Ноянов Р.С. Как Защитить JSONB-поля в POSTGRESQL от утечек данных и инъекций // Международный журнал информационных технологий и энергоэффективности.– 2024. – Т. 9 № 12(50) с. 25–28

Международный журнал информационных технологий и энергоэффективности

Сайт журнала:
http://www.openaccessscience.ru/index.php/ijcse/

УДК 004.736

КАК ЗАЩИТИТЬ JSONB-ПОЛЯ В POSTGRESQL ОТ УТЕЧЕК ДАННЫХ И ИНЪЕКЦИЙ

Ноянов Р.С.
ФГБОУ ВО САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ ИМ. ПРОФЕССОРА М. А. БОНЧ-БРУЕВИЧА, Санкт-Петербург, Россия (193232, г. Санкт-Петербург, просп. Большевиков, 22, корп. 1), e-mail: romannoyanov@gmail.com

JSONB-формат в PostgreSQL активно используется для хранения сложных данных, однако его использование сопряжено с рисками утечек и SQL-инъекций. Статья обсуждает основные угрозы безопасности, связанные с использованием JSONB в PostgreSQL, описывает распространённые сценарии атак и предлагает способы защиты, такие как валидация данных, использование параметризованных запросов и конфиденциальное шифрование.

Ключевые слова: PostgreSQL, JSONB, защита, SQL-инъекции, утечки данных, шифрование, валидация.

HOW TO PROTECT JSONB FIELDS IN POSTGRESQL FROM DATA LEAKS AND INJECTIONS

Nayanov R.S.
ST. PETERSBURG STATE UNIVERSITY OF TELECOMMUNICATIONS NAMED AFTER PROFESSOR M. A. BONCH-BRUEVICH, St. Petersburg, Russia (193232, St. Petersburg, ave. Bolshevikov, 22, bldg. 1), e-mail: romannoyanov@gmail.com

JSONB format in PostgreSQL is widely used for storing complex data, but its use brings risks of data leaks and SQL injections. The article discusses primary security threats associated with JSONB in PostgreSQL, outlines common attack scenarios, and offers protection methods such as data validation, parameterized queries, and confidential encryption.

Keywords: PostgreSQL, JSONB, security, SQL injections, data leaks, encryption, validation.

Введение

С появлением JSONB в PostgreSQL у разработчиков появилась возможность эффективно хранить и обрабатывать неструктурированные данные в виде JSON, которые часто встречаются в современных приложениях. JSONB предоставляет возможности для хранения сложных структур данных, поиска по ключам и удобной фильтрации, что делает его полезным для работы с динамическими данными, которые сложно хранить в традиционных реляционных таблицах. Однако хранение данных в JSONB имеет и свои риски: из-за особенностей формата и его гибкости JSONB становится потенциальной точкой уязвимости для утечек данных и инъекций, особенно если данные принимаются из внешних источников.

Основные угрозы для JSONB-полей включают SQL-инъекции, утечки конфиденциальных данных и неконтролируемый доступ к данным. Например, JSONB-поля, содержащие данные пользователей, могут быть подвергнуты SQL-инъекциям или

использованы злоумышленниками для получения доступа к конфиденциальной информации. Цель этой статьи — рассмотреть типичные угрозы безопасности, связанные с использованием JSONB в PostgreSQL, и описать способы защиты, которые помогут минимизировать риски. Мы обсудим методы, которые включают параметризацию запросов, шифрование, валидацию данных и настройку прав доступа для обеспечения безопасности JSONB.

Как защитить JSONB-поля в PostgreSQL от утечек данных и инъекций

JSONB, как формат хранения данных в PostgreSQL, предоставляет не только гибкость, но и требует дополнительных мер безопасности. Одной из главных проблем является SQL-инъекция, которая может возникнуть при передаче данных в SQL-запросы напрямую. Так как JSONB позволяет хранить вложенные структуры данных, злоумышленники могут использовать уязвимости в коде для внедрения вредоносных команд в запросы, которые обращаются к JSONB-полям. Это может привести к утечке данных или выполнению непредусмотренных команд в базе данных. Чтобы предотвратить SQL-инъекции, рекомендуется использовать параметризованные запросы, которые исключают возможность передачи вредоносных данных. Параметризация защищает запросы, так как данные обрабатываются как значения, а не как части SQL-кода[1].

Ещё одним важным аспектом защиты JSONB-полей является шифрование данных. В случаях, когда JSONB-поля содержат конфиденциальную информацию, такую как пароли или номера кредитных карт, рекомендуется шифровать данные перед их сохранением. Шифрование позволяет минимизировать риск утечек даже при несанкционированном доступе к базе данных. В PostgreSQL для этой задачи можно использовать сторонние библиотеки или встроенные функции для шифрования данных на уровне приложения. Кроме того, для повышения безопасности можно рассмотреть возможность использования функций PostgreSQL, таких как pgcrypto, чтобы шифровать данные в JSONB до их записи в базу[2].

Валидация данных также играет ключевую роль в обеспечении безопасности JSONB-полей. Данные, поступающие в JSONB, часто представляют собой сложные структуры, которые могут быть подвержены уязвимостям при отсутствии должной проверки. Без строгой валидации злоумышленники могут вводить данные, содержащие недопустимые или опасные значения, что может нарушить работу системы или привести к утечке информации. Настройка валидации входящих данных помогает предотвратить такие атаки, позволяя системе принимать только корректные данные. Например, проверка входных данных может включать ограничения на допустимые типы данных и структуру JSON[3].

Также важной частью защиты JSONB является управление доступом. Ограничение прав доступа к JSONB-полям позволяет минимизировать риск, связанный с несанкционированным доступом. В PostgreSQL можно настроить права доступа к отдельным таблицам и полям, чтобы только авторизованные пользователи могли получать доступ к JSONB-данным. Настройка таких ограничений особенно важна в многопользовательских системах, где данные из JSONB могут использоваться разными группами пользователей. Например, можно настроить правила доступа, позволяющие одному пользователю просматривать только собственные данные, хранящиеся в JSONB, что повысит общую безопасность системы[4].

Сегментация данных также помогает защитить JSONB от утечек. Например, если JSONB используется для хранения различных типов данных, целесообразно рассмотреть вариант разделения данных по разным таблицам, чтобы минимизировать доступ к конфиденциальным

данным и снизить вероятность инъекций. Сегментирование данных позволяет хранить более важную информацию в защищённых таблицах, к которым есть доступ только у ограниченного круга лиц[5].

Кроме того, полезным методом защиты JSONB-полей может быть журналирование всех операций с JSONB-данными. Ведение логов всех операций вставки, обновления и удаления данных помогает отслеживать подозрительные действия и выявлять потенциальные угрозы безопасности. Настройка логирования позволяет обнаружить любые необычные действия, связанные с JSONB-полями, что является важной частью контроля безопасности.

Таким образом, защита JSONB в PostgreSQL требует комплексного подхода, включающего параметризацию запросов, шифрование, валидацию данных и настройку прав доступа. Каждая из этих мер помогает устранить уязвимости, возникающие при работе с JSONB, и обеспечивает безопасность данных от утечек и инъекций. При грамотной настройке PostgreSQL с учётом всех перечисленных мер JSONB может быть надёжным инструментом для работы с гибкими и динамическими данными без угрозы для безопасности.

Заключение

JSONB в PostgreSQL открывает широкие возможности для гибкого и эффективного хранения данных, однако требует повышенного внимания к безопасности. Без надлежащей защиты JSONB-поля могут стать точкой уязвимости, через которую злоумышленники смогут получить доступ к конфиденциальным данным или даже проникнуть в систему. SQL-инъекции, утечки данных и недостатки в проверке входных данных — это лишь часть угроз, с которыми можно столкнуться при работе с JSONB.

Для обеспечения безопасности данных рекомендуется использовать параметризованные запросы, шифрование, строгую валидацию данных и настройку доступа к JSONB-полям. Эти меры позволят минимизировать риски и сделать работу с JSONB более безопасной. JSONB остаётся мощным инструментом, но для его надёжного использования в продуктивных системах необходимо соблюдать все рекомендованные меры безопасности, включая регулярные обновления PostgreSQL и внедрение лучших практик защиты данных.

Список литературы

1. Свидетельство о государственной регистрации программы для ЭВМ № 2020664289 РФ. Программа обеспечения системы компьютерного зрения на основе библиотеки OpenCV : № 2020663625 : заявл. 03.11.2020: опубл. 11.11.2020 / И.Е.Пестов, А.М.Гельфанд, Н.Н.Лансере, И.И.Фадеев, заявитель ФГБОУ ВО «С-Пб-кий гос.университет телекоммуникаций им. проф. М.А. Бонч-Бруевича». – EDN PKSCLB.
2. Леснова Е. М., Пестов И. Е. Разработка метода обнаружения и коррекции ошибок для распределенной информационной сети на основе больших данных //Региональная информатика и информационная безопасность. – 2018. – С. 236-240.
3. Пестов И. Е. Методика разработки управляющего воздействия на инстансы облачной инфраструктуры //Вестник Санкт-Петербургского государственного университета технологии и дизайна. Серия 1: Естественные и технические науки. – 2020. – №. 4. – С. 72-76.
4. Пестов И. Е. МЕТОДИКА АВТОМАТИЗИРОВАННОГО ПРОТИВОДЕЙСТВИЯ НЕСАНКЦИОНИРОВАННЫМ ВОЗДЕЙСТВИЯМ НА ИНСТАНСЫ ОБЛАЧНОЙ

ИНФРАСТРУКТУРЫ С ИСПОЛЬЗОВАНИЕМ БЕЗАГЕНТНОГО МЕТОДА СБОРА МЕТРИК.

5. Миняев А. А. Метод оценки эффективности системы защиты информации территориально-распределенных информационных систем персональных данных //Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2020). – 2020. – С. 716-719.

References

1. Certificate of state registration of the computer program No. 2020664289 Russian Federation. The program for providing a computer vision system based on the OpenCV library : No. 2020663625 : application 03.11.2020 : publ. 11.11.2020 / I. E. Pestov, A.M. Gelfand, N. N. Lancere, I.I. Fadeev ; applicant Federal State Budgetary Educational Institution of Higher Education "St. Petersburg State University of Telecommunications named after Prof. M.A. Bonch- Bruevich." – EDN PKSCLB.
2. Lesnova E. M., Pestov I. E. Development of a method of error detection and correction for a distributed information network based on big data //Regional informatics and information security. – 2018. – pp. 236-240.
3. Pestov I. E. Methodology for developing control effects on cloud infrastructure instances //Bulletin of the St. Petersburg State University of Technology and Design. Series 1: Natural and Technical Sciences. - 2020. – No. 4. – pp. 72-76.
4. Pestov I. E. METHOD OF AUTOMATED COUNTERACTION TO UNAUTHORIZED IMPACTS ON CLOUD INFRASTRUCTURE INSTANCES USING AN AGENTLESS METHOD OF COLLECTING METRICS.
5. Minyaev A. A. Method of evaluating the effectiveness of the information protection system of geographically distributed personal data information systems //Actual problems of infotelecommunications in science and education (APINO 2020). – 2020. – pp. 716-719.