УДК 004.021

# ПРИМЕРЫ ЗАЩИТЫ ОТ SQL-ИНЪЕКЦИЙ В WORDPRESS

Р.Д. Осипов, А.Ю. Савицкий, А.А. Игнатенко
*Учреждение образования «Белорусский государственный университет информатики и радиоэлектроники», г. Минск, Республика Беларусь*

**Аннотация**. В 2025 – 2026 году популярность WordPress делает данную платформу приоритетной целью для кибератак. В статье анализируются актуальные угрозы безопасности, в частности SQL-инъекции. Рассматриваются риски, связанные с использованием кода, созданного искусственным интеллектом без аудита безопасности. Оценивается вероятность эксплуатации брешей в современной экосистеме WordPress и подчеркивается критическая важность защиты данных в условиях открытой архитектуры платформы.
**Ключевые слова**: WordPress; кибербезопасность; SQL-инъекции; A03:2021 – Injection; шаблон; уязвимости; искусственный интеллект; защита данных; веб-разработка; аудит безопасности.

# EXAMPLES OF SQL INJECTION PROTECTION IN WORDPRESS

R.D. Osipov, A.Yu. Savitsky, A.A. Ignatenko
*Educational Institution "Belarusian State University of Informatics and Radioelectronics", Minsk, Republic of Belarus*

**Abstract**. In 2025–2026, the popularity of WordPress makes the platform a primary target for cyberattacks. This article analyzes current security threats, specifically focusing on SQL injections. It examines the risks associated with using AI-generated code without a proper security audit. The study evaluates the probability of exploiting vulnerabilities within the modern WordPress ecosystem and emphasizes the critical importance of data protection given the platform's open architecture.
**Keywords**: WordPress; cybersecurity; SQL injection; A03:2021 – Injection; template; vulnerabilities; artificial intelligence; data protection; web development; security audit.

## Введение

В 2025 – 2026 году WordPress продолжает доминировать на рынке веб-разработки обеспечивая работу более 43% всех сайтов в интернете [2]. Однако такая популярность имеет обратную сторону – огромная

экосистема становится главной мишенью для атакующих. По данным HackerOne и Pathstack, SQL-инъекции стабильно входят в число самых опасных угроз, составляя до 15% от всех выявленных уязвимостей в этой среде [2]. Несмотря на высокий уровень безопасности самого ядра WordPress, его открытая архитектура и массовое использование кода, созданного искусственным интеллектом без должного аудита, создают огромную поверхность для атак. Вероятность эксплуатации подобных брешей оценивается экспертами в 8,5 баллов из 10, что делает вопрос защиты данных приоритетным для любого владельца сайта [2].

## Основная часть

Ключевой риск в современной веб-разработке классифицируется международном стандарте OWASP как категория A03:2021 – Injection. Данная угроза сохраняет свою критическую актуальность и в 2026 году [1]. Рассматриваемая проблема напрямую связана с классической ошибкой CWE-89, при которой стирается граница между программной логикой и данными пользователя, что позволяет злоумышленнику «подмешивать» собственные команды в запросы к базе данных.

Основные проблемы безопасности в WordPress обычно скрыты не в самом «движке», а в коде сторонних тем и плагинов. Чаще всего это происходит из-за того, что разработчики пренебрегают правильной проверкой входящих данных. Самый опасный сценарий – когда необработанная информация из адресной строки браузера или полей формы попадает прямиком в SQL-запрос. В современных мире нарушители используют высокоскоростные сканеры, которые за секунды находят такие ошибки и позволяют мгновенно менять структуру запросов, открывая доступ к конфиденциальной информации сайта.

Рассмотрим простой пример того, как выглядит опасный код, написанный на php[3]:

```
$id = $_GET[data_id'];
$wpdb->query("DELETE FROM {$wpdb->prefix}data WHERE id = $id");
```

В данном случае нарушитель может передать вместо обычного номера команду – 123 OR 1=1. Это заставит базу данных выполнить команду для всех строк сразу, что приведет к удалению всех данных. Чтобы этого не случилось, в руководстве WordPress Developer Resources есть главный инструмент защиты – метод $wpdb->prepare(). Он работает как шаблон: сначала пользователь пишет текст запроса с «метками», а потом передает данные. Система сама очистит их и подставит на нужные места, не давая нарушителю шансов изменить логику [1].

```
$id = $_GET[data_id'];
$wpdb->query($wpdb->prepare(
        "DELETE FROM {$wpdb->prefix}data WHERE id = $d",
        $id
));
```

Одной только подготовки запроса мало. Важно использовать «очистку» данных сразу, как только они поступили на сайт. Например, если пользователь ждет число, используйте функцию absint(), которая превратит любой странный текст в обычную цифру [2]. Огромную роль играют Nonces – это специальные проверочные ключи WordPress. Данные ключи подтверждают, что запрос отправил именно настоящий пользователь с сайта, а не посторонний скрипт со стороны [3].

```
if (! isset($_POST['my_nonce']) || !  wp_verify_nonce($_ POST['my_nonce'], 'delete_action' )){
        wp_die('Текст который будет отображаться в случае ошибки');
}
```

## Заключение

Глубокое понимание механизмов работы инъекций в сочетании с регулярным сканирование системы – единственный надеждой способ защитить проект сегодня. Безопасность WordPress – это непрерывный процесс, когда разработчик фильтрует каждое входящее слово, использует подготовленные запросы и применяет современные средства мониторинга, сайт остается по-настоящему защищенным от угроз категории A03:2021.

В современных условиях, когда атаки автоматизированы и происходят за доли секунды, уже нельзя полагаться на внимательность специалистов по информационной безопасности или базовые настройки сервера. Для обеспечения реальной безопасности сайта необходимо внедрять комплексную систему контроля, работающую на опережение. Консольный интерфейс WP-CLI позволяет разработчикам не только управлять контентом, но и оперативно проводить технический аудит, выявляя подозрительные изменения в файловой структуре. В то же время интеграция с специализированными базами данных, такими как WPScan и Patchstack, превращает защиту из пассивной в проактивную, где владельцы получают оповещения о критических уязвимостях в плагинах еще до того, как нарушители успеют ими воспользоваться

Важно осознать, что безопасность WordPress – это не конечная точка, а непрерывный процесс. Только когда специалист по информационной безопасности фильтрует каждое входящее слово, строго использует параметризованные запросы $wpdb->prepare()

и применяет современные средства мониторинга в реальном времени, сайт остается по-настоящему защищенным. Во время ИИ-угроз многоуровневая оборона позволяет эффективно противостоять рискам категории A03:2021 – Injection, сохраняя репутацию проекта и конфиденциальность данных пользователей.

## Список использованных источников

1. Атаки на веб и WordPress. – СПб.: БХВ-Петербург, 2021. – 256 с.
2. OWASP Top 10: [отчет] / OWASP Foundation. – 2021. – URL: owasp.org (дата обращения: 17.03.2026).
3. WordPress с нуля. – СПб.: БХВ-Петербург, 2021. – 304 с.

## References

1. Attacks on Web and WordPress. St. Petersburg: BHV-Petersburg. 2021. 256 pages.
2. OWASP Top 10: [Report] / OWASP Foundation. 2021. URL: owasp.org (Accessed: 17.03.2026).
3. WordPress from Scratch. St. Petersburg: BHV-Petersburg 2021. 304 pages.

## Сведения об авторах

**Осипов Р.Д.**, курсант, учреждение образования «Белорусский государственный университет информатики и радиоэлектроники», osipovr695@gmail.com.
**Савицкий А.Ю.**, кандидат военных наук, старший преподаватель, учреждение образования «Белорусский государственный университет информатики и радиоэлектроники», a.savitskij@bsuir.by.
**Игнатенко А.А.,** преподаватель, учреждение образования «Белорусский государственный университет информатики и радиоэлектроники», a.ignatenko@bsuir.by

## Information about the authors

**Osipov R.D.**, Cadet, Educational Institution "Belarusian State University of Informatics and Radioelectronics", osipovr695@gmail.com.
**Savitsky A.Yu.**, PhD in Military Science, Senior Lecturer, Educational Institution "Belarusian State University of Informatics and Radioelectronics", a.savitskij@bsuir.by.
**Ignatenko A.A.,** Lecturer, Educational Institution "Belarusian State University of Informatics and Radioelectronics", a.ignatenko@bsuir.by.