УДК 004.056

ОСНОВНЫЕ МЕРЫ ПО СНИЖЕНИЮ РИСКОВ SQL-ИНЪЕКЦИЙ

Е.Г. Ручаевская, Е.Д. Буйневич

Учреждение образования «Белорусский государственный университет инфор-атики и радиоэлектроники» филиал «Минский радиотехнический колледж», г. Минск, Республика Беларусь

Аннотация. В работе рассматривается проблема SQL-инъекций как одной из наиболее актуальных угроз информационной безопасности, сохраняющей лидирующие позиции в рейтинге OWASP несмотря на многолетнюю известность. Описаны механизмы реализации атак, потенциальные последствия вплоть до полной компрометации серверов баз данных. Предложен комплексный подход к защите, сочетающий

технические меры (параметризованные запросы, валидацию входных данных по принципу белых списков, разделение прав доступа, специализированные средства защиты) и организационные практики.
Ключевые слова: SQL-инъекции; уязвимость; вредоносный код; безопасность баз данных; запрос; параметризованные запросы; валидация данных; права доступа; средства защиты; программное обеспечение.

KEY MEASURES TO REDUCING THE RISKS OF SQL INJECTION

E.G. Ruchaevskaya, E.D. Buinevich
Educational institution "Belarusian State University of Informatics and Radioelec-tronics" branch "Minsk Radio Engineering College", Minsk, Republic of Belarus

Abstract. The paper examines the problem of SQL injection as one of the most pressing threats to information security, which continues to hold a leading position in the OWASP ranking despite being known for many years. The mechanisms of attack implementation and the potential consequences, including complete compromise of database servers, are described. A comprehensive approach to protection is proposed, combining technical measures (parameterized queries, input validation based on the whitelisting principle, access right separation, specialized security tools) and organizational practices.
Keywords: SQL injection; vulnerability; malicious code; database security; query; parameterized queries; data validation; access rights; security tools; software.

Введение

Несмотря на то, что угроза SQL-инъекций известна сообществу разработчиков уже более двух десятилетий, она продолжает занимать лидирующие позиции в рейтинге опасностей OWASP (Open Web Application Security Project). SQL-инъекция – это метод внедрения злоумышленником вредоносных SQL-конструкций (ключевых слов или операторов) в поля ввода данных приложения. Успешная реализация такой атаки может привести к катастрофическим последствиям: от несанкционированного просмотра конфиденциальных данных (персональные данные, платежная информация) до полного уничтожения базы данных или получения злоумышленником административного контроля над системой [1].

Актуальность темы обусловлена тем, что многие современные приложения по-прежнему содержат уязвимый код, используют устаревшие фреймворки либо созданы с нарушением принципов безопасной разработки вследствие низкой квалификации разработчиков.

Основная часть

SQL-инъекции представляют собой класс уязвимостей, при которых вредоносный код внедряется в пользовательские входные данные для последующего выполнения на стороне сервера баз данных. Данный тип

атак нарушает семантику исходных запросов, что позволяет злоумышленнику обойти механизмы аутентификации и несанкционированно взаимодействовать с СУБД. Деструктивное воздействие SQL-инъекций может проявляться в различных формах: от инициирования ошибок приложения и переполнения буфера, ведущего к отказу в обслуживании, до несанкционированного повышения привилегий вплоть до уровня операционной системы.

Защита от SQL-инъекций не может быть обеспечена каким-то одним методом. Лучше всего применять как технические, так и организационные меры.

Одним из наиболее эффективных технических средств является использование параметризованных операторов, которые позволяют четко разделить логику SQL-запроса и пользовательские данные [2]. В подготовленном операторе структура запроса фиксируется заранее, а данные, полученные от пользователя, подставляются не через конкатенацию строк, а связываются с переменными. Благодаря этому при попытке SQL-инъекции внедренный код воспринимается не как исполняемая команда, а как строковое значение.

Однако, параметризованные операторы не обеспечивают защиту в случаях, когда требуется динамическое формирование имен таблиц или параметров сортировки. Поэтому ключевым дополнением к ним выступает валидация входных данных. Такой подход включает использование белых списков (разрешение только заранее определенных безопасных значений), проверку типа данных, ограничение длины полей ввода и контроль формата ввода.

Еще один важный принцип – минимизация прав доступа. Использование при подключении к базе данных учетной записи с правами администратора является грубой ошибкой. Для снижения рисков необходимо создавать для каждого приложения отдельную учетную запись, предоставляя ей строго минимально необходимые права, а также применять принцип разделения прав для отдельных модулей приложения.

Дополнительным барьером служат специализированные средства защиты. Программные или аппаратные решения анализируют входящий трафик и блокируют подозрительные запросы с признаками внедрения вредоносного кода еще до их передачи в приложение.

Нельзя забывать и о безопасности инфраструктурного слоя. Уязвимости могут возникать не только на уровне кода приложений, но и в самих системах управления базами данных, веб-серверах, а также в используемых фреймворках. Использование устаревшего программного обеспечения с известными уязвимостями делает все остальные меры защиты бессмысленными, поэтому своевременное обновление компонентов является критически важным.

Эффективная защита невозможна без систематического обучения разработчиков и администраторов принципам безопасного кодирования, а также без налаженного взаимодействия между командами разработки, информационной безопасности и администраторами баз данных. Практика применения библиотек и фреймворков, прошедших предварительную проверку со стороны службы информационной безопасности, позволяет снизить риск внесения уязвимостей на этапе разработки.

Заключение

SQL-инъекции остаются серьезной угрозой для информационной безопасности, однако, в отличие от многих сложных видов атак, они вполне предотвратимы. Эффективная стратегия защиты должна быть комплексной. Эффективная стратегия защиты должна быть комплексной и включать как технические меры, так и организационные практики. Такой подход не только повышает уровень защиты, но и ускоряет процессы разработки.

Список использованных источников

1. Карпиеня М.В. (2021) Защита веб-приложений от SQL-инъекций. Бизнес. Образование. Экономика : Международная научно-практическая конференция (Минск, 1-2 апр. 2021 г.), 163-165.

2. Кэмпбелл Л., Мейджорс Ч. (2020) Базы данных. Инжиниринг надежности. Санкт-Петербург, Издательство «Питер».

References

1. Karpienia M.V. (2021) Protection of Web Applications from SQL Injections. Business. Education. Economics: International Scientific and Practical Conference (Minsk, April 1-2, 2021), 163-165 (in Russion).

2. Campbell L., Majors C. (2020) Databases. Reliability Engineering. St. Petersburg, Piter Publishing (in Russion).

Сведения об авторах

Ручаевская Е.Г., канд. пед. наук, доц., преподаватель, учреждение образования «Белорусский государственный университет информатики и радиоэлектроники» филиал «Минский радиотехнический колледж», elenruch@gmail.com.

Буйневич Е.Д., преподаватель, учреждение образования «Белорусский государственный университет информатики и радиоэлектрони-ки» филиал «Минский радиотехнический кол-ледж», e.pulmanovskaya@gmail.com.

Information about the authors

Ruchaevskaya E., Cand. Sci. Ped., Associate Professor, teacher, educational institution "Belarusian State University of Informatics and Radioelectronics" branch "Minsk Radio Engineering College", elenruch@gmal.com.

Buinevich E., teacher, educational institution "Belarusian State University of Informatics and Radioelectronics" branch "Minsk Radio Engineering College", e.pulmanovskaya@gmail.com.