СПОСОБИ ЗАХИСТУ БАЗ ДАНИХ ВІД SQL ІН'ЄКЦІЙ

Праховнік Н. А., к.т.н., доцент (каф. ОППЦБ КПІ ім. Ігоря Сікорського);
Літвінова Н. О., ст. (гр. ІТ-51, ФІОТ КПІ ім. Ігоря Сікорського)

Анотація. Робота розкриває причини та наслідки виникнення такого виду хакерських атак на бази даних (БД), як SQL ін'єкція. Розглянуто найбільш ефективні методи, винайдені для захисту БД від такого роду атак.

Ключові слова: бази даних, безпека інформаційних систем, SQL ін'єкція, захист від кібератак, конфіденційність інформаційних ресурсів.

Abstract. This work reveals the conditions and consequences of such a kind of hacker attacks on the database (DB), as SQL injection. The most effective invented methods to protect the database from such attacks are also considered here.

Keywords: data bases, security of information systems, sql-injection, cyber attack protection, confidence of information resources.

Вступ. Основою діяльності сучасних компаній будь-якого розміру є конфіденційна інформація різних видів, саме тому безпека даних має надважливе значення для успіху у захисті баз даних.

Аналіз стану питання. Безпека баз даних на сьогоднішній день є найбільш актуальним питанням в сучасних умовах керуванням великими обсягами інформації.

Мета роботи. Метою роботи є розкриття причин і наслідків виникнення такого виду атак на бази даних (БД), як SQL ін'єкція, а також комплексний підхід щодо захисту сучасних сховищ даних від такого роду атак.

Методики та матеріали дослідження. Хакерська атака (кібератака) – спроба реалізації загрози. Тобто, це дії кібер-зловмисників (хакерів) або шкідливих програм, які спрямовані на захоплення інформаційних даних віддаленого комп'ютера, отримання контролю над ресурсами комп'ютера або на виведення системи з ладу.

SQL ін'єкція – один з найпоширеніших способів кібератак на сайти та програми, що працюють з базами даних, заснований на впровадженні в запит довільного SQL-коду. Вони визнані одними з найстаріших і найбільш небезпечних атак для веб-додатків [1].

SQL – це мова запитів, призначена для управління даними, що зберігаються в реляційних базах даних. Її можна використовувати для доступу, редагування та видалення даних. Багато веб-додатків і веб-сайтів зберігають всі дані в базах даних SQL таких як, наприклад, MySQL, Oracle, SQL Server тощо. У деяких випадках можна також використовувати команди SQL для запуску команд операційної системи. Тому успішна атака SQL Injection може мати дуже серйозні наслідки:

– зловмисники можуть використовувати ін'єкції SQL, щоб отримати облікові дані інших користувачів бази даних. Потім вони можуть видати себе за цих користувачів. В найгіршому випадку, такий користувач може «виявитися» адміністратором бази даних з усіма привілеями до бази даних.

– SQL дозволяє вибирати та виводити дані з бази даних. Уразливість SQL ін'єкцій може дозволити зловмиснику отримати повний доступ до всіх даних на сервері баз даних, включаючи особисту інформацію, паролі, комерційні дані та інтелектуальну власність.

– SQL також дозволяє змінювати дані в базі даних і додавати нові дані. Наприклад, у фінансовій програмі зловмисник може використовувати SQL ін’єкцію для зміни залишків, анулювання транзакцій або переказу коштів на свій рахунок.

– ви можете використовувати SQL для видалення записів з бази даних, навіть відкидання таблиць. Навіть якщо ви робити резервні копії баз даних, видалення даних може вплинути на доступність програми до відновлення бази даних. Крім того, резервні копії можуть не охоплювати останні дані.

– на деяких серверах баз даних можна отримати доступ до операційної системи, використовуючи сервер баз даних. Це може бути навмисно або випадково. У такому випадку зловмисник може використовувати SQL ін’єкцію як початковий вектор, а потім атакувати внутрішню мережу за брандмауером [2].

Прихованою небезпекою тут є те, що SQL запити можуть контролювати сервер БД, про що ваша програма і не дізнається.

Згідно з даними, взятими з останньої статистики найпопулярнішої системи керування веб-сайтами – Wordpress, кількість випадків SQL ін'єкцій налічує 51% від усіх здійснених хакерських атак. Інші ж джерела стверджують, що цей вид атак посідає почесне друге місце серед атак на веб-додатки.

Не зважаючи на таку статистику, багато розробники навіть не враховують таку загрозу і не зважають на те, що SQL-запити можуть бути підроблені. Насправді такі запити, залежно від типу СКБД та умов впровадження, можуть обійти обмеження доступу, стандартну перевірку авторизації та аутентифікації чи дати можливість зловмиснику виконати довільний запит до бази даних (наприклад, прочитати вміст будь-яких конфіденційних даних, видалити, змінити або додати їх), отримати можливість читання та/або запису локальних файлів та виконання довільних команд на сервері.

Зловмисники можуть використовувати уразливості SQL ін’єкцій для різних цілей, проте основною причиною виникнення цього «слабкого місця» є зазвичай те, що вхідні дані від користувача не фільтруються (відсутня обробка спеціальних символів, перевірка типів тощо). Розглянемо для наочності наступні приклади.

Маємо форму пошуку товару. У якості параметрів пошуку є назва товару та верхня границя його ціни.

Приклад 1: У якості ціни користувач замість цифри «100» написав (не)навмисно «100’» (зверніть уваги на символ апострофа). Пошук за таким критерієм поверне непередбачену програмою помилку виконання. Отже, в даному випадку негативний наслідок – погіршення відмовостійкості та працездатності системи.

Приклад 2: У назву товару користувач включить свій SQL код, тобто основний запит на пошук буде містити внутрішній «злочинний» запит. Таким чином, крім виконання основної операції, передбаченої бізнес-логікою програми, виконуються інші несанкціоновані операції, які можуть порушити цілісність та конфіденційність даних, що зберігаються у БД, або навантажити БД так, що інші користувачі в кращому випадку будуть змушені довго очікувати виконання їх запитів [3].

На практиці не завжди можливо провести SQL ін'єкцію за допомогою полів для вводу даних, тому для веб-сайтів, які передають дані через GET-запити, наявний ще один випадок уразливості – редагування адресного рядку браузера. Наприклад, за допомогою SQL ін'єкції у адресному рядку хакер може зчитати дані з будь-якої таблиці, використовуючи оператор UNION, наявний у більшості БД.

Для уникнення потенційних небезпек від такого виду загроз розробник програмних продуктів, що працюють з базами даних, повинен знати про таку уразливість і вживати заходів протидії впровадженню SQL [4].

Наразі до ефективних способів вирішення цієї проблеми, які успішно застосовуються для захисту від SQL ін’єкцій, належать наступні:

– Використання параметризованих запитів. Багато серверів надають можливість відправки програмою параметризованих запитів (підготовлені вирази) до БД. Таким чином параметри зовнішнього походження відправляються на сервер окремо від самого запиту або автоматично екрануються клієнтською бібліотекою, що дозволяє автоматично уникнути шкідливих значень в параметрах.

– Фільтрація рядкових параметрів. Щоб впровадження коду було неможливо, для деяких СКБД, потрібно брати в лапки всі рядкові параметри. Так, у самому параметрі треба екранувати спецсимволи, тобто замінити лапки на \", апостроф на \', а зворотну косу риску на \\.

– Фільтрація цілочисельних параметрів. У випадку, коли параметром виступає цілочисельне значення, допомагає перевірка значення параметру на тип – якщо змінна не є числом, запит взагалі не повинен виконуватися.

– Усікання вхідних параметрів. Для внесення змін в логіку виконання SQL-запиту потрібно впровадження достатньо довгих рядків. Так, мінімальна довжина такого рядка найчастіше становить 8 символів («1 OR 1=1»). Якщо максимальна довжина коректного значення параметра невелика, то одним з методів захисту може бути максимальне усікання значень вхідних параметрів.

– Зменшення кількості динамічних запитів. Також не варто забувати про запобіжні заходи, які ніколи не завадять для того, щоб захистити інформацію, що

зберігається у БД. Тож, крім, контролю самих параметрів, розробник завжди має пам’ятати про обмеження доступу до системних таблиць, коректну обробку помилок та зберігання паролів у зашифрованому вигляді.

Література

1. Justin C. SQL Injection Attacks and Defense. – Syngress Date, 2009.
2. Litchfield D, Anley C., Heasman J, Grindlay B. The Database Hacker’s Handbook: Defending Database Servers. – Wiley Date, 2005.
3. Марков А. С., Миронов С. В., Цирлов В. Л. Выявление уязвимостей в программном коде // Открытые системы. 2005, № 12.
4. Halde J. Basics of SQL Injection Analysis, Detection and Prevention. – LAP LAMBERT Academic Publishing, 2014.