МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г. ЧЕРНЫШЕВСКОГО»

Кафедра теоретических основ компьютерной безопасности и криптографии

Защищенная система корпоративной коммуникации

АВТОРЕФЕРАТ
дипломной работы

студента 6 курса 631 группы
специальности 10.05.01 Компьютерная безопасность
факультета компьютерных наук и информационных технологий
Краснобаева Александра Павловича

Научный руководитель
старший преподаватель _____ А. А. Лобов
22.01.2024 г.

Заведующий кафедрой
д. ф.-м. н., доцент _____ М. Б. Абросимов
22.01.2024 г.

Саратов 2024

ВВЕДЕНИЕ

В современном мире, где информационные потоки играют ключевую роль в работе предприятий, обеспечение безопасности корпоративной коммуникации становится неотъемлемой составляющей их успешного функционирования.

Технологическое развитие порождает новые вызовы в области информационной безопасности, требуя более сложных и интегрированных решений для защиты конфиденциальности, целостности и доступности корпоративной информации.

Несанкционированный доступ к коммуникационным системам может позволить злоумышленникам перехватывать и анализировать чувствительную информацию, что в свою очередь может привести к утрате конфиденциальности и нарушению целостности данных.

Утечки корпоративных данных представляют собой серьезный риск, поскольку они могут привести к раскрытию конфиденциальной информации клиентов, стратегических планов, технологических инноваций и других важных аспектов деятельности организации. Такие инциденты не только угрожают репутации компании, но и могут привести к финансовым убыткам, утрате доверия клиентов и снижению конкурентоспособности.

В контексте данной работы рассматриваются ключевые аспекты, касающиеся защиты корпоративной коммуникации от угроз вроде несанкционированного доступа, утечек данных и других потенциальных атак.

В представленной работе внимание сосредотачивается на проектировании и разработке системы защищенной корпоративной коммуникации, которая стремится не только обеспечить эффективный обмен информацией внутри организации, но и гарантировать безопасность данных.

Цель данной работы: реализовать защищенную систему корпоративной коммуникации.

Решаемые задачи:

- спроектировать указанную систему;
- рассмотреть инструменты, используемые для обеспечения необходимой функциональности;
- разработать систему защищенной коммуникации, соответствующую проекту.

Дипломная работа состоит из введения, 3 разделов, заключения, списка использованных источников и 2 приложений. Общий объем работы – 70 страницы, из них 36 страниц – основное содержание, включая 29 рисунков, список использованных источников из 20 наименований.

КРАТКОЕ СОДЕРЖАНИЕ

В первом разделе «Проектирование системы и описание функциональности» рассматривается проектирование системы, основанной на принципах безопасности, контроля доступа и быстрой коммуникации между исполнителями. Данная система должна защищать основные свойства информации, как конфиденциальность, целостность и доступность.

Конфиденциальность — это свойство информации, которое обеспечивает ее защиту от несанкционированного доступа, использования, раскрытия или изменения.

Целостность информации — это свойство, гарантирующее, что данные остаются неповрежденными, неизменными и не подвергнутыми несанкционированным модификациям в процессе их передачи, хранения или обработки.

Доступность информации — это свойство, обеспечивающее возможность получения к нужной информации в необходимый момент времени пользователями, имеющими соответствующие права на доступ.

Для безопасного функционирования системы необходима реализация идентификации, аутентификации и авторизации пользователей. Идентификация – это назначение объекту системы уникальной условной метки, которая позволяет однозначно определить этот объект. Под аутентификацией понимается проверка подлинности объекта, предъявившего данный идентификатор. Аутентификация основана на информации, которая может быть известна только истинному пользователю системы. Авторизация – предоставление субъекту доступа прав доступа, а также предоставление доступа в соответствии с установленными правилами управления доступом.

Для идентификации пользователей системы, в базе данных сохраняется пара логин/пароль, предоставленная пользователем. В целях аутентификации и авторизации используются токены доступа, соответствующие стандарту JSON Web Token (JWT).

Процесс аутентификации выглядит следующим образом:

1. Пользователь вводит логин и пароль;
2. При успешной аутентификации сервер создает пару токенов: access token, который будет использоваться непосредственно для авторизации и refresh token, необходимый для обновления access token. В нагрузке каждого токена указываются идентификатор и роль пользователя, а также его срок действия. Для access token рекомендуется выбирать небольшой срок (несколько минут), в то время как refresh token может быть действителен в течение нескольких часов или суток;
3. Сервер передает пару токенов клиенту. Оба токена сохраняются в куки клиента с флагами HttpOnly и SameSite;
4. При обработке запросов клиента сервер получает значение access token из куки. Сервер проверяет подпись токена, расшифровывает нагрузку и получает информацию о пользователе;
5. Имея действующий refresh token, пользователь может обновить пару токенов без необходимости ввода пары логин/пароль.
6. При истечении срока действия refresh token пользователю будет необходимо заново пройти аутентификацию.

Для авторизации сервер использует информацию о пользователе, полученную из access token, на основании которой предоставляется доступ к ресурсам системы или разрешение на выполнение определенных действий. Так, например, аутентифицированный пользователь имеет право на просмотр и обсуждение только тех задач, на выполнение которых он был назначен. В свою очередь контролировать процесс выполнения задач и назначать на них исполнителей имеет право только пользователь с ролью администратора.

Для оперативного обсуждения деталей и централизации коммуникации между исполнителями, за каждой задачей, созданной в системе, закреплен чат, работающий в реальном времени. Такая возможность появляется благодаря использованию WebSocket. Протокол WebSocket — это двусторонний протокол связи, который позволяет клиенту и серверу поддерживать постоянное

соединение и обмениваться данными в режиме реального времени. Он предназначен для решения проблемы периодических опросов (polling), которая характерна для HTTP.

Для обеспечения эффективной защиты конфиденциальных данных от несанкционированного доступа применяется механизм шифрования сообщений симметричным шифром. На стороне серверного приложения реализован механизм симметричного шифрования, где для шифрования и дешифрования используется один и тот же ключ. В качестве шифра используется стандарт AES. AES (Advanced Encryption Standard). Появившись как FIPS PUB 197 от NIST и, преемник DES, AES поддерживает размеры ключа в 128, в 192 и в 256 бит, в отличие от 56-битовых ключей, предлагаемых DES. Алгоритм Rijndael, изобретённый Йоаном Даменом и Винсентом Рэйменом, был выбран в качестве стандарта. AES является одним из самых распространённых и широко используемых блочных шифров. Он был выбран в качестве стандарта шифрования правительством США и широко применяется во всем мире.

Для удобства обработки и передачи зашифрованных данных, а также для соблюдения стандартов безопасности, зашифрованные сообщения кодируются в формат base64. Кодировка Base64 предназначена для представления произвольных последовательностей из байтов в форме, допускающей использование как прописных, так и строчных букв. Для этого используется подмножество из 65 символов ASCII, позволяя кодировать каждый символ 6 битами (дополнительный 65-й символ «=», используется для обозначения специальной функции обработки). Это обеспечивает представление данных в виде текста, что удобно для хранения и передачи, при этом не уступая в безопасности.

Во втором разделе «Реализация системы защищенной коммуникации» приводятся технические подробности реализации системы, используемые для этого программные компоненты, а также приведены примеры ее работы.

Система реализована на основе модели клиент-сервер. В качестве сервера служит веб-приложение, написанное на языке Java, использующее компоненты,

входящие в экосистему Spring Framework. Клиентом выступает приложение, состоящее из JavaScript-компонентов, созданных с помощью библиотеки React. Для хранения данных на сервере используется СУБД PostgreSQL. Для создания образов и контейнеризации приложений используются Docker и Docker Compose.

В подразделе 2.1 «Используемые программные компоненты» представлены программные компоненты с помощью, которых производилась практическая реализация системы.

В процессе работы над приложением были использованы следующие программные компоненты:

1. Язык программирования Java.
2. Элементы модуля javax.crypto. Данный пакет предоставляет классы и интерфейсы для реализации криптографических операций, таких как шифрование, дешифрование, создание и проверка цифровой подписи, генерация ключей и т. д.
3. Spring Boot — это проект в рамках экосистемы Spring, предоставляющий удобные средства для создания и развертывания самостоятельных приложений на платформе Java. Он облегчает процесс конфигурации и разворачивания приложений.
4. Spring Web MVC включает в себя инструменты для разработки веб-приложений в рамках фреймворка Spring. Он предоставляет различные модули и возможности для обработки HTTP-запросов, взаимодействия с клиентами и создания веб-приложений.
5. Spring Data JPA — это часть проекта Spring Data, предоставляющая абстракции и удобные методы для взаимодействия с базами данных через Java Persistence API (JPA). JPA является стандартом для работы с реляционными базами данных в Java-приложениях.
6. Spring Security — это гибкий фреймворк для обеспечения безопасности приложений на платформе Java. Он предоставляет обширные средства для аутентификации, авторизации и защиты от различных видов атак.

7. Spring Stomp (Simple Text Oriented Messaging Protocol) — это протокол обмена сообщениями, предназначенный для обеспечения асинхронного взаимодействия между клиентом и сервером. Он расширяет протокол WebSocket, предоставляя удобные средства для работы с сообщениями.

8. PostgreSQL – объектно-реляционная СУБД, использующая и расширяющая язык SQL.

9. Java JWT (JSON Web Token) — это библиотека для работы с токенами в формате JSON Web Token.

10. React — это библиотека JavaScript для создания пользовательских интерфейсов. React позволяет строить интерфейсы, разделяя приложение на компоненты и обеспечивая перерисовку только тех частей, которые изменились.

11. Docker — это открытая платформа для автоматизации разработки, доставки и запуска приложений в легковесных, подвижных контейнерах. Контейнеры обеспечивают среду выполнения, изолированную от окружающей системы, что позволяет упаковывать приложение и все его зависимости в единое целое.

12. Docker Compose — инструмент для определения и запуска многоконтейнерных Docker-приложений. Он позволяет описать все аспекты приложения, включая службы, сети, тома данных и другие параметры, в файле YAML, и затем одной командой запустить все сервисы.

В подразделе 2.2 демонстрируется функционал разработанной системы, а именно: регистрация новых пользователей, аутентификация, авторизация. Продемонстрирован функционал по работе с задачами и чатом как со стороны пользователя, так и администратора.

При регистрации нового пользователя его логин/пароль сохраняются в базе данных. Эта пара в дальнейшем используется для аутентификации пользователя. При успешной аутентификации сервер устанавливает клиенту куки, содержащие токен доступа и токен обновления. Для куки установлены

флаги HttpOnly для запрета доступа к ним из JavaScript, а также SameSite со значением Strict для предотвращения их установки из других источников, также для токена доступа установлен флаг Path со значением «/», чтобы эта куки могла быть использована в любом запросе, в случае с токеном обновления, для него установлен флаг Path «/auth/refresh», чтобы он мог быть использован только в запросе на обновление токенов.

Аутентифицированный пользователь имеет доступ к задачам, на выполнение которых он был назначен. При нажатии на карточку задачи появляется всплывающее окно с подробностями о задаче и секцией обсуждения. Администратор системы имеет доступ ко всем задачам, а также в праве менять их состояние (статус, набор исполнителей).

В третьем разделе «Сравнение реализованной системы с существующими аналогами» проводится обзорное сравнение реализованной системы с существующими аналогичными решениями. Сравнение затрагивает продукты различных компаний, такие как Trello, Jira, Slack, YouGile, VK Workspace, Yandex Tracker.

ЗАКЛЮЧЕНИЕ

В рамках данной работы была разработана система для корпоративной коммуникации и управления задачами. Система представляет из себя веб-приложение, серверная часть которого реализована с использованием технологий экосистемы Spring. В частности, для реализации аутентификации и авторизации с помощью JWT использовался компонент Spring Security. Работа клиентской части производилась с использованием компонентов, реализованных с помощью библиотеки React для языка JavaScript.

Поставленные задачи решены. Цель работы достигнута.