УДК 004.056

Пашнєв Г.Р., Волошин В.Ю., Кушніренко Н.І.

Національний університет «Одеська політехніка»

РОЗРОБКА АЛГОРИТМУ ПРОТИДІЇ ПОШИРЕНИМ ВРАЗЛИВОСТЯМ БЕЗПЕКИ ВЕБ-ЗАСТОСУНКІВ

Вступ.Сучасні веб-застосунки є ключовим елементом цифрової інфраструктури, через який користувачі отримують доступ до послуг, фінансових операцій, баз даних та персональних кабінетів. Через це вони часто стають об'єктом кібератак. Найпоширенішими з них є SQL-ін'єкції, XSS, CSRF, brute-force та атаки через повторне використання токенів [1].

Наявність вразливостей у веб-застосунках може призвести до витоку конфіденційної інформації, компрометації облікових записів та порушення роботи інформаційних систем.

Мета: Аналіз основних типів атак на веб-застосунки та розробка універсального алгоритму для підвищення їх захищеності.

Основна частина

Для аналізу безпеки веб-застосунків було розглянуто модель взаємодії

«користувач – веб – сервер – база даних».

Основна мета атак на застосунки, побудовані за такою моделлю полягає в порушенні конфіденційності, цілісності або доступності даних. Одним з основних видів атак на веб-застосунки є SQL-ін'єкції, які використовують некоректну обробку введених даних для виконання довільних запитів до бази, що вимагає захисту через підготовлені вирази, валідацію та використання ORM-систем.

Загрози на клієнтському боці включають:

– Cross-Site Scripting (XSS) - вбудовування шкідливого JavaScript-коду з метою викрадення сесійних даних

– Cross-Site Request Forgery (CSRF) - підміну запитів від імені користувача.

Запобігання XSS здійснюється шляхом HTML-ескейпінгу та впровадження політики Content Security Policy, тоді як захист від CSRF забезпечують токени автентичності та заголовок SameSite.

Для захисту облікових записів від brute-force атак (автоматизований підбір паролів) критично важливе обмеження спроб входу, використання CAPTCHA та обов'язкова двофакторна автентифікація (2FA).

Проти повторного використання токенів (Replay attacks) - перехоплення дійсного токена - застосовують TLS-шифрування, встановлення короткого часу життя токена та механізми оновлення токенів після входу.

Результати дослідження щодо популярності атак, отримані на основі аналізу сучасних звітів OWASP та практичного тестування, наведені на рисунку 1.

Рисунок 1 - Кількість атак

Детальні описи кожного типу атаки, їхні наслідки та можливі методи протидії наведені в таблиці 1. На основі отриманих даних сформовано рекомендації щодо підвищення рівня безпеки веб-застосунків та мінімізації ризиків несанкціонованого доступу [3].

Таблиця 1 – Поширені атаки та методи захисту

Атака / проблема	Орієнтовна поширеність (%)	Вплив	Складність експлуатації	Рекомендовані заходи захисту
SQL-ін'єкції	11.5	Високий - витік/зміна БД	Середня	Prepared statements, ORM, WAF, SAST/DAST
Ненадійне зберігання паролів	17.5	Високий - компрометація облікових записів	Низька	Argon2/Bcrypt, соль, MFA, захищені бекапи
Недостатній контроль доступу	22.5	Дуже високий - ескалація привілеїв	Низька–середня	RBAC/ABAC, централізована авторизація, тестування прав
Витік конфіденційних даних	27.5	Дуже критичний - фінансові та репутаційні збитки	Варіюється	Шифрування at-rest/in-transit, DLP, моніторинг
Відсутність обмеження кількості спроб входу	14.0	Середній-високий - компрометація облікових записів	Низька	Rate limiting, lockout, CAPTCHA, MFA
Повторне використання токенів	8.5	Середній - відтворення транзакцій, сесійне захоплення	Середня	TLS, nonce, короткий TTL, refresh token flows

Згідно з таблицею, пріоритетом має бути захист від витоку даних через шифрування та усунення слабкого контролю доступу шляхом централізованої авторизації. Для запобігання компрометації облікових записів слід використовувати сильні алгоритми хешування (Argon2/Bcrypt) із сіллю, MFA і Rate limiting/lockout для обмеження спроб входу. SQL-ін'єкції мають блокуватись через Prepared statements, а повторне використання токенів - через TLS і короткий TTL.

Розробка алгоритму протидії поширеним вразливостям безпеки передбачає системний підхід, який охоплює всі етапи життєвого циклу програмного забезпечення - від розробки до експлуатації. Основною метою такого алгоритму є запобігання виникненню вразливостей ще на етапі створення коду, а також забезпечення швидкого виявлення і усунення потенційних загроз у вже функціонуючих системах.

Серед основних кроків алгоритму передбачено перевірку та фільтрацію всіх вхідних даних для запобігання ін'єкційним атакам, зокрема SQL Injection та XSS. Важливо забезпечити надійне шифрування інформації під час її зберігання та передавання, що мінімізує ризик витоку конфіденційних даних. Для запобігання несанкціонованому доступу алгоритм передбачає впровадження багатофакторної автентифікації, обмеження кількості спроб входу та проведення регулярного аудиту облікових записів користувачів.

Висновок. У межах дослідження проаналізовано найпоширеніші типи атак на веб-застосунки (SQL-ін'єкції, XSS, brute-force) та існуючі методи захисту від них. Результатом роботи є запропонований алгоритм, що може бути використаний під час розробки веб-застосунків з підвищеним рівнем захищеності.

Перелік використаних джерел.

1. OWASP Foundation. OWASP Top 10: 2021.

2. Сидоренко П.В. Безпека веб-застосунків: методи захисту та тестування. - Київ: КПІ, 2023.

3. Ristic I. ModSecurity Handbook: The Complete Guide to Web Application Firewalls. Feisty Duck, 2021.