УДК 004.056.5:004.738 WWW
DOI: 10.31673/2409-7292.2023.030008

Федоренко А. А., Осадчий Б. І., Коржик В. В.

АНАЛІЗ МЕТОДІВ ВИЯВЛЕННЯ ВРАЗЛИВОСТЕЙ WEB-РЕСУРСІВ ДО SQL-ІН'ЄКЦІЙ

Ця стаття розглядає проблему вразливостей до SQL-ін'єкцій у веб-додатках та використання автоматизованих сканерів для виявлення цих вразливостей. Вона починається з опису SQL-ін'єкцій та їх наслідків, а також ручної перевірки на вразливості. Далі стаття аналізує різні автоматизовані сканери вразливостей, включаючи Acunetix, Burp Suite, Nessus, OpenVAS, SQLMap, OWASP ZAP та Nikto. Для кожного сканера наведені переваги та недоліки, а також рівень деталізації та функціональні можливості. Стаття закінчується висновками, які підкреслюють важливість розуміння ризиків SQL-ін'єкцій та використання правильних інструментів для їх виявлення. Наголошується на тому, що автоматизовані сканери не є універсальним рішенням, і вони повинні супроводжуватися ручною перевіркою та аналізом. Стаття вказує на необхідність постійного оновлення сканерів та комбінації автоматизованих та ручних методів для забезпечення найвищого рівня безпеки. Вона надає читачам корисний огляд різних аспектів та аспектів використання автоматизованих сканерів вразливостей до SQL-ін'єкцій у веб-додатках.

Ключові слова: SQL-ін'єкція, Web-ресурс, вразливість, сканер вразливостей, Web-додаток, кібербезпека.

Вступ

В даний час web-додатки займають важливе місце в нашому житті, адже вони використовуються в різних сферах, від онлайн-торгівлі до банківських послуг та соціальних мереж. З огляду на таку широку застосовність, зростає необхідність у забезпеченні безпеки web-додатків. Однією з основних загроз є SQL-ін'єкції, які можуть призвести до доступу до конфіденційної інформації та порушення її цілісності.

Огляд літератури

У літературі було знайдено багато робіт, присвячених виявленню вразливостей web-додатків до SQL-ін'єкцій. Були розглянуті різні методи, що використовуються для виявлення SQL-ін'єкцій, такі як ручна перевірка, автоматизовані сканери вразливостей та системи виявлення вразливостей. Також було проаналізовано роботу деяких автоматизованих сканерів, так і як Burp Suite, Acunetix, AppScan, OpenVAS, та інші. Крім того, було досліджено підходи до захисту від SQL-ін'єкцій, такі як використання параметризованих запитів, фільтрація вхідних даних, використання stored procedures та інші.

Нижче наведено огляд деяких наукових статей щодо виявлення SQL-ін'єкцій. Так стаття [1] досліджує різні методи виявлення та запобігання SQL-ін'єкцій, включаючи методи на основі підписів, методи на основі машинного навчання та методи на основі аналізу поведінки. Стаття [2] порівнює різні методи виявлення та запобігання SQL-ін'єкцій, включаючи методи на основі підписів, методи на основі машинного навчання та методи на основі аналізу поведінки. Стаття [3] пропонує новий метод виявлення та запобігання SQL-ін'єкцій на основі аналізу поведінки користувачів. Стаття [4] досліджує різні методи виявлення та запобігання SQL-ін'єкцій, включаючи методи на основі підписів, методи на основі машинного навчання та методи на основі аналізу поведінки. Стаття [5] порівнює різні методи виявлення SQL-ін'єкцій, включаючи методи на основі підписів, методи на основі машинного навчання та методи на основі аналізу поведінки.

Мета дослідження. Метою даної статті є розгляд методів виявлення вразливостей web-ресурсів до SQL-ін'єкцій.

Завдання дослідження. Для досягнення мети дослідження були визначені наступні завдання:

проаналізувати різні методи виявлення вразливостей web-ресурсів до SQL-ін'єкцій;

дослідити роботу автоматизованих сканерів вразливостей та їх можливості виявлення SQL-ін'єкцій.

Структура статті. У першому розділі статті будуть введені поняття вразливості web-ресурсів до SQL-ін'єкцій та їх наслідки. Далі будуть розглянуті різні методи виявлення

вразливостей, такі як ручна перевірка та автоматизовані сканери вразливостей, а також їх переваги та недоліки. На закінчення розділу будуть наведені приклади використання деяких сканерів вразливостей для виявлення SQL-ін'єкцій. Отже, дана стаття присвячена дослідженню методів виявлення вразливостей web-ресурсів до SQL-ін'єкцій. У наступних розділах будуть розглянуті детальніше різні підходи до виявлення вразливостей та їх ефективність в реальних умовах.

Поняття вразливості web-ресурсів до SQL-ін'єкцій та їх наслідки

Вразливість web-ресурсів до SQL-ін'єкцій виникає, коли додаток не належним чином перевіряє та обробляє вхідні дані, які передаються до бази даних. SQL-ін'єкція є методом атаки, при якому зловмисник використовує некоректно оброблені вхідні дані для виконання зловмисного SQL-коду в базі даних.

Припустимо, у веб-додатку є форма авторизації, де користувачі вводять своє ім'я користувача та пароль. Дані, введені користувачами, використовуються для створення SQL-запиту для перевірки авторизації:

```
SELECT * FROM users WHERE username = '[введений_логін]' AND password = '[введений_пароль]'
```

Проте, якщо додаток не належним чином перевіряє та обробляє вхідні дані, зловмисник може використати це для впровадження SQL-ін'єкції. Наприклад, якщо зловмисник у поле "ім'я користувача" вводить наступне:

```
' OR '1'='1
```

Тоді сформований SQL-запит буде виглядати наступним чином:

```
SELECT * FROM users WHERE username = '' OR '1'='1' AND password = '[введений_пароль]'
```

В результаті такого SQL-запиту умова '1'='1' завжди буде істинною, і зловмисник може отримати доступ до облікових записів користувачів, навіть без правильного введення пароля. Це лише простий приклад, але демонструє, як некоректна обробка вхідних даних може викликати SQL-ін'єкцію та призвести до небезпеки для безпеки системи.

Наслідки SQL-ін'єкцій можуть бути серйозними та включати:

1. Отримання несанкціонованого доступу до бази даних: Зловмисник може виконувати різноманітні дії з базою даних, такі як читання, зміна або видалення даних. Це може призвести до розкриття конфіденційної інформації, такої як особисті дані користувачів, паролі, фінансова інформація тощо.
2. Пошкодження бази даних: Зловмисник може виконати шкідливі операції, такі як видалення або зміна таблиць, що може призвести до недоступності або пошкодження даних.
3. Виконання вторгнення: Зловмисник може використовувати SQL-ін'єкцію для виконання додаткових атак, таких як віддалене виконання коду, включення зловмисного програмного забезпечення або вторгнення в інші системи, які взаємодіють з базою даних.
4. Втрата довіри користувачів: Коли користувачі виявляють, що їхні дані були скомпрометовані через SQL-ін'єкцію, це може призвести до втрати довіри до ресурсу та його власника.
5. Отже, вразливість до SQL-ін'єкцій може мати серйозні наслідки для безпеки та цілісності web-ресурсу та бази даних, тому важливо приділити належну увагу безпекі.

Ручна перевірка на вразливість до SQL-ін'єкцій

Ручна перевірка на вразливість до SQL-ін'єкцій є першим кроком у виявленні вразливостей веб-ресурсів. Вона полягає у введенні різноманітних значень в поля вводу на

сайті та спостереженні за результатами запитів до бази даних. Якщо запит відповідає нашому очікуванню, то веб-сайт не є вразливим до SQL-ін'єкцій. Але, якщо результат запиту містить непередбачувану відповідь, можна припустити, що сайт має вразливість до SQL-ін'єкцій.

Ручна перевірка на вразливість до SQL-ін'єкцій вимагає уважності та ретельної перевірки веб-додатку або коду, який обробляє вхідні дані. Ось кілька кроків, які можна вжити для ручної перевірки на наявність вразливості до SQL-ін'єкцій:

1. Відкриття форм та взаємодія з ними: Ретельно перевірте всі форми та інтерактивні елементи на вашому веб-ресурсі, де користувачі можуть вводити дані. Зверніть увагу на поля, які передаються до бази даних, такі як поля пошуку, логіна, пароля або коментарі.
2. Введення спеціальних символів: Введіть спеціальні символи, такі як одинарні або подвійні лапки, крапки з комою, косі риски та інші спеціальні символи, які можуть використовуватись в SQL-запитах.
3. Перевірка реакції додатку: Спостерігайте за реакцією веб-додатку на ваші введені дані. Якщо ви помічаєте незвичну поведінку, помилки або неправильне відображення даних, це може свідчити про наявність вразливості до SQL-ін'єкцій.
4. Введення спеціальних рядків: Спробуйте ввести спеціальні рядки, які можуть мати вплив на структуру SQL-запиту. Наприклад, введіть `' OR '1'='1 або '; DROP TABLE users; --`.
5. Виведення результатів: Спостерігайте, чи відображаються введені вами дані безпосередньо на веб-сторінці або в повідомленнях про помилку. Це може свідчити про вразливість до SQL-ін'єкцій.
6. Тестування додаткових сценаріїв:

спробуйте ввести спеціальні символи або комбінації, які можуть вплинути на синтаксис SQL-запиту. Наприклад, `' OR '1'='1'; --`;

введіть значення, яке містить SQL-коментарі, наприклад, `' UNION ALL SELECT 1, username FROM users; --`. Перевірте, чи відображаються дані з інших таблиць або поля, до яких ви не повинні мати доступу;

спробуйте ввести довжину великої послідовності символів, щоб перевірити, чи виникають помилки, що можуть свідчити про переповнення або недостатню обробку даних.

7. Перевірка вхідних обмежень: Перевірте, як додаток обробляє неправильні або недопустимі вхідні дані. Спробуйте ввести недійсні значення або спеціальні символи, щоб побачити, як додаток реагує на них. Якщо додаток не відкидає або екранує неправильні дані, це може свідчити про наявність вразливості.
8. Аналіз помилок: Уважно прочитайте повідомлення про помилку, якщо такі виникають. Вони можуть містити інформацію про виконання SQL-запиту або викривати деталі бази даних, що можуть бути використані зловмисником.

Ці кроки є загальними рекомендаціями, які можна використовувати для ручної перевірки на вразливість до SQL-ін'єкцій. Проте, важливо пам'ятати, що професійна безпекова перевірка та аудит додатків повинні бути проведені досвідченими фахівцями з безпеки, які використовують спеціалізовані інструменти та методики. Незважаючи на те, що ручна перевірка може зайняти багато часу та не завжди буде ефективною, вона є корисним інструментом у виявленні вразливостей та допомагає збільшити розуміння структури бази даних.

Автоматизовані сканери вразливостей

Автоматизовані сканери вразливостей є потужним інструментом для виявлення вразливостей веб-ресурсів. Вони дозволяють автоматизувати процес пошуку вразливостей та зекономити час та зусилля. Сканери використовують різні методи, такі як тестування вводу, тестування виводу, тестування на зміну запиту та тестування на множинне виконання запитів.

Крім того, сканери вразливостей можуть детектувати вразливості, які складно виявити ручним способом, такі як "time-based" або "blind" SQL-ін'єкції. Однак, не всі сканери

вразливостей є однаково ефективними та надійними. Деякі сканери можуть давати багато фальшивої інформації, тоді як інші можуть працювати надійно.

Існує багато автоматизованих сканерів вразливостей, які можуть допомогти виявити потенційні SQL-ін'єкції в веб-додатках. Ось декілька популярних сканерів:

1. Acunetix: Acunetix є потужним сканером вразливостей, який може виявити SQL-ін'єкції та інші безпекові проблеми. Він має широкий набір функцій, таких як автоматична ідентифікація параметрів запиту, виявлення логічних помилок та спеціальних символів, інтеграція з базами даних та інші.
2. Burp Suite: Burp Suite є популярним набором інструментів для тестування на проникнення, який також має можливості виявлення SQL-ін'єкцій. Він надає можливість перехоплення та зміни HTTP-запитів, що дозволяє аналізувати взаємодію з базою даних та виявляти потенційні вразливості.
3. Nessus: Nessus є широко використовуваним сканером вразливостей, який може виявляти різноманітні безпекові проблеми, включаючи SQL-ін'єкції. Він пропонує автоматичне сканування веб-додатків для пошуку вразливих точок взаємодії з базою даних.
4. OpenVAS: OpenVAS (Open Vulnerability Assessment System) є інструментом сканування вразливостей з відкритим кодом, який може виявити SQL-ін'єкції та інші безпекові проблеми. Він забезпечує широкий спектр функцій тестування на проникнення та вміє проводити детальний аналіз веб-додатків.
5. SQLMap: SQLMap є спеціалізованим інструментом для автоматизованого виявлення та експлуатації SQL-ін'єкцій.
6. OWASP ZAP: OWASP ZAP (Zed Attack Proxy) є іншим популярним інструментом для тестування на проникнення та виявлення вразливостей, включаючи SQL-ін'єкції. Він надає широкий спектр функцій, таких як перехоплення та зміна HTTP-запитів, автоматичне виявлення вразливостей та аналіз безпеки веб-додатків.
7. Nikto: Nikto є простим, але ефективним інструментом сканування веб-сайтів на наявність вразливостей, включаючи SQL-ін'єкції. Він аналізує веб-сайт на наявність вразливих паттернів та поведінки, що може вказувати на потенційні проблеми безпеки.

Ці інструменти допомагають автоматизувати процес виявлення SQL-ін'єкцій та інших вразливостей у веб-додатках. Вони забезпечують різні функціональні можливості та рівень деталізації, що дозволяє проводити широкий аналіз безпеки веб-додатків. Проте, варто пам'ятати, що автоматизовані сканери не завжди здатні знайти всі можливі вразливості, і ручна перевірка та аудит безпеки також є необхідними для забезпечення повної безпеки веб-додатків.

У таблиці 1 наведено переваги та недоліки розглянутих автоматизованих сканерів вразливостей до SQL-ін'єкцій. Важливо зазначити, що ця таблиця надає загальну оцінку переваг і недоліків розглянутих автоматизованих сканерів і не враховує всіх індивідуальних варіантів використання та специфічних вимог. Кожен сканер має свої особливості, і їх ефективність може залежати від конкретного сценарію тестування та налаштувань.

При виборі автоматизованого сканера важливо враховувати такі фактори, як:

1) Функціонал та можливості сканера, що відповідають потребам тестування безпеки.
2) Відкритий код або комерційна ліцензія та їх вартість.
3) Швидкість та надійність виявлення вразливостей.
4) Підтримка технічних вимог, таких як платформи та системи, типи веб-додатків та баз даних.
5) Рівень гнучкості та можливість налаштування сканера під потреби.
6) Підтримка та інтеграція з іншими інструментами безпеки.

Крім автоматизованих сканерів, варто враховувати, що ручна перевірка та аудит безпеки додатків є необхідними для забезпечення повної безпеки. Комбінація автоматизованих сканерів і ручного аналізу може забезпечити більш точну та глибоку оцінку безпеки веб-додатків.

Таблиця 1

Порівняльний аналіз автоматизованих сканерів вразливостей

Автоматизований сканер	**Переваги**	**Недоліки**
Acunetix	Потужність та широкий функціонал. Добре визначає вразливості в SQL-ін'єкціях.	Вартість ліцензії для повної функціональності. Вимагає значних ресурсів для запуску.
Burp Suite	Гнучкість та можливість вручну змінювати запити. Інтеграція з іншими інструментами.	Вимагає досвіду для оптимального використання.
Nessus	Виявлення широкого спектру вразливостей. Підтримка різних платформ і систем.	Потребує налаштування для точного сканування.
OpenVAS	Відкритий код та безкоштовна ліцензія. Гнучкість та можливість налаштування.	Менша швидкість та надійність порівняно з комерційними рішеннями.
SQLMap	Спеціалізований для виявлення SQL-ін'єкцій. Підтримка різних типів SQL-ін'єкцій.	Може давати багато фальшиво-позитивних результатів.
OWASP ZAP	Відкритий код та безкоштовна ліцензія. Широкий спектр функцій для аналізу безпеки.	Менша швидкість та надійність порівняно з комерційними рішеннями.
Nikto	Простий у використанні та налаштуванні. Швидкість сканування.	Менша точність порівняно з іншими сканерами.

Висновки

1. SQL-ін'єкція є серйозною загрозою для безпеки веб-додатків і може призвести до несанкціонованого доступу до бази даних та витоку чутливої інформації.
2. Ручна перевірка на вразливості до SQL-ін'єкцій може бути часо- та працезатратною, а також піддається помилкам. Автоматизовані сканери вразливостей є корисними інструментами для виявлення SQL-ін'єкцій та інших безпекових проблем у веб-додатках.
3. Існує кілька популярних автоматизованих сканерів, таких як Acunetix, Burp Suite, Nessus, OpenVAS, SQLMap, OWASP ZAP та Nikto, які надають різні функціональні можливості та рівень деталізації.
4. Кожен автоматизований сканер має свої переваги та недоліки, які потрібно враховувати при виборі. Важливо зрозуміти вимоги вашого проекту та налаштування, щоб вибрати найбільш підходящий сканер.
5. Автоматизовані сканери допомагають зекономити час і ресурси, забезпечуючи автоматичне виявлення потенційних вразливостей. Однак, вони не є повністю надійними і можуть потребувати додаткової ручної перевірки та аналізу.

Комбінація автоматизованих сканерів та ручного аналізу може дати кращі результати, дозволяючи знайти більше вразливостей та забезпечити повнішу безпеку веб-додатків.

Перелік посилань

1. P. Kumar and R. K. Pateriya, "A survey on SQL injection attacks, detection and prevention techniques," 2012 Third International Conference on Computing, Communication and Networking Technologies (ICCCNT'12), Coimbatore, India, 2012, pp. 1-5, doi: 10.1109/ICCCNT.2012.6396096.
2. Atefeh Tajpour, Maslin Massrum and Mohammad Zaman Heydari, "Comparison of SQL injection detection and prevention techniques," 2010 2nd International Conference on Education Technology and Computer, Shanghai, China, 2010, pp. V5-174-V5-179, doi: 10.1109/ICETC.2010.5529788.
3. P. A. Sonewar and N. A. Mhetre, "A novel approach for detection of SQL injection and cross site scripting attacks," 2015 International Conference on Pervasive Computing (ICPC), Pune, India, 2015, pp. 1-4, doi: 10.1109/PERVASIVE.2015.7087131
4. P. Kumar and R. K. Pateriya, "A survey on SQL injection attacks, detection and prevention techniques," 2012 Third International Conference on Computing, Communication and Networking Technologies (ICCCNT'12), Coimbatore, India, 2012, pp. 1-5, doi: 10.1109/ICCCNT.2012.6396096.
1. K. R. Veerabudren and G. Bekaroo, "Security in Web Applications: A Comparative Analysis of Key SQL Injection Detection Techniques," 2022 4th International Conference on Emerging Trends in Electrical, Electronic and Communications Engineering (ELECOM), Mauritius, 2022, pp. 1-6, doi: 10.1109/ELECOM54934.2022.9965264.

Надійшла: 22.08.2023 Рецензент: д.т.н., професор Кожухівський А.Д.