# РАЗРАБОТКА КОНЦЕПЦИИ ЗАЩИЩЕННОГО ЦЕНТРАЛИЗОВАННОГО ВЗАИМОДЕЙСТВИЯ РАСПРЕДЕЛЕННЫХ УСТРОЙСТВ

***М.М. Ковцур,*** *к.т.н., доцент, Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, maxkovzur@mail.ru;*
***А.А. Браницкий,*** *к.т.н., Санкт-Петербургский Федеральный исследовательский центр Российской академии наук, branitskiy@comsec.spb.ru;*
***Н.И. Казаков,*** *Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, kazakov.ni2.18@gmail.com.*

***УДК 004.057.4***

***Аннотация.*** С растущим количеством веб-сервисов возникают вопросы обеспечения безопасности передачи данных между клиентом и сервером. В данной статье рассматривается организация защищенной передачи данных в сети, состоящей из сенсоров и центрального сервера под управлением одного лица. Рассмотрены механизмы защиты с сессионного уровня и выше.

***Ключевые слова*:** централизованная сеть; *REST API*; веб-сертификаты; *JWT*.

# DEVELOPMENT OF A CONCEPT OF CENTRLALIZED NETWORKING OF DISTRIBUTED DEVICES

***Maxim Kovtsur,*** *Ph.D., Associate Professor, St. Petersburg state university of telecommunications n/a prof. M.A. Bonch-Bruevich;*
***Aleksandr Branitskiy,*** *Ph.D., St. Petersburg Federal Research Centerof the Russian Academy of Sciences (SPC RAS);*
***Nikita Kazakov,*** *St. Petersburg state university of telecommunications n/a prof. M.A. Bonch-Bruevich.*

***Annotation.*** With the growing number of web services, there are questions about the security of data transmission between the client and the server. This article discusses the organization of secure data transmission in a network consisting of sensors and a central server managed by one person. Protection mechanisms from the session level and above are considered.

***Keywords:*** centralized network*; REST API;* web-certificates; *JWT*.

## Введение

В современном мире растет количество больших распределенных систем, применяемых, в частности, в Интернете вещей и облачных сервисах [1-5]. Это порождает вопрос обеспечения безопасной передачи данных в сети таких систем [6,7]. Подавляющее большинство систем используют клиент-серверную архитектуру, в которой множество клиентов отправляют запросы к центральному серверу, на котором работает определенное приложение. Помимо этого, распространение получили сети сенсоров, также состоящие из множества микрокомпьютеров, которые собирают, обрабатывают и передают данные центральному серверу, который осуществляет дополнительную обработку, агрегацию и их хранение. Такие сети могут отслеживать физические характеристики среды (освещенность, влажность, температура), уровень излучения на различных частотах, а также информацию о беспроводных сетях. Ее в дальнейшем можно использовать для улучшения характеристик беспроводной сети

(исключение «слепых зон»), обнаружения вторжений и контроля местоположения легинтимных клиентов. В данной работе рассматривается подход к организации передачи данных приложений высокого уровня в сети сенсоров с центральным сервером.

В существующих исследованиях [8] организация предлагаемой системы описана не полно. Существуют работы, исследующие отдельные механизмы защиты на разных сетевых уровнях [9-11] и приводящие их сравнение [12, 13], а также предлагающие новые механизмы [14-16]. Однако они не предоставляют полного решения по обеспечению защищенного взаимодействия.

Подавляющая масса приложений высокого уровня используется для передачи данных *HTTP* [8]. Приложение исследуемой сети построено с использованием концепции *REST API*. *REST* (*Representational State Transfer*) – это концепция построения сетевого взаимодействия сервисных компонентов. Ее основными принципами является отсутствие состояния, модель клиент-сервер и единообразие интерфейса как модели взаимодействия. Передача данных в *REST* осуществляется с помощью *HyperText Transfer Protocol* (*HTTP*). Однако, так как сам по себе *HTTP* никак не защищен, вместо него рекомендуется применять *HTTPS* (*HTTP Secure*). Он добавляет поддержку шифрования и обеспечивает защиту от атак прослушивания сетевого трафика.

Так как протокол *HTTPS* сам по себе подвержен атакам *Man in the Middle* (*MITM*), для организации безопасной передачи данных от сенсоров к серверу и минимизации рисков внешнего вмешательства в сеть предлагается использовать следующий комплекс мер:

1) *HTTPS*;
2) двухсторонняя аутентификация подключения по сертификатам;
3) токены с возможностью отзыва.

*HTTPS* – протокол передачи гипертекста поверх криптографических протоколов *SSL/TLS*. Он обеспечивает достаточную защиту передаваемой информации при условии корректной авторизации и проверки сертификатов. Так как все сертификаты и серверы, и сенсоры в исследуемой системе будут контролироваться одним лицом, данные уязвимости можно считать неактуальными.

Авторизация сервера и сенсоров осуществляется с помощью сертификатов, выпущенных одним Центром Сертификации (ЦС) или цепочкой доверенных ЦС. В первую очередь, создается корневой сертификат организации. Затем с помощью него по цепочке выпускаются сертификаты доверенных серверов, к которым уже обращаются клиенты. Сертификаты клиентов так же должны быть выпущены корневым ЦС или доверенными ЦС.

С точки зрения сенсора достоверность сервера подтверждается доверенностью его сертификата – он должен быть выпущен тем же ЦС, что и сертификат сенсора. Для аутентификации сенсора с точки зрения сервера используется аналогичная проверка – сенсор при подключении предоставляет свою цепочку сертификатов, а сервер проверяет корректность цепочки и то, что сертификат выдан доверенным ЦС [17]. Таким образом реализуется двухсторонняя аутентификация.

Для контроля сессии после аутентификации и оптимизации повторных подключений предлагается использовать токены. Одним из наиболее распространенных видов токенов являются *JSON Web Token* (*JWT*).

*JWT* – это набор информации в формате *JSON*, который опционально может быть зашифрован и подписан. Он считается безопасным способом передачи данных между двумя участниками.

Для создания *JWT* токена используются:

- заголовок (*header*), содержащий общую информацию о токене;
- полезные данные (*payload*), которые включают в себя информацию о пользователе и его авторизационные данные;
- подпись (*signature*).

Все элементы записаны в формате *JSON*, закодированы в *base*64 и объединены через точку. Пример готового токена:

***eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJuYW1lIjoidGVzdCJ9.ml0QsTms3K9wMygTu41ZhKlTyjmW9zHQtoS8FusCCjU***

Выпущенный и подписанный токен прикрепляется к отправляемым запросам. Сервер при приеме такого запроса до обработки может проверить, что в токене содержатся корректные идентифицирующие клиента данные, а также проверить подпись, сравнив ее с собственной, которую он вычисляет хэшированием. При совпадении сервер делает вывод о том, что запрос поступил действительно от того клиента, которому изначально был выдан токен.

*JWT* хранит в себе всю информацию о клиенте и не сохраняет состояния (*stateless*). Дополнительной мерой защиты в таком случае может являться сохранение сгенерированных токенов [18] в базе данных (БД) сервера, их текущего состояния и времени истечения. Это позволит вести полный учет всех подключений и самостоятельно отзывать токены до срока их истечения. Схема взаимодействия компонентов в такой системе представлена на рис. 1. При успешном запросе сервер отправляет статус 200. Если клиент получает в ответ статус 403 *Forbidden*, это означает, что предоставленный токен не валиден и клиент должен авторизоваться заново.

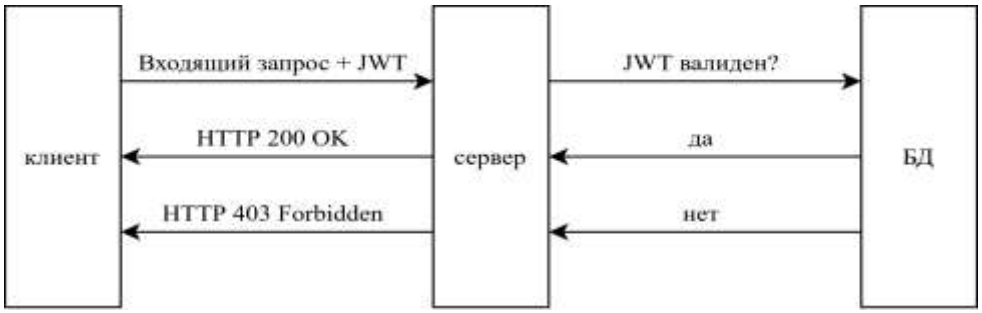


Рисунок 1

Для ускорения повторных авторизаций предлагается использовать *Refresh token*. Это долгоживущий одноразовый токен, который позволяет получить новую пару токенов без ввода аутентификационных данных. На рис. 1 показана схема проверки валидности токена.

Рассмотрим практическую реализацию предлагаемой схемы. Для этого созданы и объединены в сеть виртуальные машины сервера и сенсоров. На рис. 2 представлена схема собранного стенда.

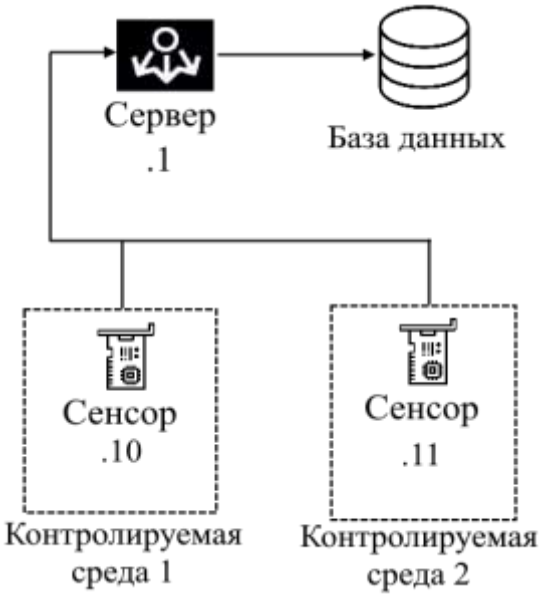


Рисунок 2

Для создания стенда использованы следующие инструменты:

- *Python* – основной язык приложений сервера и клиента.
- *Flask* – веб-фреймворк со встроенным сервером.
- *Ubuntu* – операционная система, на которой развернуты сервер и клиент.
- *Vmware* – платформа виртуализации.

В реализованной схеме клиент собирает определенную информацию, включая параметры загрузки *CPU* и *RAM* своей системы, статистику по окружающим беспроводным сетям, и в виде *POST*-запроса отправляет ее на сервер с определенной периодичностью.

**Заключение**

В результате данной работы описан подход к организации защищенного взаимодействия в сети, состоящей из сенсоров и центрального сервера. Конфиденциальность и целостность данных предлагается обеспечивать с помощью использования *HTTPS*. Для двухсторонней аутентификации предлагается использовать механизм сертификатов. Контроль сессий предлагается обеспечивать с помощью *JWT*. Собран стенд и реализованы описанные механизмы защиты. Если в полной мере осуществляется контроль выпуска доверенных сертификатов и контроль чтения из базы данных, система демонстрирует высокий уровень защищенности.

**Литература**

1. Василишин Н.С., Ушаков И.А., Котенко И.В. Исследование алгоритмов анализа сетевого трафика с использованием технологий больших данных для обнаружения компьютерных атак // Информационные технологии в управлении (ИТУ-2016). Материалы 9-й конференции по проблемам управления. Председатель президиума мультиконференции В.Г. Пешехонов, 2016. – С. 670-675.
2. Браницкий А.А. Архитектура сетевой системы обнаружения атак на основе использования методов машинного обучения и технологий обработки больших данных // Инновации в информационных технологиях, машиностроении и

автотранспорте (ИИТМА-2020). Сборник материалов IV Международной научно-практической конференции с онлайн-участием. Кемерово, 2020. – С. 160-162.
3. Альшаев И.А., Красов А.В., Ушаков И.А. Исследование принципов работы протокола openflow в программно-конфигурируемых сетях // Труды учебных заведений связи, 2017. – Т. 3. – № 2. – С. 16-27.
4. Дубровин Н.Д., Ушаков И.А., Чечулин А.А. Применение технологии больших данных в системах управления информацией и событиями безопасности // В сборнике: Актуальные проблемы инфотелекоммуникаций в науке и образовании. Сборник научных статей V международной научно-технической и научно-методической конференции, 2016. – С. 348-353.
5. Котенко И.В., Ушаков И.А. Использование технологий больших данных для мониторинга инцидентов информационной безопасности // В сборнике: Региональная информатика «РИ-2016». Материалы конференции, 2016. – С. 168-169.
6. Крылов А.В., Ушаков И.А. Метрика защищенности интернет вещей // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО-2022). XI Международная научно-техническая и научно-методическая конференция. Том 1. Санкт-Петербург, 2022. – С. 622-626.
7. Дешевых Е.А., Конюхов В.М., Крылов К.Ю., Ушаков И.А. Исследование методов защиты от инсайдерских атак // В сборнике: Актуальные проблемы инфотелекоммуникаций в науке и образовании. IV Международная научно-техническая и научно-методическая конференция: Сборник научных статей в 2 томах, 2015. – С. 310-313.
8. Бабков И.Н., Пудов К.А., Коновалова В.В., Дибиров Г.М. Исследование способов взаимодействия сетевых устройств на базе микрокомпьютеров // Научные известия, 2022. – № 26. – С. 35-38.
9. Vratonjic N., Freudiger J., Bindschaedler V., Hubaux J. The inconvenient truth about web certificates // В книге: Economics of information security and privacy III, Springer New York, 2013. – С. 79-117.
10. Rahmatulloh A., Gunawan R., Nursuwars F.M. Performance comparison of signed algorithms on JSON Web Token // В книге: IOP Conference Series: Materials Science and Engineering, Том 550, н. 1. IOP Publishing, 2019. – С. 012023.
11. Torrano-Giménez C., Perez-Villegas A. and Maranón G.A. An anomaly-based approach for intrusion detection in web traffic // Dynamic Publishers, 2010.
12. Лазарева М.В. Сравнительный анализ методов аутентификации пользователей: сессии и токены // Информационные технологии в науке, бизнесе и образовании. проблемы обеспечения цифрового суверенитета государства. Материалы XIII Международной научно-практической конференции студентов, аспирантов и молодых ученых. Под общей редакцией А.М. Прохорова, А.В. Царегородцева. Москва, 2022. – С. 40-44.
13. Visočnik V. Comparison of JWT and OAuth 2.0 authorisation and authentication techniques in REST services // Дис. докт. техн. наук; 2018; Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko.
14. Leiding B., Cap C.H., Mundt T. Rashidibajgan Authcoin: validation and authentication in decentralized networks // arXiv preprint arXiv:1609.04955, 2016.
15. Dietz M., Czeskis A., Balfanz D., Wallach D. Origin-bound certificates: A fresh approach to strong client authentication for the web // 21st USENIX Security Symposium, 2012.
16. Story H., Harbulot B., Jacobi I., Jones M. Foaf+ssl: Restful authentication for the social web // В книге: Proceedings of the First Workshop on Trust and Privacy on the Social and Semantic Web (SPOT2009), 2009.

17. URL Концепции безопасности при использовании сертификата X.509 в Центре Интернета вещей Azure https://learn.microsoft.com/ru-ru/azure/iot-hub/iot-hub-x509ca-concept (дата обращения – апрель 2023 г.).
18. Дибиров Г.М., Бабков И.Н., Ковцур М.М. Сравнительный анализ решений для контейнеризации // Молодежная школа-семинар по проблемам управления в технических системах имени А.А. Вавилова, 2022. – Т. 1. – С. 27-29.