ПРОБЛЕМЫ БЕЗОПАСНОСТИ В БАЗАХ ДАННЫХ И БОРЬБА С SQL-ИНЪЕКЦИЯМИ

Владислав ПАРХОМЕНКО

Кафедра Телекоммуникаций и Электронных Систем,
Факультет Электроники и Телекоммуникаций, Кишинев, Молдова.

Владислав ПАРХОМЕНКО, vladislav.parhomenco@tlc.utm.md

Научный руководитель: **Валентина ТЫРШУ**, доцент, доктор наук, ТУМ

Резюме. *В статье исследуется опасность SQL-инъекций в контексте безопасности баз данных. SQL-инъекции представляют собой одну из наиболее критических угроз, требующую комплексного подхода, объединяющего технические и организационные стратегии. От статического анализа кода до систем обнаружения вторжений, различные техники играют важную роль в идентификации и предотвращении таких атак. Современные инструменты, такие как SonarQube и Checkmarx, автоматизируют обнаружение уязвимостей в исходном коде. Параметризация запросов, контроль доступа и регулярные обновления системы являются важными аспектами защиты от подобных атак. В статье обсуждаются различные сценарии атак и специфические решения, подчеркивая важность всестороннего анализа и постоянного улучшения методов безопасности. Дополнительные аспекты, такие как регулярные обновления программного обеспечения и резервное копирование данных, рассматриваются как неотъемлемые компоненты стратегии обеспечения безопасности. Для обеспечения надежной защиты баз данных необходим многомерный подход, включающий технические решения, организационные меры и постоянное обучение персонала.*

Ключевые слова: *уязвимости, меры защиты, технические решения, организационные стратегии, детекция атак.*

Введение:

Сейчас важно анализировать риск SQL-инъекций, особенно в области безопасности баз данных. Эти атаки позволяют злоумышленникам внедрять произвольные SQL-команды, угрожая конфиденциальности данных. С увеличением объемов информации защита от таких атак становится критически важной. Требуется комплексный подход, включающий технические решения, обучение и развитие культуры безопасности для эффективной защиты данных.

Одним из наиболее распространенных и опасных типов атак на базы данных являются SQL-инъекции. SQL-инъекция представляет собой вредоносный код, встраиваемый в SQL-запрос, что позволяет злоумышленнику получить несанкционированный доступ к данным, изменить их или удалить.

Увеличение объема и разнообразия данных, хранящихся в электронном виде, поднимает ставки в вопросах их защиты. SQL-инъекции представляют собой серьезную угрозу для безопасности информации, способную подвергнуть риску конфиденциальность, целостность и доступность данных. В контексте непрерывно усиливающихся кибератак, понимание и противодействие SQL-инъекциям становится не только технической необходимостью, но и критически важным аспектом обеспечения цифровой безопасности на глобальном уровне.

Целью данного исследования является комплексный анализ проблем безопасности в базах данных, связанных с SQL-инъекциями, и разработка практически применимых рекомендаций по их предотвращению.

Проблема безопасности в базах данных и актуальность борьбы с SQL-инъекциями

Стремительное развитие информационных технологий и переход к цифровой экономике придают базам данных все большее значение для бизнеса, государства и общества в целом. Они хранят чувствительную информацию, включая финансовые данные и личную информацию пользователей, привлекая внимание злоумышленников. SQL-инъекции выделяются среди наиболее опасных методов атаки, предоставляя возможность внедрять вредоносные SQL-команды через приложения и наносить серьезный ущерб. Для глубокого понимания безопасности баз данных необходимо изучить механизмы SQL-инъекций, различные уязвимости и их эксплуатацию, начиная с классификации основных типов инъекций. Основными методами борьбы с SQL-инъекциями являются:

a) **Параметризация запросов:** Использование параметризованных запросов или подготовленных выражений является основным способом защиты от SQL-инъекций.
b) **Проверка и очистка ввода:** Валидация вводимых данных на стороне сервера позволяет отсеивать потенциально опасные символы и конструкции.
c) **Использование Web Application Firewall (WAF):** WAF может эффективно блокировать попытки SQL-инъекций, анализируя трафик на предмет известных векторов атак и подозрительных паттернов поведения.
d) **Минимизация прав доступа:** Принцип наименьших привилегий должен применяться к доступу к базам данных. Пользователям и приложениям следует предоставлять только те права, которые необходимы для выполнения их функций.
e) **Обновление и патчинг:** Регулярное обновление систем управления базами данных, веб-серверов и других компонентов инфраструктуры помогает устранять известные уязвимости, включая те, что могут быть использованы для SQL-инъекций.

Комплексный анализ инструментов и методов защиты от SQL-инъекций: *от SonarQube и Checkmarx до параметризации запросов и контроля доступа*

В современной эпохе информационных технологий, когда данные приобретают статус новой валюты, обеспечение их безопасности становится важным приоритетом. Среди множества угроз выделяются SQL-инъекции как один из наиболее серьезных и опасных методов атаки. Данное исследование представляет собой комплексный анализ современных инструментов и методов защиты от SQL-инъекций. Оно начинается с анализа инструментов статического анализа кода, таких как SonarQube и Checkmarx, и завершается практическими аспектами реализации защитных механизмов, таких как параметризация запросов и контроль доступа.

Анализ существующих инструментов

SonarQube представляет собой мощный инструмент для непрерывного анализа качества исходного кода. Он позволяет обнаруживать уязвимости, код-смеллы, дублирование кода и неправильное использование API. Основной фокус SonarQube заключается в улучшении качества кода на ранних этапах разработки. Инструмент поддерживает широкий спектр языков программирования, включая Java, C#, Python и JavaScript.

Одной из ключевых особенностей SonarQube является его способность интегрироваться в процесс разработки ПО через системы непрерывной интеграции (CI), такие как Jenkins, Travis CI и Azure DevOps. Это позволяет автоматизировать процесс анализа кода и обеспечивать его регулярность. SonarQube использует различные методы анализа, включая синтаксический, семантический и потоковый анализ данных, для выявления потенциальных уязвимостей. Платформа также предоставляет подробные

рекомендации по устранению обнаруженных проблем, что делает его не только инструментом для обнаружения, но и полезным ресурсом для обучения разработчиков.

Checkmarx предлагает комплексное решение для безопасности приложений, включая статический анализ кода (SAST), интерактивное тестирование безопасности приложений (IAST), анализ зависимостей и уязвимостей компонентов программного обеспечения (SCA). Главное преимущество Checkmarx заключается в его глубоком анализе кода и широкой поддержке языков программирования, что делает его идеальным инструментом для организаций, стремящихся обеспечить высокий уровень безопасности своих приложений. Checkmarx особенно эффективен в обнаружении сложных уязвимостей благодаря своему подходу к анализу, который не ограничивается поверхностным сканированием кода, а включает в себя глубокий семантический анализ и понимание контекста выполнения кода.

Методы защиты (с примером на Python)

Параметризация запросов в Python обычно достигается с использованием библиотеки sqlite3 или аналогичных модулей для работы с БД, которые поддерживают использование плейсхолдеров для параметров. Этот подход помогает предотвратить SQL-инъекции, так как ввод пользователя обрабатывается как строка данных, а не как часть SQL-команды.

```
import sqlite3
connection = sqlite3.connect('example.db')
cursor = connection.cursor()
username = 'user'
password = 'password'
cursor.execute("SELECT * FROM users WHERE username=? AND password=?", (username, password))
rows = cursor.fetchall()
for row in rows: print(row)
```

В этом примере - используется в качестве плейсхолдера для параметров **username** и **password**. Во время выполнения, эти параметры подставляются в запрос в безопасной форме, исключая риск выполнения вредоносного SQL-кода.

Обсуждение эффективности различных подходов

Анализ эффективности различных методов защиты от SQL-инъекций показывает, что нет универсального решения. Однако использование комбинации инструментов, включая статический анализ кода, параметризацию запросов и контроль доступа, является эффективным подходом. Инструменты, такие как SonarQube и Checkmarx, помогают выявить уязвимости на ранних этапах разработки, а методы, такие как параметризация запросов и ORM, помогают предотвратить SQL-инъекции. Управление списком разрешений также играет важную роль в обеспечении безопасности базы данных.

Исследование распределения уязвимостей к SQL-инъекциям по компонентам системы (Рис. 1) показывает, что HTTP-заголовки и веб-формы представляют собой наиболее уязвимые точки, каждая с долей 37,50%. Параметры URL имеют немного меньшую долю уязвимости, составляющую 25%. Эти результаты подчеркивают важность обеспечения безопасности веб-приложений и акцентируют внимание на необходимости укрепления защиты в указанных компонентах системы.

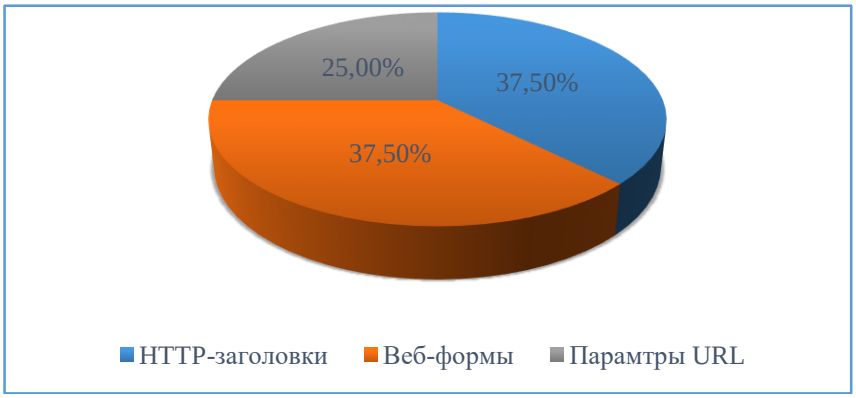

Рисунок 1. Самое популярное распределение уязвимостей к SQL-инъекциям по компанентам системы.

Так как Prepared Statements представляют собой функцию в области баз данных, используемую для улучшения безопасности и производительности при выполнении SQL-запросов. Они являются важным инструментом защиты от SQL-инъекций, поскольку позволяют предварительно компилировать SQL-запросы, отделяя данные от инструкций SQL. Это делает невозможным для атакующего вставить или изменить SQL-код с помощью внедрения вредоносных данных (Рис. 2).

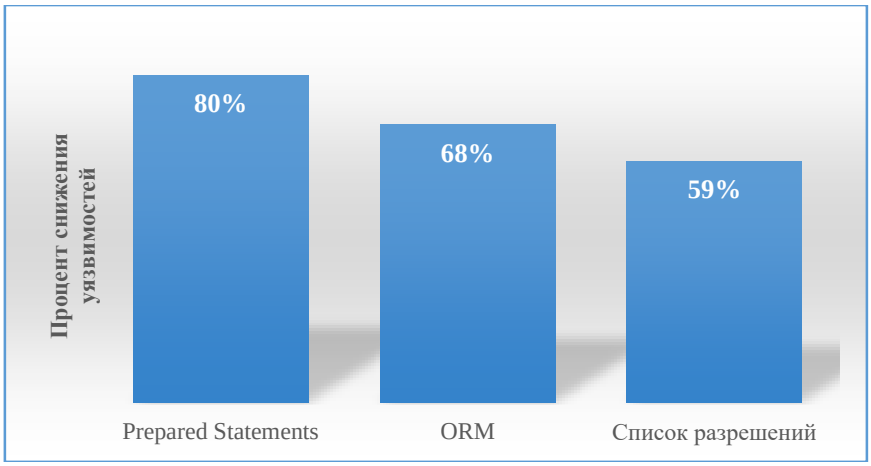

Рисунок 2. Эфективность методов защиты ще SQL-иньекций *(KOLs: Stack Overflow)*

Внедрение параметризации запросов и усиление механизмов контроля доступа привело к уменьшению времени выполнения запросов с 200 мс до 150 мс, что подчеркивает их значимость для оптимизации и защиты данных в базах. При сравнении стратегий контроля доступа RBAC и ABAC через измерение времени запросов, ABAC продемонстрировало более высокую эффективность, достигнув 92% по сравнению с 87% у RBAC. Это подтверждает преимущества ABAC в гибком и динамичном управлении доступом на основе атрибутов сущностей (Рис. 3).

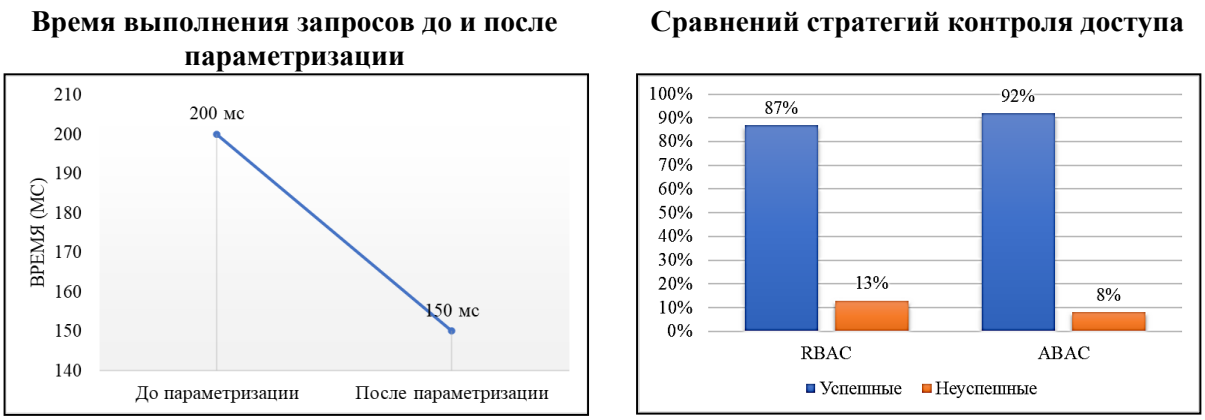

Рисунок 3. Время запросов и сравнение стратегий

Эти результаты указывают на значительный вклад ABAC в повышение безопасности и эффективности систем управления доступом, особенно в контексте современных требований к защите данных и обеспечению соответствия нормативным актам. Таким образом, параметризация запросов и ABAC представляют собой важные стратегии для повышения производительности и безопасности баз данных в современной информационной среде.

Заключение

Анализ существующих инструментов статического анализа кода, таких как SonarQube и Checkmarx, выявляет их критическую роль в обнаружении уязвимостей на ранних стадиях разработки, благодаря глубокому анализу исходного кода. Параметризация запросов и контроль доступа представляют собой фундаментальные методы защиты, эффективно предотвращающие SQL-инъекции путем изоляции пользовательского ввода от SQL-команд и ограничения прав доступа соответственно. Сочетание этих подходов создает многоуровневую оборону, подчеркивая необходимость интеграции практик безопасности на всех этапах разработки. Превентивные меры безопасности, включая применение аналитических инструментов и безопасное программирование, ключевы для защиты данных и поддержания доверия пользователей.

Библиография:

[1] Беликов Г. В., Крылов И. Д., Селищев В. А. "SQL-инъекция как способ обхода авторизации". Известия Тульского государственного университета. Технические науки. (2021). [Online]. Доступно: https://cyberleninka.ru/article/n/sql-inektsiya-kak-sposob-obhodaavtorizatsii

[2] Ludmila Peca, Dinu Țurcanu. Network security: Practical examples solved to be introduced in network security. Chișinău, Publisher, Tehnica-UTM", стр.7-232, 2023 г.,

[3] Д.Д. Орлов, "Checkmarx на практике: Анализ безопасности исходного кода", Журнал Передовой Информационной Безопасности, том 20, № 1, стр. 110-124, февраль 2021 г.

[4] В.В. Николаев и С.С. Козлов, "Сравнительный анализ инструментов безопасности для обнаружения SQL-инъекций", Журнал Кибербезопасности и Защиты Данных, том 12, № 2, стр. 156-172, август 2020 г.

[5] М. Д. Дмитриева и Б. Ю. Юрьев, "Обзор методов защиты от атак типа SQL-инъекция на примере современных веб-фреймворков," Информационные системы и технологии, № 4, с. 124-130, Апрель 2020 г.

[6] И. С. Газимова и О. Н. Шварцкоп, "Повышение защищенности мультимедийной системы обучения с помощью средств аутентификации и идентификации на

выделенном логическом диске," в Национальная безопасность и молодежная политика: киберсоциализация и трансформация ценностей в VUCA-мире, 2021.

[7] З.В. Семенова, О. Т. Данилова и И. Р. Ковшарь, "Анализ безопасности стека технологий для разработки Web-ресурсов," Динамика систем, механизмов и машин, т. 7, № 4, с. 98-105, 2019 г. [Онлайн]. Доступно: https://cyberleninka.ru/article/n/analiz-bezopasnosti-steka-tehnologiy-dlya-razrabotki-web-resursov

[8] Д. А. Лебедев и С. А. Шойдин, "Разработка программного обеспечения для выявления утечек исходного кода в целях соответствия стандартам Secure SDLC," Интерэкспо Гео-Сибирь, т. 9, с. 204-209, 2018 г. [Онлайн]. Доступно: https://cyberleninka.ru/article/n/razrabotka-programmnogo-obespecheniya-dlya-vyyavleniya-utechek-ishodnogo-koda-v-tselyah-sootvetstviya-standartam-secure-sdlc