DOI 10.6084/m9.figshare.22886720
Виклики та проблеми сучасної науки, Том 1 (2023)

Вразливості SQL-injection

Світлана Володимирівна Клименко
ORCID: https://orcid.org/0000-0003-2005-9993
Дніпровський національний університет імені Олеся Гончара, Дніпро

Ілля Юрійович Тремба
ORCID: https://orcid.org/0009-0004-0223-1176
Дніпровський національний університет імені Олеся Гончара, Дніпро

Вступ

SQL-injection (SQLi) це ін'єкційна атака, яка дозволяє втручатися в запити, які програма робить до своєї бази даних, і як зрозуміло з назви, запити відбуваються через стандартизовану мову запитів SQL. Часто це дозволяє зловмиснику переглядати дані, які вони зазвичай не можуть отримати такі як дані, що належать іншим користувачам, або будь-які інші дані, до яких має доступ сама програма. Незважаючи на те, що є інші типи ін'єкційних вразливостей (ін'єкція заголовка HTTP, ін'єкція коду, команди т. п.) SQLi є більш відомою та привабливою для використання зловмисниками при атаках на веб-додатки. Трохи нижче ми розглянемо статистику вразливостей та наслідки реалізації SQLi [1].

Уразливість SQLi може вплинути на будь-який веб-сайт або веб-програму, яка використовує базу даних SQL, таку як MySQL, Oracle, SQL Server або інші. SQL – це мова запитів, які керують даними, котрі зберігаються в реляційних базах даних, її можна використати для доступу, видалення або зміни даних. У деяких випадках можливо використати команди SQL для запуску команд операційної системи, а також враховуючи, що частина веб-сайтів та веб-додатків, які зберігають усі дані в базах даних SQL досить велика, то успішна атака SQLi може мати дуже серйозні наслідки.

Спектр використання ін'єкції досить широкий. Злочинці можуть використовувати цю уразливість для отримання несанкціонованого доступу до ваших конфіденційних даних: інформації про клієнтів, особистих даних, комерційної таємниці, інтелектуальної власності тощо. Отже, успішна реалізація SQLi може порушити конфіденційність (приватні дані користувачів або компанії), цілісність (внесення змін до системи або видалення інформації з неї), автентифікацію (можливе підключення до системи як інший користувач без попереднього знання паролю), авторизацію (зміна інформації про авторизацію, якщо вона зберігається в базі даних SQL).

Типи впровадження SQL-injection

1. Найпоширеніший типом атаки є ***Внутрішньо-смугове впровадження SQLi***: зловмисник використовує той самий канал зв'язку для атаки та збору результатів. Методи на основі внутрішньо-смугового впровадження:

– SQLi *на основі помилок*: зловмисник використовує команду SQL для отримання повідомлення про помилку з сервера бази даних, з цього повідомлення вилучається інформація про структуру баз даних

– SQLi *на основі оператора об'єднання* (UNION BASED): є найпоширенішим типом SQL-ін'єкції. За цією методикою зловмисник використовує SQL оператор UNION для об'єднання операторів SELECT і повернути лише одну відповідь HTTP.

2. ***Сліпа SQLi***: база-даних веб-сайту не передає дані зловмиснику, але він може дізнатися про структуру сервера, надсилаючи корисні дані та спостерігаючи за відповіддю. Із-за більш довгого часу реалізації є менш поширеними за внутрішньо-смугове впровадження SQLi. Методи на основі сліпої SQLi:

– *Ін'єкція на основі часу:* зловмисник надсилає SQL-запит до бази даних, змушуючи базу даних чекати певну кількість секунд перед тим як відповісти. Виходячи з кількості секунд, що минули до відповіді, визначається чи істинний результат. Наприклад, зловмисник використовує SQL-запит, який дає команду на затримку, якщо перша літера імені бази даних – С, якщо відповідь затримується – результат істинний.

– *Булева ін'єкція*: до бази даних надсилається SQL-запит, зловмисник вважає результат істинними, якщо було змінено інформацію у відповіді HTTP.

3. Найменш поширений типом атаки є ***Позасмугове впровадження SQLi***: зловмисник використовує різні канали для збору результатів та для атаки. Цей метод використовують, якщо сервер нестабільний або надто повільний.

Приклад простої реалізації – SQLi на основі помилок

Маємо сайт на якому увімкнена функція відображення помилок бази даних на сайті. Спочатку потрібно визначити скільки стовпців повертається з вихідного запиту, для цього використовуємо запит (' ORDER BY 1 --), де 1 – індекс стовпця; далі збільшуємо вказаний індекс доки не станеться помилка типу: *The ORDER BY position number 4 is out of range of the number of items in the select list.*

З цього виходить, що кількість стовпців дорівнює трьом. Наступний крок буде визначення стовпців із корисним типом даних. Наприклад, визначаємо чи містить стовпець рядкові дані запитом SELECT ‘o’ (визначається чи міститься символ ‘o’ у стовпці)

' UNION SELECT 'o',NULL,NULL--

' UNION SELECT NULL,'o',NULL--

' UNION SELECT NULL,NULL,'o'—

Якщо тип даних у стовпці не сумісний із рядковими даними, то запит спричинить помилку типу: *Conversion failed when converting the varchar value 'o' to data type int.*

В інакшому випадку, коли помилка не виникає, а відповідь містить деякий додатковий вміст (включаючи введене значення рядка), то цей стовпець підходить для отримання даних.

Далі, якщо ми визначили, який стовпець вразливий, наприклад другий, ми надсилаємо запит типу: *' UNION SELECT 1,database(),3 –*

З якого можна отримати назву бази даних, наприклад ‘authorization_data’. Потім запитом *' UNION SELECT 1,table_name,3 from INFORMATION_SCHEMA.TABLES WHERE TABLE_SCHEMA= 'authorization_data'* -- визначаємо назву таблиць. І так далі, використовуючи синтаксис SQL поступово визначаємо назви стовпців і виводимо запитами із них дані. Це найпростіший приклад реалізації SQLi.

ІСТОРИЧНИЙ ТА ХРОНОЛОГІЧНИЙ ОГЛЯД

Вище розглянута загальна суть та типи SQLi, тепер роздивимось як вона розвивалась та наскільки актуальною є загроза зараз. Враховуючи те, що SQL розроблена на початку 70-х років, то вже в 90-х були відомі перші ін’єкції. Перший експлойт SQLi був задокументований у 1998 році дослідником кібербезпеки та хакером Джеффом Форрісталом, він пояснив, що навіть хтось із базовими навичками кодування може підключити неавторизовані команди SQL до законних команд SQL і отримати конфіденційну інформацію з бази даних незахищеного веб-сайту. Пізніше, він повідомив компанію Microsoft про те як ця уразливість вплинула на їх SQL Server, але їх реакція була досить безтурботна і вони не придали цьому велике значення [2].

У 2007 році сталося перша велика атака SQLi - російські хакери використали SQL-ін’єкції, щоб зламати веб-сайт 7-Eleven (найбільша мережа міні-маркетів у США), через цей злом вони дісталися до дебетових карток клієнтів, які зберігалися у базі даних, пізніше готівка була виведена ними до Росії. Втрати готівки – 2 мільйона доларів.

У 2008 році відбулася атака на MySpace за якої було викрадено електронні листи, імена та часткові паролі майже 360 мільйонів

облікових записів.

У 2017 році при атаці на Equifax були отримані надзвичайно особисту інформацію (імена, номери соціального страхування, дати народження та адреси) для 143 мільйонів споживачів. Перед тим, як відбулася атака, дослідницька фірма з кібербезпеки навіть попередила Equifax, що вони чутливі до атаки SQLi, але кредитне бюро не вжило жодних заходів, доки не стало надто пізно.

Відомі інші жертви злому з допомогою SQLi такі як: Epic Games, TalkTalk, LinkedIn, Target, Yahoo, Zappos і Sony Pictures.

SQL-INJECTION ПОСЕРЕД ІНШИХ АТАК

Навіть незважаючи на це, все рівно SQLi не виглядає великою проблемою, бо й інші типи атак мають не менші наслідки та небезпеки, деякі навіть є більш небезпечними такі як DDoS, трояни тощо. Але SQLi потребують лише незначних технічних навичок, щоб досягти результатів, від пошуку вразливого сайту до виконання та безпечного вилучення файлів або даних.

Також, за даними сайту Malwarebytes Labs, SQLi посіла на третє місце в топ-5 найдурніших кіберзагроз, які все одно працюють посилаючись на той факт, що SQLi — це відома, передбачувана атака з простими контрзаходами. Атаки SQLi настільки прості, що фактично зловмисники можуть знайти вразливі веб-сайти за допомогою розширеного пошуку Google [3]. Одне дослідження Інституту Ponemon про загрозу впровадження SQL і недавні порушення роздрібної торгівлі показало, що 65% опитаних компаній були жертвами атак на основі SQLi (хоча дослідження 2014 року, дані все рівно вражають) [4].

За стандартним документом Top-10 OWASP, який представляє найбільш критичні ризики для безпеки веб-додатків за 2021 рік ін'єкції посідають третє місце (рис. 1), а в 2017 посідали перше місце так як і в 2013 році [2].

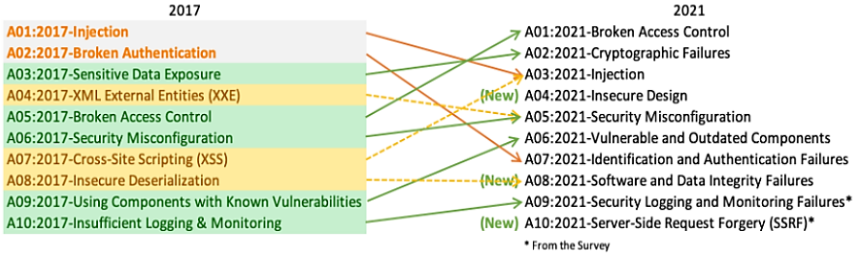

Рисунок 1- 10 найбільших ризиків для безпеки веб-додатків 2017 та 2021 років [1]

Як запобігти SQL-injection

Конкретні методи запобігання залежать від підтипу вразливості SQLi, механізму бази даних SQL і мови програмування, тому запобігти вразливості SQLi нелегко. Єдиний надійний спосіб запобігти атакам SQLi – це перевірка введених даних та параметризовані запити, включаючи підготовлені оператори. Код програми ніколи не повинен використовувати вхідні дані безпосередньо. Розробник повинен дезінфікувати всі введені дані, а не лише введені веб-форми, такі як форми входу. Потрібно видалити потенційно шкідливі елементи коду, такі як одинарні лапки. Також доцільно вимкнути видимість помилок бази даних на сайтах. Помилки бази даних можна використовувати за допомогою SQLi для отримання інформації про базу даних.

Висновок

Вразливість SQLi існує вже більше двадцяти років і судячи по всьому буде існувати і далі, вона є однією із пріоритетних загроз для веб-додатків, має пластичні методи та підходи до отримання даних з бази даних, при цьому є простою в реалізації. І враховуючи те, що від неї нелегко захиститися, SQLi однозначно це те на що треба звернути увагу при активному розвитку мережевих технологій.

Посилання

1. OWASP Top Ten | OWASP Foundation. *OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation*. URL: https://owasp.org/www-project-top-ten/ (дата звернення: 21.05.2023).

2. What is SQL Injection (SQLi) and How to Prevent Attacks. *Acunetix*. URL. https://www.acunetix.com/websitesecurity/sql-injection/ (дата звернення: 21.05.2023).

3. The SQL Injection Threat & Recent Retail Breaches : Ponemon Institute. *Ponemon Institute*. URL: https://www.ponemon.org/research/ponemon-library/security/the-sql-injection-threat-recent-retail-breaches.html (дата звернення: 21.05.2023).

4. What is SQL injection - Examples & prevention | Malwarebytes. *Malwarebytes*. URL: https://www.malwarebytes.com/sql-injection (дата звернення: 21.05.2023).