ИПМ им.М.В.Келдыша РАН

Абрау-2019 • Труды конференции

А.В. Никешин, В.З. Шнитман

Верификация реализаций клиента протокола аутентификации EAP

Рекомендуемая форма библиографической ссылки

Никешин А.В., Шнитман В.З. Верификация реализаций клиента протокола аутентификации EAP // Научный сервис в сети Интернет: труды XXI Всероссийской научной конференции (23-28 сентября 2019 г., г. Новороссийск). — М.: ИПМ им. М.В.Келдыша, 2019. — С. 541-550. — URL: http://keldysh.ru/abrau/2019/theses/55.pdf doi:10.20948/abrau-2019-55

Размещена также презентация к докладу

Верификация реализаций клиента протокола аутентификации EAP

А.В. Никешин, В.З. Шнитман

Институт системного программирования Российской академии наук

Аннотация. В данной работе представлен опыт верификации реализаций клиентов протокола аутентификации EAP. EAP – широко используемый протокол аутентификации, реализующий большой набор криптографических алгоритмов и позволяющий динамически выбирать нужный в процессе аутентификации. Протокол EAP определяет для своих целей множество методов, среди которых есть как простые методы на основе контрольной суммы, так и туннельные методы, предполагающие создание защищенного туннеля, внутри которого применяются другие методы аутентификации. В работе использовался новый тестовый набор, разработанный с использованием технологии UniTESK и методов мутационного тестирования а также наработок коллектива в тестировании сетевых протоколов. Технология UniTESK позволяет автоматизировать процесс верификации сетевых протоколов на основе их формальных моделей, а методы мутационного тестирования позволяют протестировать устойчивость реализации протокола к искаженным сообщениям.

Ключевые слова: безопасность, аутентификация, EAP, методы EAP, протоколы, тестирование, оценка устойчивости, Интернет, стандарты, формальные методы спецификации

EAP Clients verification

A.V. Nikeshin, V.Z. Shnitman

Ivannikov Institute for System Programming of the Russian Academy of Sciences

Abstract. This paper presents the experience of verification of EAP authentication Protocol client implementations. EAP – a widely used authentication Protocol that implements a large set of cryptographic algorithms and allows you to dynamically select the desired authentication process. The EAP Protocol defines a variety of methods for its purposes, including both simple checksum - based methods and tunnel methods that involve the creation of a secure tunnel within which other authentication methods are used. The paper used a new test set developed using UniTESK technology and methods of mutation testing as well as team developments in the testing of network protocols. UniTESK technology allows to automate the verification process of network protocols on the basis of their formal models, and

mutation testing methods allow to test the stability of the Protocol implementation to distorted messages.

Keywords: security, authentication, EAP, EAP methods, protocols, testing, verification, evaluate robustness, Internet, standards, formal specifications

1. Особенности тестирования реализаций клиента EAP

EAP – широко используемый протокол аутентификации, реализующий большой набор криптографических алгоритмов и позволяющий динамически выбирать нужный в процессе аутентификации. Достаточно подробно особенности данного протокола и его методов изложены в нашей статье «Обзор расширяемого протокола аутентификации и его методов» [1],[2], поэтому здесь мы на них останавливаться не будем. Несколько наших работ посвящены тестированию серверной части протокола [3-6]. Данная работа является их продолжением и сосредоточена на клиентской части протокола.

Тестирование клиента EAP имеет свои особенности. При тестировании сервера мы довольствовались видимым сетевым трафиком. Поскольку «последнее слово» в аутентификации всегда остается за сервером, сетевого трафика достаточно, чтобы сделать выводы о результатах аутентификации. Клиент же не отвечает на последнее сообщение сервера, поэтому анализ сетевых пакетов позволяет отслеживать реакцию клиента на промежуточные сообщения сервера, но не дает информации о результате аутентификации на стороне клиента. Некоторой альтернативой могут служить методы EAP, поддерживающие защищенную индикацию результата. Кроме этого, результат аутентификации активно используется аутентификатором, который на его основании решает какие права представить клиенту. Одной из областей использования протокола EAP является разграничение сетевого доступа с предоставлением ip-адресов. В этом случае получение клиентом сетевого адреса может служить индикатором успешного завершения процесса аутентификации. Также неотъемлемой частью сервера является протоколирование своей работы (механизм журнализации событий), что позволяет отслеживать правильность работы сервера и выявлять причины сбоев. Клиентские же службы и низкоуровневые приложения, как правило, не ведут отчетов, и сообщения об ошибках можно найти лишь в общих системных журналах. Еще одно отличие состоит в том, что сервер это постоянно работающий процесс и его зависание, падение или перезагрузка свидетельствуют о сбоях в работе. Клиентский процесс выполняется определенное, как правило, короткое время, и делать выводы о результате его работы часто можно лишь по косвенным событиям или по записям из журналов событий.

В нашей работе в качестве тестируемых клиентов используется мобильный узел, подключенный через проводное соединение к коммутатору и использующий протокол 802.1x (EAP over LAN) в качестве транспорта для протокола EAP [7]. Результатом аутентификации является получение сетевого

адреса, который служит индикатором успешной аутентификации. В процессе тестирования мы используем не очень эффективную, но менее трудоемкую схему работы – мы не используем каких-либо дополнительных агентов на стороне клиента, запуск процесса получения сетевого адреса происходит в ручную. Это связано с тем, что при подключении к коммутатору клиент не имеет еще ни сетевых настроек, ни адреса и удаленное управление и тем более синхронизация работы с тестовой системой не возможна. При этом мы исходим из следующих положений:

- при настройке сетевого интерфейса в случае каких-либо ошибок операционная система обычно делает несколько попыток и некоторые ОС позволяют устанавливать их количество, таким образом несколько автоматизируя процесс тестирования,
- основная часть тестовых воздействий, являются некорректными и должны завершаться с ошибкой и попыткой повторного соединения,
- общее количество тестов не очень велико, что позволяет проводить тестирование за разумное время.

Индикатором успешного завершения аутентификации служит факт получения клиентом сетевого адреса.

Таким образом, тестовый сценарий представляет собой последовательность попыток клиента пройти аутентификацию, периодически прерываемую случаями успешной настройки сетевого интерфейса и возобновляемую в ручном режиме новой переустановкой сетевых параметров для продолжения сценария.

В большинстве случаев взаимодействие между аутентификатором и сервером EAP происходит по протоколу RADIUS [8, 9]. При этом, согласно спецификации [9], часть проверок корректности сообщений ложится на аутентификатор и, таким образом, не все тестовые сообщения доходят до клиента (такие воздействия можно рассматривать, как тестирование аутентификатора). Обойти такие ограничения позволяют туннельные методы EAP, создающие сначала защищенный канал, внутри которого происходит дальнейший обмен. Аутентификатор не видит сообщения внутри канала, что позволяет передавать клиенту любые данные.

2. Тестовый стенд

В данной работе мы используем стандартную схему организации доступа по протоколу EAP, состоящую из трех сетевых узлов, выполняющих следующие роли:

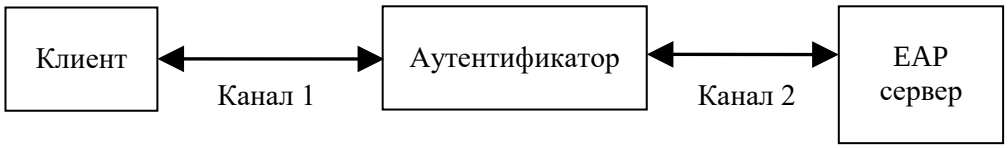

Рис. 1. Общая схема работы EAP

- Клиент: Мобильный узел, которому требуется пройти аутентификацию.
- Аутентификатор: Сетевой узел, с которым соединяется клиент. В общем случае аутентификатор используется как ретранслятор, передавая пакеты EAP между партнером и сервером EAP. Сервер EAP информирует аутентификатор о результате аутентификации. На основе этого результата аутентификатор либо предоставляет, либо запрещает доступ клиента к сети.
- Сервер EAP: Внутренний сервер, который выполняет аутентификацию клиента и определяет, прошла ли аутентификация успешно или нет. На нем исполняется основной поток управления тестовой системы под управлением UniTESK, обход тестового автомата и верификация наблюдаемых реакций. Тестовые сообщения протокола, сформированные модельной реализацией, передаются через аутентификатор тестируемой системе (клиенту), после чего регистрируются реакции тестируемого узла.

Клиент запрашивает доступ к сети, подключаясь к аутентификатору. Аутентификатор передает запрос с данными клиента серверу EAP. Сервер EAP запрашивает дополнительные данные у клиента. Обмен сообщениями между клиентом и сервером EAP продолжается до тех пор, пока выбранный метод аутентификации не завершится успешно или с ошибкой. На основании этого результата аутентификатор, принимает решение о предоставлении клиенту доступа к сети.

В качестве аутентификатора используется коммутатор Dell Networking N2048.

Протокол EAP может одновременно использоваться в разных средах передачи данных и, соответственно, выполняться через разные стеки сетевых протоколов. В нашем случае клиент и аутентификатор осуществляют обмен данными через проводной канал по протоколу 802.1x (EAP over LAN) [7]. Аутентификатор и сервер EAP осуществляют обмен данными через проводной канал поверх протокола AAA RADIUS [8].

3. Методы верификации

Тестирование реализаций сетевых протоколов направлено на решение двух важных задач: проверки совместимости различных реализаций и проверки их корректности и надежности.

В наших проектах мы используем, описанные нами неоднократно в предыдущих работах, наработанные методики по тестированию сетевых протоколов: автоматизированное тестирование на соответствие формальным спецификациям и методы мутации данных.

Поскольку в данном процессе участвуют разнородные объекты, возникает задача обеспечения совместимости реализаций, одним из основных методов решения которой является тестирование на соответствие стандарту.

В текущих экспериментах используется разработанная нами на основе спецификаций RFC модель протокола EAP и его методов, описывающая сложную схему функционирования протокола.

Тестирование реализаций на соответствие формальным спецификациям проводится с использованием технологии UniTESK, предоставляющей средства автоматизации тестирования на основе использования конечных автоматов (с использованием инструмента JavaTesK) [10],[11]. Состояния тестируемой системы определяют состояния автомата, а тестовые воздействия – переходы этого автомата. При выполнении перехода заданное воздействие передается на тестируемую реализацию, после чего регистрируются реакции реализации и автоматически выносится вердикт о соответствии наблюдаемого поведения спецификации. В UniTESK алгоритм обхода конечного автомата реализован как внутренний компонент и не зависит от протокола и тестируемой системы.

Особенность тестирования протоколов состоит в том, что требуется также проводить тестирование на нестандартных или искаженных входных данных, что довольно актуально для систем безопасности. Такие ситуации постоянно возникают из-за ошибок пользователей или из-за целенаправленных действий злоумышленника, но при этом часто недостаточно полно определены в спецификации протокола.

Методы мутационного тестирования используются для обнаружения неадекватного поведения тестируемой системы (завершение из-за фатальной ошибки, "подвисание", ошибки доступа к памяти). В сообщения, сформированные на основе разработанной модели протокола, вносятся какие-либо изменения, при этом модель протокола позволяет менять данные на любом этапе обмена, что позволяет тестовому сценарию проходить через все значимые состояния протокола и в каждом таком состоянии проводить тестирование реализации в соответствие с заданной программой.

Модель применяемого метода EAP определяет множество состояний тестируемой системы, на его основе строится конечный автомат, переходы которого сопоставлены с соответствующими тестовыми воздействиями. При выполнении перехода определенное воздействие передается на тестируемую реализацию, после чего регистрируются реакции реализации и проверяется

корректность наблюдаемого поведения системы. Обход автомата по всем достижимым состояниям модели протокола осуществляется инструментарием UniTESK. Методы мутации позволяют вносить искажения в сообщения на любом этапе обмена. При этом совместное использование корректных и измененных сообщений позволяет тестовому сценарию "преодолеть" необходимые проверки в реализации, пройти через все значимые состояния протокола и в каждом состоянии выполнить необходимый набор тестовых воздействий.

4. Тестирование реализации клиента протокола

Процесс обмена в нашем случае начинает клиент сообщением EAPOL-start, предлагая аутентификатору начать стандартный обмен EAP сообщением EAP-Request/Identity. Клиент отвечает сообщением EAP-Response/Identity, второе аутентификатор передает серверу (т.е. тестовому узлу). При запуске тестового сценария сервер переходит в режим ожидания сообщений от клиента. На их основе в соответствии с планом сценария создаются ответы сервера в модельном представлении, передаваемые затем функции отправки сообщений. В тестовом сценарии сообщения сервера являются стимулами, ответы клиента – реакциями. В предусловии спецификационных функций стимулов проверяется правильность структуры тестового сообщения и его своевременность, и на основании этого делается вывод о том, должен ли на него быть ответ, сообщение об ошибке или реализация должна его проигнорировать. При получении очередного сообщения клиента проверяется, является ли оно продолжением предыдущей сессии или это новый запрос на аутентификацию. В зависимости от этого выбирается соответствующий ответ сервера. В постусловии реакций данные проверяются на соответствие требованиям спецификации: допустимость сообщения и его структура.

5. Тестируемые реализации клиента протокола EAP

Используемые в нашем проекте реализации протокола отбирались исключительно по их доступности, популярности и использованием в предыдущих этапах проекта:

реализация клиента ОС CentOS 7 [12],

реализация клиента ОС Windows 10 [13].

6. Тестируемые методы EAP

При выборе методов мы также наблюдаем отличия в тестировании серверов и клиентов. Сервер по своему назначению должен поддерживать как можно большее количество существующих методов EAP. Набор методов поддерживаемых клиентом значительно скромнее. Поддержка конкретного метода клиентом зависит исключительно от взглядов разработчика, популярностью метода и его криптографическими возможностями. ОС

Windows 10 предлагает по-умолчанию туннельный метод PEAP собственной разработки. ОС CentOS предлагает более широкий выбор: MD5, PWD, TLS, FAST, TTLSv0, PEAP.

Исходя из наработок предыдущих этапов проекта, на данный момент мы используем методы EAP-MD5, TTLSv0 и PEAP [1],[14],[15].

7. Результаты тестирования

На текущем годовом этапе проекта в рамках технологии UniTESK выполнены следующие задачи:

- разработаны модели методов аутентификации EAP-MD5, TTLSv0, PEAP для клиентов, которые интегрированы в разработанную ранее модель базового протокола EAP,
- разработана спецификация и медиаторы для указанных методов,
- разработан набор тестов, покрывающий часть требований спецификаций.

Найдены несколько отклонений реализаций от спецификаций.

CentOS 7, метод TTLSv0:

- в первом сообщении от сервера tls-start игнорируется бит S,
- в сообщении tls-start игнорируется бит M,
- в сообщении tls-start игнорируется номер версии,
- в сообщении tls-start игнорируется наличие дополнительных tls-данных (данное сообщение не должно содержать каких-либо данных),
- бит M в сообщениях указывает на наличие дополнительных tls-фрагментов. во втором сообщении от сервера tls-ServerHello реализация собирает фрагменты, ориентируясь только на соответствующие поля длины, игнорируя бит M.

Windows 10, метод TTLSv0:

- в сообщении tls-start игнорируется номер версии,
- в сообщении tls-start игнорируется наличие дополнительных tls-данных.

Windows 10, метод PEAP:

- в сообщении tls-start игнорируется номер версии,
- в сообщении tls-start внутренние поля длины не проверяются,
- в сообщении tls-start игнорируется наличие дополнительных tls-данных.

8. Заключение

В данной работе представлен опыт верификации клиентских реализаций методов протокола аутентификации EAP, который является логическим продолжением проекта по верификации этого протокола безопасности. В работе используются как простые, так и туннельные методы, предполагающие создание защищенного туннеля, внутри которого применяются другие методы аутентификации.

В работе использовался новый тестовый набор, разработанный с использованием технологии UniTESK и наработок коллектива в тестировании сетевых протоколов. Технология UniTESK позволяет автоматизировать процесс верификации сетевых протоколов на основе их формальных моделей, методы мутационного тестирования позволяют протестировать устойчивость реализации протокола к искаженным сообщениям.

Представленный подход доказал свою эффективность в наших предыдущих проектах, обеспечив обнаружение различных отклонений от спецификации и других ошибок при тестировании сетевых протоколов [18,19].

Данная работа является частью проекта РФФИ № 16-07-00603 «Верификация функций безопасности и оценка устойчивости к атакам реализаций протокола аутентификации EAP».

Литература

1. Aboba B. et al. Extensible Authentication Protocol (EAP). June 2004. IETF RFC 3748. — URL: https://tools.ietf.org/html/rfc3748 .
2. Никешин А.В., Шнитман В.З. Обзор расширяемого протокола аутентификации и его методов // Труды ИСП РАН, том 30, вып. 2, 2018 г. — С. 113-148. — doi:10.15514/ISPRAS-2018-30(2)-7 .
3. Никешин А.В., Пакулин Н.В., Шнитман В.З. Подходы к разработке тестового набора для тестирования реализаций протокола EAP и его методов // Научный сервис в сети Интернет: труды XVIII Всероссийской научной конференции (19–24 сентября 2016 г., г. Новороссийск). — М.: ИПМ им. М.В. Келдыша, 2016. — С. 290-297. — doi:10.20948/abrau-2016-24.
4. Никешин А.В., Шнитман В.З. Верификация протокола EAP и его методов в беспроводных сетях // Научный сервис в сети Интернет: труды XIX Всероссийской научной конференции (18-23 сентября 2017 г., г. Новороссийск). — М.: ИПМ им. М.В.Келдыша, 2017. — С. 369-376. — doi:10.20948/abrau-2017-43 .
5. Никешин А.В., Шнитман В.З. Верификация туннельных методов протокола аутентификации EAP // Научный сервис в сети Интернет: труды XX Всероссийской научной конференции (17-22 сентября 2018 г., г. Новороссийск). — М.: ИПМ им. М.В.Келдыша, 2018. — С. 406-416. — doi:10.20948/abrau-2018-17 .
6. Никешин А.В., Шнитман В.З. Тестирование соответствия реализаций протокола EAP и его методов спецификациям Интернета // Труды ИСП

РАН, том 30, вып. 6, 2018. — С. 89-104. — doi:10.15514/ISPRAS-2018-30(6)-5 .
7. IEEE Standard 802.1X-2010 - IEEE Standard for Local and metropolitan area networks – Port-Based Network Access Control, 2010 .
8. C. Rigney, S. Willens, A. Rubens and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", June 2000 .
9. B. Aboba and P. Calhoun. RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP). September 2003. URL: https://tools.ietf.org/html/rfc3579 .
10. Bourdonov I., Kossatchev A., Kuliamin V., and Petrenko A. UniTesK Test Suite Architecture // Proceedings of FME 2002. LNCS 2391. — P. 77–88, Springer-Verlag, 2002 .
11. JavaTESK. — URL: http://www.unitesk.ru/content/category/5/25/60/ .
12. CentOS 7. — URL: https://www.centos.org/ .
13. Windows Server 2012 R2. — URL: https://www.microsoft.com .
14. Funk & Blake-Wilson. Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0). August 2008. IETF RFC 5281. — URL: https://tools.ietf.org/html/rfc5281 .
15. Microsoft Corporation. [MS-PEAP]: Protected Extensible Authentication Protocol (PEAP). December 2017. — URL: https://msdn.microsoft.com/en-us/library/cc238354.aspx, 25.04.2018 .

References

1. Aboba B. et al. Extensible Authentication Protocol (EAP). June 2004. IETF RFC 3748. — URL: https://tools.ietf.org/html/rfc3748 .
2. Nikeshin A.V., Shnitman V.Z. The review of Extensible Authentication Protocol and its methods. Trudy ISP RAN/Proc. ISP RAS, vol. 30, issue. 2, 2018. — P. 113-148 (inRussian). — doi:10.15514/ISPRAS-2018-30(2)-7 .
3. Nikeshin A.V., Pakulin N.V., Shnitman V.Z. Approaches to the development of a test suite for testing implementations of EAP and its methods // Nauchnyi servis v seti Internet: trudy XVIII Vserossiiskoi nauchnoi konferentsii (19-24 sentiabria 2016 g., g. Novorossiisk). — M.: IPM im. M.V.Keldysha, 2016. — P. 290-297. — doi:10.20948/abrau-2016-24 .
4. Nikeshin A.V., Shnitman V.Z. Verification of EAP and its methods in wireless networks // Nauchnyi servis v seti Internet: trudy XIX Vserossiiskoi nauchnoi konferentsii (18-23 sentiabria 2017 g., g. Novorossiisk). — M.: IPM im. M.V.Keldysha, 2017. — P. 369-376. — doi:10.20948/abrau-2017-43 .
5. Nikeshin, A.V., Shnitman, V.Z. The verification of tunnel methods of the Extensible Authentication Protocol (EAP) // (2018) CEUR Workshop Proceedings, 2260. — P. 406-416.

6. Nikeshin A.V., Shnitman V.Z. Conformance testing of Extensible Authentication Protocol implementations // Trudy ISP RAN/Proc. ISP RAS, vol. 30, issue 6, 2018. — P. 89-104 (in Russian). — doi:10.15514/ISPRAS-2018-30(6)-5 .
7. IEEE Standard 802.1X-2010 - IEEE Standard for Local and metropolitan area networks – Port-Based Network Access Control, 2010 .
8. C. Rigney, S. Willens, A. Rubens and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)", June 2000 .
9. B. Aboba and P. Calhoun. RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP). September 2003. — URL: https://tools.ietf.org/html/rfc3579 .
10. Bourdonov I., Kossatchev A., Kuliamin V., and Petrenko A. UniTesK Test Suite Architecture // Proceedings of FME 2002. LNCS 2391. — P. 77–88, Springer-Verlag, 2002 .
11. JavaTESK. — URL: http://www.unitesk.ru/content/category/5/25/60/ .
12. CentOS 7. — URL: https://www.centos.org/ .
13. Windows Server 2012 R2. — URL: https://www.microsoft.com .
14. Funk & Blake-Wilson. Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0). August 2008. IETF RFC 5281. — URL: https://tools.ietf.org/html/rfc5281 .
15. Microsoft Corporation. [MS-PEAP]: Protected Extensible Authentication Protocol (PEAP). December 2017. — URL: https://msdn.microsoft.com/en-us/library/cc238354.aspx, 25.04.2018 .