УДК 004.056 DOI https://doi.org/10.35546/kntu2078-4481.2025.4.3.7

К. І. ДОРОФЄЄВА
магістрант кафедри безпеки інформаційних технологій
Харківський національний університет радіоелектроніки
ORCID: 0009-0003-8116-9050

О. В. СЄВЄРІНОВ
кандидат технічних наук, доцент,
професор кафедри безпеки інформаційних технологій
Харківський національний університет радіоелектроніки
ORCID: 0000-0002-6327-6405

З. М. СИДОРЕНКО
асистент кафедри безпеки інформаційних технологій
Харківський національний університет радіоелектроніки
ORCID: 0000-0002-0104-6807

В. М. СУХОТЕПЛИЙ
старший викладач кафедри радіоелектронних систем
пунктів управління Повітряних Сил
Харківський національний університет Повітряних Сил
імені Івана Кожедуба
ORCID: 0000-0002-2566-4167

ЗАСТОСУВАННЯ ІНСТРУМЕНТА АНАЛІЗУ БЕЗПЕКИ ДЛЯ ВИЯВЛЕННЯ КРИТИЧНИХ ВРАЗЛИВОСТЕЙ У ВЕБ-ДОДАТКАХ

У статті розглянуто особливості застосування інструмента OWASP Zed Attack Proxy у процесі тестування на проникнення веб-додатків, що обумовлює необхідність аналізу його функціональних можливостей і місця в сучасних підходах до аудиту інформаційної безпеки. Досліджено принципи роботи проксі-механізму ZAP, що забезпечує перехоплення та аналіз HTTP(S)-трафіку, а також можливість побудови карти веб-додатка на основі спостережених запитів і відповідей. Проаналізовано методи пасивного сканування, яке дозволяє здійснювати оцінку безпеки без втручання в поведінку сервера, зокрема виявляти відсутні або некоректно налаштовані HTTP-заголовки, особливості роботи cookies та інші технічні недоліки конфігурації. Розглянуто застосування активного сканування, що передбачає виконання контрольованих тестів на наявність поширених технічних вразливостей, а також акцентовано на необхідності проводити такі дії виключно у тестових середовищах. Описано роботу інструментів ручного аналізу, включно з редактором запитів, переглядом історії трафіку та базовим фуззингом, що дозволяє дослідникові перевіряти поведінку додатка за різних умов. Окремо розглянуто можливості розширення функціональності інструмента шляхом використання додаткових модулів та застосування API ZAP для автоматизації рутинних перевірок у межах підготовлених сценаріїв. У роботі також визначено обмеження інструмента, зокрема труднощі у виявленні складних логічних вразливостей, що потребують експертного ручного дослідження. Отримані результати дозволяють визначити місце OWASP ZAP як інструмента для технічного аудиту, початкових перевірок безпеки і навчальних цілей, а також підкреслити необхідність його комбінування з ручним аналізом при дослідженні складних систем.

Ключові слова: *OWASP ZAP, тестування на проникнення, веб-додатки, веб-безпека, проксі-аналіз, пасивне сканування, активне сканування.*

K. I. DOROFIEIEVA
Master's Student at the Department of Information Technology Security
Kharkiv National University of Radio Electronics
ORCID: 0009-0003-8116-9050

O. V. SIEVIERINOV
Candidate of Technical Sciences, Associate Professor,
Professor at the Department of Information Technology Security
Kharkiv National University of Radio Electronics
ORCID: 0000-0002-6327-6405

© К. І. Дорофєєва, О. В. Сєвєрінов, З. М. Сидоренко, В. М. Сухотеплий, 2025
Стаття поширюється на умовах ліцензії CC BY 4.0

Z. M. SYDORENKO
Assistant at the Department of Information Technology Security
Kharkiv National University of Radio Electronics
ORCID: 0000-0002-0104-6807

V. M. SUKHOTEPLYI
Senior Instructor at the Department of Radioelectronic Systems
of Control Points of Air Forces
Ivan Kozhedub Kharkiv National Air Force University
ORCID: 0000-0002-2566-4167

USING A SECURITY ANALYSIS TOOL TO IDENTIFY CRITICAL VULNERABILITIES IN WEB APPLICATIONS

The article examines the features and practical applications of the OWASP Zed Attack Proxy tool in web application penetration testing, emphasizing the need to analyze its capabilities and role in modern approaches to information security assessment. The study investigates the principles of ZAP's proxy mechanism, which allows the interception and analysis of HTTP(S) traffic and the construction of a site map based on observed requests and responses. Methods of passive scanning are analyzed, which enable security assessment without affecting server behavior, including the identification of missing or improperly configured HTTP headers, issues related to cookies, and other technical configuration shortcomings. The use of active scanning is also examined, involving controlled tests for common technical vulnerabilities, with an emphasis on performing such operations only in dedicated testing environments. Manual analysis tools are described, including the request editor, traffic history viewer, and basic fuzzing functionality, which allow researchers to evaluate application behavior under various input conditions. Additionally, the paper considers the possibilities of extending ZAP's functionality through optional add-ons and using its API to automate routine tasks within predefined testing scenarios. The limitations of the tool are outlined, particularly the challenges of detecting complex logical vulnerabilities that require expert manual investigation. The findings indicate that OWASP ZAP is suitable for technical audits, initial security evaluations, and educational purposes, while highlighting the necessity of combining automated scanning with manual analysis when assessing complex web applications.

Key words: *OWASP ZAP, penetration testing, web applications, web security, proxy analysis, passive scanning, active scanning.*

Постановка проблеми

Забезпечення безпеки веб-додатків є одним із ключових завдань сучасних інформаційних систем, оскільки більшість сервісів, бізнес-процесів та доступу до даних реалізується саме через веб-інтерфейси. Розповсюдженість веб-технологій сприяє збільшенню кількості загроз, спрямованих на експлуатацію вразливостей у логіці застосунків, у компонентах серверної інфраструктури та у механізмах обробки даних користувача. Значна частина атак пов'язана не з високотехнологічними способами злому, а з типовими помилками програмування та недоліками конфігурації.

Тестування на проникнення веб-додатків дозволяє виявляти такі недоліки ще до того, як вони будуть використані зловмисником. При цьому важливо застосовувати інструменти, які дозволяють аналізувати HTTP-трафік, перевіряти реакцію веб-додатка на різні типи некоректних або небезпечних запитів та формувати об'єктивні висновки щодо рівня безпеки.

OWASP ZAP є інструментом з відкритим програмним кодом, який забезпечує базові та розширені можливості для тестування на проникнення. На відміну від комерційних продуктів, ZAP доступний безкоштовно, має активну спільноту та орієнтований як на початківців, так і на фахівців, які здійснюють ручний чи напівавтоматизований аудит веб-безпеки.

Проблема, яку розглядає дана стаття, полягає в необхідності системного використання OWASP ZAP під час виконання тестування на проникнення та у формуванні методологічного підходу до його застосування відповідно до реальних можливостей інструмента.

Аналіз останніх досліджень і публікацій

У наукових і прикладних публікаціях останніх років значну увагу приділено розвитку методологій тестування на проникнення та використанню спеціалізованих засобів аналізу безпеки веб-додатків. У роботі [1] автори наголошують, що пентест є ключовим інструментом кібербезпечного менеджменту, оскільки дозволяє виявляти вразливості, оцінювати рівень ризику та формувати рекомендації щодо підвищення стійкості інформаційних систем. Автори підкреслюють важливість інструментів динамічного аналізу, які забезпечують відтворення реальних атакувальних сценаріїв у процесі тестування. Скибун, аналізуючи мету та завдання пентесту, виокремлює значення автоматизованих сканерів і проксі-інструментів, що дають можливість як пасивного, так і активного вивчення поведінки веб-додатка під час взаємодії з ним [2]. Прямо у контексті OWASP ZAP детальний аналіз наданий

у дослідженні Maniraj та співавторів, де інструмент розглядається як комплексне рішення для сканування, перехоплення HTTP-трафіку, моделювання атаки та автоматизованого моніторингу вразливостей. Автори відзначають ефективність ZAP при виявленні XSS, SQL Injection та конфігураційних помилок на реальних веб-платформах [3]. У роботі [4] пояснюється сутність вразливостей з OWASP Top 10, їхні корені та типові шляхи експлуатації, підкреслюючи необхідність інструментів, здатних аналізувати HTTP-взаємодію та структуру веб-додатків у реальному часі. Завершальним елементом є офіційна документація ZAP, що містить опис механізмів проксі-перехоплення, роботи пасивного та активного сканерів, алгоритмів Spider і AJAX-Spider, а також API для автоматизації – ці матеріали формують практичну базу для використання інструмента в пентестингу [5]. Сукупність публікацій відображає зростаючу увагу до інтеграції OWASP ZAP у процеси тестування на проникнення та підтверджує його ефективність як інструмента динамічного аналізу безпеки веб-додатків.

Формулювання мети дослідження

Метою дослідження є оцінка реальних можливостей інструмента OWASP ZAP та формування методологічного підходу до його використання у процесі тестування на проникнення веб-додатків. Дослідження спрямоване на вивчення функціоналу ZAP, ефективності його пасивного та активного сканування, можливостей перехоплення та аналізу HTTP/HTTPS-трафіку, а також практичної придатності інструмента для виявлення типових технічних вразливостей і підтримки комплексного аудиту веб-додатків.

Викладення основного матеріалу дослідження

OWASP ZAP є інструментом відкритого коду для тестування веб-додатків, призначеним для виявлення технічних вразливостей. Основною особливістю ZAP є робота як проксі-сервера, що дозволяє перехоплювати HTTP(S)-трафік між клієнтом та сервером, аналізувати його та зберігати історію запитів і відповідей. Це дає можливість досліднику відстежувати всі параметри, передані веб-додатком, а також виявляти помилки конфігурації, небезпечні заголовки або некоректну обробку даних користувача [5].

Інструмент підтримує пасивне і активне сканування. Пасивне сканування виконується автоматично при отриманні трафіку, не впливаючи на роботу сервера, і дозволяє виявляти відсутність безпечних заголовків, небезпечні cookies, незахищені параметри та інші технічні недоліки. Активне сканування включає надсилання контрольованих тестових запитів для перевірки реакції сервера на потенційно небезпечні вхідні дані, наприклад, на SQL Injection чи XSS [3]. ZAP також надає інструменти ручного аналізу, такі як редактор запитів, історія трафіку та базовий фуззинг. Ці інструменти дозволяють досліднику перевіряти додаток за різних умов, змінювати параметри запитів і оцінювати реакцію сервера. Також доступні додаткові модулі (add-ons), що розширюють функціональність ZAP для інтеграції з іншими інструментами або додаткового аналізу специфічних протоколів.

Використання проксі-механізму

Однією з ключових функціональних можливостей OWASP ZAP є використання проксі-механізму, який дає змогу перехоплювати, аналізувати та модифікувати HTTP/HTTPS-трафік між браузером користувача та веб-додатком. Під час роботи ZAP виступає як проміжна ланка, що приймає всі вихідні запити клієнта та відповіді сервера, забезпечуючи повний контроль над переданими параметрами та протоколами взаємодії. Завдяки цьому механізму інструмент дозволяє не лише спостерігати за поведінкою веб-додатка в реальному часі, але й вручну коригувати запити для оцінки реакції сервера на змінені параметри. Такий підхід є особливо корисним під час тестування вразливостей, що потребують ручної взаємодії, включно з маніпуляціями сеансовими даними, зміною параметрів форм, тестуванням автентифікації та контролю доступу [5].

На рисунку 1 зображено принцип роботи проксі-механізму ZAP, де інструмент виступає як посередник між клієнтом і сервером, забезпечуючи прозорий огляд і можливість втручання в трафік під час тестування. Така архітектура є базовою для всіх подальших етапів аналізу та дозволяє тестувальнику повністю контролювати комунікацію веб-додатка.

Проксі-механізм ZAP тісно інтегрований з такими інструментами, як History, Request/Response Editor, Breakpoints та Fuzzer, що надає можливість не тільки фіксувати кожен елемент трафіку, але й зупиняти виконання запиту на певному етапі для внесення змін. Використання breakpoints дозволяє детально дослідити логіку обробки запитів, виявляти приховані параметри та оцінювати поведінку серверної частини при аномальних або навмисно некоректних даних.

Також ZAP підтримує роботу з HTTPS-трафіком шляхом генерації власного локального сертифіката, що дає змогу безпечно розшифровувати зашифровані дані та забезпечувати повноцінний аналіз навіть у випадку використання сучасних транспортних протоколів. Завдяки цьому проксі-механізм стає основою для більшості методів тестування вразливостей, забезпечуючи прозорий огляд усіх взаємодій між клієнтом та сервером.

Пасивне сканування

Пасивне сканування в OWASP ZAP є базовим етапом аналізу безпеки, який виконується без створення додаткового навантаження на веб-додаток. На відміну від активних методів, пасивне сканування не змінює структуру запитів та не модифікує дані, що дає змогу проводити його на робочих системах без ризику порушення

функціонування або виникнення побічних ефектів. OWASP ZAP здійснює пасивний аналіз у фоновому режимі під час роботи проксі-механізму: кожен перехоплений HTTP-запит та відповідь сервера автоматично перевіряються на наявність ознак потенційних вразливостей. До типових категорій, що можуть бути виявлені таким способом, належать: некоректні заголовки безпеки, надлишкове розкриття інформації, неправильна конфігурація хешування, відсутність політик захисту від XSS, використання слабких алгоритмів хешування тощо. Згідно з практиками, описаними OWASP, пасивне сканування виступає одним з найбільш безпечних способів первинної оцінки безпеки, оскільки воно не впливає на логіку роботи веб-додатка й дозволяє поступово формувати перелік можливих проблем, які потребують глибшого аналізу під час активного сканування або ручної перевірки [6]. Такий підхід підтримується також українськими дослідженнями, де наголошується, що пасивні техніки є оптимальними на початкових етапах тестування та дозволяють оцінити загальний рівень безпеки IT-системи без втручання у її функції. Цей етап створює основу для подальших досліджень, забезпечуючи послідовний і безпечний перехід до більш інтенсивних методів оцінювання безпеки [1]. На рисунку 2 показано, що ZAP виявив відсутність заголовка Content-Security-Policy (CSP) у відповіді сервера. Виявлення подібних недоліків на етапі пасивного сканування дозволяє зробити висновки щодо базового стану безпеки додатка ще до застосування активних методів аналізу.

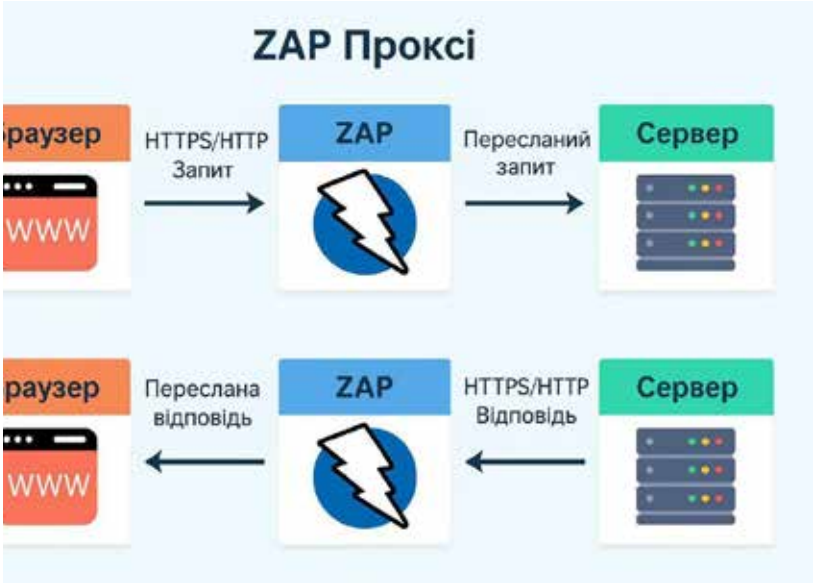

Рис. 1. Архітектура роботи проксі-механізму ZAP

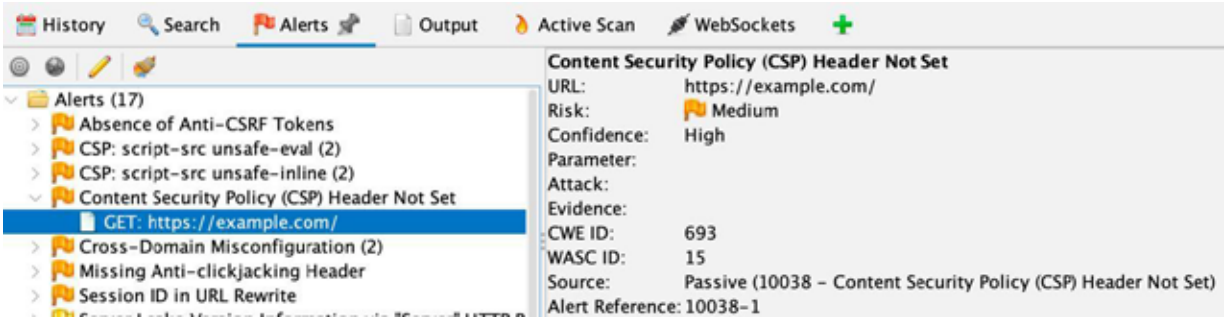

Рис. 2. Виявлення відсутності CSP у відповідях сервера під час пасивного сканування

Для систематизації результатів аналізу в таблиці 1 наведено узагальнену класифікацію типових проблем безпеки, які можуть бути виявлені під час пасивного сканування OWASP ZAP [5].

Пасивне сканування у ZAP виконується автоматично. Інструмент аналізує всі запити та відповіді, що проходять через проксі, але не надсилає додаткових запитів до веб-сервера.

Це робить пасивне сканування безпечним щодо впливу на тестований додаток. Пасивне сканування є першим етапом перевірки, оскільки воно не змінює поведінку додатка та не створює додаткового навантаження.

Таблиця 1

Категорії вразливостей, які виявляє пасивне сканування

Категорія	Приклади
Небезпечні заголовки	Відсутність X-Frame-Options, Content-Security-Policy
Проблеми cookie	Відсутність HttpOnly, Secure, SameSite
Інформаційні витоки	Версія сервера в заголовках, stack traces
Небезпечні параметри	Передача токенів у URL, параметри IDOR
Конфігураційні недоліки	Mixed content, некоректні cache-policy

Активне сканування

Активне сканування в OWASP ZAP є наступним етапом після пасивного аналізу та передбачає безпосереднє формування інструментом спеціально сконструйованих HTTP-запитів з метою виявлення вразливостей, які не можуть бути ідентифіковані лише шляхом спостереження за трафіком. На відміну від пасивного аналізу, активне сканування суттєво впливає на поведінку веб-додатка, оскільки використовує методи, що імітують потенційні атаки, включно зі спробами ін'єкцій, маніпуляцією параметрами, обходом автентифікації та тестуванням механізмів обробки помилок. В OWASP ZAP активне сканування реалізується через механізм Active Scan Rules, який використовує спеціалізовані сканери, що аналізують кожен параметр, заголовок, cookie та тіло запиту на предмет можливих точок впливу. Користувач може визначати масштаб аналізу, рівень навантаження, типи тестів та обмеження, що дозволяє адаптувати процес до середовища тестування, знижуючи ризики небажаних змін у роботі цільового додатка [2].

На рисунку 3 продемонстровано конфігурацію, яка визначає, які типи перевірок будуть виконані, рівні інтенсивності тестування та граничні значення для запитів. Такі параметри дозволяють адаптувати роботу сканера до вимог дослідження.

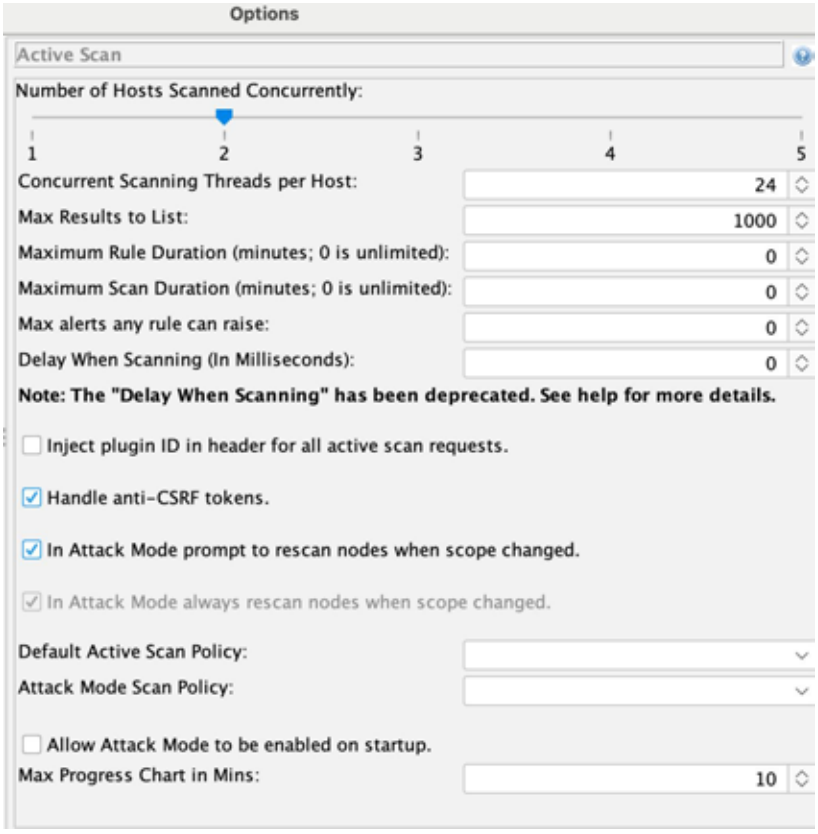

Рис. 3. Налаштування параметрів активного сканування в OWASP ZAP

Суттєвим аспектом активного сканування є ризикованість його застосування, тому слід з обережністю підходити до вибору налаштувань сканеру. Оскільки інструмент навмисно генерує запити, які можуть викликати виключення, створити нестабільність, вплинути на стан даних або спровокувати блокування облікових записів, його не рекомендується виконувати на продуктивних системах. Дослідження підкреслюють, що активне сканування повинно здійснюватися лише у контрольованому середовищі, а перед початком тестування мають бути

передбачені заходи щодо збереження даних та мінімізації шкоди [3]. У практиці тестування на проникнення активне сканування використовується не як самостійний інструмент, а як складова комплексної методики, де воно доповнює ручний аналіз та контекстно орієнтовані перевірки. Результати, отримані за допомогою OWASP ZAP, слугують відправною точкою для глибшої експлуатації виявлених вразливостей і формування детального звіту щодо реальних ризиків для веб-додатка.

В таблиці 2 наведена класифікація типових категорій вразливостей, які визначаються за допомогою активного сканування OWASP ZAP.

Таблиця 2

Категорії вразливостей, які виявляє активне сканування

Категорія	Приклади
Ін'єкційні атаки	SQL Injection, Command Injection, LDAP Injection
Cross-Site Scripting (XSS)	Reflected XSS, DOM-based XSS
Path Traversal	Доступ до заборонених директорій
Атаки на автентифікацію	Brute force, weak password checks
Проблеми із сесіями	Підбір session ID, повторне використання токенів
Логічні вразливості	Обхід авторизації, некоректна обробка бізнес-логіки

ZAP використовує набір готових тестових правил, що систематично перевіряють поведінку веб-додатка та аналізують відповіді сервера.

З огляду на різний характер пасивного та активного сканувань, доцільно здійснити їх систематичне порівняння за ключовими характеристиками, що наведено у таблиці 3. Такий аналіз дозволяє обґрунтовано обирати тактику тестування відповідно до цілей аудиту, рівня ризику та можливостей взаємодії з тестованим середовищем.

Таблиця 3

Порівняння можливостей видів сканувань в OWASP ZAP

Характеристика	Пасивне сканування	Активне сканування
Вплив на додаток	Не впливає на роботу системи	Може змінювати стан даних або викликати помилки
Додаткові запити	Не створює	Створює спеціальні тестові запити
Типи вразливостей	Заголовки безпеки, інформаційні витоки, некоректні конфігурації	Ін'єкції, XSS, обхід автентифікації, зміна параметрів
Ризик застосування	Мінімальний	Підвищений
Потребує контексту	Ні	Так, залежить від бізнес-логіки
Рекомендоване використання	Початковий етап тестування	Поглиблений аналіз та експлуатація

Інструменти ручного аналізу та автоматизації

ZAP також надає можливості ручного аналізу через Request Editor, History і Fuzzer, що дозволяє змінювати запити та перевіряти реакцію серверу. Для автоматизації рутинних завдань доступний API ZAP, який дозволяє запускати сканування, отримувати результати та формувати звіти безпосередньо із зовнішніх сценаріїв [5]. Також через Marketplace можна встановлювати додаткові модулі для специфічного аналізу. OWASP ZAP ефективний для виявлення типових технічних вразливостей, проте має обмеження у випадках складних логічних проблем або специфічних бізнес-логік, що потребують ручного аналізу. Додатково деякі складні сканування можуть вимагати значних ресурсів і часу, особливо для великих веб-додатків.

Висновки

У статті було розглянуто практичні аспекти використання інструмента OWASP ZAP у процесі тестування на проникнення веб-додатків, а також проаналізовано його можливості у контексті сучасних вимог до забезпечення кібербезпеки. Встановлено, що ZAP належить до категорії доступних і водночас функціонально повноцінних засобів, які дозволяють проводити комплексний аналіз безпеки веб-додатків на різних етапах життєвого циклу. Він підтримує як ручне тестування, так і автоматизоване сканування, що робить його універсальним інструментом як для початківців, так і для досвідчених фахівців.

У процесі дослідження було визначено, що ZAP забезпечує виявлення широкого спектра вразливостей, включно з помилками налаштування, слабкими механізмами автентифікації, вразливостями, пов'язаними з ін'єкціями, та недоліками у реалізації контролю доступу. Значною перевагою інструмента є його інтеграція з методологією OWASP, що дозволяє застосовувати його у рамках рекомендованих підходів до оцінювання безпеки. Окрему увагу приділено використанню розширень, що надають змогу адаптувати ZAP під специфічні потреби тестування та підвищувати рівень деталізації аналізу. Як показує практика, у поєднанні з ручними техніками інструмент перетворюється на ефективний компонент процесу тестування на проникнення.

Таким чином, застосування OWASP ZAP сприяє підвищенню якості безпекового аналізу веб-додатків, а його можливості дозволяють забезпечити глибоке вивчення потенційних загроз без необхідності використання комерційних рішень. Інструмент демонструє ефективність у виявленні поширених вразливостей та може бути інтегрований у регулярні процеси безпекового тестування в організаціях різного масштабу.

Список використаної літератури

1. Горбаченко С., Бойко В. Тестування на проникнення як ефективний інструмент менеджменту кібербезпеки. Information Technology and Society. 2023. № 3 (9). С. 23–29. URL: https://doi.org/10.32689/maup.it.2023.3.3 (дата звернення: 14.11.2025).
2. Скибун О. Тестування на проникнення: мета та цілі. Grail of Science. 2022. № 22. С. 161–163. URL: https://doi.org/10.36074/grail-of-science.25.11.2022.28 (дата звернення: 17.11.2025).
3. Maniraj S. P., Ranganathan C. S., Sekar S. Securing Web Applications with OWASP ZAP for Comprehensive Security Testing. International Journal of Advances in Signal and Image Sciences. 2024. Vol. 10, no. 2. P. 12–23. URL: https://doi.org/10.29284/ijasis.10.2.2024.12-23 (date of access: 17.11.2025).
4. Reddy s. OWASP Top 10 for Layman: OWASP Top 10. Independently Published, 2019.
5. ZAP – Documentation. *ZAP*. URL: https://www.zaproxy.org/docs/(date of access: 17.11.2025).
6. OWASP Automated Threat Handbook: Web Applications / T. Zaw et al. Lulu Press, Inc., 2018.

References

1. Horbachenko S., Boiko V. (2023) Penetration Testing as an Effective Tool of Cybersecurity Management. Information Technology and Society, no. 3 (9), pp. 23–29. URL: https://doi.org/10.32689/maup.it.2023.3.3. [in Ukrainian].
2. Skybun O. (2022) Penetration Testing: Purpose and Objectives. Grail of Science, no. 22, pp. 161–163. URL: https://doi.org/10.36074/grail-of-science.25.11.2022.28. [in Ukrainian].
3. Maniraj S. P., Ranganathan C. S., Sekar S. (2024) Securing Web Applications with OWASP ZAP for Comprehensive Security Testing. International Journal of Advances in Signal and Image Sciences, vol. 10, no. 2, pp. 12–23. URL: https://doi.org/10.29284/ijasis.10.2.2024.12-23 (accessed: 17.11.2025).
4. Reddy S. (2019) OWASP Top 10 for Layman: OWASP Top 10. Independently Published.
5. ZAP – Documentation. ZAP. URL: https://www.zaproxy.org/docs/ (accessed: 17.11.2025).
6. Zaw T., et al. (2018) OWASP Automated Threat Handbook: Web Applications. Lulu Press, Inc.

Дата першого надходження рукопису до видання: 11.11.2025
Дата прийнятого до друку рукопису після рецензування: 09.12.2025
Дата публікації: 31.12.2025