Санкт-Петербург, ул. Седова, 80, телефон (812) 449-0770, факс (812) 449-0771, E-mail: info@bilim.com

Network Working Group — A. Melnikov, Ed.
Request for Comments: 4422 — Isode Limited
Obsoletes: 2222 — K. Zeilenga, Ed.
Category: Standards Track — OpenLDAP Foundation
June 2006

Простой уровень аутентификации и защиты (SASL)

Simple Authentication and Security Layer (SASL)

Статус документа

В этом документе содержится спецификация протокола, предложенного сообществу Internet. Документ служит приглашением к дискуссии в целях развития и совершенствования протокола. Текущее состояние стандартизации протокола вы можете узнать из документа "Internet Official Protocol Standards" (STD 1). Документ может распространяться без ограничений.

Тезисы

Простой уровень аутентификации и защиты (SASL[1]) представляет собой схему обеспечения услуг по аутентификации и защите данных в основанных на прямых соединениях протоколов на основе механизмов, допускающих замену. Эта схема обеспечивает структурированный интерфейс между протоколами и механизмами. Схема позволяет использовать существующие механизмы для новых протоколов и добавлять новые механизмы для старых протоколов. Кроме того, эта схема обеспечивает протокол для защиты последующих протокольных транзакций внутри уровня защиты данных.

В этом документе описана структуризация механизма SASL, поддержка SASL в различных протоколах и протокол поддержки уровня защиты данных для соединений. Кроме того, в документе определен один из механизмов SASL – механизм EXTERNAL.

Данный документ отменяет действие RFC 2222.

Оглавление

[1]Simple Authentication and Security Layer

1. Введение

Простой уровень аутентификации и защиты (SASL[2]) представляет собой схему обеспечения услуг по аутентификации и защите данных в основанных на прямых соединениях протоколов на основе механизмов, допускающих замену. Уровень защиты данных может обеспечивать целостность данных, их конфиденциальность и иные средства защиты.

Устройство SASL позволяет новым протоколам использовать существующие механизмы без их реконструкции, а также позволяет добавлять новые механизмы к существующим протоколам без изменения последних.

Концептуально SASL представляет собой схему, обеспечивающую уровень абстракции между протоколами и механизмами, как показано на рисунке.

Благодаря интерфейсам этого уровня абстракции схема позволяет любому протоколу использовать любые механизмы. Хотя этот уровень в общем случае скрывает конкретные протоколы от механизмов (и наоборот), он не прячет в общем случае конкретные механизмы от реализации протокола. Например, для работы различных механизмов требуется разная информация – некоторые используют аутентификацию на основе пароля, другим нужна информация об областях (realm), третьи применяют ярлыки (ticket) Kerberos, сертификаты и т. д. Кроме того, для проверки полномочий серверные реализации в общем случае применяют отображение между объектами[3] аутентификации, форма которых определяется механизмом, и объектами проверки полномочий, чья форма определяется прикладными протоколами. Концепции идентификации рассмотрены в главе 2.

Можно разработать и реализовать схему таким образом, чтобы абстрагироваться от конкретных деталей механизмов. Реализация такой схемы, как и реализации механизмов могут использоваться не только множеством реализаций конкретного протокола, но и реализациями различных протоколов.

Схема включает интерфейсы с протоколами и механизмами, с помощью которых выполняется обмен аутентификационными данными. Процесс аутентификационного обмена SASL рассматривается в главе 3.

Для использования SASL каждый протокол обеспечивает метод идентификации используемого механизма, метод обмена обусловленными механизмом запросами со стороны сервера[4] и откликами клиентов[5], а также метод обмена результатами аутентификации. Требования SASL к протоколам обсуждаются в главе 4.

Каждый механизм SASL определяет последовательности серверных запросов и откликов клиентов, которые обеспечивают аутентификацию и согласование сервиса защиты данных. Требования SASL к механизмам рассмотрены в главе 5.

В главе 6 обсуждаются вопросы безопасности, глава 7 содержит информацию о согласовании с IANA. В приложении A определен механизм SASL EXTERNAL.

1.1. Аудитория

Этот документ рассчитан на несколько категорий читателей:

- разработчики протоколов, которые используют спецификацию для поддержки этими протоколами аутентификации;
- разработчики механизмов, создающие новые механизмы SASL;
- разработчики клиентских и серверных программ для протоколов, поддерживающих SASL.

Хотя документ организован так, чтобы сосредоточить внимание читателей на деталях, относящихся к их разработкам, читателям рекомендуется прочесть и понять все отраженные в документе вопросы.

1.2. Связь с другими документами

Данный документ прекращает действие RFC 2222. Он заменяет все части RFC2222, за исключением параграфов 7.1 (механизм KERBEROS_IV), 7.2 (механизм GSSAPI) и 7.3 (механизм SKEY). Механизмы KERBEROS_IV и SKEY в настоящее время представляются устаревшими и их спецификации в RFC 2222 получают статус Historic. Спецификация механизма GSSAPI в настоящее время содержится в отдельном документе [SASL-GSSAPI].

В приложении B приводится список отличий от RFC 2222.

1.3. Использование терминов

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не следует** (SHALL NOT), **следует** (SHOULD), **не нужно** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе интерпретируются в соответствии с BCP 14 [RFC2119].

[2]Simple Authentication and Security Layer
[3]В оригинале используется термин identity (подлинность, тождество, идентичность). *Прим. перев.*
[4]server-challenge
[5]client-response

Символы, используемые в данном документе для обозначения кодов (code point) и имен, соответствуют стандарту Unicode [Unicode]. Например, буква "a" может быть представлена как <U+0061> или <LATIN SMALL LETTER A>[6].

В примерах префиксы "C:" и "S:" указывают строки, передаваемые клиентом и сервером, соответственно. Строки могут переноситься для удобства чтения документа.

2. Концепции идентификации

На практике аутентификация и проверка полномочий (authorization) может включать множество объектов проверки тождественности, возможно в разных формах (простые имена пользователей - username, Kerberos principal, X.500 Distinguished Name и т. п.) и с различным представлением (например строки ABNF в кодировке UTF-8, Distinguished Name в кодировке BER). Хотя технические спецификации часто описывают форму и представление объектов идентификации, используемых в сети, реализации могут использовать различные формы объектов и/или варианты их представления. Отношения между объектами идентификации различных форм в общем случае определяются локально. Используемые реализацией формы и варианты представления объектов также определяются локально самой реализацией.

Однако концептуально схема SASL включает два объекта идентификации:

1) идентификация, связанная с предъявленными при аутентификации полномочиями (authentication credentials), обозначаемая термином authentication identity (аутентификационная идентификация), and

2) идентификация при проверке полномочий, обозначаемая термином authorization identity.

Спецификации механизмов SASL описывают форму полномочий (например, сертификаты X.509, ярлыки Kerberos, простые пары username/password), предъявляемых при аутентификации клиента, включая (когда это приемлемо) синтаксис и семантику объектов идентификации, передаваемых в свидетельствах полномочий (credentials). Спецификации протокола SASL описывают форму объектов идентификации, используемых при проверке полномочий (authorization) и, в частности, синтаксис и семантику строк объектов идентификации, передаваемых механизмами.

Клиент представляет свидетельства полномочий (которые включают или подразумевают authentication identity) и, возможно, символьную строку, представляющую запрашиваемую как часть обмена SASL аутентифкационную идентификацию. Когда эта строка отсутствует или пуста, это говорит о том, что клиент запрашивает использование идентификации, связанной с предъявленными полномочиями (например, пользователь просит принять authentication identity).

Сервер отвечает за проверку того, что предъявленные клиентом свидетельства и идентификация, связанная с этими свидетельствами клиента (например, authentication identity), может использоваться как идентификация при проверке полномочий. Обмен SASL прерывается отказом при отрицательном результате любой из этих проверок (обмен SASL может также прерываться отказом по иным причинам, например, в результате отказа службы проверки полномочий).

Однако конкретные формы authentication identity (используемые сервером для верификации) и authorization identity (используемые для проверки полномочий) не входят в спецификацию SASL. В некоторых случаях конкретные формы идентификации, используемые в том или ином контексте за пределами обмена SASL, могут диктоваться другими спецификациями. Например, спецификация правил identity assumption authorization (proxy authorization) может задавать представление идентификации при проверке полномочий и аутентификации в правилах политики.

3. Аутентификационный обмен

Каждый аутентификационный обмен состоит из клиентского сообщения серверу с запросом аутентификации на основе определенного механизма. После этого следует одна или несколько пар запросов (challenge) от сервера и откликов (response) от клиента. Завершает обмен сообщение от сервера, указывающее результат проверки. Отметим, что обмен может быть прерван, как описано в параграфе 3.5.

Ниже приводится схематическое представление процесса аутентификационного обмена.

```
C: Запрос аутентификационного обмена
S: Первоначальный запрос (challenge) сервера
C: Первоначальный отклик
<дополнительные сообщения challenge/response>
S: Результат аутентификационного обмена
```

Если обмен завершился с положительным результатом проверки и согласован уровень защиты, этот уровень устанавливается, как описано в параграфе 3.7. Это правило применимо также к представленным ниже иллюстрациям.

Некоторые механизмы указывают, что первыми данными в процессе аутентификационного обмена являются данные от клиента, передаваемые серверу. Протоколы могут обеспечивать дополнительное поле начального отклика в сообщении с запросом для передачи таких данных. Когда механизм задает, что первыми передаются данные от клиента к серверу, протокол обеспечивает дополнительное поле начального отклика и клиент использует это поле, что позволяет сократить время обмена на один интервал кругового обхода:

```
C: Запрос аутентификационного обмена + Первоначальный отклик
<дополнительные сообщения challenge/response>
S: Результат аутентификационного обмена
```

Когда механизм задает передачу первыми данных от клиента к серверу и упомянутое поле недоступно или не используется, за клиентским запросом передается пустой запрос (challenge) от сервера.

```
C: Запрос аутентификационного обмена
S: Пустой запрос (Challenge)
C: Первоначальный отклик
<дополнительные сообщения challenge/response>
S: Результат аутентификационного обмена
```

Если клиент включит начальный отклик в свой запрос на организацию аутентификационного обмена в том случае, когда механизм не позволяет клиенту передавать данные первым, аутентификационный обмен завершится отказом.

[6]Глоссарий терминов, используемых в Unicode, можно найти в документе [Glossary]. Информация о модели кодирования символов Unicode имеется в документе [CharModel].

Некоторые механизмы задают передачу сервером дополнительных данных, которые показывают клиенту успешное завершение обмена. Протоколы могут обеспечивать дополнительное поле данных в итоговом сообщении для переноса такой информации. Когда механизм задает передачу сервером дополнительной информации об успешном завершении обмена, протокол обеспечивает поле для передачи таких данных и сервер использует это поле, процесс обмена сокращается на один период кругового обхода:

```
C: Запрос аутентификационного обмена
S: Первоначальный запрос (challenge) сервера
C: Первоначальный отклик
<дополнительные сообщения challenge/response>
S: Результат аутентификационного обмена с дополнительными данными об успехе
```

Когда механизм задает возврат сервером дополнительных данных для клиента, а поле для таких данных недоступно или не используется, эти данные передаются как запрос, отклик на который будет пустым. После получения отклика сервер передает сообщение об успешном завершении.

```
C: Запрос аутентификационного обмена
S: Первоначальный запрос (challenge) сервера
C: Первоначальный отклик
<дополнительные сообщения challenge/response>
S: Дополнительный запрос (challenge)
C: Пустой отклик
S:  Результат аутентификационного обмена
```

Когда механизм задает, что первыми в процессе обмена передаются данные от клиента к серверу, клиенту передаются дополнительные данные об успешном завершении обмена и протокол поддерживает поля для передачи дополнительной информации, обмен сокращается на два периода кругового обхода:

```
C: Запрос аутентификационного обмена + Первоначальный отклик
<дополнительные сообщения challenge/response>
S: Результат аутентификационного обмена с дополнительными данными об успехе
```

вместо:

```
C: Запрос аутентификационного обмена
S: Пустой запрос (Challenge)
C: Первоначальный отклик
<дополнительные сообщения challenge/response>
S: Дополнительный запрос (challenge)
C: Пустой отклик
S:  Результат аутентификационного обмена
```

3.1. Именование механизмов

Механизмы SASL именуются символьными строками длиной от 1 до 20 знаков, содержащими буквы верхнего регистра кода ASCII [ASCII], цифры, знаки дефиса (-) и символы подчеркивания (_). В приведенной ниже форме ABNF[7] [RFC4234] <sasl-mech> определяет синтаксис имен механизмов SASL.

```
sasl-mech    = 1*20mech-char
mech-char    = UPPER-ALPHA / DIGIT / HYPHEN / UNDERSCORE
; mech-char может содержать только буквы A-Z (заглавные), 0-9, -, и _ из набора ASCII.

UPPER-ALPHA  = %x41-5A  ; A-Z (только заглавные)
DIGIT        = %x30-39  ; 0-9
HYPHEN       = %x2D ; hyphen (-)
UNDERSCORE   = %x5F ; underscore (_)
```

Регистрация имен механизмов SASL обсуждается в параграфе 7.1.

3.2. Согласование механизмов

Согласование механизмов зависит от протокола.

В общем случае протокол будет задавать анонсирование сервером поддерживаемых механизмов своим клиентам с помощью того или иного способа, обеспечиваемого протоколом. Клиент будет выбирать “лучший” механизм из числа поддерживаемых им и подходящих для данного случая.

Отметим, что согласование механизма не защищено последующим аутентификационным обменом и, следовательно, может служить объектом атак с целью выбора наименее надежного механизма[8], если не принять дополнительных мер защиты.

Для детектирования таких атак протокол может позволять клиенту определять доступные механизмы после завершения аутентификационного обмена и установки уровня защиты данных, обеспечивающего по крайней мере защиту целостности. Это позволит клиенту обнаружить изменения в списке механизмов, поддерживаемых сервером.

3.3. Запрос аутентификационного обмена

Аутентификационный обмен инициируется клиентом путем передачи запроса на аутентификацию с использованием указанного механизма. Клиент передает серверу сообщение, которое содержит имя механизма. Конкретный формат сообщений зависит от протокола.

Отметим, что имя механизма не защищено этим механизмом и может быть подменено атакующим, если не используется иных средств защиты.

Когда определен механизм, позволяющий клиенту первым передавать данные и протокольное сообщение для передачи запроса включает дополнительное поле начального отклика, клиент может включить отклик на изначальный запрос сервера (initial challenge) в свой запрос на аутентификацию.

[7] Augmented Backus-Naur Form – расширенная форма Бэкуса-Наура. *Прим. перев.*
[8] В оригинале “downgrade attack”. *Прим. перев.*

3.4. Запросы и отклики

Аутентификационный обмен включает одну или несколько пар запросов сервера (server-challenge) и откликов клиента (client-response), детали которых определяются используемым механизмом. Эти запросы и отклики передаются в протокольных сообщениях, формат которых определяется протоколом.

С помощью таких запросов и откликов механизм может:

- аутентифицироать клиента на сервере;
- аутентифицировать сервер на стороне клиента;
- передать строку идентификации для проверки полномочий (authorization identity);
- согласовать уровень защиты;
- обеспечить другие типы сервиса.

Согласование уровня защиты может включать согласование защитных служб, обеспечиваемых этим уровнем, способов обеспечения защиты, а также индикацию максимального размера буфера шифрованных данных на каждой стороне, который уровень способен принять (см. параграф 3.6).

После получения запроса на аутентификацию или любого отклика от клиента сервер может передать свой запрос (challenge), прервать обмен или указать на его завершение. После получения запроса механизм на стороне клиента может передать отклик или прервать обмен.

3.4.1. Строка идентификации для проверки полномочий

Строка идентификации для проверки полномочий (authorization identity string) представляет собой последовательность (возможно нулевой длины) символов Unicode [Unicode], за исключением символа NUL (U+0000).

Если строка идентификации отсутствует, это означает, что клиент запрашивает у сервера идентификацию, связанную с представленными клиентом свидетельствами (credentials). Пустая строка идентификации эквивалентна ее отсутствию.

Непустая строка идентификации показывает, что клиент хочет представиться. В этом случае форма идентификации, представляемая этой строкой, а также точный синтаксис и семантика строки определяются протоколом.

Когда используемая в аутентификационном обмене для передачи строк идентификации схема символьного представления зависит от механизма, предполагается, что механизм может передавать все символы Unicode (за исключением символа NUL).

3.5. Прерывание аутентификационного обмена

Клиент или сервер может принять решение о прерывании аутентификационного обмена, если они не хотят или не могут его продолжить (или начать).

Клиент может прервать аутентификационный обмен, передавая серверу специальное сообщение, формат которого зависит от протокола. Протокол может требовать от сервера передачи клиенту сообщения в ответ на запрос того оп прерывании обмена.

Подобно этому сервер может прервать аутентификационный обмен путем передачи клиенту специального сообщения, формат которого определяется протоколом.

3.6. Результат аутентификации

По завершении аутентификационного обмена сервер передает клиенту сообщение, показывающее результат обмена. Точный формат сообщения определяется протоколом.

Аутентификация завершается отказом, если

- аутентификационный обмен прерван по любой причине;
- клиентские свидетельства (credentials) не были подтверждены;
- сервер не может связать идентификацию со свидетельствами клиента;
- представленная сервером для проверки полномочий строка идентификации имеет некорректный формат;
- идентификация, связанная со свидетельствами клиента, не может использоваться в качестве идентификации при проверке полномочий;
- согласованный уровень защиты (или его отсутствие) не удовлетворяет требованиям;
- сервер по какой-либо причине не желает предоставлять свои услуги клиенту.

Протокол может включать дополнительное поле данных в сообщении о результате аутентификации. Это поле может включать дополнительную информацию лишь в тех случаях, когда аутентификация завершилась успешно.

Если результат аутентификации положительный и уровень защиты согласован, этот уровень устанавливается. Если аутентификация завершилась отказом или уровень защиты не был согласован, сохраняется текущий уровень защиты.

Передаваемое сервером сообщение о результате аутентификации может обеспечивать для клиента способ различать ошибки, при которых лучше всего повторно запросить у пользователя его свидетельства (credentials), от ошибок, при которых лучше всего рекомендовать клиенту обратиться к серверу позднее, или ошибок,при которых пользователю следует обратиться к системному администратору для решения проблем (см. коды откликов SYS и AUTH POP [RFC3206] в качестве примера). Такая возможность полезна, в частности, в периоды запланированного обслуживания сервера, поскольку она позволяет снизить расходы на поддержку. Важно также настроить сервер так, чтобы сообщения о результате аутентификации не позволяли отличить корректного пользователя с некорректными свидетельствами от некорректного пользователя.

3.7. Уровни защиты

Механизмы SASL могут предлагать широкий спектр средств защиты. Обычно сюда включается защита целостности и конфиденциальности данных. Механизмы SASL, которые не обеспечивают уровня защиты, трактуются как согласующие отсутствие уровня защиты.

Если использование уровня защиты согласовано в процессе аутентификационного обмена, этом уровень устанавливается сервером после передачи сообщения о результате аутентификации, а клиентом – при получении этого сообщения. В обоих случаях уровень устанавливается до передачи последующих протокольных данных. Точный момент начала использования уровня защиты для потоков данных протокола определяется самим протоколом.

После того, как уровень защиты вступает в силу для потоков данных протокола, он остается в действии до тех пор, пока не будет установлен заново согласованный уровень защиты или нижележащее транспортное соединение не будет разорвано.

Когда уровень защиты активен, он переносит протокольные данные в буферы защищенных данных. Если в какой-то момент уровень защиты не пожелает или не сможет продолжить создание буферов, защищающих протокольные данные, нижележащее транспортное соединение **должно** быть разорвано. Если уровень защиты не способен декодировать приемный буфер, нижележащее транспортное соединение **должно** быть разорвано. В обоих случаях транспортное соединение **следует** закрывать корректно.

Каждый буфер защищенных данных передается через нижележащее транспортное соединение в виде последовательности октетов, которой предшествует 4-октетное поле с сетевым порядком байтов, которое представляет размер буфера. Размер буфера защищенных данных **должен** быть не больше максимального размера, который ожидает принять другая сторона. При получении данных, размер которых превышает максимально допустимый, приемной стороне **следует** закрыть соединение и она может трактовать это как атаку.

Максимальный размер, который ожидает получить каждая сторона, определяется используемым механизмом (согласуется или задается спецификацией).

3.8. Множественная аутентификация

Если протоколом явно (в технической спецификации данного протокола) не указано иное, в протокольной сессии допускается лишь одна успешная аутентификация SASL. После успешного завершения аутентификации дальнейшие попытки организации аутентификационного обмена приводят к отказу.

Когда протокол разрешает многократную аутентификацию SASL, может выполняться множество сеансов аутентификационного обмена, но ни в коем случае одновременно не может использоваться множество уровней защиты. Если уровень защиты уже используется и при следующем согласовании SASL выбирается другой уровень, новый уровень будет использоваться взамен старого. Если уровень защиты используется, а при новом согласовании SASL выбрано отсутствие защитного уровня, сохранится прежний уровень защиты.

Когда протокол разрешает многократную успешную аутентификацию SASL, влияние отказа при последующей попытке аутентификации SASL на ранее проведенную аутентификацию и проверку полномочий определяется протоколом. Следует обратиться к технической спецификации протокола, чтобы определить следует сохранять ранее аутентифицированное состояние, переходить в состояние для анонимного пользователя или принимать иные меры. Независимо от принятого для протокола влияния на предыдущее состояние, согласованный ранее уровень защиты сохраняется.

4. Требования к протоколам

Для того, чтобы протокол мог предлагать сервис SASL, его спецификация **должна** содержать следующую информацию:

1) Имя сервиса, которое может быть выбрано из регистра элементов "service" для связанного с хостом имени сервиса GSSAPI[9], как описано в параграфе 4.1 документа [RFC2743]. Отметим, что этот регистр совместно используется всеми механизмами GSSAPI и SASL.

2) Детали всех аспектов согласования механизма, обеспечиваемых протоколом (см. параграф 3.2).

   Протоколу следует задавать средства, с помощью которых клиент может (как до начала обмена SASL, так и после установки уровня защиты в результате обмена) имена механизмов SASL, которые сервер делает доступными для клиента. Это важно для обеспечения клиенту возможности детектирования атак, направленных на снижение уровня защиты. Такие средства обычно обеспечиваются с помощью функций детектирования расширений протокола и согласования возможностей.

3) Определения сообщений, требуемых при аутентификационном обмене, включая следующие:

   a) сообщения для запроса аутентификационного обмена (см. параграф 3.3).

      Такие сообщения **должны** содержать поле для передачи имени механизма, выбранного клиентом.

      В такие сообщения **следует** включать необязательное поле для передачи начального отклика. Если сообщение определено с таким полем, спецификация **должна** описывать как сообщения с пустым начальным откликом отличаются от сообщений без такого отклика. Это поле **должно** обеспечивать возможность передачи произвольной последовательности октетов (включая последовательности нулевой длины и последовательности с нулевыми октетами).

   b) Сообщения для передачи запросов (challenge) сервера и откликов (response) клиента (см. параграф 3.4).

      Каждое из таких сообщений **должно** обеспечивать возможность передачи произвольных последовательностей октетов (включая последовательности нулевой длины и последовательности с нулевыми октетами).

   c) Сообщение для индикации результата аутентификационного обмена (см. параграф 3.6).

      В такие сообщения **следует** включать необязательное поле для передачи дополнительных данных при успешном завершении обмена. Если для сообщения определено такое поле, спецификация **должна** описывать способ различить сообщения с пустым полем дополнительной информации от сообщений без такого поля. Поде **должно** обеспечивать возможность передачи произвольной последовательности октетов (включая последовательности нулевой длины и последовательности с нулевыми октетами).

[9] Generic Security Service Application Program Interface – базовый программный интерфейс служб защиты. *Прим. перев.*

4) Описание синтаксиса и семантики непустых строк идентификации при проверке полномочий (см. параграф 3.4.1).

   Чтобы избавиться от проблем интероперабельности в результате использования разной нормализации, спецификация протокола **должна** точно описывать как и где (клиент или сервер) готовятся непустые строки идентификации при проверке полномочий, включая любую нормализацию, для операций сравнения и других функций, обеспечивающих корректную работу.

   Рекомендуется включать в спецификацию описание применения существующих форм идентификации при проверке полномочий, а также существующих вариантов представления строк таких, как обычные имена пользователей [RFC4013].

   Когда спецификация не описывает точно как идентификация в SASL связана с идентификацией, используемой другими компонентами протокола (например, в правилах контроля доступа), для протокола может оказаться полезным поддержка для клиентов механизма обнаружения информации (например, представление идентификации, используемой при решении вопроса о предоставлении доступа) об имеющихся способах идентификации для таких целей.

5) Детали всех способов, которые позволяют клиенту и/или серверу прервать аутентификационный обмен (см. параграф 3.5).

   Протоколы, поддерживающие множественную аутентификацию, обычно позволяют клиенту прервать происходящий аутентификационный обмен путем инициирования нового обмена. Протоколы, которые не поддерживают множественную аутентификацию, могут требовать от клиента закрывать соединение и открывать новое для прерывания происходящего аутентификационного обмена.

   Протоколы обычно позволяют серверу прерывать происходящий аутентификационный обмен путем возврата сообщения о неудачной аутентификации.

6) Точного указания момента, с которого начинает использоваться вновь согласованный уровень защиты для обоих направлений (см. параграф 3.7).

   Обычно спецификации требуют начинать использование уровня защиты, начиная с первого октета, передаваемого после результата аутентификации, для сообщений от сервера и первого октета, передаваемого после приема сообщения о результате аутентификации, для клиента.

7) Если протокол поддерживает другие службы обеспечения уровня защиты (такие, как TLS[10] [RFC4346], спецификация **должна** описывать, как эти уровни применяются к данным протокола.

   Например, когда протокол поддерживает уровни защиты TLS и SASL, спецификация может указать любой из перечисленных ниже вариантов:

   a) уровень защиты SASL всегда применяется первым для передаваемых данных и последним для принимаемых;

   b) уровень защиты SASL всегда применяется последним для передаваемых данных и первым для принимаемых;

   c) уровни защиты применяются в порядке их установки;

   d) уровни защиты применяются в порядке, обратном порядку их установки;

   e) оба уровня TLS и SASL не могут быть установлены.

8) Индикация поддержки множественной аутентификации (см. параграф 3.8). Если такая аутентификация поддерживается, **должно** быть детально описано поведение при отказе от аутентификации SASL для тех случаев, когда предыдущая аутентификация была завершена успешно.

В спецификации протокола **следует** избегать требований к реализации, препятствующих замене применимых механизмов. В общем случае спецификации протокола **следует** быть нейтральной по отношению к механизмам. Однако из этого правила существует множество исключений, в том числе перечисленные ниже:

- детализация того, как в в протоколе происходит управление свидетельствами (которые связаны с механизмами);
- детализация того, как аутентификационная идентификация (определяется механизмом) и идентификация при проверке полномочий (определяется протоколом) связаны одна с другой;
- детализация применимости механизмов для данного протокола.

5. Требования к механизмам

Спецификации механизмов SASL **должны** обеспечивать следующую информацию:

1) Имя механизма (см. параграф 3.1). Это имя **должно** быть зарегистрировано, как описано в параграфе 7.1.

2) Определение запросов сервера (server-challenge) и откликов клиента (client-response) в процессе аутентификационного обмена, а также:

   a) Индикацию того, начинает работу механизма клиент (client-first), сервер (server-first) или это может изменяться (variable). Если механизм SASL определен, как client-first, а клиент не передает начального отклика в запросе на аутентификацию, первый запрос (challenge) сервера **должен** быть пустым (механизм EXTERNAL является примером такого случая). Если механизм SASL определен, как variable, спецификация должна указывать поведение сервера для тех случаев, когда начальный отклик в клиентском запросе на аутентификацию опущен (механизм DIGEST-MD5 [DIGEST-MD5] является примером для этого случая). Если механизм SASL определен, как server-first, для клиента **недопустимо** передавать начальный отклик в запросе на аутентификацию (примером для этого случая может служить механизм CRAM-MD5 [CRAM-MD5]).

   b) Индикация того, предполагается ли со стороны сервера предоставление дополнительных данных, показывающих успех при аутентификации. Если такие данные ожидаются и сервер шлет их как запрос (challenge), спецификация **должна** показывать, что в ответ на такой запрос передается пустой отклик.

   При разработке механизмов SASL **следует** минимизировать число запросов и откликов, требуемых для выполнения аутентификационного обмена.

[10] Transport Layer Security – защита на транспортном уровне. *Прим. перев.*

3) Идентификация возможности поддержки механизмом передачи строк идентификации при проверке полномочий (см. параграф 3.4.1). Хотя некоторые унаследованные механизмы не способны передавать строки authorization identity (это означает, что для данного механизма authorization identity во всех случаях является пустой строкой), новым механизмам **следует** поддерживать передачу строк идентификации при проверке полномочий. Механизмам **не следует** поддерживать одновременно передачу отсутствия строк authorization identity и передачу пустых строк идентификации при проверке полномочий.

   Механизмы, способные передавать строки authorization identity, **должны** поддерживать возможность передачи произвольных непустых последовательностей символов Unicode, включая строки с символами NUL (U+0000). Механизмам **следует** использовать формат преобразования UTF-8 [RFC3629]. В спецификации **должно** быть указано, каким образом включаются в строки идентификации все последовательности символов, имеющие специальное значение для данного механизма, во избежание неоднозначности при декодировании строк authorization identity. Обычно механизмы, использующие специальные символы, требуют для таких символов добавки специального escape-символа или представления таких символов в виде последовательности (после преобразования в заданный формат Unicode) с использованием схем кодирования данных типа Base64 [RFC3548].

4) Спецификация **должна** указывать, предлагает ли механизм уровень защиты. Если механизм предлагает такой уровень, спецификация **должна** детально описывать защиту и другие услуги, обеспечиваемые этим уровнем, а также способы реализации сервиса.

5) Если используемая механизмом криптографическая технология поддерживает защиту целостности данных, спецификация механизма **должна** защищать целостность при передаче строк authorization identity и согласовании уровня защиты.

Механизмам SASL **следует** быть нейтральными по отношению к протоколам.

Механизмам SASL **следует** использовать существующие формы свидетельств (credential) и идентификации (identity), а также связанный с ними синтаксис и семантику.

Механизмам SASL **следует** использовать формат преобразования UTF-8 [RFC3629] для представления передаваемых кодов (code point) Unicode [Unicode].

Во избежание проблем интероперабельности в результате различной нормализации, когда механизм вызывается для символьных данных (отличных от строк authorization identity), которые будут использоваться в качестве входной информации для криптографических функций и/или функций сравнения, спецификация **должна** подробно описывать когда и где готовятся символьные данные (включая любую нормализацию) для передачи их функциям, чтобы обеспечить корректную работу.

Для простых имен пользователей и паролей в аутентификационных свидетельствах в качестве алгоритма подготовки **следует** указывать SASLprep [RFC4013] (профиль алгоритма подготовки StringPrep [RFC3454]).

Механизмам **не следует** использовать строки authorization identity при генерации каких-либо долгоживущих криптографических ключей или хэш-значений, поскольку не существует требования, по которому строки authorization identity должны быть каноническими. Долгосрочный в данном случае означает существование в течение срока, превышающего продолжительность аутентификационного обмена. Т. е., поскольку разные клиенты (одного или различных протоколов) могут предоставлять различные строки authorization identity, которые семантически будут эквивалентны, использование таких строк для генерации криптографических ключей и хэш-значений может приводить к проблемам интероперабельности и иным сложностям.

6. Вопросы безопасности

Вопросы безопасности обсуждаются на протяжении всего документа.

Многие из существующих механизмов SASL не обеспечивают адекватной защиты от пассивных атак, блокируя лишь активные атаки в процессе аутентификационного обмена. Многие из существующих механизмов SASL не поддерживают уровня защиты. Есть надежда, что новые механизмы SASL будут обеспечивать сильную защиту от пассивных атак, активных атак в процессе аутентификационного обмена, а также защитные уровни с сильными базовыми средствами защиты данных (например, защита целостности и конфиденциальности). Есть надежда и на то, что будущие механизмы обеспечат более эффективные средства защиты типа регенерации ключей (см. параграф 6.3).

Кроме того, схема SASL чувствительна к атакам, направленным на снижение уровня защиты (downgrade attack). В параграфе 6.1.2 рассматриваются различные способы детектирования и предотвращения таких атак. В некоторых случаях можно воспользоваться внешними по отношению к SASL средствами защиты целостности данных (например, TLS) для защиты от атак, направленных на снижение уровня защиты в SASL. Применение внешних средств защиты важно также для тех случаев, когда доступные механизмы сами по себе не обеспечивают адекватной защиты целостности и/или конфиденциальности аутентификационного обмена и/или данных протокола.

6.1. Активные атаки

6.1.1. Перехват

Когда клиент выбирает уровень защиты SASL, обеспечивающий по крайней мере защиту целостности, такая защита работает как средство против активных атак с перехватом соединений и модификацией данных протокола, передаваемых после организации защитного уровня. Реализациям **следует** разрывать соединение, когда службы уровня защиты SASL сообщают о потере целостности данных.

6.1.2. Атаки, направленные на снижение уровня защиты

Важно, чтобы все чувствительные к защите согласования протокола выполнялись после организации уровня защиты с обеспечением целостности данных. Протоколы следует разрабатывать так, чтобы согласование, выполненное до установки защитного уровня, можно было проверить после организации этого уровня. Согласование механизмов SASL относится к числу таких согласований.

Когда клиент согласует с сервером механизм аутентификации и/или другие функции защиты, активный атакующий может вынудить к выбору наименее защищенного варианта сервиса из числа возможных. Например, атакующий может подменить список анонсируемых сервером механизмов или список поддерживаемых клиентом функций в отклике с выбором механизма. Для защиты от этого типа атак реализациям **не следует** анонсировать механизмы и/или функции, которые не соответствуют минимальным требованиям безопасности, а также **не следует** начинать или продолжать аутентификационный обмен, который не

удовлетворяет минимальным требованиям безопасности. Кроме того, **следует** проверять, что результат аутентификационного обмена в части защиты соответствует требованиям безопасности. Отметим, что каждая из конечных точек должна независимо проверять выполнение требований безопасности.

Для детектирования атак, направленных на снижение уровня защиты до минимального из числа поддерживаемых (или ниже), клиент может определить поддерживаемые сервером механизмы SASL до аутентификационного обмена и после установки согласованного уровня защиты SASL (обеспечивающего по крайней мере целостность данных) с помощью механизма определения возможностей, предоставляемого протоколом. Если клиент видит, что полученный с обеспечением целостности список (полученный после создания защитного уровня), предлагает более сильный механизм, нежели содержатся в полученном ранее списке, клиенту разумно предположить, что первый список был изменен атакующим и **следует** закрыть нижележащее соединение транспортного уровня.

Инициированный клиентом обмен SASL, включающий выбор механизма SASL, происходит в открытой форме и может быть изменен активным атакующим. При разработке новых механизмов SASL важно обеспечить предотвращение вынужденного атакующим выбора наименее надежного варианта защиты путем простого изменения имени механизма SASL и/или запросов и откликов.

Многоуровневое согласование функций защиты более открыто для атак, направленных на снижение уровня. Разработчикам протоколов следует избегать в протоколах предложения использовать согласование функций защиты на верхних уровнях (например, “поверх” согласования механизма SASL), а разработчикам механизмов следует избегать низкоуровневого согласования защитных функций в своих механизмах (например, ниже согласования механизма SASL).

6.1.3. Атаки с воспроизведением данных

Некоторые механизмы могут быть подвержены воздействию атак на основе воспроизведения ранее собранных данных (replay attack), если не используются внешние механизмы защиты данных (например, TLS).

6.1.4. Атаки с отсечением

Многие из существующих механизмов SASL сами по себе не обеспечивают защиты от атак с отсечением. В таких атаках активный атакующий вынуждает закрыть протокольную сессию, что ведет к блокированию возможности использования потоков данных с защитой целостности в результате чего поведение одной или обеих сторон соединения может давать атакующему недопустимые преимущества. От атак этого типа достаточно просто защитить ориентированные на соединения протоколы прикладного уровня. Протокол может защитить себя от таких атак, обеспечивая для каждого информационного обмена явную проверку результата, проверяя корректность закрытия каждой протокольной сессии и применения защиты целостности.

6.1.5. Другие активные атаки

При использовании уровня защиты согласованного в процессе аутентификационного обмена, получателю **следует** быть очень аккуратным с буферами защищенных данных размером больше заданного или согласованного максимума. В частности, **недопустимо** выделять, не глядя, количество памяти, указанное в поле размера буфера, поскольку это может привести к нехватке памяти (out of memory). При обнаружении большого блока получателю **следует** закрыть соединение.

6.2. Пассивные атаки

Многие механизмы могут быть объектами различных пассивных атак, включая простое подслушивание незащищенной информации, используемой при аутентификации (credential information), а также объектами атак с использованием словарей по отношению к защищенным данным, используемым при аутентификации.

6.3. Замена ключей

Безопасные или административно разрешенные значения времени жизни уровней защиты в механизмах SASL конечны. Эффективность криптографических ключей снижается с течением времени при их использовании – наличие времени и/или зашифрованного текста, который криптоаналитик получает при первом использовании ключа упрощает анализ и организацию атак.

Административные ограничения на продолжительность существования защитного уровня могут задаваться в форме сертификатов X.509, ярлыков Kerberos V или каталогов. Такие ограничения часто весьма желательны. На практике же очевидным результатом административного ограничения срока жизни является то, что приложение может столкнуться с прекращением работы уровня защиты во время операции прикладного уровня (например, при передаче большого объема данных). В результате соединение будет закрыто (см. параграф 3.7), что приведет к негативной реакции пользователей.

Замена ключей (Re-keying – процесс повторного согласования ключей) является способом решения проблемы. Схема SASL сама по себе не обеспечивает замены ключей, но это могут делать механизмы SASL. Разработчикам новых механизмов SASL следует обратить внимание на вопрос замены ключей.

Реализациям, которые пожелают регенерировать ключи уровня защиты SASL в тех случаях, когда механизм не обеспечивает замены ключа, **следует** заново аутентифицировать те же идентификаторы и заменить уровень защиты, срок работы которого истек или близок к завершению. Этот вариант требует поддержки повторной аутентификации в протоколах прикладного уровня (см. параграф 3.8).

6.4. Прочие вопросы

При разработке и реализации протоколов следует обращать внимание на вопросы безопасности механизмов, чтобы выбирать механизмы, соответствующие требованиям задачи.

При реализации распределенных серверов нужно аккуратно отнестись к вопросу доверия другим компонентам сервера. В частности, аутентификационные данные (secret) следует открывать только тем компонентам сервера, которым разрешено управлять этими данными и использовать их тем способом, который приемлем для открывающей стороны. Приложения, использующие SASL, предполагают, что уровни защиты SASL обеспечивают конфиденциальность данных и защиту даже в тех случаях, когда атакующий выбирает текст, который будет защищен этим уровнем. Приложения также полагают, что уровень защиты SASL безопасен даже в тех случаях, когда атакующий может манипулировать результатами шифрования на защитном уровне. Предполагается, что новые механизмы SASL соответствуют этим предположения.

Вопросы безопасности Unicode [UTR36] имеют отношение к строкам authorization identity, а при использовании UTF-8 становятся актуальными и вопросы безопасности UTF-8 [RFC3629]. Следует принимать во внимание и вопросы безопасности SASLprep [RFC4013] и StringPrep [RFC3454] в тех случаях, когда эти алгоритмы применяются.

7. Согласование с IANA

7.1. Реестр механизмов SASL

Реестр механизмов SASL поддерживается агентством IANA. В настоящее время реестр доступен по адресу http://www.iana.org/assignments/sasl-mechanisms.

Задачей этого реестра является не только обеспечение уникальности значений, используемых для именования механизмов SASL, но и предоставление ссылок на технические спецификации, определяющие каждый механизм SASL и доступные через Internet.

Для механизмов SASL не существует соглашений по именованию и любое имя, соответствующее синтаксису имен механизмов SASL, может быть зарегистрировано.

Для регистрации имени отдельного механизма используется процедура, описанная в параграфе 7.1.1.

Для регистрации семейства имен связанных между собой механизмов используется процедура, описанная в параграфе 7.1.2.

Вопросы включения комментариев в реестр рассмотрены в параграфе 7.1.3, а вопросы изменения имеющихся комментариев – в параграфе 7.1.4.

Реестр механизмов SASL был обновлен с внесением данного документа как технической спецификации SASL и указанием данного параграфа в качестве описания процедур регистрации в реестре механизмов.

7.1.1. Процедура регистрации имен механизмов

IANA будет регистрировать новые имена механизмов SASL в порядке подачи заявок (процедура First Come First Served), как описано в BCP 26 [RFC2434]. IANA имеет право отвергать явно фиктивные запросы на регистрацию, но не будет рассматривать заявления прав, сделанные в регистрационной форме.

Регистрация имен механизмов SASL осуществляется путем заполнения приведенной ниже формы

Тема: регистрация механизма SASL с именем X

Имя механизма SASL (или префикс для семейства имен):

Вопросы безопасности:

Опубликованная спецификация (рекомендуется):

Контактное лицо и адрес электронной почты для обмена информацией:

Предполагаемое использование: (Один из вариантов COMMON, LIMITED USE, OBSOLETE)

Владелец/контролер изменений:

Примечания: (Здесь может быть добавлена любая информация, которую автор сочтет нужной).

и ее отправки по электронной почте в IANA по адресу iana@iana.org.

Хотя эта процедура не требует экспертного обзора, авторам механизмов SASL настоятельно рекомендуется получить отклики и комментарии сетевого сообщества в тех случаях, когда это возможно. Авторы могут получить отклики сетевого сообщества, опубликовав спецификацию предлагаемого механизма в качестве Internet-Draft. Механизмы SASL, рассчитанные на общее применение, следует стандартизировать с использованием нормального процесса IETF в тех случаях, когда это возможно.

7.1.2. Процедура регистрации имен семейств

Как было отмечено выше, для механизмов SASL не существует общего соглашения об именовании. Однако спецификация может резервировать часть пространства имен механизмов SASL для набора связанных между собой механизмов SASL (семейства механизмов SASL). Каждое семейство механизмов SASL обозначается уникальным префиксом типа X-. регистрация нового семейства имен механизмов SASL требует экспертного обзора в соответствии с BCP 26 [RFC2434].

Регистрация имен семейств механизмов SASL осуществляется путем заполнения приведенной ниже формы

Тема: регистрация семейства механизмов SASL с именем X

Имя семейства SASL (или префикс для семейства имен):

Вопросы безопасности:

Опубликованная спецификация (рекомендуется):

Контактное лицо и адрес электронной почты для обмена информацией:

Предполагаемое использование: (Один из вариантов COMMON, LIMITED USE, OBSOLETE)

Владелец/контролер изменений:

Примечания: (Здесь может быть добавлена любая информация, которую автор сочтет нужной).

И ее отправки по электронной почте в конференцию IETF SASL по адресу ietf-sasl@imc.org с копией по адресу iana@iana.org.

После двухнедельного рассмотрения комментариев сетевого сообщества в почтовой конференции IETF SASL эксперт будет решать вопрос о возможности регистрации имени семейства и одобрять или отвергать запрос, уведомляя об этом запрашивающую сторону и IANA, а также помещая соответствующее сообщение в список рассылки.

Обозревателю следует обратить внимание на применимость запрошенного имени семейства с учетом предполагаемого использования и планами регистрации для имеющихся и будущих имен механизмов данного семейства. При решении вопроса

эксперт может принимать во внимание имеющие отношение к делу аспекты любой представленной технической спецификации (такой, как раздел IANA Considerations). Однако этот обзор сфокусирован на уместности запрошенной регистрации и не связан деталями технической спецификации.

Авторам настоятельно рекомендуется получить отзывы и комментарии сетевого сообщества, опубликовав технические спецификации как Internet-Draft и поместив их в подходящие почтовые конференции IETF.

7.1.3. Комментарии к регистрации механизмов SASL

Комментарии к зарегистрированным механизмам и семействам SASL следует сначала направить “владельцу” имени или семейства и/или в почтовую конференцию по адресу ietf-sasl@imc.org.

Автор комментариев может после разумного числа попыток контакта с владельцем запросить в IANA включение своих комментариев в регистрацию этого механизма SASL, направив письмо по адресу iana@iana.org. По своему разумению IANA может включить комментарии в регистрацию механизма SASL.

7.1.4. Контроль изменений

После того, как регистрация механизма SASL опубликована IANA, автор может запросить изменение зарегистрированного определения. Запрос на изменение осуществляется в соответствии с такой же процедурой, как использовалась при регистрации.

Владелец механизма SASL может передать ответственность за этот механизм другому лицу или организации, уведомив об этом IANA; такая передача не требует обсуждения или обзора.

IESG может сменить ответственного за механизм SASL. Чаще всего это делается в тех случаях, когда требуется внести изменения в механизм, автор которого умер или по каким-либо причинам недоступен или не может внести изменения, требуемые для сетевого сообщества.

Регистрация механизмов SASL не может быть отменена; механизмы, которые перестали быть приемлемыми для использования, помечаются как OBSOLETE в поле "intended usage"; такие механизмы SASL явно маркируются в списках, публикуемых IANA.

IESG рассматривается как владелец механизмов SASL, имеющих статус IETF.

7.2. Изменение регистрации

Агентство IANA внесло в реестр механизмов SASL следующие изменения:

1) Изменено поле "Intended usage" для механизмов KERBEROS_V4 и SKEY на OBSOLETE.
2) Изменено поле "Published specification" для механизма EXTERNAL с указанием данной спецификации, как показано ниже:

   Subject: Updated Registration of SASL mechanism EXTERNAL

   Family of SASL mechanisms: NO

   SASL mechanism name: EXTERNAL

   Security considerations: See A.3 of RFC 4422

   Published specification (optional, recommended): RFC 4422

   Person & email address to contact for further information:

   Alexey Melnikov <Alexey.Melnikov@isode.com>

   Intended usage: COMMON

   Owner/Change controller: IESG <iesg@ietf.org>

   Note: Updates existing entry for EXTERNAL

8. Литература

8.1. Нормативные документы

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119[11], March 1997.

[RFC2244] Newman, C. and J. G. Myers, "ACAP – Application Configuration Access Protocol", RFC 2244, November 1997.

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[RFC2743] Linn, J., "Generic Security Service Application Program Interface Version 2, Update 1", RFC 2743, January 2000.

[RFC3454] Hoffman, P. and M. Blanchet, "Preparation of Internationalized Strings ("stringprep")", RFC 3454, December 2002.

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

[RFC4013] Zeilenga, K., "SASLprep: Stringprep Profile for User Names and Passwords", RFC 4013, February 2005.

[RFC4234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", RFC 4234[11], October 2005.

[ASCII] Coded Character Set--7-bit American Standard Code for Information Interchange, ANSI X3.4-1986.

[Unicode] The Unicode Consortium, "The Unicode Standard, Version 3.2.0" is defined by "The Unicode Standard, Version 3.0" (Reading, MA, Addison-Wesley, 2000. ISBN 0-201-61633-5), as amended by the "Unicode Standard Annex #27: Unicode 3.1" (http://www.unicode.org/reports/tr27/) and by the "Unicode Standard Annex #28: Unicode 3.2" (http://www.unicode.org/reports/tr28/).

[11]На сайте http://www.protocols.ru имеется перевод этого документа на русский язык. *Прим. перев.*

[CharModel] Whistler, K. and M. Davis, "Unicode Technical Report #17, Character Encoding Model", UTR17, <http://www.unicode.org/unicode/reports/tr17/>, August 2000.

[Glossary] The Unicode Consortium, "Unicode Glossary", <http://www.unicode.org/glossary/>.

8.2. Дополнительная литература

[RFC3206] Gellens, R., "The SYS and AUTH POP Response Codes", RFC 3206, February 2002.

[RFC3548] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 3548, July 2003.

[RFC4301] Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

[RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006.

[SASL-GSSAPI] Melnikov, A. (Editor), "The Kerberos V5 ("GSSAPI") SASL Mechanism", Work in Progress, May 2006.

[UTR36] Davis, M., "(Draft) Unicode Technical Report #36, Character Encoding Model", UTR17, <http://www.unicode.org/unicode/reports/tr36/>, February 2005.

[CRAM-MD5] Nerenberg, L., "The CRAM-MD5 SASL Mechanism", Work in Progress.

[DIGEST-MD5] Leach, P., C. Newman, and A. Melnikov, "Using Digest Authentication as a SASL Mechanism", Work in Progress, March 2006.

9. Благодарности

Этот документ является пересмотром RFC 2222, автором которого является John Myers.

Документ является результатом работы группы IETF Simple Authentication and Security Layer (SASL).

Перечисленные здесь люди внесли значительный вклад в подготовку документа: Abhijit Menon-Sen, Hallvard Furuseth, Jeffrey Hutzelman, John Myers, Luke Howard, Magnus Nystrom, Nicolas Williams, Peter Saint-Andre, RL 'Bob' Morgan, Rob Siemborski, Sam Hartman, Simon Josefsson, Tim Alsop, Tony Hansen.

Приложение A. Механизм SASL EXTERNAL

Данное приложение является нормативным.

Механизм EXTERNAL позволяет клиенту запросить у сервера использования свидетельств (credentials), созданных с использованием внешних по отношению к механизму средств, для аутентификации клиента. Внешними средствами могут быть, например, службы IP Security [RFC4301] или TLS [RFC4346]. В отсутствие некой предварительной договоренности между клиентом и сервером клиент не может делать каких-либо предположений об использовании сервером внешних средств для получения клиентских свидетельств или о форме этих свидетельств. Например, клиент не может предполагать, что сервер будет использовать свидетельства, созданные клиентом с помощью TLS.

A.1. Техническая спецификация механизма EXTERNAL

Название механизма - "EXTERNAL".

Механизм не обеспечивает защитного уровня.

Механизм может передавать строки authorization identity. Пустая строка означает, что клиент запрашивает у сервера использование для него идентификации, которую сервер связывает с клиентскими свидетельствами. Непустая строка говорит о том, что клиент запрашивает идентификацию с использованием этой строки.

Предполагается, что клиент первым передает данные в процессе аутентификационного обмена. Когда клиент не включает начальный отклик в запрос на организацию аутентификационного обмена, сервер принимает на ответственность за передачу сообщения с пустым начальным запросом (challenge), на который клиент будет давать начальный отклик.

Клиент передает начальный отклик, содержащий строку идентификации для проверки полномочий в кодировке UTF-8 [RFC3629]. Этот отклик будет непустым, когда клиент запрашивает у сервера аутентификацию с использованием представленной (непустой) строки. Пустой отклик будет передаваться в тех случаях, когда клиент запрашивает у сервера аутентификацию с использованием аутентификационных свидетельств (authentication credentials).

Синтаксис начального отклика задается как значение <extern-initial-resp> описанное ниже в формате ABNF [RFC4234].

```
external-initial-resp = authz-id-string
authz-id-string       = *( UTF8-char-no-nul )
UTF8-char-no-nul      = UTF8-1-no-nul / UTF8-2 / UTF8-3 / UTF8-4
UTF8-1-no-nul         = %x01-7F
```

<UTF8-2>, <UTF8-3> и <UTF8-4> определены в [RFC3629].

Механизм не использует дополнительных запросов (challenge) и откликов.

Следовательно, сервер возвращает результат аутентификационного обмена.

Обмен завершается отказом в следующих случаях:

- клиент не создал свидетельств с использованием внешних средств;
- клиент использует неподходящие свидетельства;
- клиент передал пустую строку authorization identity, а сервер не желает или не может связать аутентификационную идентификацию с клиентскими свидетельствами;
- клиент представил непустую строку authorization identity, которая некорректна с точки зрения синтаксических требований, предъявляемых спецификацией прикладного протокола;

- клиент представил непустую строку authorization identity, представляющую идентификацию, которая не разрешена для клиента;
- сервер не желает или не может обслужить клиента по той или иной причине.

В остальных случаях аутентификационный обмен завершается успешно. При индикации успешного завершения дополнительных данных не передается.

A.2. Примеры SASL EXTERNAL

В этом параграфе рассматриваются примеры аутентификационного обмена с использованием механизма EXTERNAL. Эти примеры рассчитаны на то, чтобы помочь читателям понять приведенный выше текст. Примеры не являются определяющими. В примерах используется протокол ACAP[12] [RFC2244].

Первый пример показывает использование механизма EXTERNAL с пустой строкой authorization identity. В этом примере начальный отклик не включается в запрос клиента на организацию аутентификационного обмена.

```
S: * ACAP (SASL "DIGEST-MD5")
C: a001 STARTTLS
S: a001 OK "Begin TLS negotiation now"
<согласование TLS, остальные команды выполняются на уровне TLS>
S: * ACAP (SASL "DIGEST-MD5" "EXTERNAL")
C: a002 AUTHENTICATE "EXTERNAL"
S: + ""
C: + ""
S: a002 OK "Authenticated"
```

Второй пример показывает использование механизма EXTERNAL со строкой authorization identity "fred@example.com". В этом примере клиентский запрос на организацию аутентификационного обмена содержит начальный отклик, Такой подход экономит время на один период кругового обхода.

```
S: * ACAP (SASL "DIGEST-MD5")
C: a001 STARTTLS
S: a001 OK "Begin TLS negotiation now"
<согласование TLS, остальные команды выполняются на уровне TLS >
S: * ACAP (SASL "DIGEST-MD5" "EXTERNAL")
C: a002 AUTHENTICATE "EXTERNAL" {16+}
C: fred@example.com
S: a002 NO "Cannot assume requested authorization identity"
```

A.3. Вопросы безопасности

Механизм EXTERNAL не обеспечивает защиты; он уязвим по отношению к подстановкам (spoofing) со стороны клиента или сервера, активным атакам и подслушиванию. Этот механизм следует использовать лишь в тех случаях, когда имеется надежная защита.

Приложение B. Изменения по отношению к RFC 2222

Это приложение не является нормативным.

Текст RFC 2222 при создании этого документа был в значительной степени переработан.

RFC 2222 не указывал, что текст строк authorization identity использует символы Unicode, позволяя лишь символьные данные, подразумевающие, что authorization identity представляет собой строку октетов.

- Строки authorization identity сейчас определены как строки символов Unicode. Использование символа NUL (U+0000) не допускается. Хотя за определение формы строк authorization identity, синтаксис строк Unicode и связанную с ними семантику отвечает спецификация протокола, спецификация механизма несет ответственность за определение способа передачи строк Unicode в сеансе аутентификационного обмена.
- Удалена фраза "If so, when the client does not send data first, the initial challenge MUST be specified as being an empty challenge.[13]"

В механизм EXTERNAL были внесены следующие технические изменения:

- Строки authorization identity используют кодировку UTF-8.

Отметим, что были существенно усилены требования к спецификациям протоколов и механизмов. Существующие протоколы и механизмы должны быть обновлены с учетом этих требования.

Адреса редакторов

Alexey Melnikov

Isode Limited

5 Castle Business Village

36 Station Road

Hampton, Middlesex,

TW12 2BX, United Kingdom

[12]Application Configuration Access Protocol

[13]Если так, то в тех случаях, когда клиент не начинает первым передачу данных, изначальный запрос (challenge) сервера **должен** быть задан как пустой запрос. *Прим. перев.*

EMail: Alexey.Melnikov@isode.com

URI: http://www.melnikov.ca/

Kurt D. Zeilenga

OpenLDAP Foundation

EMail: Kurt@OpenLDAP.org

Перевод на русский язык

Николай Малых

nmalykh@bilim.com

Полное заявление авторских прав

Интеллектуальная собственность

Подтверждение

Финансирование функций RFC Editor обеспечено IETF Administrative Support Activity (IASA).