

Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра
на тему: «Проект підсистеми сканування безпеки веб-
додатків за допомогою програмного забезпечення
OWASP ZAP»

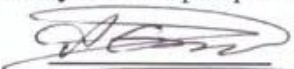
Виконав: студент 4-го курсу групи КБ-21-1
спеціальності 125 «Кібербезпека»

 Єлизавета ОПАНАСЕНКО

Керівник: Старший викладач кафедри КІ та КБ

 Олександра ПОКОТИЛО

Рецензент: Завідувач кафедри КІ та КБ, к.т.н., доцент

 Андрій ЄФІМЕНКО

Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки
Спеціальність 125 «Кібербезпека»
Освітня програма «Кібербезпека»

ЗАТВЕРДЖУЮ

Завідувач

кафедри комп'ютерної

інженерії та кібербезпеки

 Андрій Єфіменко

3 березня 2025 р.

ЗАВДАННЯ

на випускню кваліфікаційну роботу бакалавра

Студент: Опанасенко Єлизавета Романівна

Тема роботи: «Проект підсистеми сканування безпеки веб-додатків за допомогою програмного забезпечення OWASP ZAP»

затверджена Наказом університету від 3 березня 2025 р. № 80/с

Термін здачі студентом закінченої роботи 20 червня 2025 р.

Вихідні дані роботи (зазначається предмет і об'єкт дослідження):

Об'єкт дослідження: процес перевірки безпеки веб-додатків із використанням утиліти OWASP ZAP.

Предмет дослідження: методи та засоби тестування безпеки веб-додатків з використанням інструментарію OWASP ZAP.

Обладнання та програмне забезпечення:

1. Windows 10
2. Owasp ZAP
3. Google Chrome
4. XAMPP
5. DVWA

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Пірог О.В.	15.04.2025	15.04.2025
2	Пірог О.В.	20.05.2025	20.05.2025
3	Пірог О.В.	13.06.2025	13.06.2025

Студент Єлизавета ОПАНАСЕНКО

Керівник Олександра ПОКОТИЛО

Календарний план

№ з/п	Назви етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка завдання	10.04.2025	виконано
2	Розробка плану роботи	15.04.2025	виконано
3	Аналіз літературних джерел	18.04.2025	виконано
4	Огляд та встановлення утиліти OWASP ZAP	29.04.2025	виконано
5	Налаштування утиліти та дослідження її можливостей	01.05.2025	виконано
7	Встановлення цільового додатку	16.05.2025	виконано
8	Проведення сканування	20.05.2025	виконано
9	Моделювання атак	22.05.2025	виконано
10	Аналіз результатів тестування	26.05.2025	виконано
11	Оформлення пояснювальної записки	30.05.2025	виконано
12	Підготовка презентації	12.06.2025	виконано
13	Отримання відгуку керівника та рецензії	17.06.2025	виконано
14	Захист роботи	20.06.2025	виконано

Студент Єлизавета ОПАНАСЕНКО

Керівник Олександра ПОКОТИЛО

АНОТАЦІЯ

Кваліфікаційна робота представляє собою проєкт підсистеми сканування безпеки веб-додатків за допомогою програмного забезпечення Owasp Zар.

Метою роботи є розробка проєкту підсистеми сканування, яка дозволяє підвищити ефективність виявлення вразливостей та загальний рівень кібербезпеки веб-ресурсів. Безпека є першочерговим питанням у розробці веб-сайтів, оскільки вразливості можуть призвести до серйозних фінансових та репутаційних втрат. Розробники можуть допускати помилки при написанні коду або не усвідомлювати всю важливість використання прийомів безпечного програмування, що може призвести до появи вразливостей.

В рамках проєкту було проаналізовано найпоширеніші типи вразливостей, такі як SQL-ін'єкції та міжсайтовий скриптинг (XSS), а також сучасні підходи до тестування безпеки. Проведено сканування веб-сайту на наявність вразливостей та досліджено поведінку веб-сайту під їхнім впливом.

Пояснювальна записка складається зі вступу, трьох розділів, висновків, списку умовних скорочень, списку використаної літератури. Теоретична частина була викладена на 25 сторінках друкованого тексту. Список використаних джерел складається з 34 позиції та займає 5 сторінок. В роботі наведено 77 рисунки. Загалом обсяг роботи складає 84 сторінок.

КЛЮЧОВІ СЛОВА: ПЕНТЕСТИНГ, ТЕСТУВАННЯ, OWASP ZAP, ВЕБ-ДОДАТОК, СКАНУВАННЯ БЕЗПЕКИ, КІБЕРБЕЗПЕКА, ВРАЗЛИВІСТЬ, SQL-ІН'ЄКЦІЯ, XSS.

					KKIK.KP.B – 125 – 14 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Проект підсистеми сканування безпеки веб-додатків за допомогою програмного забезпечення OWASP ZAP	Лім.	Арк.	Аркушів
Розроб.		Опанасенко Є.Р.		18.06.25				
Керівник		Покотило О.А.		18.06.25			4	84
Реценз.		Єфіменко А.А.		18.06.25				
Н. контр.		Покотило О.А.		18.06.25				
Затверд.		Єфіменко А.А.		18.06.25				
						ФІКТ, Гр. КБ-21-1		

Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

ПОЯСНЮВАЛЬНА ЗАПИСКА

до випускної кваліфікаційної роботи бакалавра
на тему: «Проект підсистеми сканування безпеки веб-
додатків за допомогою програмного забезпечення
OWASP ZAP»

Виконав: студент 4-го курсу групи КБ-21-1
спеціальності 125 «Кібербезпека»

_____ Єлизавета ОПАНАСЕНКО

Керівник: Старший викладач кафедри КІ та КБ

_____ Олександра ПОКОТИЛО

Рецензент: Завідувач кафедри КІ та КБ, к.т.н., доцент

_____ Андрій ЄФІМЕНКО

Житомир – 2025

Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки
Спеціальність 125 «Кібербезпека»
Освітня програма «Кібербезпека»

ЗАТВЕРДЖУЮ
Завідувач
кафедри комп'ютерної
інженерії та кібербезпеки
_____ Андрій Єфіменко
3 березня 2025 р.

ЗАВДАННЯ

на випускню кваліфікаційну роботу бакалавра

Студент: Опанасенко Єлизавета Романівна

Тема роботи: «Проект підсистеми сканування безпеки веб-додатків за допомогою програмного забезпечення OWASP ZAP»

затверджена Наказом університету від 3 березня 2025 р. № 80/с

Термін здачі студентом закінченої роботи 20 червня 2025 р.

Вихідні дані роботи (зазначається предмет і об'єкт дослідження):

Об'єкт дослідження: процес перевірки безпеки веб-додатків із використанням утиліти OWASP ZAP.

Предмет дослідження: методи та засоби тестування безпеки веб-додатків з використанням інструментарію OWASP ZAP.

Обладнання та програмне забезпечення:

1. Windows 10
2. Owasp ZAP
3. Google Chrome
4. XAMPP
5. DVWA

Консультанти з випускної кваліфікаційної роботи із зазначенням розділів, що їх стосуються

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Пірог О.В.	15.04.2025	15.04.2025
2	Пірог О.В.	20.05.2025	20.05.2025
3	Пірог О.В.	13.06.2025	13.06.2025

Студент _____ Єлизавета ОПАНАСЕНКО

Керівник _____ Олександра ПОКОТИЛО

Календарний план

№ з/п	Назви етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка завдання	10.04.2025	виконано
2	Розробка плану роботи	15.04.2025	виконано
3	Аналіз літературних джерел	18.04.2025	виконано
4	Огляд та встановлення утиліти OWASP ZAP	29.04.2025	виконано
5	Налаштування утиліти та дослідження її можливостей	01.05.2025	виконано
7	Встановлення цільового додатку	16.05.2025	виконано
8	Проведення сканування	20.05.2025	виконано
9	Моделювання атак	22.05.2025	виконано
10	Аналіз результатів тестування	26.05.2025	виконано
11	Оформлення пояснювальної записки	30.05.2025	виконано
12	Підготовка презентації	12.06.2025	виконано
13	Отримання відгуку керівника та рецензії	17.06.2025	виконано
14	Захист роботи	20.06.2025	виконано

Студент _____ Єлизавета ОПАНАСЕНКО

Керівник _____ Олександра ПОКОТИЛО

АНОТАЦІЯ

Кваліфікаційна робота представляє собою проєкт підсистеми сканування безпеки веб-додатків за допомогою програмного забезпечення Owasp Zap.

Метою роботи є розробка проєкту підсистеми сканування, яка дозволяє підвищити ефективність виявлення вразливостей та загальний рівень кібербезпеки веб-ресурсів. Безпека є першочерговим питанням у розробці веб-сайтів, оскільки вразливості можуть призвести до серйозних фінансових та репутаційних втрат. Розробники можуть допускати помилки при написанні коду або не усвідомлювати всю важливість використання прийомів безпечного програмування, що може призвести до появи вразливостей.

В рамках проєкту було проаналізовано найпоширеніші типи вразливостей, такі як SQL-ін'єкції та міжсайтовий скриптинг (XSS), а також сучасні підходи до тестування безпеки. Проведено сканування веб-сайту на наявність вразливостей та досліджено поведінку веб-сайту під їхнім впливом.

Пояснювальна записка складається зі вступу, трьох розділів, висновків, списку умовних скорочень, списку використаної літератури. Теоретична частина була викладена на 25 сторінках друкованого тексту. Список використаних джерел складається з 34 позиції та займає 5 сторінок. В роботі наведено 77 рисунки. Загалом обсяг роботи складає 84 сторінок.

КЛЮЧОВІ СЛОВА: ПЕНТЕСТИНГ, ТЕСТУВАННЯ, OWASP ZAP, ВЕБ-ДОДАТОК, СКАНУВАННЯ БЕЗПЕКИ, КІБЕРБЕЗПЕКА, ВРАЗЛИВІСТЬ, SQL-ІН'ЄКЦІЯ, XSS.

					ККІК.КР.Б – 125 – 14 – ПЗ			
Змн.	Арк.	№ докум.	Підпис	Дата	Проект підсистеми сканування безпеки веб-додатків за допомогою програмного забезпечення OWASP ZAP	Літ.	Арк.	Аркушів
Розроб.		Опанасенко Є.Р.						
Керівник		Покотило О.А.						
Реценз.		Єфіменко А.А.				ФІКТ, Гр. КБ-21-1		
Н. контр.		Покотило О.А.						
Затверд.		Єфіменко А.А.						

ANNOTATION

The qualification work is a project for a web application security scanning subsystem using Owasp Zap software.

The aim of the work is to develop a scanning subsystem project that improves the effectiveness of vulnerability detection and the overall level of cybersecurity of web resources. Security is a top priority in website development, as vulnerabilities can lead to serious financial and reputational losses. Developers may make mistakes when writing code or fail to realize the importance of using secure programming techniques, which can lead to vulnerabilities.

The project analyzed the most common types of vulnerabilities, such as SQL injections and cross-site scripting (XSS), as well as modern approaches to security testing. The website was scanned for vulnerabilities and its behavior under their influence was investigated.

The explanatory note consists of an introduction, three sections, conclusions, a list of abbreviations, and a list of references. The theoretical part was presented on 25 pages of printed text. The list of references consists of 34 items and occupies 5 pages. The work contains 77 figures. In total, the work consists of 84 pages.

KEYWORDS: PENETRATION TESTING, TESTING, OWASP ZAP, WEB APPLICATION, SECURITY SCANNING, CYBERSECURITY, VULNERABILITY, SQL INJECTION, XSS.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ІЗ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ВЕБ-ДОДАТКІВ.....	10
1.1 Концепція сканування безпеки веб-додатків.....	10
1.1.1 Сканування безпеки: сутність, призначення	10
1.1.2 Підстави для проведення сканування безпеки та його об'єкти.....	11
1.1.3 Сучасні виклики та тенденції кіберзагроз у сфері веб-технологій	13
1.2 Веб-додатки як об'єкти дослідження.....	16
1.2.1 Сучасні веб-додатки, особливості взаємодії із ними.....	16
1.2.2 Сфери застосування веб-додатків та їх вразливості.....	20
1.3 Інструменти для сканування безпеки веб-додатків.....	24
1.3.1 Огляд сучасних рішень для аналізу безпеки	24
1.3.2 OWASP ZAP: функціональні можливості, модулі, розширення.....	27
1.3.3 Переваги та обмеження OWASP ZAP	31
РОЗДІЛ 2. АНАЛІЗ І ПІДГОТОВКА СЕРЕДОВИЩА ДЛЯ ПРОВЕДЕННЯ СКАНУВАННЯ БЕЗПЕКИ	35
2.1 Аналіз веб-додатків як об'єктів тестування.....	35
2.1.1 Поняття веб-додатків для пентестингу	35
2.1.2 Навчальна тестова платформа DVWA	37
2.2 Технічна підготовка інструментарію для проведення сканування	44
2.2.1 Встановлення ZAP та огляд інтерфейсу	44
2.2.2 Конфігурація параметрів проксі та сертифікатів.....	48
2.2.3 Підготовка локального середовища тестування	53
РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ПІДСИСТЕМИ СКАНУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ.....	56
3.1 Проведення ручного сканування на прикладі цільового веб-додатку	56
3.2. Виявлення вразливостей, їх аналіз та генерація звітів	61
3.2.1 Проведення сканування з використанням користувача.....	61
3.2.2. Створення звіту із результатів сканування.....	63
3.3. Впровадження підсистеми сканування в робоче середовище	66
3.4. Рекомендації щодо захисту веб-додатків та аналіз роботи	71
ВИСНОВКИ.....	78
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ	80

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API – Application Programming Interface
CI/CD – Continuous Integration/Continuous Delivery
CVE – Common Vulnerabilities and Exposures
CSS – Cascading Style Sheets
CSP – Content Security Policy
DAST – Dynamic Application Security Testing
DevSecOps – Development, Security, and Operations
DVWA – Damn Vulnerable Web Application
HIDS – Host-based Intrusion Detection System
HTTP – Hypertext Transfer Protocol
HTTPS – Hypertext Transfer Protocol Secure
HUD – Heads-Up Display
IoT – Internet of Things
JDK – Java Development Kit
JWT – JSON Web Token
MFA – Multi-Factor Authentication
MPA – Multi-Page Application
NVT – Network Vulnerability Tests
OWASP – Open Web Application Security Project
PWA – Progressive Web App
PHP – Hypertext Preprocessor
SQL – Structured Query Language
SSL – Secure Sockets Layer
SPA – Single-Page Application
TLS – Transport Layer Security
XDR – Extended Detection and Response
XSS – Cross-Site Scripting
ZAP – Zed Attack Proxy

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

ВСТУП

Актуальність теми. В епоху цифрових технологій веб-додатки стали ключовим елементом функціонування бізнесу та повсякденного життя. Фактично, більшість важливих сервісів, від онлайн-банкінгу до соціальних комунікацій та державних послуг, базуються на веб-інтерфейсах, через які щодня циркулюють величезні обсяги конфіденційної інформації. У цій ситуації зростання кіберзагроз, а також постійне ускладнення методів атак на веб-ресурси виводять питання їхньої безпеки на передній план. Недостатня увага до захисту веб-додатків може призвести до серйозних фінансових втрат, репутаційних ризиків та порушення приватності користувачів. Аналіз статистики кіберінцидентів демонструє невтішну тенденцію: значна частина успішних проникнень у корпоративні системи відбувається саме через слабкі місця у веб-додатках.

Зловмисники постійно вдосконалюють свої техніки, використовуючи як добре відомі вразливості, так і новітні, ще невідомі розробникам ("zero-day" експлойти). Різноманітність векторів атак, від SQL-ін'єкцій та міжсайтового скриптингу (XSS) до складних атак на рівні автентифікації та авторизації, вимагає комплексного підходу до забезпечення безпеки веб-додатків. У цьому контексті превентивне тестування безпеки веб-додатків перестає бути просто бажаною опцією чи формальністю, а стає нагальною необхідністю на кожному етапі їхнього життєвого циклу – від проектування до безпосередньої експлуатації.

Регулярне та систематичне сканування веб-додатків на наявність потенційних вразливостей дозволяє своєчасно ідентифікувати загрози та мінімізувати ризики серйозних наслідків кібератак. Виявлення слабких місць на ранніх стадіях розробки або експлуатації є значно ефективнішим та економічно вигіднішим, ніж усунення наслідків вже успішно проведеної атаки. Однак ефективність такого тестування безпосередньо залежить від якості використовуваних інструментів, глибини застосовуваних методик та кваліфікації фахівців, які його проводять. Важливо усвідомлювати, що безпека веб-додатків – це не одноразова процедура чи встановлення окремого програмного забезпечення, а безперервний, динамічний процес, який має супроводжувати всю їхню експлуатацію, враховуючи постійну появу нових загроз та оновлення технологій. Саме

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				8
Змн.	Арк.	№ докум.	Підпис	Дата		

тому дослідження ефективних методів та інструментів для виявлення вразливостей у веб-додатках є надзвичайно актуальним на сьогоднішній день.

Метою кваліфікаційної роботи є розробка проекту підсистеми сканування безпеки веб-додатків з використанням програмного забезпечення OWASP ZAP для підвищення ефективності виявлення вразливостей та покращення загального рівня кібербезпеки веб-ресурсів.

Визначена мета обумовлює наступні завдання:

- проаналізувати сучасні підходи та методи тестування безпеки веб-додатків;
- дослідити функціональні можливості та архітектуру програмного забезпечення OWASP ZAP;
- розробити архітектуру підсистеми сканування безпеки веб-додатків;
- провести тестування розробленої підсистеми та оцінити її ефективність;
- вивчити основні типи вразливостей веб-додатків;
- дослідити методи та підходи до усунення виявлених вразливостей;
- реалізувати програмне рішення для сканування з використанням функціональності OWASP ZAP;
- оцінити ефективність розробленого рішення.

Об'єктом дослідження є процес сканування безпеки веб-додатків.

Предметом дослідження є методи та засоби тестування безпеки веб-додатків з використанням інструментарію OWASP ZAP.

У процесі виконання кваліфікаційної роботи було розроблено проект підсистеми сканування безпеки веб-додатків, що забезпечує комплексне виявлення вразливостей з використанням різноманітних інструментів OWASP ZAP та формування детальних звітів з рекомендаціями щодо підвищення рівня захищеності досліджуваних веб-ресурсів.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 1. ТЕОРЕТИЧНІ ОСНОВИ ІЗ ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ ВЕБ-ДОДАТКІВ

1.1 Концепція сканування безпеки веб-додатків

1.1.1 Сканування безпеки: сутність, призначення

Сканування безпеки, відоме також як сканування вразливостей, є проактивним процесом у сфері кібербезпеки, спрямованим на виявлення слабких місць, недоліків або вразливостей в інформаційно-технологічній інфраструктурі, додатках, мережах чи системах організації, які можуть бути використані зловмисниками. Такі сканування відіграють критично важливу роль у зменшенні ризиків кібербезпеки та забезпеченні дотримання нормативних вимог. Використовуються різні методології та інструменти для методичного пошуку прогалин у безпеці, неправильних конфігурацій, застарілого програмного забезпечення, слабкого контролю доступу та інших проблем, які можуть скомпрометувати цифрове середовище організації.

Процес сканування безпеки розпочинається з визначення обсягу роботи: необхідно вирішити, які аспекти системи, такі як пристрої, діапазони IP-адрес, програмне забезпечення, тощо, підлягають скануванню. Сканування конфігурації є одним із загальноприйнятих підходів, що забезпечує відповідність параметрів системи заздалегідь визначеним політикам безпеки. Цей процес може сприяти організаціям у виконанні вимог щодо захисту даних, як це передбачено відомими законодавчими актами чи стандартами в галузі кібербезпеки[1].



Рисунок 1.1 – Методологія тестування безпеки веб-додатків

Існує кілька типів сканування безпеки, включаючи сканування мережі, сканування на основі хосту, сканування бездротової мережі та сканування програм. Кожен тип сканування спеціалізується на окремих областях. Сканування мережі має справу з виявленням IP-адрес і вразливостей у мережі, тоді як сканування на

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

основі хоста має справу з виявленням проблем безпеки на рівні системи чи пристрою. Тим часом сканування бездротових мереж пов'язане зі скануванням бездротових мереж на наявність вразливостей, а сканування програм зосереджено на оцінці безпеки кодів програм. Ці комплексні набори перевірок безпеки є критично важливими у сучасному цифровому середовищі, зокрема в бізнес-моделях, що включають великі мережі, які потребують жорстких заходів кібербезпеки. Сканування безпеки забезпечують організації необхідними інструментами для проактивного управління своїми кібербезпековими стратегіями, а не реагування на загрози.

Незважаючи на величезні переваги, які отримують користувачі від сканування безпеки, ними слід користуватися обережно. Багаторазове сканування може сповільнити продуктивність систем, а несанкціоноване сканування системи третьої сторони без дозволу власника може мати серйозні юридичні наслідки.

1.1.2 Підстави для проведення сканування безпеки та його об'єкти

Сканування безпеки може здійснюватися як вручну фахівцями, так і автоматично за допомогою спеціалізованих скануючих інструментів, які порівнюють системи з базами даних відомих вразливостей. Сканування може виконуватися з різних перспектив (внутрішньої або зовнішньої), з різними рівнями доступу (автентифікованим або неавтентифікованим) та з різною глибиною аналізу, відповідно до вимог безпеки організації. Основою процесу сканування системи безпеки в галузі кібербезпеки є можливість оцінювати як загальнодоступні, так і приватні комп'ютерні мережі на предмет виявлення потенційних зловмисних дій, крадіжки даних або шкідливих програм, які загалом відомі як зловмисне програмне забезпечення.

Сканування безпеки значною мірою впливають на захист від вірусів. Антивірусне програмне забезпечення викорінює зловмисне програмне забезпечення, виконуючи ретельне сканування безпеки сканованих дисків або пам'яті операційної системи протягом певних періодів часу. Коли ми активуємо захист у режимі реального часу, антивірусне програмне забезпечення може

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

виконувати активне сканування безпеки щоразу, коли дані змінюються або доступ до них здійснюється.

Об'єкти, що скануються:

- Мережева інфраструктура (маршрутизатори, комутатори, брандмауери)
- Серверні середовища (фізичні, віртуальні, хмарні)
- Кінцеві пристрої (робочі станції, ноутбуки, мобільні пристрої)
- Програми (веб, мобільні, настільні)
- Бази даних і системи зберігання даних
- Пристрої IoT та операційна технологія
- Контейнерні середовища та мікросервіси
- Хмарні середовища та сервіси

Сканування безпеки не обмежується лише перевіркою конфігурації. Воно також включає оцінку потенційних вразливостей, які можуть дозволити несанкціонований доступ до системи, залишаючи це непоміченим для користувачів. Такі вразливості можуть бути наслідком помилок у дизайні, недбалості користувачів або відсутності оновлень безпеки й необхідних патчів. Після виявлення вразливостей їх зазвичай оцінюють за допомогою спеціальних систем ранжування. Зазвичай це базується на ймовірності та потенційній серйозності експлойту, що сприяє компаніям та окремим особам у визначенні пріоритетності виправлень і інших важливих заходів безпеки[2].

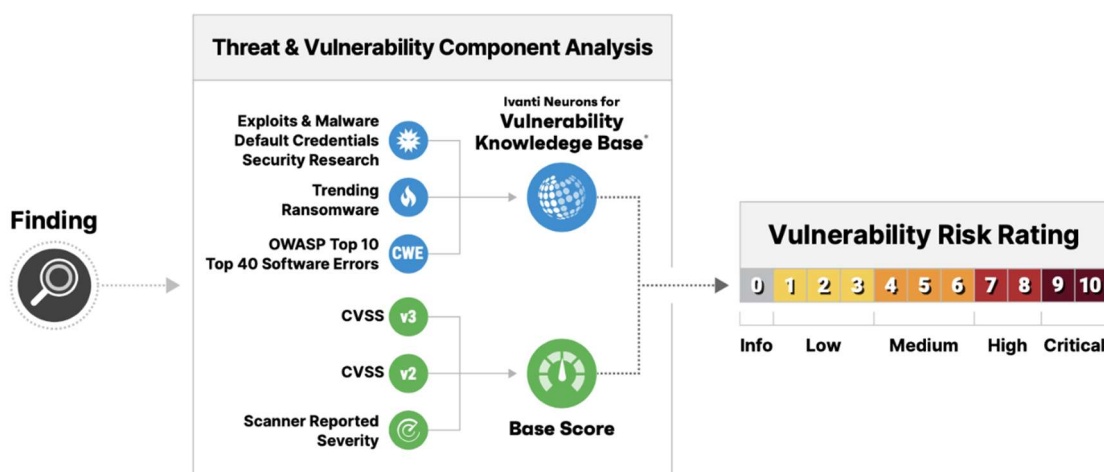


Рисунок 1.2 - Що таке рейтинг ризику вразливості?

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

Серед типових загроз, які виявляються під час такого сканування, найбільш поширеними є неправильно налаштовані параметри безпеки, слабкі паролі, використання застарілого програмного забезпечення з відомими вразливостями, наявність несанкціонованих пристроїв і додатків, а також некоректно виставлені дозволи на файли і каталоги.

Слід зазначити, що сканування безпеки може бути визнане ефективним лише за умови, що організація систематично вживає заходів для вирішення виявлених проблем у сфері безпеки. У рамках стратегії швидкого реагування, багато доступних інструментів сканування також оснащені системами виявлення та запобігання вторгненням. Вони ефективно захищають мережі та системи, аналізуючи загрозливі атаки та блокуючи потенційно шкідливі дії[33]. Зокрема, перевірка безпеки виявляє неправильно налаштовані мережі, які можуть стати вразливими точками для хакерів, що намагаються отримати доступ до системи. Для запобігання цій загрозі, організаціям необхідно терміново виправити конфігурації після їх виявлення. Невірна конфігурація може проявлятися в різних формах, таких як: неактивні бази даних, які залишаються відкритими, системи з заводськими налаштуваннями, незахищені архіви даних, а також користувачі, які мають зайві адміністративні права.

1.1.3 Сучасні виклики та тенденції кіберзагроз у сфері веб-технологій

Сфера цифрових технологій сьогодні стикається з небаченою раніше кількістю та складністю кіберзагроз, що пояснюється як збільшенням числа пристроїв, з'єднаних в Інтернеті, так і постійним удосконаленням інструментів атаки. Зокрема, розширення концепції Інтернету речей (ІоТ) створює нові вектори для проникнення всередину систем, а зловмисники дедалі частіше використовують автоматизовані методи для пошуку вразливостей у веб-додатках, серверах та API. В той самий час, сучасне глобальне суспільство є високопідключеним, що сприяє швидкому поширенню нових видів атак, серед яких особливо небезпечними залишаються цілеспрямовані атаки на інфраструктуру, фішинг, шкідливе програмне забезпечення та складні багатовекторні кампанії.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

Окрему роль у сучасних викликах відіграє поява складних атак, які важко виявити традиційними методами. Зокрема, класичні системи виявлення загроз, що працюють на основі сигнатур, демонструють обмежену ефективність щодо нових або модифікованих загроз, оскільки їм властива залежність від регулярного оновлення баз даних та неможливість ідентифікувати невідомі патерни поведінки. Це зумовлює необхідність впровадження інноваційних підходів до аналізу та захисту, які дозволяють оперативно реагувати на еволюцію загроз. Для кращого розуміння особливостей кібербезпеки важливо дослідити основні напрями еволюції та ключові тенденції розвитку кібербезпеки:

- Використання штучного інтелекту для аналізу великих обсягів даних і виявлення аномалій забезпечує швидку і точну ідентифікацію можливих загроз. А разом із машинним навчанням дозволяє вдосконалити методи виявлення загроз і знизити кількість хибних тривог завдяки глибокому аналізу даних.

- Посилення захисту даних у хмарних платформах та використання спеціалізованих інструментів для моніторингу та захисту вже встановлюється як стандарт для компаній, які прагнуть забезпечити надійну безпеку своєї інформації.

- Технологія розширеного виявлення та реагування (XDR) об'єднує різноманітні інструменти кібербезпеки в єдину систему, що сприяє ефективному визначенню загроз у складних середовищах.

- Модель "нульової довіри" (Zero Trust) передбачає постійний контроль доступу користувачів і пристроїв незалежно від місця їхнього розташування, що дозволяє досягти максимальної безпеки.

- Автоматизація процесів у сфері кіберзахисту допомагає суттєво скоротити час реагування на потенційні загрози завдяки швидкому моніторингу й аналізу.

- Підготовка працівників до розпізнавання кіберзагроз, включаючи запобігання фішинговим атак і дотримання політик безпеки, є ключовим заходом.

- Інтеграція практик безпеки на ранніх етапах розробки (DevSecOps) забезпечує виявлення вразливостей ще до запуску продукту, зменшуючи ризики у подальшій експлуатації.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

- DDoS-атаки залишаються поширеною загрозою, і для їхнього відбиття використовуються спеціалізовані сервіси та хмарні рішення, які забезпечують стабільну роботу сайтів навіть під час масованих атак.

- Перехід до безперервного моніторингу вразливостей замість періодичного сканування сприяє оперативнішій реакції на загрози та підвищенню ефективності ризик-менеджменту.

- Інтеграція автоматизованих систем усунення недоліків із процесами сканування дозволяє оперативно вирішувати проблеми, пов'язані з поширеними вразливостями.

- Спеціалізація на захисті контейнерів і Kubernetes забезпечує надійну безпеку сучасних додатків завдяки адаптованим технологіям сканування контейнерних середовищ.

- Прогресивні методи оцінювання безпеки в середовищах IoT та OT дозволяють ефективніше захищати мережеві пристрої й операційні технології у контексті їхнього стрімкого розвитку.

- Посилене тестування безпеки API включає ретельний аналіз кінцевих точок як критично важливих компонентів цифрової інфраструктури[3].



Рисунок 1.3 – Майбутні тренди у сфері кібербезпеки

У зв'язку зі зростанням складності й витонченості кібератак компанії змушені впроваджувати більш ефективні методи перевірки безпеки. Такі підходи не лише сприяють виявленню вразливостей у системах, але й забезпечують їхнє

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

швидке усунення. Це дозволяє захистити дані, підтримати репутацію підприємств і уникнути фінансових збитків. Тож сучасна кібербезпека орієнтована на розробку комплексних рішень, які здатні гарантувати найвищий рівень захисту в умовах динамічного цифрового середовища.

1.2 Веб-додатки як об'єкти дослідження

1.2.1 Сучасні веб-додатки, особливості взаємодії із ними

Веб-додатки можна поділити за їхнім архітектурним підходом до доставки контенту та взаємодії з користувачем.

Односторінкові додатки (SPA) базуються на концепції динамічного перезавантаження контенту без повного оновлення сторінки. Вся логіка додатка виконується в браузері, а сервер надає лише API для обміну даними. Такі додатки забезпечують швидку взаємодію користувача з інтерфейсом після початкового завантаження, але можуть мати проблеми з SEO та великим розміром початкового пакета коду.

Багатосторінкові додатки (MPA) представляють традиційний підхід, де кожна сторінка генерується окремо на сервері і повністю перезавантажується при навігації. Це забезпечує кращу індексацію пошуковими системами та менше навантаження на клієнтський пристрій, але може призводити до повільнішої навігації між сторінками.

Прогресивні веб-додатки (PWA) поєднують веб-технології з можливостями нативних мобільних додатків. Вони працюють офлайн завдяки Service Workers, можуть бути встановлені на пристрій користувача та надсилати push-повідомлення. PWA забезпечують швидкість завантаження та надійність роботи навіть при поганому інтернет-з'єднанні.

Додатки на основі мікросервісів розбивають функціональність на незалежні сервіси, кожен з яких має власну базу даних та логіку. Цей підхід дозволяє різним командам працювати над окремими компонентами, забезпечує горизонтальне масштабування та технологічну різноманітність, але вимагає складнішої інфраструктури та координації.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				16
Змн.	Арк.	№ докум.	Підпис	Дата		

Модель доступності та розповсюдження веб-додатків суттєво впливає на вимоги до їхньої безпеки та підходи до управління ними.

Загальнодоступні додатки призначені для широкої аудиторії через відкритий інтернет. Вони мають публічні URL-адреси, доступні для всіх користувачів та часто монетизуються через рекламу або електронну комерцію.

Інтранет-додатки функціонують у внутрішній мережі організації та доступні лише співробітникам. Вони забезпечують внутрішні бізнес-процеси, корпоративну комунікацію та управління ресурсами з підвищеним рівнем безпеки.

Екстранет-додатки надають обмежений доступ зовнішнім партнерам, постачальникам або клієнтам до певних внутрішніх ресурсів організації. Такі системи забезпечують контрольований обмін інформацією між компанією та її діловими партнерами.

Технологічна реалізація веб-додатків відображає еволюцію парадигм розробки.

Традиційні серверні додатки виконують більшість логіки на сервері, генеруючи HTML-сторінки для відправки в браузер. Цей підхід забезпечує централізоване управління даними та логікою, але може бути менш інтерактивним для користувача.

Клієнтські JavaScript-додатки переносять основну логіку в браузер, використовуючи JavaScript-фреймворки для створення динамічних інтерфейсів. Сервер слугує лише як API для надання даних, що дозволяє створювати більш зручні користувацькі інтерфейси.

Гібридні додатки поєднують серверний та клієнтський рендеринг, дозволяючи розробникам використовувати переваги обох підходів. Початкова сторінка може рендеритися на сервері для швидкого завантаження, а подальша взаємодія відбувається через клієнтський код.

Безсерверні додатки використовують хмарні функції, які виконуються за потреби без постійно працюючих серверів. Цей підхід забезпечує автоматичне масштабування, зменшені операційні витрати та оплату лише за фактичне використання ресурсів.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

Відкриті веб-ресурси - це загальнодоступні цифрові активи, які функціонують за принципами відкритості, доступності та сумісності. Їхня структура та робота мають низку характеристик, що визначають взаємодію користувачів із контентом, а також їхню роль у ширшій екосистемі Інтернету[4].

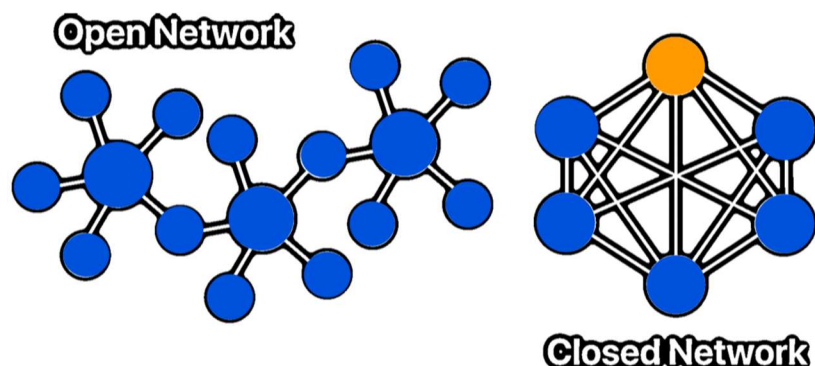


Рисунок 1.4 – Відкриті та закриті мережі

Необмежений доступ і доступність є основоположною рисою таких ресурсів. Вони орієнтовані на глобальний масштаб використання та мінімізують бар'єри доступності. Завдяки використанню відкритих веб-протоколів і уникненню пропрієтарних технологій контент залишається доступним на різних пристроях та операційних системах без вимог до спеціалізованого програмного забезпечення.

Розподілена інфраструктура та відмовостійкість забезпечують необхідну стабільність відкритої веб-екосистеми. На відміну від централізованих платформ, такі ресурси часто використовують розподілені хостингові мережі та мережі доставки контенту. Це покращує продуктивність для глобальних користувачів і мінімізує затримки, навіть у разі сплеску трафіку чи часткових збоїв.

Гіперпосилання та взаємозв'язок є ключовими елементами, які формують когнітивний фундамент відкритого Інтернету. Їх основне завдання — створення зв'язків між різними частинами контенту та веб-сайтами, що забезпечує користувачам свободу переміщення між ресурсами та відкриває нові інформаційні маршрути. Пошукові системи та веб-сканери постійно індексують ці зв'язки, полегшуючи доступ до відкритих ресурсів через пошукові механізми.

Реалізація на основі стандартів забезпечує сумісність і довговічність ресурсів. Відкриті веб-ресурси зазвичай відповідають встановленим технічним стандартам, забезпечуючи інтеграцію між різними платформами. Це дозволяє

контенту зберігати свою функціональність, незалежно від технологій, які використовуються для його створення. Основні стандарти включають HTML для організації структури, CSS для оформлення, JavaScript для інтерактивності та HTTP/HTTPS для передачі даних — усі вони базуються на прозорих і публічно задокументованих специфікаціях.

Технології динамічної генерації контенту змінюють статичні ресурси на адаптивне середовище. Сучасні веб-ресурси використовують серверну обробку для створення контенту, адаптованого до потреб користувача, його вподобань або актуальних даних. Це забезпечує персоналізацію без втрати відкритості ресурсу. Адаптивний дизайн автоматично змінює вигляд і функціональність ресурсу відповідно до розміру екрану й характеристик пристроїв.

Інтерактивність та внесок користувачів виходять за рамки пасивного споживання контенту. Відкриті веб-ресурси часто включають механізми зворотного зв'язку, коментарів, створення користувацького контенту чи спільного редагування. Такі платформи роблять контент активним та живим завдяки колективному розвитку. Однак для забезпечення якості й запобігання зловживанням потрібні ефективні механізми модерації та управління.

Аналітичні інструменти сприяють оптимізації й постійному вдосконаленню. Збір даних щодо особливостей користування, ефективності контенту чи поведінки користувачів дозволяє власникам ресурсів приймати обґрунтовані рішення щодо розвитку й оптимізації роботи платформ при цьому враховуючи конфіденційність. Наприклад, аналіз часу завантаження сторінок, рівня взаємодії та конверсії допомагає спрямовувати зусилля на ключові аспекти вдосконалення.

Моделі безпеки та довіри захищають як інфраструктуру, так і користувачів, незважаючи на відкритий доступ. Впроваджують різні механізми безпеки, зокрема шифрування SSL/TLS для захисту передачі даних, політики безпеки контенту для захисту від міжсайтових скриптових атак і політики CORS, що регулюють взаємодію контенту між доменами. Сигнали довіри, такі як індикатори HTTPS,

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

політики конфіденційності та прозора атрибуція, допомагають користувачам оцінити легітимність ресурсу та стан безпеки.

Проблеми довгострокового збереження контенту залишаються значним викликом для стійкості Інтернету. Відкриті ресурси часто стикаються з гниттям посилань, яке виникає при зникненні сайтів чи сторінок, на які вони вказують, а також дрейфом контенту — суттєвими змінами матеріалів із часом, які можуть порушити структуру взаємозв'язків. Систематичне архівування веб-контенту стає важливим інструментом для вирішення цих проблем. Проте комплексних рішень поки що не існує через масштаби й динамічний характер сучасного Інтернету[16].

1.2.2 Сфери застосування веб-додатків та їх вразливості

Таблиця 1.1

Сфери застосування веб-додатків

Категорія	Опис	Основні напрямки застосування
Додатки, орієнтовані на вміст (Drupal, WordPress, Vimeo)	Платформи, орієнтовані на створення, управління та розповсюдження інформації	- Корпоративні сайти та портали - Видання новин та ЗМІ - Сховища освітнього контенту - Урядові інформаційні ресурси - Системи документації
Додатки для спільної роботи (Slack, Microsoft Teams, Trello)	Інструменти для командної роботи та спільної продуктивності	- Системи управління проектами - Платформи для спільної роботи з документами - Системи календаря та планування - Віртуальні робочі простори
Додатки для соціальних мереж (Facebook, Twitter, Reddit)	Платформи, що об'єднують користувачів і полегшують взаємодію	- Загальні соціальні мережі - Професійні мережеві платформи - Послуги для знайомств і стосунків - Спільноти для обміну контентом - Форуми та дискусійні дошки
Освітні програми (Moodle, Canvas, Kahoot)	Системи для навчання, викладання та передачі знань	- Віртуальні класні середовища - Платформи освітнього контенту - Системи оцінювання - Академічна співпраця - Додатки для вивчення мов
Фінансові програми (PayPal, Mint, Stripe)	Системи, що обробляють грошові транзакції та управління ними	- Банківські інтерфейси - Інвестиційні платформи - Гаманці та особисті фінанси - Послуги з обробки платежів - Інструменти планування фінансів
IoT та Smart Home додатки	Платформи для управління та моніторингу підключених пристроїв та розумних систем	- Системи розумного дому - Промислові IoT платформи - Платформи моніторингу сенсорів - Системи автоматизації - Енергетичний менеджмент

1	2	3
Додатки для охорони здоров'я (Teladoc, WebMD, Zocdoc, Practice Fusion)	Платформи для підтримки медичних послуг та управління охороною здоров'я	- Системи електронних медичних карток (ЕМК) - Телемедичні платформи - Портали пацієнтів - Додатки для моніторингу здоров'я - Управління медичною практикою - Керування фармацевтичними препаратами та рецептами
Додатки для уряду та державних служб (USA.gov, IRS.gov, Census.gov)	Системи, що сприяють громадській активності та залученню державних ресурсів	- Публічні документи, інформаційні системи - Портали для обслуговування громадян - Платформи для голосування та громадської участі - Системи сповіщення про надзвичайні ситуації - Послуги громадського транспорту
Аналітичні додатки	Системи для збору, обробки та візуалізації даних з метою прийняття бізнес-рішень	- Платформи бізнес-аналітики - Системи веб-аналітики - Інструменти візуалізації даних - Системи звітності - Платформи прогнозової аналітики

Вкрай важливо глибоко розуміти вразливості веб-ресурсів. Це знання є ключовим для всіх, хто займається розробкою, підтримкою або захистом веб-додатків. У цьому контексті заслуговують на увагу найбільш актуальні загрози, визначені галузевими стандартами, такими як OWASP Top 10. Їх детальне вивчення дає змогу зрозуміти, як такі уразливості можуть призвести до серйозних наслідків: витоку даних, порушення цілісності систем або навіть відмови в обслуговуванні. Без належного розуміння цих аспектів навіть найнадійніші системи захисту можуть виявитися неспроможними[5].

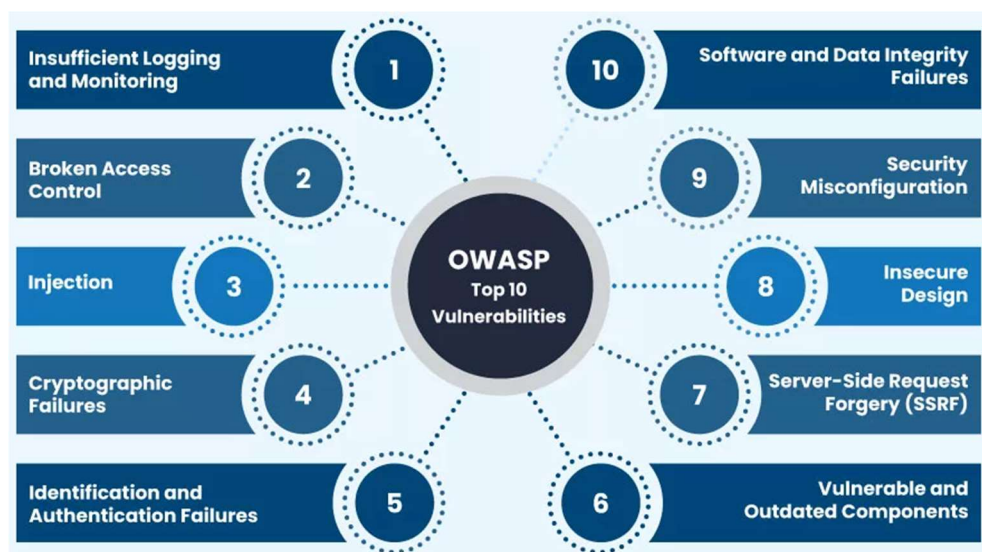


Рисунок 1.5 – Топ 10 загроз OWASP ZAP

- SQL-ін'єкції (SQL Injection)

SQL-ін'єкція залишається однією з найнебезпечніших та найпоширеніших вразливостей веб-додатків. Ця атака виникає, коли зловмисники вставляють шкідливий SQL-код в поля введення, які не були належним чином валідовані або санітизовані перед передачею до бази даних. Атакуючі можуть маніпулювати SQL-запитами для отримання несанкціонованого доступу до конфіденційних даних, зміни або видалення інформації з бази даних, а в деяких випадках навіть виконання системних команд на сервері бази даних.

- Неправильна конфігурація безпеки (Security Misconfiguration)

Включає незабезпечені налаштування за замовчуванням, неповні або спеціальні конфігурації, відкриті хмарні сховища, неправильно налаштовані HTTP заголовки, детальні повідомлення про помилки та невикористовувані функції безпеки. Ці проблеми можуть виникати на будь-якому рівні технологічного стеку додатка.

- Порушення контролю доступу (Broken Access Control)

Ця категорія включає широкий спектр проблем, від простого обходу авторизації до складних схем привілейованого ескалації. Атакуючі можуть отримати доступ до функцій адміністратора, переглядати або змінювати чужі облікові записи, отримувати доступ до конфіденційних документів або виконувати дії, для яких у них немає відповідних прав. Особливо небезпечним є вертикальне ескалація привілеїв, коли звичайний користувач отримує права адміністратора, та горизонтальне ескалація, коли користувач може отримати доступ до ресурсів інших користувачів того ж рівня.

- Порушена автентифікація (Broken Authentication)

Охоплює широкий спектр проблем, пов'язаних з неправильною реалізацією функцій ідентифікації користувачів та управління сесіями. Ці вразливості дозволяють атакуючим компрометувати паролі, ключі або сесійні токени, або тимчасово чи постійно брати на себе ідентичність інших користувачів. Проблеми можуть виникати через слабкі паролі, неправильне зберігання автентифікаційних даних, вразливості у відновленні паролів або неадекватне управління сесіями.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		22

- Вразливі та застарілі компоненти (Vulnerable and Outdated Components)

Використання компонентів з відомими вразливостями є проблемою, оскільки сучасні додатки інтенсивно використовують сторонні бібліотеки, фреймворки та інші програмні компоненти. Ускладнюється усе тим, що багато організацій не знають версій усіх компонентів, які вони використовують, включаючи вкладені залежності. Багато відомих порушень безпеки були результатом експлуатації вразливостей у бібліотеках та фреймворках, які не оновлювалися роками.

- Криптографічні збої (Cryptographic Failures)

Виникають через неправильне використання або повну відсутність криптографічного захисту конфіденційних даних. Ця категорія включає широкий спектр проблем - від передачі даних у відкритому вигляді до використання слабких криптографічних алгоритмів, неправильного управління ключами та порушень у реалізації криптографічних протоколів.

- Міжсайтове скриптування (Cross-Site Scripting, XSS)

XSS-атаки дозволяють зловмисникам впроваджувати шкідливий JavaScript-код на веб-сторінки, який потім виконується в браузерах інших користувачів. Ці атаки можуть використовуватися для крадіжки cookies та сесійних токенів, перенаправлення користувачів на шкідливі сайти, збору конфіденційної інформації або виконання дій від імені жертви.

- Розкриття конфіденційних даних (Sensitive Data Exposure)

Воно відбувається, коли додатки не забезпечують належний захист чутливої інформації, такої як фінансові дані, медичні записи, персональні ідентифікатори або автентифікаційні дані. Це може статися через відсутність шифрування даних при передачі або зберіганні, використання слабких криптографічних алгоритмів, неправильне управління ключами шифрування або випадкове включення конфіденційних даних у логи або повідомлення про помилки.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		23

- Недостатнє логування та моніторинг (Insufficient Logging & Monitoring)

Це дозволяють атакам залишатися непоміченими протягом тривалого часу, що дає зловмисникам можливість завдати більшої шкоди та ускладнює розслідування інцидентів безпеки. Багато організацій не ведуть належних логів подій безпеки, не моніторять підозрілу активність або не мають ефективних процедур реагування на інциденти[33].

1.3 Інструменти для сканування безпеки веб-додатків

1.3.1 Огляд сучасних рішень для аналізу безпеки

Обрання надійного та продуктивного сканера вразливостей відіграє критично важливу роль у забезпеченні безпеки ваших веб-додатків і серверів. Влучно підібраний інструмент здатен істотно підвищити ефективність процесів виявлення та усунення проблем безпеки, що дозволяє значно мінімізувати ризики кібератак. У динамічному світі кіберзагроз, який постійно розвивається, використання перевірених і потужних механізмів сканування стає одним із головних елементів для захисту інформаційних активів.

1. OpenVAS



Рисунок 1.6 – Логотип OpenVAS

OpenVAS (Open Vulnerability Assessment System) — це потужний відкритий сканер вразливостей, який використовується для комплексного аналізу безпеки мереж, систем і веб-додатків. Має велику базу даних з понад 50 000 мережевих тестів вразливостей (NVT), яка регулярно оновлюється та надає детальні звіти з рекомендаціями щодо усунення знайдених проблем. Система підтримує сканування різних високорівневих і низькорівневих інтернет-протоколів, а також промислових протоколів, що робить її універсальним інструментом для оцінки безпеки в різних середовищах. OpenVAS має масштабовану архітектуру, що дозволяє проводити сканування як невеликих, так і великих корпоративних мереж

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		24

із можливістю налаштування продуктивності для оптимізації часу перевірки. Програма є відкритим програмним забезпеченням, що робить її доступною для широкого кола користувачів і дозволяє адаптувати під специфічні потреби організації[6].

2. Nmap



Рисунок 1.7 – Логотип Nmap

Nmap (Network Mapper) — це багатофункціональний та надзвичайно гнучкий інструмент з відкритим кодом для аналізу мереж. Його основне завдання — виконувати мапування мереж, виявляти активні пристрої, визначати відкриті порти і служби, а також аналізувати їхні конфігурації. Nmap підтримує різні техніки сканування (TCP SYN, TCP connect, UDP тощо) та оснащений потужним скриптовим рушієм (NSE — Nmap Scripting Engine), що дозволяє автоматизувати пошук вразливостей та адаптувати його до різних потреб і середовищ. Nmap підтримує роботу на таких операційних системах, як Windows, Linux і macOS, і є широко визнаним стандартом у сфері пентестингу мереж та оцінки безпеки IT-інфраструктури. Його було представлено в сотнях журнальних статей, кількох фільмах, десятках книг та одній серії коміксів[7].

3. Wazuh



Рисунок 1.8 – Логотип Wazuh

Wazuh — це потужна платформа для моніторингу безпеки та управління подіями, яка об'єднує в собі функції системи виявлення вторгнень (HIDS), аналізу

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		25

журналів, управління безпековими політиками та сканування вразливостей. Wazuh створює докладні звіти, його можна легко розширити за допомогою додаткових модулів та інтегрувати з іншими інструментами безпеки. Агенти Wazuh здатні проводити регулярний аналіз кінцевих точок, визначаючи слабкі місця за допомогою зіставлення з базою CVE. Платформа використовує передові механізми кореляції та дослідження загроз для покращення точності їх виявлення, а завдяки масштабованій архітектурі, Wazuh може використовуватися як у малих, так і у великих середовищах, локально чи у хмарі. Система інтегрується з Elastic Stack для візуалізації та аналізу даних[8].

4. Nikto



Рисунок 1.9 – Логотип Nikto

Nikto – безкоштовний інструмент з відкритим вихідним кодом для виявлення вразливостей на веб-серверах. Утиліта належить до категорії сканерів типу "чорний ящик" (blackbox), що означає відсутність знань про внутрішню будову програми чи сайту з фокусом на перевірку функціональності. Nikto здатний виявляти понад 6700 різних потенційно небезпечних файлів та вразливостей. База даних сканера регулярно оновлюється з додаванням нових уразливостей. Окрім пошуку вразливостей, інструмент визначає застарілі версії бібліотек та фреймворків, що використовуються. Сканер може працювати з HTTP та HTTPS протоколами, підтримує автентифікацію через різні методи та може сканувати через проксі-сервери. Важливо зазначити, що Nikto не є стелс-сканером, адже методика роботи передбачає завершення TCP-з'єднання, через що процес сканування не залишається непомітним. У логах вебсайту або у системах виявлення вторгнень, якщо вони активні, буде зафіксовано факт сканування[9].

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		26

5. ClamAV



Рисунок 1.10 – Логотип ClamAV

ClamAV — це антивірусна система, яка спеціалізується на виявленні шкідливого програмного забезпечення, вірусів, троянських програм та експлойтів у різних типах файлів. Основне призначення ClamAV — забезпечення антивірусного захисту, однак інструмент також здатен ідентифікувати потенційно небезпечні елементи у файловій системі, виконуючи базове сканування на вразливості. Його часто використовують для перевірки вкладень на поштових серверах або аналізу файлів, що передаються через веб-додатки. У пакунок входить масштабований багатонитевий демон `clamd`, керований з командного рядка сканер `clamscan`, а також модуль оновлення сигнатур з Інтернету `freshclam`. Завдяки підтримці численних форматів файлів та простій інтеграції з іншими сервісами, ClamAV є ефективним компонентом загальної інфраструктури кіберзахисту[10].

1.3.2 OWASP ZAP: функціональні можливості, модулі, розширення

OWASP ZAP (Zed Attack Proxy) — це безкоштовний інструмент для сканування безпеки веб-додатків із відкритим кодом, створений і підтримуваний Open Web Application Security Project (OWASP). Він належить до найбільш популярних у світі засобів забезпечення безпеки, призначених для ідентифікації вразливостей у веб-додатках на етапах їх розробки та тестування[11][31].

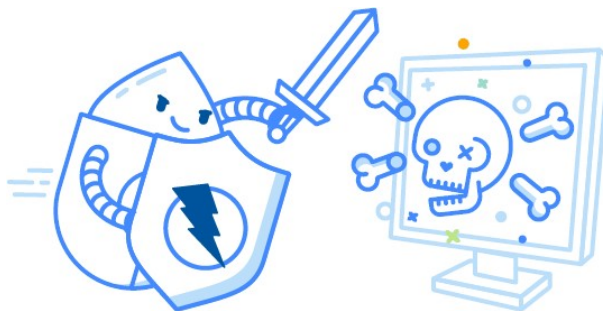


Рисунок 1.11 – OWASP ZAP

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		27

ZAP виконує роль проксі-сервера-перехоплювача, що надає змогу фахівцям із безпеки, розробникам і тестувальникам аналізувати та змінювати трафік між браузерами та веб-додатками. Інструмент функціонує за принципом проксі-сервера "людина посередині", перебуваючи між браузером і цільовим веб-додатком. Така архітектура дозволяє перехоплювати й аналізувати всі HTTP/HTTPS-запити та відповіді, надаючи повний доступ до даних комунікації. ZAP має модульну структуру на базі плагінів, завдяки чому функціонал можна розширювати через додаткові компоненти, які підвищують можливості тестування. Інструмент побудований на Java, що забезпечує його кросплатформну сумісність із системами Windows, Linux, macOS, а також адаптованість для використання в Docker-середовищі[15].

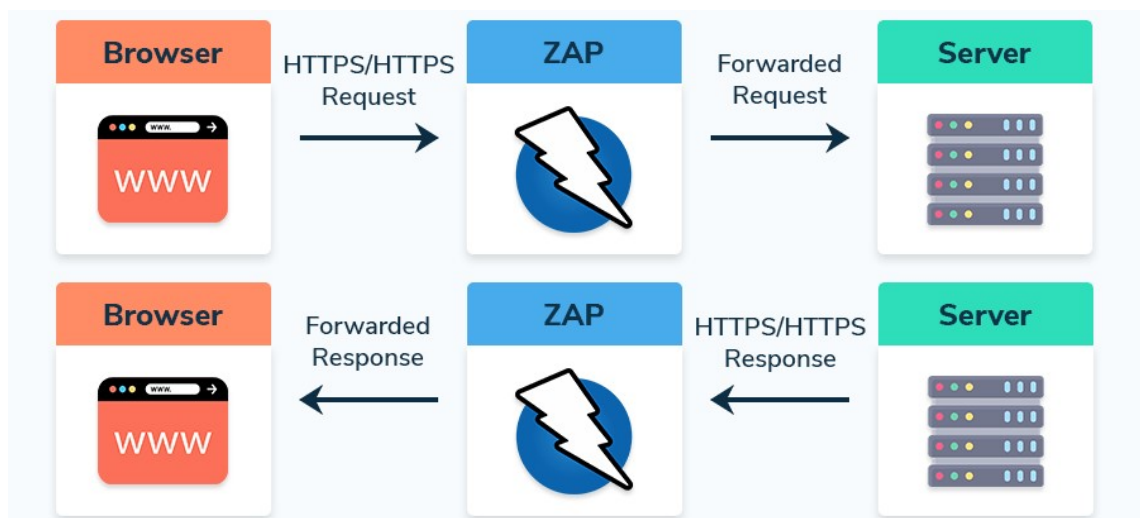


Рисунок 1.12 – Схематична робота ZAP Proxu

ZAP надає *потужні можливості автоматизованого сканування*, здатні виявляти потенційні вразливості без необхідності глибоких знань у сфері безпеки:

- Активний сканер здійснює автоматичний запуск відомих атак на веб-додатки, щоб виявити такі уразливості, як впровадження SQL, міжсайтовий сценарій (XSS) і впровадження команд.
- Пасивний сканер аналізує весь трафік, який проходить через проксі-сервер, без активного втручання в роботу цільового додатка. Він виявляє можливі проблеми, наприклад, відсутність заголовків безпеки, незахищені атрибути файлів cookie або витік інформації у відповідях.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		28

- AJAX Spider орієнтується на сканування додатків із великою кількістю JavaScript, що їх стандартні сканери можуть не помітити. Використовуючи браузерний механізм, він взаємодіє з динамічним вмістом та знаходить приховані області програми.

Для фахівців із безпеки, які потребують детального контролю, ZAP забезпечує *комплексні можливості ручного тестування*:

- Редактор запитів дає змогу створювати та надсилати спеціальні HTTP-запити з повним керуванням заголовками, параметрами та основним вмістом.

- Аналіз веб-сокетів дозволяє перевіряти та маніпулювати зв'язками WebSocket для тестування безпеки додатків у реальному часі.

- Fuzzer полегшує тестування шляхом автоматичної заміни значень у запитах на тестові дані з вбудованих або користувацьких списків слів для виявлення потенційних уразливостей обробки вхідних даних.

OWASP ZAP пропонує гнучкі механізми аутентифікації, що дозволяють тестувати захищені веб-додатки. Основні підходи включають використання форм, JSON-даних, HTTP/NTLM-аутентифікації та кастомних скриптів. Налаштування через форми передбачає перехоплення POST-запиту з параметрами логіну (наприклад, username та password), створення контексту з цими даними та вказівку успішного локатора для перевірки авторизації. Сесійне управління реалізовано через cookie, заголовки або скрипти. Новітній підхід із заголовками дозволяє автоматично підставляти токени у заголовок Authorization. Автоматизація процесу здійснюється через YAML-конфігурації Automation Framework або REST API. Експорт конфігурацій у XML/контекстні файли дозволяє повторне використання профілів[13].

ZAP генерує структуровані документи, в яких вразливості класифікуються за рівнем ризику — від низького до критичного, а також за ступенем достовірності. Звіти можуть бути згенеровані у різних форматах, а їх особливістю є можливість налаштування контенту звіту, де можна вибирати, які саме розділи включати, наприклад, детальний опис знайдених проблем, статистику за ризиками або фільтрацію за певними контекстами чи сайтами. Щодо підтримки плагінів і

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		29

розширень, ZAP має гнучку архітектуру, яка дозволяє додавати нові функції через додатки та скрипти[18]. Це дає змогу розширювати можливості інструменту залежно від потреб тестувальника. Наприклад, існують плагіни для визначення технологій, що використовуються на сайті, такі як CMS або серверні платформи, що допомагає краще зрозуміти цільову систему. Також є розширення для імпорту OpenAPI-специфікацій, що дозволяє ефективно тестувати API, а також інструменти для сканування відкритих портів або використання наборів патернів для фаззингу різних типів вразливостей[14].

Key Features of OWASP ZAP



Рисунок 1.13 Ключові особливості OWASP ZAP

Ще однією важливою перевагою ZAP є виявлення та запобігання поширеним атакам. Інструмент автоматично розпізнає та управляє анти-CSRF токенами, забезпечуючи таким чином високу точність сканування. Користувачі мають змогу створювати власні правила та шаблони вразливостей, що дозволяє налаштовувати процес тестування відповідно до потреб своєї організації. Для детальнішого аналізу ZAP пропонує широкі функціональні можливості[28]. Зокрема, користувачі можуть встановлювати точки зупинки, що сприяє тоншій настройці процесу тестування. Крім того, реалізовано тестування на основі стану, яке дає змогу перевіряти логіку переходів усередині додатка.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				30
Змн.	Арк.	№ докум.	Підпис	Дата		

ZAP знаходить широке застосування у різних сценаріях упродовж життєвого циклу розробки. Як флагманський проєкт OWASP, ZAP має потужну підтримку спільноти та активно розвивається. Широка база документації, форуми та ресурси допомагають користувачам ефективно освоювати інструмент. Також OWASP надає великий обсяг освітніх матеріалів — посібників, підручників та навчальних відео — що робить ZAP доступним для спеціалістів будь-якого рівня.

1.3.3 Переваги та обмеження OWASP ZAP

Переваги

- Спеціалізація на безпеці веб-додатків

OWASP ZAP створений із фокусом саме на безпеку веб-додатків, чим вигідно відрізняється від таких інструментів, як OpenVAS, Nmap, Wazuh або ClamAV, що охоплюють ширший спектр завдань безпеки систем, мереж або кінцевих точок. ZAP ефективно виявляє уразливості на кшталт SQL-ін'єкцій, XSS (міжсайтовий скриптинг) та проблем із небезпечним керуванням сесіями.

- Динамічне тестування (DAST)

Інструмент виконує динамічний аналіз працюючих додатків, активно ідентифікуючи вразливості в реальному часі. Це відрізняє його від OpenVAS чи Nmap, орієнтованих здебільшого на статичні перевірки вразливостей чи мережеве сканування, без аналогічних динамічних функцій. ZAP здатен симулювати різні типи атак та аналізувати відповіді додатка, що дозволяє виявляти логічні помилки та контекстуальні вразливості, які неможливо знайти статичними методами.

- Функціональність проксі-сервера

OWASP ZAP виконує роль проміжного проксі-сервера, що дозволяє перехоплювати та змінювати HTTP/HTTPS-трафік для ручного тестування, і у свою чергу робить його зручним для детальної перевірки.

- Тестування WebSocket та API

ZAP забезпечує аналіз і перевірку безпеки трафіку WebSocket та API (зокрема REST і GraphQL), що значно розширює можливості тестування додатків. Інструмент підтримує автентифікацію через різні методи (OAuth, JWT, API ключі)

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		31

та може автоматично генерувати тести на основі OpenAPI/Swagger специфікацій, що прискорює процес тестування сучасних мікросервісних архітектур.

- Простота використання для веб-тестування

OWASP ZAP пропонує інтуїтивно зрозумілий інтерфейс, який підходить як для новачків, так і для досвідчених професіоналів. На відміну від нього, наприклад, Nmap чи Wazuh можуть вимагати вищого рівня технічних знань для налаштування та аналізу.

- Підтримка спільноти та налаштування

Інструмент має активну спільноту користувачів, регулярні оновлення, широкий вибір плагінів і можливості для кастомізації через скриптинг. Хоча OpenVAS і Nmap також мають свої спільноти, їхні можливості розширення здебільшого стосуються саме мережевого сканування.

Недоліки

- Обмеженість застосування поза межами веб-додатків

OWASP ZAP ефективний для тестування веб-додатків, однак не охоплює інші важливі напрямки, як-от безпека мережі (зокрема, використання Nmap), захист кінцевих точок (Wazuh) чи виявлення шкідливих програм (ClamAV). Для організацій, які прагнуть комплексного аналізу всіх ІТ-активів, більш доцільно обрати інші інструменти.

- Хибні позитивні спрацьовування

Подібно до Nikto, OWASP ZAP може видавати чимало хибних позитивів під час автоматичних сканувань, що вимагає подальшої ручної перевірки. Інструменти на кшталт OpenVAS чи Wazuh зазвичай забезпечують більш точну діагностику системних вразливостей завдяки використанню ретельно курованих баз даних.

- Продуктивність у масштабних середовищах

OWASP ZAP не адаптований для роботи у великих корпоративних середовищах, які потребують можливості розподіленого сканування між кількома системами. Його краще застосовувати для тестування окремих додатків чи невеликих розгортань.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		32

- Відсутність функцій захисту кінцевих точок

На відміну від таких рішень, як Wazuh чи ClamAV, OWASP ZAP не виконує завдань зворотнього захисту кінцевих точок або виявлення шкідливих програм. Його сфера застосування обмежується аналізом вразливостей веб-додатків.

- Нестача мережевого сканування

Інструменти на кшталт Nmap або OpenVAS є значно кращими варіантами для детального аналізу мережевих служб та конфігурацій. OWASP ZAP не пропонує достатньо засобів для глибокої оцінки мережевих вразливостей[34].

В результаті порівняння різних сканерів, було вирішено таки зупинитися на OWASP ZAP. Такий вибір обумовлений його спеціалізацією на всебічному динамічному тестуванні веб-додатків, яке поєднує автоматизовані процеси з ручними підходами. Завдяки архітектурі проксі-сервера цей інструмент забезпечує можливість детального аналізу HTTP/HTTPS-трафіку, перехоплення та модифікації запитів. OWASP ZAP пропонує широкий спектр функціональних можливостей, а завдяки модульній структурі та сумісності з плагінами ZAP легко інтегрується в процеси DevSecOps. Особливо важливою перевагою є відкритий код цього інструмента. Інтуїтивно зрозумілий інтерфейс робить OWASP ZAP доступним навіть для новачків, а можливість налаштування дозволяє використовувати його як для невеликих проєктів, так і для тестування складних систем.

Висновки до розділу:

У першому розділі проведено детальний аналіз теоретичних засад забезпечення безпеки веб-додатків і застосування інструментів для їх сканування. Дослідження концепції сканування безпеки дозволило з'ясувати, що сучасні кіберзагрози у сфері веб-технологій характеризуються зростанням складності та різноманітністю векторів атак. Аналіз веб-додатків як об'єктів дослідження виявив, що сучасні архітектурні підходи створюють нові виклики для захисту. Це пояснюється збільшеною поверхнею атак і складною взаємодією компонентів.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		33

Порівняльний аналіз інструментів для сканування безпеки веб-додатків продемонстрував переваги OWASP ZAP у порівнянні з універсальними сканерами, такими як OpenVAS, Nmap, Wazuh і ClamAV. З'ясовано, що OWASP ZAP забезпечує глибший та більш деталізований аналіз веб-додатків завдяки функціям проксі-сервера, підтримці динамічного тестування. Однак було виявлено певні обмеження інструменту, серед яких можливість хибнопозитивних результатів та необхідність додаткових налаштувань для роботи зі складними автентифікаційними механізмами.

Результати дослідження підтверджують доцільність вибору OWASP ZAP як основного інструменту для реалізації системи автоматизованого сканування безпеки веб-додатків у цьому проекті. Це підкріплює перспективи подальшої розробки практичних рішень на його основі.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		34

РОЗДІЛ 2. АНАЛІЗ І ПІДГОТОВКА СЕРЕДОВИЩА ДЛЯ ПРОВЕДЕННЯ СКАНУВАННЯ БЕЗПЕКИ

2.1 Аналіз веб-додатків як об'єктів тестування

2.1.1 Поняття веб-додатків для пентестингу

Пентестинг (penetration testing), або тестування на проникнення, є однією з основних методологій у сфері кібербезпеки, яка спрямована на моделювання реальних атак на інформаційні системи з метою виявлення слабких місць. У світі кібербезпеки, де постійно з'являються нові загрози, практичний досвід є ключовим. І тут на допомогу приходять веб-додатки з вразливостями — спеціально розроблені програмні рішення, що містять навмисні недоліки безпеки[17]. Це не помилка розробників, а свідоме інженерне рішення, покликане створити безпечне та контрольоване середовище для навчання, відпрацювання навичок та тестування інструментів кібербезпеки[19].

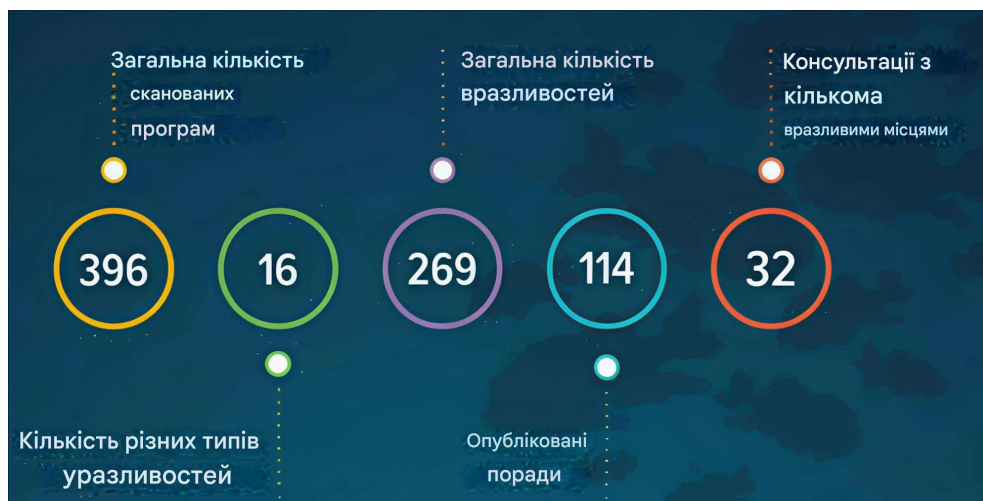


Рисунок 2.1 – Скановані веб-застосунки проти виявлених вразливостей

Подібні додатки, які нерідко називають «вразливими лабораторними стендами», являють собою своєрідні симулятори реальних інформаційних систем, але з однією принциповою особливістю — вони свідомо створені з певними ідентифікованими слабкими місцями. Їх головним завданням є надання фахівцям у сфері кібербезпеки, розробникам програмного забезпечення та студентам унікальної можливості без жодних ризиків практикуватися у дослідженні та використанні різноманітних видів вразливостей. Завдяки цьому можна здобути

цінний практичний досвід без загрози для функціонування реальних сервісів чи порушення законодавства. Такі симулятори дозволяють створювати сценарії атак, що не лише сприяє глибшому розумінню принципів роботи вразливостей, але й допомагає ефективно засвоїти підходи до їх виявлення та подальшої нейтралізації.

Вразливі програми створюють, беручи до уваги поширені помилки програмування та архітектурні недоліки, які зазвичай трапляються у комерційних веб-додатках. Серед таких проблем можуть бути найрізноманітніші: від SQL-ін'єкцій, що дозволяють маніпулювати базами даних, і міжсайтового скриптингу (XSS), який використовується для впровадження шкідливого коду на сторінки, до складніших питань, як-от проблеми з автентифікацією (наприклад, ненадійні механізми входу) або небезпечна десеріалізація. Зазвичай список таких вразливостей відповідає загальновизнаним класифікаціям, наприклад, OWASP Top 10 — актуальному переліку найкритичніших ризиків для безпеки веб-додатків.

Використання веб-додатків з вразливостями дає багато переваг у різних сферах їх використання:

- Вони слугують чудовим інструментом для практичного навчання: студенти та початківці в галузі кібербезпеки отримують можливість освоювати методи виявлення та використання вразливостей у безпечному середовищі, що значно ефективніше, ніж лише теоретичні заняття.
- Такі програми надають професійним пентестерам змогу перевіряти свої інструменти, як-от сканери вразливостей чи проксі-сервери, до їхнього використання на реальних системах, що підвищує якість і надійність тестувань.
- Веб-додатки з вразливостями створюють ідеальне середовище для безпечного вивчення кібербезпеки, дозволяючи дослідникам експериментувати з новими методами атак, розробляти власні експлойти та тестувати сучасні засоби захисту без ризику порушення закону або завдання шкоди реальним системам.
- Водночас для розробників програмного забезпечення такі платформи надають унікальну можливість вдосконалювати захисні механізми через практичний аналіз конкретних вразливостей та способів їх експлуатації, що

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		36

забезпечує отримання цінних знань для створення більш стійкого та безпечного програмного забезпечення.

- Такі програми є ідеальним вибором для демонстрацій під час тренінгів, семінарів чи конференцій, адже вони наглядно демонструють техніку проведення кібератак та ефективні способи протидії їм.
- Нарешті, робота з такими системами допомагає розвивати навички етичного хакінгу, стимулюючи креативне мислення, необхідне для виявлення навіть найменш очевидних вразливостей, які часто залишаються поза увагою автоматичних інструментів.

Вразливі веб-додатки відіграють важливу роль у сфері кібербезпеки. Вони дозволяють безпечно отримувати практичний досвід, випробовувати різноманітні інструменти та вдосконалювати ключові навички, необхідні для захисту цифрового середовища.

На мою думку, дослідження можливостей OWASP ZAP краще проводити саме на вразливих додатках, оскільки це дає змогу перевірити, чи справді сканер виявляє ті проблеми безпеки, які мають бути знайдені. Крім того, на таких платформах можна безпечно експериментувати з різними налаштуваннями та функціями інструменту, не ризикуючи порушити роботу реальних систем або створити правові проблеми. Це особливо важливо для розуміння того, як правильно інтерпретувати результати сканування та відрізнити справжні загрози від помилкових спрацювань.

2.1.2 Навчальна тестова платформа DVWA

Damn Vulnerable Web Application (DVWA) — це навмисно вразлива веб-програма, яку широко рекомендують для практичного тестування безпеки веб-програм. Він розроблений, щоб забезпечити безпечне та законне середовище, де люди можуть навчатися та вдосконалювати свої навички з виявлення та використання вразливостей, які зазвичай зустрічаються у веб-додатках. DVWA — це програма з відкритим вихідним кодом, яку можна встановити на локальній або віртуальній машині, що робить її легко доступною для освітніх цілей[20].

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		37



Username

Password

Login

Рисунок 2.2 – Сторінка авторизації у DVWA

Однією з головних переваг використання DVWA для практичних занять із безпеки веб-додатків є його навчальна цінність. Цей інструмент забезпечує інтерактивний підхід до навчання, дозволяючи користувачам працювати з реальною веб-програмою, яка містить різноманітні вразливості. Завдяки цим вразливостям користувачі отримують можливість наочно зрозуміти ключові проблеми та ризики, які вони створюють. DVWA охоплює широкий діапазон типів уразливостей, що робить його комплексним інструментом для навчання. Деякі з уразливостей, які можна знайти в DVWA, включають впровадження SQL, міжсайтовий сценарій (XSS), впровадження команд, віддалене включення файлів тощо. Кожна вразливість ретельно розроблена для імітації реальних сценаріїв, гарантуючи, що користувачі наражаються на різноманітний набір проблем безпеки, які зазвичай зустрічаються у веб-додатках[12].

Використовуючи DVWA, люди можуть практикувати різні техніки та методології, що використовуються під час тестування безпеки веб-додатків. Наприклад, вони можуть використовувати методи ручного тестування для виявлення вразливостей, такі як перевірка вихідного коду, аналіз мережевого трафіку та маніпулювання полями введення. Крім того, вони можуть використовувати засоби автоматизованого сканування для виявлення потенційних вразливостей і проведення оцінки безпеки.

DVWA також пропонує інтегрований підручник і детальну документацію, яка допомагає користувачам освоїти процес взаємодії з уразливостями. У цих матеріалах детально описуються можливі вразливості, включаючи покрокові інструкції щодо їх експлуатації, аналіз потенційного впливу, а також рекомендації

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		38

стосовно методів їх усунення. Завдяки такій вичерпній документації навчальний процес стає ефективнішим, забезпечуючи користувачів необхідними знаннями й інструментами для розуміння та вирішення вразливостей, з якими вони стикаються.

Damn Vulnerable Web Application (DVWA) є одним із найрекомендованіших інструментів для практичного навчання у сфері безпеки веб-додатків. Це зумовлено його високою дидактичною цінністю, ретельним охопленням ключових вразливостей і практичним підходом до навчання. Застосовуючи DVWA, користувачі мають змогу здобути реальний досвід виявлення та використання типових уразливостей веб-додатків, що сприяє вдосконаленню їхніх навичок у галузі тестування безпеки веб-систем[21].

Для того щоб обрана навчальна платформа Damn Vulnerable Web Application (DVWA) злагоджено працювала, необхідно мати вебсервер (Apache), сервер бази даних (MySQL) та інтерпретатор PHP. Щоб уникнути ускладнень із налаштуванням кожного компонента окремо, доцільно використовувати готовий дистрибутив XAMPP, який об'єднує всі необхідні сервіси. Для цього із офіційного сайту завантажуюмо інсталяційний пакет найновішої версії для відповідної операційної системи. У мене це Windows 10 64-бітної версії, що розташована у віртуальному середовищі VirtualBox, яка у свою чергу налаштована на режим мережі NAT. Зроблено у такий спосіб, оскільки веб-застосунок є дуже вразливим та не рекомендованим для завантаження на власний комп'ютер.

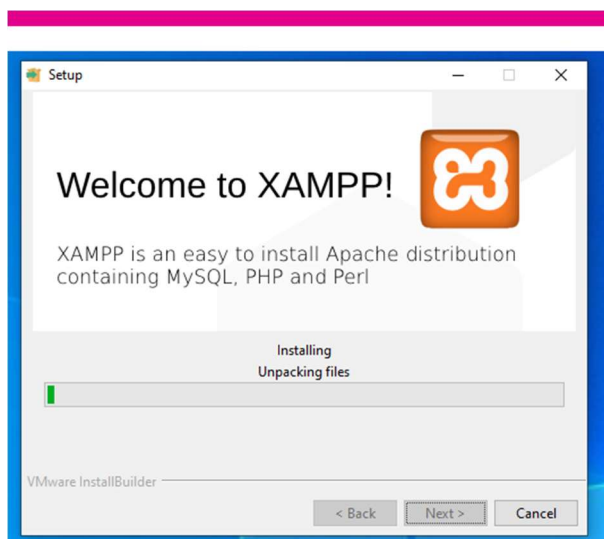


Рисунок 2.3 – Установка XAMPP

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		39

Після завершення встановлення, відкривається XAMPP Control Panel. З її допомогою необхідно активувати необхідні сервіси - вебсервер Apache та сервер бази даних MySQL. Через декілька секунд після запуску кожного із них, з'являється спливаюче вікно брандмауера для Захисника Windows, яке перевіряє чи дійсно ми хочемо запустити дані сервіси. Підтверджуємо для обох та спостерігаємо як вони загоряються зеленим кольором, а отже запуск виконано успішно. Було протестовано роботу Apache шляхом пошуку localhost у браузері. Усе було виконано правильно, а тому у браузері можна спостерігати відповідну сторінку.

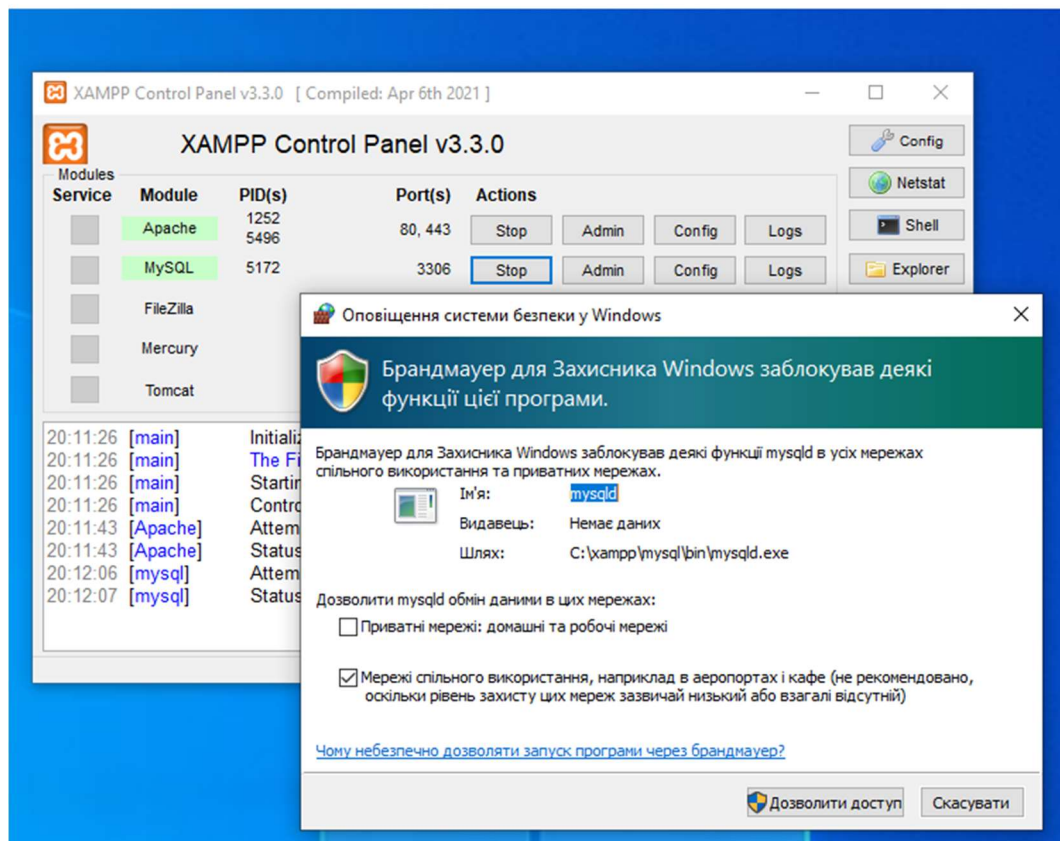


Рисунок 2.4 – Установка XAMPP

Наступним етапом є завантаження DVWA із офіційного репозиторію GitHub. Завантажений архів розпаковано та перенесено папку із файлами до каталогу XAMPP, що був створений під час встановлення даного дистрибутива на попередньому етапі. Перевірити чи правильно виконуються дії можна у браузері. Під час пошуку виникає системна помилка, але це так і повинно бути, оскільки система нам сама підказує, що потрібно змінити файл конфігурації, а саме прибрати розширення .dist.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				40
Змн.	Арк.	№ докум.	Підпис	Дата		

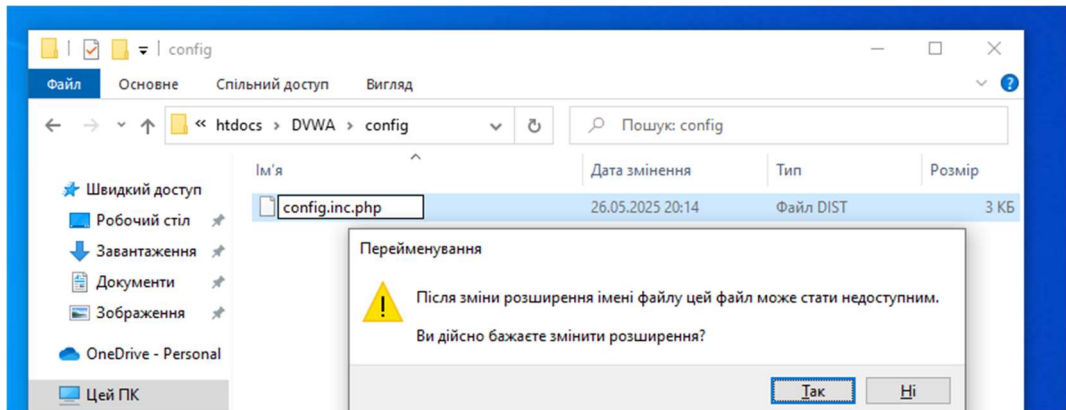


Рисунок 2.5 – Встановлення DVWA

Відкриваємо конфігураційний файл за допомогою будь-якого текстового редактора. Звертаємо увагу на наступні головні дані, що нам потрібні. Вони знадобляться на наступних кроках налаштування вебдодатку.

```
config.inc: Блокнот
Файл Редагування Формат Вигляд Довідка
<?php

# If you are having problems connecting to the MySQL database and all of the variables below are correct
# try changing the 'db_server' variable from localhost to 127.0.0.1. Fixes a problem due to sockets.
# Thanks to @digininja for the fix.

# Database management system to use
$DBMS = getenv('DBMS') ?: 'MySQL';
#$DBMS = 'PGSQL'; // Currently disabled

# Database variables
# WARNING: The database specified under db_database WILL BE ENTIRELY DELETED during setup.
# Please use a database dedicated to DVWA.
#
# If you are using MariaDB then you cannot use root, you must use create a dedicated DVWA user.
# See README.md for more information on this.

$_DVWA = array();
$_DVWA[ 'db_server' ] = getenv('DB_SERVER') ?: '127.0.0.1';
$_DVWA[ 'db_database' ] = getenv('DB_DATABASE') ?: 'dvwa';
$_DVWA[ 'db_user' ] = getenv('DB_USER') ?: 'dvwa';
$_DVWA[ 'db_password' ] = getenv('DB_PASSWORD') ?: 'p@ssw0rd';
$_DVWA[ 'db_port' ] = getenv('DB_PORT') ?: '3306';
```

Рисунок 2.6 – Перегляд конфігураційного файлу

У XAMPP Control Panel перейдемо до налаштувань бази даних MySQL через вкладку Admin. Відкривається вкладка у браузері, яка у свою чергу дає нам повну можливість контролю та управління базами даних. Поки іще нема потрібної нам бази dvwa, отже необхідно її створити. Слідкуємо за тим, щоб назва співпадала із тією, що вказана у конфігураційному файлі DVWA.

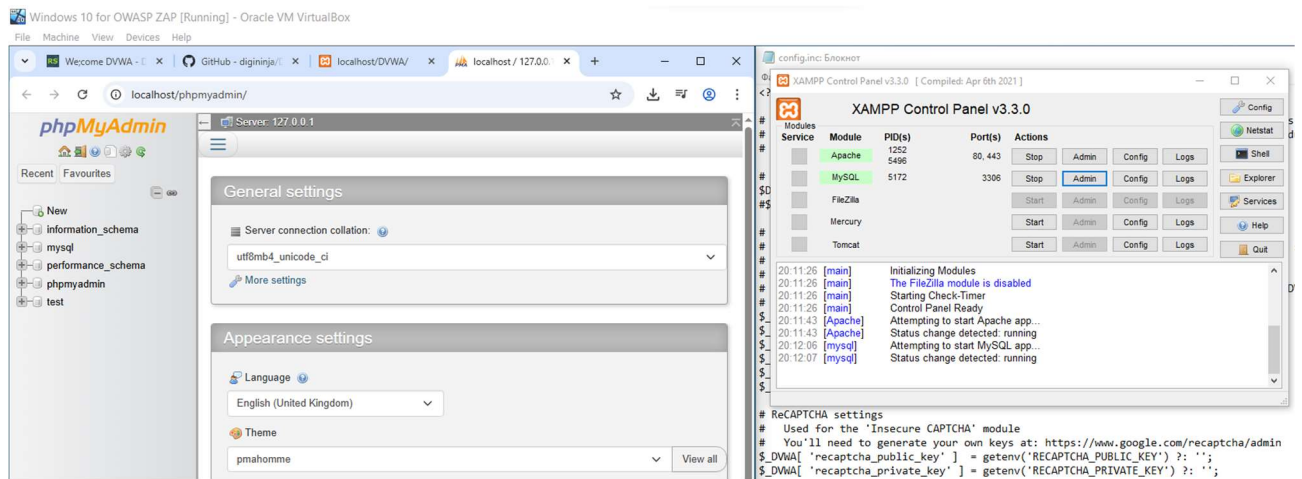


Рисунок 2.7 – Створення бази даних

Наступний крок це створення користувача у новоствореній базі даних dvwa. Логін та пароль обираємо теж із конфігураційного файлу. Надаємо йому повний доступ, щоб надалі могли вільно користуватися даним вебдодатком.

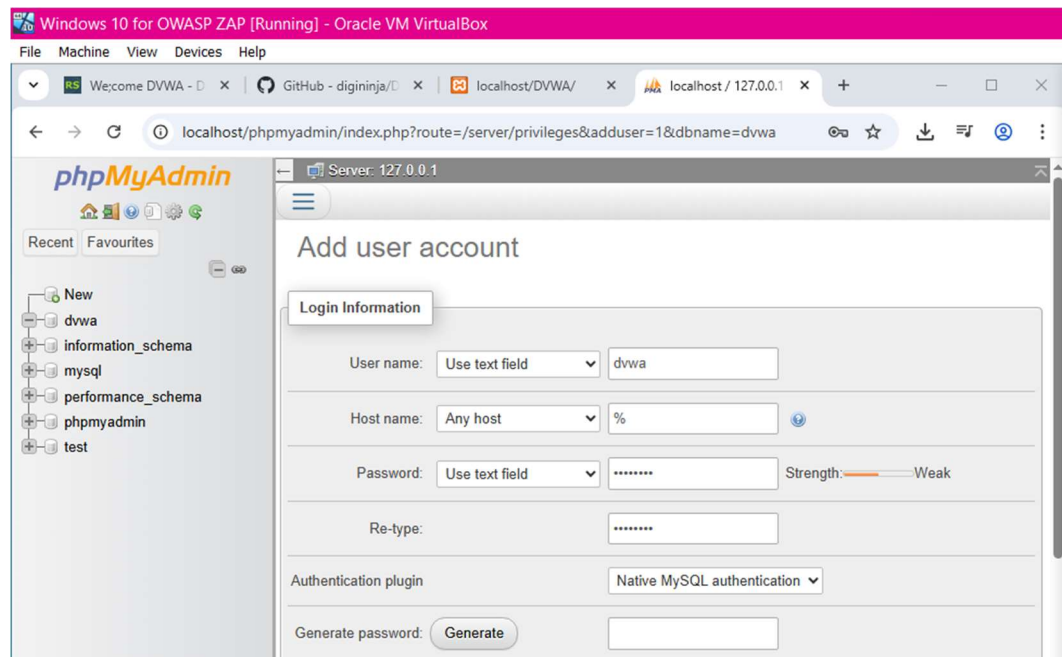


Рисунок 2.8 – Створення користувача

Повертаємося до браузера із DVWA, оновлюємо сторінку та перевіряємо роботу вебдодатку. Навіть без введення логіну та паролю користувача, нам дозволено увійти у додаток, що у свою чергу означає, що налаштування ще не завершено. Пролістуючи у кінець сторінки, натискаємо на кнопку Create/Reset Database. Якщо на попередніх етапах все було виконано правильно, то зараз база даних буде успішно створена і після перезавантаження сторінки ми зможемо залогуватися за відомими нам логіном та паролем користувача admin.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				42
Змн.	Арк.	№ докум.	Підпис	Дата		

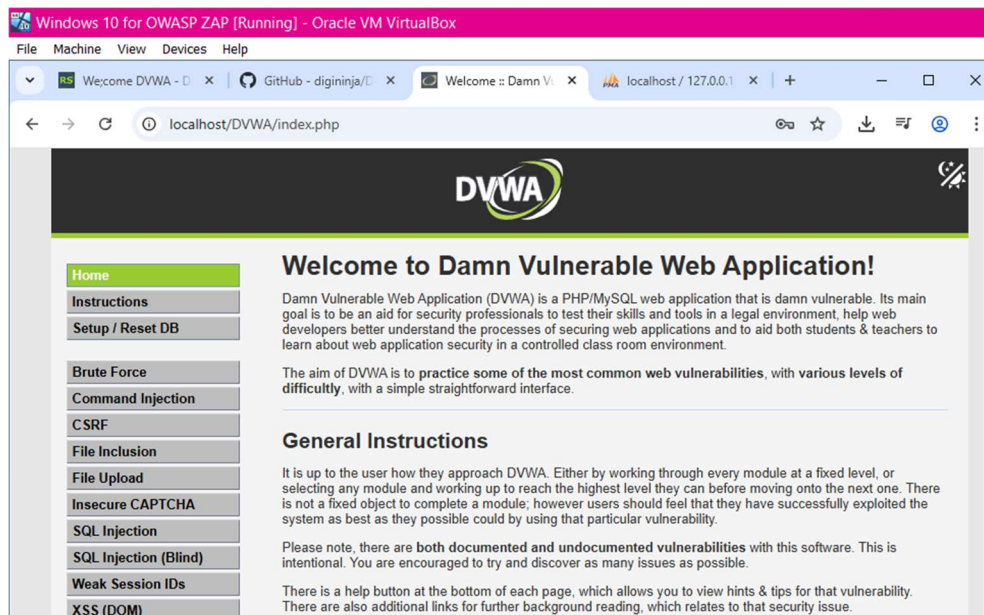


Рисунок 2.9 – Головна сторінка DVWA

Надалі виконаємо ще декілька необов'язкових налаштувань, просто щоб прибрати червоні помилки. У XAMPP Control Panel відкриваємо конфігурацію Apache, а саме `php.ini` файл.

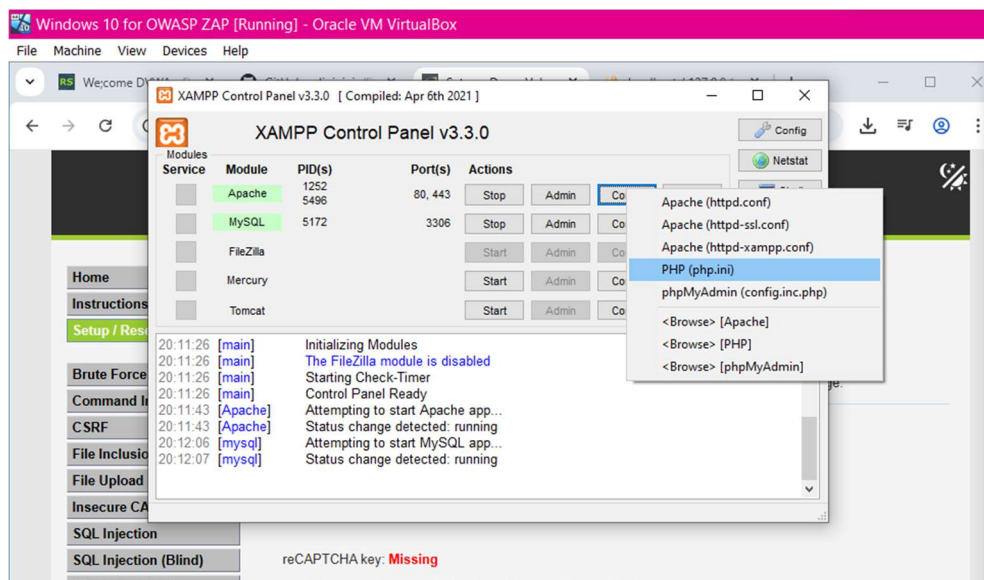


Рисунок 2.10 – Конфігурація Apache

Шукаємо необхідні поля та вмикаємо їх. Це буде дозвіл на директиву `allow_url_include`, яка використовується в деяких сценаріях атак та модуль `gd` для графічної бібліотеки PHP, яка використовується для роботи із зображеннями. Після усіх виконаних змін необхідно перезавантажити Apache сервер, щоб вони вступили у дію. Повертаємося на сторінку DVWA та бачимо, що усі наявні функції та модулі запущені.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				43
Змн.	Арк.	№ докум.	Підпис	Дата		

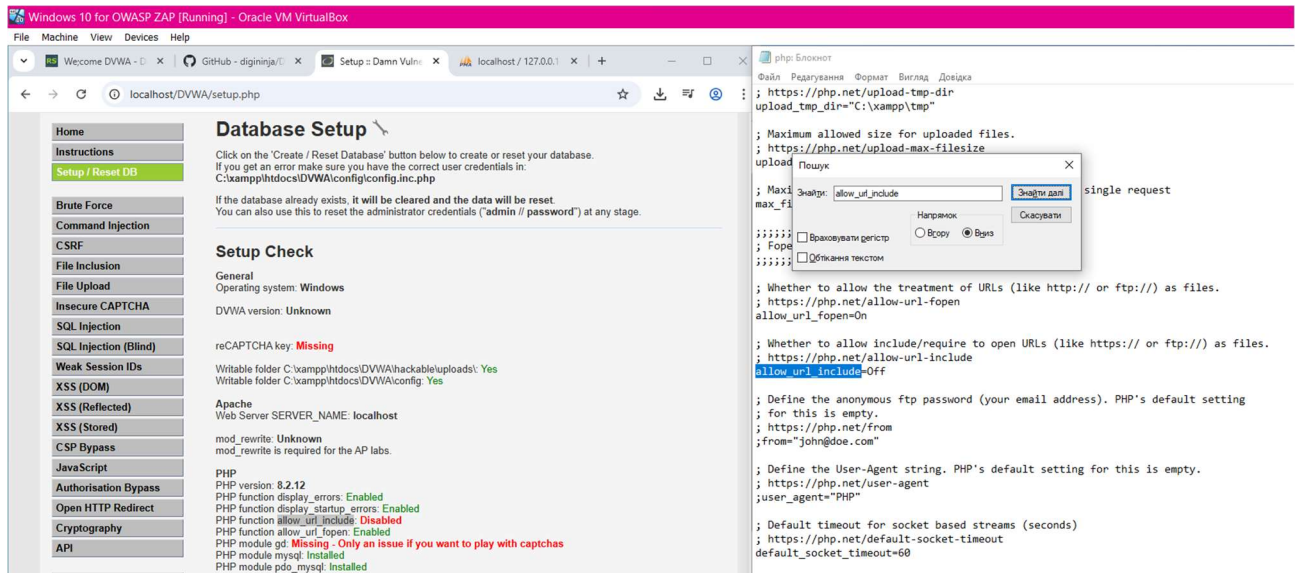


Рисунок 2.11 – Зміна конфігураційного файлу

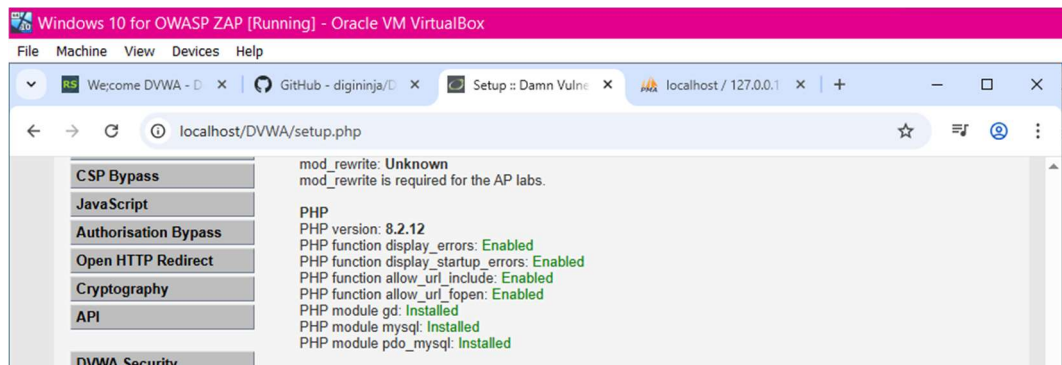


Рисунок 2.12 – Результат зміни конфігураційного файлу

2.2 Технічна підготовка інструментарію для проведення сканування

2.2.1 Встановлення ZAP та огляд інтерфейсу

OWASP ZAP – це безкоштовний і відкритий проксі-сервер з графічним інтерфейсом. Процес встановлення є дуже легким і триває не довго. Спершу переходимо на офіційну вебсторінку проєкту та обираємо інсталятор Windows-версії програми.

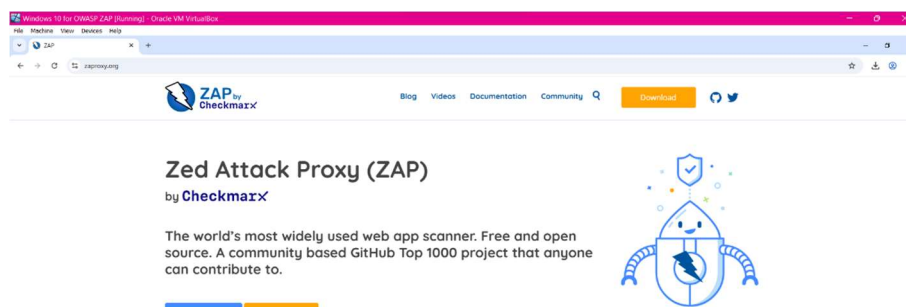


Рисунок 2.13 – Завантаження дистрибутива OWASP ZAP з офіційного сайту

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				44
Змн.	Арк.	№ докум.	Підпис	Дата		

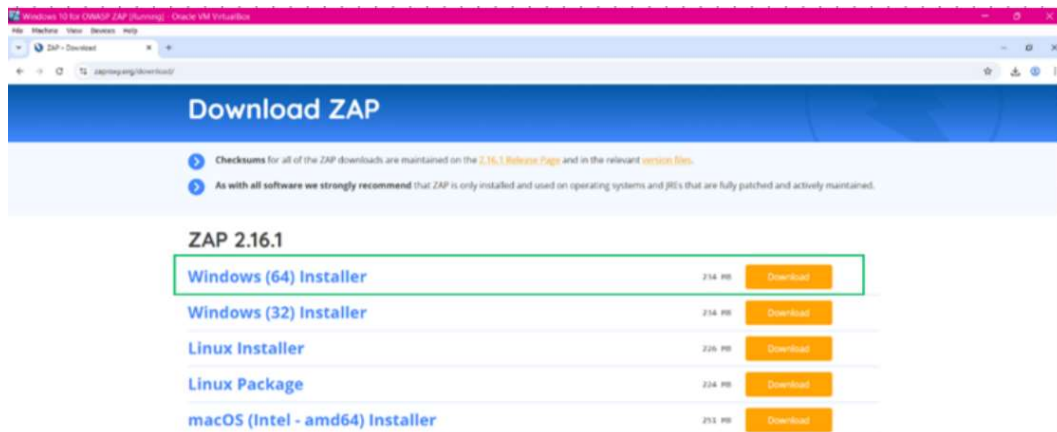


Рисунок 2.14 – Обираємо файл, що відповідає нашій операційній системі та версії

Запускаючи інсталятор від імені адміністратора, уникаємо потенційних обмежень доступу. Під час встановлення необхідно лише обрати мову графічного інтерфейсу додатку, погодити умови ліцензії, обрати тип інсталяції та, за бажання, змінити місце збереження на комп'ютері.

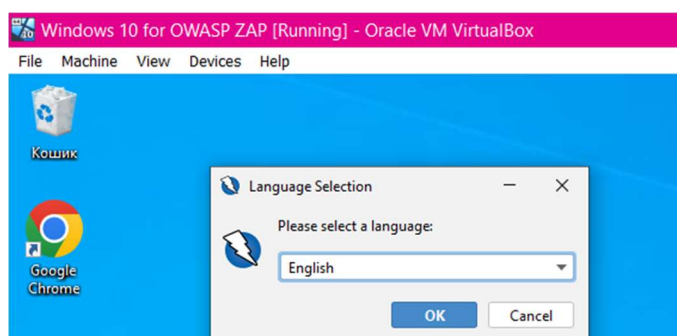


Рисунок 2.15 – Запуск інсталяційного файлу та вибір мови

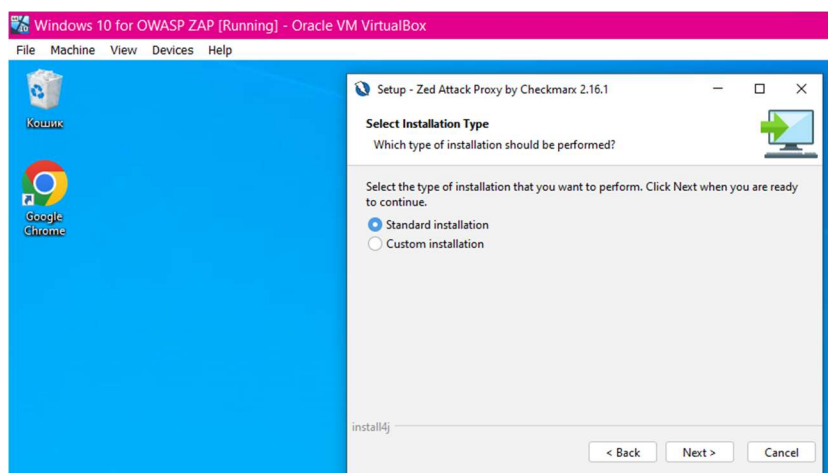


Рисунок 2.16 – Вибір типу інсталяції. У моєму випадку, стандартна

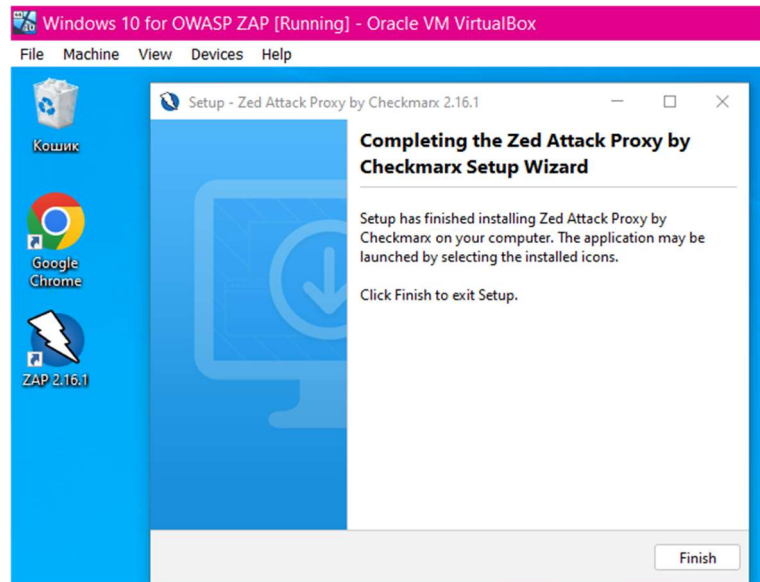


Рисунок 2.17 – Вікно із повідомленням про успішну установку

Для коректної роботи OWASP ZAP необхідною умовою є наявність встановленого середовища виконання Java. Це обумовлено тим, що ZAP повністю реалізований мовою програмування Java, яка є кросплатформною та вимагає спеціального середовища для виконання прикладних програм. Перед інсталяцією самого сканера, за декілька кроків встановлюємо Java Development Kit (JDK) з офіційного сайту Oracle, де було обрано відповідну версію для Windows. Варто зазначити, що OWASP ZAP офіційно підтримує версії Java починаючи з 11-ї.



Рисунок 2.18 – Установка середовища Java

На початку запуску ZAP завжди запитують, чи хочемо ми зберегти даний сеанс. Наразі це не є великою необхідністю, тому можна обрати варіант "Ні, я не хочу продовжувати цей сеанс на даний момент часу" та почати роботу без збереження.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		46

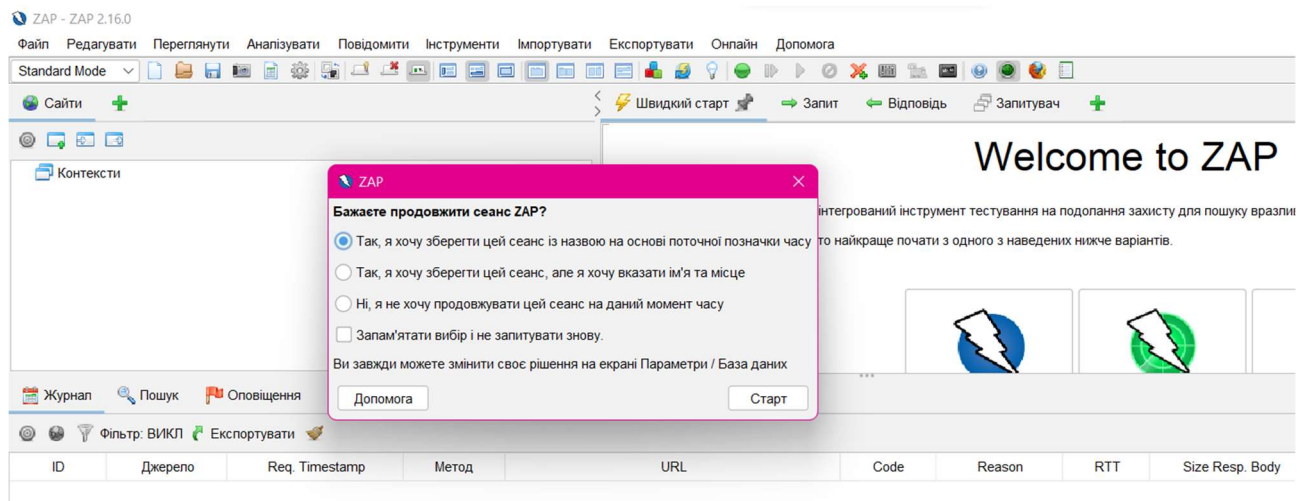


Рисунок 2.19 – Перший запуск OWASP ZAP

Одразу ознайомимося із інтерфейсом програми. При огляді робочого столу OWASP ZAP, можна побачити основні функції та інструменти, доступні користувачеві для тестування безпеки веб-додатків.

1. Рядок меню – забезпечує доступ до широкого спектру як автоматизованих, так і ручних інструментів, що дозволяють ефективно виконувати різні операції та налаштовувати робоче середовище під індивідуальні потреби.

2. Панель інструментів – включає набір кнопок, які спрощують доступ до найбільш популярних та найбільш затребуваних функцій системи, оптимізуючи стандартні робочі процеси.

3. Вікно дерева – виконує роль інтерактивного навігатора, що демонструє упорядковане дерево сайтів і скриптів.

4. Вікно робочої області – служить ключовим компонентом для роботи з запитами, відповідями та скриптами.

5. Інформаційне вікно – відображає всю необхідну деталізовану інформацію щодо стану автоматизованих і ручних інструментів, забезпечуючи користувача важливими технічними деталями.

6. Нижній колонтитул використовується для відображення підсумкового звіту про виявлені сповіщення, а також для представлення актуального стану основних автоматизованих інструментів.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		47

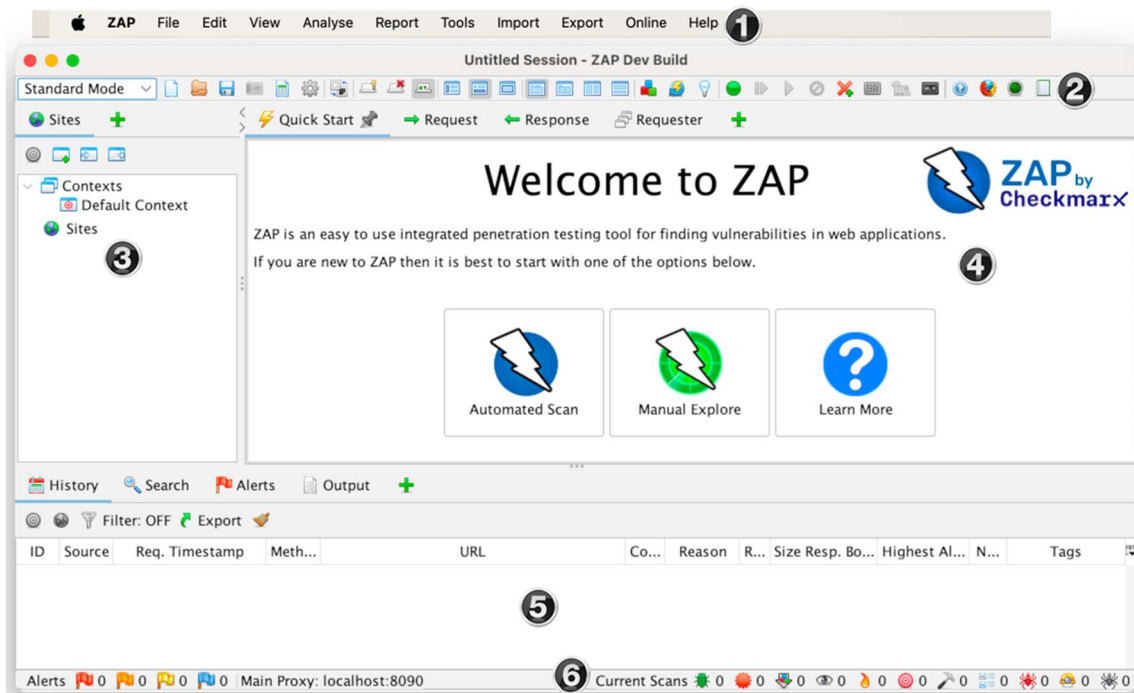


Рисунок 2.20 – Графічний інтерфейс Owasp Zap

2.2.2 Конфігурація параметрів проксі та сертифікатів

Оскільки ZAP функціонує як проксі-сервер, він потребує відповідної конфігурації з боку веббраузера та на рівні системи, щоб уся HTTP- та HTTPS-комунікація могла бути направлена через нього.

Проксі-сервер є сервером-посередником між користувачем та Інтернетом. Він працює як фільтр проти кібератак — запобігає проникненню зловмисників у приватну мережу й захищає ваш комп'ютер від шкідливих програм та інших кіберзагроз. На відміну від VPN, проксі просто маскує вашу IP-адресу і може працювати лише для окремих програм чи браузера, але не шифрує дані[22][23].

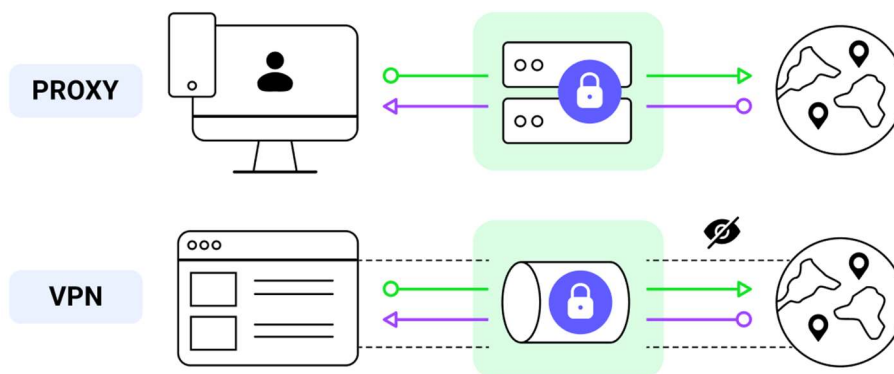


Рисунок 2.21 – Відмінність Проксу від VPN

Але не зважаючи на це, проксі-сервер OWASP ZAP чудово підходить для перехоплення і аналізу веб-запитів під час тестування безпеки. Він зручно демонструє заголовки та вміст HTTP, що дозволяє тестувальникам ефективно оцінювати дані, які передаються між клієнтом і сервером.

Перед тим, як почати використовувати ZAP важливо перевірити налаштування як самої програми, так і конфігурацію нашого комп'ютера. Для цього необхідно перейти в налаштування проксі сервера та змінити налаштування на наведені нижче.

Рисунок 2.22 – Налаштування проксі-сервера на віртуальній машині

Далі обов'язково необхідно перевірити налаштування у програмному забезпеченні. Вони повинні співпадати із тими, що були виконані на проксі сервері робочої станції. Може бути іще можливість налаштування проксі-сервера у браузері. Але залежно від вибору програмного забезпечення, дані налаштування можуть бути відсутні. На даній робочій станції встановлено браузер Chrome і тут при спробі налаштувати проксі-сервер відкривається його налаштування у робочій станції. Тому даний етап є завершеним.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		49

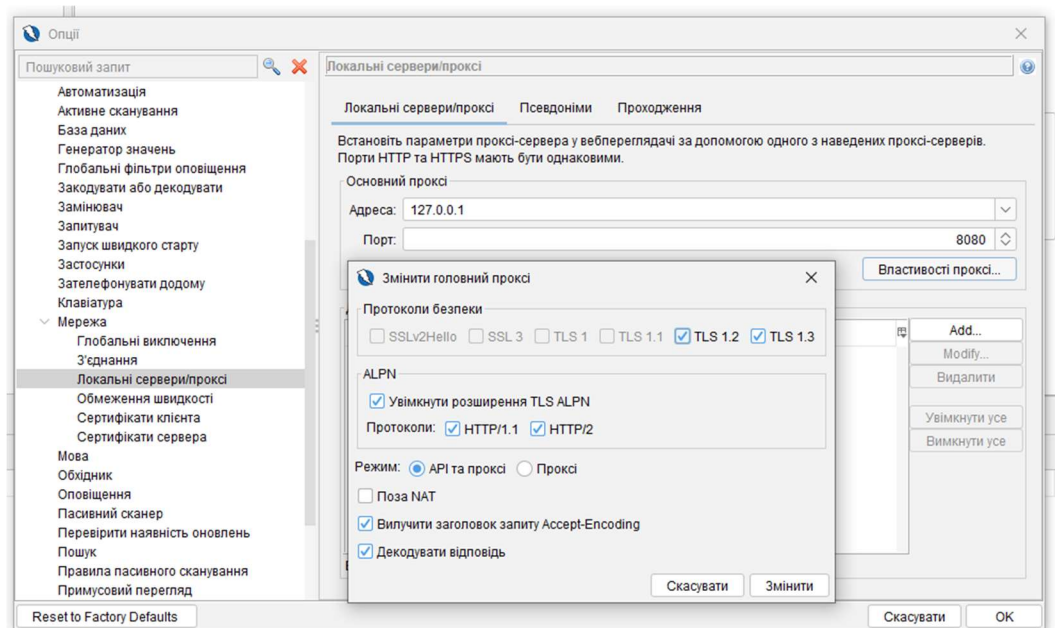


Рисунок 2.23 – Налаштування проксі-сервера у програмі OWASP ZAP

Сучасні вебсайти зазвичай захищають з'єднання через HTTPS, що шифрує дані між клієнтом і сервером за допомогою SSL/TLS. Це запобігає читанню перехоплених даних сторонніми особами. OWASP ZAP, як проксі-сервер для аналізу HTTP(S)-трафіку, потребує доступу до розшифрованих даних для якісного тестування. Для цього ZAP створює власний псевдо-сертифікат, який слугує тимчасовим центром сертифікації[25].



Рисунок 2.24 Вміст SSL-сертифікату

Сертифікат дозволяє ZAP замінювати серверні SSL-сертифікати своїми. Коли користувач переходить на HTTPS-сайт через проксі сервера ZAP, той емулює справжнє з'єднання, підписуючи його власним сертифікатом. Щоб уникнути попереджень у браузері про "небезпечне з'єднання", цей сертифікат потрібно додати як довірений[24].

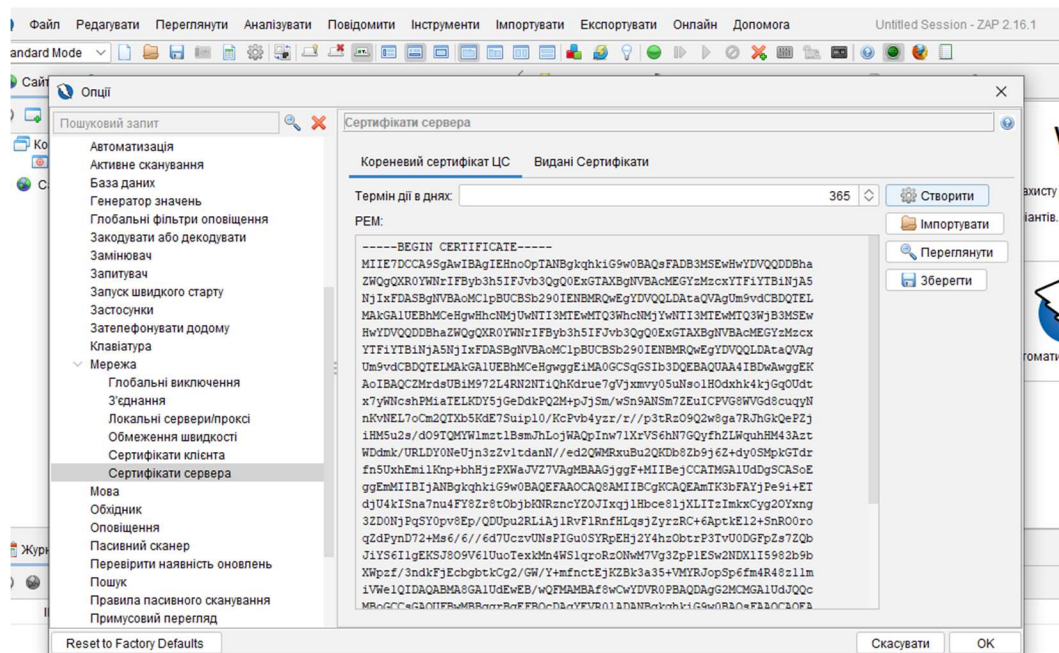


Рисунок 2.25 – Генерація SSL-сертифікату

Сертифікат створюється під час першого запуску програми і зберігається на локальному пристрої. Після того, як у налаштуваннях ZAP було згенеровано новий сертифікат, зберігаємо його у своєму форматі на диску робочої станції. Далі необхідно у обраному браузері знайти менеджер сертифікатів та імпортувати туди сертифікат OWASP ZAP. Нижче показано процес імпорту файлу до розділу «Проміжні центри сертифікації». У цьому розділі зібрані сертифікати проміжних центрів, видані надійними кореновими центрами.

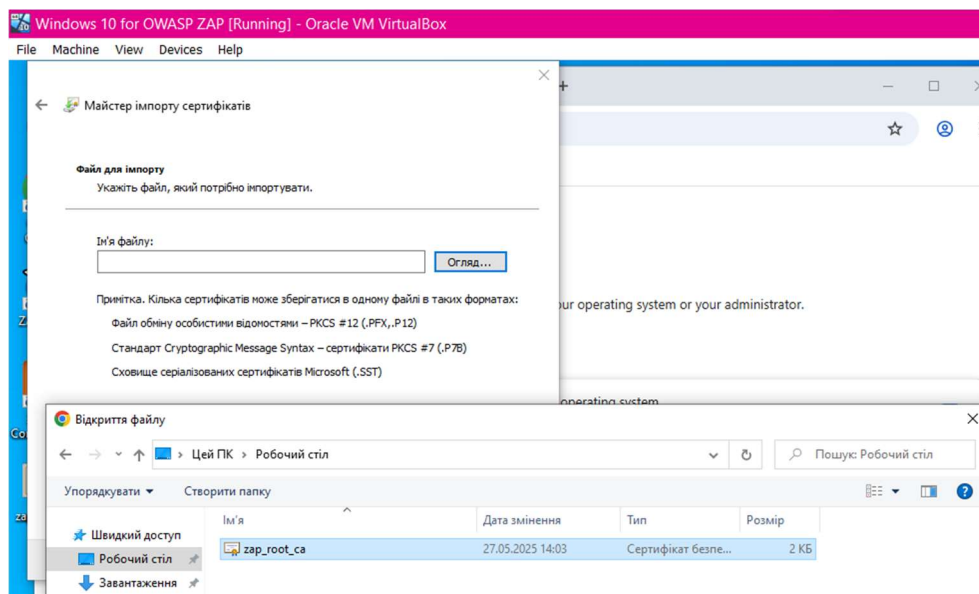


Рисунок 2.26 – Імпорт SSL-сертифікату у Chrome

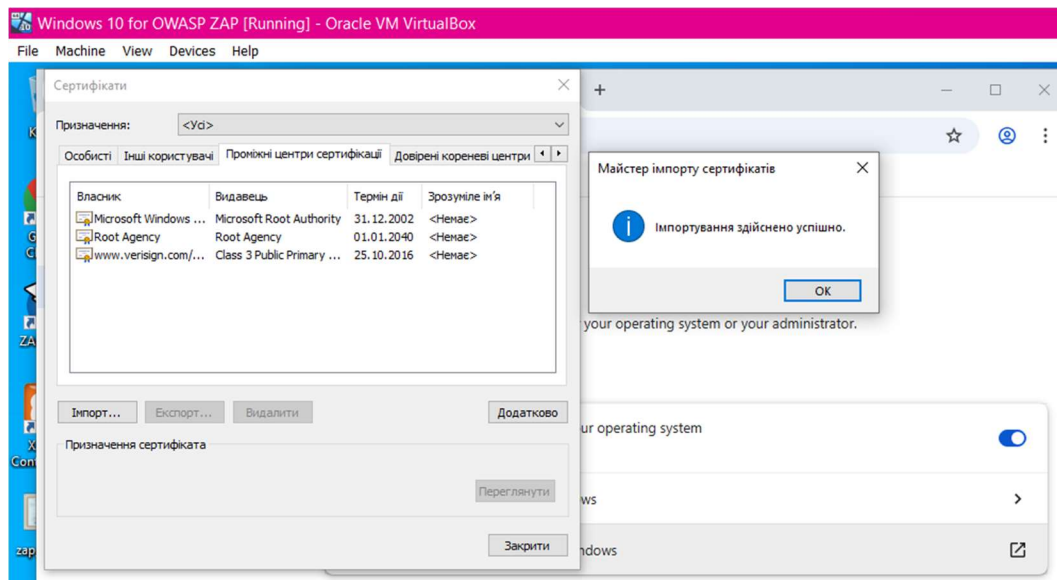


Рисунок 2.27 – Результат імпорту SSL-сертифікату

Також сертифікат було додано до розділів «Довірені кореневі центри сертифікації» та «Довірені видавці». З метою повної довіри операційної системи та браузера. Перший розділ є найважливішим для контролю вебз'єднань HTTPS, а ось другий призначений для довіри до цифрових підписів програмного забезпечення.

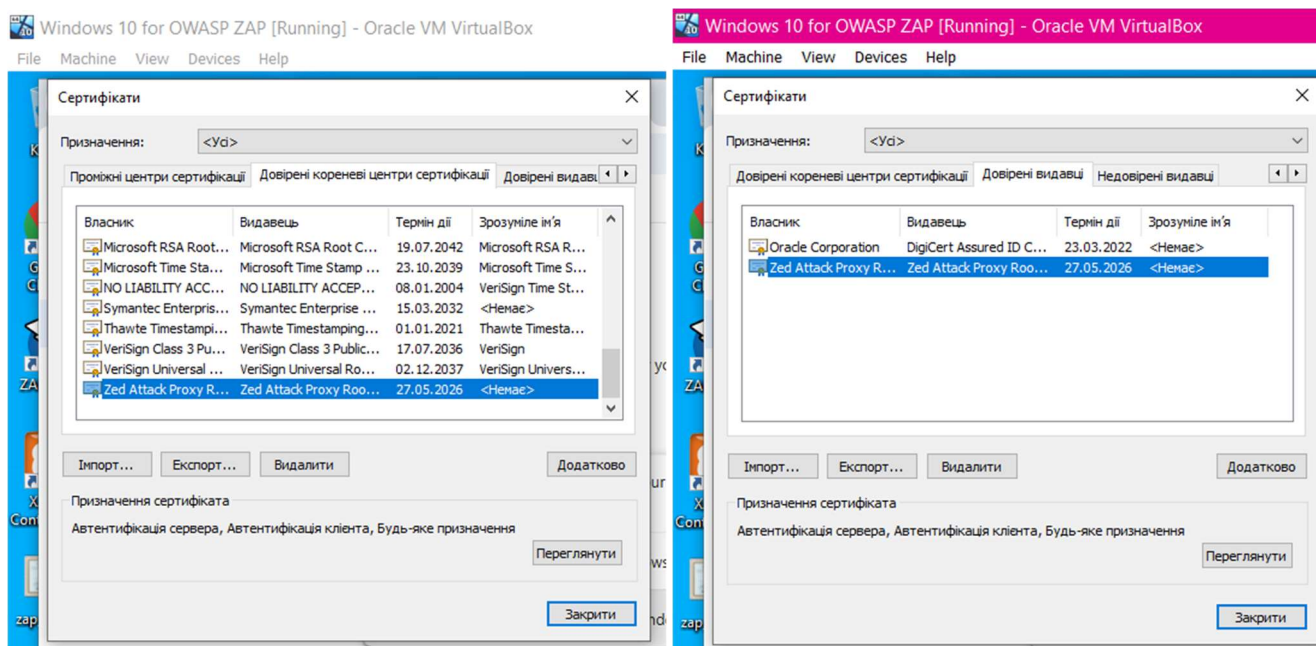


Рисунок 2.28 та Рисунок 2.29 – Результати імпорту SSL-сертифікату

Налаштувавши проксі-сервер та підключивши сертифікати, можна надалі повноцінно використовувати OWASP ZAP. Тепер усі запити проходять через ZAP, і можна зручно тестувати сайти, знаходити вразливості та бачити, як працює передача даних у вебдодатках.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк. 52
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		

Наступним кроком є безпосереднє створення користувача в ZAP. У вікні що з'явилося, вводимо ім'я користувача та відповідний пароль. Після цього користувача потрібно обов'язково "Увімкнути" (Enabled), поставивши відповідну галочку.

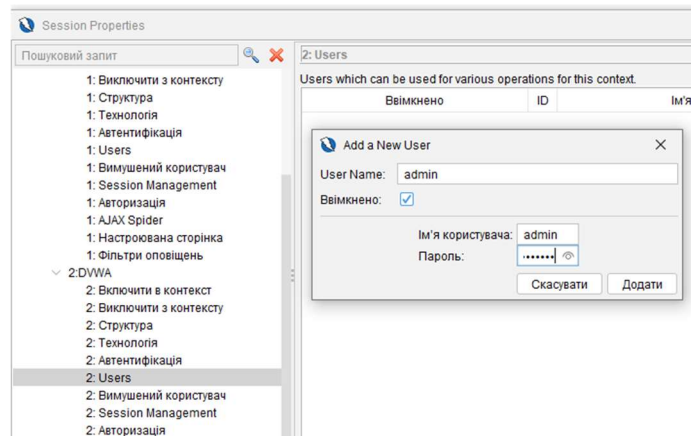


Рисунок 2.32 – Створення користувача

Після успішного створення користувача, необхідно перевірити правильність налаштувань автентифікації. Це можна зробити за допомогою інструмента «Тестувальник автентифікації». У його вікні вказується URL-адреса для входу, обирається створений контекст та ім'я користувача разом із паролем. Обирається бажаний браузер та можна починати тестування.

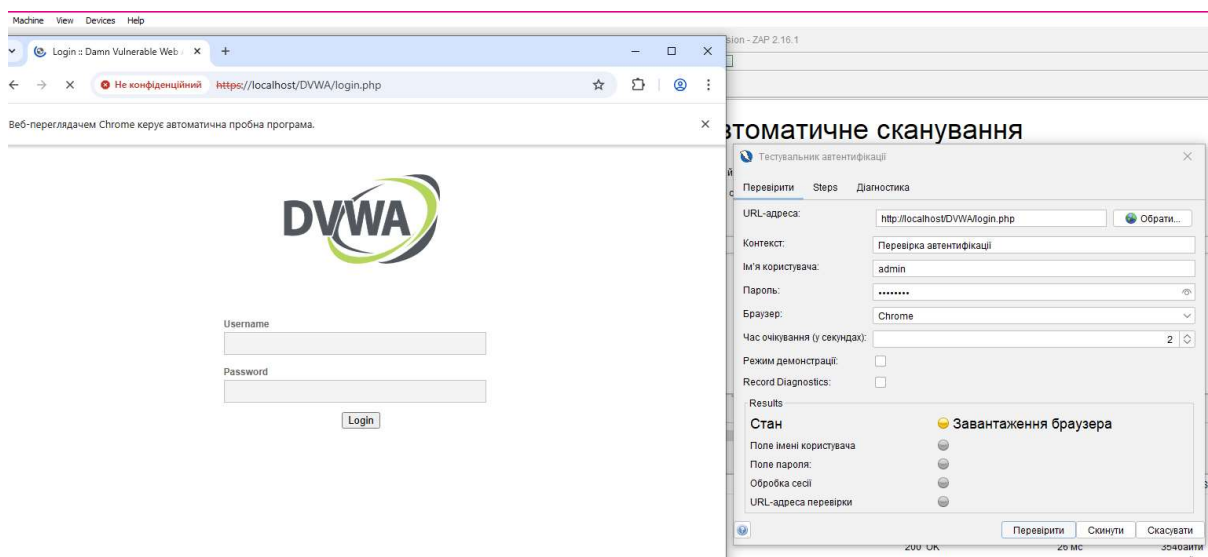


Рисунок 2.33 – Тестувальник аутентифікації

Після запуску перевірки всі налаштування були коректними, тому ZAP показує статус "Виконано" для полів імені користувача, пароля та обробки сесії, що підтверджує успішну імітацію входу в систему.

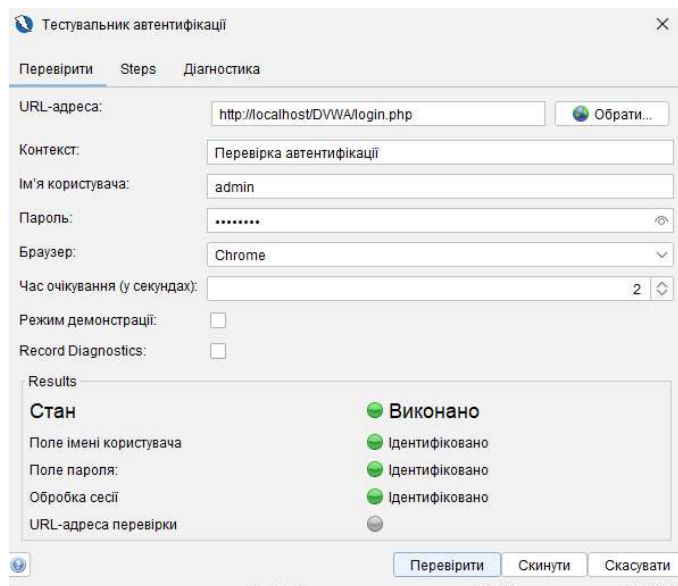


Рисунок 2.34 Результат тестування автентифікації

Висновки до розділу:

У другому розділі було проведено аналіз та повну підготовку середовища. Було досліджено концепцію спеціалізованих вразливих веб-додатків, які виконують роль безпечних і контрольованих «лабораторних стендів» для відпрацювання навичок пентестингу. Як об'єкт тестування, було вибрано навчальну платформу DVWA. Вона охоплює широкий спектр популярних вразливостей, таких як SQL-ін'єкції та XSS, що робить її ідеальним інструментом для освітніх цілей та тестування сканерів.

Була виконана повна установка та налаштування DVWA в безпечному ізолюваному середовищі. Для цього використано дистрибутив XAMPP, що включає веб-сервер Apache та сервер баз даних MySQL, на віртуальній машині з Windows 10. Основним інструментом для сканування обрано і встановлено OWASP ZAP — безкоштовний проксі-сервер з відкритим кодом. Було підготовлено локальне тестове середовище безпосередньо в OWASP ZAP.

Створено контекст для цільового додатка DVWA та налаштовано параметри автентифікації на основі форми. Також створено користувача з відомими обліковими даними, а успішність автоматичного входу перевірено за допомогою вбудованого тестувальника автентифікації.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		55

РОЗДІЛ 3. ПРАКТИЧНА РЕАЛІЗАЦІЯ ПІДСИСТЕМИ СКАНУВАННЯ ТА АНАЛІЗ РЕЗУЛЬТАТІВ

3.1 Проведення ручного сканування на прикладі цільового веб-додатку

OWASP ZAP (Zed Attack Proxy) підтримує різні режими сканування, включаючи ручне сканування, яке дає максимальний контроль над процесом тестування безпеки веб-додатків. Ручне сканування має ряд переваг, серед яких повний контроль над процесом тестування, можливість фокусуватися на специфічних уразливостях, краще розуміння логіки додатку та менше false positive результатів.

Розпочнемо роботу з панелі швидкого запуску OWASP ZAP. У поле «URL для дослідження» введемо адресу нашого локально розгорнутого тестового додатку — `http://localhost/DVWA/`. Для зручності активуємо опцію «Увімкнути HUD» (Heads-Up Display), яка інтегрує основні функції ZAP безпосередньо в інтерфейс браузера. Після цього запустимо браузер Chrome, який тепер працював через проксі-сервер ZAP, що дозволило нам перехоплювати та аналізувати весь трафік.

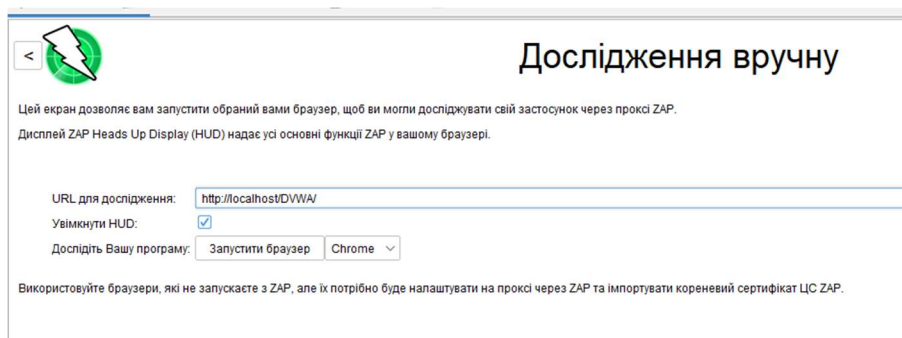


Рисунок 3.1 – Запуск ручного тестування

Запускаємо браузер за допомогою відповідної кнопки. В результаті відкриється вікно, з двох боків якого можна бачити доступні для тестування інструменти. Аутентифікуємося на сторінці входу у додатку DVWA за допомогою стандартних облікових даних. Цей додаток дозволяє обирати рівень складності вразливостей: низький (Low), середній (Medium), високий (High) та неможливий (Impossible). Увійшовши в систему, свідомо налаштовуємо рівень безпеки на "low". Це найнижчий рівень, на якому захисні механізми практично відсутні, що

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		56

дозволяє легше знаходити та демонструвати різні типи вразливостей. Далі наше завдання, вручну переглянути основні розділи сайту, щоб зрозуміти його структуру та функціонал.

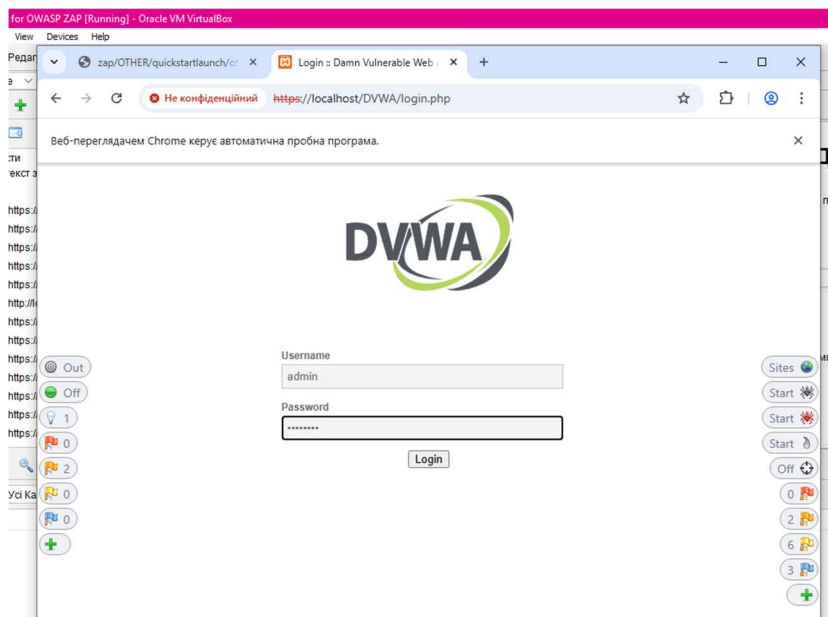


Рисунок 3.2 – Ручне сканування

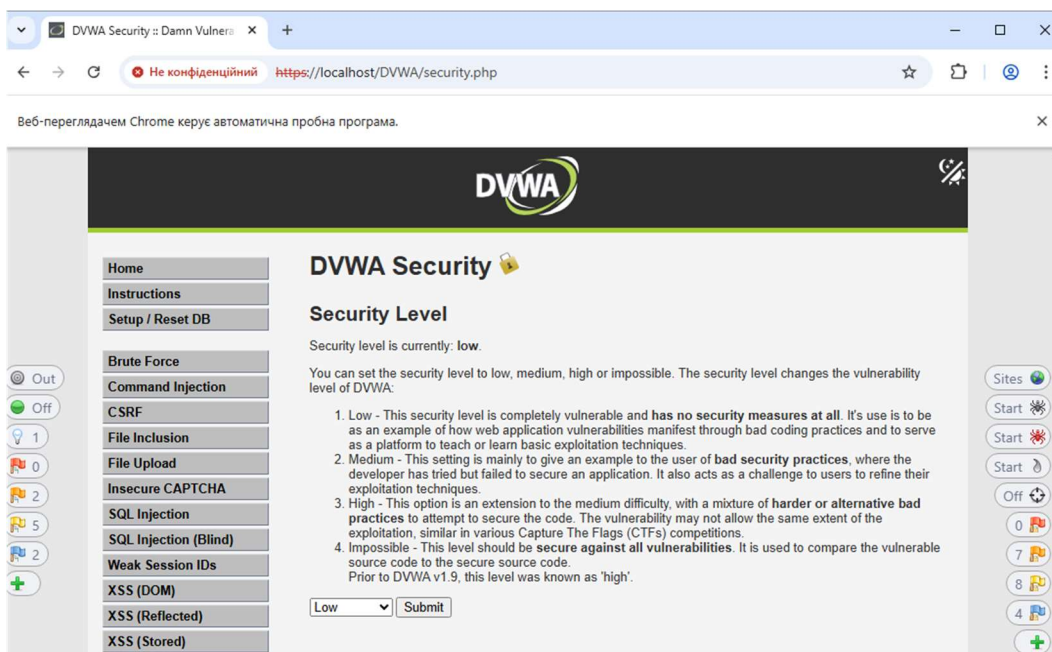


Рисунок 3.3 – Налаштування рівня захищеності системи

Під час ручного відвідування усіх вкладок сайту, виконувалась імітація дій звичайного користувача. В результаті якого уже було знайдено декілька вразливостей, про що свідчать лічильники зліва та справа біля прапорців різного кольору. Перебуваючи на будь якому місці сайту, можна запустити інструменти ручного сканування.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				57
Змн.	Арк.	№ докум.	Підпис	Дата		

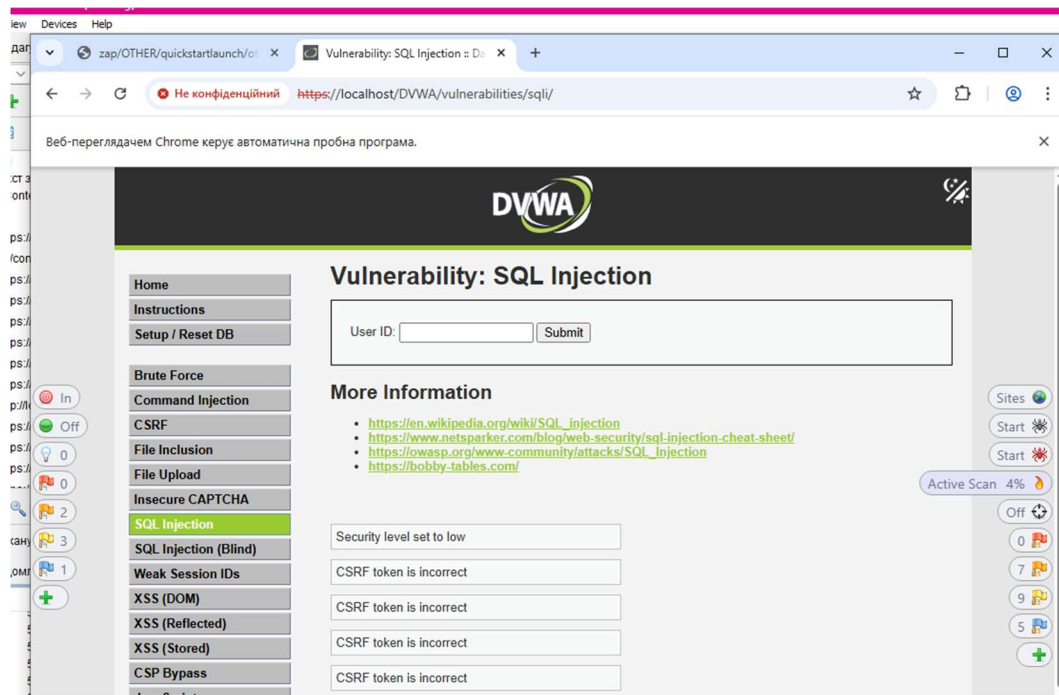


Рисунок 3.4 –Запуск активного сканування

Залежно від структури сайту та його вмісту, процес сканування може займати значний час. Для перегляду атак, що застосовуються під час активного сканування, а також для моніторингу їх прогресу, слід скористатися опцією «Показати деталі сканування процесу». Назва вразливості відображається у стовпці «Аналізатор», у стовпці «Минуло» зазначається час, витрачений на процес сканування цієї вразливості. Будь-яку перевірку можна припинити, натиснувши синю кнопку «Пропустити активне сканування».

Допомога

Untitled Session - ZAP 2.16.1

http://localhost Перебіг сканування

Прогрес

Діаграма відповідей

Хост: http://localhost

	Сила	Прогрес:	Минуло	Вимоги	Оповідження	Стан
Аналізатор			00:00.242	39		
Plugin						
Віддалене включення файлів	Середній		01:34.264	200	0	✓
Зовнішнє перенаправлення	Середній		01:23.824	180	0	✓
Серверна сторона включас	Середній		00:39.545	80	0	✓
SQL впровадження	Середній		03:36.938	550	0	✓
Впровадження коду на стороні сервера	Середній		01:16.224	160	0	✓
Віддалене впровадження команд OS	Середній		03:56.257	681	0	✗
Впровадження XPath	Середній		00:25.929	80	0	✓
Атака на зовнішні об'єкти XML	Середній		00:00.145	0	0	✓
Метадані хмари потенційно розкриті	Середній		00:00.628	18	0	✓
Впровадження шаблону на стороні сервера	Середній		00:55.384	216	0	✗
Впровадження шаблону на стороні сервера (spine ...	Середній		00:29.994	180	0	✗
Перегляд каталогів	Середній		00:04.131	97	13	✓
Переповнення буферу	Середній		00:07.350	20	0	✓
Помилка форматування рядка	Середній		00:26.037	60	0	✓
CRLF введення	Середній		01:09.951	140	0	✓
Підбірка параметру	Середній		01:02.060	138	1	✓
Витік інформації Spring Actuator	Середній		00:00.029	2	0	✓
Ін'єкція XSLT	Середній		00:36.795	79	0	⚡
Правила активного сканування скриптів	Середній			0	0	⚡
Підбірка дій SOAP	Середній			0	0	⚡

Рисунок 3.5 – Перегляд прогресу сканування

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				58
Змн.	Арк.	№ докум.	Підпис	Дата		

Після завершення активного сканування було вирішено одразу провести атаку із використанням вбудованого інструменту "Павук". Його теж можна запустити одразу із браузера із режимом HUD. Цей інструмент автоматично переходить за всіма посиланнями на сайті, щоб скласти повну карту його структури. Він використовується як одна з перших фаз тестування вразливостей. Спробувавши запустити його для localhost, система попередила, що цей сайт не входить до області сканування, і запропонувала додати його.

Після запуску «Павука» варто повернутися на головний екран програми, щоб спостерігати за процесом проведення сканування. У нижній частині інтерфейсу ZAP можна побачити список виявлених URL-адрес. Цей процес допоміг нам зібрати максимальну кількість посилань для подальшого аналізу.

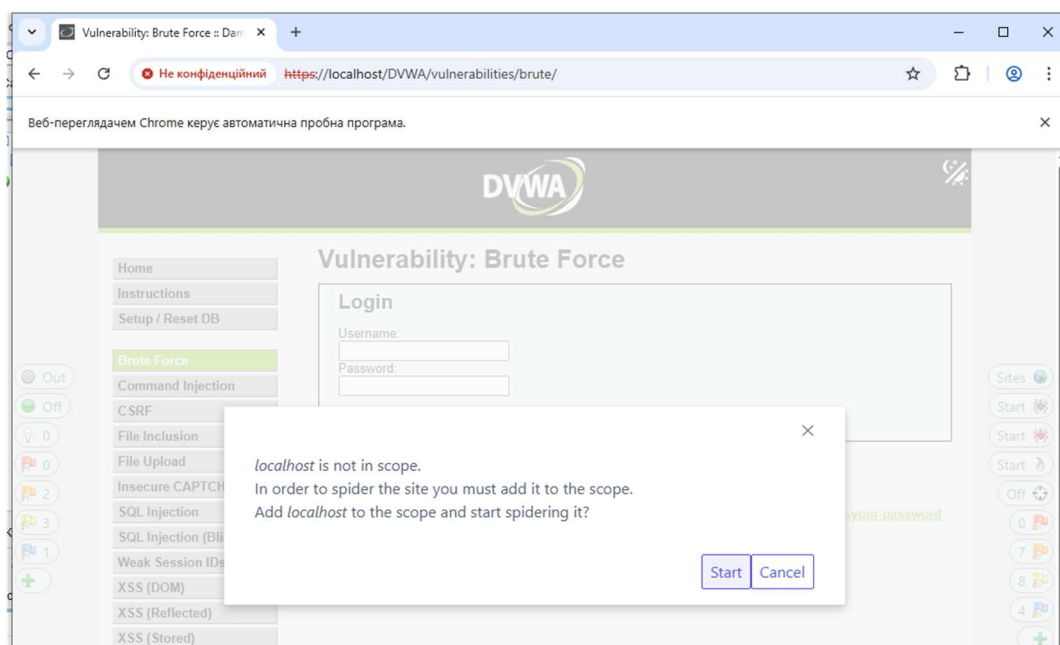


Рисунок 3.6 – Запуск «Павука»

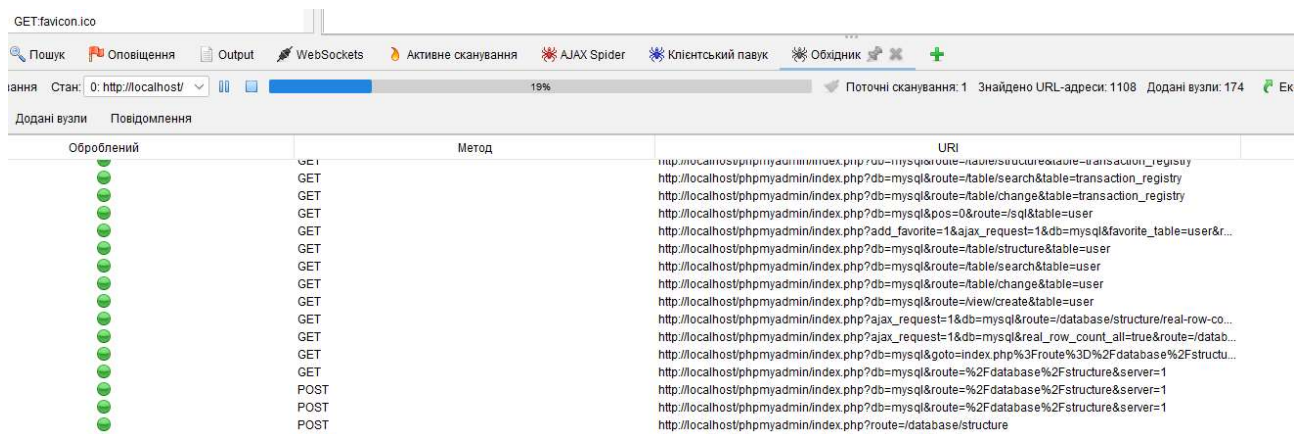


Рисунок 3.7 – Перегляд прогресу

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				59
Змн.	Арк.	№ докум.	Підпис	Дата		

За результатами ручного дослідження та роботи «павука» ми отримали детальну структуру сайту, яка відобразилася у вкладці «Сайти» в ZAP, а також відображаються нові вразливості у вкладці «Оповіщення». Це дало нам ширше уявлення про всі доступні кінцеві точки, скрипти та параметри. За рахунок даного механізму, було заощаджено трохи часу, уникнено пропуску важливих елементів та і загалом це послугувало ефективною підготовкою для подальшого тестування.

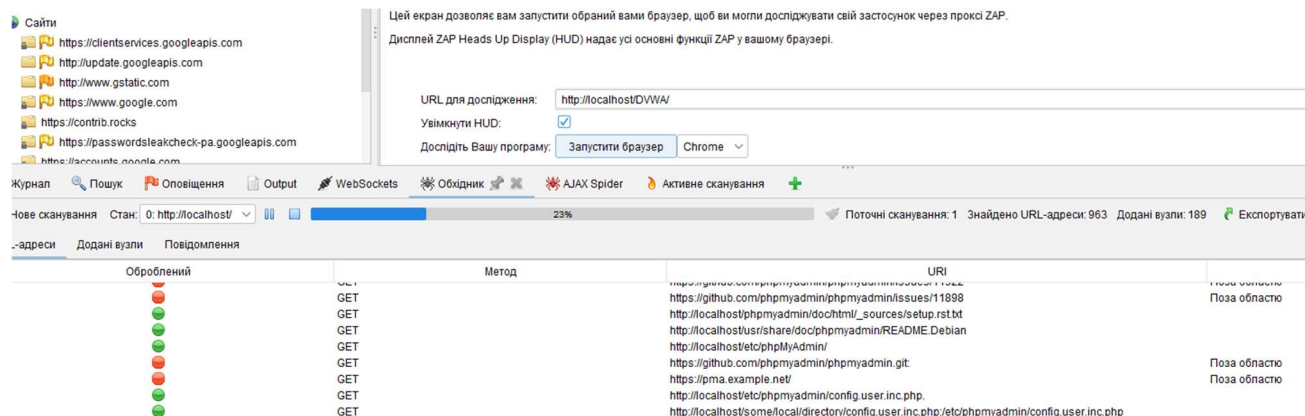


Рисунок 3.8 – Пошук сторонніх URL усередині сайту

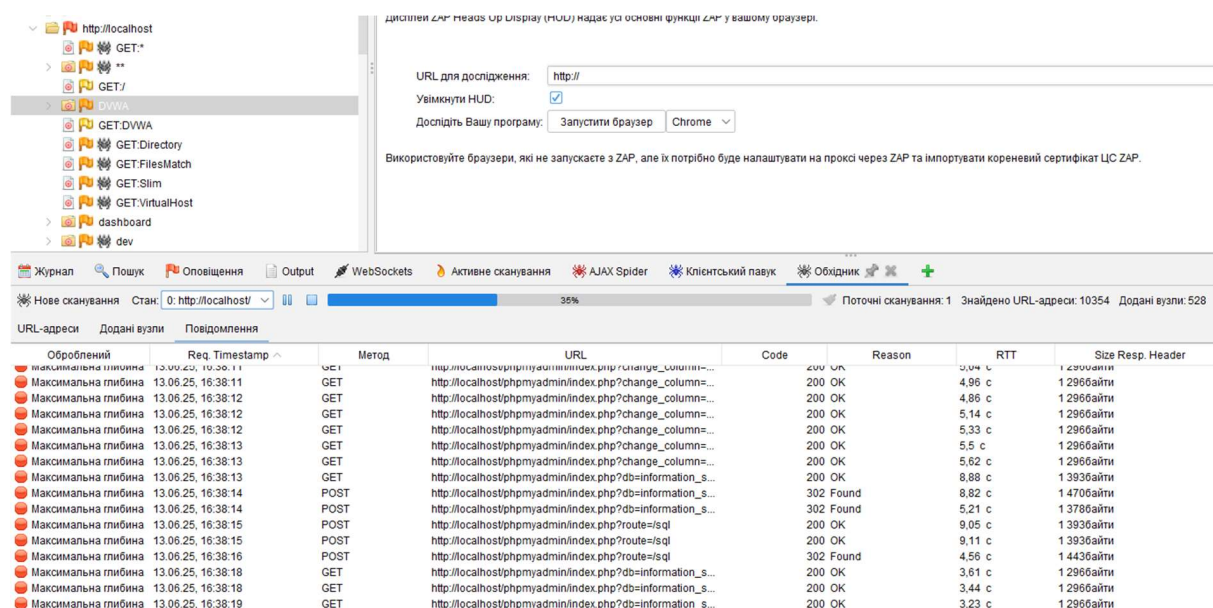


Рисунок 3.9 – Досягнення максимального поглиблення у сайт

Використання цих двох основних інструментів стало хорошим початком для ознайомлення із функціоналом OWASP ZAP та базового дослідження структури вразливого додатку DVWA. Це заклало основу для подальшого глибшого тестування на вразливості, такого як SQL-ін'єкції, XSS або командне виконання, та дало змогу краще зрозуміти типові механізми атак і захисту в веб-додатках.

3.2. Виявлення вразливостей, їх аналіз та генерація звітів

3.2.1 Проведення сканування з використанням користувача

Важливою особливістю нашої підсистеми є можливість сканування від імені автентифікованого користувача. Це дозволяє перевіряти ті частини додатку, які доступні лише після входу в систему. З цією метою було налаштовано активне сканування, вказавши початкову точку, контекст (DVWA) та з використанням користувача admin, якого було попередньо налаштували. Це дозволило сканеру отримати доступ до всіх розділів, доступних адміністратору.

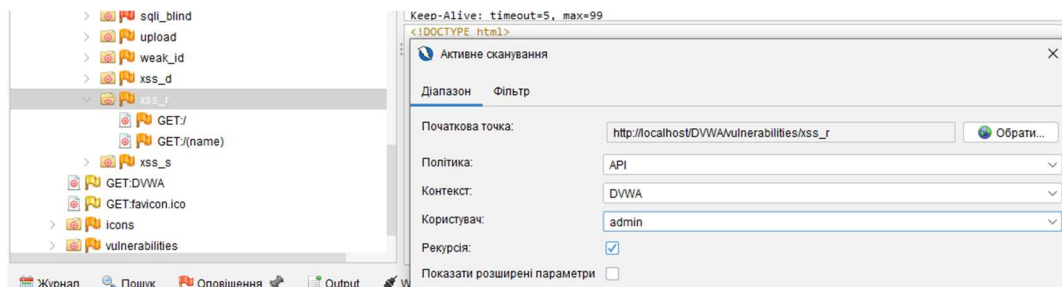


Рисунок 3.10 – Застосування контексту та користувача для сканування

Під час сканування було виявлено низку вразливостей. Наприклад, було знайдено попередження середнього рівня ризику "Відсутній заголовок для захисту від клікджекінгу" (Missing Anti-Clickjacking Header). Це техніка, при якій зловмисник накладає прозорий або непрозорий елемент (наприклад, кнопку чи посилання) на вебсторінку. Користувач, намагаючись клікнути на елемент на підробленій сторінці, насправді виконує дію на прихованій легітимній сторінці.

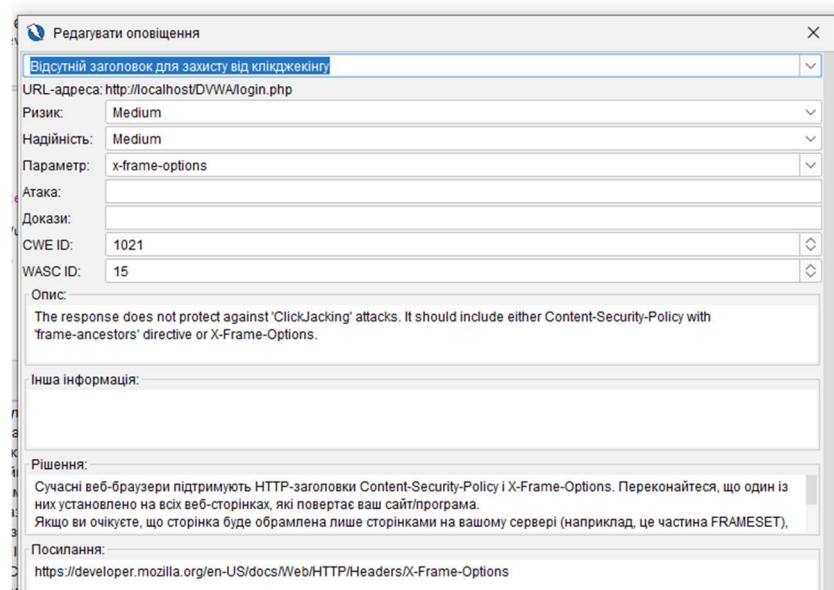


Рисунок 3.11 – Інформація про знайдену вразливість

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				61
Змн.	Арк.	№ докум.	Підпис	Дата		

Для перевірки однієї з найпоширеніших вразливостей, Reflected Cross-Site Scripting (XSS), було проведено ручний тест. Перейшовши на відповідну сторінку, на якій було знайдено попередню вразливість, необхідно ввести спеціальний скрипт у поле вводу. Наприклад, простий JavaScript-код: `<script>alert('Test')</script>`, який не несе ніякого шкідливого впровадження.

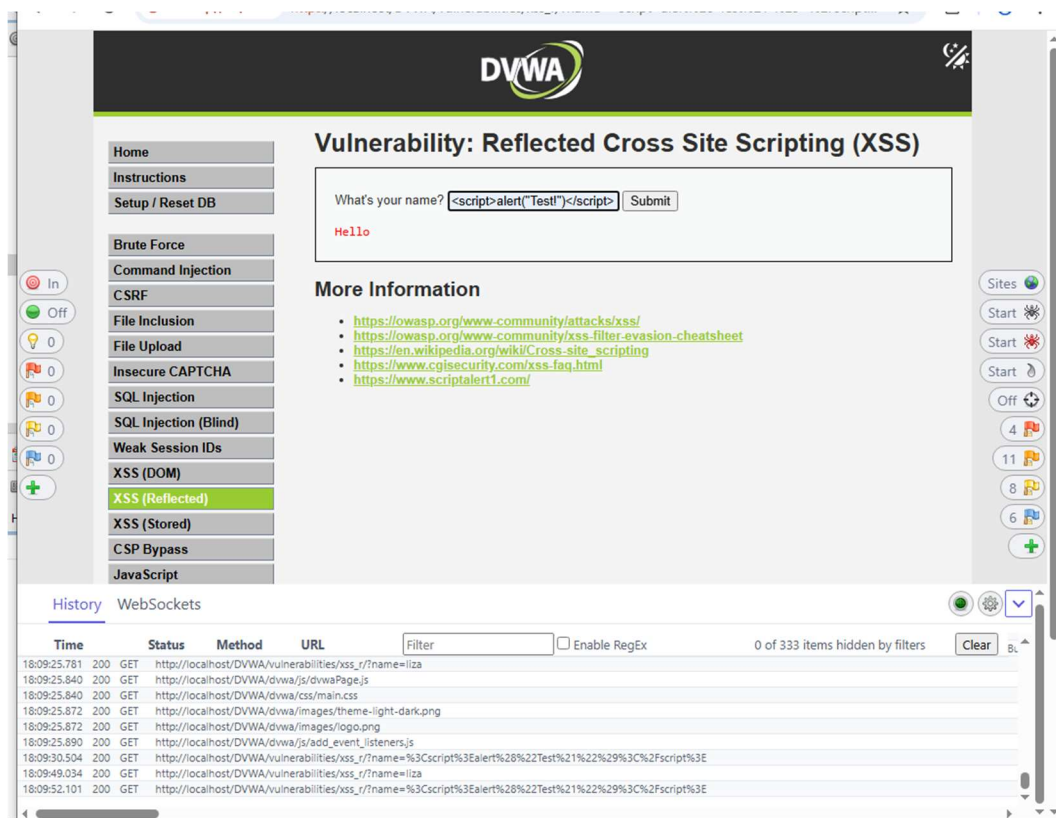


Рисунок 3.12 – Перегляд нових записів під час заповнення форми

Окрім того, що запити до форми відображаються у списку історії подій у OWASP ZAP, введений потенційно шкідливий скрипт виконався. Це підтверджується появою спливаючого вікна із повідомленням "Test!". Таким чином, було наочно доведено наявність XSS-вразливості, яку автоматичний сканер також успішно виявив[27].

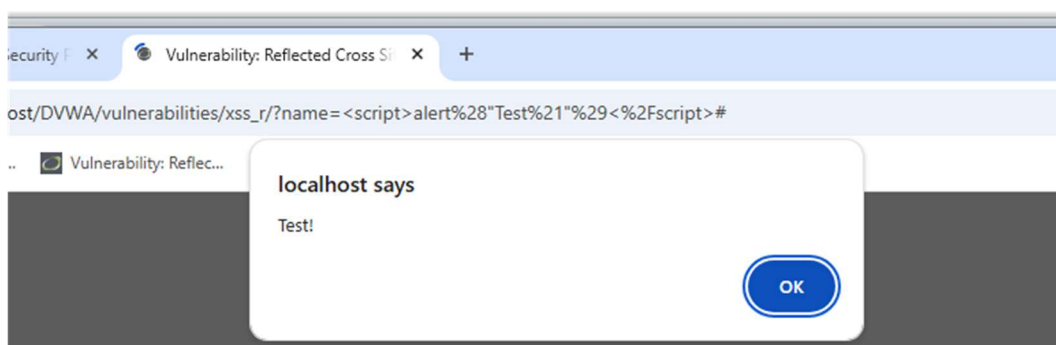


Рисунок 3.13 – Підтвердження наявності вразливості XSS

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				62
Змн.	Арк.	№ докум.	Підпис	Дата		

3.2.2. Створення звіту із результатів сканування

Оскільки виконання сканувань для пошуку вразливостей більше не заплановано, доцільно підготувати детальний та наочний звіт, який дозволить зручно ознайомитися із результатами виконаної роботи та проаналізувати усі виявлені вразливі місця. У підсумковому звіті всі знайдені сповіщення згруповано за рівнем ризику. Під час виконаного сканування було виявлено такі результати: 4 вразливості високого рівня, 14 середнього, 12 низького рівня, а також 11 інформаційних повідомлень.

Рівень ризику	Кількість сповіщень
Високий	4
Середній	14
Низький	12
Відомості	11

Рисунок 3.14 – Кількість сповіщень

Звіт також містить візуалізації, які допомагають краще зрозуміти інформацію. Наприклад, кругова діаграма демонструє розподіл вразливостей залежно від рівня їхньої небезпеки, а стовпчаста діаграма під назвою «Count of Vulnerability Occurrences» ілюструє, які типи помилок трапляються найчастіше.

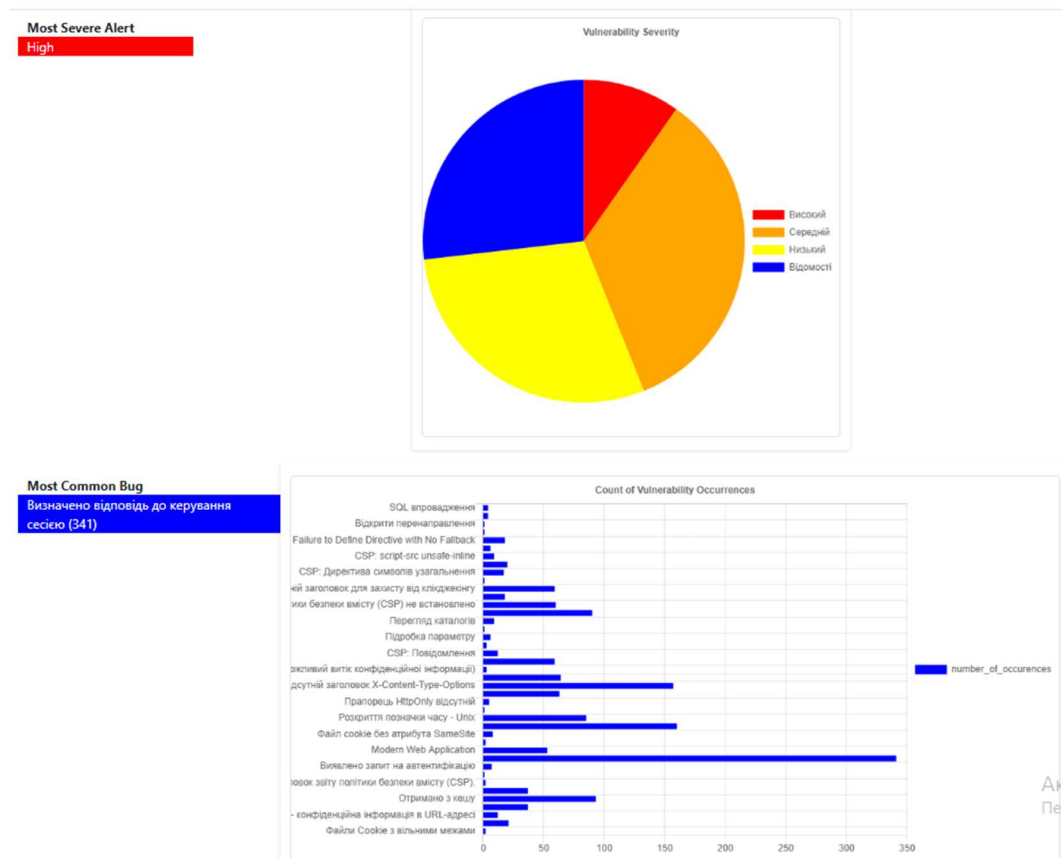


Рисунок 3.15 – Діаграми із результатами сканувань

Окрім візуальних графіків, звіт містить вичерпну таблицю всіх виявлених вразливостей. Кожна проблема в таблиці детально описана, включаючи її рівень ризику (від критичного до низького) та кількість виявлених екземплярів. Ця детальна таблиця охоплює весь спектр виявлених проблем: від SQL-ін'єкцій, які несуть високий ризик компрометації даних, до менш значних, але все ще важливих, таких як відсутність певних заголовків безпеки, що можуть призвести до клікджекінгу або інших атак. Такий формат дозволяє системним адміністраторам та розробникам не лише швидко оцінити поточний стан безпеки, але й ефективно пріоритизувати завдання з усунення недоліків, зосереджуючись на найбільш критичних загрозах.

Ім'я	Рівень ризику	Кількість екземплярів
SQL впровадження	Високий	4
SQL впровадження - MySQL	Високий	4
Відкрити перенаправлення	Високий	1
Зовнішнє перенаправлення	Високий	1
CSP: Failure to Define Directive with No Fallback	Середній	18
CSP: script-src unsafe-eval	Середній	6
CSP: script-src unsafe-inline	Середній	9
CSP: style-src unsafe-inline	Середній	20
CSP: Директива символів узагальнення	Середній	17
Ін'єкція XSLT	Середній	1
Відсутній заголовок для захисту від клікджекінгу	Середній	59
Відсутність токенів Anti-CSRF	Середній	18
Заголовок політики безпеки вмісту (CSP) не встановлено	Середній	60
Міждоменна неправильна конфігурація	Середній	90
Перегляд каталогів	Середній	9
Переповнення буферу	Середній	1
Підробка параметру	Середній	6
Розкриття помилок програми	Середній	3
CSP: Повідомлення	Низький	12
Server Leaks Information via "X-Powered-By" HTTP Response Header Field(s)	Низький	59
Виявлено велике перенаправлення (можливий витік конфіденційної інформації)	Низький	3
Включення міждоменного вихідного файлу JavaScript	Низький	64
Відсутній заголовок X-Content-Type-Options	Низький	157
Заголовок Strict-Transport-Security не встановлено	Низький	63
Прапорця HttpOnly відсутній	Низький	5
Розголошення інформації - повідомлення про помилки налагодження	Низький	1
Розкриття позначки часу - Unix	Низький	85
Сервер поширює інформацію про версію через поле заголовка HTTP-відповіді «Сервер».	Низький	160
Файл cookie без атрибута SameSite	Низький	8
Файли Cookie з атрибутом SameSite відсутні	Низький	2
Modern Web Application	Відомості	53
Визначено відповідь до керування сесією	Відомості	341
Виявлено запит на автентифікацію	Відомості	7
Виявлено запит на перевірку	Відомості	1
Знайдено лише заголовок звіту політики безпеки вмісту (CSP)	Відомості	2
Керований користувачем атрибут HTML-елемента (потенційний XSS)	Відомості	37
Отримано з кешу	Відомості	93
Перегляньте директиви керування кеш-пам'яттю	Відомості	37
Розголошення інформації - конфіденційна інформація в URL-адресі	Відомості	12
Розголошення інформації - підозрілі коментарі	Відомості	21
Файли Cookie з вільними межами	Відомості	2

Рисунок 3.16 – Таблиця вразливостей

Під час аудиту було виявлено кілька серйозних критичних уразливостей, які можуть становити потенційну загрозу для стабільної та безпечної роботи системи. У зв'язку з цим, важливо не лише уважно переглянути підготовлений звіт, але й докладно проаналізувати виявлені недоліки. Лише ретельне вивчення проблемних місць дозволить скласти чіткий план дій для їх усунення та попередження подібних ситуацій у майбутньому.

Отже за результатами сканування веб-додатку DVWA на рівні "Low" за допомогою OWASP ZAP було виявлено значну кількість вразливостей, що підкреслює низький рівень безпеки та відсутність базових механізмів захисту. Наявно декілька рекордних, за своєю кількістю виявлених екземплярів, вразливостей, які становлять найбільший ризик з точки зору їх повсюдного поширення в додатку.

1. Виявлено запит на автентифікацію (Кількість екземплярів: 341)

Ризик: Низький (Інформаційний)

Опис: Це найпоширеніший тип виявлення, який становить рекордну кількість — 341 випадок. Хоча це не є прямою вразливістю, така значна кількість запитів на автентифікацію свідчить про активне використання механізмів входу, форм авторизації або функцій, що вимагають ідентифікації користувача. Це означає, що ці численні точки можуть стати потенційними мішенями для атак, таких як брутфорс чи підбір облікових даних. У той же час, у разі наявності інших вразливостей (наприклад, відсутності захисту від CSRF чи XSS), це може призвести до компрометації облікових записів. Інструмент OWASP ZAP класифікує таке виявлення як інформаційне попередження, однак для забезпечення реального захисту необхідно ретельно перевірити ці точки на наявність супутніх проблем.

2. Розкриття політики безпеки вмісту (CSP) не встановлено (Кількість екземплярів: 90)

Ризик: Середній

Опис: Відсутність політики безпеки вмісту (CSP), яка була зафіксована у 90 випадках, посідає друге місце за своєю поширеністю і свідчить про системну проблему. CSP є важливим механізмом безпеки веб-додатків, який дозволяє

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		65

контролювати джерела вмісту, що можна завантажувати на сторінку (скрипти, стилі, зображення тощо). У разі відсутності політики або її неправильного налаштування веб-додаток стає вразливим до численних атак, зокрема XSS, впровадження шкідливих кодів і крадіжки даних. Хакери можуть використовувати такі проломи для вставлення своїх скриптів або завантаження небезпечних ресурсів із зовнішніх доменів, обходячи стандартні засоби захисту браузера. Той факт, що значна кількість екземплярів демонструє повну відсутність або некоректну конфігурацію CSP, свідчить про вагомі ризики для безпеки всього додатка.

3. Міждоменна неправильна конфігурація (Кількість екземплярів: 60)

Ризик: Середній

Опис: Ще однією значною проблемою є 60 випадків неправильної конфігурації міждоменних запитів. Неправильні налаштування CORS, як-от некоректно визначені заголовки Access-Control-Allow-Origin або інші параметри, можуть дозволити виконання небезпечних запитів із сторонніх доменів без належних обмежень. Це створює умови для атак типу Cross-Origin Resource Sharing (CORS), через які зломисники можуть отримати доступ до конфіденційної інформації або функціоналу веб-додатка. Якщо ці проблеми не усунути, це може призвести до несанкціонованого доступу до даних користувачів чи виконання шкідливих дій від їхнього імені.

На жаль, після проведеного тестування, була можливість згенерувати лише один формат звіту. Решта викликали помилки або виглядали лише як код і, можливо, були призначені для використання іншими додатками. Звіт у цьому форматі добре висвітлює головну інформацію про знайдені вразливості. Але хотілося б більше діаграм та схем, а також більше інформації про сам процес проведення сканувань.

3.3. Впровадження підсистеми сканування в робоче середовище

Однією з критичних вразливостей, знайдених сканером, була сліпа SQL-ін'єкція (Blind SQL Injection). SQL-ін'єкція - це один з найпоширеніших способів атак на веб-додатки, який дозволяє зломиснику втручатися в запити до бази даних. Ця вразливість виникає, коли вхідні дані від користувача без належної обробки

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		66

включаються до SQL-запиту. ZAP ідентифікував цю проблему, вказавши точну URL-адресу, вразливий параметр (id) та тип атаки, який був успішно застосований (1' AND '1'='1).

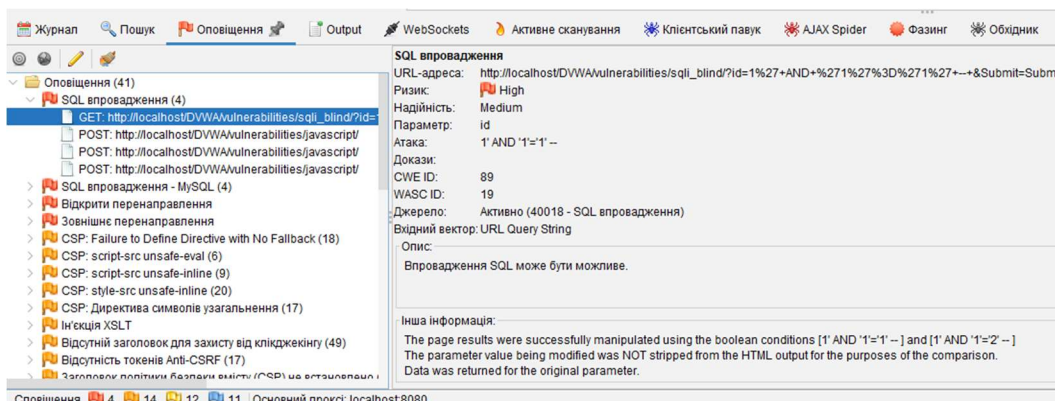


Рисунок 3.17 – Вразливість SQL-ін'єкції

Особливість цього типу атаки полягає в тому, що сервер не повертає явних повідомлень про помилки, і про успішність атаки доводиться судити за непрямими ознаками. Спершу почнемо із найпростішого тесту. Перейшовши за посиланням, що було виявлене за допомогою OWASP ZAP, введемо у активне поле числове значення «1» та упевнимось, що система коректно відобразила повідомлення про наявність у базі користувача із ID номером 1.

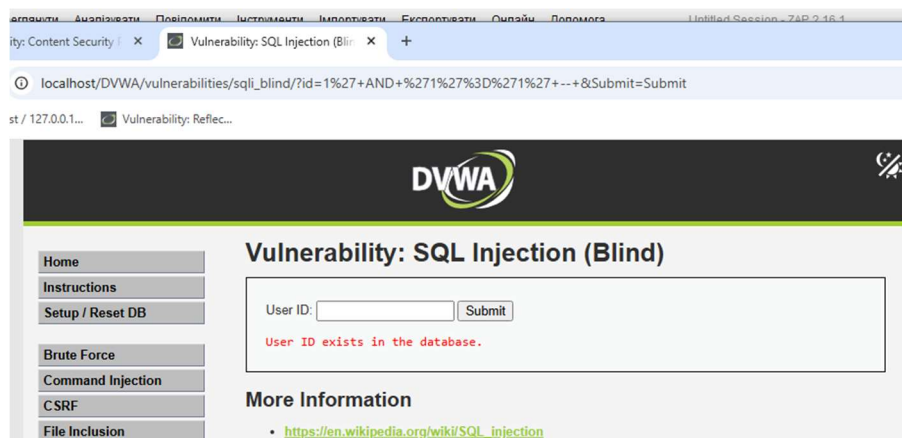


Рисунок 3.18 – Сторінка сайту, на якій була виявлена вразливість

Було вирішено перевірити, як система реагує на логічні умови. Спочатку до параметра ID у URL було додано умову AND 1=1. Оскільки ця умова завжди істинна, система обробила її та повернула повідомлення про наявність користувача з зазначеним ID у базі даних. Це свідчило про те, що додаток виконує SQL-код без належної фільтрації.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				67
Змн.	Арк.	№ докум.	Підпис	Дата		

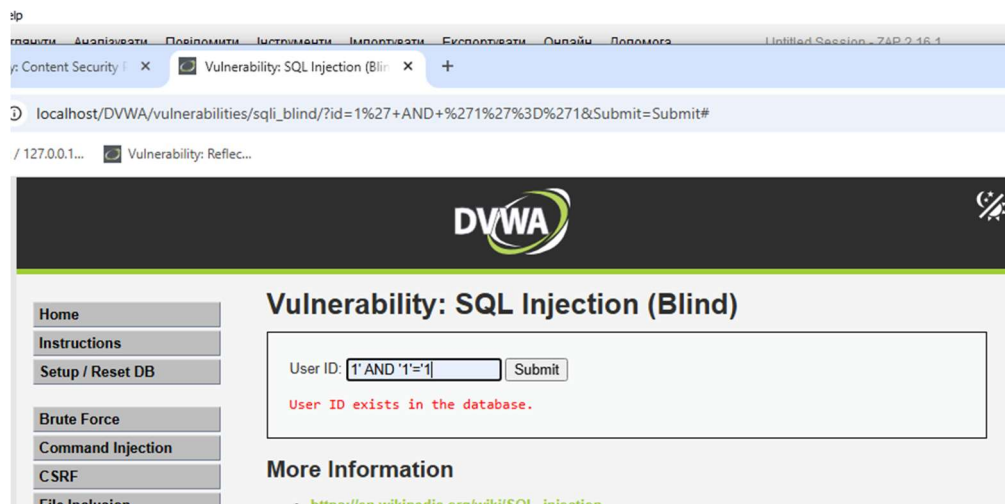


Рисунок 3.19 – Тестування форми

Потім умову було змінено на $AND\ 1=2$, яка завжди хибна. Цього разу сторінка у відповідь видала повідомлення «User ID is MISSING from the database». А отже знову успішно зреагувала на потенційно небезпечний скрипт. Цікавим аспектом є те, що додаток успішно обробив обидва запити, включаючи потенційно небезпечні умови, і продемонстрував різну реакцію залежно від їх логічного значення. Така поведінка підтверджує наявність вразливості до SQL-ін'єкції.

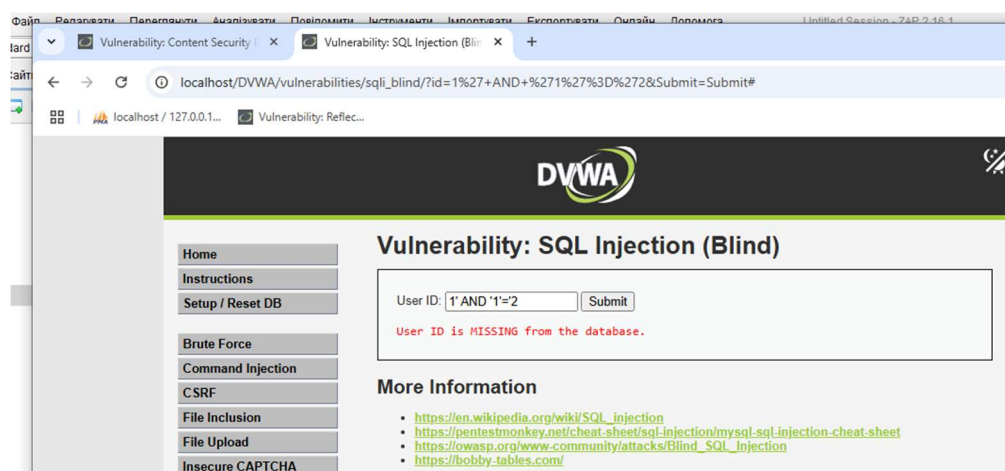


Рисунок 3.20 – Тестування форми

Ще одна важлива перевірка, можливість реалізації якої забезпечує OWASP ZAP – є тестування стійкості форми входу до перебору паролів (Brute Force). Переконавшись, що середовище працює правильно, необхідно перейти на відповідну сторінку та спробувати увійти, ввівши будь-які невірні дані в поля "Логін" та "Пароль", наприклад, test і test. Ця дія змусить браузер відправити POST-запит, який ZAP перехопить та запише у свою історію, що є необхідним кроком для подальшого налаштування фазингу.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				68
Змн.	Арк.	№ докум.	Підпис	Дата		



Рисунок 3.21 – Спроба входу

У головному вікні OWASP ZAP перейдемо на вкладку "Історія". Необхідно знайти запис про POST запит, той що був згенерований при спробі входу із хибними даними. Для даного запиту обираємо тип атаки «Фазинг». Відкриється нове діалогове вікно, у якому надалі будуть налаштовуватися параметри атаки.

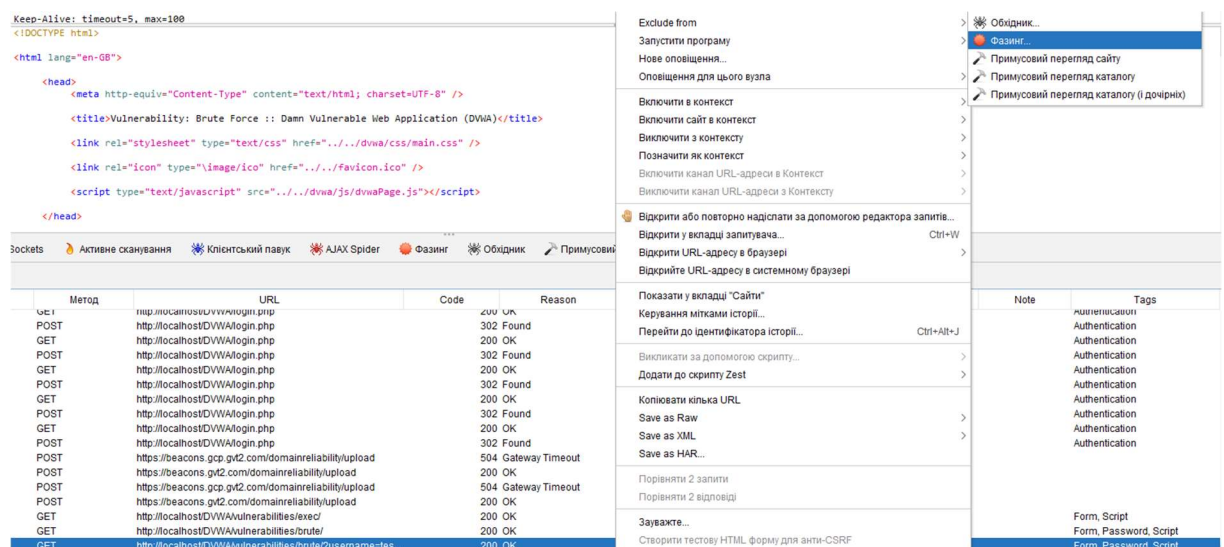


Рисунок 3.22 – Налаштування Фазингу

У тілі даного POST запиту необхідно знайти фрагменти, що відповідають полям username та password. Виділяємо лише значення та додаємо фрагменти як місця для фазингу. Після того, як ми вказали місця для підстановки, необхідно надати ZAP списки (словники) логінів та паролів, які він буде використовувати для перебору. Для першого елемента у списку Розташування, що відповідає username, обираємо файл зі списком логінів. Повторюємо ті самі дії для другого елемента, що відповідає password, обираємо файл зі списком паролів. У даному випадку текстові файли users.txt та password.txt були створені заздалегідь.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				69
Змн.	Арк.	№ докум.	Підпис	Дата		

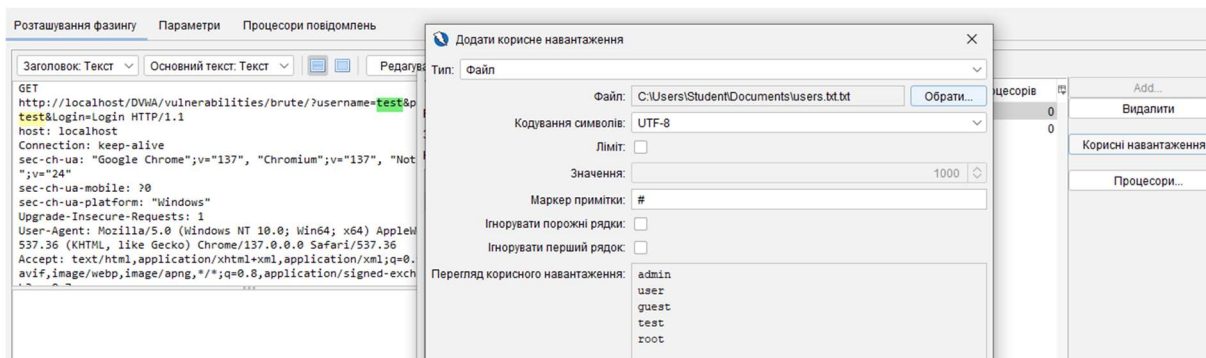


Рисунок 3.23 – Додавання списку можливих користувачів

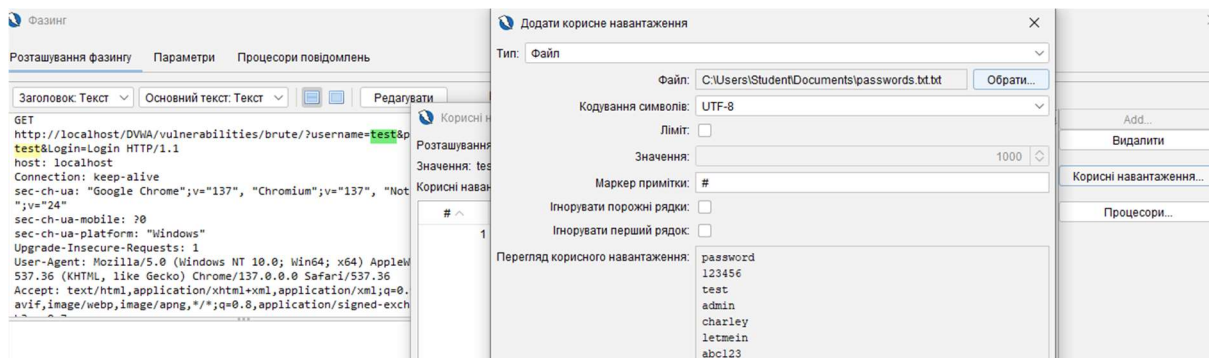


Рисунок 3.24 – Додавання списку можливих паролів

На цьому основні налаштування фазера завершені і програма готова до запуску процесу переборів паролів та логінів у виділених місцях із використанням власних словників.

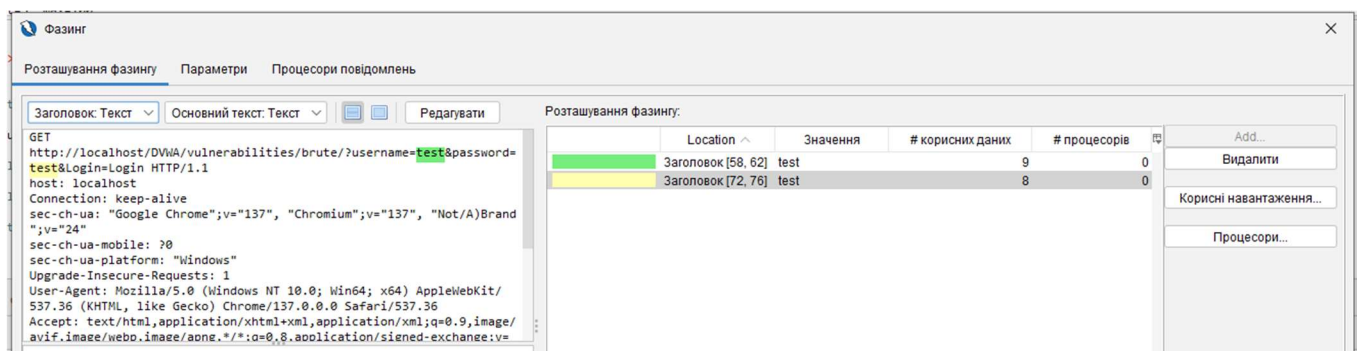


Рисунок 3.25 – Налаштування Фазингу

Після запуску інструмент почав автоматично надсилати сотні запитів на вхід. А в результаті на вкладці «Фазинг» можна ознайомитися із таблицею, яка відображає всі запити, які ZAP відправив. У даному випадку найважливішим індикатором при пошуку розбіжностей став «Розмір». Відсортувавши результати від найменшого до найбільшого, було знайдено декілька аномальних результатів. Розмір відповіді у них значно відрізняється від більшості, хоча решта параметрів збігаються із звичайними невдалими спробами пошуку. Було знайдено чотири

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				70
Змн.	Арк.	№ докум.	Підпис	Дата		

таких відповіді і для кожної з них необхідно уважно переглянути текстові фрагменти у коді. Оскільки це успішні входи, нам повідомляють, що «Вітаємо у середовищі захищених паролів».

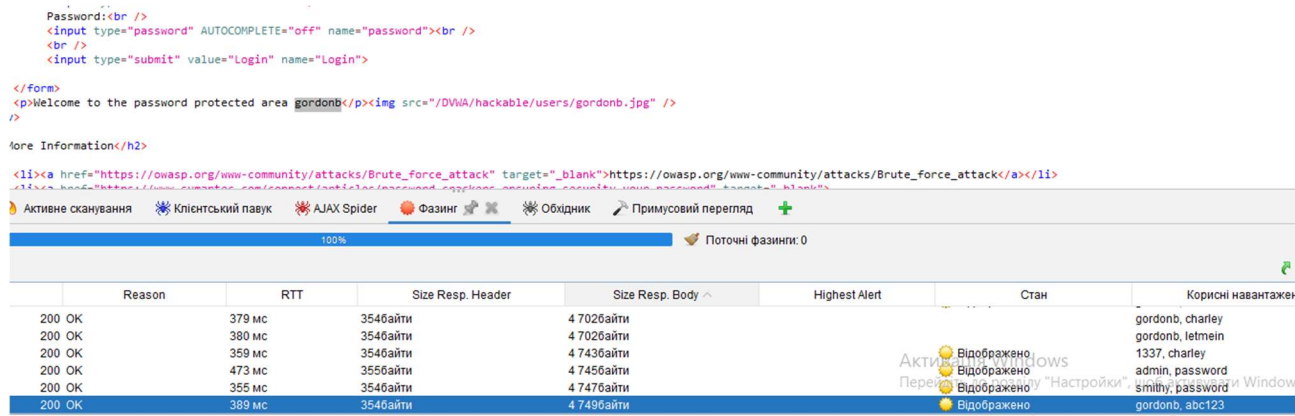


Рисунок 3.26 – Результат перебірки паролів

Заключним кроком стало використання одного з логінів та паролів зі знайденого списку для успішного входу через форму. Було вибрано обліковий запис smithy, і пароль виявився правильним. Система підтвердила це, привітавши з поверненням і підтвердивши особу користувача smithy.

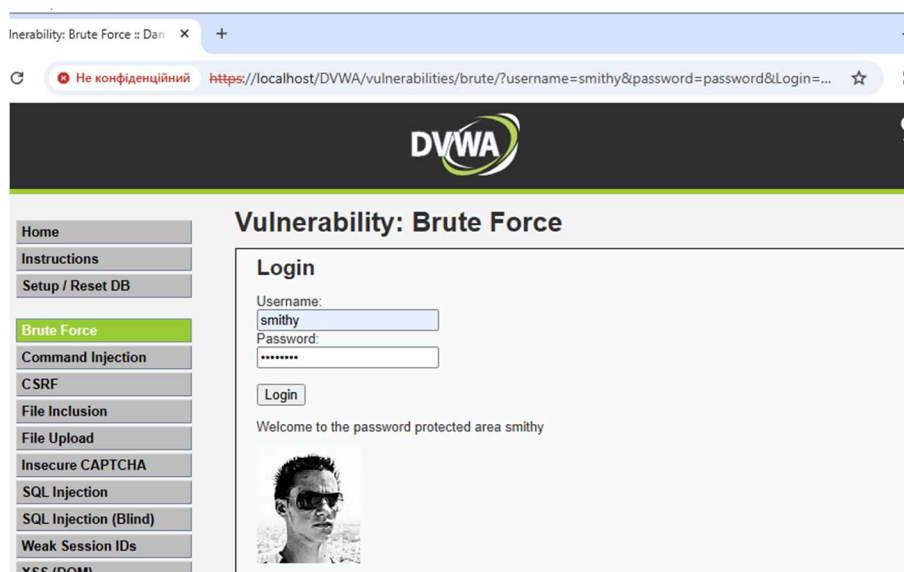


Рисунок 3.27 – Перевірка знайдених логіну та паролю

3.4. Рекомендації щодо захисту веб-додатків та аналіз роботи

Під час аналізу та сканування веб-додатку за допомогою OWASP ZAP було виявлено кілька важливих моментів, які варто враховувати розробникам, архітекторам і DevOps-фахівцям. Щоб забезпечити належний рівень безпеки, варто впроваджувати її ще на ранніх етапах розробки, а не залишати на останній момент

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				71
Змн.	Арк.	№ докум.	Підпис	Дата		

перед релізом. Інтеграція безпеки в процеси CI/CD у межах підходу DevSecOps допомагає вчасно виявляти та виправляти потенційні вразливості до того, як продукт буде розгорнуто в робочому середовищі.

Основним принципом є недовіра до будь-яких вхідних даних. Усі дані, що надходять до застосунку, мають обов'язково проходити ретельну валідацію на стороні сервера. Щоб уникнути SQL-ін'єкцій, рекомендовано використовувати параметризовані запити (prepared statements) або ORM-системи, які позбавляють розробника необхідності безпосередньо формувати SQL-запити. Практика демонструє, що навіть у разі сліпої SQL-ін'єкції її можна успішно використати, якщо обробка даних виконується неналежним чином.

Щоб запобігти міжсайтовому скриптингу (XSS), усі дані, які виводяться на сторінку, необхідно екранувати. Контент, що надходить від користувача і виводиться у HTML, слід обробляти, замінюючи спеціальні символи на їх безпечні аналоги. Додатковим рівнем захисту є застосування політики Content Security Policy, яка обмежує джерела завантаження скриптів, стилів та інших ресурсів, ускладнюючи потенційну експлуатацію вразливостей XSS[33].

Надійна система автентифікації та авторизації має ключове значення для захисту від атак перебором паролів і несанкціонованого доступу до системи. Для цього необхідно впроваджувати обмеження на кількість невдалих спроб входу, активувати блокування облікових записів після досягнення критичного порогу, застосовувати складні парольні політики та використовувати багатофакторну автентифікацію (MFA). Крім того, контроль доступу слід реалізовувати на серверній стороні, ретельно перевіряючи права користувача на кожен запит — як на виконання певної дії, так і на доступ до захищених ресурсів.

Особливу увагу варто звернути на коректну конфігурацію безпеки всього технологічного стеку. Це передбачає використання актуальних версій програмного забезпечення, вимкнення режиму налагодження на продуктивних серверах, а також правильне налаштування HTTP-заголовків безпеки для захисту від атак, таких як clickjacking, MIME-sniffing тощо. Вразливості, які виявляють сканери, часто зумовлені саме відсутністю або помилковим налаштуванням цих заголовків[34].

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		72

Комплексна безпека веб-додатків є безперервним процесом, а не разовим заходом. Вона включає в себе безпечне написання коду, правильну конфігурацію систем, надійну автентифікацію, регулярний моніторинг і своєчасне оновлення. Інструменти на кшталт OWASP ZAP допомагають ефективно знаходити поширені вразливості, проте жоден інструмент не замінить цілісного підходу до безпеки, що ґрунтується на принципах перевірки й недовіри, а також концепції «безпека за замовчуванням». Для реального захисту веб-додатків, окрім сканерів, варто впроваджувати комплексні рішення, які доповнюють і посилюють загальну систему безпеки[29].

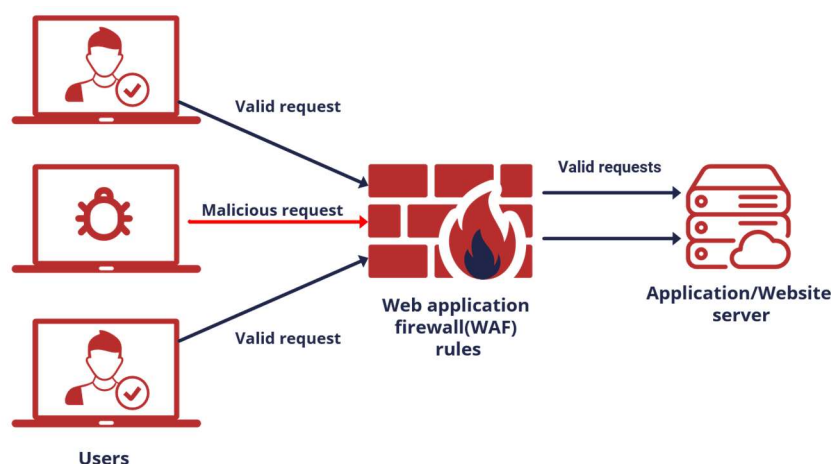


Рисунок 3.28 – Брандмауер веб-додатків

Web Application Firewall (WAF) виконує роль захисного бар'єра між вашим додатком і мережею Інтернет, забезпечуючи фільтрацію HTTP-трафіку, блокування атак типу XSS і SQL-ін'єкцій, захист API, а також пропонуючи можливості для тонкого налаштування правил доступу. Сучасні рішення WAF, такі як NSFocus, Cloudflare, AWS WAF та Azure WAF, мають підтримку віртуального патчування, захисту від ботів, аналізу поведінки трафіку, продуктивну інтеграцію з хмарними платформами та опції для гнучкого розгортання як у хмарному середовищі, так і на локальних серверах. Ці системи не лише виявляють загрози, але й ефективно блокують їх у реальному часі, мінімізуючи ризик компрометації навіть при наявності не виправлених вразливостей[26].

Runtime Application Self-Protection (RASP)

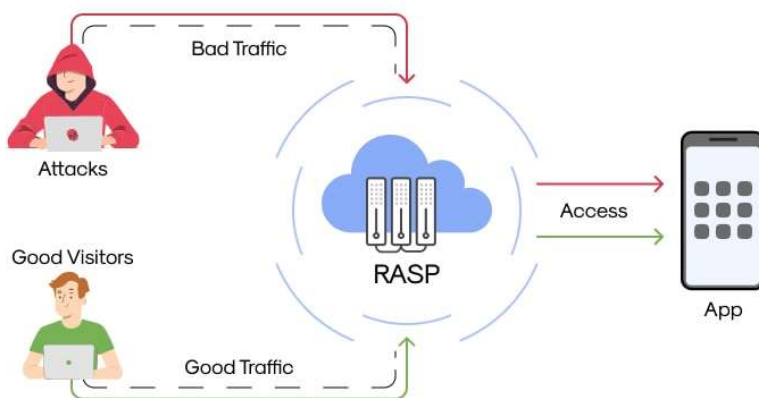


Рисунок 3.29 – Самозахист програми під час їх використання

Runtime Application Self-Protection (RASP) — це сучасна технологія кібербезпеки, яка інтегрується безпосередньо в додаток, забезпечуючи захист у реальному часі. Вона контролює виконання коду, аналізує контекст запитів та поведінку програми під час її роботи. У разі виявлення підозрілої чи шкідливої активності RASP здатна автоматично зупиняти виконання потенційно небезпечних операцій, запобігаючи використанню вразливостей. Окрім цього, технологія надає детальну аналітику про зафіксовані загрози, що дозволяє швидко реагувати на інциденти. Завдяки RASP застосунок залишається захищеним навіть за наявності недоліків у його коді[32].



Рисунок 3.30 – Приклади популярних менеджерів конфігурацій

Менеджери секретів та конфігурацій — це спеціалізовані інструменти для безпечного зберігання, управління та контролю доступу до конфіденційних даних, таких як паролі, API-ключі, токени доступу, сертифікати та інші критично важливі елементи. Вони дозволяють централізовано керувати чутливою інформацією, автоматизувати її оновлення та встановлювати політики доступу на основі ролей користувачів або сервісів. Інструменти на кшталт HashiCorp Vault, AWS Secrets Manager або Azure Key Vault сприяють зниженню ризиків, пов'язаних з витоком

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				74
Змн.	Арк.	№ докум.	Підпис	Дата		

даних через неналежне зберігання секретів у коді, репозиторіях або конфігураційних файлах.

Використання такого набору інструментів дає можливість не лише виявляти та ефективно нейтралізувати атаки, але й зменшувати площу уразливості, відповідати сучасним вимогам кібербезпеки та підвищувати загальний рівень довіри до веб-додатка. Дотримання зазначених рекомендацій разом із впровадженням актуальних засобів захисту дозволить значно знизити ризики компрометації, зміцнити довіру користувачів та забезпечити стабільну й безпечну роботу веб-додатка в умовах зростаючої кількості кіберзагроз.

Враження від роботи з OWASP ZAP та DVWA загалом можна оцінити як позитивні, адже ці інструменти демонструють високий рівень функціональності та забезпечують зручний процес використання. Вони дійсно належать до числа найкращих застосунків у своїй галузі, що зробило взаємодію з ними ефективною та приємною.

Використання DVWA як цільової платформи для тестування внесло відчутні переваги. Цей спеціалізований вебдодаток створений саме для навчання та тренувань у сфері інформаційної безпеки. Він містить навмисно вбудовані вразливості в контрольованому середовищі, що дозволило проводити повноцінне тестування без ризику завдати шкоди реальним сервісам чи порушити законодавство. Важливою перевагою DVWA є наявність кількох рівнів складності (Low, Medium, High, Impossible). Усе тестування було проведено на рівні «Low», щоб сканер точно захопив більшість вразливостей. Та і при встановленні даний веб-додаток не має ніяких проблем. Процес розгортання тестового середовища був значно спрощений завдяки використанню дистрибутива XAMPP, який об'єднує вебсервер Apache, сервер бази даних MySQL та інтерпретатор PHP в єдиному інсталяційному пакеті.

Сканер OWASP ZAP, у свою чергу, зарекомендував себе як потужний, багатофункціональний та надійний помічник для проведення тестів безпеки. Будучи безкоштовним продуктом із відкритим вихідним кодом, він залишається доступним для широкого кола користувачів і водночас підтримує високу якість

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				75
Змн.	Арк.	№ докум.	Підпис	Дата		

виконання завдань[30]. Його інтуїтивно зрозумілий графічний інтерфейс забезпечує легкість навігації, навіть для початківців. Особливо зручним у ручному режимі став інструмент HUD (Heads-Up Display), який інтегрує панелі ZAP прямо у вікно браузера. Вбудовані автоматичні інструменти значно прискорили процес, автоматично виявивши безліч посилань та потенційних проблем. Коли під час аналізу виявлялася вразливість, ZAP представляв детальний опис проблеми разом із рекомендаціями щодо її усунення. Крім того, програма надавала посилання на додаткові джерела для більш глибокого вивчення питання і чітко вказувала на конкретні параметри чи типи атак.

Незважаючи на численні переваги та широкий функціонал, під час практичної реалізації було виявлено ряд труднощів і недоліків. По-перше, спостерігалось значне навантаження на системні ресурси. Одночасне використання віртуальної машини з Windows, вебсервера XAMPP і сканера OWASP ZAP, який потребує середовища виконання Java, створювало істотний тиск на апаратну складову. По-друге, суттєвою проблемою виявився початковий етап налаштування інтеграції між сканером і браузером. Оскільки ZAP працює як проксі-сервер, для перехоплення трафіку довелося виконати кілька ручних дій, які не завжди успішно співпрацювали між собою. А тому багато часу було витрачено на узгодження усіх необхідних налаштувань для подальшої роботи[34].

OWASP ZAP є хорошим самостійним сканером безпеки, тому тяжко визначити перспективи, що варто внести у підсистему. Але з огляду на виявлені недоліки, все таки є можливим визначити напрямки для вдосконалення:

- Було б доцільно оптимізувати початкову конфігурацію. Можна розглянути варіант розробки скриптів або використання вбудованих механізмів для автоматизованого налаштування системного проксі та впровадження кореневого SSL-сертифіката ZAP у сховища довірених сертифікатів популярних браузерів.

- Крім того, варто звернути увагу на зменшення вимог до апаратного забезпечення. Перспективним напрямком може бути дослідження альтернативних, менш ресурсомістких підходів для розгортання тестового середовища, таких як використання контейнеризації (Docker) замість застосування повноцінної

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				76
Змн.	Арк.	№ докум.	Підпис	Дата		

віртуальної машини. Це дозволить підвищити продуктивність та зробить підсистему доступнішою для використання на менш потужних робочих станціях.

Висновки до розділу:

У третьому розділі виконано реалізовано та проаналізовано роботу розробленої підсистеми сканування безпеки на прикладі веб-додатка DVWA. Початку було здійснено ручне дослідження додатка за допомогою інструменту OWASP ZAP із використанням HUD, що спростило взаємодію з сайтом і аналіз трафіку. Успішна аутентифікація та налаштування низького рівня безпеки дозволили виявити максимум потенційних вразливостей. За допомогою інструмента «Павук» виконано автоматизоване сканування структури сайту, що дозволило створити повну карту URL-адрес для подальшого поглибленого аналізу.

Використовуючи контекст та облікові дані, активний сканер ZAP зафіксував 41 сповіщення про вразливості: 4 високого рівня ризику, 14 середнього, 12 низького і 11 інформаційного характеру. Було успішно підтверджено наявність Reflected Cross-Site Scripting, а також сліпої SQL-ін'єкції. Крім цього, за допомогою інструменту «Фазинг» вдалося реалізувати атаку перебору паролів.

Результати сканування узагальнені в детальному звіті, який включає візуалізацію статистики, таблицю виявлених вразливостей із їх описами, рівнями ризику та рекомендаціями щодо усунення кожної проблеми. На основі отриманих даних створено список заходів для посилення безпеки веб-додатка. Практична реалізація підтвердила високу ефективність розробленої підсистеми.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		77

ВИСНОВКИ

У даній кваліфікаційній роботі було розроблено проект підсистеми сканування безпеки веб-додатків з використанням програмного забезпечення OWASP ZAP для підвищення ефективності виявлення вразливостей та покращення загального рівня кібербезпеки веб-ресурсів.

Перший розділ роботи заклав міцний теоретичний фундамент, досліджуючи комплексну концепцію безпеки веб-додатків як безперервного процесу. Проведено аналіз сучасних підходів до тестування безпеки та найпоширеніших вразливостей веб-додатків, таких як SQL-ін'єкції та XSS. Проведено порівняльний аналіз сучасних інструментів для сканування, таких як OpenVAS, Nmap, Wazuh, Nikto та ClamAV, а також обґрунтовано вибір OWASP ZAP як основного інструменту завдяки його спеціалізації на динамічному тестуванні безпеки веб-додатків, функціональності проксі-сервера та гнучкості налаштувань, що якнайкраще відповідає меті роботи. У теоретичній частині детально розглянуто найпоширеніші вектори атак та вразливості згідно з класифікацією OWASP Top 10, зокрема SQL-ін'єкції, міжсайтовий скриптинг, порушення контролю доступу та неправильні конфігурації безпеки. Цей розділ підкреслив, що комплексний підхід до безпеки, що базується на принципах «не довіряй, перевіряй» та «безпека за замовчуванням», є єдиним ефективним шляхом у протидії сучасним кіберзагрозам.

Другий розділ був присвячений практичній підготовці та проектуванню архітектури підсистеми сканування, що ґрунтується на застосуванні OWASP ZAP. Вона складається з ізолюваного тестового середовища, розгорнутого на базі віртуальної машини, де функціонує навмисно вразливий веб-додаток DVWA. Було детально описано процес встановлення, конфігурації та налаштування OWASP ZAP, включаючи включи конфігурацію системного проксі-сервера для перехоплення трафіку та імпорт кореневого SSL-сертифіката ZAP у браузер для аналізу зашифрованих HTTPS-з'єднань. На завершення було створено контекст для цільового додатка та налаштовано параметри автентифікації для можливості сканування захищених сторінок від імені користувача. Таким чином, другий розділ

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				78
Змн.	Арк.	№ докум.	Підпис	Дата		

успішно продемонстрував функціональність розробленої підсистеми та її здатність ефективно виявляти вразливості в контрольованому середовищі.

У третьому розділі проведено практичне тестування DVWA за допомогою OWASP ZAP. Виконано ручне сканування, активне сканування та атаку з використанням інструменту "Павук" для складання карти структури сайту. Також проведено тестування атаки перебором паролів (Brute Force) за допомогою фазингу, що підтвердило ефективність OWASP ZAP у виявленні слабких місць у системах автентифікації. Виявлено критичні вразливості, такі як SQL-ін'єкції, XSS та відсутність захисту від клікджекінгу. А загалом, під час сканування за допомогою OWASP ZAP у веб-додатку було виявлено 41 вразливість, що сумарно нараховує близько 1552 екземплярів. Згенеровано звіти, які наочно демонструють результати сканування, включаючи рівні ризику та рекомендації щодо усунення проблем. На основі результатів проведеного сканування були розроблені практичні рекомендації для захисту веб-додатків. До них відносяться валідація вхідних даних, використання параметризованих SQL-запитів, впровадження політики CSP, застосування HTTPS із сучасними алгоритмами шифрування, а також додаткові методи захисту, такі як WAF і RASP.

Таким чином, усі поставлені завдання кваліфікаційної роботи були повністю виконані, що дозволило досягти її мети. Створений проект підсистеми сканування є завершеним практичним рішенням, що може бути використане для підвищення рівня безпеки веб-додатків. Отримані результати можуть слугувати основою для подальших досліджень, таких як розширення функціоналу підсистеми або адаптація її для роботи з хмарними середовищами.

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		79

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Penetration Testing (PT) – Consulting services in Cyber and Information Security [Електронний ресурс]. – Режим доступу: <https://www.securifynet.com/service/penetration-testing-pt/> (дата звернення: 30.05.2025).
2. How vulnerability risk ratings are used [Електронний ресурс]. – Режим доступу: https://help.ivanti.com/iv/help/en_US/RS/vNow/How-Vulnerability-Risk-Ratings-VRR-Are-Used.htm (дата звернення: 30.05.2025).
3. David (2025): Cybersecurity: Detecting trends and implementing Strategies - Glamourglom, Glamourglom, [Електронний ресурс]. – Режим доступу: <https://glamourglom.com/cybersecurity-detecting-trends-and-implementing-strategies/> (дата звернення: 31.05.2025).
4. Open vs. closed Networks | blog [Електронний ресурс]. – Режим доступу: <https://www.coinmena.com/en/blog/1YXFJTzDU5lFUVXxdgIrtY> (дата звернення: 30.05.2025).
5. Mehta, Janki (2023): Mitigating the OWASP Top 10 Vulnerabilities: Strategies for Protecting Your Systems, EncryptedFence by Certera - Web & Cyber Security Blog, [Електронний ресурс]. – Режим доступу: <https://certera.com/blog/mitigating-the-owasp-top-10-vulnerabilities/> (дата звернення: 01.06.2025).
6. TryHackMe (n.d.): OpenVAS, TryHackMe, [Електронний ресурс]. – Режим доступу: <https://tryhackme.com/room/openvas> (дата звернення: 29.04.2025).
7. Nmap: мережевий маппер — безкоштовний сканер безпеки [Електронний ресурс]. – Режим доступу: <https://nmap.org/> (дата звернення: 28.04.2025).
8. Avada (2025): How many endpoints can Wazuh handle?, KingsGuard, [Електронний ресурс]. – Режим доступу: <https://kings-guard.com/how-many-endpoints-can-wazuh-handle/> (дата звернення: 29.04.2025).

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				80
Змн.	Арк.	№ докум.	Підпис	Дата		

9. Shaman (2023): Обзор сканера Nikto для поиска уязвимостей в веб-серверах, Хабр, [Електронний ресурс]. – Режим доступу: <https://habr.com/ru/companies/first/articles/731696/> (дата звернення: 29.04.2025).
10. Вікімедіа, Учасники Проектів (2008): ClamAV, [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki/ClamAV> (дата звернення: 29.04.2025)
11. Beschokov, Mukhadin (2025): OWASP ZAP - ZEd Attack Proxy, Wallarm, [Електронний ресурс]. – Режим доступу: <https://www.wallarm.com/what/owasp-zap-zed-attack-proxy> (дата звернення: 30.04.2025).
12. Testing for XSS and other vulnerabilities with OWASP ZAP | Блог та тренінги з тестування безпеки від Логин Свята (n.d.): Блог Та Тренінги З Тестування Безпеки Від Логин Свята, [Електронний ресурс]. – Режим доступу: <https://svyat.tech/testing-for-xss-and-other-vulnerabilities-with-owasp-zap> (дата звернення: 22.05.2025).
13. Academy, Eitca (2024): Яка основна мета використання OWASP ZAP у тестуванні на проникнення веб-додатків? - Академія ЕІТСА, ЕІТСА Academy, [Електронний ресурс]. – Режим доступу: <https://uk.eitca.org/cybersecurity/eitc-is-wapt-web-applications-penetration-testing/hidden-files/discovering-hidden-files-with-zap/examination-review-discovering-hidden-files-with-zap/what-is-the-primary-purpose-of-using-owasp-zap-in-web-application-penetration-testing/> (дата звернення: 04.06.2025).
14. Ltd, Fintex (2023): RoleCatcher | Посібник OWASP ZAP: Оволодіння навичками тестування безпеки веб-додатків, RoleCatcher.com, [Електронний ресурс]. – Режим доступу: <https://rolecatcher.com/uk/skills/knowledge/information-and-communication-technologies/software-and-applications-development-and-analysis/owasp-zap/> (дата звернення: 29.04.2025).

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				81
Змн.	Арк.	№ докум.	Підпис	Дата		

- 15.Hub, CyberSkills (2025): How to detect web app vulnerabilities with OWASP ZAP, CyberSkillsHub, [Електронний ресурс]. – Режим доступу: <https://cyberskillshub.com/en/detect-web-app-vulnerabilities-owasp-zap/> (дата звернення: 29.04.2025).
- 16.Admin/Admin (2024): Як захистити вебсайт від кіберзагроз у 2025 році, Як захистити вебсайт від кіберзагроз: Тренди кібербезпеки у 2025 році › WEB DEV & SEO, [Електронний ресурс]. – Режим доступу: <https://webdevandseo.com/how-to-protect-a-website-from-cyber-threats/> (дата звернення: 05.06.2025).
- 17.Borges, Esteban (2024): Top 12 vulnerable websites for legal penetration testing training, Recorded Future, [Електронний ресурс]. – Режим доступу: <https://www.recordedfuture.com/threat-intelligence-101/vulnerability-management-threat-hunting/vulnerable-websites-for-penetration-testing> (дата звернення: 16.05.2025).
- 18.Infosec Train (2025): What is OWASP Zed Attack Proxy (ZAP)?, InfosecTrain, [Електронний ресурс]. – Режим доступу: <https://www.infosectrain.com/blog/what-is-owasp-zed-attack-proxy-zap/> (дата звернення: 01.05.2025).
- 19.Administrator (2025): Top 3 most popular web application vulnerabilities - security scans of 396 open source web applications, Firewall.cx, [Електронний ресурс]. – Режим доступу: <https://www.firewall.cx/security/web-application-vulnerability-scanners/most-popular-web-application-vulnerabilities.html> (дата звернення: 02.06.2025).
- 20.Academy, Eitca (2023): Що таке Damn Vulnerable Web Application (DVWA) і чому його рекомендують для практичного тестування безпеки веб-додатків? - Академія ЕІТСА, ЕІТСА Academy, [Електронний ресурс] . – Режим доступу: <https://uk.eitca.org/cybersecurity/eitc-is-wapt-web-applications-penetration-testing/spidering/spidering-and-dvwa/examination-review-spidering-and-dvwa/what-is-the-damn-vulnerable-web-application> (дата звернення: 16.05.2025).

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				82
Змн.	Арк.	№ докум.	Підпис	Дата		

21. Diginiinja (n.d.): GitHub - diginiinja/DVWA: Damn Vulnerable Web Application (DVWA), GitHub, [Електронний ресурс]. – Режим доступу: <https://github.com/diginiinja/DVWA> (дата звернення: 16.05.2025).
22. Денисенко, Андрій (2025): Посібник із проксі-серверів для початківців, in: Robot_Dreams, 13.05.2025, [Електронний ресурс]. – Режим доступу: <https://robotdreams.cc/uk/blog/481-posibnik-iz-proksi-serveriv-dlya-pochatkivciv> (дата звернення: 03.06.2025).
23. Hung Nguyễn (2025): Những điểm khác nhau giữa Proxy và VPN, Vietnix, [Електронний ресурс]. – Режим доступу: <https://vietnix.vn/proxy-va-vpn/> (дата звернення: 03.06.2025).
24. ZAP – Сертифікати сервера [Електронний ресурс]. – Режим доступу: <https://www.zaproxy.org/docs/desktop/addons/network/options/servercertificates/> (дата звернення: 08.05.2025).
25. Khine, Paing Thet/Paing Thet Khine (2025): What is SSL?, Fusion Solution, [Електронний ресурс]. – Режим доступу: <https://www.fusionsol.com/blog/ssl-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3/> (дата звернення: 08.05.2025).
26. Ilyin, Stepan (2025): SAST, DAST, IAST, and RASP, Wallarm, [Електронний ресурс]. – Режим доступу: <https://www.wallarm.com/what/sast-dast-iaast-and-rasp> (дата звернення: 04.06.2025).
27. Marriott, Paul (2023): ZAP Proxy and DVWA - Paul Marriott - Medium, in: Medium, 18.11.2023, [Електронний ресурс]. – Режим доступу: <https://paulmarriott.medium.com/zap-proxy-and-dvwa-cc02d5b72a75> (дата звернення: 22.05.2025).
28. Shaman (2023): Обзор OWASP ZAP. Сканер для поиска уязвимостей в веб-приложениях, Хабр, [Електронний ресурс]. – Режим доступу: <https://habr.com/ru/companies/first/articles/709586/> (дата звернення: 10.06.2025).

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				83
Змн.	Арк.	№ докум.	Підпис	Дата		

29. Adam Novotný (2022): What is a Web Application Firewall (WAF) and Why Use it?, [Електронний ресурс]. – Режим доступу: <https://www.stormit.cloud/blog/what-is-a-web-application-firewall/> (дата звернення: 10.06.2025).
30. Matsiiako, Vlad (2025): Top-10 Secrets Management Tools in 2025, Infisical Blog, [Електронний ресурс]. – Режим доступу: <https://infisical.com/blog/best-secret-management-tools?> (дата звернення: 06.06.2025).
31. Magalhães, Manu (2024): OWASP Zap: DAST tool for Automated Security Testin [Електронний ресурс]. – Режим доступу: <https://manumagalhaes.medium.com/owasp-zap-for-automated-dast-b70322b8309a> (дата звернення: 10.06.2025).
32. Yadav, Abhishek (2023): Secrets unveiled: AWS Secrets Manager - Abhishek Yadav - medium, in: Medium, [Електронний ресурс]. – Режим доступу: <https://aplusbsquare.medium.com/secrets-unveiled-aws-secrets-manager-2bf659a1d778> (дата звернення: 10.06.2025).
33. Penetration Testing A Hands-On Introduction to Hacking [навчальний посібник] / Georgia Weidman — San Francisco: no starch press, 2014. — 531 с (дата звернення: 21.04.2025)
34. Network Vulnerability Assessment. Identify security loopholes in your network's infrastructure. [навчальний посібник] / Sagar Rahalkar — Birmingham: Packt Publishing Ltd., 2018. — 243 с (дата звернення: 20.04.2025)

		Опанасенко Є.Р.			ККІКБ.КР.Б – 125 – 14 – ПЗ	Арк.
		Покотило О.А.				
Змн.	Арк.	№ докум.	Підпис	Дата		84