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

Network Working Group
Request for Comments: 2865
Obsoletes: 2138
Category: Standards Track

C. Rigney
S. Willens
Livingston
A. Rubens
Merit
W. Simpson
Daydreamer

June 2000

Протокол RADIUS

Remote Authentication Dial In User Service (RADIUS)

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.

Авторские права

Примечание IESG

Этот протокол имеет множество реализаций и используется достаточно широко. Опыт показывает возможность снижения производительности и потери данных при использовании протокола в больших масштабируемых системах. Это обусловлено отчасти тем, что протокол не включает средств контроля насыщения. Интересующиеся читатели могут обратиться к документам рабочей группы AAA в составе IETF, результатом деятельности которой может стать следующий вариант протокола лучше приспособленный для больших систем и обеспечивающий контроль насыщения.

Тезисы

В этом документе описан протокол, используемый для идентификации (authentication), проверки полномочий (authorization) и установки конфигурационных параметров в серверах доступа, которые хотят работать с идентифицированными пользователями, информация о которых содержится на разделяемом сервере идентификации (Authentication Server).

Замечания для разработчиков

В этом документе содержится спецификация протокола RADIUS. Первые версии протокола RADIUS использовали протокол UDP через порт 1645, что приводило к возникновению конфликтов со службами datametrics. Официально для протокола RADIUS выделен порт 1812.

Оглавление

1. Введение

Данный документ заменяет RFC 2138 [1]. Сводка изменений по сравнению с RFC 2138 приведена в приложении “Журнал изменений”.

Управление распределенной системой доступа по телефонным линиям и модемными пулами в сетях с большим числом пользователей может потребовать от администраторов значительных усилий. Поскольку модемный пул по определению является открытой дверью, требуется повышенное внимание к идентификации пользователей, проверке их полномочий и учету работы (accounting). Наиболее эффективное решение может быть обеспечено путем создания единой “базы данных” о пользователях, которая применяется при идентификации (проверка имени пользователя и пароля), установке конфигурационных параметров и выборе типа сервиса, предоставляемого пользователю (например, SLIP, PPP, telnet, rlogin).

Основными преимуществами протокола RADIUS являются:

Архитектура “клиент-сервер”

Сервер доступа (NAS[1]) выступает в качестве клиента RADIUS. Клиент отвечает за передачу сведений о пользователе заданным серверам RADIUS и дальнейшие действия в зависимости от возвращенной сервером информации.

Серверы RADIUS отвечают за прием клиентских запросов, идентификацию пользователей и возврат клиенту всех конфигурационных параметров, требуемых для предоставления пользователю соответствующих услуг.

[1] Network Access Server – сервер доступа в сеть.

Сервер RADIUS может выступать в качестве клиента-посредника (proxy client) других серверов RADIUS или серверов идентификации иного типа.

Безопасность

Аутентификация транзакций между клиентом и сервером RADIUS осуществляется с использованием разделяемого ключа (shared secret), который никогда не передается через сеть. В дополнение к этому пользовательские пароли между клиентами и серверами RADIUS передаются в зашифрованном виде во избежание перехвата паролей при их передаче через незащищенные сети.

Гибкость механизмов идентификации

Сервер RADIUS может поддерживать широкий спектр методов идентификации пользователей. При получении регистрационного имени и пароля, указанных пользователем, сервер может может поддерживать дополнительные механизмы, включая PPP PAP или CHAP, UNIX login и т. п.

Возможность расширения

Все протокольные транзакции представляются в форме триплетов “атрибут-размер-значение”[2]. Новые атрибуты могут добавляться без нарушения работы существующих реализаций протокола.

1.1. Спецификация уровня требований

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не нужно** (SHALL NOT), **следует** (SHOULD), **не следует** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе должны интерпретироваться в соответствии с BCP 14 [2]. Значение этих слов не зависит от шрифтового выделения.

Реализация протокола считается “несовместимой” со стандартом, если она не соответствует хотя бы одному из обязательных (необходимо, недопустимо) условий. Реализация, в которой выполняются все обязательные и желательные (следует, не следует) условия, считается “безусловно совместимой”, а реализации, в которых выполнены все обязательные условия, но не все желательные, считаются “условно совместимыми”.

Для серверов NAS, которые не поддерживают тот или иной сервис, **недопустимо** использование атрибутов RADIUS для такого сервиса. Например, для сервера NAS, который не поддерживает ARAP **недопустима** реализация атрибутов RADIUS для ARAP. NAS **должен** MUST трактовать RADIUS Access-Accept (доступ разрешен) для недоступного сервиса как Access-Reject (доступ закрыт).

1.2. Терминология

Ниже приведены определения некоторых терминов, часто встречающихся в документе:

Service – сервис, служба, обслуживание

Сервер NAS обеспечивает сервис (например, PPP или Telnet) для подключающихся по коммутируемым линиям пользователей.

Session – сессия, сеанс

Каждый тип сервиса, обеспечиваемого NAS пользователям, предоставляется в форме сеанса, начало которого определяется моментом предоставления первой услуги, а завершение – моментом выполнения последней услуги. Пользователь может организовать множество параллельных (одновременных) сеансов, если NAS поддерживает такой режим.

Silently discard – отбрасывание без уведомления

Отбрасывание пакетов без дальнейшей обработки. Реализациям протокола **следует** обеспечивать возможность ведения журнала ошибок, включающего содержимое отбрасываемых без уведомления пакетов. Также **следует** поддерживать статистику (счетчики) таких событий.

2. Работа протокола

Когда клиент настроен на использование RADIUS, каждый пользователь должен предоставить клиенту свою идентификационную информацию. Это может выполняться в форме приглашения на вход в систему (login prompt), где пользователь должен ввести свое регистрационное имя и пароль. Другим вариантом может быть использование канальных протоколов типа PPP[3], в которых поддерживаются специальные пакеты идентификации для передачи сведений о пользователе.

После того того, как клиент получит идентификационные данные пользователя, он может провести процесс идентификации с использованием RADIUS. Для этого клиент создает пакет Access-Request[4], содержащий такие атрибуты, как регистрационное имя пользователя, пароль, идентификатор клиента и порта (Port ID), через который регистрируется в системе данный пользователь. При наличии пароля он кодируется с использованием алгоритма RSA MD5 [3].

Пакет Access-Request передается серверу RADIUS через сеть. Если в течение продолжительного времени на запрос не будет получено отклика, передача запроса повторяется несколько раз. Клиент может также пересылать запросы другим серверам в тех случаях, когда основной сервер не работает или недоступен. Дополнительные (альтернативные) серверы могут использоваться после некоторого числа неудачных попыток или в режиме кругового обхода известных серверов. Алгоритмы повтора и перебора сервером являются предметом специального исследования и не рассматриваются детально в этом документе.

После получения клиентского запроса сервер RADIUS проверяет передавшего этот запрос клиента. Запросы от клиентов, для которых сервер RADIUS не имеет разделяемого ключа, **должны** отбрасываться без уведомления. Если проверка клиента завершилась успешно, сервер RADIUS обращается к базе данных о пользователях для поиска указанного в запросе имени. Пользовательская запись в базе данных содержит список требований, которым пользователь должен удовлетворять для получения доступа в сеть. К таким требованиям относится проверка пароля, но в базе данных может также указываться клиент (клиенты) и порт (порты), через которые разрешен доступ пользователя.

---

[2] Attribute-Length-Value. В последнее время для таких триплетов чаще используют обозначение TLV (Type-Length-Value – тип-размер-значение). *Прим. перев.*

[3] Point-to-Point Protocol.

[4] Запрос доступа.

Сервер RADIUS **может** делать запросы к другим серверам, выступая в таких случаях как клиент.

Если в запросе Access-Request присутствует хотя бы один атрибут Proxy-State, такой атрибут **должен** без каких-либо изменений копироваться в пакет отклика. Другие атрибуты могут до атрибутов Proxy-State, после них и даже между такими атрибутами.

При невыполнении любого из условий сервер RADIUS передает отклик Access-Reject, показывающий некорректность данного пользовательского запроса. При желании сервер **может** включать в Access-Reject текстовое сообщение, которое **может** передаваться пользователю клиентом. Никакие иные атрибуты (за исключением Proxy-State) не могут включаться в Access-Reject.

Если все условия выполнены и сервер RADIUS хочет "задать пользователю дополнительные вопросы", на которые последний должен ответить, сервер RADIUS передает отклик Access-Challenge. Такой отклик **может** включать текстовое сообщение, выводимое клиентом для пользователя с приглашением ответить на вопросы сервера. Кроме того, отклик **может** включать атрибут State.

Если клиент получает отклик Access-Challenge и поддерживает режим challenge/response, он **может** вывести текстовое сообщение (если оно имеется в пакете отклика) для пользователя, чтобы получить от того отклик. После этого клиент снова передает первоначальный запрос Access-Request с новым идентификатором (request ID), заменой атрибута User-Password на полученную от пользователя информацию (зашифрованную) и включением атрибута State из Access-Challenge (если этот атрибут присутствовал в отклике). В запрос **не следует** включать более одного атрибута State. Сервер может передать в ответ на новый запрос Access-Request отклик типа Access-Accept, Access-Reject или Access-Challenge.

При выполнении всех условий в отклик Access-Accept включается список всех конфигурационных параметров для данного пользователя. К таким параметрам относятся тип сервиса (например, SLIP, PPP, Login User) и все требуемые для предоставления этого сервиса значения. Для протоколов SLIP и PPP могут включаться такие параметры, как адрес IP, маска подсети, MTU, желательность использования компрессии и идентификаторы желаемых фильтров. Для терминальных пользователей эти параметры могут указывать желаемый протокол и хост.

2.1. Режим Challenge/Response

При использовании режима challenge/response пользователю передается непредсказуемое число и ожидается возврат зашифрованного отклика на это число. У легитимных пользователей имеется специальное устройство (например, смарт-карта) или программа для вычисления корректного отклика. Пользователь, не имеющий нужного устройства или программы и не знающий секретного ключа, который позволит эмулировать устройство/программу, может лишь попытаться угадать правильный отклик.

Пакет Access-Challenge обычно содержит атрибут Reply-Message, включающий передаваемый пользователю запрос (challenge), в качестве которого могут использоваться редко повторяющиеся числа. Обычно запрос получают от внешнего сервера, которому известен тип идентификатора, доступного пользователю и который, следовательно, может выбрать случайное или редко повторяющееся число с подходящим основанием и длиной.

Пользователь вводит это число в свое устройство (или программу), вычисляющее отклик, которого ожидает сервер RADIUS во втором запросе Access-Request. Если полученное от пользователя значение соответствует ожидаемому, сервер RADIUS возвращает отклик Access-Accept, а при несоответствии - Access-Reject.

Пример: Сервер NAS передает серверу RADIUS пакет Access-Request с атрибутами NAS-Identifier, NAS-Port, User-Name, User-Password (это может быть просто фиксированная строка типа challenge или пустое значение). Сервер возвращает пакет Access-Challenge с атрибутами State и Reply-Message (строка "Challenge 12345678, enter your response at the prompt[5]", которую NAS передает пользователю). NAS принимает введенное пользователем значение и передает серверу **новый** запрос Access-Request с новым идентификатором, атрибутами NAS-Identifier, NAS-Port, User-Name, User-Password (зашифрованный отклик от пользователя) и значение атрибута State из пакета Access-Challenge. Сервер в ответ шлет отклик Access-Accept или Access-Reject в зависимости от результатов проверки введенного пользователем значения. Допускается также возврат сервером другого отклика Access-Challenge.

2.2. Взаимодействие с PAP и CHAP

Для PAP сервер NAS принимает PAP ID и пароль, передавая их в запросе Access-Request как атрибуты User-Name и User-Password. NAS **может** включать атрибуты Service-Type = Framed-User и Framed-Protocol = PPP как указание серверу RADIUS на использование сервиса PPP.

Для CHAP сервер NAS генерирует случайное число - challenge (предпочтительно 16 октетов) и передает его пользователю, который возвращает CHAP-отклик вместе с CHAP ID и CHAP username. После этого NAS передает запрос Access-Request серверу RADIUS со значением CHAP username для атрибута User-Name и значениями CHAP ID и CHAP-отклик в качестве CHAP-Password (атрибут 3). Случайное число (challenge) может быть включено в атрибут CHAP-Challenge или (если размер числа равен 16 октетам) в поле Request Authenticator пакета Access-Request. Сервер NAS **может** включать атрибуты Service-Type = Framed-User и Framed-Protocol = PPP как указание серверу RADIUS на использование сервиса PPP.

Сервер RADIUS находит пароль для пользователя, указанного атрибутом User-Name, шифрует значение challenge с использованием алгоритма MD5, октета CHAP ID, найденного пароля и CHAP challenge (из атрибута CHAP-Challenge или Request Authenticator при отсутствии этого атрибута) и сравнивает результат с атрибутом CHAP-Password. При совпадении сервер возвращает Access-Accept, в противном случае - Access-Reject.

Если сервер RADIUS не способен выполнить запрошенную идентификацию, он **должен** возвращать Access-Reject. Например, CHAP требует чтобы пользовательский пароль был доступен серверу в открытом виде для шифрования CHAP challenge и сравнения с откликом CHAP. Если незашифрованный пароль недоступен, сервер RADIUS **должен** возвращать клиенту Access-Reject.

2.3. Сервер-посредник (Proxy)

При работе в режиме посредника (proxy) сервер RADIUS принимает от клиента (например, NAS) запросы идентификации или учета и пересылает эти запросы другому серверу RADIUS, а получив от этого сервера отклики, пересылает их клиенту (возможно с внесением изменений в соответствии с локальной политикой администрирования). Наиболее распространенным вариантом

[5] Аналог обмена "пароль – отзыв", используемого часовыми. *Прим. перев.*

использования RADIUS-посредников является организация систем роуминга (roaming), когда два (или более) провайдера принимают запросы не только от своих клиентов, но и от клиентов своих партнеров.

NAS передает серверу RADIUS запрос Access-Request, который сервер-посредник переправляет “удаленному серверу”. Последний возвращает серверу-посреднику отклик (Access-Accept, Access-Reject или Access-Challenge), который пересылается NAS. Атрибут User-Name **может** содержать идентификатор NAI[8] для работы с RADIUS Proxy. Выбор сервера, которому будут пересылаться клиентские запросы **следует** производить на основе областей идентификации (authentication "realm"). Область идентификации **может** быть частью идентификатора NAI[6] (named realm). Кроме того, **возможен** выбор сервера для пересылки запросов на основе других параметров, например, Called-Station-Id (numbered realm).

Сервер RADIUS может работать одновременно в режиме посредника (forwarding server) и отвечающего сервера (remote server), выбирая тот или иной режим в зависимости от области идентификации. Один сервер-посредник может пересылать запросы неограниченному числу удаленных серверов. Точно так же отвечающий сервер может принимать запросы от любого числа серверов-посредников. Сервер-посредник даже может пересылать запросы другому посреднику для создания proxy-цепочек, но этого следует остерегаться во избежание возникновения петель.

Ниже приведен пример обмена информацией между NAS, сервером-посредником и отвечающим сервером RADIUS:

1. NAS передает запрос Access-Request серверу-посреднику.
2. Посредник пересылает запрос удаленному серверу.
3. Удаленный сервер возвращает посреднику отклик Access-Accept, Access-Reject или Access-Challenge (для нашего примера пусть это будет Access-Accept).
4. Посредник передает полученный отклик серверу NAS.

Пересылающий сервер **должен** трактовать имеющиеся атрибуты Proxy-State как непонятные данные (opaque data). Зависимость работы пересылающего сервера от ранее добавленных атрибутов Proxy-State **недопустима**.

Если атрибуты Proxy-State присутствуют в запросе, полученном от клиента, сервер-посредник **должен** включить эти атрибуты в возвращаемый клиенту отклик. Сервер-посредник **может** включать атрибуты Proxy-State в Access-Request при пересылке запроса или опускать такие атрибуты при пересылке. Если при пересылке запроса атрибуты Proxy-State были опущены, сервер-посредник **должен** включить их в отклик, возвращаемый клиенту.

Рассмотрим этапы процесса более детально.

1. NAS передает свой запрос Access-Request серверу-посреднику. Пересылающий сервер расшифровывает значение атрибута User-Password (если атрибут присутствует) с использованием известного ему разделяемого ключа для данного NAS. Если в пакете присутствует атрибут CHAP-Password и нет атрибута CHAP-Challenge, пересылающий сервер **должен** сохранить атрибут Request-Authenticator неизменным или скопировать его значение в атрибут CHAP-Challenge.
   Пересылающий сервер **может** добавить в пакет атрибут Proxy-State, но добавление каких-либо других атрибутов **недопустимо**. При добавлении атрибута Proxy-State этот атрибут **должен** помещаться в пакет после всех имеющихся в пакете атрибутов Proxy-State. Для пересылающего сервера **недопустимо** изменение имеющихся в пакете атрибутов Proxy-State (сервер может отказаться от пересылки этих атрибутов, но менять их **недопустимо**). Также **недопустимо** для пересылающего сервера изменять порядок любых однотипных атрибутов, включая Proxy-State.
2. Пересылающий сервер шифрует User-Password (если этот атрибут присутствует) с использованием ключа, разделяемого с удаленным сервером, устанавливает значение поля Identifier и передает пакет Access-Request удаленному серверу.
3. Удаленный сервер (если он не является посредником) проверяет пользователя с помощью атрибутов User-Password, CHAP-Password или иных методов, которые могут появиться в будущих версиях, и возвращает серверу посреднику пакет Access-Accept, Access-Reject или Access-Challenge. В нашем примере передается отклик Access-Accept. Удаленный сервер **должен** скопировать из запроса Access-Request все атрибуты Proxy-State (и только Proxy-State) с сохранением их порядка.
4. Сервер-посредник проверяет Response Authenticator с использованием ключа, разделяемого с удаленным сервером, и при несоответствии отбрасывает пакет без уведомления. При успешной проверке пересылающий сервер удаляет последний атрибут Proxy-State (если он был добавлен), подписывает Response Authenticator с использованием ключа, разделяемого с NAS, восстанавливает Identifier в соответствии с исходным запросом от NAS и передает отклик Access-Accept серверу NAS.

Пересылающему серверу **может** потребоваться изменение атрибутов в соответствии с локальной политикой. Такие вопросы выходят за пределы данного документа, однако спецификация протокола вносит ряд ограничений на изменение атрибутов пересылающими серверами. Для серверов посредников **недопустимо** изменять существующие атрибуты Proxy-State, State, Class. Разработчикам таких серверов следует внимательно относиться к принимаемым сервером значениям атрибута Service-Type. Особая осторожность требуется для атрибута Service-Type со значениями NAS-Prompt или Administrative в пересылаемых пакетах Access-Accept и разработчики могут использовать механизмы блокирования таких сообщений, а также сообщений иных типов. Рассмотрение механизмов блокировки выходит за пределы настоящей спецификации.

2.4. Почему UDP?

Достаточно часто спрашивают, почему протокол RADIUS использует транспорт UDP, а не TCP. Выбор протокола UDP был обусловлен техническими причинами.

Здесь нужно разобраться со множеством вопросов. Протокол RADIUS работает на основе транзакций, что ведет к ряду интересных особенностей:

1. **При отказе основного сервера идентификации запросы должны передаваться резервному серверу.**
   Для выполнения этого требования копия запроса должна сохраняться выше транспортного уровня чтобы обеспечить возможность повторного запроса. Это требует поддержки таймеров повтора передачи.
2. **Временные требования протокола существенно отличаются от обеспечиваемых TCP временных параметров.**
   С одной стороны, RADIUS не требует “нести ответственность” за детектирование потери данных. Пользователь может подождать завершения процедуры идентификации в течение нескольких секунд. Не требуется агрессивная политика TCP в части передачи повторов (на основе среднего времени кругового обхода), а также передача подтверждений, увеличивающая

[6] Network Access Identifier – идентификатор доступа в сеть.

уровень служебного трафика.
С другой стороны, пользователя вряд ли устроит процедура идентификации, занимающая несколько минут. Следовательно гарантированная доставка данных на основе TCP в течение 2 минут не принесет пользы. Передача запроса альтернативному серверу позволит пользователю быстрее завершить процедуру идентификации.

3. **Протокол по своей природе не требует организации прямых соединений.**
   Клиенты и серверы “приходят и уходят”. Системы могут перезагружаться или выключаться. В общем случае это не вызывает проблем и средства обнаружения обрыва соединений TCP вкупе с механизмами тайм-аутов позволяют обрабатывать аномальные ситуации. Однако использование протокола UDP полностью избавляет от таких ситуаций без какой-либо специальной обработки. Каждый клиент или сервер может активизировать свой транспорт UDP и сохранять его в активном состоянии независимо от возникающих в сети проблем.

4. **UDP упрощает реализацию серверов.**
   Первые реализации серверов RADIUS были однопотоковыми. Это означало, что запросы принимались и обрабатывались по одному. Такое решение оказалось неприемлемым для сред, где реализация механизмов безопасности занимала достаточно продолжительное время (1 секунду или более). Очередь запросов сервера могла содержать достаточно много запросов и в средах, где каждую минуту проверяется идентификация сотен пользователей, пользователи были вынуждены ждать завершения идентификации слишком долго. Ожидание увеличивалось дополнительно при обращении к базам данных или серверам DNS и могло затянуться на 30 секунд и более того. Обычно такие проблемы решаются путем создания многопотоковых (multi-threaded) серверов. Реализация таких серверов существенно упрощается при использовании протокола UDP. Для обработки каждого запроса порождается отдельный процесс и этот процесс может напрямую взаимодействовать с клиентом NAS, обмениваясь с ним пакетами UDP.

Однако ничего не достается даром и отказ от использования TCP приводит к необходимости реализации протоколом некоторых функций, которые в TCP поддерживаются транспортным уровнем. В частности, при работе по протоколу UDP требуется поддержка таймеров повторной передачи для сервера, хотя и не со столь жесткими требованиями, как в TCP. Это достаточно невысокая плата за те преимущества, которые обеспечивает работа на основе протокола UDP.

Для протокола RADIUS транспорт UDP является оптимальным решением.

2.5. Рекомендации по передаче повторов

Если основной и дополнительный серверы RADIUS используют общий разделяемый ключ, разумно пересылать пакеты дополнительному серверу с теми же значениями полей ID и Request Authenticator, поскольку значения атрибутов сохраняются при повторной передаче. При желании можно для запросов к дополнительному серверу использовать другое значение для поля Request Authenticator.

При изменении User-Password или значения любого другого атрибута, нужно задать новое значение поля Request Authenticator и, следовательно, новый идентификатор ID.

Если NAS передает повторный запрос тому же серверу, которому был отправлен первичный, и атрибуты запроса не изменились, **должны** использоваться такие же значения Request Authenticator, ID и номер порта отправителя. При изменении атрибутов **должны** указываться новые значения Request Authenticator и ID.

Сервер NAS **может** использовать одинаковые значения ID для всех серверов или выбирать ID для каждого сервера по усмотрению разработчиков. Если серверу NAS требуется более 256 значений ID для исходящих запросов, **можно** воспользоваться другими номерами портов отправителя и сохранять значения ID для каждого из таких портов. Это позволит создать до 16 миллионов одновременных запросов к одному серверу.

2.6. Пагубность запросов Keep-Alive

В некоторых реализациях используется передача тестовых запросов RADIUS для проверки работоспособности сервера. Это порочная практика, от которой следует отказаться, поскольку при таком способе проверки работоспособности без всякой практической пользы возрастает нагрузка на сервер и снижается уровень масштабируемости. Вместо отправки тестовых запросов лучше передать серверу нормальный запрос и полученный от сервера отклик подтвердит работоспособность сервера. Если же у вас нет запросов для отправки серверу RADIUS, вам нет особой нужды беспокоиться о его работоспособности.

Если вы хотите организовать мониторинг сервера RADIUS, используйте протокол SNMP, разработанный специально для таких задач.

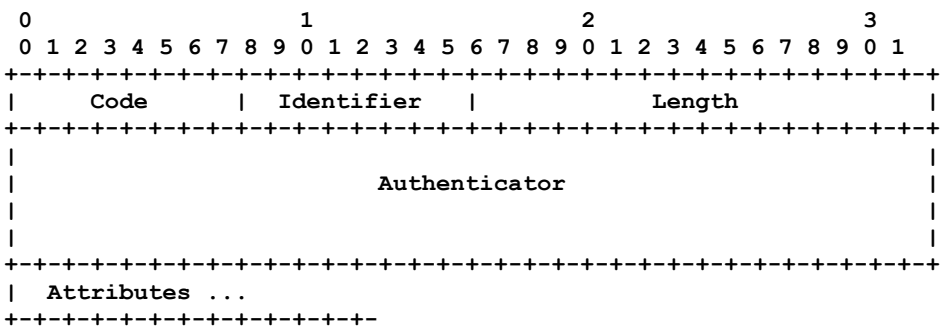
3. Формат пакетов

В поле данных пакетов UDP [4] инкапсулируется по одному пакету RADIUS и поле UDP Destination Port для протокола RADIUS должно содержать десятичное значение 1812.

При генерации откликов номера портов отправителя и получателя меняются местами.

В этом документе содержится спецификация протокола RADIUS. Ранние версии RADIUS использовали порт UDP 1645, что приводило к конфликтам со службами datametrics. Официально выделенный для протокола RADIUS порт имеет номер 1812.

Ниже показан формат типового пакета RADIUS. Поля передаются слева направо и сверху вниз.



Code

Поле Code имеет размер 1 октет и содержит идентификатор типа пакета RADIUS. При получении пакета с некорректным значением поля Code такой пакет отбрасывается без уведомления.

Десятичные значения кодов для пакетов RADIUS показаны в таблице.

Код	*Тип пакета*
1	Access-Request
2	Access-Accept
3	Access-Reject
4	Accounting-Request
5	Accounting-Response
11	Access-Challenge
12	Status-Server (экспериментальный)
13	Status-Client (экспериментальный)
255	Зарезервирован

Коды 4 и 5 описаны в документе RADIUS Accounting [5]. Коды 12 и 13 зарезервированы и могут использоваться, но не рассматриваются в данном документе.

Identifier

Поле Identifier размером 1 октет используется для сопоставления запросов с откликами. Сервер RADIUS может детектировать дубликаты запросов по совпадению IP-адреса отправителя, номеру порта отправителя и значению поля Identifier, если такие пакеты получены в течение короткого промежутка времени.

Length

Поле Length имеет размер 2 октета и показывает размер пакета с учетом полей Code, Identifier, Length, Authenticator и Attribute. Октеты за пределами указанного в поле размера значения **должны** трактоваться как заполнение и оставляться без внимания. Если размер пакета меньше значения поля Length, пакет **должен** отбрасываться без уведомления. Минимальный размер пакета составляет 20, а максимальный - 4096.

Authenticator

Поле Authenticator имеет размер 16 октетов. Старший октет поля передается первым. Значение поля применяется для идентификации откликов от сервера RADIUS, а также используется алгоритмом сокрытия паролей.

Request Authenticator

В пакетах Access-Request в качестве значения поля Authenticator используется 16-октетное случайное значение Request Authenticator. **Следует** использовать непредсказуемые значения, уникальные в течение срока жизни ключа (пароля, используемого при обмене информацией между клиентом и сервером RADIUS), поскольку повторное использование значений вкупе с тем же ключом позволит атакующему использовать перехваченные отклики. В предположении что разделяемый секрет **может** использоваться в географически удаленных серверах, поле Request Authenticator следует делать уникальным в пространственном и временном аспекте.

В пакетах Access-Request также **следует** использовать непредсказуемые значения поля Request Authenticator, чтобы атакующий не мог обмануть сервер, предсказав будущий запрос, и использовать полученный от сервера отклик, прикинувшись сервером для будущих запросов Access-Request.

Хотя такие протоколы, как RADIUS, не обеспечивают защиты сеансов идентификации от перехвата путем прямого прослушивания, использование уникальных и непредсказуемых запросов может обеспечить защиту от множества типов атак на систему идентификации пользователей.

Серверы NAS и RADIUS используют разделяемый ключ (пароль). Этот ключ вместе с Request Authenticator передаются необратимой функции MD5 для создания 16-октетного значения, которое объединяется с введенным пользователем паролем (логическая операция XOR) и результат помещается в атрибут User-Password пакета Access-Request. Более подробное описание этих операций приведено ниже при рассмотрении атрибута User-Password.

Response Authenticator

Поле Authenticator в пакетах Access-Accept, Access-Reject и Access-Challenge называют Response Authenticator. Это поле содержит необратимое хэш-значение MD5 рассчитанное для потока октетов, состоящего из пакета RADIUS, начиная с поля Code и включая поля Identifier, Length и Request Authenticator из пакета Access-Request, атрибутов отклика и разделяемого ключа. Таким образом,

ResponseAuth = MD5(Code+ID+Length+RequestAuth+Attributes+Secret),

где знак + обозначает конкатенацию (объединение) строк.

Замечания для администраторов

Разделяемый ключ (пароль, известный клиенту и серверу RADIUS) **следует** создавать с соблюдением обычных требований, предъявляемых к хорошим паролям. Предпочтительно использовать ключи размером не менее 16 октетов. Это существенно затруднит атаки путем подбора ключей. **Недопустимо** использование пустых (нулевой длины) ключей, поскольку в этом случае перехват пакетов становится тривиальной задачей.

Сервер RADIUS **должен** использовать IP-адрес отправителя из пакетов RADIUS UDP для выбора разделяемого с клиентом секрета, чтобы можно было передавать запросы RADIUS через серверы-посредники.

При использовании пересылающего proxy, этот сервер-посредник должен быть способен отличать пакеты, передаваемые в каждом направлении. При пересылке запросов посредник **может** добавлять атрибут Proxy-State, а при пересылке откликов посредник **должен** удалить добавленный атрибут Proxy-State. Удаляемый или добавляемый атрибут Proxy-State всегда должен быть последним среди одноименных атрибутов, но делать иные допущения о месте данного атрибута среди прочих атрибутов не следует. Поскольку для откликов Access-Accept и Access-Reject аутентификация осуществляется на уровне всего пакета, удаление атрибута Proxy-State делает сигнатуру некорректной и proxy-сервер должен заново “подписать” пакет.

Другие детали реализации серверов-посредников RADIUS выходят за пределы данной спецификации.

4. Типы пакетов

Тип пакетов RADIUS определяется значением поля Code (код) в первом октете пакета.

4.1. Пакет Access-Request

Пакеты Access-Request передаются серверу RADIUS и содержат информацию, которая используется для того, чтобы определить имеет ли пользователь право доступа к указанному серверу NAS и службам, запрошенным пользователем. Реализации, желающие применять аутентификацию пользователей **должны** передавать пакет RADIUS с Code = 1 (Access-Request).

При получении пакета Access-Request от легитимного клиента **должен** передаваться соответствующий отклик.

В пакеты Access-Request **следует** включать атрибут User-Name. Пакет **должен** содержать по крайней мере один из атрибутов NAS-IP-Address и NAS-Identifier.

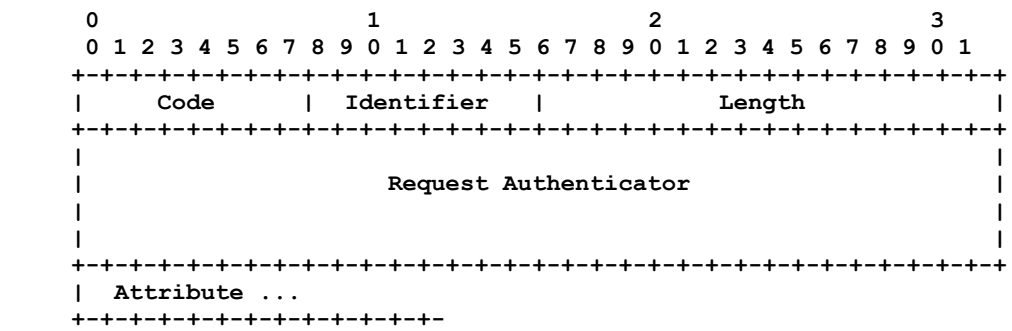
Пакет Access-Request **должен** включать атрибут User-Password, CHAP-Password или State. **Недопустимо** помещать в пакет оба атрибута User-Password и CHAP-Password. Если в будущих расширениях появятся новые типы идентификационной информации для включения в пакеты Access-Request, соответствующие атрибуты могут использоваться взамен User-Password или CHAP-Password.

В пакеты Access-Request **следует** включать атрибут NAS-Port или NAS-Port-Type (возможно включение обоих атрибутов), за исключениям тех случаев, когда запрашиваемый тип доступа включает порт или NAS не различает портов.

Пакет Access-Request **может** дополнительные атрибуты, служащие рекомендациями для сервера, но сервер не обязан использовать эти атрибуты.

При наличии атрибута User-Password для сокрытия значения пароля используется алгоритм RSA MD5 [3].

Формат пакета Access-Request показан ниже. Поля пакета передаются слева направо и сверху вниз.



Code = 1

Identifier

Значение поля Identifier **должно** меняться при изменении содержимого полей атрибутов и при получении корректного отклика на предыдущий запрос. При повторной передаче значение поля Identifier **должно** сохраняться.

Request Authenticator

Значение поля Request Authenticator **должно** меняться всякий раз при смене значения поля Identifier.

Attribute

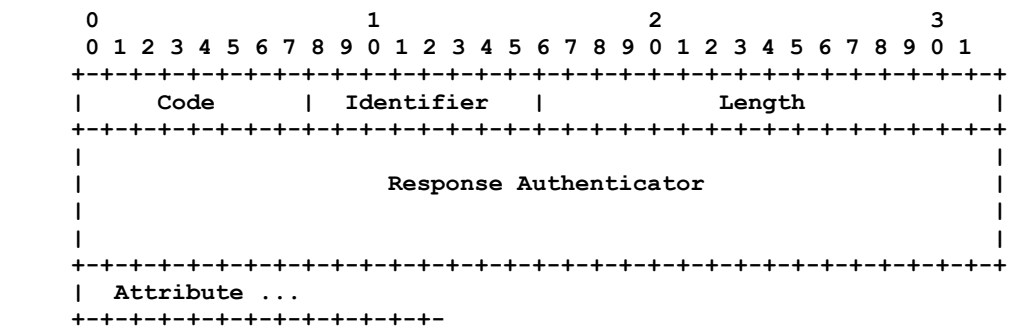
Поле Attribute имеет переменную длину и содержит список атрибутов, которые требуются для запрашиваемого типа сервиса, а также дополнительных атрибутов.

4.2. Пакет Access-Accept

Пакеты Access-Accept передаются сервером RADIUS и содержат конфигурационные параметры, необходимые для начала предоставления услуг пользователю. Если все значения атрибутов, полученные в пакете Access-Request, приемлемы, реализация RADIUS **должна** передать пакет с Code = 2 (Access-Accept).

При получении пакета Access-Accept значение поля Identifier сравнивается с ожидающим запросом Access-Request. Поле Response Authenticator **должно** содержать корректный отклик для ожидающего запроса Access-Request. Некорректные пакеты отбрасываются без уведомления.

Формат пакета Access-Accept показан ниже. Поля пакета передаются слева направо и сверху вниз.



Code = 2

Identifier

Поле Identifier содержит копию значения одноименного поля из пакета Access-Request, с которым связан данный отклик.

Response Authenticator

Значение поля Response Authenticator вычисляется на основе полей запроса Access-Request, как описано выше.

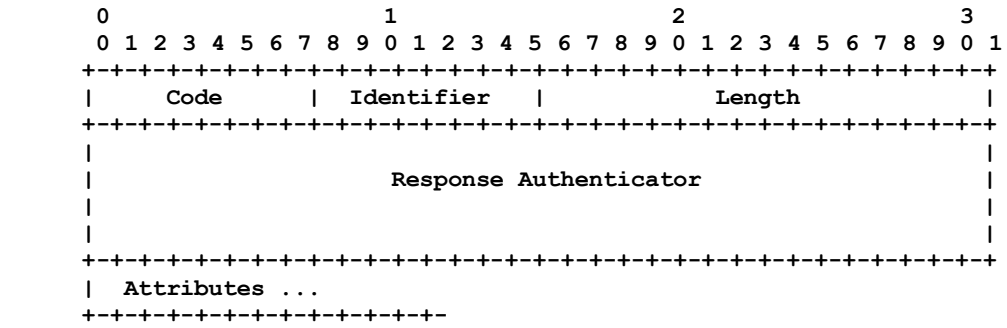
Attribute

Необязательное поле атрибутов имеет переменную длину.

4.3. Пакет Access-Reject

Если какой-либо из принятых в запросе атрибутов неприемлем, сервер RADIUS **должен** передать пакет с Code = 3 (Access-Reject). В пакет **может** быть включен один или несколько атрибутов Reply-Message с текстовым сообщением, которое NAS **может** передавать пользователю.

Формат пакета Access-Reject показан ниже. Поля пакета передаются слева направо и сверху вниз.



Code = 3

Identifier

Поле Identifier содержит копию значения одноименного поля из пакета Access-Request, с которым связан данный отклик.

Response Authenticator

Значение поля Response Authenticator вычисляется на основе полей запроса Access-Request, как описано выше.

Attributes

Необязательное поле атрибутов имеет переменную длину.

4.4. Пакет Access-Challenge

Если сервер RADIUS хочет отправить пользователю запрос на ввод дополнительной информации (challenge), требующий отклика, сервер RADIUS **должен** ответить на запрос Access-Request передачей пакета с Code = 11 (Access-Challenge).

Необязательное поле атрибутов такого пакета **может** содержать один или несколько атрибутов Reply-Message и один атрибут State. Допускается также включение в отклик атрибутов Vendor-Specific, Idle-Timeout, Session-Timeout и Proxy-State. Остальные атрибуты, описанные в данной спецификации, не должны включаться в пакеты Access-Challenge.

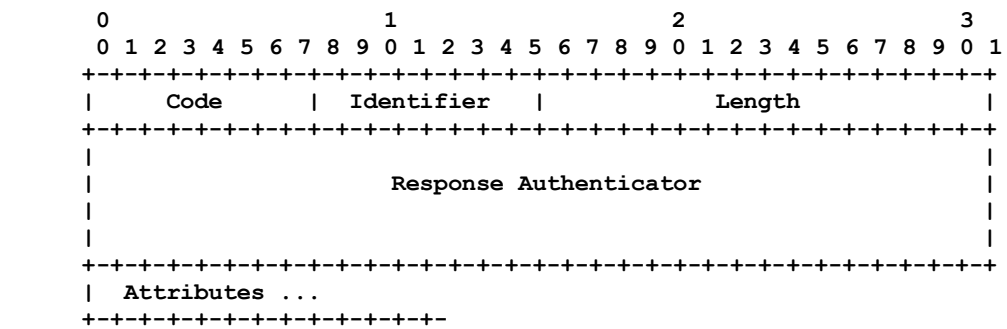
При получении пакета Access-Challenge значение поля Identifier сравнивается с идентификатором в ожидающем запросе Access-Request. Кроме того поле Response Authenticator **должно** содержать корректный отклик для ожидающего Access-Request. Некорректные пакеты отбрасываются без уведомления.

Если сервер NAS не поддерживает режим challenge/response, он **должен** трактовать пакеты Access-Challenge как Access-Reject.

Если NAS поддерживает режим challenge/response, получение корректного пакета Access-Challenge показывает, что **следует** передать новый пакет Access-Request. Сервер NAS **может** передавать пользователю текстовое сообщение (если оно есть) и тогда запрашивать у пользователя отклик. После получения отклика передается исходный запрос Access-Request с новым идентификатором и полем Request Authenticator, а также с заменой значения атрибута User-Password на введенную пользователем информацию (в шифрованном виде) и включением атрибута State из пакета Access-Challenge (если этот атрибут присутствует). В пакете Access-Request может присутствовать не более 1 атрибута State.

Сервер NAS, поддерживающий протокол PAP, **может** пересылать Reply-Message вызывающему клиенту и принимать от того отклик PAP, который может использоваться как введенный пользователем отклик. Если сервер NAS не может это сделать, он **должен** трактовать пакет Access-Challenge как Access-Reject.

Формат пакета Access-Challenge показан ниже. Поля пакета передаются слева направо и сверху вниз.



Code = 11

Identifier

Поле Identifier содержит копию значения одноименного поля из пакета Access-Request, с которым связан данный отклик.

Response Authenticator

Значение поля Response Authenticator вычисляется на основе полей запроса Access-Request, как описано выше.

Attributes

Необязательное поле атрибутов имеет переменную длину.

5. Атрибуты

Атрибуты RADIUS служат для передачи сведений, используемых для идентификации, проверки полномочий, конфигурации, а также для передачи пользователю той или иной информации.

Завершение списка атрибутов определяется по значению поля Length в пакетах RADIUS.

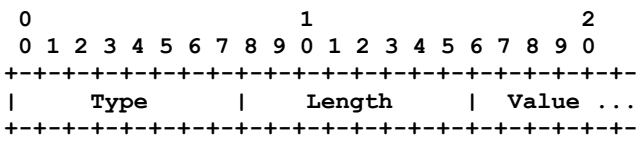
Некоторые атрибуты **могут** включаться в пакет в нескольких экземплярах. Эффект от включения нескольких однотипных атрибутов определяется конкретным атрибутом и рассматривается ниже при описании соответствующих атрибутов. Сводная таблица атрибутов приведена ниже.

При наличии в пакете нескольких однотипных атрибутов серверы-посредники **должны** сохранять порядок этих атрибутов. Сохранение порядка для разнотипных атрибутов не требуется. Для серверов и клиентов RADIUS **недопустимо** принятие каких-либо решений на основе порядка расположения разнотипных атрибутов. **Недопустимо** также требование непрерывности однотипных атрибутов.

Если при описании того или иного атрибута указан тип пакетов, в которых этот атрибут может присутствовать, это ограничение применимо только к типам пакетов, описанным в данном документе, а именно - Access-Request, Access-Accept, Access-Reject и Access-Challenge (коды 1, 2, 3, 11). Другие документы, определяющие иные типы пакетов также могут использовать описанные здесь атрибуты. Для определения допустимости использования атрибутов в пакетах Accounting-Request и Accounting-Response (код 4 и 5) обращайтесь к документу RADIUS Accounting [5].

В данной спецификации указано какие атрибуты допустимо использовать с определенными в этом документе типами пакетов. В будущих документах при определении новых атрибутов также следует указывать в каких типах пакетов может присутствовать атрибут.

Формат поля атрибута показан ниже. Поля атрибута передаются слева направо.



Type

Однооктетное поле, определяющее тип атрибута. Актуальные значения поля типа для атрибутов RADIUS вы можете узнать из последнего варианта документа Assigned Numbers [6]. Значения 192-223 предназначены для экспериментальных целей, значения 224-240 зарезервированы для разработчиков (специфические для реализации типы), а значения 241-255 являются резервными и не должны использоваться.

Сервер RADIUS **может** игнорировать атрибуты неизвестных типов.

Клиент RADIUS **может** игнорировать атрибуты неизвестных типов.

Определенные в данной спецификации атрибуты перечислены в таблице 1:

Таблица 1: Атрибуты RADIUS

Тип	*Атрибут*	*Тип*	*Атрибут*	*Тип*	*Атрибут*
1	User-Name	16	Login-TCP-Port	31	Calling-Station-Id
2	User-Password	17	(не используется)	32	NAS-Identifier
3	CHAP-Password	18	Reply-Message	33	Proxy-State
4	NAS-IP-Address	19	Callback-Number	34	Login-LAT-Service
5	NAS-Port	20	Callback-Id	35	Login-LAT-Node
6	Service-Type	21	(не используется)	36	Login-LAT-Group
7	Framed-Protocol	22	Framed-Route	37	Framed-AppleTalk-Link
8	Framed-IP-Address	23	Framed-IPX-Network	38	Framed-AppleTalk-Network
9	Framed-IP-Netmask	24	State	39	Framed-AppleTalk-Zone
10	Framed-Routing	25	Class	40-59	(зарезервированы для учета)
11	Filter-Id	26	Vendor-Specific	60	CHAP-Challenge
12	Framed-MTU	27	Session-Timeout	61	NAS-Port-Type
13	Framed-Compression	28	Idle-Timeout	62	Port-Limit
14	Login-IP-Host	29	Termination-Action	63	Login-LAT-Port
15	Login-Service	30	Called-Station-Id		

Length

Однооктетное поле Length указывает размер данного атрибута с учетом полей Type, Length и Value. При получении в пакете Access-Request атрибута с некорректно указанным размером **следует** передавать отклик Access-Reject. При получении атрибута с некорректно указанным размером в пакетах Access-Accept, Access-Reject или Access-Challenge пакет **должен** трактоваться как Access-Reject или отбрасываться без уведомления.

Value

Необязательное поле Value содержит значение атрибута. Формат и размер значения атрибута определяются значениями полей Type и Length.

Отметим, что ни один из типов RADIUS не использует в качестве завершения NUL-символ (hex 00). В частности, значения типа text и string в протоколе RADIUS не завершаются NUL-символом. Для каждого атрибута имеется поле размера, поэтому символы завершения не требуются. Значения типа text представляет собой последовательность символов в кодировке UTF-8 10646 [7], а значения типа string содержат 8-битовые бинарные данные. Серверы и клиенты RADIUS **должны** быть способны работать со строками, содержащими NUL-символы. При реализации RADIUS на основе языка C не следует использовать для обработки строк функцию strcpy().

Значение атрибута может относиться к одному из пяти поддерживаемых типов данных. Отметим, что тип text является частным случаем (подмножеством) типа string.

text от 1 до 253 октетов, содержащих символы в кодировке UTF-8 10646 [7]. **Недопустима** передача текстовых атрибутов нулевой длины. В таких случаях следует просто исключить атрибут.

string от 1 до 253 октетов, содержащих бинарные данные (значения от 0 до 255, включительно). **Недопустима** передача string-атрибутов нулевой длины. В таких случаях следует просто исключить атрибут.

address 32-битовое значение, первый октет является старшим.

integer 32-битовое беззнаковое целое, первый октет является старшим.

time 32-битовое беззнаковое целое (первый октет является старшим), показывающее число секунд, прошедших с 1 января 1970 г. (00:00:00 по Гринвичу – UTC). Стандартные атрибуты RADIUS не используют этот тип, но он добавлен для будущих расширений.

5.1. User-Name

Этот атрибут показывает имя пользователя, для которого выполняется аутентификация. При наличии у пользователя имени этот атрибут **должен** передаваться в пакетах Access-Request.

Атрибут **может** передаваться в пакетах Access-Accept – в этом случае клиенту **следует** использовать возвращенное в пакете Access-Accept имя пользователя для всех пакетов Accounting-Request в данном сеансе. Если Access-Accept включает Service-Type = Rlogin и атрибут User-Name, NAS **может** использовать возвращенный атрибут User-Name при выполнении функции Rlogin.

Формат полей атрибута User-Name показан ниже. Поля передаются слева направо.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|     Type      |    Length     |  String ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Type = 1

Length ≥ 3

String

Поле String может содержать 0 и более октетов. Сервер NAS может ограничивать размер поля User-Name, но рекомендуется обрабатывать поля по крайней мере до 63 октетов длиной.

Имя пользователя может указываться одним из трех способов:

text – строка символов в кодировке UTF-8 10646 [7].

network access identifier – идентификатор NAI, описанный в RFC 2486 [8].

distinguished name – имя в формате ASN.1, используемое в системах аутентификации Public Key.

5.2. User-Password

Этот атрибут показывает пароль идентифицируемого пользователя или данные, введенные пользователем в ответ на пакет Access-Challenge. Пароль может передаваться только в пакетах Access-Request.

При передаче пакетов используется сокрытия паролей. Сначала к паролю добавляются NUL-символы до значения, кратного 16 октетам. Далее вычисляется необратимая хэш-функция MD5 для потока октетов разделяемого ключа и значения Request Authenticator. Полученное значение объединяется (логическая операция XOR) с первыми 16 октетами пароля и помещается в первые 16 октетов поля String в атрибуте User-Password.

Если размер пароля превышает 16, вычисляется вторая необратимая хэш-последовательность MD5 для потока октетов, состоящего из разделяемого ключа и результата шифрования первых 16 октетов пароля. Полученный результат объединяется (операция XOR) со вторым 16-октетным сегментом пароля и помещается в следующие 16 октетов поля String в атрибуте User-Password.

При необходимости эта операция повторяется для каждого 16-октетного сегмента пароля с использованием разделяемого ключа и результата предыдущей операции. Размер пароля не может превышать 128 символов.

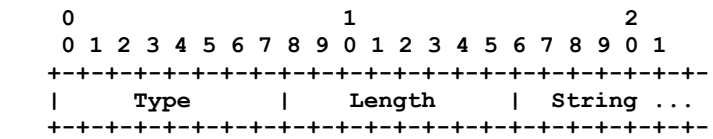
Этот метод шифрования взят из книги "Network Security" (Kaufman, Perlman и Speciner) [9], стр. 109-110. Ниже приведено более детальное описание метода:

Вызывается функция MD5 с разделяемым ключом S 128-битовым псевдослучайным значением Request Authenticator (RA). Пароль делится на 16-октетные сегменты p1, p2 и т. д. с дополнением последнего сегмента NUL-символами для выравнивания по 16-октетной границе. Используется операция XOR (исключающее или) для хэш-функции и соответствующего сегмента пароля. Для второго и последующих сегментов вместо RA используется хэш-функция, полученная на предыдущем этапе. Зашифрованный пароль сохраняется как конкатенация промежуточных результатов c(1), c(2) и т. д. в поле String атрибута User-Password.

```
        b1 = MD5(S + RA)       c(1) = p1 xor b1
        b2 = MD5(S + c(1))     c(2) = p2 xor b2
               .                      .
               .                      .
               .                      .
        bi = MD5(S + c(i-1))   c(i) = pi xor bi
```

На приемной стороне процесс выполняется в обратном порядке для получения исходного пароля[7].

Формат полей атрибута User-Password показан ниже. Поля передаются слева направо.



Type = 2

Length

От 18 до 130.

String

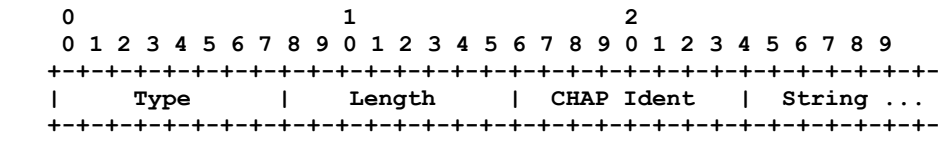
Строка типа String размером от 16 до 128 октетов (кратные 16 значения), содержащая зашифрованный пароль.

5.3. CHAP-Password

Этот атрибут показывает значение отклика, представленное пользователем протокола CHAP[8] в ответ на запрос (challenge). Атрибут может включаться только в пакеты Access-Request.

Запрос CHAP можно найти в атрибуте CHAP-Challenge (60), если он присутствует в пакете, или в поле Request Authenticator.

Формат атрибута CHAP-Password показан ниже. Поля передаются слева направо.



Type = 3

Length = 19

CHAP Ident

Это поле имеет размер 1 октет и содержит идентификатор CHAP из пользовательского CHAP Response.

String

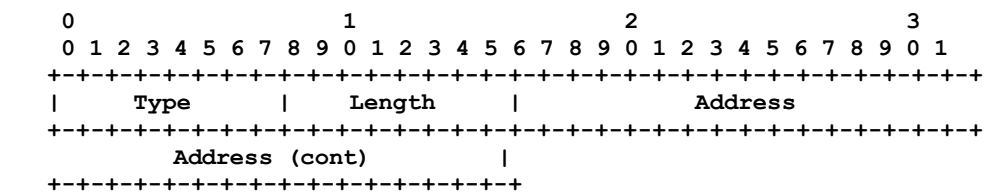
Это 16-октетное поле содержит значение CHAP Response, принятое от пользователя.

5.4. NAS-IP-Address

Этот атрибут показывает IP-адрес сервера NAS, который запрашивает аутентификацию пользователя. **Следует** иметь уникальные адреса всех серверов NAS в пределах сферы действия сервера RADIUS. В каждом пакете Access-Request **должен** присутствовать атрибут NAS-IP-Address или NAS-Identifier.

Отметим, что значение атрибута NAS-IP-Address **недопустимо** использовать для выбора разделяемого ключа, применяемого при аутентификации запроса. Такой выбор **должен** осуществляться на основе адреса отправителя в пакете Access-Request.

Формат атрибута NAS-IP-Address показан ниже. Поля передаются слева направо.



Type = 4

Length = 6

Address

Четырехоктетное значение IP-адреса.

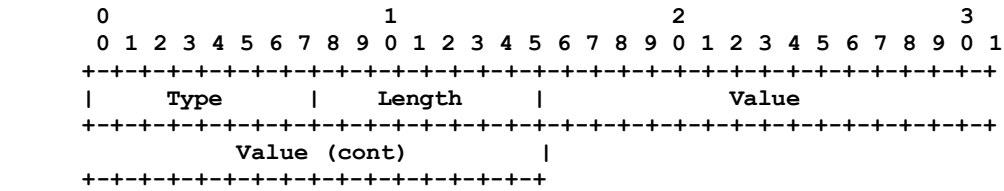
5.5. NAS-Port

Этот атрибут указывает физический порт сервера NAS, через который обратился идентифицируемый пользователь. Этот атрибут используется только в пакетах Access-Request. Отметим, что номер порта NAS, не имеет никакого отношения к номерам используемых портов TCP или UDP. Если сервер NAS способен различать свои порты, в пакеты Access-Request **следует** включать атрибут NAS-Port или NAS-Port-Type (61), допускается одновременное включение обоих атрибутов.

[7] Процесс восстановления пароля является корректным, несмотря на использование при его шифровании необратимых хеш-функций MD5. *Прим. перев.*

[8] PPP Challenge-Handshake Authentication Protocol

Формат атрибута NAS-Port показан ниже. Поля передаются слева направо.



Type = 5

Length = 6

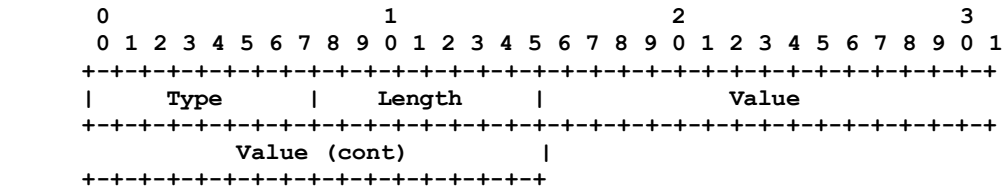
Value

Четырехоктетное значение идентификатора порта NAS.

5.6. Service-Type

Этот атрибут показывает тип сервиса, запрошенного пользователем или тип обеспечиваемого пользователю сервиса. Атрибут **может** использоваться в пакетах Access-Request и Access-Accept. От серверов NAS не требуется реализация всех типов сервиса, они просто **должны** трактовать неизвестные типы как неподдерживаемые значения Service-Type (как при получении ответа Access-Reject).

Формат атрибута Service-Type показан ниже. Поля передаются слева направо.



Type = 6

Length = 6

Value

Четырехоктетное поле Value содержит один из перечисленных в таблице идентификаторов типа сервиса.

Тип	*Сервис*	*Тип*	*Сервис*	*Тип*	*Сервис*
1	Login	5	Outbound	9	Callback NAS Prompt
2	Framed	6	Administrative	10	Call Check
3	Callback Login	7	NAS Prompt	11	Callback Administrative
4	Callback Framed	8	Authenticate Only		

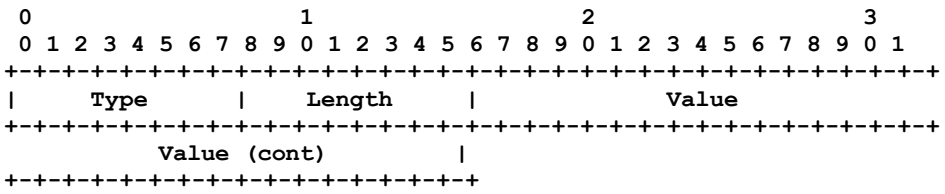
Ниже приведены определения различных типов сервиса для использования в пакетах Access-Accept. Типы сервиса в пакетах Access-Request **могут** рассматриваться как рекомендации серверу RADIUS от сервера NAS, который имеет основания предполагать, что пользователь предпочитает указанный тип сервиса. Рекомендации не являются обязательными для сервера.

Тип	*Описание*
Login	Пользователь подключается к хосту.
Framed	Для пользователя должен быть запущен Framed-протокол (например, PPP или SLIP).
Callback Login	Пользователь должен быть отключен и соединен с хостом после вызова со стороны хоста.
Callback Framed	Пользователь должен быть отключен и соединен с хостом после вызова со стороны хоста, после чего для пользователя должен быть запущен Framed-протокол (например, PPP или SLIP).
Outbound	Пользователю следует предоставить возможность доступа к устройствам для организации исходящих соединений.
Administrative	Пользователю следует предоставить административный интерфейс с сервером NAS, на котором будут выполняться команды, требующие определенных привилегий.
NAS Prompt	Пользователю следует предоставить возможность ввода консольных команд NAS, не требующих специальных привилегий.
Authenticate Only	Запрошена только идентификация пользователя и не требуется возвращать сведений о проверке полномочий (авторизации) в отклике Access-Accept. Этот тип сервиса обычно используется серверами-посредниками.
Callback NAS Prompt	Пользователь должен быть отключен и соединен с NAS по инициативе последнего с предоставлением пользователю возможности ввода консольных команд NAS, не требующих специальных привилегий.
Call Check	Используется NAS в пакетах Access-Request для индикации приема вызова и факта, что серверу RADIUS следует возвратить пакет Access-Accept для ответа или Access-Reject для отказа от приема вызова (обычно это решается на основе атрибутов Called-Station-Id или Calling-Station-Id). Рекомендуется в таких запросах Access-Requests использовать значение Calling-Station-Id как значение User-Name.
Callback Administrative	Пользователь должен быть отключен и соединен с NAS по инициативе последнего с предоставлением пользователю административного интерфейса с сервером NAS, на котором будут выполняться команды, требующие определенных привилегий.

5.7. Framed-Protocol

Этот атрибут показывает тип кадров, которые будут использоваться для соединения. Атрибут **может** передаваться в пакетах Access-Request и Access-Accept.

Формат атрибута Framed-Protocol показан ниже. Поля передаются слева направо.



Type = 7

Length = 6

Value

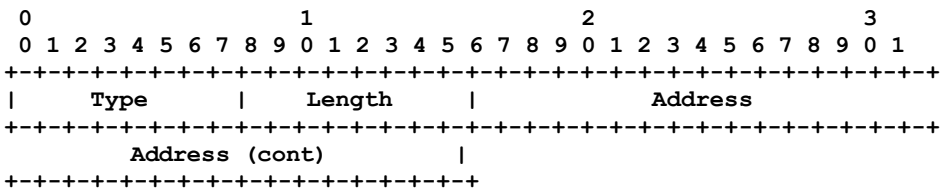
Четырехоктетное поле Value содержит идентификатор протокола.

Значение	*Протокол*	*Значение*	*Протокол*	*Значение*	*Протокол*
1	PPP	3	AppleTalk Remote Access Protocol (ARAP)	5	Xylogics proprietary IPX/SLIP
2	SLIP	4	Gandalf proprietary SingleLink/MultiLink protocol	6	X.75 Synchronous

5.8. Framed-IP-Address

Этот атрибут показывает предоставленный пользователю адрес. Атрибут **может** использоваться в пакетах Access-Accept. Этот атрибут **можно** также включать в запросы Access-Request как совет от сервера NAS по поводу предпочтительного адреса. Сервер не обязан принимать во внимание такие советы.

Формат атрибута Framed-IP-Address показан ниже. Поля передаются слева направо.



Type = 8

Length = 6

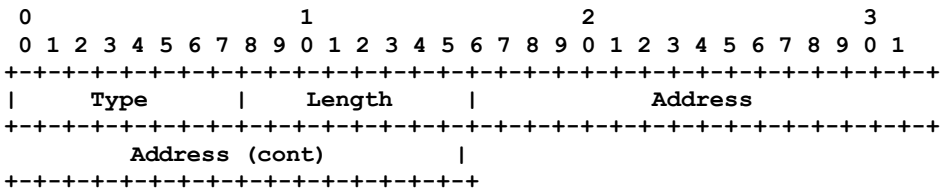
Address

Поле Address имеет размер 4 октета. Значение 0xFFFFFFFF показывает, что серверу NAS следует позволить пользователю выбрать адрес (например, Negotiated). Значение 0xFFFFFFFE показывает, что серверу NAS следует выбрать адрес для пользователя (например, из пула адресов, выделенного для NAS). Остальные корректные значения указывают серверу NAS значение IP-адреса, предоставляемого пользователю.

5.9. Framed-IP-Netmask

Этот атрибут показывает маску подсети IP, которая предоставляется пользователю в тех случаях, когда этот пользователь является маршрутизатором для сети. Этот атрибут **может** использоваться в пакетах Access-Accept. Атрибут можно также включать в пакеты Access-Request в качестве рекомендации со стороны NAS значения предпочтительной для пользователя маски. Сервер не обязан принимать эти рекомендации во внимание.

Формат атрибута Framed-IP-Netmask показан ниже. Поля передаются слева направо.



Type = 9

Length = 6

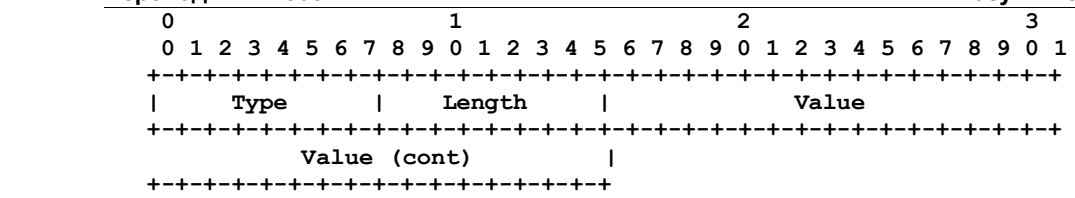
Address

Четырехоктетное поле Address содержит значение маски подсети, выделенное для пользователя.

5.10. Framed-Routing

Этот атрибут показывает метод маршрутизации для пользователя в тех случаях, когда пользователь является маршрутизатором для сети. Атрибут может использоваться только в пакетах Access-Accept.

Формат атрибута Framed-Routing показан ниже. Поля передаются слева направо.



Type = 10

Length = 6

Value

Четырехоктетное поле Value задает тип маршрутизации.

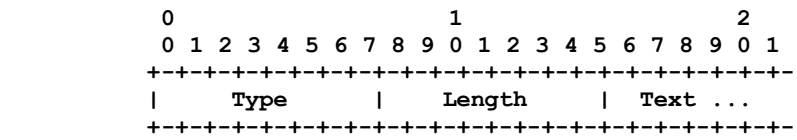
Значение	*Тип маршрутизации*	*Значение*	*Тип маршрутизации*
0	Нет	2	Принимать пакеты маршрутизации
1	Передавать пакеты маршрутизации	3	Принимать и передавать пакеты маршрутизации

5.11. Filter-Id

Этот атрибут показывает имя списка фильтров для данного пользователя. Пакеты Access-Accept **могут** включать в себя множество атрибутов Filter-Id.

Идентификация списков по именам позволяет использовать фильтры на различных NAS независимо от деталей реализации списков фильтрации.

Формат атрибута Filter-Id Attribute показан ниже. Поля передаются слева направо.



Type = 11

Length ≥ 3

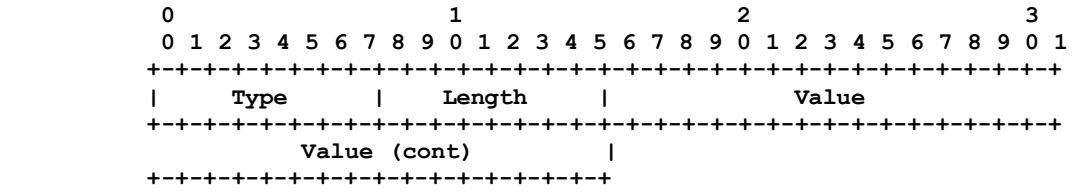
Text

Поле Text содержит содержит 0 или более октетов, содержимое которых зависит от реализации. Идентификаторы фильтров должны быть понятны человеку; **недопустимо** влияние этих идентификаторов на работу протокола. Рекомендуется в качестве значений атрибута использовать текстовые сообщения в кодировке UTF-8 10646 [7].

5.12. Framed-MTU

Этот атрибут показывает устанавливаемое для пользователя значение MTU[9], если это значение не согласуется иным способом (например, на уровне PPP). Атрибут **может** использоваться в пакетах Access-Accept. **Возможно** включение этого атрибута в пакеты Access-Request в качестве рекомендации серверу NAS, но сервер не обязан следовать этим рекомендациям.

Формат атрибута Framed-MTU показан ниже. Поля передаются слева направо.



Type = 12

Length = 6

Value

Поле Value содержит 4 октета, указывающие значение MTU, которое должно находиться в диапазоне от 64 до 65535 (несмотря на то, что формат поля позволяет задать большие значения).

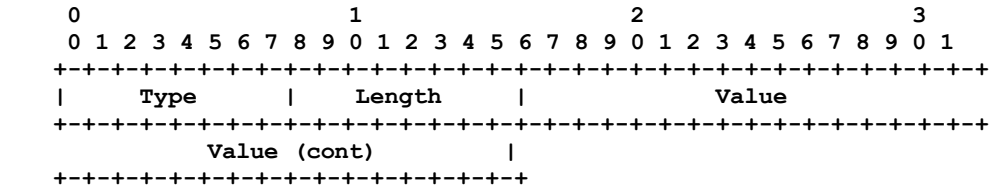
5.13. Framed-Compression

Этот атрибут показывает протокол компрессии, который будет использоваться для соединения. Атрибут **может** использоваться в пакетах Access-Accept. **Можно** включать этот атрибут в пакеты Access-Request в качестве рекомендации, но сервер не обязан принимать этот совет во внимание.

Пакет **может** содержать несколько атрибутов. Ответственность за поддержку протоколов компрессии ложится на сервер NAS.

Формат атрибута Framed-Compression показан ниже. Поля передаются слева направо.

[9] Maximum Transmission Unit – максимальный размер передаваемого блока информации (кадра). *Прим. перев.*



Type = 13

Length = 6

Value

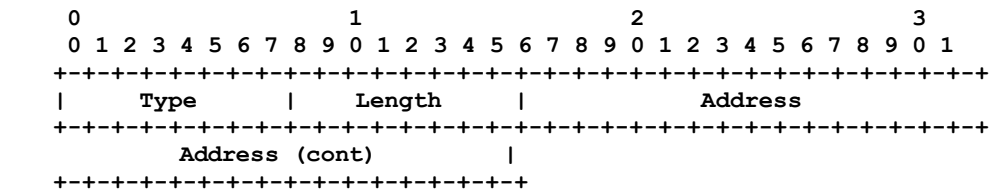
Четырехоктетное поле, указывающее протокол сжатия данных.

Значение	*Тип компрессии*	*Значение*	*Тип компрессии*
0	Без компрессии	2	Сжатие заголовков IPX
1	Сжатие заголовков VJ [10]	3	Сжатие Stac-LZS

5.14. Login-IP-Host

Этот атрибут указывает систему (хост) к которой хочет подключиться пользователь при наличии в пакете атрибута Login-Service. Атрибут **может** включаться в пакеты Access-Accept. **Можно** использовать атрибут в запросах Access-Request как рекомендацию, но сервер не обязан следовать такие рекомендациям.

Формат атрибута Login-IP-Host показан ниже. Поля передаются слева направо.



Type = 14

Length = 6

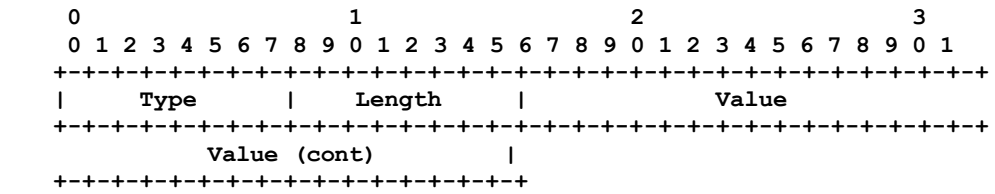
Address

Поле Address имеет размер 4 октета. Значение 0xFFFFFFFF показывает, что серверу NAS **следует** позволить пользователю указать адрес хоста. Нулевое значение адреса говорит, что серверу NAS **следует** выбрать адрес самостоятельно. Все прочие значения указывают адрес, который серверу NAS **следует** выбрать для подключения пользователя.

5.15. Login-Service

Этот атрибут указывает службу, которая будет использоваться для входа пользователя в систему. Атрибут включается только в пакеты Access-Accept.

Формат атрибута Login-Service показан ниже. Поля передаются слева направо.



Type = 15

Length = 6

Value

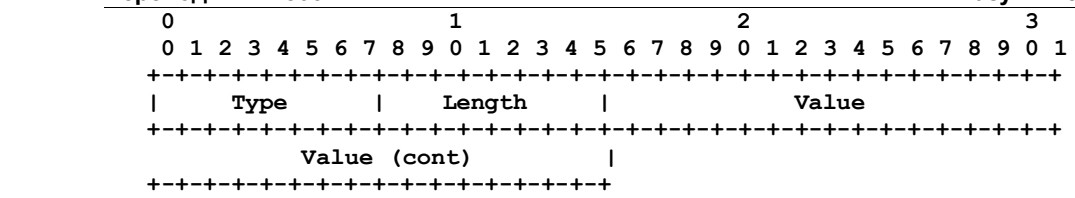
Четырехоктетное поле Value указывает тип сервиса для удаленного входа в систему.

Значение	*Служба*	*Значение*	*Служба*	*Значение*	*Служба*
0	Telnet	3	PortMaster (фирменный)	6	X25-T3POS
1	Rlogin	4	LAT	8	TCP Clear Quiet (подавляется любой строкой connect от NAS)
2	TCP Clear	5	X25-PAD		

5.16. Login-TCP-Port

Этот атрибут показывает порт TCP, к которому пользователь будет подключаться, если в пакете присутствует атрибут Login-Service. Атрибут включается только в пакеты Access-Accept.

Формат атрибута Login-TCP-Port показан ниже. Поля передаются слева направо.



Type = 16

Length = 6

Value

Четырехоктетное поле Value может содержать значение в диапазоне от 0 до 65535.

5.17. (не используется)

Атрибуты типа 17 не используются.

5.18. Reply-Message

Этот атрибут содержит текст, который **может** выводиться на консоль пользователя.

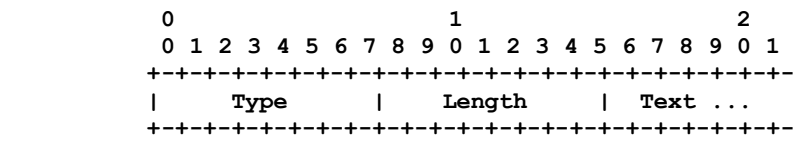
В пакетах Access-Accept этот атрибут содержит информацию об успешной аутентификации пользователя.

В пакетах Access-Reject этот атрибут служит для передачи информации об отказе. Атрибут **может** содержать приглашение пользователю для ввода дополнительной информации перед новой попыткой передачи пакета Access-Request.

В пакетах Access-Challenge этот атрибут **может** содержать приглашение пользователя к диалогу для ввода дополнительной информации (отклика).

Пакет **может** содержать несколько атрибутов данного типа. Если такие атрибуты отображаются пользователю, они **должны** выводиться в порядке их следования в пакете.

Формат атрибута Reply-Message показан ниже. Поля передаются слева направо.



Type = 18

Length ≥ 3

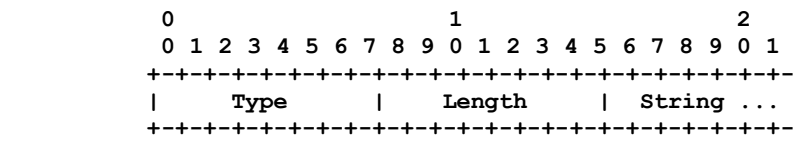
Text

Поле Text содержит по крайней мере один октет. Содержимое поля определяется реализацией. Предполагается, что значение атрибута понятно человеку. **Недопустимо** влияние этого атрибута на работу протокола. Рекомендуется в качестве значения атрибута использовать текст в кодировке UTF-8 10646 [7].

5.19. Callback-Number

Этот атрибут показывает строку набора номера, используемую для организации обратного звонка (callback). Атрибут **может** использоваться в пакетах Access-Accept. **Можно** использовать также в пакетах Access-Request как совет серверу по выбору службы Callback, но сервер не обязан принимать во внимание такие советы.

Формат атрибута Callback-Number показан ниже. Поля передаются слева направо.



Type = 19

Length ≥ 3

String

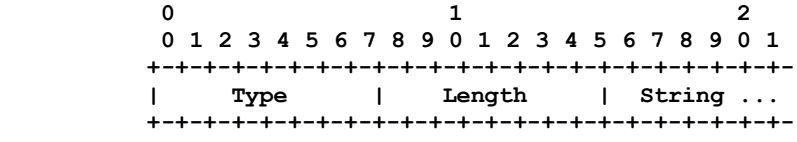
Поле String содержит по крайней мере один октет. Формат реального значения поля зависит от сайта и приложения, в целях повышения устойчивости рекомендациям **следует** поддерживать трактовку этого поля как неразобранных данных.

Рассмотрение возможных значений поля выходит за пределы данной спецификации.

5.20. Callback-Id

Этот атрибут показывает с кем будет организовано “обратное соединение” с точки зрения NAS. Атрибут может использоваться в пакетах Access-Accept.

Формат атрибута Callback-Id показан ниже. Поля передаются слева направо.



Type = 20

Length ≥ 3

String

Поле String содержит по крайней мере один октет. Формат реального значения поля зависит от сайта и приложения, в целях повышения устойчивости рекомендациям **следует** поддерживать трактовку этого поля как неразобранных данных.

Рассмотрение возможных значений поля выходит за пределы данной спецификации.

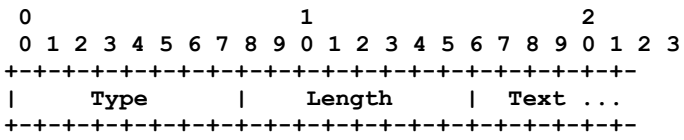
5.21. (не используется)

Атрибуты типа 21 не используются.

5.22. Framed-Route

Этот атрибут содержит маршрутную информацию для пользователя NAS. Атрибут включается в пакеты Access-Accept и может присутствовать в нескольких экземплярах.

Формат атрибута Framed-Route показан ниже. Поля передаются слева направо.



Type = 22

Length ≥ 3

Text

Поле Text содержит по крайней мере один октет. Конкретное содержимое поля зависит от реализации. Содержащаяся в поле информация предназначена для человека, воздействие значения этого поля на работу протокола **недопустимо**. Рекомендуется включать в это поле текстовое сообщение в кодировке UTF-8 10646 [7].

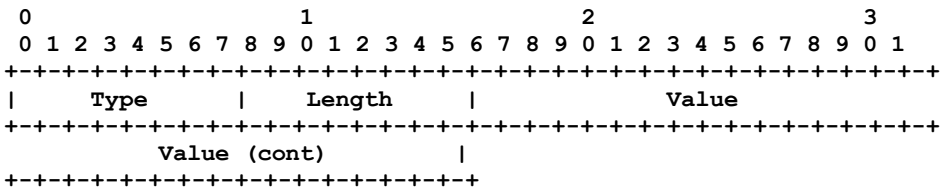
Для маршрутов IP в поле **следует** включать префикс адресата[10] в десятичном формате с разделением точками и указанием размера маски[11] после символа дробной черты (слэш). После префикса следует пробел, адрес шлюза в десятичном формате с разделением точками, пробел и одно или несколько значений метрики, разделенных пробелами. Примером может служить строка "192.168.1.0/24 192.168.1.1 1 2 -1 3 400". Размер маски может быть опущен; в таких случаях предполагается принятая по умолчанию маска /8 для префиксов класса A, /16 для префиксов класса B и 24 для префиксов класса C. Например, строка может иметь вид "192.168.1.0 192.168.1.1 1".

В тех случаях когда шлюз указан в форме "0.0.0.0" в качестве IP-адреса шлюза **следует** указывать выделенный пользователю адрес.

5.23. Framed-IPX-Network

Этот атрибут показывает номер сети IPX для пользователя. Атрибут включается в пакеты Access-Accept.

Формат атрибута Framed-IPX-Network показан ниже. Поля передаются слева направо.



Type = 23

Length = 6

Value

Поле Value содержит 4 октета. Значение 0xFFFFFFFE показывает, что серверу NAS **следует** выбрать для пользователя номер сети IPX (например, из пула адресов NAS). Другие значения указывают конкретный номер сети IPX для пользователя.

5.24. State

Этот атрибут доступен для передачи от сервера к клиентам в пакетах Access-Challenge и **должен** передаваться без изменения от клиента к серверу в новом пакете Access-Request, являющемся откликом на запрос (challenge), если таковой отклик передается.

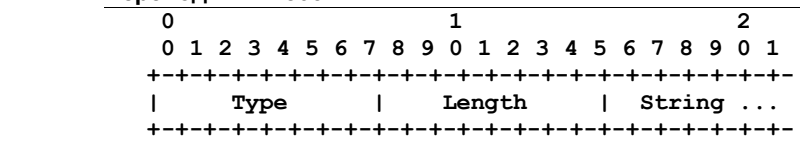
Этот атрибут доступен для передачи от сервера к клиентам в пакетах Access-Accept, которые включают также атрибут Termination-Action Attribute со значением RADIUS-Request. Если сервер NAS выполняет операцию Termination-Action путем передачи нового пакета Access-Request при разрыве текущего сеанса, он **должен** включить атрибут State в этот пакет Access-Request без изменения атрибута.

В любом случае для клиента **недопустима** локальная интерпретация атрибута. Пакет не может включать более одного атрибута State. Использование атрибутов State зависит от реализации.

Формат атрибута State показан ниже. Поля передаются слева направо.

[10] В оригинале - destination prefix. *Прим. перев.*

[11] Количество старших битов префикса, которые имеют значение.



Type = 24

Length ≥ 3

String

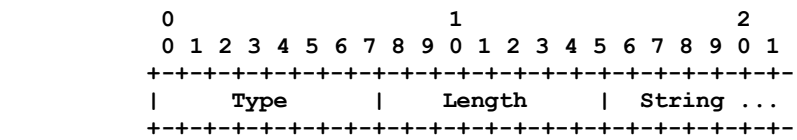
Поле String содержит по крайней мере один октет. Формат реального значения поля зависит от сайта и приложения, в целях повышения устойчивости рекомендациям **следует** поддерживать трактовку этого поля как неразобранных данных.

Рассмотрение возможных значений поля выходит за пределы данной спецификации.

5.25. Class

Этот атрибут доступен для передачи от сервера к клиенту в пакетах Access-Accept. Клиентам **следует** пересылать атрибут без его изменения серверам учета как часть пакета Accounting-Request, если в системе поддерживается учет. Для клиентов **недопустима** локальная интерпретация данного атрибута.

Формат атрибута Class показан ниже. Поля передаются слева направо.



Type = 25

Length ≥ 3

String

Поле String содержит по крайней мере один октет. Формат реального значения поля зависит от сайта и приложения, в целях повышения устойчивости рекомендациям **следует** поддерживать трактовку этого поля как неразобранных данных.

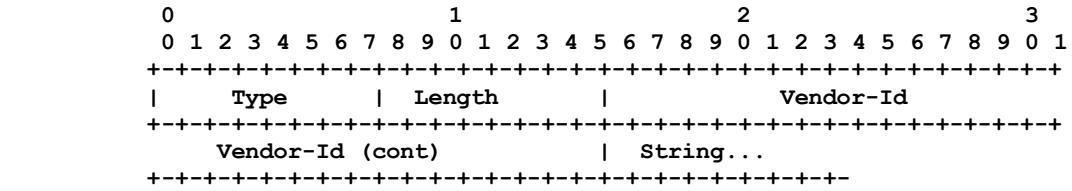
Рассмотрение возможных значений поля выходит за пределы данной спецификации.

5.26. Vendor-Specific

Этот атрибут позволяет разработчикам поддерживать фирменные атрибуты, недоступные или неподходящие для общего пользования. **Недопустимо** влияние таких атрибутов на работу протокола RADIUS.

Серверы, не понимающие полученные от клиента фирменные атрибуты, **должны** игнорировать их (допускается генерация отчетов о получении таких атрибутов). Клиентам, не получившим желаемого отклика на фирменный атрибут, **следует** предпринять попытку работы без такого атрибута, хотя бы в усеченном режиме (генерируя соответствующий отчет).

Формат атрибута Vendor-Specific показан ниже. Поля передаются слева направо.



Type = 26

Length ≥ 7

Vendor-Id

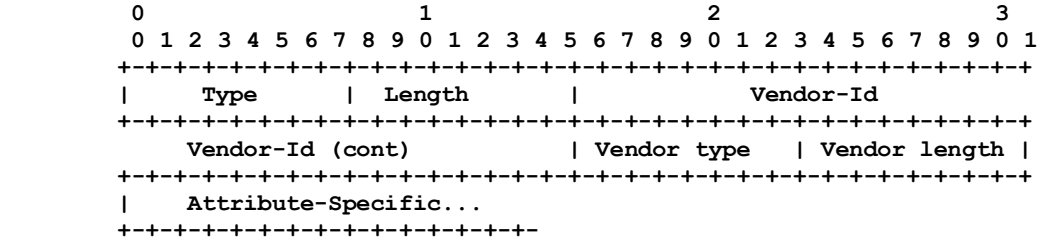
Старший октет этого поля имеет значение 0, а три младших октета содержат код SMI Network Management Private Enterprise для производителя в соответствии с "Assigned Numbers" RFC [6].

String

Поле String содержит по крайней мере один октет. Формат реального значения поля зависит от сайта и приложения, в целях повышения устойчивости рекомендациям **следует** поддерживать трактовку этого поля как неразобранных данных.

Рассмотрение возможных значений поля выходит за пределы данной спецификации.

Это поле следует представлять в формате vendor type / vendor length / value, как показано ниже. Поле Attribute-Specific зависит от принятого разработчиком определения для этого поля. Пример атрибута Vendor-Specific показан ниже:

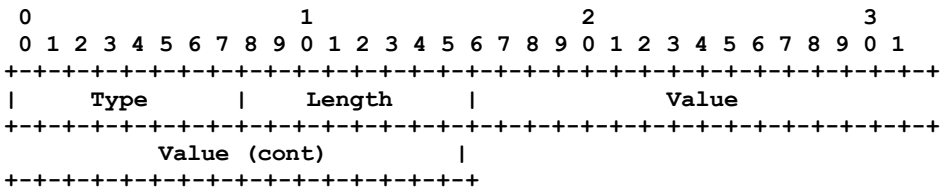


В одном атрибуте Vendor-Specific **может** содержаться множество определенных разработчиком субатрибутов.

5.27. Session-Timeout

Этот атрибут задает максимальную продолжительность (в секундах) пользовательского сеанса. Атрибут доступен для передачи в пакетах Access-Accept и Access-Challenge.

Формат атрибута Session-Timeout показан ниже. Поля передаются слева направо.



Type = 27

Length = 6

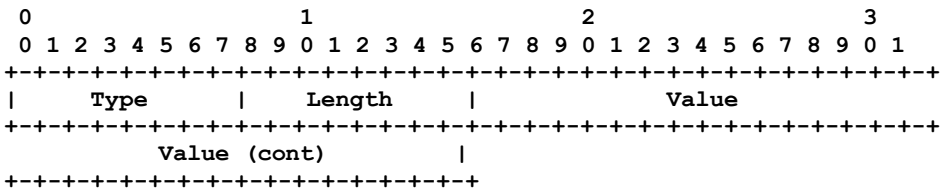
Value

Четырехоктетное поле, содержащее 32-битовое беззнаковое целое число, определяющее продолжительность пользовательского сеанса (соединения с NAS) в секундах.

5.28. Idle-Timeout

Этот параметр задает максимальный период непрерывного бездействия, по истечении которого пользовательский сеанс прерывается. Атрибут доступен для передачи от сервера к клиентам в пакетах Access-Accept и Access-Challenge.

Формат атрибута Idle-Timeout показан ниже. Поля передаются слева направо.



Type = 28

Length = 6

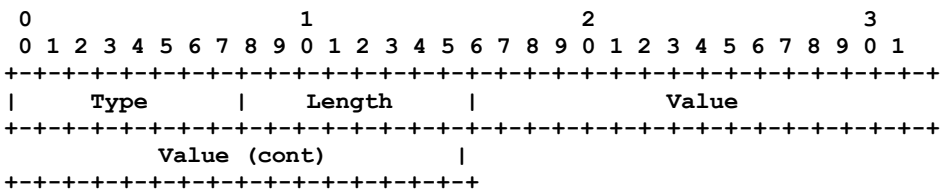
Value

Четырехоктетное поле, содержащее 32-битовое беззнаковое целое число, которое определяет максимальное непрерывное время бездействия в секундах, по истечении которого пользователь отключается от сервера NAS.

5.29. Termination-Action

Атрибут указывает действие, выполняемое сервером NAS по завершении сеанса, и может использоваться только в пакетах Access-Accept.

Формат атрибута Termination-Action показан ниже. Поля передаются слева направо.



Type = 29

Length = 6

Value

Четырехоктетное поле Value может содержать одно из двух значений.

0 принятое по умолчанию поведение

1 RADIUS-Request

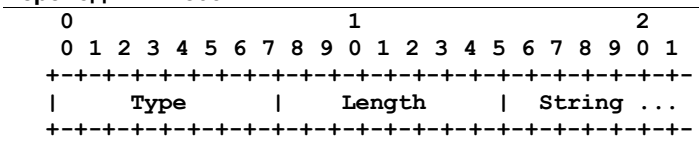
При установке значения RADIUS-Request в случае прерывания обслуживания сервер NAS **может** передать серверу RADIUS новый запрос Access-Request, включив в него атрибут State при наличии такового.

5.30. Called-Station-Id

Этот атрибут позволяет серверу NAS передать в пакете Access-Request номер телефона, набранный пользователем и определенный с помощью DNIS[12]) или иной технологии. Отметим, что этот номер может отличаться от реального номера, с которым было организовано соединение. Атрибут может использоваться только в пакетах Access-Request.

Формат атрибута Called-Station-Id показан ниже. Поля передаются слева направо.

[12] Dialed Number Identification – идентификация вызывающего номера.



Type = 30

Length ≥ 3

String

Поле String содержит по крайней мере один октет.

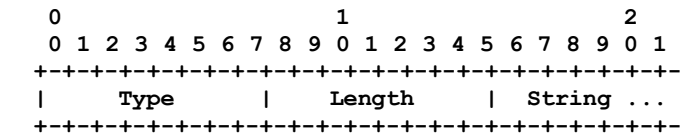
Формат реального значения поля зависит от сайта и приложения. Рекомендуется использовать текст в кодировке UTF-8 10646 [7], а в целях повышения устойчивости рекомендациям **следует** поддерживать трактовку этого поля как неразобранных данных.

Рассмотрение возможных значений поля выходит за пределы данной спецификации.

5.31. Calling-Station-Id

Этот атрибут позволяет серверу NAS передать в пакете Access-Request телефонный номер пользователя, определенный с помощью ANI[13] или иной технологии. Атрибут может использоваться только в пакетах Access-Request.

Формат атрибута Calling-Station-Id показан ниже. Поля передаются слева направо.



Type = 31

Length ≥ 3

String

Поле String содержит по крайней мере один октет.

Формат реального значения поля зависит от сайта и приложения. Рекомендуется использовать текст в кодировке UTF-8 10646 [7], а в целях повышения устойчивости рекомендациям **следует** поддерживать трактовку этого поля как неразобранных данных.

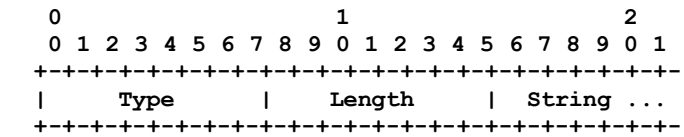
Рассмотрение возможных значений поля выходит за пределы данной спецификации.

5.32. NAS-Identifier

Этот атрибут содержит строку идентификации сервера NAS, передавшего пакет Access-Request. Атрибут используется только в пакетах Access-Request. Один из атрибутов NAS-IP-Address или NAS-Identifier **должен** присутствовать в каждом пакете Access-Request.

Отметим, что значение атрибута NAS-Identifier **недопустимо** использовать для выбора разделяемого ключа в процессе аутентификации запроса. Для выбора такого ключа **должно** использоваться значение IP-адреса отправителя в пакете Access-Request.

Формат атрибута NAS-Identifier показан ниже. Поля передаются слева направо.



Type = 32

Length ≥ 3

String

Поле String содержит по крайней мере один октет и должно быть уникальным идентификатором NAS в пределах видимости сервера RADIUS. Например, в качестве NAS-Identifier может использоваться полное доменное имя сервера доступа.

Формат реального значения поля зависит от сайта и приложения, в целях повышения устойчивости рекомендациям **следует** поддерживать трактовку этого поля как неразобранных данных.

Рассмотрение возможных значений поля выходит за пределы данной спецификации.

5.33. Proxy-State

Этот атрибут доступен для передачи сервером-посредником другому серверу при пересылке пакета Access-Request и **должен** быть возвращен в неизменном виде в пакете Access-Accept, Access-Reject или Access-Challenge. Когда proxy-сервер получает отклик на свой запрос, он **должен** удалить свой атрибут Proxy-State (последний атрибут данного типа в пакете) до пересылки отклика серверу NAS.

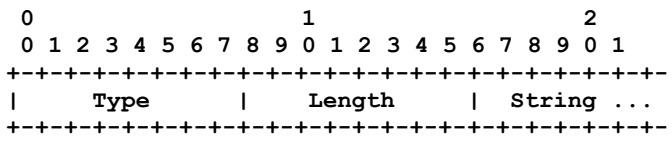
При добавлении атрибута Proxy-State в пересылаемый пакет этот атрибут **должен** добавляться после всех имеющихся в пакете атрибутов Proxy-State.

Содержимое любого атрибута Proxy-State, кроме добавленного данным сервером атрибута этого типа, должно трактоваться как непонятные данные; **недопустимо** влияние этих атрибутов на работу протокола.

[13] Automatic Number Identification – AOH.

Использование атрибутов Proxy-State зависит от реализации. Описание этих функций выходит за пределы настоящей спецификации.

Формат атрибута Proxy-State показан ниже. Поля передаются слева направо.



Type = 33

Length ≥ 3

String

Поле String содержит по крайней мере один октет. Формат реального значения поля зависит от сайта и приложения, в целях повышения устойчивости рекомендациям **следует** поддерживать трактовку этого поля как неразобранных данных.

Рассмотрение возможных значений поля выходит за пределы данной спецификации.

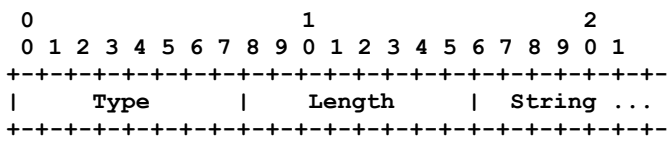
5.34. Login-LAT-Service

Этот атрибут указывает систему, к которой пользователь подключается по протоколу LAT. Атрибут **может** использоваться в пакетах Access-Accept при условии, что протокол LAT указан как Login-Service. **Можно** включать этот атрибут в пакеты Access-Request в качестве рекомендации серверу, но сервер не обязан следовать таким советам.

Администраторы используют этот атрибут в системах с кластерами (например, VAX или Alpha). В таких средах несколько хостов могут поддерживать ресурс совместного использования (диски, принтеры и т. п.) в режиме разделения времени и администраторы зачастую настраивают систему таким образом, чтобы предоставлялся доступ к каждому из разделяемых ресурсов. В этом случае каждый хост в кластере анонсирует свой сервис с помощью широковещательных пакетов LAT.

Изощренные пользователи зачастую знают какая из машин (поставщиков услуг) работает быстрее и при организации соединений LAT задают имя узла. Возможны также случаи, когда администраторы хотят привязать обслуживание конкретных пользователей к определенным машинам (как примитивный вариант распределения нагрузки), хотя хосты LAT сами могут балансировать загрузку.

Формат атрибута Login-LAT-Service показан ниже. Поля передаются слева направо.



Type = 34

Length ≥ 3

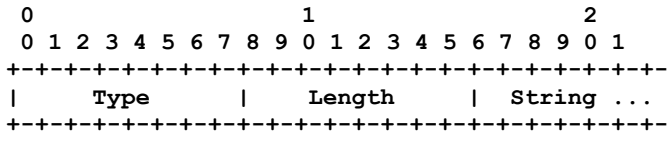
String

Поле String содержит по крайней мере один октет, указывающий сервис LAT. Архитектура LAT позволяет включать в строку символы $ (доллар), - (дефис), . (точка), _ (подчеркивание), числа, строчные м прописные буквы, а также символы расширенного набора ISO Latin-1 [11]. При сравнении строк LAT прописные и строчные буквы не различаются.

5.35. Login-LAT-Node

Этот атрибут указывает узел, к которому пользователь будет автоматически подключен по протоколу LAT. Атрибут **может** использоваться в пакетах Access-Accept, но только при указании LAT в качестве Login-Service. **Возможно** использование атрибута в пакетах Access-Request как рекомендации серверу, но сервер не обязан следовать таким рекомендациям.

Формат атрибута Login-LAT-Node показан ниже. Поля передаются слева направо.



Type = 35

Length ≥ 3

String

Поле String содержит по крайней мере один октет, идентифицирующий узел LAT, к которому подключается пользователь. Архитектура LAT позволяет включать в строку символы $ (доллар), - (дефис), . (точка), _ (подчеркивание), числа, строчные м прописные буквы, а также символы расширенного набора ISO Latin-1 [11]. При сравнении строк LAT прописные и строчные буквы не различаются.

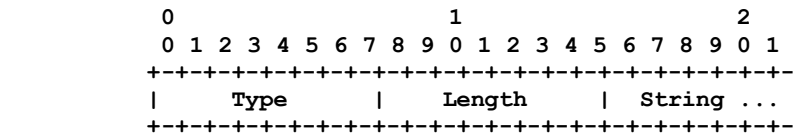
5.36. Login-LAT-Group

Этот атрибут содержит строку, идентифицирующую коды группы LAT, которую пользователю разрешено применять. Атрибут **можно** использовать в пакетах Access-Accept, но только при указании LAT в качестве Login-Service. **Можно** включать атрибут в пакеты Access-Request как рекомендацию для сервера, но сервер не обязан принимать такие рекомендации во внимание.

LAT поддерживает 256 различных групп, которые служат для управления правами доступа пользователей. Для представления кодов используются битовые маски размером 256.

Администраторы могут выбрать один или множество кодов групп доступа для сервис-провайдера LAT. Провайдер будет принимать только такие соединения, которые имеют соответствующие биты в маске прав. Администраторы создают маску прав доступа для каждого пользователя. LAT получает биты прав от операционной системы и использует их при передаче запросов сервис-провайдерам.

Формат атрибута Login-LAT-Group показан ниже. Поля передаются слева направо.



Type = 36

Length = 34

String

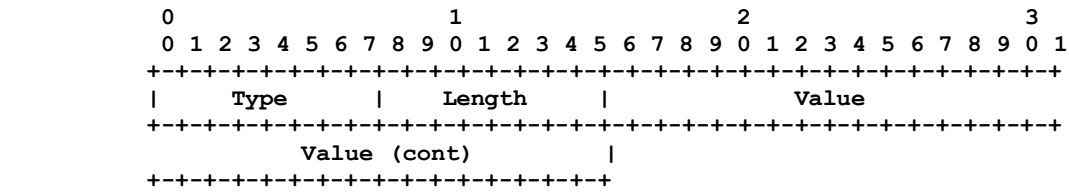
Поле имеет размер 32 октета. Для повышения устойчивости реализациям протокола следует поддерживать эти поля как необрабатываемые данные.

Кодирование прав доступа выходит за пределы данной спецификации.

5.37. Framed-AppleTalk-Link

Этот атрибут показывает номер сети AppleTalk, который следует для последовательного канала пользователя, который является другим маршрутизатором AppleTalk. Атрибут передается только в пакетах Access-Accept. Этот атрибут не применяется, если пользователь не является маршрутизатором.

Формат атрибута Framed-AppleTalk-Link показан ниже. Поля передаются слева направо.



Type = 37

Length = 6

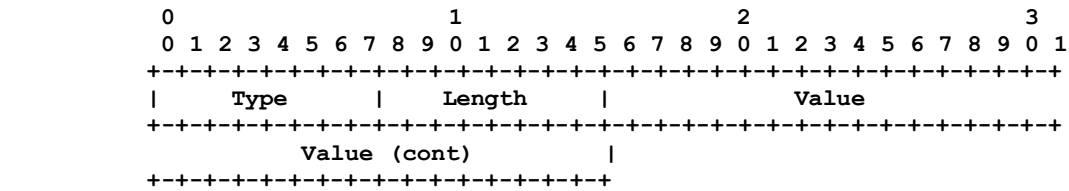
Value

Поле Value имеет размер 4 октета. Диапазон допустимых значений атрибута составляет 0 – 65535. Специальное значение 0 указывает на безадресный (unnumbered) последовательный канал. Значение от 1 до 65535 говорят о том, что последовательному каналу между NAS и пользователем следует присвоить указанное значение в качестве номера сети AppleTalk.

5.38. Framed-AppleTalk-Network

Этот атрибут показывает номер сети AppleTalk, который серверу NAS следует проверить для выделения номера узла AppleTalk пользователю. Атрибут передается только в пакетах Access-Accept. Этот атрибут никогда не применяется, если пользователь является маршрутизатором. Наличие более одного экземпляра данного атрибута показывает, что сервер NAS может пытаться использовать любой из указанных номеров сетей.

Формат атрибута Framed-AppleTalk-Network показан ниже. Поля передаются слева направо.



Type = 38

Length = 6

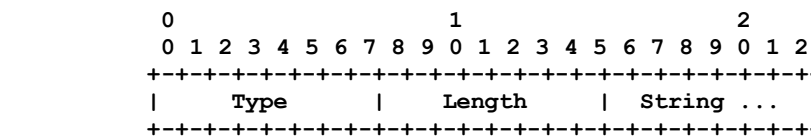
Value

Поле Value имеет размер 4 октета. Диапазон допустимых значений атрибута составляет 0 – 65535. Специальное значение 0 показывает, что серверу NAS следует выбрать для пользователя номер сети из принятого по умолчанию диапазона. Значения от 1 до 65535 (включительно) показывают номер сети AppleTalk, которую серверу NAS следует проверить для выделения адреса.

5.39. Framed-AppleTalk-Zone

Этот атрибут указывает принятую по умолчанию зону AppleTalk для данного пользователя. Атрибут передается только в пакетах Access-Accept и не допускается использование в одном пакете нескольких экземпляров атрибута.

Формат атрибута Framed-AppleTalk-Zone показан ниже. Поля передаются слева направо.



Type = 39

Length ≥ 3

String

Имя Default AppleTalk Zone для данного пользователя. В целях повышения устойчивости реализациям протокола **следует** поддерживать это поле как необрабатываемые данные.

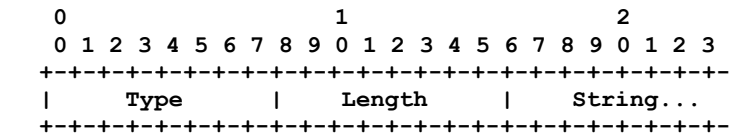
Кодирование этого поля выходит за пределы настоящей спецификации.

5.40. CHAP-Challenge

Этот атрибут содержит запрос CHAP Challenge, передаваемый сервером NAS пользователю PPP CHAP[14]. Атрибут применяется только в пакетах Access-Request.

Если значение CHAP challenge имеет размер 16 октетов, оно **может** быть помещено в поле Request Authenticator вместо использования данного атрибута.

Формат атрибута CHAP-Challenge показан ниже. Поля передаются слева направо.



Type = 60

Length ≥ 7

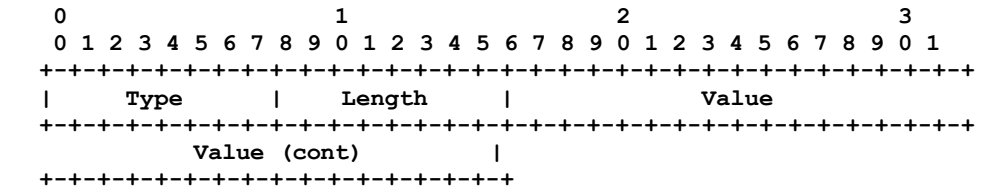
String

Поле String содержит значение CHAP Challenge.

5.41. NAS-Port-Type

Этот атрибут показывает тип физического порта сервера NAS, через который подключается пользователь. Атрибут может передаваться вместо атрибута NAS-Port (5) или в дополнение к нему. Атрибут передается только в пакетах Access-Request. В каждый пакет Access-Request **следует** включать атрибут NAS-Port (5) или NAS-Port-Type (или оба атрибута), если сервер NAS может различать свои порты.

Формат атрибута NAS-Port-Type показан ниже. Поля передаются слева направо.



Type = 61

Length = 6

Value

Поле Value имеет размер 4 октета. Значение Virtual указывает, что соединение с NAS осуществляется через некий протокол транспортного уровня вместо физического порта. Например, если пользователь обращается к NAS из сессии telne для идентификации себя как Outbound-User, пакет Access-Request может включать атрибут -Port-Type = Virtual в качестве указания серверу RADIUS что пользователь не связан с физическим портом.

Значение	*Тип порта*	*Значение*	*Тип порта*
0	Async	10	G.3 Fax
1	Sync	11	SDSL - Symmetric DSL
2	ISDN Sync	12	ADSL-CAP - Asymmetric DSL, модуляция CAP
3	ISDN Async V.120	13	ADSL-DMT - Asymmetric DSL, Discrete Multi-Tone
4	ISDN Async V.110	14	IDSL - ISDN Digital Subscriber Line
5	Virtual	15	Ethernet
6	PIAFS	16	xDSL – DSL неизвестного типа
7	HDLC Clear Channel	17	Cable
8	X.25	18	Беспроводные каналы (не IEEE 802.11)
9	X.75	19	Беспроводные каналы IEEE 802.11

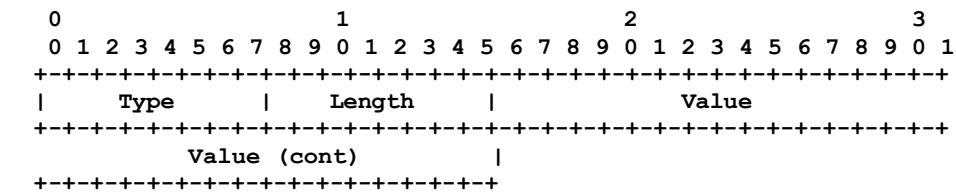
PIAFS[15] представляет собой беспроводный вариант ISDN, используемый в Японии.

[14] Challenge-Handshake Authentication Protocol
[15] PHS (Personal Handyphone System) Internet Access Forum Standard

5.42. Port-Limit

Этот атрибут устанавливает максимальное число портов NAS, открытых для подключения пользователя. Атрибут **может** передаваться сервером клиенту в пакетах Access-Accept. Назначение атрибута состоит в управлении числом портов для организации "параллельных" соединений с помощью Multilink PPP [12] или аналогичных протоколов. Атрибут **может** также передаваться NAS в качестве рекомендации серверу относительно желаемого числа портов, но сервер не обязан следовать таким рекомендациям.

Формат атрибута Port-Limit показан ниже. Поля передаются слева направо.



Type = 62

Length = 6

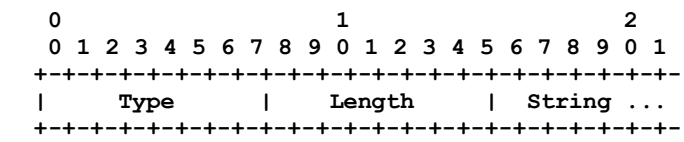
Value

Четырехоктетное поле содержит 32-битовое беззнаковое целое число, указывающее количество портов, которые сервер NAS может выделить для подключения пользователя.

5.43. Login-LAT-Port

Этот атрибут указывает порт, через который пользователь будет подключаться по протоколу LAT. Атрибут **может** передаваться в пакетах Access-Accept, но только при указании протокола LAT в качестве Login-Service. **Можно** использовать атрибут в пакетах Access-Request как рекомендацию для сервера, но сервер не обязан принимать такие рекомендации во внимание.

Формат атрибута Login-LAT-Port показан ниже. Поля передаются слева направо.



Type = 63

Length ≥ 3

String

Поле String содержит по крайней мере один октет, идентифицирующий порт LAT, который будет использоваться для подключения. Архитектура LAT позволяет включать в строку символы $ (доллар), - (дефис), . (точка), _ (подчеркивание), числа, строчные м прописные буквы, а также символы расширенного набора ISO Latin-1 [11]. При сравнении строк LAT прописные и строчные буквы не различаются.

5.44. Таблица атрибутов

В таблице приведен список атрибутов с указанием типа пакетов, в которые каждый атрибут может быть включен и допустимого числа экземпляров атрибута в пакете.

Request	*Accept*	*Reject*	*Challenge*	#	*Attribute*
0-1	0-1	0	0	1	User-Name
0-1	0	0	0	2	User-Password[16]
0-1	0	0	0	3	CHAP-Password[16]
0-1	0	0	0	4	NAS-IP-Address[17]
0-1	0	0	0	5	NAS-Port
0-1	0-1	0	0	6	Service-Type
0-1	0-1	0	0	7	Framed-Protocol
0-1	0-1	0	0	8	Framed-IP-Address
0-1	0-1	0	0	9	Framed-IP-Netmask
0	0-1	0	0	10	Framed-Routing
0	0+	0	0	11	Filter-Id
0-1	0-1	0	0	12	Framed-MTU
0+	0+	0	0	13	Framed-Compression
0+	0+	0	0	14	Login-IP-Host

[16] Пакет Access-Request **должен** включать атрибут User-Password, CHAP-Password или State. **Недопустимо** одновременное включение в пакеты Access-Request атрибутов User-Password и CHAP-Password. Если в будущем появятся новые типы аутентификации для использования в пакетах Access-Request, соответствующие атрибуты будут применяться взамен User-Password или CHAP-Password.

[17] Пакеты Access-Request **должны** включать по крайней мере один из атрибутов NAS-IP-Address и NAS-Identifier.

Request	*Accept*	*Reject*	*Challenge*	#	*Attribute*
0	0-1	0	0	15	Login-Service
0	0-1	0	0	16	Login-TCP-Port
0	0+	0+	0+	18	Reply-Message
0-1	0-1	0	0	19	Callback-Number
0	0-1	0	0	20	Callback-Id
0	0+	0	0	22	Framed-Route
0	0-1	0	0	23	Framed-IPX-Network
0-1	0-1	0-1	0	24	State[18]
0	0+	0	0	25	Class
0+	0	0	0+	26	Vendor-Specific
0	0-1	0	0-1	27	Session-Timeout
0	0-1	0	0-1	28	Idle-Timeout
0	0-1	0	0	29	Termination-Action
0-1	0	0	0	10	Called-Station-Id
0-1	0	0	0	31	Calling-Station-Id
0-1	0	0	0	32	NAS-Identifier[19]
0+	0+	0+	0+	33	Proxy-State
0-1	0-1	0	0	34	Login-LAT-Service
0-1	0-1	0	0	35	Login-LAT-Node
0-1	0-1	0	0	36	Login-LAT-Group
0	0-1	0	0	37	Framed-AppleTalk-Link
0	0+	0	0	38	Framed-AppleTalk-Network
0	0-1	0	0	39	Framed-AppleTalk-Zone
0-1	0	0	0	60	CHAP-Challenge
0-1	0	0	0	61	NAS-Port-Type
0-1	0-1	0	0	62	Port-Limit
0-1	0-1	0	0	63	Login-LAT-Port

Приведенные для каждого типа пакетов количественные значения имеют следующий смысл.

0 **недопустимо** включение данного атрибута в пакеты этого типа;

0+ атрибут является необязательным и **может** присутствовать в нескольких экземплярах;

0-1 необязательный атрибут, который **может** присутствовать в единственном экземпляре;

1 обязательный атрибут, который **должен** присутствовать в единственном экземпляре;

6. Согласование с IANA

В этом разделе содержатся рекомендации по регистрации в IANA (Internet Assigned Numbers Authority) значения, связанных с протоколом RADIUS, как это указано в документе BCP 26 [13].

существует три области имен RADIUS, требующие регистрации: коды типа пакетов (Packet Type Code), типы атрибутов (Attribute Type) и, в отдельных случаях, значения атрибутов (Attribute Value).

Протокол RADIUS не предназначен для использования в качестве протокола управления общего назначения для серверов NAS (Network Access Server) и выделение значений не должно осуществляться для целей, отличных от AAA (Authentication, Authorization, Accounting – идентификация, проверка полномочий, учет).

6.1. Определения терминов

BCP 26 содержит определения используемых здесь терминов "name space" (пространство имен), "assigned value" (присвоенное значение), "registration" (регистрация).

BCP 26 содержит определения используемых здесь правил: "Private Use" (приватное использование), "First Come First Served" (первым пришел – первого обслужили), "Expert Review" (требуется одобрение экспертов), "Specification Required" (требуется спецификация), "IETF Consensus" (согласие IETF), "Standards Action" (значения присваиваются только для RFC, одобренных IESG).

[18] Пакет Access-Request **должен** включать атрибут User-Password, CHAP-Password или State. **Недопустимо** одновременное включение в пакеты Access-Request атрибутов User-Password и CHAP-Password. Если в будущем появятся новые типы аутентификации для использования в пакетах Access-Request, соответствующие атрибуты будут применяться взамен User-Password или CHAP-Password.

[19] Пакеты Access-Request **должны** включать по крайней мере один из атрибутов NAS-IP-Address и NAS-Identifier.

6.2. Рекомендуемая политика регистрации

Для регистрационных запросов, которые требуют экспертизы (Designated Expert), экспертов назначает IESG Area Director for Operations.

Для регистрационных запросов, требующих обзора экспертов (Expert Review), следует обращаться к списку рассылки ietf-radius.

Коды типа пакетов (Packet Type Code) имеют значения в диапазоне от 1 до 254, из которого значения 1 – 5, 11 - 13 уже использованы. Поскольку новые типы пакетов будут оказывать влияние на взаимодействие реализаций, выделение новых типов рассматривается как Standards Action. Для новых типов следует использовать коды, начиная с 14.

Идентификаторы типа атрибутов имеют значения в диапазоне от 1 до 255. Для протокола RADIUS такое количество идентификаторов достаточно мало, поэтому их распределение требует внимания. Значения 1 – 53, 55, 60 – 88, 90 - 91 уже распределены, при этом атрибуты 17 и 21 доступны для повторного использования. Для выделения значений 17, 21, 54, 56 - 59, 89, 92 - 191 требуется уровень Expert Review вкупе со Specification Required. Выделение блоков значения (более 3) требует согласования с IETF (IETF Consensus). Рекомендуется использовать значения 17 и 21 лишь после того, как будут распределены все другие значения.

Отметим, что протокол RADIUS определяет механизм расширений Vendor-Specific (атрибут 26) и следует по возможности пользоваться этим расширением вместо выделения новых глобальных типов, если атрибут связан с реализациями RADIUS одного производителя и для его использования не требуется интероперабельности с другими реализациями.

Как было отмечено выше в разделе "Атрибуты":

"[Type] Значения 192-223 предназначены для экспериментальных целей, значения 224-240 зарезервированы для разработчиков (специфические для реализации типы), а значения 241-255 являются резервными и не должны использоваться.."

Следовательно, атрибуты 192-240 рассматриваются как приватные (Private Use), а значения 241-255 требуют Standards Action.

Некоторые атрибуты (например, NAS-Port-Type) протокола RADIUS определяют список значений, которые могут иметь разный смысл. Для каждого из таких атрибутов возможны 4 миллиарда (2^{32}) значений. Добавление в такие списки новых значений осуществляется по принципу First Come, First Served[20] в понимании IANA.

7. Примеры

Ниже представлено несколько примеров пакетов, содержащих достаточно типичные варианты атрибутов. Список примеров не является исчерпывающим – можно найти множество других вариантов. Шестнадцатеричные дампы пакетов приводятся с использованием сетевого порядка следования байтов для разделяемого ключа xyzzy5461.

7.1. Telnet-доступ к заданному хосту

Сервер NAS с адресом 192.168.1.16 передает пакет UDP типа Access-Request серверу RADIUS для пользователя с именем nemo, подключающегося через порт 3 с паролем arctangent.

Поле Request Authenticator (16 октетов) содержит случайное значение, созданное NAS.

Поле User-Password является результатом применения операции XOR по отношению к 16-октетному паролю (может быть дополнен нулями до 16 октетов) и MD5(shared secret|Request Authenticator).

```
01 00 00 38 0f 40 3f 94 73 97 80 57 bd 83 d5 cb
98 f4 22 7a 01 06 6e 65 6d 6f 02 12 0d be 70 8d
93 d4 13 ce 31 96 e4 3f 78 2a 0a ee 04 06 c0 a8
01 10 05 06 00 00 00 03
```

1 Code = Access-Request (1)

1 ID = 0

2 Length = 56

16 Request Authenticator

Атрибуты:

6 User-Name = "nemo"

18 User-Password

6 NAS-IP-Address = 192.168.1.16

6 NAS-Port = 3

Сервер RADIUS проверяет пользователя nemo и передает пакет UDP типа Access-Accept серверу NAS, позволяющий клиенту nemo доступ по протоколу telnet к хосту 192.168.1.3.

Поле Response Authenticator содержит 16-октетную контрольную сумму MD5 полей code (2), id (0), Length (38), поля Request Authenticator из предыдущего пакета, атрибутов отклика и разделяемого ключа.

```
02 00 00 26 86 fe 22 0e 76 24 ba 2a 10 05 f6 bf
9b 55 e0 b2 06 06 00 00 00 01 0f 06 00 00 00 00
0e 06 c0 a8 01 03
```

1 Code = Access-Accept (2)

1 ID = 0 (тот же самый, что в пакете Access-Request)

2 Length = 38

16 Response Authenticator

[20] Т. е., явочным порядком. *Прим. перев.*

Атрибуты:

6 Service-Type (6) = Login (1)

6 Login-Service (15) = Telnet (0)

6 Login-IP-Host (14) = 192.168.1.3

7.2. Framed-сервис с использованием аутентификации CHAP

Сервер NAS с адресом 192.168.1.16 передает UDP-пакет Access-Request серверу RADIUS для аутентификации пользователя flopsy, подключающегося к порту 20 по протоколу PPP с использованием CHAP. Сервер NAS вместе в атрибутами Service-Type и Framed-Protocol передает серверу RADIUS сведения о том, что пользователь желает работать по протоколу PPP, хотя NAS не обязан передавать такие рекомендации.

Атрибут Request Authenticator содержит 16-октетное случайное значение, созданное NAS, которое также используется в CHAP Challenge.

Атрибут CHAP-Password содержит 1-октетное значение CHAP ID (в данном случае 22), за которым следует 16 октетов отклика CHAP response.

```
01 01 00 47 2a ee 86 f0 8d 0d 55 96 9c a5 97 8e
0d 33 67 a2 01 08 66 6c 6f 70 73 79 03 13 16 e9
75 57 c3 16 18 58 95 f2 93 ff 63 44 07 72 75 04
06 c0 a8 01 10 05 06 00 00 00 14 06 06 00 00 00
02 07 06 00 00 00 01
```

1 Code = 1 (Access-Request)

1 ID = 1

2 Length = 71

16 Request Authenticator

Атрибуты:

8 User-Name (1) = "flopsy"

19 CHAP-Password (3)

6 NAS-IP-Address (4) = 192.168.1.16

6 NAS-Port (5) = 20

6 Service-Type (6) = Framed (2)

6 Framed-Protocol (7) = PPP (1)

Сервер RADIUS идентифицирует flopsy и передает UDP-пакет Access-Accept серверу NAS, говорящий тому о возможности использования PPP и выделении пользователю адреса из динамического пула.

Атрибут Response Authenticator содержит 16-октетное хэш-значение MD5 для (2), id (1), Length (56), поля Request Authenticator из предыдущего пакета, атрибутов данного отклика и разделяемого ключа.

```
02 01 00 38 15 ef bc 7d ab 26 cf a3 dc 34 d9 c0
3c 86 01 a4 06 06 00 00 00 02 07 06 00 00 00 01
08 06 ff ff ff fe 0a 06 00 00 00 02 0d 06 00 00
00 01 0c 06 00 00 05 dc
```

1 Code = Access-Accept (2)

1 ID = 1 (совпадает со значением в пакете Access-Request)

2 Length = 56

16 Response Authenticator

Атрибуты:

6 Service-Type (6) = Framed (2)

6 Framed-Protocol (7) = PPP (1)

6 Framed-IP-Address (8) = 255.255.255.254

6 Framed-Routing (10) = None (0)

6 Framed-Compression (13) = VJ TCP/IP Header Compression (1)

6 Framed-MTU (12) = 1500

7.3. Пользователь подключается с помощью карты Challenge-Response

NAS с адресом 192.168.1.16 передает UDP-пакет Access-Request серверу RADIUS для пользователя mopsy, подключающегося через порт 7. Пользователь ввел пароль challenge. Генерируемые смарт-картой запрос (challenge) и отклик имеют значения 32769430 и 99101462, соответственно.

Атрибут Request Authenticator содержит 16-октетное случайное значение, созданное NAS.

Атрибут User-Password содержит результат операции XOR для 16 октетов пароля (в данном случае challenge, с дополнением нулями справа) и MD5(shared secret|Request Authenticator).

```
01 02 00 39 f3 a4 7a 1f 6a 6d 76 71 0b 94 7a b9
30 41 a0 39 01 07 6d 6f 70 73 79 02 12 33 65 75
73 77 82 89 b5 70 88 5e 15 08 48 25 c5 04 06 c0
a8 01 10 05 06 00 00 00 07
```

1 Code = Access-Request (1)

1 ID = 2

2 Length = 57

16 Request Authenticator

Атрибуты:

7 User-Name (1) = "mopsy"

18 User-Password (2)

6 NAS-IP-Address (4) = 192.168.1.16

6 NAS-Port (5) = 7

Сервер RADIUS передает пользователю mopsy дополнительный запрос, на который ожидает ответа. Следовательно, RADIUS передает UDP-пакет Access-Challenge серверу NAS.

Атрибут Response Authenticator содержит 16-октетной хэш-значение MD5 для code (11), id (2), length (78), атрибута Request Authenticator из предыдущего пакета, атрибутов данного отклика и разделяемого ключа.

Атрибут Reply-Message имеет значение "Challenge 32769430. Enter response at prompt."

Атрибут State представляет собой значение magic cookie, возвращаемое с откликом пользователя (в нашем примере это 8 октетов - 33 32 37 36 39 34 33 30 в шестнадцатеричном представлении).

```
0b 02 00 4e 36 f3 c8 76 4a e8 c7 11 57 40 3c 0c
71 ff 9c 45 12 30 43 68 61 6c 6c 65 6e 67 65 20
33 32 37 36 39 34 33 30 2e 20 20 45 6e 74 65 72
20 72 65 73 70 6f 6e 73 65 20 61 74 20 70 72 6f
6d 70 74 2e 18 0a 33 32 37 36 39 34 33 30
```

1 Code = Access-Challenge (11)

1 ID = 2 (совпадает со значение в пакете Access-Request)

2 Length = 78

16 Response Authenticator

Атрибуты:

48 Reply-Message (18)

10 State (24)

Пользователь вводит свой отклик и NAS передает новый пакет Access-Request, включая в него атрибут State.

Атрибут Request Authenticator содержит новое 16-октетное случайное значение.

Атрибут User-Password представляет собой 16-октетное значение, полученное с помощью операции XOR для пользовательского отклика (в данном случае 99101462), дополненного справа нулями и MD5(shared secret|Request Authenticator).

Атрибут State содержит значение magic cookie из пакета Access-Challenge.

```
01 03 00 43 b1 22 55 6d 42 8a 13 d0 d6 25 38 07
c4 57 ec f0 01 07 6d 6f 70 73 79 02 12 69 2c 1f
20 5f c0 81 b9 19 b9 51 95 f5 61 a5 81 04 06 c0
a8 01 10 05 06 00 00 00 07 18 10 33 32 37 36 39
34 33 30
```

1 Code = Access-Request (1)

1 ID = 3 (отметим, что значение идентификатора изменилось)

2 Length = 67

16 Request Authenticator

Атрибуты:

7 User-Name = "mopsy"

18 User-Password

6 NAS-IP-Address (4) = 192.168.1.16

6 NAS-Port (5) = 7

10 State (24)

Значение атрибута Response (отклик пользователя) было некорректным (для примера), поэтому сервер RADIUS говорит NAS о необходимости отвергнуть попытку подключения.

Атрибут Response Authenticator представляет собой 16-октетное хэш-значение MD5 для code (3), id (3), length(20), атрибута Request Authenticator из предыдущего пакета, атрибутов данного отклика (если они имеются) и разделяемого ключа.

```
03 03 00 14 a4 2f 4f ca 45 91 6c 4e 09 c8 34 0f
9e 74 6a a0
```

1 Code = Access-Reject (3)

1 ID = 3 (совпадает со значением в пакете Access-Request)

2 Length = 20

16 Response Authenticator

Атрибуты:

Атрибуты отсутствуют, хотя включение Reply-Message допускается.

8. Вопросы безопасности

Вопросы безопасности являются основной темой данного документа.

На практике в каждом сервере RADIUS (или на связанном с ним дополнительном сервере) имеется база данных, содержащая список имен пользователей и сопоставленных каждому пользователю идентификационных данных (secret). Не делается предположений, что пользователь с данным именем будет идентифицироваться с использованием множества методов. Такой подход является уязвимым для атак с использованием наименее безопасного метода идентификации из числа поддерживаемых. Для предотвращения таких атак следует для каждого пользователя указать единственный метод идентификации. Если пользователю нужны различные варианты идентификации в разных случаях, для решения проблемы **следует** создавать такому пользователю несколько записей с различными именами, каждая из которых будет использовать свою схему идентификации.

Пароли и другая конфиденциальная информация должны храниться на отвечающей стороне так, чтобы доступ к ним был максимально ограничен. В идеальном случае доступ к таким сведениям следует предоставлять только процессу, выполняющему функции аутентификации.

Ключи следует передавать с использованием механизмов, обеспечивающих минимальное число участвующих в передаче (и следовательно, способных узнать ключи) объектов. В идеальном случае о наличии ключа следует знать лишь лицам, наделенным соответствующими полномочиями. Передачу ключей можно обеспечить с помощью протоколов SNMP Security [14], но рассмотрение этих механизмов выходит за пределы данной спецификации.

Прочие методы распространения ключей требуют дополнительных исследований и экспериментов. В описывающем протокол SNMP Security документе [14] также содержится превосходный обзор связанных с сетевыми протоколами опасностей.

Механизм сокрытия User-Password, описанный в параграфе 5.2 не подвергался достаточному криптоанализу по опубликованным источникам. Некоторые члены сообщества IETF высказывают сомнения [15] в том, что этот механизм обеспечивает достаточный уровень конфиденциальности при передаче паролей в системах RADIUS. Пользователям следует оценить уровень безопасности своей среды и при необходимости подключить дополнительные механизмы обеспечения конфиденциальности.

9. Журнал изменений

Ниже приведен список изменений, внесенных в спецификацию, по сравнению с RFC 2138:

- Переменные типа string должны использовать кодировку UTF-8 вместо US-ASCII, и трактовать их следует как 8-битовые данные.
- Целые числа и даты задаются как 32-битовые беззнаковые целые числа.
- Для обеспечения совместимости с таблицей атрибутов обновлен список атрибутов, которые могут включаться в Access-Challenge.
- User-Name ссылается на идентификаторы NAI (Network Access Identifier).
- User-Name можно передавать в пакетах Access-Accept для учета и использования с rlogin.
- Добавлены значения для атрибутов Service-Type, Login-Service, Framed-Protocol, Framed-Compression и NAS-Port-Type.
- Атрибут NAS-Port может использовать все 32 бита.
- Примеры приведены с шестнадцатеричными дампами пакетов.
- Порт отправителя UDP должен использоваться вместе с атрибутом Request Identifier для идентификации дубликатов.
- Допускается включение множества субатрибутов в атрибуты Vendor-Specific.
- Пакет Access-Request должен содержать по крайней мере один из атрибутов NAS-IP-Address или NAS-Identifier.
- В раздел “Работа протокола” добавлена информация о серверах-посредниках (proxy), повторной передаче и пакетах keep-alive.
- При наличии множества однотипных атрибутов все серверы-посредники **должны** сохранять порядок таких атрибутов.
- Даны пояснения для Proxy-State.
- Даны пояснения об отсутствии зависимости трактовки атрибутов от их положения в пакете и необходимости сохранения порядка однотипных атрибутов.
- Добавлен раздел “Согласование с IANA”.
- Обновлен параграф “Сервер-посредник (Proxy)" в главе "Работа протокола".
- Атрибуты Framed-MTU могут передаваться в пакетах Access-Request в качестве рекомендации.
- Обновлен раздел “Вопросы безопасности”.

- Текстовые строки рассматриваются как подмножество string, для прояснения использования UTF-8.

10. Литература

[1] Rigney, C., Rubens, A., Simpson, W. and S. Willens, "Remote Authentication Dial In User Service (RADIUS)", RFC 2138, April 1997.

[2] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119[21], March, 1997.

[3] Rivest, R. and S. Dusse, "The MD5 Message-Digest Algorithm", RFC 1321[21], April 1992.

[4] Postel, J., "User Datagram Protocol", STD 6, RFC 768[21], August 1980.

[5] Rigney, C., "RADIUS Accounting", RFC 2866[21], June 2000.

[6] Reynolds, J. and J. Postel, "Assigned Numbers", STD 2, RFC 1700[22], October 1994.

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

[8] Aboba, B. and M. Beadles, "The Network Access Identifier", RFC 2486[21], January 1999.

[9] Kaufman, C., Perlman, R., and Speciner, M., "Network Security: Private Communications in a Public World", Prentice Hall, March 1995, ISBN 0-13-061466-1.

[10] Jacobson, V., "Compressing TCP/IP headers for low-speed serial links", RFC 1144, February 1990.

[11] ISO 8859. International Standard -- Information Processing -- 8-bit Single-Byte Coded Graphic Character Sets -- Part 1: Latin Alphabet No. 1, ISO 8859-1:1987.

[12] Sklower, K., Lloyd, B., McGregor, G., Carr, D. and T. Coradetti, "The PPP Multilink Protocol (MP)", RFC 1990, August 1996.

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

[14] Galvin, J., McCloghrie, K. and J. Davin, "SNMP Security Protocols", RFC 1352, July 1992.

[15] Dobbertin, H., "The Status of MD5 After a Recent Attack", CryptoBytes Vol.2 No.2, Summer 1996.

11. Подтверждение

Исходный вариант протокола RADIUS был разработан Стивом Виллисом (Steve Willens) из Livingston Enterprises для линейки серверов доступа PortMaster.

12. Адрес руководителя группы

Взаимодействием участников рабочей группы руководил:

Carl Rigney

Livingston Enterprises

4464 Willow Road

Pleasanton, California 94588

Phone: +1 925 737 2100

EMail: cdr@telemancy.com

13. Адреса авторов

Связанные с этим документом вопросы можно адресовать авторам:

Carl Rigney

Livingston Enterprises

4464 Willow Road

Pleasanton, California 94588

Phone: +1 925 737 2100

EMail: cdr@telemancy.com

Allan C. Rubens

Merit Network, Inc.

4251 Plymouth Road

Ann Arbor, Michigan 48105-2785

EMail: acr@merit.edu

William Allen Simpson

Daydreamer

[21] Перевод документа на русский язык вы найдете на сайте www.protocols.ru. *Прим. перев.*

[22] В соответствии с RFC 3232 документ STD 2 утратил силу. Значения Assigned Numbers следует искать в базе данных, доступной на сайте www.iana.org/numbers.html. *Прим. перев.*

Computer Systems Consulting Services

1384 Fontaine

Madison Heights, Michigan 48071

EMail: wsimpson@greendragon.com

Steve Willens

Livingston Enterprises

4464 Willow Road

Pleasanton, California 94588

EMail: steve@livingston.com

Перевод на русский язык

Николай Малых

nmalykh@bilim.com

14. Полное заявление авторских прав

Подтверждение

Финансирование функций RFC Editor обеспечивалось Internet Society.