УДК 004.056:355.4511:004.75 https://doi.org/10.30837/IYF.PCEIP.2024.091

SQL ІН'ЄКЦІЯ: ЗАГРОЗА БЕЗПЕЦІ ВЕБ-ЗАСТОСУНКІВ

Блінна В.С.
Науковий керівник – ст., викл. В'юхін Д.О.
Харківський Національний університет радіоелектроніки, каф. БІТ,
м. Харків, Україна
e-mail: veronika.blinna@nure.ua

This work is devoted to exploring the threat of SQL Injection to the security of web applications. Five major types of SQL injection attacks were reviewed, their methodologies and potential impact on system integrity were examined. The paper delineates the consequences of a successful SQL Injection attack, highlighting the severity of the threat it poses to web application security. Additionally, the significance of implementing robust security measures and defenses to effectively mitigate the risks associated with SQL Injection vulnerabilities in web applications is emphasized.

У сучасному цифровому світі веб-додатки відіграють ключову роль у забезпеченні доступу до даних і виконанні різних операцій. Однак, разом із розширенням функціоналу веб-додатків зростає і рівень загроз безпеки. Одним із поширених методів атак є SQL ін'єкція, що становить серйозну загрозу для даних.

Метою доповіді є розгляд вразливостей SQL-запитів. SQL-ін'єкція - це вразливість веб-безпеки, яка дозволяє зловмиснику втручатися в запити, які додаток робить до внутрішньої бази даних. Як правило, це дає змогу переглядати дані, які він зазвичай не може отримати. Це можуть бути інші користувачі, або будь-які інші дані, доступ до яких має сам додаток [1]. У багатьох випадках зловмисник може змінювати або видаляти ці дані, викликаючи постійні зміни у вмісті або поведінці програми.

SQL-ін'єкції потребують мінімальних навичок і найчастіше не потребують будь-яких спеціальних програм. Зловмисник просто вводить в адресний рядок потрібні значення і отримує доступ до даних, а іноді й адміністративні права на базу даних [2]. За допомогою цих атак зломщик може отримати дані про конфіденційну інформацію. При цьому структура таблиць не порушується, і виявити злом можна тільки в разі зміни (зловмисником) чогось на сайті або в паролі. Існує 5 основних типів SQL ін'єкцій [3]:

1. Класична (In-Band або Union-based) [4]. Найнебезпечніша і найрідкісніша сьогодні атака. Дає змогу відразу отримувати будь-які дані з бази. У цьому методі хакер використовує результати з бази даних і зламує базу даних, щоб досягти мети. Це також називається внутрішньо смуговою ін'єкцією SQL.

2. На основі помилок (Error-based). Дає змогу отримувати інформацію про базу, таблиці та дані на основі виведеного тексту помилки СУБД. У цьому типі впровадження хакер аналізує різні операції і знаходить зразок помилки в базі даних. Потім він/вона отримує доступ до нього, щоб зламати/пошкодити базу.

3. Булева сліпа атака (Boolean-based). Замість отримання всіх даних, зловмисник може поштучно їх перебирати, орієнтуючись на просту відповідь типу true/false.

4. Сліпа атака, заснована на часі (Time-based). Схожа на попередню атаку принципом перебору, маніпулюючи часом відгуку бази. Атакуючі направляють SQL-запит до бази даних, змушуючи її зробити затримку на кілька секунд, перш ніж вона підтвердить або спростує отриманий запит.

5. Поза смугове впровадження (Out-of-Band). Дуже рідкісні та специфічні типи атак, засновані на індивідуальних особливостях баз даних. Така атака відбувається у двох випадках: коли атакуючі не можуть провести атаку і зібрати дані через один і той самий канал зв'язку, або коли сервер працює занадто повільно чи нестабільно, щоб досягти потрібного результату.

Успішна атака на основі SQL-ін'єкції може мати серйозні наслідки для бізнесу, крім розкриття конфіденційних даних, атака також може призвести до отримання зловмисниками адміністративного доступу до системи, з можливістю втручання в її функціонування. Водночас можливе порушення приватності користувачів, включно з розкриттям адрес, номерів телефонів і відомостей про банківські картки, що спричинить фінансові втрати та втрату довіри з боку клієнтів.

Для запобігання SQL-ін’єкціям критично важливо впроваджувати комплексні стратегії безпеки. По-перше, слід використовувати параметризовані запити, що дозволяють передавати значення окремо від запиту та автоматично екранувати спеціальні символи, таким чином запобігаючи впровадженню шкідливого коду. Крім того, важливо проводити валідацію всіх введених даних перед їх використанням у запитах, щоб запобігти можливим атакам. Необхідно також належно керувати привілеями доступу до бази даних для уникнення несанкціонованого доступу та злому системи. Використання об'єктно-реляційних маперів (ORM) може додатково забезпечити захист від SQL-ін'єкцій, оскільки вони автоматично генерують безпечні запити на основі об'єктів моделей даних. Не менш важливим є регулярне оновлення бази даних, програмного забезпечення та бібліотек з метою виправлення вразливостей та недоліків, включаючи потенційні ін'єкції, тим самим забезпечуючи максимальний рівень безпеки веб-додатка.

Однак, додавання додаткових перевірок або постійних перевірок запитів може сповільнити роботу сайту. Це через те, що система буде витрачати

додатковий час на виконання цих перевірок перед обробкою запитів. Таким чином, існує дилема: ми можемо забезпечити високий рівень захисту від SQL-ін'єкцій за рахунок сповільнення роботи сайту, оскільки він постійно проводитиме перевірки та оновлення даних, або ми можемо робити сайт швидким, але менш безпечним, залишаючи деякі ризики.

Прикладом такої ситуації є Denuvo Anti-Tamper[5], яка була розроблена австрійською компанією Denuvo Software Solutions GmbH. Ця технологія використовується для захисту від несанкціонованого доступу до комп'ютерних ігор та програм. Вона запобігає піратству шляхом ускладнення процесу копіювання та модифікації програмного коду. Проте, використання Denuvo може призвести до деякої затримки в роботі гри або програми, оскільки система постійно перевіряє їх автентичність під час виконання. Таким чином, вона впливає на продуктивність, крім того, може спричинити незручність через довгий час завантаження або затримки під час гри, проте забезпечує високий рівень захисту від несанкціонованого доступу.

Отже, SQL-ін'єкції являють собою серйозну загрозу безпеці інформаційних систем, оскільки вони дають змогу зловмисникам несанкціоновано отримувати доступ до баз даних і впливати на дані. Такий тип атак може призвести до втрати конфіденційної інформації, порушення цілісності даних і впливу на доступність системи. Розуміння різних видів цих атак та їхніх можливих наслідків є важливим кроком для запобігання таким атакам і забезпечення безпеки інформаційних систем. Окрім цього, провадження комплексних стратегій безпеки, таких як використання параметризованих запитів, валідація даних і керування привілеями доступу, є критично важливими для запобігання SQL-ін'єкціям і забезпечення безпеки веб-додатків. Проте, додаткові перевірки запитів можуть призвести до сповільнення роботи сайту, створюючи дилему між високим рівнем захисту та продуктивністю.

Список використаних джерел

1. Clarke J. SQL Injection Attacks and Defense. Elsevier Science & Technology Books, 2009. 496 p.
2. Imperva.What is SQL Injection. SQL (Structured query language) Injection, 2023. URL: https://www.imperva.com/learn/application-security/sql-injection-sqli/(дата звернення: 20.01.2024).
3. Ettore Galluccio, Edoardo Caselli, Gabriele Lombari. SQL Injection Strategies. Packt Publishing, 2020. 210 p.
4. Ляшко М. С. Безпека даних в веб-додатках: як захистити від Union-based SQL Injection / М. С. Ляшко, Д. О. В'юхін // Проблеми інформатизації : тези доповідей одинадцятої міжнар. наук.-техн. конф., 16–17 листопада 2023 р. – Баку–Харків–Бельсько-Бяла, 2023. – Т. 2, секція 3,6. – С. 67.
5. Denuvo. The global #1 Games Protection and Anti-Piracy technology, 2023. URL: https://irdeto.com/denuvo/(дата звернення: 05.02.2024)