КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 3 (31), 2025
ISSN 2663 - 4023
DOI 10.28925/2663-4023.2025.31.1068
УДК 004.056.5

Хома Володимир Васильович
д.т.н.,професор кафедри захисту інформації
Національний Університет «Львівська Політехніка», Львів, Україна
ORCID: 0000-0001-9391-6525
volodymyr.v.khoma@lpnu.ua

Партика Андрій Ігорович
к.т.н., старший викладач кафедри захисту інформації
Національний Університет «Львівська Політехніка», Львів, Україна
ORCID: 0000-0003-3037-8373
andrii.i.partyka@lpnu.ua

Сабодашко Дмитро Володимирович
доктор філософії, старший викладач кафедри захисту інформації
Національний Університет «Львівська Політехніка», Львів, Україна
ORCID: 0000-0003-1675-0976
dmytro.v.sabodashko@lpnu.ua

СИСТЕМАТИЗАЦІЯ ТА ПОРІВНЯЛЬНИЙ АНАЛІЗ ІНСТРУМЕНТІВ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ НА ПРОНИКНЕННЯ ВЕБ-ДОДАТКІВ

Анотація. У статті досліджено сучасні підходи до автоматизації процесу тестування на проникнення веб-додатків. Зростання кількості та складності веб-загроз, обмеженість ресурсів на ручне тестування зумовлюють критичну потребу в ефективному використанні інструментів автоматизованого тестування. Проте різноманіття архітектур, принципів роботи та функціональних можливостей наявного інструментарію пентестингу створює проблему обґрунтованого вибору та ефективного поєднання. Метою статті є систематизація основних інструментів з відкритим кодом та проведення їх порівняльного аналізу на основі комплексу розроблених критеріїв. У роботі запропоновано класифікувати інструменти за чотирма категоріями: активні сканери на основі проксі (наприклад, OWASP ZAP), сканери на основі шаблонів (Nuclei), спеціалізовані інструменти експлуатації (sqlmap) та фаззери/сканери параметрів (ffuf). Для порівняльного аналізу визначено систему критеріїв, що включає функціональні (покриття OWASP Top-10, підтримка сучасних технологій), операційні (інтеграція в CI/CD, зручність) та технічні (ліцензія, активність розвитку) аспекти. На основі цих критеріїв проаналізовано інструменти, що представляють кожну з категорій. Результати дослідження показали, що жоден інструмент не є універсальним, а ефективність залежить від специфіки завдання. Встановлено, що найбільш універсальним для глибинного аналізу є OWASP ZAP, найшвидшим для масштабованого сканування – Nuclei, а найефективнішим для розвідки – ffuf. Ключовим висновком є рекомендація щодо синергетичного комбінування інструментів у єдиному робочому процесі (workflow) для максимального покриття фаз тестування: від розвідки до глибокої експлуатації. Отримані результати формують теоретичну основу для обґрунтованого вибору інструментів та побудови ефективних процесів автоматизованого тестування безпеки в рамках підходів DevSecOps.

Ключові слова: тестування на проникнення; вразливість; автоматизоване сканування безпеки; OWASP ZAP; Nuclei; sqlmap; систематизація; порівняльний аналіз; інформаційна безпека.

ВСТУП

У сучасних умовах цифрової трансформації веб-додатки є основним інтерфейсом для надання послуг, обробки конфіденційних даних та підтримки бізнес-процесів. Це

робить їх пріоритетною мішенню для кібератак, про що послідовно свідчить рейтинг OWASP Top Ten [1]. Водночас, ручне тестування безпеки таких додатків є ресурсоємним та залежить від кваліфікації фахівця, що обмежує його масштабованість. Автоматизоване тестування на проникнення за допомогою спеціалізованих інструментів стає критичною складовою життєвого циклу розробки безпечного ПЗ (DevSecOps). Водночас, ефективність автоматизованих інструментів значною мірою залежить від їх відповідності окремим етапам пентесту — розвідці, ідентифікації вразливостей, експлуатації та валідації [2]. Чітке розуміння того, яку фазу покриває конкретний клас інструментів, дозволяє будувати коректні workflow та уникати методологічних прогалин. Існує широка екосистема відкритих (open-source) інструментів для автоматизованого сканування, таких як OWASP ZAP [3], Nuclei [4], sqlmap [5], кожен з яких реалізує власний підхід. Однак, різноманіття архітектур, принципів роботи та функціональних можливостей створює значну проблему для фахівців. Це призводить до суб'єктивного, часто неефективного вибору інструментів, що може залишати критичні вразливості непоміченими, а відтак знижувати рівень безпеки веб-додатків. Тому важливо розробити комплексні критерії для порівняльного аналізу наявних інструментів автоматизації пентестингу і на цій основі виконати систематизацію їх функціоналу. Це посприяє обґрунтованому вибору оптимального інструментарію для конкретних сценаріїв тестування, формуванню уніфікованих підходів до інтегрування засобів автоматизації в DevSecOps-процеси та підвищенню безпеки веб-додатків. Така систематизація дозволить не лише мінімізувати ризики пропуску критичних вразливостей, а й оптимізувати витрати часу та ресурсів, а також створити передумови для стандартизації підходів до автоматизованого пентестигу веб-додатків.

Аналіз останніх досліджень і публікацій. Питання автоматизації пентесту та ефективності інструментів розглядаються як в академічних колах, так і в практичній площині. Методологічною основою слугує керівництво OWASP Web Security Testing Guide (WSTG) [2]. Більшість існуючих порівняльних матеріалів представлена у форматі технічних блогів, відеоогляді або статей, які часто обмежуються тестуванням однієї-двох утиліт на конкретному вразливому додатку, що не дозволяє сформувати узагальнену картину ефективності інструментів на різних етапах пентесту [7-8]. Такі роботи мають прикладну цінність, але їм бракує системності, глибини аналізу архітектурних особливостей та урахування сучасних вимог до інтеграції в CI/CD конвеєри [9-14]. Наукових праць, присвячених саме систематизації та багатокритеріальному порівнянню open-source інструментів автоматизованого пентесту веб-додатків, є обмежена кількість**.** Це створює прогалину, яка потребує заповнення структурованим аналізом, що поєднує теоретичну класифікацію з практичними критеріями оцінки.

Мета статті. Метою даної статті є систематизація основних відкритих інструментів автоматизованого тестування на проникнення веб-додатків та проведення їх порівняльного аналізу на основі комплексу структурних, функціональних та операційних критеріїв. Для досягнення поставленої мети сформульовано наступні завдання:

- Провести аналіз сучасних підходів до автоматизації тестування безпеки веб-додатків та визначити їх місце в загальній методології пентесту.
- Запропонувати класифікацію відкритих інструментів автоматизованого пентесту за ключовими ознаками: архітектурою, принципом дії та основним призначенням.
- Визначити та обґрунтувати систему критеріїв для порівняльного аналізу, що враховує технічні можливості (покриття класів вразливостей, підтримку сучасних

технологій) та практичні аспекти використання (зручність інтеграції в CI/CD, ліцензійні умови, активність розвитку).

- На основі обраних критеріїв провести детальний порівняльний аналіз найбільш релевантних представників кожної категорії інструментів.
- Сформулювати рекомендації щодо вибору та оптимального комбінування інструментів в залежності від конкретних цілей тестування та етапів життєвого циклу розробки ПЗ.

ТЕОРЕТИЧНІ ОСНОВИ ДОСЛІДЖЕННЯ

Місце автоматизованих інструментів у методології тестування на проникнення. Згідно з методологією OWASP Web Security Testing Guide (WSTG), повний цикл тестування веб-додатків включає як ручні техніки дослідження бізнес-логіки, так і автоматизовані перевірки на наявність типових вразливостей. Автоматизовані інструменти, основним завданням яких є сканування (англ. Vulnerability Scanning), ефективно доповнюють роботу пентестера, дозволяючи швидко охопити великий обсяг цілей (елементи інтерфейсу, параметри запитів, точки API) та виявити стандартні помилки, пов'язані з недостатньою валідацією вхідних даних, небезпечною конфігурацією тощо. Таким чином, вони виконують критичну функцію на етапах активного пошуку вразливостей (Active Vulnerability Detection) та частково – на етапі збору інформації (Reconnaissance).

Класифікація інструментів автоматизованого тестування за архітектурою та призначенням. На основі аналізу принципів функціонування та цільового призначення, сучасні open-source інструменти можна систематизувати за чотирма основними категоріями, представленими в *Таблиці 1*.

Таблиця 1

Класифікація інструментів автоматизованого тестування веб-додатків

Категорія	Принцип дії	Основне призначення	Ключові представники
Активні сканери на основі проксі	Інструмент працює як MITM проксі	Комплексне, глибоке сканування	OWASP ZAP, Burp Suite
Сканери на основі шаблонів (Template-based)	Використовують базу попередньо написаних шаблонів	Швидке масштабоване сканування на великій кількості на предмет відомих, добре визначених вразливостей	Nuclei, Sn1per
Спеціалізовані інструменти експлуатації	Сконцентровані на максимально ефективній експлуатації одного конкретного класу вразливостей	Глибоке тестування та демонстрація критичності конкретного типу	Sqlmap, XSStrike, SSDFmap
Фаззери та сканери параметрів	Систематично перебирають можливі значення параметрів	Виявлення прихованого контенту, нерозмічених точок входу API	Ffuf, wfuzz, Param Miner

Система критеріїв для порівняльного аналізу. Для об'єктивного порівняння інструментів різних категорій запропоновано систему критеріїв, яка поєднує технологічні та операційні аспекти.

А. Функціональні критерії оцінюють технічні можливості інструменту:

- **Покриття OWASP Top-10:** Здатність виявляти вразливості з актуального списку OWASP Top Ten [5].
- **Підтримка сучасних технологій:** Можливість тестувати REST API, GraphQL, WebSocket, односторінкові додатки (SPA).
- **Режими роботи:** Наявність повністю автоматичного, напівавтоматичного (з участю тестувальника) та пасивного сканування.

Б. Операційні критерії стосуються практичної інтеграції та використання:

- **Інтеграція в CI/CD:** Наявність командного рядка (CLI), Docker-образу, плагінів для Jenkins, GitLab CI, GitHub Actions.
- **Зручність використання:** Якість графічного інтерфейсу (GUI), доступність та повнота документації, крива навчання.
- **Формати звітів:** Підтримка популярних форматів для експорту результатів (HTML, PDF, JSON, XML).

В. Технічні та організаційні критерії враховують довгострокову підтримку та правові аспекти:

- **Ліцензія:** Тип ліцензії (наприклад, Apache 2.0, GPL) та обмеження на комерційне використання.
- **Активність розвитку:** Регулярність оновлень, розмір та активність спільноти на GitHub (кількість зірок, частота комітів, вирішені питання).
- **Масштабованість:** Ефективність роботи при скануванні великої кількості цілей або складних додатків.

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

На основі запропонованої класифікації та системи критеріїв проведено порівняльний аналіз чотирьох ключових інструментів, які є найбільш релевантними представниками своїх категорій: OWASP ZAP (активний сканер), Nuclei (сканер на шаблонах), sqlmap (спеціалізований інструмент) та [6] (фаззер). Результати зведені в узагальнену таблицю (Таблиця 2), а нижче наведено детальний аналіз за категоріями.

Таблиця 2

Порівняльна характеристика інструментів автоматизованого тестування

Критерій	OWASP ZAP	Nuclei	sqlmap	ffuf	Примітки/пояснення
Категорія	Активний сканер	Сканер на шаблонах	Спеціалізований інструмент	Фаззер/ сканер параметрів	Відповідає класифікаїї таблиці 1
Покриття OWASP Top-10	Високе	Середнє-високе	Дуже низьке	Низьке	ZAP покриває 8-9 категорій. Nuclei залежить від шаблонів
Підтримка API/ Сучасних технологій	Так(REST, GraphQL, WebSocket)	Так(REST, GraphQL)	Обмежена	Так	ZAP має вбудовані сканери для API. Nuclei має спеціальні шаблони

Режим роботи	Авто, напівавто, пасивний	Автоматичний	Авто, напівавто	Авто	ZAP – єдиний з напіватоматичним режимом проксі
Інтеграція в CI/CD	Висока(CLI, Docker, REST, API)	Дуже висока (CLI, Docker, GitHub Actions)	Середня (CLI, Docker)	Дуже висока (CLI, Docker)	Усі мають CLI та Docker. ZAP має REST API для автоматизації. Nuclei та ffuf – ідеальні для CL
Зручність	Висока	Низька-середня	Низька	Низька	ZAP має зручний GUI. Інші – консольні інструменти з потужним, але складними опціями
Формат звітів	HTML, XML, JSON, PDF	JSON, HTML, Markdown	TXT, CSV, JSON	JSON, HTML, CSV, EJSON	ZAP пропонує найбільш структуровані звіти
Ліцензія	Apache 2.0	MIT	GNU GPLv3	MIT	Усі ліцензії дозволяють безкоштовне комерційне використання
Активність розвитку	Дуже висока(Фонд OWASP)	Дуже висока (ProjectDiscovery)	Висока (понад 10 років розвитку)	Висока (популярний інструмент)	Усі проекти маю активну спільноту, регулярні оновлення
Основна перевага	Універсаль-ність, глибина, ручний контроль	Швидкість, масштабність, актуальність шаблонів	Неперевершена ефективність для SQL - інструкцій	Неперевершена швидкість та гнучкість для розвідки	-
Основне обмеження	Високі вимоги до ресурсів, складність автоматиза-ція повного циклу	Залежність від якості шаблонів спільноти, обмежена глибина аналізу	Вузька спеціалізація, складність налаштування для нестандартних випадків	Не виявляє вразливостей без додаткових скриптів або інтеграцій	-

Аналіз результатів за категоріями інструментів

Активні сканери (OWASP ZAP). OWASP ZAP демонструє найвищий рівень універсальності та глибини аналізу серед розглянутих інструментів. Його ключова перевага – поєднання потужного автоматизованого сканування з можливістю повного ручного контролю через проксі-режим, що робить його незамінним для тестування складної бізнес-логіки. Недоліками є відносно високе споживання ресурсів та складність повної інтеграції в CI/CD через необхідність керування сесіями та контекстами.

Сканери на основі шаблонів (Nuclei). Nuclei виділяється видатною швидкістю та зручністю масштабування завдяки простій архітектурі та використанню шаблонів YAML. Активна спільнота ProjectDiscovery забезпечує швидке оновлення бази шаблонів для нових вразливостей (CVEs), що робить інструмент ідеальним для первинного рекогносцирування та моніторингу. Головним обмеженням є поверхневий аналіз: Nuclei

перевіряє лише на відповідність шаблону і не може глибоко досліджувати власну логіку додатку.

Спеціалізовані інструменти експлуатації (sqlmap). Як представник цієї категорії, sqlmap залишається «золотим стандартом» для виявлення та експлуатації SQL-ін'єкцій, демонструючи ефективність, недосяжну для універсальних сканерів. Він використовує складні алгоритми для визначення типу СКБД, техніки внесення та отримання даних. Його недолік – вузька спеціалізація. Він не призначений для пошуку інших класів вразливостей і потребує попереднього виявлення потенційно вразливих параметрів.

Фаззери та сканери параметрів (ffuf). Основна перевага ffuf полягає у надзвичайній швидкості та гнучкості для завдань розвідки: знаходження прихованих каталогів, файлів, віртуальних хостів та параметрів. Він є фундаментом для побудови складних workflow, де його результати передаються іншим інструментам (наприклад, ZAP чи Nuclei). Самостійно ffuf не виконує тестування на вразливості, що є його основним функціональним обмеженням.

Результати аналізу підтверджують, що жоден інструмент не забезпечує повного покриття всіх фаз та цілей тестування. Найбільш ефективним підходом є комбіноване використання інструментів у синергетичному ланцюжку:

1. **Етап розвідки:** Використання ffuf для знаходження цілей (ендпоінтів, параметрів).
2. **Первинне сканування:** Запуск Nuclei для швидкої перевірки знайдених цілей на відомі вразливості.
3. **Глибоке сканування та ручне тестування:** Запуск OWASP ZAP для детального аналізу критичних функцій додатку, тестування логіки та сесій.
4. **Експертне тестування:** Застосування sqlmap для поглибленої перевірки параметрів, підозрілих на SQL-ін'єкцію, виявлених на попередніх етапах.

Таке поєднання дає змогу максимально використати сильні сторони кожного інструменту, компенсуючи їхні слабкості, та забезпечити високий рівень покриття тестами.

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Проведене дослідження дозволило систематизувати підходи до автоматизації тестування на проникнення веб-додатків та визначити місце основних відкритих інструментів у цьому процесі. Автоматизоване тестування є критично важливою, але не самодостатньою складовою пентесту. Воно ефективно вирішує завдання масштабного пошуку типових вразливостей та розвідки, але не може повністю замінити аналіз складної бізнес-логіки, який вимагає участі фахівця.

Запропоновано поділ інструментів пентестингу на чотири категорії (активні сканери, сканери на шаблонах, спеціалізовані інструменти експлуатації, фаззери), які відображають фундаментальні відмінності в їх архітектурі та призначенні. OWASP ZAP залишається найбільш універсальним рішенням для глибинного аналізу, Nuclei – найшвидшим для масштабованого сканування на відомі загрози, sqlmap – найпотужнішим інструментом для всіх задач, пов’язаних з SQL-ін’єкціями, а ffuf – основним інструментом розвідки.

Критерій «Покриття OWASP Top-10» є ключовим, але недостатнім для вибору**.** Як показав аналіз, ефективність інструменту також суттєво залежить від підтримки

сучасних технологій (API, SPA), можливостей інтеграції в CI/CD, активної спільноти та ліцензійної чистоти. Максимальна ефективність досягається шляхом синергетичного комбінування інструментів у єдиному робочому процесі (workflow), де кожен наступний етап використовує результати попереднього. Оптимальний ланцюжок включає: розвідку (ffuf) -> швидке сканування на відомі вразливості (Nuclei) -> глибоке автоматизоване та ручне тестування (ZAP) -> експертну експлуатацію (спеціалізовані інструменти, наприклад, sqlmap).

Подальші дослідження доцільно зосередити на таких напрямках:

1. Розроблення метрик для кількісного порівняння продуктивності інструментів автоматизації пентестингу веб-додатків. Перспективним кроком виглядає створення стандартизованого тестового стенду (бенчмарку) з набором вразливих веб-додатків різних архітектур (моноліт, мікросервіси, SPA) з метою отримання порівняльних даних за метриками часу сканування, точності (Precision/Recall) *та* споживання ресурсів.

2. Інтегрування інструментів пентестингу в реальні конвеєри DevSecOps із подальшим дослідженням їх ефективності. Наприклад, впровадження зв'язок типу «Nuclei + GitHub Actions» або «ZAP + Jenkins» для виявлення типових проблем, вартості підтримки та реального впливу на безпеку продукту на ранніх стадіях розробки.

3. Аналіз інструментів нового покоління, що базуються на машинному навчанні. Зростає кількість рішень, які намагаються зменшити частку хибно-позитивних спрацювань або автоматизувати складені етапи тестування за допомогою інструментів машинного навчання. Системне порівняння таких інструментів з традиційними відкритими рішеннями є також перспективним завданням.

4. У контексті інтегрування інструментів пентестингу актуальним стає стандартизація формату обміну даними між ними На цей час відсутність єдиного формату для передачі результатів розвідки (від ffuf) сканерам (ZAP, Nuclei) створює практичну проблему. Дослідження та пропозиція такого стандарту могли б значно спростити створення автоматизованих конвеєрів (pipeline) тестування.

Таким чином, результати даної роботи формують теоретичну основу для обґрунтованого вибору інструментів пентестингу та побудови на їх базі конвеєра ефективних процесів автоматизованого тестування безпеки веб-додатків, а також визначають напрями для подальшого поглибленого вивчення цієї критично важливої сфери кібербезпеки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. OWASP Foundation. (2021). *OWASP Top Ten Web Application Security Risks*. OWASP. https://owasp.org/Top10/
2. OWASP Foundation. (2021). *OWASP Web Security Testing Guide (WSTG) Version 4.2*. OWASP. https://owasp.org/www-project-web-security-testing-guide/
3. OWASP Foundation. (2023). *OWASP Zed Attack Proxy (ZAP) – User Guide*. OWASP. https://www.zaproxy.org/docs/
4. ProjectDiscovery. (2024). *Nuclei Documentation*. ProjectDiscovery.io. https://docs.nuclei.sh/
5. sqlmap developers. (2024). *sqlmap: Automatic SQL injection and database takeover tool – User Manual*. sqlmap.org. http://sqlmap.org/
6. ffuf project. (2024). *ffuf – Fast web fuzzer written in Go – Documentation*. GitHub. https://github.com/ffuf/ffuf
7. Duchene, F., Rawat, S., Gupta, V., & Balzarotti, D. (2018). *A Case Study of Automated Penetration Testing: The Status Quo and Future Challenges*. In Proceedings of the 13th International Conference on Availability, Reliability and Security (ARES '18). ACM. https://doi.org/10.1145/3230833.3233284

8. Al-Saleh, M. I., & Espinoza, A. M. (2019). *A Survey on Web Application Vulnerability Scanning Tools*. International Journal of Advanced Computer Science and Applications (IJACSA), 10(5), 384-390. https://doi.org/10.14569/IJACSA.2019.0100549
9. Antunes, N., & Vieira, M. (2015). *Assessing and comparing vulnerability detection tools for web services: Benchmarking approach and examples*. IEEE Transactions on Services Computing, 8(2), 269-283. https://doi.org/10.1109/TSC.2013.2295792
10. Doupé, A., Cova, M., & Vigna, G. (2010). *Why Johnny Can't Pentest: An Analysis of Black-Box Web Vulnerability Scanners*. In Proceedings of the 7th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA '10). Springer. https://doi.org/10.1007/978-3-642-14215-4_11
11. Bau, J., Bursztein, E., Gupta, D., & Mitchell, J. (2010). *State of the Art: Automated Black-Box Web Application Vulnerability Testing*. In Proceedings of the 2010 IEEE Symposium on Security and Privacy (SP '10). IEEE. https://doi.org/10.1109/SP.2010.27
12. Garside, J. (2022). *Integrating Security into DevOps: A Survey of Open-Source Tools for Automated Penetration Testing*. Journal of Cybersecurity and Privacy, 2(3), 512-527. https://doi.org/10.3390/jcp2030026
13. Sokolov, V., Skladannyi, P., & Platonenko, A. (2023). *Jump-stay jamming attack on Wi-Fi systems*. In 2023 IEEE 18th International Conference on Computer Science and Information Technologies (CSIT) (pp. 1–5). IEEE. https://doi.org/10.1109/CSIT61576.2023.10324031
14. Воронцов, А. М., & Іванченко, І. В. (2021). *Сучасні методи та засоби тестування безпеки веб-застосунків*. Кібербезпека: освіта, наука, техніка, 2(14), 87–101. https://doi.org/10.28925/2663-4023.2021.14.87101

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 3 (31), 2025
ISSN 2663 - 4023

Volodymyr V. Khoma
Dr.Sc., Professor of the Department of Information Security
Lviv Polytechnic National University, Lviv, Ukraine
ORCID: 0000-0001-9391-6525
volodymyr.v.khoma@lpnu.ua

Andrii I. Partyka
Phd, Senior Lecturer of the Department of Information Security
Lviv Polytechnic National University, Lviv, Ukraine
ORCID: 0000-0003-3037-8373
andrii.i.partyka@lpnu.ua

Dmytro V. Sabodashko
Phd, Senior Lecturer of the Department of Information Security
Lviv Polytechnic National University, Lviv, Ukraine
ORCID: 0000-0003-1675-0976
dmytro.v.sabodashko@lpnu.ua

SYSTEMATISATION AND COMPARATIVE ANALYSIS OF AUTOMATED PENETRATION TESTING TOOLS FOR WEB APPLICATIONS

Abstract. The article examines modern approaches to automating the process of testing web applications for penetration. The growing number and complexity of web threats, as well as limited resources for manual testing, create a critical need for the effective use of automated testing tools. However, the diversity of architectures, operating principles, and functional capabilities of existing pentesting tools creates a problem of informed choice and effective combination. The purpose of the article is to systematise the main open source tools and conduct a comparative analysis based on a set of developed criteria. The paper proposes classifying tools into four categories: proxy-based active scanners (e.g., OWASP ZAP), template-based scanners (Nuclei), specialised exploitation tools (sqlmap), and fuzzers/parameter scanners (ffuf). For comparative analysis, a system of criteria was defined, including functional (OWASP Top-10 coverage, support for modern technologies), operational (integration into CI/CD, usability) and technical (licence, development activity) aspects. Based on these criteria, the tools representing each category were analysed. The results of the study showed that no tool is universal, and effectiveness depends on the specifics of the task. It was found that OWASP ZAP is the most universal for in-depth analysis, Nuclei is the fastest for scalable scanning, and ffuf is the most effective for reconnaissance. The key conclusion is a recommendation for the synergistic combination of tools in a single workflow for maximum coverage of the testing phases: from reconnaissance to deep exploitation. The results obtained form the theoretical basis for the informed selection of tools and the construction of effective automated security testing processes within the DevSecOps approaches.

Keywords: penetration testing; vulnerability; automated security scanning; OWASP ZAP; Nuclei; sqlmap; systematisation; comparative analysis; information security.

REFERENCES

1. OWASP Foundation. (2021). *OWASP Top Ten Web Application Security Risks*. OWASP. https://owasp.org/Top10/
2. OWASP Foundation. (2021). *OWASP Web Security Testing Guide (WSTG) Version 4.2*. OWASP. https://owasp.org/www-project-web-security-testing-guide/
3. OWASP Foundation. (2023). *OWASP Zed Attack Proxy (ZAP) – User Guide*. OWASP. https://www.zaproxy.org/docs/
4. ProjectDiscovery. (2024). *Nuclei Documentation*. ProjectDiscovery.io. https://docs.nuclei.sh/
5. sqlmap developers. (2024). *sqlmap: Automatic SQL injection and database takeover tool – User Manual*. sqlmap.org. http://sqlmap.org/

6. ffuf project. (2024). *ffuf – Fast web fuzzer written in Go – Documentation*. GitHub. https://github.com/ffuf/ffuf
7. Duchene, F., Rawat, S., Gupta, V., & Balzarotti, D. (2018). *A Case Study of Automated Penetration Testing: The Status Quo and Future Challenges*. In Proceedings of the 13th International Conference on Availability, Reliability and Security (ARES '18). ACM. https://doi.org/10.1145/3230833.3233284
8. Al-Saleh, M. I., & Espinoza, A. M. (2019). *A Survey on Web Application Vulnerability Scanning Tools*. International Journal of Advanced Computer Science and Applications (IJACSA), 10(5), 384-390. https://doi.org/10.14569/IJACSA.2019.0100549
9. Antunes, N., & Vieira, M. (2015). *Assessing and comparing vulnerability detection tools for web services: Benchmarking approach and examples*. IEEE Transactions on Services Computing, 8(2), 269-283. https://doi.org/10.1109/TSC.2013.2295792
10. Doupé, A., Cova, M., & Vigna, G. (2010). *Why Johnny Can't Pentest: An Analysis of Black-Box Web Vulnerability Scanners*. In Proceedings of the 7th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA '10). Springer. https://doi.org/10.1007/978-3-642-14215-4_11
11. Bau, J., Bursztein, E., Gupta, D., & Mitchell, J. (2010). *State of the Art: Automated Black-Box Web Application Vulnerability Testing*. In Proceedings of the 2010 IEEE Symposium on Security and Privacy (SP '10). IEEE. https://doi.org/10.1109/SP.2010.27
12. Garside, J. (2022). *Integrating Security into DevOps: A Survey of Open-Source Tools for Automated Penetration Testing*. Journal of Cybersecurity and Privacy, 2(3), 512-527. https://doi.org/10.3390/jcp2030026
13. Sokolov, V., Skladannyi, P., & Platonenko, A. (2023). *Jump-stay jamming attack on Wi-Fi systems*. In 2023 IEEE 18th International Conference on Computer Science and Information Technologies (CSIT) (pp. 1–5). IEEE. https://doi.org/10.1109/CSIT61576.2023.10324031
14. Vorontsov, A. M., & Ivanchenko, I. V. (2021). *Suchasni metody ta zasoby testuvannia bezpeky veb-zastosunkiv* [Modern methods and tools for web application security testing]. Cybersecurity: Education, Science, Technique, 2(14), 87–101. https://doi.org/10.28925/2663-4023.2021.14.87101