УНИВЕРСИТЕТ ИТМО
НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
январь–февраль 2019 Том 19 № 1 ISSN 2226-1494 http://ntv.ifmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
January–February 2019 Vol. 19 No 1 ISSN 2226-1494 http://ntv.ifmo.ru/en
НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ

УДК 004.773

ПРОТОКОЛ ПЕРЕДАЧИ ДАННЫХ MQTT В МОДЕЛИ УДАЛЕННОГО УПРАВЛЕНИЯ ПРАВАМИ ДОСТУПА ДЛЯ СЕТЕЙ ИНТЕРНЕТА

Д.И. Дикий[a], В.Д. Артемьева[b]

[a] Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация
[b] Балтийский федеральный университет имени И. Канта, Калининград, 236016, Российская Федерация
Адрес для переписки: dimandikiy@mail.ru

Информация о статье
Поступила в редакцию 22.10.18, принята к печати 30.11.18
doi: 10.17586/2226-1494-2019-19-1-109-117
Язык статьи – русский

Ссылка для цитирования: Дикий Д.И., Артемьева В.Д. Протокол передачи данных MQTT в модели удаленного управления правами доступа для сетей Интернета // Научно-технический вестник информационных технологий, механики и оптики. 2019. Т. 19. № 1. С. 109–117. doi: 10.17586/2226-1494-2019-19-1-109-117

Аннотация
Рассмотрены вопросы безопасности в интернете вещей, а именно организация безопасного разграничения доступа при использовании протокола MQTT. Проанализированы методы и механизмы безопасности, реализованные или поддерживаемые MQTT. Так, протокол реализует аутентификацию по логину и паролю, а также поддерживает криптографические преобразования над передаваемой информацией по протоколу TLS. Для аутентификации могут быть использованы сторонние сервисы по протоколу OAuth. Авторизация происходит путем настройки ACL-файлов или через сторонние сервисы и базы данных. Предложена модель управления дискреционным разграничением доступа устройств для межмашинного взаимодействия по протоколу MQTT, которая основана на модели Харрисона–Руззо–Ульмана. Модель предусматривает шесть операторов: добавление и удаление субъекта, добавление и удаление объекта, добавление и удаление прав доступа. Модель разграничения прав доступа имеет вид матрицы доступа и содержит три вида прав: чтение, запись и владение. Модель реализована таким образом, чтобы быть совместимой с протоколом широко распространенной версии v3.1.1. Изменение прав доступа происходит с помощью доступных в протоколе MQTT видов сообщений. Рассмотрен алгоритм такого построения служебного блока данных, чтобы этот блок можно было легко распознать в теле сообщения. Используя предлагаемую модель, можно минимизировать участие администратора за счет того, что устройства сами будут определять права доступа к информационным ресурсам без участия человека. Приведены рекомендации по политике безопасности при организации информационного обмена в соответствии с протоколом MQTT.

Ключевые слова
интернет вещей, передача данных, модель разграничения доступа, безопасность, MQTT, матрица доступа

MQTT DATA PROTOCOL IN REMOTE ACCESS CONTROL MANAGEMENT MODEL FOR INTERNET NETWORKS

D.I. Dikii[a], V.D. Artemeva[b]

[a]ITMO University, Saint Petersburg, 197101, Russian Federation
[b]Immanuel Kant Baltic Federal University, Kaliningrad, 236016, Russian Federation
Corresponding author: dimandikiy@mail.ru

Article info
Received 22.10.18, accepted 30.11.18
doi: 10.17586/2226-1494-2019-19-1-109-117
Article in Russian

For citation: Dikii D.I., Artemeva V.D. MQTT data protocol in remote access control management model for Internet networks. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2019, vol. 19, no. 1, pp. 109–117 (in Russian). doi: 10.17586/2226-1494-2019-19-1-109-117

Abstract
The paper deals with security issues in the environment of "Internet of things" and, in particular, the management of safety access control at MQTT protocol application. We analyzed the most widespread data transfer protocols, CoAP and MQTT, and carried out the analysis of safety methods and means for the MQTT protocol being realized in it or maintained by it. The protocol implements authentication by login and password and allows for cryptographic transformations over the transmitted information via TLS protocol. Third-party services via OAuth protocol and others can be applied for authentication. The

authentication takes place by the setting of ACL files or the third-party services and databases. A model is proposed for remote access control management of devices for machine-to-machine interaction under the MQTT protocol based on the Harrison-Ruzzo-Ullman model. The model provides six operators: addition and removal of the subject, addition and removal of the object, addition and deletion of access rights. The proposed model has the form of an access matrix and includes three types of rights: reading, writing and holding. The model is implemented with the result that it is compatible with the version v3.1 of MQTT protocol widely used at the moment. The change of access rights is performed on the basis of the types of messages available in MQTT protocol. An algorithm is considered for service data block creation so that this block can be easily recognized in the message body. The proposed model application gives the possibility to minimize administrator's participation by determination of access rights via the devices themselves without human involvement. Recommendations are given for security policy during information traffic management under MQTT protocol.

Keywords
Internet of things, communication, access control model, security, MQTT, access matrix

Введение

Согласно прогнозам [1], к 2020 году количество устройств, используемых в рамках интернета вещей, превысит 25 млрд. Для обеспечения беспрерывной работы всей сети устройств, с учетом их ограниченных вычислительных и энергетических мощностей, разрабатываются «легкие» протоколы передачи данных, охватывающие весь стек OSI[1]. Так, на канальном уровне используется протокол Lora [2], основанный на PHY IEEE 802.15.4 [3], в отличие от обычного проводного соединения по стандарту IEEE 802.3 (Ethernet) либо беспроводного соединения: IEEE 802.11 (Wi-Fi), IEEE 802.15.1 (Bluetooth). На более высоком уровне разрабатываются протоколы, работающие выше транспортного уровня, а именно поверх TCP и UDP протоколов. К таким протоколам можно отнести CoAP (Constrained Application Protocol) [4, 5] и MQTT (Message Queue Telemetry Transport) [6] протоколы. Первый работает поверх UDP по принципу клиент-сервер. Протокол MQTT, наоборот, работает поверх TCP/IP и имеет структуру «издатель-подписчик». Пример организации обмена сообщениями по протоколу MQTT представлен на рис. 1.

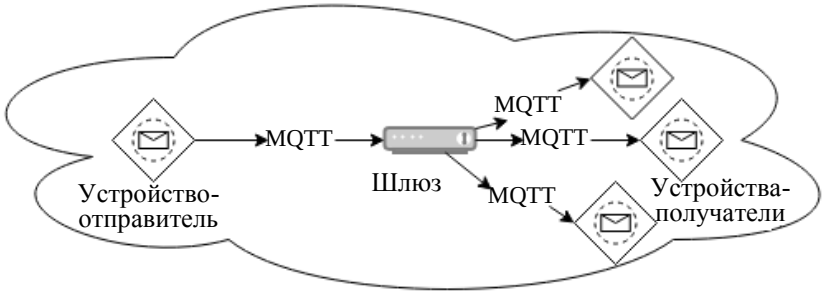

Рис. 1. Процесс обмена сообщениями по протоколу MQTT

Сообщение, отправленное на шлюз, перенаправляется получателю или группе получателей. Маршрутизацию сообщений обеспечивает устройство-шлюз. Таким образом, данный протокол позволяет снизить энергопотребление при передаче сообщения группе других устройств.

Таблица. Виды сообщений в протоколе MQTT

Вид сообщения	Описание	Направление передачи
CONNECT	Подключение к шлюзу	Клиент–Шлюз
CONNACK	Подтверждение подключения	Шлюз–Клиент
PUBLISH	Публикация сообщения с темой	Клиент–Шлюз
PUBACK	Подтверждение публикации сообщения	Клиент–Шлюз, Шлюз–Клиент
PUBREC	Публикация получена	Клиент–Шлюз, Шлюз–Клиент
PUBREL	Удаление сообщения	Клиент–Шлюз, Шлюз–Клиент
PUBCOMP	Публикация окончена	Клиент–Шлюз, Шлюз–Клиент
SUBSCRIBE	Запрос на подписку темы	Клиент–Шлюз
SUBACK	Ответ об успешной подписке на тему	Шлюз–Клиент
UNSUBSCRIBE	Запрос на отписку от темы	Клиент–Шлюз
UNSUBACK	Ответ об успешной отписке от темы	Шлюз–Клиент
PINGREQ	Запрос проверки соединения	Клиент–Шлюз
PINGRESP	Ответ проверки соединения	Шлюз–Клиент
DISCONNECT	Отключение от шлюза	Клиент–Шлюз

Разработанный для межмашинного взаимодействия (M2M) [7] протокол MQTT основан на нескольких типах сообщений (см. таблицу). Согласно [8, 9], в протоколе широко используются две меры

[1]ГОСТ Р ИСО/МЭК 7498-1-99 Информационная технология (ИТ). Взаимосвязь открытых систем. Базовая эталонная модель. Часть 1. Базовая модель.

защиты информации: авторизация устройства по логину и паролю и протокол TLS для криптографической защиты передаваемой информации. Так, при подключении к шлюзу connect сообщение содержит логин и пароль помимо служебной информации. После успешной проверки соединение между шлюзом и устройством поддерживается, пока оно не будет закрыто. Сообщение publish содержит служебные заголовки, полезную нагрузку и название «темы». Чтобы получить это сообщение, другое устройство должно авторизоваться на шлюзе (отправить сообщение connect) и подписаться на тему (отправить сообщение subscribe с названием темы).

Модель

Из документации протокола версии v.3.1.1 [6] MQTT следует, что этот протокол предназначен для межмашинного взаимодействия. Однако безопасности обмена данными уделено мало внимания. Так, один пользователь, имеющий уникальную пару «логин–пароль», может иметь неограниченное число устройств, каждое из которых имеет уникальный в данной сети идентификатор *clientId*. Таким образом, для каждого устройства одного пользователя пара «логин–пароль» одинакова. В случае компрометации, например, при соединении без криптографической защиты по протоколу TLS или успешной атаки на перебор паролей, злоумышленник сможет «прослушивать» всю информацию, циркулирующую в сети путем подписок на все возможные темы (рис. 2).

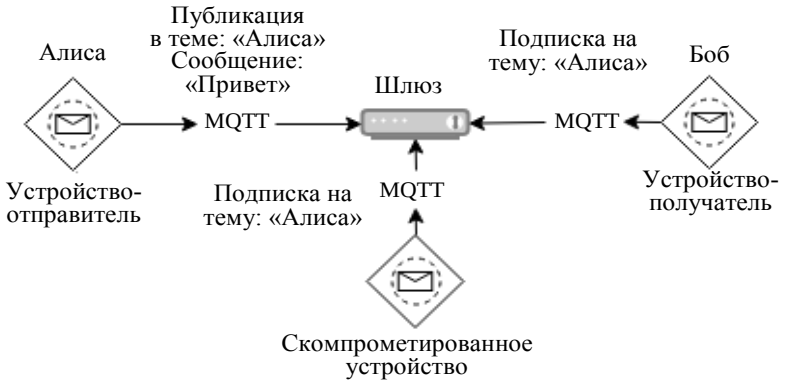

Рис. 2. Схема утечки информации при использовании протокола MQTT

Пусть Алиса и Боб – авторизованные пользователи. Устройство Алисы хочет отправить сообщение Бобу. Для этого создается тема «Алиса». Боб подписывается на тему «Алиса», отправляя соответствующее subscribe-сообщение. Алиса отправляет publish-сообщение с темой «Алиса» и полезной информацией «*привет*». Сообщение попадает на шлюз, где проверяется список всех подписчиков, и рассылается соответствующим клиентам, в данном случае – Бобу. Но если скомпрометированное устройство отправит такое же subscribe-сообщение на тему «Алиса», то оно тоже получит сообщение «*привет*» от Алисы.

Протокол MQTT поддерживает групповые широковещательные рассылки. Для этого используются спецсимволы «/», «+», «#» [10]. Наименование темы представляет собой многоуровневый список, например, тема «*home/room/temperature*» имеет три уровня, каждый из которых разделен символом «/». Пусть существует три темы: «*home/kitchen/temperature*», «*home/room/temperature*», «*home/room/pressure*». Спецсимвол «+» позволяет подписаться на все темы одного уровня, например, subscribe-сообщение с темой «*home/+/temperature*» произведет подписку на все темы, соответствующие регулярному выражению: «*home/kitchen/temperature*», «*home/room/temperature*» и т.д. Спецсимвол «#» позволяет подписаться на все темы ниже по уровню. Например, subscribe-сообщение с темой «*home/#*» оформит подписку на темы «*home/kitchen/temperature*», «*home/room/temperature*» и «*home/room/pressure*». Таким образом, чтобы получить всю информацию, циркулирующую в сети, злоумышленнику достаточно отправить subscribe-сообщение с темой «#», чтобы подписаться на все существующие темы. Во избежание полной утечки информации при компрометации одной учетной записи предложена следующая модель разграничения доступа к темам.

Чтобы нивелировать вероятность утечки информации используются дополнительные плагины поверх протокола MQTT, которые формируют ACL (Access Control List) файлы или записи в базах данных. Так, брокер Mosquitto [11] использует отдельный файл для хранения прав пользователей по полю username или клиентов по полю *clientId* в виде строк «*user <username> topic <readwrite||read||write> <topic name>*», представлено три вида прав: чтение, запись, чтение и запись. Аналогично организовано разграничение прав при использовании баз данных. Основным недостатком данного подхода является отсутствие у пользователя возможности самому задавать права доступа к создаваемым им же темам. В Mosquitto [11] для назначения прав необходимо обращаться к администратору, что очень неудобно при

эксплуатации многопользовательских систем. По такому же принципу работают многие другие проекты, например, проект [12] использует web-интерфейс для задания прав пользователей, как правило, через HTTP-протокол.

В работе [13] предложено управление доступом с помощью протокола OAuth, что подразумевает использование сторонних сервисов для авторизации пользователей. Следовательно, функционирование устройств зависит от работоспособности других сервисов. Согласно работе [8], для идентификации и авторизации, помимо протоколов OAuth, также используется протокол LDAP [14].

Основной целью разработки другого подхода к модели управления ограничением доступа для протокола MQTT является поддержка возможности пользователям самим устанавливать права на доступ к темам (введение права на владение) с помощью самих устройств интернета вещей, без использования сторонних каналов связи и сервисов. Это позволит организовать публичные MQTT-шлюзы. Предлагаемая модель основана на матричной [15] (дискреционной) модели доступа Харрисона–Руззо–Ульмана.

Пусть существуют два множества: $C = \{C_1,\ldots, C_n\}$ линейно упорядоченное множество пользователей сети, имеющих уникальный логин (субъектов), и $O = \{O_1,\ldots, O_k\}$ – множество названий тем (объекты доступа), а $R = \{o, w, r\}$ – ограниченное множество прав доступа (o – владение, w – запись, r – чтение), политика разграничения доступа будет описываться матрицей доступа **M**, каждая ячейка матрицы $\mathbf{M}=(m_{n,k})$ размером $N\times K$, где $n=1\ldots N$, $k=1\ldots K$, $m_{n,k}\in R$ содержит набор прав для субъекта из множества $C = \{C_1,\ldots, C_n\}$ к объекту из множества $O = \{O_1,\ldots, O_k\}$. Тогда для каждого момента времени система Q будет описываться состоянием $Q = (C,O,\mathbf{M})$ [16].

Если $c`\notin C$, процесс добавления нового пользователя описывается выражением:

$$\begin{aligned}&C`=C\cup\{c`\},\\&O`=O,\\&\text{если } (c\quad o)\in C\times O, \text{ то } M`[c\quad o]=M[c\quad o],\\&\text{если } o\in O`, \text{ то } M`[c`\quad o]=\varnothing\end{aligned}\tag{1}$$

Процесс добавления новой темы описывается выражением, если $o`\notin O$, $s\in C$:

$$\begin{aligned}&O`=O\cup\{o`\},\\&C`=C,\\&\text{если } (c\quad o)\in C\times O, \text{ то } M`[c\quad o]=M[c\quad o],\\&\text{если } c\in C`, \text{ то } M`[c`\quad o]=\varnothing,\\&\text{если } c=s, \text{ то } M`[c`\quad o]=r\{o,w,r\}\end{aligned}\tag{2}$$

Процесс удаления пользователя описывается выражением, если $c`\in C$:

$$\begin{aligned}&C`=C\setminus\{c`\},\\&O`=O,\\&\text{если } (c\quad o)\in C`\times O`, \text{ то } M`[c\quad o]=M[c\quad o].\end{aligned}\tag{3}$$

Процесс удаления темы описывается выражением, если $o`\in O$:

$$\begin{aligned}&\text{если } M[s\quad o]=r\{o\}, \text{ то}\\&O`=O\setminus\{o`\},\\&C`=C,\\&\text{если } (c\quad o)\in C`\times O`, \text{ то } M`[c\quad o]=M[c\quad o].\end{aligned}\tag{4}$$

Процесс добавления права описывается выражением, если $o`\in O$, $c`\in C$, $s\in C$:

$$\begin{aligned}&\text{если } M[s\quad o]=r\{o\}, \text{ то}\\&C`=C,\\&O`=O,\\&M`[c\quad o]=M[c\quad o]\cup r\{\ \},\\&\text{если } (c`\quad o`)\neq(c\quad o), \text{ то } M`[c`\quad o`]=M[c`\quad o`]\end{aligned}\tag{5}$$

Процесс удаления права описывается выражением, если $o`\in O$, $c`\in C$, $s\in C$:

$$\begin{aligned}&\text{если } M[s\quad o]=r\{o\}, \text{ то}\\&C`=C,\\&O`=O, M`[c\quad o]=M[c\quad o]\setminus r\{\ \},\\&\text{если } (c`\quad o`)\neq(c\quad o), \text{ то } M`[c`\quad o`]=M[c`\quad o`]\end{aligned}\tag{6}$$

где s – субъект (пользователь), который инициализирует транзакцию; $r\{\}$ – какое-либо право из множества R $\{o,w,r\}$, где w – запись, r – чтение, o – владение, что подразумевает возможность назначать права на данную тему для других пользователей. Например, $r\{o\}$ – означает право владения.

Согласно формуле (1) создается новый пользователь путем создания учетной записи на шлюзе MQTT средствами администрирования. Права других пользователей не изменяются, новый пользователь не имеет прав.

Согласно формуле (2) создается новая тема. Здесь учитывается, что если тема новая для системы, или у нее отсутствует владелец, то пользователю (s), вызвавшему инициализацию этой процедуры, предоставляются права владения над этой темой, записи и чтения. В рамках протокола MQTT при отправке сообщения «publish» с новой для шлюза темой за отправителем остается право владения этой темой. Отправитель может задать перечень пользователей, которым будут доступны чтение и запись, а также передать право владения. Для задания прав для новой темы не потребуется обращаться к администратору шлюза.

По формуле (3) удаляется пользователь, который до этого момента существовал в системе. Права других пользователей не изменяются, пользователь лишается всех прав. Удаление пользователя происходит путем удаления учетной записи на шлюзе MQTT средствами администрирования.

Согласно формуле (4) удаляется тема. В рамках протокола MQTT для удаления темы со шлюза необходимо отправить publish-сообщение с соответствующим содержанием. При этом от имени всех пользователей, подписанных на удаляемую тему, отправляется unsubscribe-сообщение. Это необходимо для того, чтобы обновить список тем на шлюзе, на которые подписано устройство, и избежать утечки информации.

В формулах (5) и (6) добавляются и удаляются права пользователей. Чтобы пользователь мог внести изменения в права доступа к теме, он должен обладать правом владения $r\{o\}$. Если право владения присутствует, то права изменяются только у перечисленных в дополнительном заголовке пользователей. Если изымается право чтения, то необходимо от имени пользователя, которого лишают права, отправить unsubscribe-сообщение, как и при удалении темы. Изменение прав происходит удаленно самими пользователями. Взаимодействие с администратором не потребуется.

Встраивание в протокол MQTT

Для применения вышеизложенной модели в рамках протокола MQTT необходимо рассмотреть структуру заголовков протокола. Протокол MQTT состоит из фиксированного заголовка, переменного заголовка и полезной нагрузки. Фиксированный заголовок имеет размер 2 Б и определяет тип сообщения (см. таблицу), флаг дубликата сообщения – 1 бит, флаг качества QoS (quality of service, качество обслуживания) – 2 бита, флаг сохранения сообщения на шлюзе – 1 бит. В переменном заголовке содержится 1 Б информации. Переменный заголовок зависит от типа сообщения. Таким образом, наиболее подходящим местом для встраивания информации в систему разграничения доступа является блок полезной нагрузки (тело сообщения).

Для того чтобы иметь возможность назначать права пользователям, в начало publish-сообщения встраивается дополнительный заголовок в следующем формате, как представлено на рис. 3.

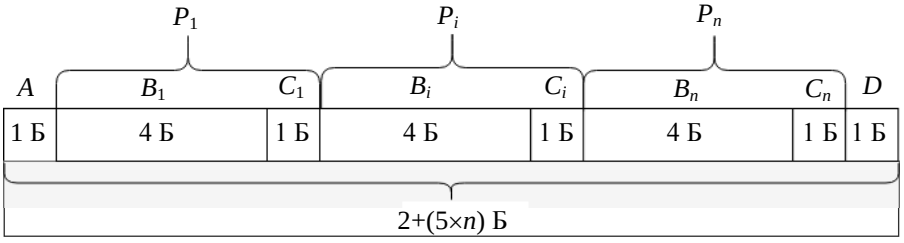

Рис. 3. Предлагаемый формат дополнительного заголовка, содержащего информацию о правах доступа к теме

На рис. 3 символом A обозначен обязательный байт дополнительного заголовка, определяющий наличие сведений о правах доступа в сообщении (значение 1) либо их отсутствие (значение 0). Если в заголовке указываются права доступа, то затем следуют n блоков P_i $(i=1\ldots n)$, соответствующих количеству пользователей, которым назначаются права. Блок P_i размером 5 Б состоит из двух частей: блок B размером 4 Б, идентифицирует субъект, и блок C размером 1 Б, в котором закодированы предоставляемые права (чтение, запись, владение). Размер B выбирается исходя из предполагаемого максимального количества пользователей. Блок B размером 4 Б позволяет закодировать более 4 млрд уникальных значений. Затем следует блок D размером 1 Б, обозначающий окончание дополнительного заголовка. Чтобы удалить все имеющиеся права доступа к теме, пользователь-владелец темы отправляет publish-сообщение, первый байт которого (блок A) должен принимать значение 2. Таким образом, для

сообщения, не содержащего информацию о назначении прав к теме, и сообщения удаления всех прав размер дополнительного заголовка составляет 1 Б. Для сообщений, содержащих права доступа, размер дополнительного заголовка V для предлагаемой конфигурации вычисляется по формуле:

$$V = 2 + 5n, \quad (7)$$

где n – число субъектов (пользователей), которым назначаются права. Если сообщение первое для этой темы, то права на владение, чтение и запись предоставляются отправителю автоматически, и дополнительный заголовок не используется.

Если не стоит задача экономии объема передаваемой информации, то для этих целей также можно использовать XML-формат. Пример использования XML-языка для передачи назначаемых прав доступа:

```
<users>
        <user>
                <username> username</username>
                <read>true</read>
                <write>true</write>
        </user>
<users/>
<payload>payload<payload>
```

При использовании спецсимволов «/», «+», «#» процесс управления правами доступа также соответствует предлагаемой модели. Для subscribe-сообщения с темой, содержащей спецсимволы, составляется регулярное выражение, и производится проверка на соответствие названия каждой темы этому регулярному выражению. Если совпадения найдены, то проверяется наличие прав на чтение каждой найденной темы. Таким образом, в случае компрометации учетной записи злоумышленник, отправляя subscribe-сообщение с темой «#», получит доступ только к тем темам, на которые у скомпрометированной учетной записи есть право чтения, а не ко всей информации в сети. Предлагаемая модель не нарушает принципов использования протокола MQTT для широковещательных рассылок и позволяет уменьшить вероятность утечки информации.

На рис. 4 представлена UML-схема подключения устройства, отправки и получения сообщений с учетом предлагаемой модели.

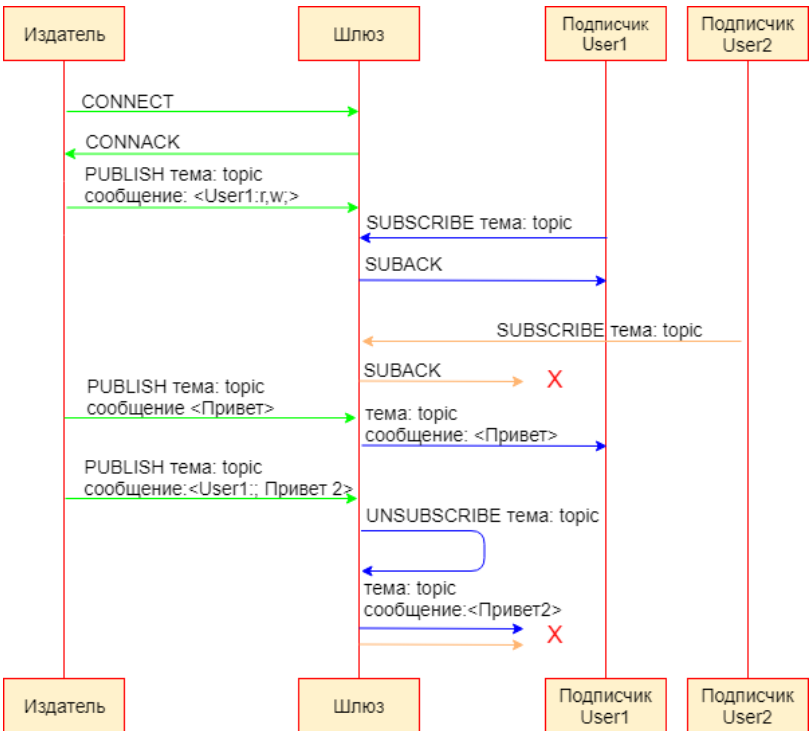

Рис. 4. UML-диаграмма сообщений при использовании MQTT с QoS 0

После отправки авторизованным пользователем сообщения «*publish topic <User1:r,w;>*» создается тема «*topic*», и назначаются права чтения и записи *r*{*r,w*} для пользователя *User1*. При последующей подписке на тему «*topic*» пользователь *User1* получит сообщение об успешной подписке и начнет получать сообщения, а также сможет отправлять их. В то же время пользователю *User2* будет отказано в получении и отправке сообщений для темы «*topic*». При отправке сообщения «*publish topic <User1:; Привет 2>*» пользователь *User1* лишается всех прав на тему, так как множество прав *r* определено как

пустое. Здесь *User1* и *User2* – идентификаторы пользователей, соответствующие параметру B_i на рис. 3.

Для того чтобы злоумышленник не мог создать множество тем с наиболее популярными названиями (например, «*home*»), в политике безопасности следует учесть, что каждый пользователь может создавать темы с названиями, первый уровень которых соответствует имени пользователя, например, «*username1/hom*e». Также с целью предотвращения утечки права владения в политике безопасности следует ограничить операцию передачи права владения.

Оценка производительности

Предлагаемая модель управления разграничением доступа реализована на базе открытого исходного кода брокера [17]. Данный брокер поддерживает управление доступом через локальный ACL-файл, который загружается в шлюз при его включении. Доработанный шлюз позволяет сохранять динамически права доступа в базе данных. В проекте использовалась СУБД Postgresql. Экспериментальная установка состояла из следующих компонентов: два клиента (персональные компьютеры – ПК), шлюз – Raspberry pi 3 model B, телекоммуникационное оборудование – маршрутизатор. База данных (БД) расположена на одном из двух ПК. Была оценена производительность шлюза в условиях, когда управление доступом:

1) отсутствует;
2) организуется через локальный ACL-файл;
3) организуется через БД по предлагаемой модели: сообщения без сведений об изменении прав доступа (отсутствует операция записи в БД) и сообщения с дополнительным заголовком, имеющим сведения об изменении прав доступа (присутствует операция записи в БД).

Для оценки производительности шлюза при лавинообразном отправлении 1000 сообщений за 250–400 мс (3 сообщения за 1 мс) измерялась разница между временем отправления и получения сообщений. На рис. 5 представлены результаты эксперимента.

Исходя из рис. 5 можно сделать вывод, что наибольшие временные затраты происходят при использовании БД для чтения и записи информации о правах доступа. При довольно небольших объемах передаваемой информации разница во времени незначительна. По мере возрастания количества сообщений за единицу времени разница становится очевидной. На доставку последнего сообщения с дополнительным заголовком, содержащим сведения об изменении прав доступа, затрачено более 1 с. Использование управления доступом через локальный ACL-файл позволило доставить сообщение чуть более чем за 0,5 с.

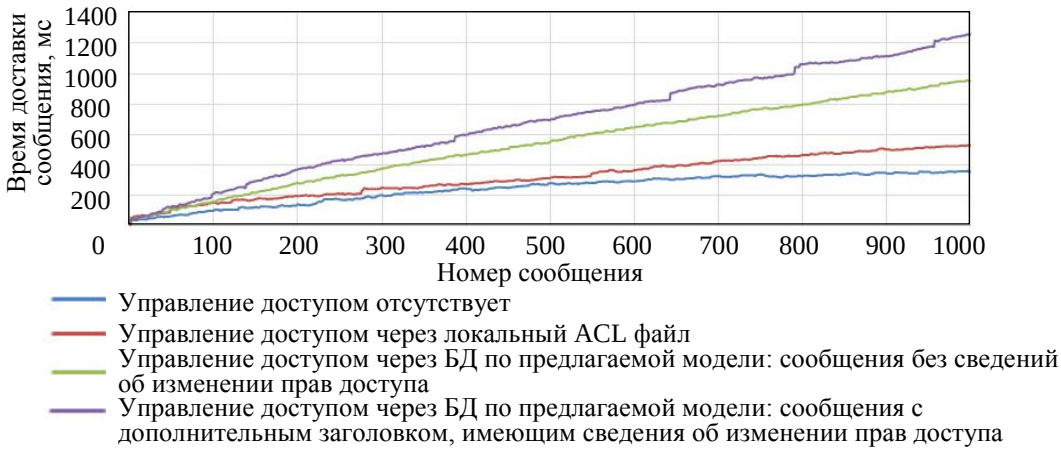

Рис. 5. Время доставки сообщений при различном подходе к управлению доступом

Заключение

В работе рассмотрены вопросы безопасности интернета вещей, проанализированы наиболее распространенные протоколы передачи данных. Проведен анализ методов и средств безопасности, используемых в протоколе MQTT. Анализ показал возможность утечки информации ввиду отсутствия какого-либо инструмента разграничения доступа в самом протоколе. Поэтому во многих проектах используются такие средства авторизации, как ACL-файлы и базы данных, протоколы LDAP и OAuth. С целью повысить безопасность и нивелировать угрозу утечки информации предложена модель управления правами доступа устройств при межмашинном взаимодействии по протоколу MQTT на базе модели Харрисона–Руззо–Ульмана. Предложенная модель учитывает особенности протокола MQTT и реализована на языке программирования Java на основе MQTT-брокера (шлюза) с открытым исходным кодом [17], что подтверждает практическую применимость данной модели. Программная реализация поддерживает возможность использования регулярных выражений при использовании MQTT-шлюза

таким образом, что функционирование брокера не нарушается при использовании дискреционной модели разграничения доступа. При использовании данной модели реализуется возможность дистанционного управления правами доступа самими пользователями без использования иных каналов связи, что способствует развитию публичных MQTT-шлюзов и снижению нагрузки на администраторов шлюза. Произведено тестирование на базе шлюза следующей конфигурации: аппаратно – микрокомпьютер Raspberry pi 3 model B, программно – доработанный MQTT-брокер с открытым исходным кодом. Моделировалось лавинообразное отправление и прием 1000 сообщений между двумя клиентами. Было показано, что при небольшом количестве сообщений за единицу времени ($1\ мс^{-1}$) шлюз примерно одинаково справляется с нагрузкой, как с предлагаемой моделью, так и без какого-либо управления доступом. Однако при увеличении объема обрабатываемых сообщений ($3\ мс^{-1}$) появляется значительная разница из-за постоянного обращения к базе данных. Предложены рекомендации для политики безопасности при использовании протокола MQTT.

Литература

1. Станки выходят на связь // Российская газета. 2016. № 7086. Режим доступа: https://rg.ru/2016/09/27/chislo-podkliuchennyh-k-seti-ustrojstv-k-2020-godu-dostignet-25-milliardov.html (дата обращения: 21.10.2018).
2. de Carvalho Silva J., Rodrigues J.J.P.C., Alberti A.M., Solic P., Aquino A.L.L. LoRaWAN – a low power WAN protocol for Internet of Things: a review and opportunities // Proc. 2nd Int. Multidisciplinary Conference on Computer and Energy Science. Split, Croatia, 2017.
3. Granjal J., Monteiro E., Sa Silva J. Security for the Internet of Things: a survey of existing protocols and open research issues // IEEE Communication Surveys and Tutorials. 2015. V. 17. N 3. P. 1294–1312. doi: 10.1109/comst.2015.2388550
4. Bormann C., Castellani A.P., Shelby Z. CoAP: an application protocol for billions of tiny Internet nodes // IEEE Internet Computing. 2012. V. 16. N 2. P. 62–67. doi: 10.1109/comst.2015.2388550
5. Гойхман В., Абраменкова Д. Протокол Интернета вещей CoAP // Технологии и средства связи. 2017. № 4. С. 20–24.
6. MQTT Version 3.1.1 OASIS Standard. 2014.
7. Pticek M., Cackovic V., Pavelic M., Kusek M., Jezic G. Architecture and functionality in M2M standards // Proc. 38th Int. Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO). Opatija, Croatia, 2015. doi: 10.1109/mipro.2015.7160306
8. Perrone G., Vecchio M., Pecori R., Giaffreda R. The day after mirai: a survey on MQTT security solutions after the largest cyber-attack carried out through an army of IoT devices // Proc. 2nd Int. Conf. on Internet of Things, Big Data and Security. 2017. P. 246–253. doi: 10.5220/0006287302460253
9. Fremantle P., Aziz B., Kopecky J., Scott P. Federated identity and access management for the Internet of Things // International Workshop on Secure Internet of Things. 2014. doi: 10.1109/siot.2014.8
10. Soni D., Makwana A. A survey on MQTT protocol for the Internet of Things // Proc. Int. Conf. on Telecommunication, Power Analysis and Computing Techniques. 2017.
11. Eclipse Mosquitto™. mosquitto.conf - the configuration file for Mosquitto. Режим доступа: https://mosquitto.org/man/mosquitto-conf-5.html (дата обращения: 21.10.2018).
12. Documentation. CloudMQTT. Режим доступа: https://www.cloudmqtt.com/docs.html (дата обращения: 21.10.2018).
13. Cruz-Piris L., Rivera D., Marsa-Maestre I., de la Hoz E., Velasco J.R. Access control mechanism for IoT environments based on modelling communication procedures as resources // Sensors. 2018. V. 18. N 3. P. 917. doi: 10.3390/s18030917
14. User Guide EMQ 2.2. Режим доступа: http://emqtt.io/docs/v2/guide.html (дата обращения: 21.10.2018).
15. Harrison M.A., Ruzzo W.L., Ullman J.D. Protection in operating systems // Communication of ACM. 1976. V. 19. N 8. P. 461–471. doi: 10.1145/360303.360333
16. Щеглов А.Ю. Модели, методы и средства контроля доступа к ресурсам вычислительных систем. СПб: Университет ИТМО, 2014. 95 с.
17. Moquette Java MQTT lightweight broker. Режим доступа: https://github.com/andsel/moquette (дата обращения: 21.10.2018).

References

1. Machines get in touch. *Rossiiskaya Gazeta*, 2016, no. 7086. Available at: https://rg.ru/2016/09/27/chislo-podkliuchennyh-k-seti-ustrojstv-k-2020-godu-dostignet-25-milliardov.html (accessed: 21.10.2018).
2. de Carvalho Silva J., Rodrigues J.J.P.C., Alberti A.M., Solic P., Aquino A.L.L. LoRaWAN – a low power WAN protocol for Internet of Things: a review and opportunities. *Proc. 2nd Int. Multidisciplinary Conference on Computer and Energy Science*. Split, Croatia, 2017.
3. Granjal J., Monteiro E., Sa Silva J. Security for the Internet of Things: a survey of existing protocols and open research issues. *IEEE Communication Surveys and Tutorials*, 2015, vol. 17, no. 3, pp. 1294–1312. doi: 10.1109/comst.2015.2388550
4. Bormann C., Castellani A.P., Shelby Z. CoAP: an application protocol for billions of tiny Internet nodes. *IEEE Internet Computing*, 2012, vol. 16, no. 2, pp. 62–67. doi: 10.1109/comst.2015.2388550
5. Goikhman V., Abramenkova D. CoAP Internet of Things protocol. *Communication Technologies and Equipment*, 2017, no. 4, pp. 20–24. (in Russian)
6. *MQTT Version 3.1.1* OASIS Standard. 2014.
7. Pticek M., Cackovic V., Pavelic M., Kusek M., Jezic G. Architecture and functionality in M2M standards. *Proc. 38th Int. Convention on Information and Communication Technology, Electronics and Microelectronics*. Opatija, Croatia, 2015. doi: 10.1109/mipro.2015.7160306
8. Perrone G., Vecchio M., Pecori R., Giaffreda R. The day after mirai: a survey on MQTT security solutions after the largest cyber-attack carried out through an army of IoT devices. *Proc. 2nd Int. Conf. on Internet of Things, Big Data and Security*, 2017, pp. 246–253. doi: 10.5220/0006287302460253
9. Fremantle P., Aziz B., Kopecky J., Scott P. Federated identity and access management for the Internet of Things. *International Workshop on Secure Internet of Things*, 2014. doi: 10.1109/siot.2014.8
10. Soni D., Makwana A. A survey on MQTT protocol for the Internet of Things. *Proc. Int. Conf. on Telecommunication, Power Analysis and Computing Techniques*, 2017.
11. *Eclipse Mosquitto™. mosquitto.conf - the configuration file for Mosquitto*. Available at: https://mosquitto.org/man/mosquitto-conf-5.html (accessed: 21.10.2018).
12. *Documentation. CloudMQTT*. Available at: https://www.cloudmqtt.com/docs.html (accessed: 21.10.2018).
13. Cruz-Piris L., Rivera D., Marsa-Maestre I., de la Hoz E., Velasco J.R. Access control mechanism for IoT environments based on modelling communication procedures as resources. *Sensors*, 2018, vol. 18, no. 3, p. 917. doi: 10.3390/s18030917
14. *User Guide EMQ 2.2*. Available at: http://emqtt.io/docs/v2/guide.html (accessed: 21.10.2018).
15. Harrison M.A., Ruzzo W.L., Ullman J.D. Protection in operating systems. *Communication of ACM*, 1976, vol. 19, no. 8, pp. 461–471. doi: 10.1145/360303.360333
16. Shcheglov A.Yu. *Models, Methods and Means for Access Control of Computing System Resources*. St. Petersburg, ITMO University Publ., 2014, 95 p. (in Russian)
17. *Moquette Java MQTT lightweight broker*. Available at: https://github.com/andsel/moquette (accessed: 21.10.2018).

Авторы

Дикий Дмитрий Игоревич – аспирант, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, Scopus ID: 56998707400, ORCID ID: 0000-0002-8819-8423, dimandikiy@mail.ru

Артемьева Виктория Денисовна – студент, Балтийский федеральный университет имени И. Канта, Калининград, 236016, Российская Федерация, ORCID ID: 0000-0001-6478-7162, vika_med2019@mail.ru

Authors

Dmitrii I. Dikii – postgraduate, ITMO University, Saint Petersburg, 197101, Russian Federation, Scopus ID: 56998707400, ORCID ID: 0000-0002-8819-8423, dimandikiy@mail.ru

Viktoriia D. Artemeva – student, Immanuel Kant Baltic Federal University, Kaliningrad, 236016, Russian Federation, ORCID ID: 0000-0001-6478-7162, vika_med2019@mail.ru