Эссе по курсу «Защита информации», кафедра радиотехники,
Московский физико-технический институт(ГУ МФТИ),
http://re.mipt.ru/infsec

Протокол EAP

Студент: *Затеса Александр*
Васильевич, 114 группа

г.Долгопрудный
2005 г.

1. Введение

В последние годы беспроводные локальные сети(WLAN – Wireless Local Area Network) являются основным и самым быстроразвивающимся направлением развития сетевой индустрии. С каждым годом они находят всё большее применение в расширении LAN(решение проблемы «последней мили»), организации беспроводных сетей с инфраструктурой, с расширением рынка портативных компьютеров беспроводные технологии стали применяться и для обеспечения доступа мобильных устройств к ресурсам Internet.

Одним из наиболее распространённых стандартов, используемых при построении беспроводных сетей, является IEEE 802.11. Он описывает физический и канальный уровни (согласно модели OSI ISO) сети. Защита же в этом стандарте определена опционально. Поэтому проблема обеспечения безопасности в беспроводных сетях стоит очень остро, в связи с тем, что рост популярности беспроводных сетей повлёк за собой рост попыток несанкционированного доступа к ним. До недавнего времени широко применялся протокол WEP (Wired Equivalent Privacy). Но из-за ряда уязвимостей в нём, таких как отсутствие сервиса распределения ключей, линейность CRC, недостаточная длина секретного ключа и др., возникла необходимость в усовершенствовании и разработке новых процедур аутентификации и работы самого протокола. Одним из результатов этих разработок явился протокол EAP (Extended Authentication Protocol), который получил очень широкое применение для обеспечения безопасности в WLAN.

2. Описание протокола EAP

Протокол EAP предназначен для обеспечения расширенной аутентификации, когда IP протокол недоступен. Будучи изначально предназначенным для использования вместе с PPP(Point-to-Point Protocol) протоколом, протокол EAP нашёл также широкое применение в беспроводных сетях (IEEE-802.1X). Главной особенностью данного протокола является то, что механизм аутентификации определяется на более поздней фазе, уже после установления непосредственного соединения. Это позволяет аутентификатору получить некую дополнительную информацию о клиенте, который хочет быть авторизован. Итак, для начала надо ввести основные термины, которыми мы будем впоследствии оперировать:

- Аутентификатор(authenticator) - один из концов соединения, требующий аутентификацию.
- Клиент(peer) – другой конец соединения, который будет проходить процедуру аутентификации на аутентификаторе.

- AAA– аутентификация (Authentication), авторизация (Authorization) и учёт (Accounting). Наиболее известные AAA-протоколы с поддержкой EAP это RADIUS [RFC3579] and Diameter [I-D.ietf-aaa-eap].

Итак, как же происходит аутентификация с помощью протокола EAP?

1. После процедуры установления соединения аутентификатор посылает запрос клиенту, в котором содержится поле «тип», оно показывает, что именно запрашивает аутентификатор. Запрос может посылаться несколько раз.
2. Затем клиент посылает пакет с ответом аутентификатору, причем на каждый из запросов. В ответе также содержится поле «тип», отвечающее за то, на какой именно запрос делается ответ.
3. Аутентификатор заканчивает процедуру аутентификации, посылая пакет, сигнализирующий об успешной аутентификации(Success packet) или о неудавшейся аутентификации(Failure packet).

Последовательность пунктов 1-2 может повторяться необходимое число раз, определяемое реализацией. Причём EAP протокол пошаговый(lock-step), т.е. каждый последующий пакет с запросом посылается только после получения ответа на предыдущий запрос.

3. Формат пакетов протокола EAP

Пакет в протоколе EAP состоит из следующих полей

код	идентификатор	длина	тип	тип данных

1. Код – равен 1 для пакета с запросом, 2 - для пакета с ответом, 3 – для пакета, сигнализирующего об успешной аутентификации(success-пакет), 4 – аутентификацию не прошла(failure-пакет)
2. Идентификатор – поле, длиной в 8 бит, в ответе идентификатор должен быть таким же, что и в запросе. Во время сессии для каждого из запросов идентификатор выбирается уникальным в пределах одной сессии, однако при ретрансляции запроса по таймауту, идентификатор не должен меняться. Если клиент получает запросы с одинаковым идентификатором (например, ожидается ввод с консоли) то все дублирующиеся запросы он просто отбрасывает, оставляя один.
3. Длина – поле с длиной всего EAP-пакета.
4. Тип – поле, описывающее тип запроса или ответа. Обычно тип ответа такой же, как и тип запроса, но у ответа есть один тип Nak – который говорит, что клиенту не подходит тип запроса от аутентификатора.

Теперь стоит рассмотреть каждый из пакетов более подробно.
Назначение пакетов, сигнализирующих об успешной или неуспешной аутентификации, интуитивно понятно, поэтому стоит более детально рассмотреть пакеты с запросом и ответом. И более подробно следует остановиться на различных типах пакетов (определяемых полем «тип»). Поле «тип» имеет длину 8 бит. Первые четыре типа зарезервированы стандартом RFC 3748 как особые:
0 – reserved;
1 – identity;
2 – notification;
3 – Nak(reponse only).
Помимо этих типов, стандарт **обязывает** все реализации EAP протокола поддерживать ещё тип 4 – MD5-Challenge. Поддержка остальных типов опциональна. Однако очень широкое применение имеют также типы 5 – OTP (One-Time Password) и 6 – GTC (Generic Token Card). Типы с 9 по 45 уже определены, с 46 по 191 – свободны, с 192 по 253 – зарезервированы для дальнейшей стандартизации, 254 – Expanded type, 255 – экспериментальный тип. Типы с номерами 7 и 8 не определены.
Теперь рассмотрим более подробно каждый из основных типов.
Тип Identity.
Запрос с таким типом аутентификатором посылается в самую первую очередь во время процедуры аутентификации. Запрос служит для идентификации клиента и начала фазы аутентификации. В основном этот тип пакетов используется для определения, какой метод EAP использовать. Ответом на запрос с типом 1, должен быть пакет только с типом 1.
Тип Notification
Данный тип запросов применяется для передачи отображаемого сообщения от аутентификатора клиенту. Клиент может отображать сообщение пользователю, а может вести лог сообщений, если его нельзя по каким-либо причинам отобразить. Запросы с таким типом могут отсылаться в любое время процедуры аутентификации, ответом на такой пакет могут служить пакеты с таким же типом, ответы с типом Nak запрещены. Очень часто пакеты с таким типом не требуются, и в реализации можно запретить такие пакеты, тогда клиент будет просто удалять их.
Тип Nak
Данный тип пакетов используется только в ответах. Он сигнализирует о том, что предлагаемый в запросе тип аутентификации не приемлем у клиента. Ответ может содержать в себе один или несколько типов аутентификации, которые поддерживаются у

клиента. В ответе может быть 0, это говорит о том, что клиент не поддерживает другие типы аутентификации. После получения ответа типа Nak с кодом 0, аутентификатору следует прекратить слать пакеты с запросами на тип аутентификации. Данный тип Nak ответа называется legacy Nak(с кодом равным 3), помимо него существует также expanded Nak, он также используется только в ответах. Ответ с таким типом отсылается только на запросы с кодом 254 (expanded type), сам ответ с типом expanded Nak имеет также код 254. Предназначение у него такое же, как и у legacy Nak.

Тип MD5-Challenge

Когда передаются пакеты с таким типом, то это означает что периодически во время сессии проходит подтверждение идентичности клиента посредством «троекратного рукопожатия» (3-way handshake). Первое «рукопожатие» происходит сразу после установления связи.

1. Аутентификатор отправляет клиенту запрос содержащее «сообщение-вопрос»(challenge-message);
2. Клиент отвечает на этот запрос ответом либо с кодом 4, где в ответе содержится хэш-функция от определённой комбинации «сообщения-запроса» и секретного пароля (хэширование происходит по алгоритму MD-5), либо с кодом 3(Nak), где в ответе описан желаемый способ аутентификации. В этом случае обмен пакетами с типом 4 прекращается.
3. В противном случае при получении ответа с типом 4, аутентификатор проверяет полученный ответ и сверяет присланную хэш-функцию со своей, которую он посчитал сам (секретный пароль также хранится и на сервере аутентификации). Если хэш-функции разные то, аутентификатор прекращает сеанс связи, в противном случае сессия продолжается.

Далее во время сессии такое «рукопожатие» происходит через произвольные промежутки времени. Преимущества данного метода аутентификации состоит в том, что секретный пароль, который хранится у пользователя и аутентификатора не передаётся по сети. Важной частью данного метода также является уникальность посылаемых «сообщений-запросов», т.е. чтобы последовательность «сообщений-запросов» не содержала одинаковые элементы и не была предсказуемой. Данное условие является гарантией устойчивости к reply-атакам, когда подслушивающая сторона запоминает ответы на предыдущие запросы и посылает сохранённый ответ на повторно-встретившийся запрос. Недостатком данного метода аутентификации является то, что секретный пароль должен быть доступен в виде обычного текста, следовательно, данный механизм неприменим для систем, где пароли хранятся в базе данных в необратимо-зашифрованном виде.

Тип OTP (One-Time Password)

Данный тип пакетов используется для обеспечения аутентификации на основе последовательности «одноразовых паролей» (one-time password). Суть этого метода заключается в следующем. Аутентификатор и клиент имеют «секретный ключ» (или секретный пароль). В самом начале процедуры аутентификации аутентификатор отсылает клиенту «начальное сообщение» (seed). Это «начальное сообщение» каким-то образом соединяется с секретным паролем и преобразуется неким образом (алгоритмы разные). После этого результат этих преобразований, назовём его S, прогоняется N раз через хэш-функцию у клиента и аутентификатора. Таким образом, получается последовательность из N одноразовых паролей. Первый раз посылается N раз прохэшированное S, и аутентификатор сверяет свой и присланный результаты. Затем, когда аутентификатор присылает очередной запрос на подтверждение аутентификации, пусть в i-ый раз, в качестве пароля используется (N-i) раз прохэшированное S. При достижении N единицы, аутентификатор вновь инициирует передачу «начального сообщения» и далее всё повторяется по циклу. Основным преимуществом данного метода является то, что аутентификация повторяется во время сеанса связи, но пароль не передаётся по сети, а инициируется передача лишь последовательности одноразовых паролей. Безопасноть данного метода в основном определяется устойчивостью к коллизиям выбранного алгоритма хэширования.

4. Применение EAP-протокола в WLAN

Помимо рассмотрения формата пакетов и их основных типов, следует также кратко рассмотреть, как используется протокол EAP в беспроводных сетях. Его инкапсуляция определена в стандарте IEEE-802.1x. В этом стандарте также вводиться новое понятие – точка доступа. Аутентификатор и точка доступа разнесены физически. Клиент общается с аутентификатором через точку доступа, причём точка доступа не просто переправляет пакеты через себя, но ещё и просматривает их содержимое на предмет успешной или неуспешной аутентификации. При передаче Success-пакета точка доступа переправляет его клиенту, сама же открывает ему доступ к своим сервисам.

Как было описано выше, EAP предоставляет удобный и надёжный механизм по доставке ключей, но не по их распределению. Распределением ключей занимается специальный AAA-сервер, в его задачи входит генерация свежего сессионного ключа, определения основных параметров механизмов шифрования (механизмы подтверждения доставки, время жизни ключа и др.) и непосредственное распределение ключей. Все эти задачи AAA-сервер начинает выполнять, после того как пройдёт аутентификация одним из EAP-протоколов. Вид EAP протокола определяется в сервере доступа и в операционной

системы на стороне клиента. Точка доступа только перенаправляет пакеты между клиентами и аутентификатором и не требует изменения настроек или даже оборудования на данной точке доступа при смене вида EAP.

Как видно IEEE 802.1x предоставляет удобный, легко настраиваемый и эффективный унифицированный интерфейс для услуг в области аутентификации, авторизации и учета (AAA) при помощи EAP-протокола. Однако протокол EAP нельзя назвать именно методом аутентификации. Он определяет только протокольную основу для механизмов аутентификации и распределения ключей. Преимущество состоит в том, что производителям сетевого оборудования требуется внедрить исключительно протокол EAP, независимо от того, каким образом регулируется сама аутентификация между пользователем или клиентом и сервером аутентификации. Таким образом, разработчики могут выбирать между различными методами EAP, что представляется очевидным преимуществом. Однако это преимущество является таким же недостатком, т.к. современные тенденции роста беспроводных сетей влекут за собой рост количества методов аутентификации. А стандартом протокола EAP не установлен обязательный метод, который должны поддерживать все. Тогда возникает резонный вопрос, кто же будет определять этот метод. Если для небольшой WLAN это ещё как-то реализуемо, то для распределённой корпоративной WLAN это уже представляется нелёгкой задачей. Еще неприятнее окажется ситуация в "горячих точках" общего доступа, само назначение которых предполагает, что каждый клиент получает по возможности легкий и простой доступ в сеть. К сожалению, этот вопрос пока остаётся открытым и остается только ждать окончательной редакции стандарта 802.11i

5. Используемые материалы:

1. RFC 3748 “Extensible Authentication Protocol (EAP)”: http://rfc.net/rfc3748.html
2. RFC 3579 “RADIUS support for EAP” http://rfc.net/rfc3748.html#s7.13.
3. EAP Key Management Framework: http://bgp.potaroo.net/ietf/ids/draft-ietf-eap-keying-06.txt
4. IEEE 802.1X: EAP over LAN (EAPOL) for LAN/WLAN Authentication & Key Management http://www.javvin.com/protocol8021X.html
5. RFC 2289 “A One-Time Password System”: http://rfc.net/rfc2289.html
6. RFC 1994 “PPP Challenge Handshake Authentication Protocol (CHAP)”:

http://rfc.net/rfc1994.html

7. Extensible Authentication Protocol (EAP) Registry

http://www.iana.org/assignments/eap-numbers