УДК 004.056.5

Опанасенко Є.Р., магістрантка
Бродський Ю. Б., к.т.н., доцент
Россінський Ю.М., к.т.н., доцент
Державний університет «Житомирська політехніка»

АНАЛІЗ ТА ОЦІНЮВАННЯ ЕФЕКТИВНОСТІ ЗАСТОСУВАННЯ СКАНЕРА OWASP ZAP НА ПРИКЛАДІ ВЕБ-ДОДАТКУ DVWA

Стрімка цифровізація та міграція бізнес-процесів у веб-середовище загострюють проблему захисту інформаційних ресурсів від кібератак. В умовах, коли ручний пошук вразливостей вимагає значних часових ресурсів, інструменти автоматизації, такі як OWASP Zed Attack Proxy (ZAP), набувають критичного значення. Проте сліпа довіра до автоматичних алгоритмів несе в собі ризики пропуску складних загроз. Дослідження можливостей цього сканера дає змогу сформувати чітке уявлення про реальну ефективність автоматизованих засобів та визначити оптимальну стратегію їх використання. Отримані результати підтверджують, що успішність аудиту безпеки залежить не стільки від вибору інструменту, скільки від методології його застосування, зокрема поєднання автоматики з людським інтелектом.

Особливу цінність для розуміння меж можливостей автоматизованих сканерів становить аналіз їх роботи в середовищі DVWA (Damn Vulnerable Web Application). Цей додаток є не просто набором вразливих скриптів, а повноцінним полігоном, що імітує реальні помилки розробників на різних рівнях захищеності. Специфіка DVWA полягає у градації рівнів безпеки («Low», «Medium», «High», «Impossible»), що дозволяє простежити еволюцію коду від повної відсутності захисту до впровадження фільтрації вхідних даних та використання підготовлених запитів. Тестування на такій платформі чітко демонструє момент, коли автоматичний сканер втрачає ефективність: якщо на низькому рівні ZAP легко ідентифікує вразливості за стандартними сигнатурами, то вже на середньому рівні, де застосовуються базові методи санітизації (наприклад, функція mysql_real_escape_string або прості регулярні вирази), автоматика часто зазнає невдачі без додаткового налаштування контексту та ручного корегування векторів атаки[3].

Аналіз роботи OWASP ZAP показує, що ефективність сканування прямо пропорційна глибині попереднього налаштування та розумінню природи виявлених загроз. Використання базових автоматичних

режимів дозволяє виявити поверхневі вразливості, покриваючи не більше 60–70% потенційних загроз. Однак практична користь від результатів роботи ZAP значно зростає при детальному розборі звітів. Сканер не лише сигналізує про наявність проблеми, а й надає доказову базу: повний текст HTTP-запиту, використаний шкідливий пейлоад (payload) та відповідь сервера, що підтверджує успішну експлуатацію. Це дає розробникам точні інструкції для відтворення помилки та її виправлення[2].

Ключовим інсайтом роботи є доведення переваги гібридного підходу над суто автоматичним. Експериментальні дані свідчать, що найвищий рівень виявлення вразливостей (до 95%) досягається при використанні OWASP ZAP у режимі перехоплюючого проксі. Це дозволяє поєднувати автоматичний пошук типових помилок із ручним фаззингом (fuzzing) — підстановкою специфічних пейлоадів у місця зі складною бізнес-логікою. Саме такий метод дає змогу обходити фільтри безпеки та виявляти критичні вразливості високого рівня складності, наприклад, збережені XSS або сліпі SQL-ін'єкції, які автоматичні алгоритми часто ігнорують через неможливість коректної інтерпретації відповіді сервера[1,2].

Таким чином, результати дослідження дають підстави стверджувати, що OWASP ZAP є потужним елементом у системі кіберзахисту, проте його роль має зводитися до інструменту підсилення можливостей фахівця, а не повної заміни ручного тестування. Впровадження комплексної методики, де автоматизація пришвидшує рутинні процеси, а ручний контроль забезпечує глибину аналізу, виступає запорукою мінімізації ризиків та підвищення загального рівня захищеності веб-додатків.

Список використаних джерел

1. OWASP Zed Attack Proxy Documentation. OWASP Foundation. [Електронний ресурс]. – Режим доступу: URL: https://www.zaproxy.org/docs/
2. Scarfone, K., et al. (2008). Technical Guide to Information Security Testing and Assessment. NIST Special Publication 800-115.
3. Damn Vulnerable Web Application (DVWA). GitHub Repository. [Електронний ресурс]. – Режим доступу: URL: https://github.com/digininja/DVWA