
          xml:lang    %LANG;              #IMPLIED>
```

7.2. Профиль транспортной защиты TLS

```
<!--
  DTD для профиля TLS Transport Security от 2000-09-04

  Для ссылки на это DTD служит:

    <!ENTITY % TLS PUBLIC "-//IETF//DTD TLS//EN"
              "http://xml.resource.org/profiles/TLS/tls.dtd">
    %TLS;
  -->

<!--
  Сообщения TLS, передаваемые как application/beep+xml

     роль       MSG         RPY         ERR
    ======      ===         ===         ===
    I или L     ready       proceed     error
  -->

<!ELEMENT ready       EMPTY>
<!ATTLIST ready
          version     CDATA               "1">

<!ELEMENT proceed     EMPTY>
```

7.3. Семейство профилей SASL

```
<!--
  DTD для семейства профилей SASL от 2000-09-04

  Для ссылки на это DTD служит:

    <!ENTITY % SASL PUBLIC "-//IETF//DTD SASL//EN"
              "http://xml.resource.org/profiles/sasl/sasl.dtd">
    %SASL;
  -->

<!--
  Сообщения SASL, передаваемые как application/beep+xml

     роль       MSG         RPY         ERR
    ======      ===         ===         ===
    I или L     blob        blob        error
  -->

<!ELEMENT blob        (#PCDATA)>
<!ATTLIST blob
          xml:space   (default|preserve)
                                        "preserve"
          status      (abort|complete|continue)
                                        "continue">
```

8. Коды откликов

Код	Значение
200	Успех.
421	Сервис не доступен.
450	Запрошенная операция не выполнена (например, блокировка уже существует).
451	Запрошенная операция прервана (например, локальная ошибка при обработке).
454	Временный отказ при проверке подлинности.
500	Синтаксическая ошибка (например, некорректный формат XML).
501	Синтаксическая ошибка в параметрах (например, недействительный XML).
504	Параметр не реализован.
530	Требуется аутентификация.
534	Недостаточно механизма проверки подлинности (например, слишком слабый).
535	Отказ при проверке подлинности.
537	Действие не разрешено для пользователя.
538	Механизм проверки подлинности требует шифрования.
550	Запрошенная операция не выполнена (например, недоступны запрошенные профили).
553	Непригодный параметр.
554	Отказ транзакции (например, нарушение правил).

9. Вопросы безопасности

Механизм кадрирования BEEP, сам по себе, не обеспечивает защиты от атак, однако разумные начальные настройки профилей обеспечивают ту или иную степень уверенности.

1. При использовании одного из профилей SASL следует принять во внимание раздел 9 в [4], где рассмотрены вопросы безопасности.
2. При использовании транспортной защиты TLS (или согласованном уровне защиты SASL) следует принимать во внимание описанные ниже аспекты.
    1. Посредник MITM[1] может удалить связанные с защитой профили из приветствия BEEP или создать негативный отклик для элемента ready профиля транспортной защиты TLS. Узел BEEP можно настроить на отказ от обработки без приемлемого уровня защиты.
    2. MITM-посредник может вынудить к принятию наиболее слабого шифра. Узлам BEEP следует поддерживать возможность настройки для отказа от слабых шифров.
    3. MITM-посредник может может изменить любой протокольный обмен до успешного согласования. После согласования узел BEEP должен отбрасывать ранее кэшированную информацию о сессии BEEP.

    Поскольку разные шифронаборы TLS обеспечивают разную степень защиты, администраторам следует аккуратно выбирать предлагаемые шифры.

Одноранговая природа протокола BEEP требует перед выполнением любой задачи, связанной с сообщением, применять средства контроля доступа с учетом аутентификации и уровня конфиденциальности сессии BEEP.

Литература

[1] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

[2] World Wide Web Consortium, "Extensible Markup Language (XML) 1.0", W3C XML, February 1998, <http://www.w3.org/TR/1998/REC-xml-19980210>.

[3] Dierks, T., Allen, C., Treese, W., Karlton, P., Freier, A. And P. Kocher, "The TLS Protocol Version 1.0", RFC 2246, January 1999.

[4] Myers, J., "Simple Authentication and Security Layer (SASL)", RFC 2222, October 1997.

[5] Rose, M., "Mapping the BEEP Core onto TCP", RFC 3081, March 2001.

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

[7] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 2234, November 1997.

[8] Elz, R. and R. Bush, "Serial Number Arithmetic", RFC 1982, August 1996.

[9] Alvestrand, H., "Tags for the Identification of Languages", RFC BCP 47, RFC 3066, January 2001.

[10] Berners-Lee, T., Fielding, R. and L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax", RFC 2396[2], August 1998.

[11] Newman, C., "The One-Time-Password SASL Mechanism", RFC 2444, October 1998.

[1]Man-in-the-middle - человек посередине.
[2]Этот документ заменен RFC 3986. *Прим. перев.*

[12] Kent, S. and R. Atkinson, "Security Architecture for the Internet Protocol", RFC 2401[2], November 1998.

[13] Yergeau, F., "UTF-8, a transformation format of ISO 10646", RFC 2279, January 1998.

[14] Linn, J., "Generic Security Service Application Program Interface, Version 2", RFC 2078[3], January 1997.

[15] <http://www.isi.edu/in-notes/iana/assignments/sasl-mechanisms>

Адрес автора

Marshall T. Rose

Invisible Worlds, Inc.

1179 North McDowell Boulevard

Petaluma, CA 94954-6559

US

Phone: +1 707 789 3700

EMail: mrose@invisible.net

URI: http://invisible.net/

Перевод на русский язык

Николай Малых

nmalykh@gmail.com

Приложение A. Благодарности

Автор благодарен за вклад в работу David Clark, Dave Crocker, Steve Deering, Wesley Michael Eddy, Huston Franklin, Marco Gazzetta, Danny Goodman, Steve Harris, Robert Herriot, Ken Hirsch, Greg Hudson, Ben Laurie, Carl Malamud, Michael Mealling, Keith McCloghrie, Paul Mockapetris, RL 'Bob' Morgan, Frank Morton, Darren New, Chris Newman, Joe Touch, Paul Vixie, Gabe Wachob, Daniel Woods и James Woodyatt. В частности, Dave Crocker внес важные предложения по природе сегментации в механизме кадрирования.

Приложение B. Взаимодействие с IANA

Агентство IANA зарегистрировало beep в качестве имени службы GSSAPI [14], как указано в параграфе 4.1.

IANA поддерживает списки:

- стандартизуемых профилей BEEP (5.1. Шаблон регистрации профиля);
- стандартизуемых возможностей для профиля управления каналом (5.2. Шаблон регистрации функции).

Для каждого списка IESG отвечает за выделение эксперта (designated expert) для рецензирования спецификации до включения в реестры IANA. Разработчики не стандартизуемых профилей и фкнций управления каналом BEEP могут запросить комментарии к своим предложениям через список рассылки bxxpwg@invisible.net.

Агентство IANA выполнило регистрации, указанные в параграфах 6.2 и 6.3. Рекомендуется использовать префикс IANA в URI и включать в эти URI соответствующие регистрации.

Полное заявление авторских прав

Подтверждение

Финансирование функций RFC Editor обеспечено Internet Society.

[2]Этот документ заменен RFC 4301. *Прим. перев.*
[3]Этот документ заменен RFC 2743. *Прим. перев.*