Network Working Group
Request for Comments: 2554
Category: Standards Track

J. Myers
Netscape Communications

March 1999

Расширение сервиса SMTP для аутентификации

SMTP Service Extension for Authentication

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.

Авторские права

Copyright (C) The Internet Society (1999). All Rights Reserved.

1. Введение

Этот документ определяет расширение сервиса SMTP [ESMTP], посредством которого клиент SMTP может указать серверу механизм аутентификации, провести обмен данными протокола аутентификации и дополнительно согласовать уровень защиты для последующих транзакций протокола. Это расширение является вариантом SASL[1] [SASL].

2. Используемые в документе соглашения

В примерах строки "C:" и "S:" показывают строки, передаваемые клиентом и сервером, соответственно.

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **следует** (SHOULD), **не следует** (SHOULD NOT), **возможно** (MAY) в данном документе должны интерпретироваться в соответствии с документом "Ключевые слова для обозначения уровня требований в RFC" [KEYWORDS].

3. Расширение для аутентификации

(1) Имя расширения сервиса SMTP - Authentication

(2) Ключевое слово EHLO, связанное с этим расширением, - AUTH

(3) Ключевое слово AUTH EHLO содержит в качестве параметра список разделенных пробелами имен поддерживаемых механизмов SASL.

(4) Определяется новая команда (verb) SMTP - AUTH.

(5) Дополнительный параметр, использующий ключевое слово AUTH, добавляется к команде MAIL FROM и максимальный размер этой команды расширяется до 500 символов.

(6) Данное расширение подходит для протокола подачи сообщений [SUBMIT].

4. Команда AUTH

```
AUTH mechanism [initial-response]
```

Аргументы

Строка, идентифицирующая механизм аутентификации SASL. В строку может также включаться необязательный отклик, представленный в формате base64.

Ограничения

После успешного выполнения команды AUTH в той же сессии не могут вводиться дополнительные команды AUTH. После успешного выполнения команды AUTH сервер **должен** отвергать любые последующие команды AUTH с возвратом кода 503.

Команды AUTH не допускаются в процессе выполнения почтовых транзакций.

Обсуждение

Команда AUTH показывает серверу механизм аутентификации. Если сервер поддерживает запрошенный механизм, он выполняет обмен данными протокола аутентификации и идентифицирует пользователя. В дополнение к этому может согласовываться уровень защиты для последующих транзакций протокола. Если запрошенный механизм аутентификации не поддерживается, сервер отвергает команду AUTH с возвратом кода 504.

Обмен данными протокола аутентификации состоит из последовательности запросов (challenge) сервера и откликов (answer) клиента, определяемых используемым механизмом аутентификации. Запрос сервера (server challenge), называемый также индикацией готовности (ready response) представляет собой код 334 с текстом, содержащим строку в коде BASE64. Ответ клиента состоит из строки в коде BASE64. Если клиент отказывается от аутентификационного обмена, он возвращает строку, содержащую только символ *. Если сервер получает такой отклик, он **должен** отвергнуть команду AUTH и возвратить код 501.

Необязательный аргумент initial-response в команде AUTH служит для “замыкания круга” (round trip) при использовании механизмов аутентификации, которые не передают данных в начальном запросе (initial challenge).

[1] Simple Authentication and Security Layer - уровень простой аутентификации и защиты.

При использовании аргумента initial-response с таким механизмом клиенту не передается пустой изначальный запрос и вместо этого сервер передает данные, указанные параметром initial-response, как будто он шлет их в ответ на пустой запрос. В отличие от пустого (zero-length) ответа клиента на код 334 пустой изначальный отклик передается в виде одного символа - знака равенства (=). Если клиент использует в команде AUTH параметр initial-response для механизма, который передает данные в изначальном запросе, сервер отвергает команду AUTH с возвратом кода 535.

Если сервер не может декодировать аргумент BASE64, он отвергает команду AUTH с возвратом кода 501. Если сервер отвергает аутентификационные данные, ему **следует** отвергнуть и команду AUTH с возвратом кода 535, если нет подходящего более специфичного кода из числа перечисленных в главе 6. Для успешного завершения клиентом процесса обмена аутентификационными данными, сервер SMTP возвращает код 235.

Имя сервиса, задаваемое данным вариантом SASL, - smtp.

Если в процессе аутентификационного обмена SASL согласуется уровень защиты, он вступает в силу сразу же после последовательности CRLF, завершающей аутентификационный обмен для клиента и CRLF в отклике о позитивном результате для сервера. После вступления в силу уровня защиты протокол SMTP сбрасывается в начальное состояние (состояние SMTP после возврата сервером кода готовности 220). Сервер **должен** отбрасывать любые полученные от клиента данные (такие, как аргументы команды EHLO), которые не были получены непосредственно из согласования SASL. Клиент **должен** отбрасывать любую полученную от сервера информацию (такую, как список расширений сервиса SMTP), которая не была получена непосредственно из согласования SASL (исключением является **возможность** клиента сравнивать список анонсируемых механизмов SASL до и после аутентификации для детектирования активных атак по срыву согласования[1]). Клиенту **следует** передавать EHLO в качестве первой команды после успешного завершения SASL-согласования, которое приводит к включению уровня защиты.

От сервера не требуется поддержки любого конкретного механизма аутентификации или механизмов аутентификации, требуемых для поддержки любых уровней защиты. При отказе, полученном в ответ на команду AUTH, клиент может попытаться использовать другой механизм аутентификации с помощью новой команды AUTH.

При отказе от выполнения команды AUTH сервер **должен** вести себя так, как будто клиент не вводил команду AUTH совсем.

Строка BASE64 в общем случае может иметь произвольную длину. Клиенты и серверы **должны** поддерживать строки запросов и откликов такого размера, который использует поддерживаемый ими механизм аутентификации, независимо от всех прочих ограничений на размер, которые могут вносить другие компоненты реализации протокола для сервера или клиента.

Примеры

```
S: 220 smtp.example.com ESMTP server ready
C: EHLO jgm.example.com
S: 250-smtp.example.com
S: 250 AUTH CRAM-MD5 DIGEST-MD5
C: AUTH FOOBAR
S: 504 Unrecognized authentication type.
C: AUTH CRAM-MD5
S: 334
PENCeUxFREJoU0NnbmhNWitOMjNGNndAZWx3b29kLmlubm9zb2Z0LmNvbT4=
C: ZnJlZCA5ZTk1YWVlMDljNDBhZjJiODRhMGMyYjNiYmFlNzg2ZQ==
S: 235 Authentication successful.
```

5. Параметр AUTH в команде the MAIL FROM

AUTH=addr-spec

Аргументы

Параметр addr-spec указывает “личность” (identity) подающего сообщение в систему доставки или содержит два символа <>, показывающие, что “личность” неизвестна или недостаточно аутентифицирована. С учетом ограничений, накладываемых на параметры ESMTP значение addr-spec кодируется в xtext. Синтаксис xtext описан в главе 5 документа [ESMTP-DSN].

Обсуждение

Необязательный параметр AUTH в команде MAIL FROM позволяет кооперировать агенты, находящиеся в защищенной среде для аутентификации отдельных сообщений.

Если сервер доверяет аутентифицированной “личности” клиента, заявляющего, что сообщение уже было изначально подано addr-spec, серверу **следует** передать это значение addr-spec в параметре AUTH при трансляции сообщения любому серверу, который поддерживает расширение AUTH.

Команда MAIL FROM с параметром AUTH=<> показывает, что изначально подавший сообщение неизвестен. Для сервера **недопустимо** трактовать такое сообщение как изначально поданное клиентом.

Если параметр AUTH не включен в команду MAIL FROM, клиент аутентифицирован и сервер полагает, что сообщение было изначально подано клиентом, сервер **может** указать “личность” клиента как addr-spec в параметре AUTH при трансляции сообщения любому серверу, который поддерживает расширение AUTH.

Если сервер не совсем доверяет аутентифицированной “личности” клиента или клиент попросту не аутентифицирован, сервер **должен** вести себя как при получении параметра AUTH=<>. Сервер **может** в таких случаях записать значение AUTH в системный журнал.

Если параметр AUTH=<> задан явно или в результате выполнения требований предыдущего параграфа, сервер **должен** указать параметр AUTH=<> при трансляции сообщения любому серверу, который поддерживает расширение AUTH.

Сервер **может** трактовать расширение[2] списка рассылки как новую подачу, указывая в параметре AUTH адрес списка рассылки или администратора этого списка при трансляции сообщения получателям.

Реализация может трактовать всех клиентов как недостаточно доверенных в этом случае реализация не делает ничего кроме разбора и отбрасывания синтаксически корректных параметров AUTH команды MAIL FROM и указания AUTH=<> всем серверам, использующим расширение AUTH.

Примеры

```
C: MAIL FROM:<e=mc2@example.com> AUTH=e+3Dmc2@example.com
```

[1] down-negotiation attack

[2] Извлечение адресов конкретных получателей из сообщения, направленного по адресу списка рассылки. *Прим. перев.*

```
      S: 250 OK
```

6. Коды ошибок

Перечисленные ниже коды ошибок могут использоваться для индикации соответствующих условий.

432 A password transition is needed

Такой отклик на команду AUTH показывает, что пользователю нужно перейти к выбранному механизму аутентификации. Обычно это делается путем однократного использования механизма аутентификации PLAIN.

534 Authentication mechanism is too weak

Этот отклик на команду AUTH показывает, что выбранный механизм аутентификации слабее, чем сервер может позволить для этого пользователя.

538 Encryption required for requested authentication mechanism

Этот отклик на команду AUTH показывает, что выбранный механизм аутентификации может использоваться только для шифрованных соединений SMTP.

454 Temporary authentication failure

Этот отклик на команду AUTH показывает, что аутентификация завершилась неудачей в результате временных проблем на сервере.

530 Authentication required

Этот отклик может возвращаться любой командой, за исключением AUTH, EHLO, HELO, NOOP, RSET и QUIT. Он показывает, что политика сервера требует аутентификации для выполнения запрошенной операции.

7. Формальный синтаксис

Спецификация формального синтаксиса использует расширенную нотацию Бэкуса-Наура (BNF), описанную в документе [ABNF].

За исключением явно указанных случаев регистр символов для букв не принимается во внимание. Использование строчных и прописных букв обусловлено исключительно наглядностью. Реализации **должны** принимать эти строки независимо от регистра символов.

```
UPALPHA         = %x41-5A             ;; верхний регистр: A-Z
LOALPHA         = %x61-7A             ;; нижний регистр: a-z
ALPHA           = UPALPHA / LOALPHA   ;; регистр не имеет значения
DIGIT           = %x30-39             ;; цифры 0-9
HEXDIGIT        = %x41-46 / DIGIT     ;; шестнадцатеричные цифры (верхний регистр для A - F)
hexchar         = "+" HEXDIGIT HEXDIGIT
xchar           = %x21-2A / %x2C-3C / %x3E-7E
                                ;; символы US-ASCII за исключением "+", "=", пробела и CTL
xtext           = *(xchar / hexchar)
AUTH_CHAR       = ALPHA / DIGIT / "-" / "_"
auth_type       = 1*20AUTH_CHAR
auth_command    = "AUTH" SPACE auth_type [SPACE (base64 / "=")] *(CRLF [base64]) CRLF
auth_param      = "AUTH=" xtext
                        ;; декодированная форма xtext ДОЛЖНА совпадать с addr-spec или "<>"
base64          = base64_terminal / ( 1*(4base64_CHAR) [base64_terminal] )
base64_char     = UPALPHA / LOALPHA / DIGIT / "+" / "/"
                                ;; регистр принимается во внимание
base64_terminal = (2base64_char "==") / (3base64_char "=")
continue_req    = "334" SPACE [base64] CRLF
CR              = %x0C                ;; ASCII CR (возврат каретки)
CRLF            = CR LF
CTL             = %x00-1F / %x7F      ;; все коды управления ASCII, а также символ DEL
LF              = %x0A                ;; ASCII LF (перевод строки)
SPACE           = %x20                ;; ASCII SP (пробел)
```

8. Литература

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

[CRAM-MD5] Klensin, J., Catoe, R. and P. Krumviede, "IMAP/POP AUTHorize Extension for Simple Challenge/Response", RFC 2195, September 1997.

[ESMTP] Klensin, J., Freed, N., Rose, M., Stefferud, E. and D. Crocker, "SMTP Service Extensions", RFC 1869, November 1995.

[ESMTP-DSN] Moore, K, "SMTP Service Extension for Delivery Status Notifications", RFC 1891, January 1996.

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

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

[SUBMIT] Gellens, R. and J. Klensin, "Message Submission", RFC 2476, December 1998.

[RFC821] Postel, J., "Simple Mail Transfer Protocol", STD 10, RFC 821, August 1982.

[RFC822] Crocker, D., "Standard for the Format of ARPA Internet Text Messages", STD 11, RFC 822, August 1982.

9. Вопросы безопасности

В этом документе затрагиваются вопросы безопасности.

Если клиент использует это расширение для создания шифрованного туннеля через открытую сеть до корпоративного сервера требуется обеспечить такую настройку клиента, чтобы почта никогда не передавалась на этот сервер без соответствующей аутентификации и шифрования. В противном случае атакующий сможет красть клиентскую почту

путем захвата соединений SMTP и ложной индикации отсутствия на сервере поддержки данного расширения или генерации ложных отказов от выполнения команд AUTH.

До начала согласования SASL все протокольные транзакции выполняются в открытом виде и могут быть изменены в результате активной атаки. По этой причине клиенты и серверы **должны** отбрасывать любые сведения, полученные до начала согласования SASL после того, как будет завершено SASL-согласование, приводящее к использованию защищенного уровня.

Этот механизм не защищает порт TCP, поэтому при активной атаке возможно перенаправить попытки соединения с портом транслятора в порт подачи сообщений [SUBMIT]. Параметр AUTH=<> предотвращает для таких атак возможность заставить транслируемое сообщение без аутентификации конверта “подбирать” аутентификацию клиента транслятора.

Клиент подачи сообщений может требовать от пользователя аутентификации всякий раз, когда анонсируется подходящий механизм SASL. Следовательно, для сервера подачи сообщений [SUBMIT] может оказаться нежелательным анонсирование механизма SASL в тех случаях, когда использование этого механизма не дает клиенту преимуществ по сравнению с анонимной подачей сообщений.

Это расширение не предназначено для замены сквозных систем поддержки цифровых подписей или шифрования типа S/MIME или PGP. Данное расширение предназначено для решения иных задач и ниже перечислены основные отличия от сквозных (end-to-end) систем:

(1) данное расширение в общем случае полезно только на защищенных территориях;

(2) это расширение защищает “конверт” сообщения в целом, а не только тело сообщения;

(3) расширение аутентифицирует подачу сообщений, а не подтверждает авторство содержимого письма;

(4) расширение может дать отправителю некоторые гарантии того, что сообщение будет доставлено на следующий этап (next hop) в тех случаях, когда отправитель имеет взаимную аутентификацию со следующим этапом и согласовал подходящий уровень безопасности.

Дополнительное рассмотрение вопросов безопасности приводится в спецификации SASL [SASL].

10. Адрес автора

John Gardiner Myers

Netscape Communications

501 East Middlefield Road

Mail Stop MV-029

Mountain View, CA 94043

EMail: jgmyers@netscape.com

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru

11. Полное заявление авторских прав

Copyright (C) The Internet Society (1999). Все права защищены.

Этот документ и его переводы могут копироваться и предоставляться другим лицам, а производные работы, комментирующие или иначе разъясняющие документ или помогающие в его реализации, могут подготавливаться, копироваться, публиковаться и распространяться целиком или частично без каких-либо ограничений при условии сохранения указанного выше уведомления об авторских правах и этого параграфа в копии или производной работе. Однако сам документ не может быть изменён каким-либо способом, таким как удаление уведомления об авторских правах или ссылок на Internet Society или иные организации Internet, за исключением случаев, когда это необходимо для разработки стандартов Internet (в этом случае нужно следовать процедурам для авторских прав, заданных процессом Internet Standards), а также при переводе документа на другие языки.

Предоставленные выше ограниченные права являются бессрочными и не могут быть отозваны Internet Society или правопреемниками.

Этот документ и содержащаяся в нем информация представлены "**как есть**" и **автор, организация, которую он/она представляет или которая выступает спонсором (если таковой имеется), Internet Society и IETF отказываются от каких-либо гарантий (явных или подразумеваемых), включая (но не ограничиваясь) любые гарантии того, что использование представленной здесь информации не будет нарушать чьих-либо прав, и любые предполагаемые гарантии коммерческого использования или применимости для тех или иных задач**.