

ОГЛЯД МЕТОДІВ БОРОТЬБИ З SQL-ІН'ЄКЦІЯМИ

SQL-ін'єкція – це вразливість, яка дозволяє маніпулювати даними з бази даних (читати, додавати, змінювати, записувати) за допомогою мови SQL. Дана ін'єкція може призвести до витоку конфіденційної інформації, втрати даних або крадіжки грошей. Найбільш вразливими до таких SQL-ін'єкцій є вебзастосунки, а також інші програми, які працюють з базами даних [1].

Перша задокументована згадка про SQL-ін'єкції була записана в 1998 році дослідником кібербезпеки, Джеффом Форрісталом у журналі Phrack [2]. У його статті [3] описано метод отримання конфіденційної інформації за допомогою SQL-ін'єкції з баз даних незахищеного вебсайту. Проблема лишається актуальною і зараз. За даними компанії OWASP, що займається захистом вебзастосунків, однією з найпоширеніших атак на програмне забезпечення є SQL-ін'єкція [4]. Це підтверджується прикладами компаній, які зазнали SQL-атаки: 7-Eleven (2007 рік), MySpace (2008), Equifax (2017), Vtech (2015) [2].

Для ефективного захисту від SQL-ін'єкцій існують методи, які наведені нижче. Один з загальних методів – це використання параметризованих запитів. Параметризований запит - це запит до бази даних, в якому дані передаються як параметри, а не вбудовуються в сам запит. Параметри захищаються від SQL-ін'єкції, оскільки будь-який SQL-код, який був введений в якості параметру, буде розглядатися як дані, а не як частина запиту до бази даних [5].

У випадку, якщо не можна використовувати параметризовані запити, то найкращим підходом є забезпечення належного екранування спеціальних символів рядка у вхідних параметрах. Екранування входів - це захисний механізм, який дозволяє запобігти SQL-ін'єкції, коли зловмисник використовує спеціально сформований SQL-запит з метою отримати несанкціонований доступ до бази даних або змінити її вміст. У процесі екранування входів всі вхідні дані, які можуть містити шкідливі символи, будуть видозмінені. Наприклад, якщо програма отримує від користувача його ім'я, то перед тим, як використовувати його в SQL-запиті, необхідно прибрати спеціальні символи, такі як одинарні лапки, які можуть бути використані для введення шкідливого коду [5].

Одним з інструментів запобігання SQL-ін'єкціям є ORM(об'єктно-реляційне відображення/перетворення). ORM – це інструмент, який конвертує SQL-команди в код мови програмування і навпаки, однак використання ORM не робить програмний продукт автоматично захищеним від SQL-ін'єкції [5].

Метод Sanitizing in puts (санітизація входів) застосовує процес валідації вхідних даних, щоб переконатися, що вони відповідають потрібному формату

або типу даних. Наприклад, перевірка поля електронної адреси на наявність обов'язкових даних або видалення пробілів та символів нового рядка, де вони недоречні.

Атаки SQL-ін'єкції є серйозною загрозою кібербезпеці. Тому застосування методів боротьби з SQL-ін'єкціями дозволить програмістам зробити свій продукт більш захищеним та ефективним. Треба зазначити, що використання якогось лише одного методу не зможе повністю захистити програмний продукт. Для більшої ефективності варто комбінувати наведені методи.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. WhatIs SQL Injection? Identification & Prevention Tips [Electronic recourse]. – Access mode : <https://www.varonis.com/blog/what-is-sql-injection>.
2. What is the history of SQL injections? [Electronic recourse]. – Access mode : <https://www.malwarebytes.com/sql-injection#:~:text=The%20SQL%20injection%20exploit%20was,researcher%20and%20hacker%20Jeff%20Forristal>.
3. The Internet Explained, Part III: The SQL Tutorial [Electronic recourse]. – Access mode : <http://phrack.org/issues/54/8.html#article>
4. OWASP Top 10:2021 [Electronic recourse]. - Access mode : <https://owasp.org/Top10/>
5. Protecting against sqlinjection [Electronic recourse]. - Access mode : <https://www.hacksplaining.com/prevention/sql-injection>.