Системная модель обнаружения SQL-инъекций на основе комбинированного анализа синтаксических структур и поведенческих характеристик запросов

О.С. Безнос[1], С. А. Зарубина[1], С.Е. Кошевая[1], Е.К. Согомонян[1], Е.А. Васильева[2]

[1]Кубанский Государственный Технологический университет, Краснодар
[2]Nanjing university of information science and technology, China

Аннотация: Статья представляет собой системное исследование информационных потоков в связке «приложение–СУБД» и предлагает комплексную модель защиты от SQL-инъекций, основанную на многоуровневом анализе. В рамках системного анализа рассматривается полный цикл обработки запроса, что позволяет преодолеть фрагментарность существующих подходов. Проанализированы ограничения существующих методов, основанных на сигнатурном анализе, машинном обучении и синтаксической проверке. Для повышения надежности и точности детектирования предложен новый комбинированный метод, интегрирующий статический синтаксический анализ абстрактных синтаксических деревьев (AST) запросов с динамическим поведенческим анализом сессий. Ключевой особенностью синтаксического модуля является применение коэффициента Жаккара для оценки структурного сходства путей в AST, что обеспечивает эффективное выявление полиморфных инъекций. Поведенческий модуль анализирует временные и статистические паттерны последовательности запросов, что позволяет обнаруживать сложные time-based атаки. Предложенный метод демонстрирует практическую значимость для защиты современных информационных систем.

Ключевые слова: SQL-инъекции, системный анализ, машинное обучение, синтаксический анализ, абстрактное синтаксическое дерево, поведенческий анализ, коэффициент Жаккара, полиморфные атаки, time-based атаки.

Введение

SQL-инъекции остаются одним из наиболее критичных векторов атак на веб-приложения согласно рейтингу OWASP Top-10 [1]. Анализ современных исследований показывает, что несмотря на длительную историю существования данной уязвимости, ее актуальность только возрастает в связи с усложнением архитектур приложений и появлением новых техник обхода защиты. Особую опасность представляют полиморфные и time-based атаки, которые эффективно обходят традиционные системы обнаружения.

Полиморфные SQL-инъекции характеризуются способностью динамически изменять свою синтаксическую структуру при сохранении семантической эквивалентности. Это достигается за счет использования различных техник обфускации, кодирования и изменения структуры запросов. Time-based инъекции относятся к категории слепых атак и используют задержки выполнения запросов для извлечения информации из базы данных. Оба типа атак представляют серьезную проблему для традиционных систем безопасности.

Как отмечается в работе по сравнительному анализу подходов к обнаружению SQL-инъекций с помощью методов машинного обучения [2], существующие методы защиты можно условно разделить на три категории: сигнатурный анализ, статический анализ кода приложения и динамический анализ во время выполнения. Однако каждый из подходов обладает существенными ограничениями, что подтверждается исследованиями, проведенными в работах [3 - 5].

Данная статья представляет собой системное исследование информационных потоков в связке «приложение–СУБД». Целью исследования является разработка целостной системной модели защиты, преодолевающей фрагментарность существующих подходов за счет многоуровневого анализа, охватывающего как синтаксическую корректность единичных запросов, так и поведенческие паттерны их последовательностей в сессии.

1. Критический анализ современных методов

Проведенный анализ современных методов обнаружения SQL-инъекций выявил ряд системных проблем и ограничений. В работе [2] был проведен детальный анализ эффективности различных алгоритмов машинного обучения. Авторы исследовали такие алгоритмы как деревья решений, метод опорных векторов и нейронные сети. Деревья решений показали точность 87.3%, метод опорных векторов продемонстрировал точность 89.1%, а нейронные сети достигли точности 91.4%. Однако исследование выявило ключевую проблему - высокую зависимость качества классификации от полноты обучающей выборки и слабую эффективность против полиморфных атак, которые способны обходить модели машинного обучения за счет динамического изменения синтаксической структуры запросов. Кроме того, авторы работы [2] отмечают, что традиционные методы машинного обучения требуют постоянного переобучения при изменении структуры легитимных запросов, что снижает их практическую применимость в динамично развивающихся системах.

В исследовании [3] предложен метод обнаружения SQL-инъекций с использованием распознавания грамматических паттернов и анализа поведения доступа. Авторы используют подход, основанный на синтаксическом анализе SQL-запросов через построение грамматических деревьев и последующее сравнение с эталонными шаблонами. Метод включает анализ поведения доступа к данным, что позволяет выявлять аномальные последовательности операций. Но данный подход имеет ограничения при работе с полиморфными инъекциями, которые изменяют грамматическую структуру запросов, сохраняя их семантику, а также требует значительных вычислительных ресурсов для анализа грамматических паттернов в реальном времени, что может ограничивать его применение в высоконагруженных системах.

В исследовании [4] демонстрируется подход, основанный на синтаксическом анализе SQL-запросов. Авторы предлагают метод нормализации запросов, который заключается в замене параметров на унифицированные плейсхолдеры. Например, исходный запрос SELECT * FROM users WHERE id = 1 OR 1=1 преобразуется в нормализованную форму SELECT * FROM users WHERE id = param. Основным ограничением данного подхода является неспособность обнаруживать сложные инъекции, сохраняющие синтаксическую корректность, а также полиморфные атаки, которые используют семантически эквивалентные, но синтаксически различные конструкции.

В работе [5] предложена система, использующая следующие признаки для анализа: длину SQL-запроса, количество ключевых слов и наличие специальных символов. Однако метод демонстрирует высокий уровень ложных срабатываний при работе со сложными легитимными запросами, которые могут содержать нестандартные конструкции или работать с большими объемами данных. Кроме того, данный подход слабо эффективен против time-based инъекций, которые не оставляют явных синтаксических маркеров в запросах.

Анализ показал, что ни один из существующих методов в отдельности не способен эффективно противостоять всем типам SQL-инъекций, особенно при наличии целенаправленных атак с использованием продвинутых техник обфускации и полиморфизма. Это обосновывает необходимость разработки нового комбинированного подхода, который бы интегрировал преимущества существующих методов и минимизировал их недостатки.

2. Предлагаемый комбинированный метод

2.1. Общая архитектура системы

Предлагаемый метод реализует трехуровневую архитектуру детектирования SQL-инъекций, основанную на системном анализе информационных потоков между приложением и базой данных. Архитектура системы включает последовательную обработку входящих запросов через три основных модуля.

Пусть Q представляет множество всех возможных SQL-запросов, тогда для каждого запроса $q \in Q$ выполняется последовательная обработка:

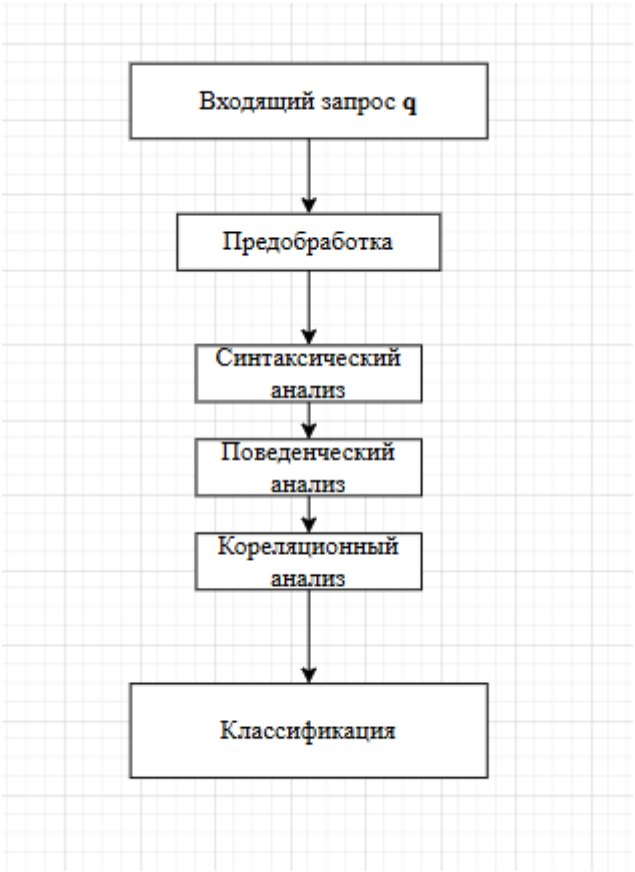

Рис. 1 – Схема работы комбинированного метода

На первом уровне выполняется предварительная обработка и нормализация SQL-запросов. Процесс нормализации можно формализовать следующим образом (1):

$$N(q) = replace\_params(q, P) \ \forall q \in Q \quad (1)$$

где $P = \{p_1, p_2, \ldots, p_n\}$ представляет множество параметров запроса, а $N(q)$ - нормализованная форма запроса после замены всех параметрических значений на унифицированные плейсхолдеры.

2.2. Синтаксический анализ на основе абстрактного синтаксического дерева

Для каждого нормализованного запроса строится абстрактное синтаксическое дерево. Формально абстрактное синтаксическое дерево (AST) представляется как ориентированный ациклический граф (2):

$$AST(q) = (V, E, L, R) \quad (2)$$

где $V = \{v_1, v_2, \ldots, v_m\}$ - множество вершин, соответствующих токенам SQL, $E \subseteq V \times V$ - множество ребер, отражающих синтаксические отношения, $L: V \to T$ - функция маркировки вершин типами токенов, $R \in V$ - корневая вершина дерева, T - множество типов SQL-токенов.

Метрика синтаксической аномальности вычисляется на основе сравнения с эталонными шаблонами (3):

$$P_{syntax}(q) = 1 - max\{sim(AST(N(q)), AST_{ref}) \mid AST_{ref} \in R_{type}(q)\} \quad (3)$$

где $R_{type}(q)$ - множество эталонных абстрактных синтаксических деревьев для типа запроса q, $sim(AST_1, AST_2)$ - функция схожести синтаксических деревьев.

Функция схожести деревьев объединяет компоненты (4):

$$sim(AST^1, AST^2) = w_{jaccar} \times J(P^1, P^2) + \\ + w_{depth} \times D(AST_1, AST_2) + w_{structure} \times S(AST_1, AST_2) \quad (4)$$

где $w_{jaccar} = 0.5$, $w_{depth} = 0.2$, $w_{structure} = 0.3$ - весовые коэффициенты, причем $w_{jaccar} + w_{depth} + w_{structure} = 1$.

Коэффициент Жаккара для путей вычисляется как (5):

$$J(P_1, P_2) = |P_1 \cap P_2| / |P_1 \cup P_2| \quad (5)$$

где P_1, P_2 - множества путей от корня к листьям в AST и AST_2 Данная метрика позволяет эффективно выявлять структурные различия между запросами за счет анализа совпадения путей в синтаксических деревьях.

Метрика глубины оценивает различия в сложности структуры запросов (6):

$$D(AST_1, AST_2) = 1 - |depth(AST^1) - \\ - depth(AST_2)| / max(depth(AST_1), depth(AST_2)) \quad (6)$$

где $depth(AST)$ - максимальная глубина дерева.

Структурная схожесть основана на анализе топологических характеристик (7):

$$S(AST_1, AST_2) = (2 \times |M|) / (|V_1| + |V_2|) \quad (7)$$

где M - максимальное паросочетание в двудольном графе вершин AST и AST_2.

2.3. Поведенческий анализ сессий запросов

На втором уровне выполняется поведенческий анализ сессии запросов. Сессия определяется как временная последовательность запросов от одного источника (8):

$$S = \{(q_1, t_1), (q_2, t_2), \ldots, (q_n, t_n)\} \quad (8)$$

где q_i - i-й SQL-запрос в сессии, t_i - временная метка запроса, n - количество запросов в сессии.

Для сессии S вычисляется вектор признаков (9):

$$F(S) = [f_1(S), f_2(S), f_3(S), f_4(S), f_5(S)] \quad (9)$$

Компоненты вектора признаков включают:

1. Стандартное отклонение длин запросов (10):
$$f_1(S) = \sigma(\{length(q_i) \mid i = 1..n\}) \quad (10)$$
2. Соотношение операций чтения/записи (11):
$$f_2(S) = count_{SELECT(S)} / (count_{UPDATE(S)} + \\ + count_{INSERT(S)} + count_{DELETE(S)} + \varepsilon) \quad (11)$$
3. Энтропия последовательности операторов (12):

$$f_3(S) = -\Sigma\, p(o) \times log_2(p(o)) \quad (12)$$

для $o \in$ {SELECT, UPDATE, INSERT, DELETE}

4. Максимальный временной интервал (13):

$$f_4(S) = max(\{t_{i+1} - t_i \mid i = 1..n-1\}) \quad (13)$$

5. Коэффициент уникальности таблиц (14):

$$f_5(S) = |unique\_tables(S)| / |total\_table\_references(S)| \quad (14)$$

Каждый признак нормализуется на основе статистик обучающей выборки (15):

$$norm(f_i) = (f_i - \mu_i) / \sigma_i \quad (15)$$

где μ_i - среднее значение признака f_i на нормальных сессиях, σ_i - стандартное отклонение признака f_i на нормальных сессиях.

Оценка поведенческой аномальности вычисляется как (16):

$$P_{behavior}(S) = \Sigma\, w_i \times |norm(f_i)| \text{ для } i = 1..5 \quad (16)$$

где весовые коэффициенты w_i вычисляются методом главных компонент.

2.4. Интегральная модель классификации

На третьем уровне реализуется корреляционный анализ, который объединяет результаты предыдущих этапов. Итоговая оценка аномальности вычисляется как (17):

$$P_{final}(q, S) = \alpha \times P_{syntax}(q) + \beta \times P_{behavior}(S) + \gamma \times P_{con}(q, S) \quad (17)$$

где $\alpha + \beta + \gamma = 1, \alpha, \beta, \gamma \geq 0$. Эмпирически установлены значения: $\alpha = 0.5, \beta = 0.3, \gamma = 0.2$.

Контекстная компонента включает (18):

$$P_{con}(q, S) = \delta_1 \times C_{time}(t) + \delta_2 \times C_{privilege}(u) + \delta_3 \times C_{frequency}(S) \quad (18)$$

Классификация запроса выполняется на основе порогового правила (19):

$$class(q) = \{ \text{"INJECTION"}, \text{если } P_{final}(q, S) > \theta(t) \quad (19)$$

$$\{ \text{"NORMAL"}, \text{иначе}$$

где $\theta(t) = \theta_{base} + \Delta\theta(t)$ - адаптивный порог, $\theta_{base} = 0.7$ - базовый порог.

3. Экспериментальные исследования

Для валидации предложенного метода были проведены комплексные экспериментальные исследования с использованием нескольких специализированных датасетов. В качестве исходных данных использовались датасет из исследования [2] содержащий 15,000 записей, данные из работы [5] включающие 8,000 записей, а также датасет [6], содержащий >30000 реальных SQL-запросов. Результаты работ представлены в таблице №1.

Таблица № 1

Сравнительный анализ эффективности методов обнаружения SQL-инъекций

Метод	Precision	Recall	F1-score	Ложные срабатывания
Деревья решений [2]	0.873	0.845	0.859	14.2%
Метод опорных векторов [2]	0.891	0.862	0.876	12.5%
Нейронные сети [2]	0.914	0.882	0.898	9.8%
Метод грам. паттернов [3]	0.889	0.857	0.873	11.3%
Метод классификации [5]	0.901	0.866	0.883	10.7%
Предлагаемый метод	0.942	0.918	0.930	6.1%

Сравнительный анализ эффективности предложенного метода проводился в сопоставлении с тремя современными подходами. Результаты сравнения демонстрируют значительное преимущество предложенного метода по всем основным метрикам качества. Полнота (Recall) предложенного метода достигла 91.8%, что свидетельствует о его способности обнаруживать подавляющее большинство атак. Высокие показатели для полиморфных инъекций объясняются эффективностью коэффициента Жаккара в выявлении структурных аномалий, даже при сохранении семантической эквивалентности запросов.

Заключение

В данной работе в рамках системного анализа информационных потоков был разработан комбинированный метод обнаружения SQL-инъекций, демонстрирующий существенное преимущество за счет рассмотрения информационного обмена «приложение–СУБД» как целостной системы с многоуровневой моделью анализа. Научная новизна предлагаемого решения заключается в нескольких ключевых аспектах. Для теории системного анализа разработана формальная модель, объединяющая синтаксический анализ на основе модифицированной метрики схожести абстрактного синтаксического дерева (с использованием коэффициента Жаккара для путей) с поведенческим анализом сессий запросов. Для практики защиты информации предложен и верифицирован адаптивный алгоритм классификации, показавший на экспериментальных данных прирост точности на 3.1% и снижение уровня ложных срабатываний на 37.8% относительно наиболее эффективных аналогов. Метод проявил максимальную результативность против сложных полиморфных и time-based атак, традиционно ускользающих от детектирования. Таким образом, работа вносит вклад в развитие методов системного анализа и обработки информации, предлагая не просто новый алгоритм, а целостный подход к мониторингу и защите информационных потоков в корпоративных системах.

Литература

1. OWASP Foundation. OWASP Top Ten Web Application Security Risks (2021). – URL: owasp.org/Top10/2021/A03_2021-Injection/index.html (дата обращения: 10.12.2025).

2. Юдова Е. А., Лапонина О. Р. Сравнительный анализ подходов к обнаружению sql-инъекций с помощью методов машинного обучения // International Journal of Open Information Technologies. 2023.

№6. URL: cyberleninka.ru/article/n/sravnitelnyy-analiz-podhodov-k-obnaruzheniyu-sql-inektsiy-s-pomoschyu-metodov-mashinnogo-obucheniya (дата обращения: 03.12.2025).

3. Gao H., Zhu J., Liu L., et al. Detecting SQL Injection Attacks Using Grammar Pattern Recognition and Access Behavior Mining // 2019 IEEE International Conference on Energy Internet (ICEI) : IEEE, 2019. pp. 493-498.

4. Оглов В.А. Метод обнаружения SQL-инъекций на основе синтаксического анализа запросов: магистерская диссертация; Тольяттинский гос. ун-т. – Тольятти, 2023. – 142 с. – URL: dspace.tltsu.ru/bitstream/123456789/33719/1/%D0%9E%D0%B3%D0%BB%D0%BE%D0%B2%20%D0%92.%D0%90._%D0%9F%D0%98%D0%BC-2303%D0%B0.pdf (дата обращения: 10.12.2025).

5. Митрофанов М. В., Крибель А. М., Фроленков А. С., Спицын О. Л. Методика обнаружения атак, типа SQL-инъекция на основе алгоритмов искусственного интеллекта, методом задачи классификации // Наукоемкие технологии в космических исследованиях Земли. 2021. №5. URL: cyberleninka.ru/article/n/metodika-obnaruzheniya-atak-tipa-sql-inektsiya-na-osnove-algoritmov-iskusstvennogo-intellekta-metodom-zadachi-klassifikatsii (дата обращения: 05.12.2025).

6. Sajid576. SQL Injection Dataset // Kaggle. – 2021. – URL: kaggle.com/datasets/sajid576/sql-injection-dataset (дата обращения: 10.12.2025).

7. Атрощенко В. А., Шаповало А. А., Безнос О. С. База данных для информационной системы по источникам и накопителям электроэнергии постоянного тока в системах электроснабжения// Электронный сетевой политематический журнал "Научные труды

КубГТУ". – 2024. – № 5. – С. 150-159. – DOI: 10.26297/2312-9409.2024.5.5. – EDN GNJUMO

8. Алгазали С. М. М., Айвазов В. Г., Кузнецова А. В. Совершенствование процесса поиска неэффективных SQL-запросов в СУБД Oracle // Инженерный вестник Дона. 2017. №4. URL: ivdon.ru/ru/magazine/archive/n4y2017/4511.
9. Верещагина Е. А., Колесникова Д. С., Рудниченко А. К. Особенности разработки информационной системы для предприятия // Инженерный вестник Дона. 2019. №1. URL: ivdon.ru/ru/magazine/archive/n1y2019/5533.
10. Paul Alan, Sharma Vishal, Olukoya Oluwafemi, SQL injection attack: Detection, prioritization & prevention, Journal of Information Security and Applications, Volume 85, 2024, 103871, ISSN 2214-2126, doi.org/10.1016/j.jisa.2024.103871.(sciencedirect.com/science/article/pii/S2214212624001737X)

References

1. OWASP Foundation. OWASP Top Ten Web Application Security Risks (2021). URL: owasp.org/Top10/2021/A03_2021-Injection/index.html.
2. Yudova E. A., Laponina O. R. International Journal of Open Information Technologies. 2023. №6. URL: cyberleninka.ru/article/n/sravnitelnyy-analiz-podhodov-k-obnaruzheniyu-sql-inektsiy-s-pomoschyu-metodov-mashinnogo-obucheniya.
3. Gao H., Zhu J., Liu L. et al. 2019 IEEE International Conference on Energy Internet (ICEI): IEEE, 2019. p.p. 493-498.
4. Oglov V.A. Metod obnaruzheniya SQL-in``ekcij na osnove sintaksicheskogo analiza zaprosov [SQL Injection Detection Method Based on Query Parsing]: magisterskaya dissertaciya; Tol`yattinskij gos. un-t.

Tol`yatti, 2023. 142 p. URL: dspace.tltsu.ru/bitstream/123456789/33719/1/%D0%9E%D0%B3%D0%BB%D0%BE%D0%B2%20%D0%92.%D0%90._%D0%9F%D0%98%D0%BC-2303%D0%B0.pdf.

5. Mitrofanov M. V., Kribel` A. M., Frolenkov A. S., Spicyn O. L. Naukoemkie texnologii v kosmicheskix issledovaniyax Zemli. 2021. №5. URL: cyberleninka.ru/article/n/metodika-obnaruzheniya-atak-tipa-sql-inektsiya-na-osnove-algoritmov-iskusstvennogo-intellekta-metodom-zadachi-klassifikatsii.

6. Sajid576. SQL Injection Dataset Kaggle. 2021. URL: kaggle.com/datasets/sajid576/sql-injection-dataset.

7. Atroshhenko V. A., Shapovalo A. A., Beznos O. S. E`lektronny`j setevoj politematicheskij zhurnal Nauchny`e trudy` KubGTU. 2024. № 5. pp. 150-159. DOI: 10.26297/2312-9409.2024.5.5. – EDN GNJUMO

8. Algazali S. M. M., Ajvazov V. G., Kuzneczova A. V. Inzhenernyj vestnik Dona. 2017. №4. URL: ivdon.ru/ru/magazine/archive/n4y2017/4511.

9. Vereshhagina E. A., Kolesnikova D. S., Rudnichenko A. K. Inzhenernyj vestnik Dona. 2019. №1. URL: URL: ivdon.ru/ru/magazine/archive/n1y2019/5533.

10. Paul Alan, Sharma Vishal, Olukoya Oluwafemi, Journal of Information Security and Applications, Volume 85, 2024, 103871, ISSN 2214-2126, doi.org/10.1016/j.jisa.2024.103871.(sciencedirect.com/science/article/pii/S2214212624001730X)

Дата поступления: 23.12.2025

Дата публикации: 7.02.2026