УДК 004.056.5 DOI: 10.26565/2519-2310-2019-3-03

# SQL-ИНЪЕКЦИИ: ОБЗОР ПОТЕНЦИАЛЬНЫХ СПОСОБОВ ЗАЩИТЫ

Юрий Попов, Сабина Рузудженк, Карина Погорелая

Харьковский национальный университет имени В.Н. Каразина, пл. Свободы, 4, Харьков, 61022, Украина
yuriipopov18@gmail.com, ruzudzhenk.jb@gmail.com, karina.pogorelka@gmail.com

**Рецензент:** Ирина Лисицкая, д.т.н., проф., ХНУ имени В.Н. Каразина, пл. Свободы, 6, Харьков, 61022, Украина
lisitska@karazin.ua

Поступила: Ноябрь 2019.

***Аннотация.*** *В статье представлен краткий обзор известных техник взлома программ и веб-сайтов, работающих с базами данных. На основе проведенного анализа основных разновидностей SQL-атак выделены наиболее серьезные типы угроз: – внутриполосная, слепая и внеполосная. Утверждается, что по совокупности характеристик, внеполосная SQL-атака является наиболее опасной. Обращено внимание на необходимость периодического тестирования и мониторинга веб-сайтов, что является актуальным средством защиты от SQL-инъекций. Отмечено, что наилучший метод тестирования – попытка подвергнуть код SQL-инъекции. Рассмотренные способы защиты способны повысить общий уровень безопасности программных продуктов от атак типа SQL-инъекция, обеспечивают корректную работу приложений и целостность пользовательских данных.*

***Ключевые слова****: SQL-инъекция; базы данных; внеполосная SQL-атака; внутриполосная SQL-атака.*

## 1 Введение

В последние годы практически во всех современных компаниях, работающих в сфере высоких технологий, всё большую популярность приобретает тенденция использования в бизнес-процессах разнообразных web-приложений, информационные ресурсы которых, обрабатывают и хранят персональные данные клиентов, компаний подрядчиков и, непосредственно, владельцев компаний [1]. Использование web-приложений наделяет производственные и бизнес-процессы новыми качествами, прежде всего, такими как: - высокая мобильность бизнеса; - доступность сервисов; - непрерывность бизнес-процессов; - масштабируемость получаемого эффекта и т.п.. Учитывая все эти обстоятельства, вопросы обеспечения информационной безопасности (ИБ) при обработке и хранении персонифицированной и «чувствительной» корпоративной информации, сохраняют высочайший приоритет и являются крайне актуальным направлением деятельности, как для специалистов соответствующих подразделений компаний (отделов и служб ИБ), так и для профильных специалистов отрасли ИБ. Так, согласно статистике Positive Technologies [2], около 70% веб-сайтов поддаются различным атакам, среди которых одно из первых мест занимают атаки типа SQL-инъекция.

SQL-инъекция – одна из самых распространённых техник взлома программ и веб-сайтов, работающих с различными базами данных [3]. Атака, как правило, производится на основе внедрения в различные типы запросов некорректных SQL операторов, что позволяет злоумышленнику получить, практически, полный несанкционированный доступ к соответствующей базе данных (БД), локальным файлам, а также возможность удалённого выполнения произвольных операций на сервере. Кроме того, SQL-атаки, зачастую, являются результатом неэкранированного ввода, передаваемого сайту и используемого как часть запроса к БД [3].

Таким образом, вопросы организации противодействия атакам типа «SQL-инъекция» являются актуальным направлением деятельности и требуют постоянной модификации уже существующих и разработки новых способов защиты и методик противодействия.

## 2 Механизм атаки на основе SQL-инъекции

Атака на основе SQL-инъекции производится путем запросов к БД на основе вводимых пользователем данных с применением некорректно фильтруемых escape-символов.

 

С точки зрения архитектуры самого приложения пользователь взаимодействует с веб-сервером, посредством веб-клиента, который взаимодействует с SQL-сервером, по протоколу HTTP (Рис. 1). Как правило, веб-сервер требует от пользователя аутентификации в системе (пользователю необходимо предоставить *name* и *password*). Для этого веб-сервер выполняет операцию: *SELECT * FROM Users WHERE userid='name' AND userpass='password'*, где *Users* – таблица БД, содержащая персональные данные пользователей; *name* – имя пользователя; *password* – пароль, соответствующий имени пользователя. При определённом подборе злоумышленником информации для ввода, есть вероятность обхода действующего механизма идентификации в системе и возможность последующей модификации исполняемого запроса (*например, использование комментариев дает возможность деактивировать проверку пароля*) [4].

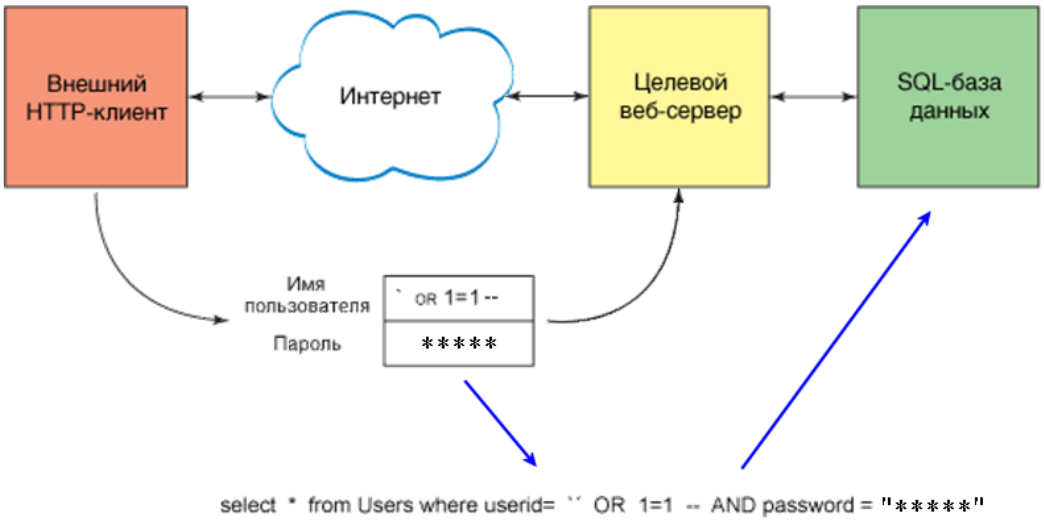


Рис. 1 – Механизм атаки на основе SQL-инъекции

Таким образом, данные, вводимые на веб-странице, способны использовать уязвимости при обмене с БД. SQL-инъекция использует введенные пользователем некорректные данные для получения разрешения на прямое взаимодействие с внутренней БД, в результате чего атакующий (хакер) получает доступ ко всем записям данной таблицы БД.

### 2.2 Основные разновидности SQL-атак

В настоящий момент известно несколько типов атак, применяющих SQL-инъекции. Необходимо отметить, что, в целом, уязвимость типа SQL-инъекция известна уже на протяжении семнадцати лет, однако продолжает интересовать специалистов по безопасности до сих пор. Эта разновидность уязвимости впервые была описана в декабре 1998 года на примере сервера Microsoft SQL, в котором было возможно получение конфиденциальных данных посредством использования команд в обычных пользовательских вводах, таких как «имя» или «номер телефона» (rain.forest.puppy, 1998) [8]. Хотя это событие и было впервые задокументировано в 1998, однако SQL-инъекция не привлекала большого внимания в сообществе информационной безопасности, вплоть до 2002 года. По мере развития SQL-инъекций, появлялись их новые типы, основное отличие которых заключается в способах и сложности их внедрения, а также в используемых способах защиты (будут рассмотрены ниже). Коротко рассмотрим 3 основных типа атак, применяющих технику SQL-инъекции [5].

*Внутриполосная SQLi (классическая SQLi)*. Данный тип является самым распространенным, инъекция, в основном, происходит, когда потенциальный злоумышленник использует один и тот же канал связи для запуска атаки и последующего сбора результатов.

При этом внутриполосные SQL-инъекции делятся на две разновидности:

- SQL-инъекция на основе ошибок. Основана на сообщении об ошибке, выдаваемом сервером БД, для получения информации о её структуре;

- внедрение SQL на основе объединения. Основана на использовании оператора SQL UNION для объединения результатов двух или более операторов SELECT в один результат, который затем возвращается, как HTTP ответ [6].

*Инференциальная SQL-инъекция (слепая или Blind SQLi).* При атаках с использованием слепой SQL-инъекции злоумышленник не может увидеть результат своей атаки внутри группы, поскольку данные не передаются через веб-приложение. По этой причине она также называется Blind SQLi [6]. Инференциальные SQL-инъекции бывают двух типов:

- булевая слепая SQLi. Основана на отправке SQL-запроса в базу данных, что вынуждает приложение возвращать другой результат в зависимости от того, возвращает ли запрос результат «TRUE» или «FALSE»;

- базирующаяся на времени слепая SQLi. Основана на отправке соответствующего SQL-запроса к БД, что заставляет базу данных ждать определенное время (в секундах), прежде чем ответить. Собственно, время ответа и укажет злоумышленнику, является ли результатом запроса «ИСТИНА» или «ЛОЖЬ».

Атака типа внедрения SQL может быть возможна из-за некорректной обработки входных данных, используемых в SQL-запросах.

*Внеполосная SQLi.* Внедрение SQL-инъекции происходит, когда злоумышленник не может использовать один и тот же канал для запуска атаки и сбора результатов.

Внеполосные методы предлагают злоумышленнику альтернативу логическим методам SQL-инъекции, основанным на времени, особенно если ответы сервера не очень стабильны, делая вывод, основанный на времени, ненадежным [6].

Эти методы зависят от способности сервера БД делать DNS (*Domain Name Server*) или HTTP-запросы для доставки необходимых данных хакеру.

Анализ проблематики по данному направлению позволяет утверждать, что наибольшую угрозу целостности информации представляют именно вышеперечисленные типы SQL-атак. При комплексном использовании данных разновидностей атак (*интеграции или коллаборации друг с другом*) практически не остается возможности сохранить конфиденциальность персональных данных. Соответственно, так как утечка информации подобного рода крайне недопустима, то владельцам различных web-приложений необходимо продумывать адекватную стратегию защиты, демпфирующих возможный эффект от подобного рода атак.

## 3 Способы защиты программных продуктов от атак типа SQL-инъекция

Во избежание подобного рода атак необходимо максимально ограничить программному продукту доступ к серверным данным, т.е. разработать и внедрить приложения, работающие исключительно с параметризованными запросами. Таким образом, при атаках соответствующих приложений, не имеющих достаточных прав доступа к исходным таблицам, исключается возможность получения злоумышленником нелегитимного доступа к локальным данным или БД. В этой связи коротко рассмотрим наиболее известные способы защиты от атак типа SQL-инъекция, а также выделим рекомендуемые меры защиты при разработке компонентов БД и сформируем общие рекомендации для разработки web-ориентированных систем, использующих сервера БД.

*Первый способ* предусматривает необходимость обеспечить фильтрацию данных, поступающих на сервер: - т.е. специальные символы должны экранироваться, а численная информация – подвергаться проверке на вводимый тип. Кроме того необходимо ограничивать вход (*например, количество вводимой информации, после проверки на сервере; запросы, превышающие установленное количество – отклоняются*).

Кроме того, важным аспектом при защите от подобного рода атак является безопасность самого процесса хранения конфиденциальных данных. Так, например, используемая БД не должна содержать такие данные в виде простого текста или таблицы (*пароли должны быть*

*хэшированы, а также содержать случайным образом генерируемую строку, добавляемую перед шифрованием и др.*) [7].

*Вторым способом* обеспечения безопасности является использование серверами БД параметризованных запросов. В общем случае параметризованные запросы представляют собой способ передачи данных, при котором внешние параметры передаются серверу отдельно от SQL-запросов. В большинстве языков программирования реализация данных функций уже предусмотрена [7]:

1. *Delphi* – свойство *TQuery.Params;*
2. *Java* – класс *PraparedStatement;*
3. *C#* – свойство *SqlCommand.Parameters;*
4. *PHP* – свойство *MySQLi.*

*Третий способ* – это максимальное ограничение отображения сообщений об ошибках пользователей (*отображаются общие сообщения об ошибках, возможные для всех сбоев*). Однако, при этом, на стороне сервера необходимо отслеживать все неудачные запросы для возможности их последующего просмотра и анализа (*аудита инцидента*) в случае атаки.

Периодическое тестирование и мониторинг, также, можно смело отнести к довольно эффективным *способам защиты* от SQL-инъекций. При этом наилучшим способом тестирования, является попытка подвергнуть код SQL-инъекции. Существует множество сканеров для обнаружения подобных атак, которые находят уязвимые места, а также тестируют различные разновидности атак.

В целом, следует акцентировать внимание на том, что для обеспечения потенциально более высокого уровня защиты системы, необходимо использовать (*по возможности*) сочетание всех приведенных выше способов защиты. При настройке/программировании таких систем, нужно проверить программный код на выявление уязвимостей и подвергнуть код SQL-инъекции, что в последующем поможет практически мгновенно отслеживать реакцию программы на подобные атаки.

## 5 Выводы

Анализ информации об известных инцидентах безопасности, обзор соответствующей периодики и обобщение мнения соответствующих отраслевых специалистов позволяет утверждать, что сокрытие уязвимостей и защита конфиденциальных данных являются важнейшими направлениями обеспечения информационной безопасности, которые не теряют своей актуальности и до настоящего времени. Вследствие этого при разработке программных продуктов не стоит пренебрегать приемами проверки и фильтрации данных.

Выделенные в работе способы защиты могут существенно обезопасить программные системы от атак, основанных на механизме SQL-инъекции, уменьшить общую подверженность атакам, а также обеспечить корректную работу приложений и целостность пользовательских данных.

Опираясь на обзор известных способов защиты от атак типа SQL-инъекция и анализ эффективности возможных мер противодействия, следует констатировать, что:

1- к основным способам защиты от атак типа «SQL-инъекция» можно отнести:

а) обеспечение фильтрации данных, поступающих на сервер БД;

б) использование серверами БД параметризованных запросов;

2 - к рекомендуемым мерам предосторожности при разработке компонентов БД, следует отнести:

а) интеграция дополнительных мер безопасности для всей хранимой информации (*например, хэширование паролей или использование ЭЦП*);

б) выявление устранение потенциальных уязвимостей зависящих/независящих от данных (*например, путем периодического тестирования и мониторинга*);

3 - для обеспечения более высокого уровня защиты системы, крайне необходимо комплексирование нескольких способов защиты.

## Ссылки

[1] *Популярность языков программирования: рейтинг 2018*. [Электронный ресурс]. Режим доступу: https://techrocks.ru/2018/07/29/programming-languages-popularity-2018/
[2] Статистика Positive Technologies. [Электронный ресурс]. Режим доступу: https://www.ptsecurity.com/ru-ru/research/analytics/
[3] Д. Евтеев, *SQL Injection от А до Я*. [Электронный ресурс]. Режим доступу: https://www.ptsecurity.com/upload/corporate/ru-ru/analytics/PT-devteev-AdvancedSQL-Injection.pdf.
[4] П. Яновски та Є.Бурмакин, *Основы веб-хакинга*. [Электронный ресурс]. Режим доступу: white-hat-hacking-ru-sample.pdf.
[5] М. Егоров, "Выявление и эксплуатация SQL-инъекций в приложениях", *Защита информации. INSIDE*, № 2, с. 2-8, 2011.
[6] *SQL инъекции. Проверка, взлом, защита*. [Электронный ресурс]. Режим доступа: https://habr.com/ru/post/130826/.
[7] К.И. Колесникова, Ю.И.Кошкарёва, *SQL-инъекции*. [Электронный ресурс]. Режим доступа: https://docplayer.ru/46070414-Sql-inekcii-nauchnyy-rukovoditel-gapanyuk-yu-e-k-t-n-docent.html
[8] *SQL Injection: The Longest Running Sequel in Programming History*. [Электронный ресурс]. Режим доступа: https://www.researchgate.net/publication/324227697_SQL_Injection_The_Longest_Running_Sequel_in_Programming_History/link/5ac6a25d4585151e80a37b27/download

**Reviewer:** Irina Lisitska, Doctor of Sciences (Engineering), Full Prof., Academician of the Academy of Sciences of Applied Radio Electronics, V. N. Karazin Kharkiv National University, Kharkiv, 61022, Ukraine.
E-mail: lisitska@karazin.ua

Received: November 2019.

**Authors:**
Yuri Popov, student of CSD, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine. E-mail: yuriipopov18@gmail.com
Sabina Ruzudzhenk, student of CSD, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
E-mail: ruzudzhenk.jb@gmail.com
Karina Pogorelaya, student of CSD, V. N. Karazin Kharkiv National University, Kharkiv, Ukraine.
E-mail: karina.pogorelka@gmail.com

**SQL-injections: an overview of potential protection methods.**

**Abstract**. This work exposes a brief review of well-known hacking techniques for programs and websites working with databases. Based on a comprehensive analysis of the main types of SQL attacks, the most profound threats are identified. They include in-band, blind and out-of-band types of SQL injections. An out-of-band SQL attack is considered to be the most dangerous because of its characteristics' combination. Attention was also paid on the need of periodic testing and monitoring, which is an actual method of protection against SQL injections. It is emphasized, that the best testing method is undertaking code by the SQL injection. The protection methods, reviewed above, can increase the overall security of software products from attacks such as SQL injection, ensure the correct functionality of applications and the integrity of user data.

**Keywords**: SQL-injection; Protection methods; Data bases (DB); In-band SQLi; Out-Of-Band SQLi.

**Рецензент**: Ірина Лисицька, д-р тех. наук, проф., Харківський національний університет імені В.Н. Каразіна, Харків, 61022, Україна. E-mail: lisitska@karazin.ua

Надійшло: Листопад 2019.

**Автори:**
Юрій Попов, студент факультету комп'ютерних наук, Харківський національний університет імені В.Н. Каразіна, Харків, 61022, Україна. E-mail: yuriipopov18@gmail.com
Сабіна Рузудженк, студентка факультету комп'ютерних наук, Харківський національний університет імені В.Н. Каразіна, Харків, 61022, Україна. E-mail: ruzudzhenk.jb@gmail.com
Карина Погоріла, студентка факультету комп'ютерних наук, Харківський національний університет імені В.Н. Каразіна, Харків, 61022, Україна. E-mail: karina.pogorelka@gmail.com

**SQL-ін'єкції: огляд потенційних способів захисту.**

**Анотація.** У статті надано короткий огляд відомих технік злому програм і веб-сайтів, що працюють з базами даних. На основі проведеного аналізу основних різновидів SQL-атак виділені найбільш серйозні типи загроз: - внутріполосна, сліпа та позасмугова. Стверджується, що за сукупністю характеристик, позасмугова SQL-атака є найбільш небезпечною. Звернуто увагу на необхідність періодичного тестування і моніторингу веб-сайтів, що є актуальним засобом захисту від SQL-ін'єкцій. Відзначено, що найкращий метод тестування - спроба піддати код SQL-ін'єкції. Розглянуті способи захисту здатні підвищити загальний рівень безпеки програмних продуктів від атак типу SQL-ін'єкція, забезпечують коректну роботу додатків та цілісність призначених для користувача даних.

**Ключові слова:** SQL-ін'єкція; бази даних (БД); внутріполосна SQL-атака; позасмугова SQL-атака.