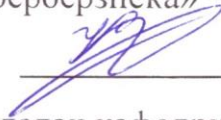


Міністерство освіти і науки України
Державний університет «Житомирська політехніка»
Факультет інформаційно-комп'ютерних технологій
Кафедра комп'ютерної інженерії та кібербезпеки

ПОЯСНЮВАЛЬНА ЗАПИСКА

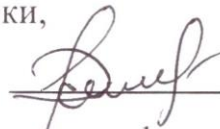
до випускної кваліфікаційної роботи бакалавра
на тему: «Проект підсистеми сканування безпеки веб-
додатків за допомогою програмного забезпечення Owasp
Zap»

Виконала студентка 4-го курсу групи КБ-19-2
спеціальності 125 «Кібербезпека»



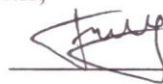
Наталія Ковбан

Керівник старший викладач кафедри комп'ютерної
інженерії та кібербезпеки,



Єлизавета Байлюк

Рецензент старший викладач кафедри комп'ютерної
інженерії та кібербезпеки,



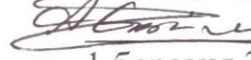
Олександра Покотило

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
1 березня 2023 р.

ЗАВДАННЯ

на випускн кваліфікаційну роботу бакалавра

Студент: Ковбан Наталія Сергіївна

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

затверджена Наказом університету від 1 березня 2023 р. № ____

Термін здачі студентом закінченої роботи 23 червня 2023 р.

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

Об'єкт дослідження: процес конфігурації та використання за призначенням інструменту Owasp Zap.

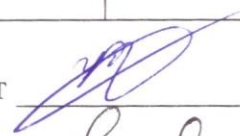
Предмет дослідження: технологія сканування безпеки веб-сайту за допомогою утиліти Owasp Zap.

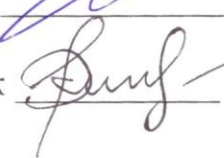
Обладнання та програмне забезпечення:

1. Windows 10
2. Mozilla Firefox
4. OWAP Zap

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

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Байлюк Є.М.	01.03.2023	
2	Єфіменко А.А.	01.03.2023	

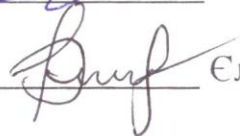
Студент  Наталія Ковбан

Керівник  Єлизавета Байлюк

Календарний план

№ з/п	Назви етапів випускної кваліфікаційної роботи	Термін виконання етапів роботи	Примітка
1	Постановка завдання	01.03.2023	виконано
2	Розробка плану роботи	10.03.2023	виконано
3	Пошук та огляд літературних джерел	25.03.2023	виконано
4	Аналіз можливих загроз ІБ	29.03.2023	виконано
5	Дослідження стратегій безпеки веб-сайтів	03.04.2023	виконано
6	Огляд утиліт для пентестингу	10.04.2023	виконано
7	Ознайомлення з утилітою Owasp Zap	14.04.2023	виконано
8	Дослідження можливостей пз Owasp Zap	19.04.2023	виконано
9	Встановлення та налаштування утиліти	23.04.2023	виконано
10	Проведення сканування	27.04.2023	виконано
11	Моделювання атак	01.05.2023	виконано
12	Аналіз результатів тестування	15.05.2023	виконано
13	Написання пояснювальної записки	20.05.2023	виконано
14	Підготовка презентації	01.06.2023	виконано
15	Отримання відгуку керівника та рецензії	15.06.2023	виконано
16	Захист роботи	23.06.2023	виконано

Студент  Наталія Ковбан

Керівник  Єлизавета Байлюк

АНОТАЦІЯ

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

Метою роботи є аналіз та моделювання кіберінцидентів, які можуть вказати на лазівки в безпеці веб-сайту для подальшого їх усунення.

Безпека являється першочерговим важливим питанням в розробці веб сайту, оскільки вразливості можуть привести до серйозних втрат з боку бізнесу. Розробники можуть допускати помилки при написанні коду, або не усвідомлювати всю важливість використання прийомів безпечного програмування – все це може призвести до появи вразливостей.

В рамках проєкту було проаналізовано певні типи вразливостей, а також правила та політики, які допоможуть запобігти даним кіберінцидентам. Проведено сканування веб-сайту на наявність вразливостей та досліджено поведінку веб-сайту під їхнім впливом.

Предметом дослідження є технологія сканування безпеки веб-сайту за допомогою утиліти Owasp Zар.

Об'єктом дослідження є процес конфігурації та використання за призначенням інструменту Owasp Zар.

Пояснювальна записка складається зі вступу, трьох розділів, висновків, списку умовних скорочень, списку використаної літератури.

Теоретична частина була викладена на 62 сторінках друкованого тексту. Список використаних джерел складається з 24 позиції та займає 2 сторінки. В роботі наведено 46 рисунків. Загалом обсяг роботи складає 70 сторінок.

КЛЮЧОВІ СЛОВА: ПЕНТЕСТИНГ, ТЕСТУВАННЯ, ПРОНИКНЕННЯ, ВЗЛОМ, ТЕСТУВАЛЬНИК, КІБЕРБЕЗПЕКА, ЗАХИЩЕНІСТЬ, САЙТ, КІБЕРІНЦИДЕНТ, ВРАЗЛИВОСТІ, МОДЕЛЬ ЗАГРОЗ.

					КБ.КР.Б-125-23-ПЗ				
Змн.	Арк.	№ докум.	Підпис	Дата	Проект підсистеми сканування безпеки веб-додатків за допомогою програмного забезпечення Owasp Zap	Літ.	Арк.	Аркушів	
Розроб.		Ковбан Н. С.		20.06.23					
Керівник		Байлюк Є.М.		20.06.23			4	70	
Реценз.		Покотило О.А.		19.06.23		ФІКТ Гр. КБ-19-2			
Н. контр.									
Затверд..		Сфіменко А.А.		19.06.23					

ANNOTATION

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

The aim of the work is to analyze and model cyber incidents that may indicate security loopholes in the website for further elimination.

Security is a primary important issue in website development, as vulnerabilities can lead to serious business losses. Developers can make mistakes during writing code or not realize the importance of using secure programming techniques, all of which can lead to vulnerabilities.

The project analyzed certain types of vulnerabilities, also rules, and policies that would help prevent these cyber incidents. The website was scanned for vulnerabilities and the behavior of the website under their influence was investigated.

The subject of the study is the technology of scanning website security using the Owasp Zap utility.

The object of research is the process of configuration and intended use of the Owasp Zap tool.

The explanatory note consists of an introduction, three chapters, conclusions, a list of abbreviations, and a list of references.

The theoretical part was presented on 62 pages of printed text. The list of used sources consists of 24 items and occupies 2 pages. The work contains 46 drawings. In total, the volume of the work is 70 pages.

KEYWORDS: PENTESTING, TESTING, PENETRATION, HACKING, TESTER, CYBERSECURITY, SECURITY, WEBSITE, CYBER INCIDENT, VULNERABILITIES, THREAT MODEL.

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		5

ЗМІСТ

АНОТАЦІЯ	Ошибка! Закладка не определена.
ANNOTATION.....	5
СПИСОК УМОВНИХ СКОРОЧЕНЬ	7
ВСТУП.....	8
РОЗДІЛ 1. ТЕОРЕТИЧНІ ТА ПРИКЛАДНІ АСПЕКТИ ТЕСТУВАННЯ БЕЗПЕКИ ВЕБ-САЙТІВ	10
1.1 Обґрунтування актуальності теми	10
1.2 Розгляд існуючих загроз інформаційній безпеці та їх типи.....	15
1.3 Рекомендації щодо вдосконалення стратегії безпеки веб-сайтів	19
РОЗДІЛ 2. ОГЛЯД ІСНУЮЧОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ТЕСТУВАННЯ ВЕБ-САЙТУ НА ВРАЗЛИВОСТІ ТА ПІДВИЩЕННЯ РІВНЯ БЕЗПЕКИ	23
2.1 Аналіз та порівняння інструментів для сканування веб-сайту щодо можливої загрози	23
2.2 Обґрунтування вибору інструменту для проведення тестування веб-сайту та огляд можливостей застосування	29
2.3 Варіації можливих атак в утиліті Owasp Zap	38
РОЗДІЛ 3. НАЛАШТУВАННЯ, ПЕРЕВІРКА ФУНКЦІОНАЛУ ТА ПРОВЕДЕННЯ ТЕСТУВАННЯ ВЕБ-САЙТУ ЗА ДОПОМОГОЮ ДОДАТКУ Owasp Zap.....	44
3.1 Встановлення та налаштування програмного забезпечення.....	44
3.2 Моделювання кіберінцидентів	49
3.3 Методи захисту від змодельованих атак та аналіз результатів тестування веб-сайту.....	59
ВИСНОВКИ.....	68
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	69

СПИСОК УМОВНИХ СКОРОЧЕНЬ

Owasp – Open Web Application Security Project

Zap – Zed Attack Proxy

DDos – Denial-of-service Attack

XSS – Cross Site Scripting

HTTPS – Hypertext Transfer Protocol Secure

SSL – Secure Sockets Layer

2FA – Two-Factor Authentication

WAF – Web Application Firewall

CLI – Command line interface

GUI – Graphical user interface

HUD – Heads Up Display

ІБ – інформаційна безпека

ПЗ – програмне забезпечення

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		7

ВСТУП

Актуальність теми. Інформаційні технології прогресивно розвиваються і захоплюють усе більше сфер нашої життєдіяльності. Сучасний інформаційний вік переніс майже всі звичайні підприємства на онлайн-платформи. Через це одним із найпопулярніших способів ведення бізнесу є створення веб-додатків. Основна причина такої популярності полягає в тому що Інтернет – це дешевий, простий і швидкий спосіб для спілкування та обміну інформацією. Але ця зручність існує разом з великою кількістю серйозних кіберзагроз. Як і інше програмне забезпечення, сайти в Інтернеті, схильні до різноманітних вразливостей, які дозволяють зловмисникам отримати доступ до конфіденційних і важливих даних або виконувати інші незаконні дії. Деякі вразливі місця небезпечні і виникають частіше, ніж хотілося б. Важливо розуміти, що таке вразливість, як вона відбувається, а також вчасно перевіряти ресурс, для того щоб ці вразливості виправити.

Відповідальними, в першу чергу, за безпеку веб-додатків є самі розробники. Зазвичай розробка веб-застосунків включає в себе команду розробників та деяку визначену методику для розробки програмного забезпечення, але не всі методики розробки веб-застосунків виносять на перший план питання безпеки. Тому є досить актуальним те, що розробка веб-застосунків потребує впровадженню безпеки уже з перших кроків розробки застосунку. Є безліч компаній готових платити фахівцям для дослідження їхніх продуктів з метою усунення вразливостей. Попри великий попит на подібних спеціалістів – кадрів недостатньо, а вартість послуг може бути досить значною, внаслідок чого лише невелика частина компаній здатні виділяти ресурси для підтримки безпеки програмного забезпечення у належному стані. Метою діяльності відділу з питань кібербезпеки та тестування на вразливості веб додатків є оцінка фізичної безпеки систем, серверів, мереж та веб додатків, для визначення області, що потребує фізичного захисту, створення нових методів тестування та сканування для

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		8

виявлення вразливостей, пошук слабких місць та документування результатів тестування інформаційної безпеки.

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

Предметом дослідження є дослідження процесу конфігурації та використання за призначенням інструменту Owasp Zар.

Основною метою кваліфікаційної роботи є пошук лазівок в безпеці веб-сайтів, дослідження поведінки веб-сайтів на кіберінциденти та їх моделювання.

Встановлена мета обумовлює наступні завдання:

- розгляд існуючих загроз інформаційній безпеці та їх типи;
- створення рекомендацій щодо встановлення стратегії безпеки веб-сайтів;
- вибір інструментів для сканування веб-сайтів щодо можливих загроз;
- дослідження програмного застосунку для сканування веб-сайту Owasp Zар;
- моделювання кіберінцидентів;
- пошук лазівок в безпеці додатку та дослідження поведінки сайту на певні види кібератак;
- аналіз результатів проведення тестування веб-сайту.

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		9

РОЗДІЛ 1. ТЕОРЕТИЧНІ ТА ПРИКЛАДНІ АСПЕКТИ ТЕСТУВАННЯ БЕЗПЕКИ ВЕБ-САЙТІВ

1.1 Обґрунтування актуальності теми

Рік за роком світ все дедалі віртуалізується, шляхом переходу з реального простору в кіберпростір. Кіберпростір визначається як віртуальний простір, де функціонують переважно мережеві цифрові активи. Його можна описати як віртуальну бібліотеку, де можна легко і швидко отримати доступ до великої кількості інформації за будь-якою темою. До того ж, за наявності відповідного інтернет-з'єднання це можливо в будь-який час. Кіберпростір має багато переваг для людства. Крім того, завдяки кіберпростору, спілкування стало набагато простішим і дешевшим. Можна просто зателефонувати комусь чи скористатися іншими послугами, такими як обмін повідомленнями та електронна пошта. Одного дня соціальні мережі зможуть повністю замінити фізичні мережі, оскільки все дедалі набирають значної популярності. Кіберпростір став величезним місцем необмежених розваг, де можна знайти пісні, які хочеться слухати, фільми, які хочеться дивитися, ігри, в які хочеться грати. Найголовніше, що кіберпростір дозволяє користувачам отримати можливості для роботи чи навчання, і що може бути дуже корисним. В освітньому секторі стає все більш популярним дистанційне навчання. Адже воно дозволяє налагодити безперервний процес самонавчання.

З кожним роком все дадалі більш зростає попит на використання дистанційних форм та на онлайн послуги. Ніша інтернет магазинів переповнює ринок, оскільки набагато зручніше вибрати та оцінити товар за допомогою онлайн сервісу, не виходячи з дому. Тому більшого успіху та перспектив має бізнес, котрий присутній на онлайн майданчиках. Особливо популярним дане дійство стало за часів карантину. Всього за декілька хвилин можна знайти необхідну інформацію чи оплатити товар. Також за допомогою онлайн сервісів зручніше вести облік даних, чи систематизувати велику кількість інформації. Онлайн черги

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

в лікарню, покупка чи продаж послуг та товарів, навчання, обмін даними – все виконується за допомогою мережі Інтернет.

Кіберпростір включає в себе такі компоненти:

- фізична інфраструктура та телекомунікаційні пристрої;
- комп'ютерні системи;
- підключення до Інтернету;
- мережа між пристроями;
- вузли доступу користувачів.

Кіберпростір іноді утотожнюють із визначенням Інтернет. Це досить хибне твердження, оскільки Інтернет є частиною кіберпростору. Це дві різні речі.

Кіберпростір є безмежним явищем, тому в епоху інформаційних технологій неможливо почуватись захищеним. Коли користувачі використовують Інтернет для онлайн-банкінгу, соціальних мереж або інших послуг, то існує великий ризик крадіжки особистих даних, наприклад номерів банківських карт, імені чи адреси користувача. Через незахищене з'єднання або встановлення зловмисного програмного забезпечення, зловмисники можуть отримати доступ до конфіденційних даних і використовувати їх для власної вигоди. Іноді, це призводить до серйозних проблем.

Для того щоб запобігти зловживанню кібертехнологіями, було розроблено кіберзакони. Загальна ідея кіберзакону полягає в тому, щоб ніхто не порушував права інших осіб у кіберпросторі. Будь-яке порушення кіберправ вважається порушенням кіберпростору та карається згідно до кіберзаконів. Окремий набір кіберзаконів розроблено урядом для забезпечення кібербезпеки для кіберкористувачів. Такі кіберзакони необхідні для контролю та запобігання будь-якій аморальній чи незаконній поведінці людей. Техніка захисту підключених до Інтернету систем, таких як комп'ютери, сервери, мобільні пристрої, електронні системи, мережі та дані від зловмисних атак, відома як кібербезпека.

Кібербезпеку можна розділити на п'ять різних типів:

- безпека критичної інфраструктури;

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

- безпека програми;
- безпека мережі;
- хмарна безпека;
- безпека Інтернету речей (IoT).

Для того щоб охопити всі свої основи, організаціям необхідно розробити комплексний план, який включає не лише ці п'ять видів кібербезпеки, але й три компоненти, які відіграють активну роль у забезпеченні кібербезпеки: люди, процеси та технології.

Кібербезпеку також можна розділити на дві окремі частини: перша – це «кібер» та «безпека». «Кібер» відноситься до технологій, які включають системи, мережі, програми та дані. А «безпека» це захист систем, мереж, програм та інформації. Також у деяких випадках її називають електронною інформаційною безпекою або безпекою інформаційних технологій.

Оскільки зберігання даних в онлайн-сервісах стає все дедалі поширеним явищем, то зростає і кількість злочинців, які намагаються викрасти ці дані та використати їх в своїх цілях. Кожного дня хакери викрадають кошти з особових рахунків, персональні дані, блокують діяльність та збирають безліч приватної та конфіденційної інформації. Багато сучасних сайтів містить велику кількість вразливостей та лазівок, про які навіть і не здогадується розробник чи користувач. Часто розробники не дотримуються правил безпеки, або взагалі забувають за них. Через це бізнес може постраждати або і взагалі буде зруйнованим. У зв'язку з цим росте попит на фахівців з кібербезпеки. Компанії зацікавлені в наймі на роботу спеціального фахівця, котрий буде контролювати заходи безпеки веб-сайту.

Тестувальники на проникнення, також відомі як пентестери, допомагають організаціям виявляти та усувати вразливі місця безпеки, що впливають на цифрові активи та ком'ютерні мережі компанії. Пентестери — це тестувальники на проникнення, які мають дозвіл на проникнення в систему. Галузі, що мають справу з конфіденційною, особистою, секретною або конфіденційною інформацією, як правило, наймають тестувальників на проникнення. Дані фахівці

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

допомагають оцінити, наскільки легко хакери можуть отримати доступ до активів компаній та даних системи. Вони визначають можливу кількість загроз, а також аналізують негативні наслідки для бізнесу від реальних атак. Під час симуляції атаки тестувальники документують свої дії для створення докладних звітів із зазначенням того, як їм вдалося обійти встановлені протоколи безпеки.

Щоб зменшити кількість вразливостей на сайті та знайти можливі варіанти витоку інформації, пентестери проводять сканування веб-сайту. Сканування можуть варіюватись від звичайного тестування мережі до моделювання серйозних кібератак.

Пентест, або тест на проникнення – це симуляція кібератаки на комп'ютерну систему з метою виявлення вразливостей. У контексті безпеки веб-додатків, тестування на проникнення зазвичай використовується для посилення брандмауера веб-додатків. Даний тест допомагає виявити наскільки система є вразливою до хакерських атак. До того ж, тести на проникнення дозволяють вжити попереджувальних заходів для мінімізації ризиків.

Необхідність та важливість тестування на проникнення в сучасному світі з такими просунутими загрозами, як відмова в обслуговуванні, крадіжка особистих даних, крадіжка послуг, втрата даних та витік даних є звичайним явищем. Проникнення в систему забезпечує протидію атакам шкідливих загроз за рахунок запобіжних методів. Деякі інші важливі переваги та потреба в тестуванні на проникнення полягають у виявленні вразливостей в системах та розгортанні безпеки таким же чином, як зловмисник отримує доступ:

- визначити загрози та вразливі місця для активів організації;
- щоб забезпечити комплексну оцінку політики, процедур, дизайну;
- щоб установити дії з виправлення, перш ніж вони будуть використані хакером для порушення безпеки;
- щоб визначити, до чого зловмисник може отримати доступ для крадіжки;
- визначити, яка інформація може бути крадіжкою та її використання;

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

- перевірити та підтвердити захист безпеки та визначити потребу в будь-якому додатковому рівні захисту;
- модифікація та оновлення поточної архітектури безпеки;
- щоб зменшити витрати на ІТ-безпеку за рахунок підвищення рентабельності інвестицій у безпеку.

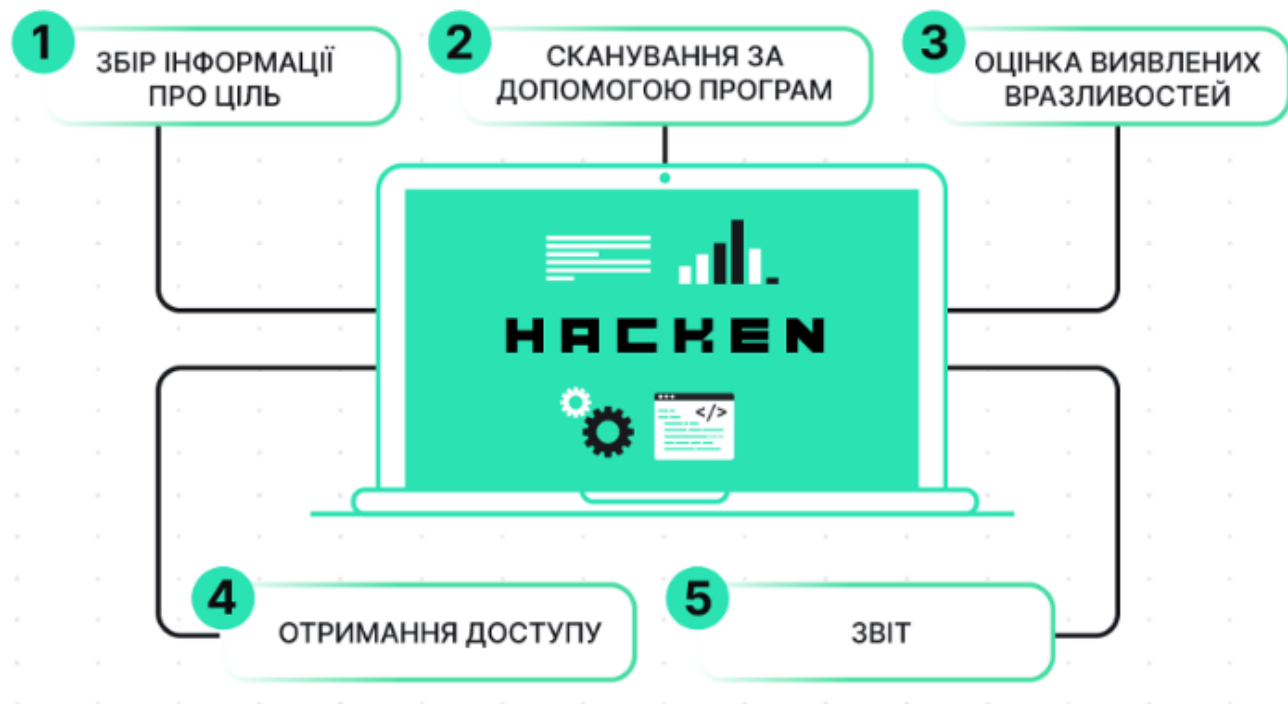


Рисунок 1.1 Фази пентесту

Збір інформації про ціль – покриття даних, які хакери можуть знайти у відкритому доступі. Наприклад, назва користувача, засоби масової інформації, якими вони користуються, відкриті порти та інформацію про працівників, пов'язаних з конкретними компаніями.

Сканування за допомогою програм – цей крок необхідний для визначення медіа з відкритими портами та використання їх послуг. Крім того, хакери завжди перевіряють ім'я та пароль користувача за замовчуванням.

Оцінка виявлених вразливостей – після збору даних проходить їх аналіз, який необхідний для формулювання подальших планів атак.

Отримання доступу – доступ до системи через виявлену раніше вразливість.

Звіт – звіт про створення всіх вразливих місць у клієнтській системі. На додаток до звіту, також надаються дані, що усувають вразливість.

Також важливо розрізняти типи тестування на проникнення. Існує всього три типи тестування: чорний, сірий та білий ящики.

Чорний ящик — це тип тестування на проникнення, у якому пентестер проводить сліпе тестування, тобто виконує його без попередніх знань про систему чи наявності будь-якої інформації про ціль. Чорний бокс призначений для демонстрації емульованої ситуації як нападник у протидії атаці.

Сірий ящик — це тип тестування на проникнення, у якому пентестер має дуже обмежені попередні знання про систему або будь-яку інформацію про цілі, таку як IP-адреси, операційна система або інформація про мережі. Даний тип призначений для демонстрації емульованої ситуації, оскільки інсайдер може мати цю інформацію, і для протидії атаці, оскільки пентестер має базову, обмежену інформацію щодо цілі.

Білий ящик — це тип тестування на проникнення, у якому пентестер має повні знання про систему та інформацію про ціль. Цей тип проникнення здійснюється групами внутрішньої безпеки або групами аудиту безпеки для проведення аудиту.

1.2 Розгляд існуючих загроз інформаційній безпеці та їх типи

Загроза кібербезпеці стосується будь-якої можливої зловмисної атаки, спрямованої на незаконний доступ до даних, порушення цифрових операцій або пошкодження інформації. Кіберзагрози можуть походити від різних суб'єктів, включаючи корпоративних шпигунів, хакерів, терористичні групи, ворожі національні держави, злочинні організації, хакерів-одинаків і незадоволених співробітників.

Кіберзловмисники можуть використовувати конфіденційні дані особи чи компанії, щоб викрасти інформацію або отримати доступ до їхніх фінансових рахунків, серед інших потенційно шкідливих дій, тому фахівці з кібербезпеки є важливими для захисту особистих даних.

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				15
Змн.	Арк.	№ докум.	Підпис	Дата		

Поширені способи отримати доступ до комп'ютера чи мережі:

- змінні носії, такі як флеш-накопичувачі;
- атака грубої сили методом спроб і помилок;
- веб-атаки або атаки електронною поштою;
- несанкціоноване використання системних привілеїв організації;
- втрата або крадіжка пристроїв, що містять конфіденційну інформацію.

Фахівці з кібербезпеки повинні мати глибоке розуміння наступних типів загроз кібербезпеці:

1. Зловмисне програмне забезпечення є однією з найпоширеніших загроз інформаційній безпеці. Це шкідливі засоби, наприклад шпигунська програма, програмне забезпечення-вимагач, віруси та черв'яки. Зловмисне програмне забезпечення активується, коли користувач натискає шкідливе посилання або вкладення, що призводить до встановлення небезпечених пакетів прикладних програм.

Можливості зловмисного програмного забезпечення після активації:

- блокування доступу до ключових мережевих компонентів (вимагачів);
- встановлення додаткових шкідливих програм;
- таємне отримання інформації шляхом передачі даних із жорсткого диска (шпигунське програмне забезпечення);
- виведення з ладу окремих частин системи, в результаті чого система буде непрацездатною.

Шкідливе програмне забезпечення розроблене для навмисного пошкодження комп'ютерів і мереж, використання особистої інформації жертв для отримання фінансової вигоди.

Зловмисне програмне забезпечення зазвичай поширюється електронною поштою у вигляді посилання або файлу для завантаження. Для того щоб його розповсюдити користувачеві потрібно буде натиснути посилання або відкрити файл. Наприклад, вірус прив'язує свій оманливий код до чистого коду та чекає, поки мимовільний користувач запустить його. Потім вірус поширюється,

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				16
Змн.	Арк.	№ докум.	Підпис	Дата		

завдаючи шкоди ключовим функціям, пошкоджуючи дані та блокуючи користувачам доступ до їхніх пристроїв.

2. Іншим поширеним методом, який використовують хакери, є фішинг, коли з користувачами зв'язується хтось, який видає себе за законну компанію, щоб спонукати їх передати конфіденційну та особисту інформацію. Коли ця інформація надана, кіберзлочинці можуть використовувати її для доступу до важливих облікових записів, що призведе до крадіжки особистих даних і фінансових втрат.

Фішингові атаки використовують фальшиві повідомлення, як-от електронні листи, щоб обманом змусити одержувача відкрити його та виконати вказівки в ньому, як-от надання номера кредитної картки. Мета полягає в тому, щоб викрасти конфіденційні дані, такі як дані кредитної картки та дані для входу, або встановити зловмисне програмне забезпечення на машину жертви.

Успіх фішингових атак залежить від людської помилки. Тому найкращий спосіб для організацій захистити себе від фішингових атак – переконатися, що всі співробітники мають чітке розуміння загрози та основних ознак, на які слід звернути увагу.

3. Розподілені атаки на відмову в обслуговуванні (DDoS).

DDoS-атаки роблять онлайн-сервіс недоступним, перевантажуючи його надмірним трафіком із багатьох місць і джерел. Час відповіді веб-сайту сповільнюється, запобігаючи доступу під час DDoS-атаки. Кіберзлочинці створюють великі мережі заражених комп'ютерів, які називаються ботнетами, розміщуючи зловмисне програмне забезпечення. DDoS-атака може не бути основним кіберзлочином. Атаки часто відволікають увагу, тоді як здійснюються інші види шахрайства та кібервиторгнення.

Варто зауважити, що DDoS та Dos атаки це відрізняються одна від одної. Атака на відмову в обслуговуванні (DoS) переповнює сервер трафіком, роблячи веб-сайт або ресурс недоступним. В той час коли розподілена атака на відмову в

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

обслуговуванні (DDoS) – це атака DoS, яка використовує кілька комп'ютерів або машин для вимикання мережі.

4. SQL-ін'єкція (SQLi) – це кібератака, яка впроваджує зловмисний код SQL у програму, дозволяючи зловмиснику переглядати або змінювати базу даних. Це може включати дані, що належать іншим користувачам, або будь-які інші дані, до яких має доступ сама програма. У багатьох випадках зловмисник може змінити або видалити ці дані, викликаючи постійні зміни у вмісті або поведінці програми.

У деяких ситуаціях зловмисник може посилити атаку SQL-ін'єкції, щоб скомпрометувати базовий сервер чи іншу серверну інфраструктуру, або здійснити атаку типу «відмова в обслуговуванні».

5. Недоліки функції аутентифікації. Функціональність програми, яка взаємодіє з керуванням автентифікації, часто реалізована неправильно, що може дозволити зловмиснику зламати паролі, ключі або токени сеансу. Помилки також можна використовувати для присвоєння (тимчасових або постійних) параметрів облікового запису користувача.

Це відбувається в результаті поганого кодування або дизайну, що впливає на автентифікацію та авторизацію доступу та функціональність програми.

Помилка логіки програми може бути спричинена зловживанням функціональністю, слабкими заходами безпеки або пропущеним кроком у процедурі перевірки. Ця вразливість дозволяє кіберзловмисникам легко обійти автентифікацію та отримати незаконний доступ до таких облікових записів.

6. Міжсайтові сценарії (XSS) – це тип ін'єкцій, під час яких шкідливі сценарії впроваджуються на безпечні та надійні веб-сайти. XSS-атаки відбуваються, коли зловмисник використовує веб-програму для надсилання шкідливого коду, як правило, у формі сценарію сторони браузера, іншому кінцевому користувачеві. Недоліки, які дозволяють цим атакам бути успішними, досить широко поширені та виникають у будь-якому місці, де веб-додаток використовує вхідні дані від користувача в межах вихідних даних, які він генерує, без їх перевірки чи кодування.

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

Зловмисник може використовувати XSS, щоб надіслати шкідливий сценарій нічого не підозрюючому користувачеві. Браузер кінцевого користувача не може дізнатися, що сценарію не можна довіряти, і виконає сценарій. Оскільки він вважає, що сценарій надійшов із надійного джерела Зловмисний сценарій може отримати доступ до будь-яких файлів cookie, маркерів сеансу чи іншої конфіденційної інформації, яка зберігається веб-переглядачем і використовується на цьому сайті. Ці сценарії можуть навіть переписати вміст сторінки HTML.

7. Брутфорс (атака грубої сили) може проявлятися багатьма різними способами, але головним чином полягає в тому, що зловмисник налаштовує заздалегідь визначені значення, робить запити до сервера, використовуючи ці значення, а потім аналізує відповідь. Заради ефективності зловмисник може використовувати словникову атаку (з мутаціями або без них) або традиційну атаку грубої сили (із заданими класами символів, наприклад: буквенно-цифрові, спеціальні, (не)чутливі до регістру). Враховуючи даний метод, кількість спроб, ефективність системи, яка проводить атаку, і оцінену ефективність системи, яка атакується, зловмисник може приблизно розрахувати, скільки часу знадобиться, щоб подати всі вибрані заздалегідь визначені значення.

Даний перелік кіберзагроз можна нескінченно продовжувати, оскільки він постійно оновлюється і хакери намагаються будь-якими способами досягнути своїх незаконних цілей. Тим самим реалізують нові атаки та удосконалюють вже наявні. Підсумовуючи, можна дійти висновку, що чим більш глибокі розуміння кіберзагроз має фахівець відповідальний за кібербезпеку, тим більше шансів пом'якшити ці процеси чи навіть і зовсім не допустити.

1.3 Рекомендації щодо вдосконалення стратегії безпеки веб-сайтів

Безпека веб-сайту є одночасно простим і складним процесом. Є принаймні десять важливих кроків, які можна зробити, щоб покращити безпеку веб-сайту, поки не стане надто пізно. Навіть у світі онлайн, власники повинні зберігати інформацію про клієнтів у безпеці та вживати всіх необхідних заходів обережності.

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

Найголовнішим аспектом безпеки – це своєчасне оновлення програмного забезпечення. Важливо постійно оновлювати всі встановлені платформи чи скрипти. Хакери агресивно атакують недоліки безпеки в популярному веб-програмному забезпеченні, і програми потрібно оновлювати, щоб виправити дірки в безпеці. Підтримуючи все в актуальному стані, це допоможе у запобіганню використання вразливостей.

Політика надійних паролів є першим кроком до захисту веб-сайту від зловмисників. Вимоги до надійних і унікальних паролів можуть значно ускладнити зловмисникам доступ до вашого сайту. Обов'язково необхідно переконатися, що серверна частина веб-сайту добре захищена і що доступ мають лише авторизовані користувачі. Для цього слід розглянути можливість використання менеджера паролів для створення та зберігання надійних паролів для сайту.

Двофакторна автентифікація (2FA) є важливим заходом безпеки, який слід застосувати для веб-сайту. 2FA додає додатковий рівень безпеки, вимагаючи від користувачів надати дві частини інформації, перш ніж вони зможуть отримати доступ до сайту. Це може включати пароль і одноразовий код, який генерує програма на пристрої. 2FA може допомогти запобігти зловмисникам отримати доступ до сайту, навіть якщо вони знають пароль користувача.

Також на веб-сайті варто використовувати HTTPS та SSL-сертифікати. URL-адреса повинна бути безпечною, щоб захистити веб-сайт. Якщо відвідувачі сайту повинні надіслати свою особисту інформацію, потрібен HTTPS, а не HTTP, щоб її доставити.

HTTPS (Hypertext Transfer Protocol Secure) – це протокол, який використовується для забезпечення безпеки в Інтернеті. HTTPS запобігає перехопленню, коли передається вміст. За допомогою HTTPS трафік шифрується таким чином, що навіть якщо пакети прослуховуються або іншим чином перехоплюються, вони виглядатимуть як безглузді символи. HTTPS використовує

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

протокол шифрування для шифрування зв'язку. Протокол називається Transport Layer Security (TLS), хоча раніше він був відомий як Secure Sockets Layer (SSL).

SSL (Secure Sockets Layer) – необхідний сайту протокол, що передає особисту інформацію відвідувача між веб-сайтом і базою даних. SSL шифрує інформацію, щоб запобігти її читанню іншими під час передачі. Він також позбавляє доступу до даних тим, хто не має відповідних повноважень. TLS є оновленою версією протоколу SSL, яка містить виправлення вразливостей що містились в SSL.

Цей протокол забезпечує захист зв'язку за допомогою так званої інфраструктури асиметричного відкритого ключа. Цей тип системи безпеки використовує два різні ключі для шифрування зв'язку між двома сторонами:

Приватний ключ – цей ключ контролюється власником веб-сайту, і він зберігається в закритому вигляді. Цей ключ знаходиться на веб-сервері та використовується для розшифрування інформації, зашифрованої відкритим ключем. Відкритий ключ – цей ключ доступний кожному, хто хоче взаємодіяти з сервером у безпечний спосіб. Інформацію, зашифровану відкритим ключем, можна розшифрувати лише закритим ключем.

Брандмауер веб-додатків (WAF) – це частина програмного забезпечення, яка знаходиться між веб-сайтом та Інтернетом. Він фільтрує трафік до сайту та блокує будь-які запити, які він вважає зловмисними. WAF можуть бути дуже ефективними для припинення атак, таких як впровадження SQL і міжсайтовий сценарій (XSS).

Реєстрація та моніторинг безпеки є критично важливим заходом безпеки для будь-якого веб-сайту. Реєструючи всю активність на сайті, це допоможе відстежити будь-яку зловмисну активність і вжити відповідних заходів. Також важливо регулярно перевіряти журнали, щоб виявити будь-яку незвичну діяльність.

Спам є серйозною проблемою для багатьох веб-сайтів. Він може засмітити розділи коментарів, контактні форми та навіть базу даних веб-сайту. Існує кілька

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				21
Змн.	Арк.	№ докум.	Підпис	Дата		

способів боротьби зі спамом, наприклад використання кодів CAPTCHA та вимога реєстрації для коментарів.

Головним моментом безпеки є сканування веб-сайтів на наявність вразливостей, щоб своєчасно їм запобігти. Важливо регулярно виконувати сканування веб-безпеки, щоб перевірити веб-сайт і сервер на вразливі місця. Сканування веб-безпеки слід виконувати за розкладом і після будь-яких змін або доповнень до веб-компонентів. Залучення професіонала для сканування безпеки на веб-сайті забезпечить поглиблений огляд і пояснення вразливостей на веб-сайті.

Існує ряд заходів безпеки, яких повинен вжити кожен веб-сайт. Впроваджуючи ці заходи, це допоможе захистити сайт від атак і захистити дані. Крім того, важливо навчати співробітників питанням безпеки веб-сайтів і мати чітку політику боротьби з порушеннями безпеки.

Висновки до першого розділу

В даному розділі було проаналізовано актуальність використання кібертехнологій у сучасному світі та досліджено головні поняття кібертехнологій. Було розглянуто хто такий спеціаліст з кібербезпеки та чому компанії з розробки програмного забезпечення зацікавлені в наймі такого працівника. Також досліджено існуючі загрози інформаційній безпеці і наведено рекомендації щодо поліпшення безпеки веб-сайту та умови для захисту даних.

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 2. ОГЛЯД ІСНУЮЧОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ДЛЯ ТЕСТУВАННЯ ВЕБ-САЙТУ НА ВРАЗЛИВОСТІ ТА ПІДВИЩЕННЯ РІВНЯ БЕЗПЕКИ

2.1 Аналіз та порівняння інструментів для сканування веб-сайту щодо можливої загрози

Інструменти пентестингу використовуються для автоматизації певних завдань, підвищення ефективності тестування та виявлення проблем, які важко виявити за допомогою лише ручних методів аналізу. Після оцінки загроз і вразливостей тестувальники на проникнення надають звіт, який може допомогти організаціям усунути виявлені ризики для покращення кіберзахисту.

Інструменти пентестингу мають вирішальне значення для тестування безпеки в сучасних широкомасштабних ІТ-середовищах. Вони підтримують виявлення активів у складних гібридних середовищах і можуть допомогти тестувальникам оцінити системи за контрольними показниками безпеки та вимогам відповідності. Хоча жоден інструмент не може замінити винахідливість талановитого тестувальника, інструменти можуть розширити та поглибити сферу тестування на проникнення, допомагаючи йому досягти кращих результатів.

Різновид наборів інструментів може варіюватись. Основні категорії інструментів:

1. Сканери портів: виявляють відкриті порти в системі. Це допомагає тестувальникам ідентифікувати операційну систему та програми, які зараз працюють у мережі, до якої вони намагаються отримати доступ. Сканери портів використовуються в розвідці та можуть надавати інформацію про потенційні вектори атак.

2. Сканери вразливостей: шукають відомі вразливості в серверах, операційних системах і програмах, а також неправильні конфігурації, якими можна скористатися під час тестування. Звіти, надані сканерами вразливостей,

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				23
Змн.	Арк.	№ докум.	Підпис	Дата		

допомагають тестувальникам на проникнення вибрати вразливість, яку можна використовувати для доступу в систему.

3. Мережевий аналізатор – відстежує інформацію про мережевий трафік, зокрема його джерело, пункт призначення, пристрої, що взаємодіють у мережі, протоколи та використання порти. Дані утиліти можуть бути корисними для перевірки та визначення чи дані зашифровані, і шляхів зв'язку, які можна використовувати під час тесту на проникнення.

4. Веб-проксі: дозволяє тестувальникам перехоплювати та змінювати трафік між браузером і веб-серверами організації. Це дозволяє виявляти приховані поля форм та інші функції HTML, які можуть сприяти атакам, таким як міжсайтовий сценарій (XSS) або підrobка міжсайтового запиту (CSRF).

5. Зломщик паролів – служить для злому паролів операційних систем. Зломщики паролів дозволяють тестувальникам на проникнення визначити, чи використовують співробітники організації слабкі паролі, які становлять ризик злому.

Існує безліч інструментів для пентестингу, а саме для прискорення робочих процесів. Кожен із цих інструментів використовує автоматизацію для виявлення нових форм атак.

Командам часто потрібні різноманітні інструменти для виконання повного тесту на проникнення, тому використання широкого спектру інструментів пентестування з відкритим кодом допомагає знизити витрати. І багато пентестерів уже знайомі з такими відомими інструментами, як Nmap, Burp Suite, Owasp Zarp, Nikto чи Wireshark.

Таблиця 2.1

Перелік інструментів для пентестингу

Назва програми	Можливості	Ключові особливості
NMAP	Сканування мережі	Створення інвентаризації мережевих пристроїв, сканування портів

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				24
Змн.	Арк.	№ докум.	Підпис	Дата		

Продовження таблиці 2.1

Перелік інструментів для пентестингу

Назва програми	Можливості	Ключові особливості
Wireshark	Аналіз протоколів, моніторинг мережі	Аналіз мережевого трафіку, відображення даних в режимі реального часу, перевірка протоколів
Nikto	Сканер вразливостей	Сканує безліч небезпечних програм і порталів
Burp Suite	Тестування безпеки веб-додатків	Веб-сканер, проксі. Містить широкий вибір інструментів

Nmap, скорочення від Network Mapper, – це безкоштовний інструмент із відкритим вихідним кодом, який використовується для перевірки вразливостей, сканування портів і картографування мережі. Nmap дозволяє мережевим адміністраторам визначати, які пристрої працюють у їхній мережі, а також виявляти відкриті порти та служби.

По-перше, Nmap допоможе швидко скласти карту мережі без складних команд або конфігурацій. Він також підтримує прості команди (наприклад, щоб перевірити, чи хост працює) і складні сценарії за допомогою механізму сценаріїв Nmap. Nmap включає в себе наступні функції:

- можливість швидкого розпізнавання всіх пристроїв, включаючи сервери, маршрутизатори, комутатори, мобільні пристрої тощо (в одній або кількох мережах);
- допомагає визначити служби, запущені в системі, включаючи веб-сервери, DNS-сервери та інші поширені програми. Nmap також може виявляти версії програми з високою точністю, щоб допомогти виявити наявні вразливості;

- Nmap може знайти інформацію про операційну систему, що працює на пристроях;

- під час аудиту безпеки та сканування вразливостей можна використовувати Nmap для атаки на системи за допомогою наявних сценаріїв із механізму створення сценаріїв Nmap.

Дана програма підтримує як CLI, так і GUI інтерфейс. Nmap має графічний інтерфейс користувача під назвою Zenmap. Дана утиліта допомагає розробити візуальне відображення мережі для кращого використання та звітності.

Wireshark – це рішення для моніторингу мережі, яке фіксує та аналізує мережевий трафік через різноманітні канали зв'язку. Тестувальники на проникнення можуть автоматично зчитувати дані в реальному часі з різних типів мереж. Також дана утиліта дозволяє досліджувати проблеми безпеки в мережі, ідентифікувати елементи мережі, які не працюють і можуть бути використані в атаці, а також виявляти помилки реалізації протоколу або конфігурації.

Додаткові функції включають:

- шифрування даних;
- моніторинг і оповіщення сервера;
- імпорт/експорт даних.

Wireshark в основному використовується для захоплення пакетів даних, що переміщуються через мережу. Інструмент дозволяє користувачам переводити контролери мережевого інтерфейсу у безладний режим, щоб спостерігати за більшістю трафіку, навіть одноадресного, який не надсилається на MAC-адресу контролера. На більш високому рівні Wireshark можна використовувати для пошуку та виділення дуже специфічних пакетів, наприклад тих, що відповідають відомому шаблону атаки. Це робить його корисним інструментом для полювання на загрози, оскільки окремі пакети виділяються червоним кольором (або будь-яким іншим кольором, який забажає користувач), щоб попередити дослідників про їхню присутність у мережі.

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				26
Змн.	Арк.	№ докум.	Підпис	Дата		

Nikto – це програмне забезпечення з відкритим вихідним кодом, написане мовою Perl, яке використовується для сканування веб-сервера на наявність вразливості, якою можна скористатися та скомпрометувати сервер. Він також може перевіряти застарілу інформацію про версію сервера та виявляти проблеми з конкретними відомостями про версію.

Проект сканера вразливостей Nikto – це швидка робота, яка часто оновлюється останніми відомими вразливими місцями. Це дає змогу впевнено сканувати веб-сервери під час пошуку будь-яких можливих проблем.

Основні переваги програмного забезпечення:

- Nikto є безкоштовним у використанні, має відкритий код і часто оновлюється;
- можна використовувати для сканування будь-якого веб-сервера (Apache, Nginx, Lighttpd, Litespeed тощо);
- сканує проблеми, пов'язані з конфігурацією, наприклад відкриті каталоги індексів;
- сканування сертифіката SSL;
- можливість сканування кількох портів на сервері з кількома запущеними веб-серверами;
- можливість сканування через проксі та з http-автентифікацією;
- можливість вказати максимальний час сканування, виключити певні типи сканувань і незвичайні заголовки звітів.

Єдиним недоліком утиліти є відсутність графічного інтерфейсу, усі команди запускаються в командному рядку.

Burp Suite – один із популярних інструментів ручного тестування на проникнення, який є дуже корисним для етичних хакерів, пентестерів та інженерів безпеки.

Burp Proху дозволяє тестувальникам на проникнення проводити атаки типу "людина посередині" між веб-сервером і браузером. Вони дозволяють перевіряти

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				27
Змн.	Арк.	№ докум.	Підпис	Дата		

мережевий трафік, що може допомогти виявити і використати вразливості та витоки даних у веб-додатках.

Ключові особливості Burp Suite:

- використання спеціального клієнта для виконання ручного тестування на позасмугові вразливості;
- тестування та підтвердження атак клікджекінгу за допомогою спеціальних засобів;
- глибоке ручне тестування, що дає змогу побачити відображені або збережені вхідні дані для перевірки XSS і подібних уразливостей;
- записує результати атак і дозволяє тестувальникам налаштовувати їх під час наступних атак;
- забезпечує швидкий брутфорсинг і фазинг за допомогою спеціальних послідовностей HTTP-запитів, що містять кілька наборів корисного навантаження.

Клікджекінг, також відомий як «атака відшкодування інтерфейсу користувача», – це коли зловмисник використовує кілька прозорих або непрозорих шарів, щоб обманом змусити користувача натиснути кнопку чи посилання на іншій сторінці, коли він мав намір натиснути сторінку верхнього рівня. Таким чином, зловмисник «викрадає» кліки, призначені для його сторінки, і спрямовує їх на іншу сторінку, яка, швидше за все, належить іншій програмі, домену або обом.

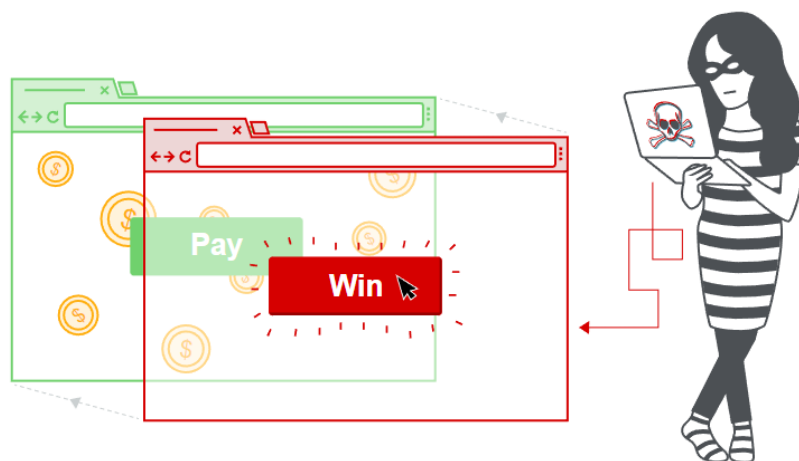


Рисунок 2.1 – Атака клікджекінгу

2.2 Обґрунтування вибору інструменту для проведення тестування веб-сайту та огляд можливостей застосунку

Спільнота Owasp створює низку програм, інструментів, навчальних посібників і стандартів, які сприяють загальному здоров'ю Інтернету та допомагають організаціям планувати, розробляти, підтримувати та керувати веб-програмами, яким можна довіряти. Усі проекти, інструменти, документи, розділи та форуми Owasp керуються спільнотою та мають відкритий код, вони дають можливість перевірити теорії чи ідеї та отримати професійну пораду та підтримку від спільноти Owasp.

Owasp Zap (Zed Attack Proxy) є частиною безкоштовної спільноти Owasp. Він ідеально підходить для розробників і тестувальників, які тільки починають знайомство з тестуванням на проникнення. Дана утиліта є кросплатформною, його можна використовувати для систем Linux, Microsoft і Mac.

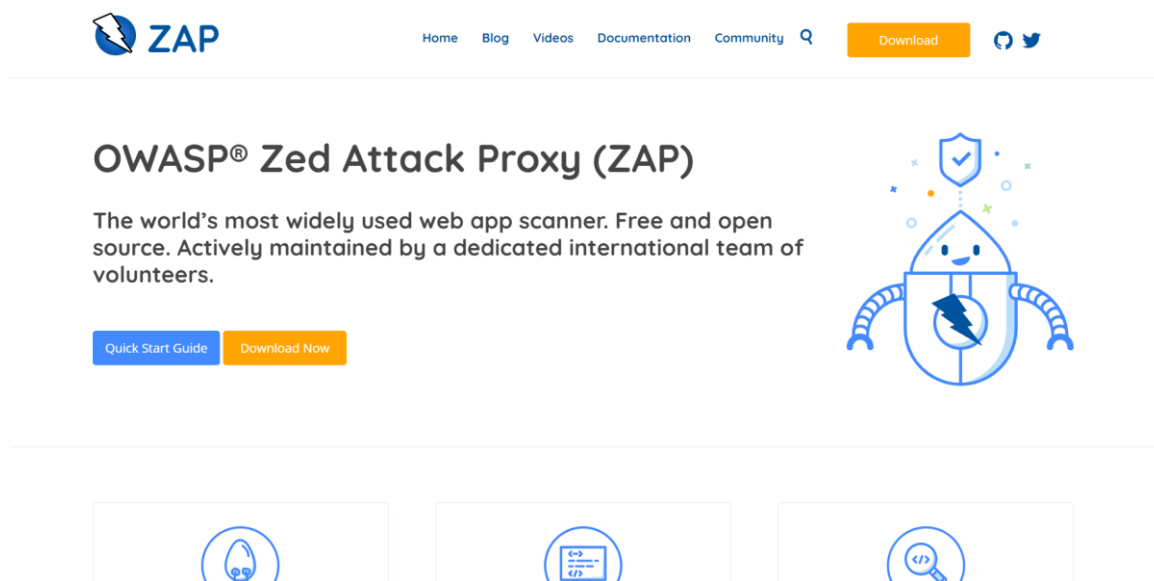


Рисунок 2.2 – Owasp Zap

Zap – це сканер безпеки веб-додатків із відкритим вихідним кодом, який може допомогти знайти та використовувати поширені веб-вразливості, такі як впровадження SQL, міжсайтовий сценарій, несправна автентифікація тощо. Він також може діяти як проксі, дозволяючи перехоплювати та змінювати трафік між браузером і веб-сервером. Доки користувач пересувається по сайту, він фіксує усі дії, а потім виконує атаку відомими методами. Owasp Zap можна використовувати

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				29
Змн.	Арк.	№ докум.	Підпис	Дата		

вручну або автоматизувати його за допомогою сценаріїв, API або плагінів. Він підтримує різні режими, такі як пасивне сканування, активне сканування, спайдеринг і фаззинг. З огляду на ризики, він є дуже потужним інструментом, через це його використання повинно бути обмежено середовищами, де тестувальник має згоду на виконання тесту та впевненість, що неможливо заподіяти постійну шкоду, оскільки існує певна форма захисту веб-програми.

Можна виділити найголовніші функції утиліти:

1. Контроль політики сканування – дозволяє користувачам створювати політики сканування відповідно до своїх вимог для кожної програми. Створюючи політику сканування, необхідно враховувати пропускну здатність сканера, можливості обробки та тип сканера. Політики можна зберігати як шаблони, що дозволить легко використовувати їх в майбутньому.

2. Сканування портів. Owasp Zarp може виконувати сканування портів, щоб ідентифікувати відкриті порти, вразливі до атак. Це тип тестування безпеки, який сканує мережеві порти системи, щоб визначити, відкриті вони чи закриті. Дана функція використовується для того, щоб визначити будь-які служби чи програми, запущені в системі, і допомогти виявити будь-які слабкі місця в них. Це корисний інструмент для тестувальників на проникнення та дослідників безпеки, оскільки він може надати детальну інформацію про конфігурацію системи та потенційні вектори атак.

3. Проксі дозволяє користувачеві перехоплювати запити та відповіді, зроблені браузером. Це дозволяє користувачеві спостерігати та змінювати ці запити та відповіді в режимі реального часу, перш ніж вони будуть надіслані або отримані. Перехоплення запитів/відповідей можна використовувати для налагодження додатків, виявлення вразливостей безпеки або виконання тестування вручну. Простіше кажучи, він діє як людина посередині між браузером користувача та цільовою веб-програмою, дозволяючи користувачам перевіряти, змінювати та відтворювати HTTP-запити.

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				30
Змн.	Арк.	№ докум.	Підпис	Дата		

4. Управління доступом допомагає обмежити доступ до певних ресурсів на сервері. Використовується для того, щоб заблокувати певні IP-адреси, обмежити доступ користувачів або вимагати автентифікації. Це гарантує, що лише авторизовані користувачі можуть отримати доступ до ресурсів на сервері та захищає від злоумисників.

5. Активне і пасивне сканування. Активне сканування – це оцінка вразливості, яка перевіряє мережу на наявність потенційних загроз безпеці. Воно передбачає надсилання запитів до цільової системи, аналіз відповідей і пошук вразливостей.

Пасивне сканування – це тип перевірки вразливості, яка жодним чином не взаємодіє з цільовою системою. Натомість – пасивно відстежує мережевий трафік на наявність потенційних загроз безпеці. Воно не надсилає жодних запитів чи перевірок, натомість шукає підозрілу активність, наприклад злоумисне корисне навантаження чи аномальну поведінку користувачів. Зар за замовчуванням пасивно сканує всі HTTP-повідомлення (запити та відповіді), надіслані веб-додатку, що тестується.

6. Zar Fuzzer. За допомогою цієї функції можна створити власні корисні навантаження для надсилання в перевірену програму, використовуючи вбудовані навантаження як відправну точку. Він має чотири режими: безпечний, захищений, стандартний і атакуючий. Кожен режим пропонує різні функції та можливості для тестування різних програм і векторів атак. Наприклад, режим атаки дозволяє більш гнучко тестувати, дозволяючи користувачам вказати тип програми та операційної системи, яку вони хотіли б атакувати.

7. Використання AJAX спайдерингу під час тесту на проникнення, щоб ідентифікувати запити у веб-програмі, яка підтримує AJAX. Дані запити неможливо ідентифікувати за допомогою стандартного інструменту павука. Спайдер використовує просту методологію, коли він переходить за всіма посиланнями, які може знайти через браузер, навіть за тими, які генеруються

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		31

кодом на стороні клієнта, що допомагає йому ефективно переглядати широкий спектр програм.

AJAX – це аббревіатура, що означає асинхронний Javascript і XML. Коли на сайті використовується технологія AJAX, немає необхідності щоразу оновлювати всю сторінку, оскільки оновлюється лише певна її частина.

Отже можна підбити підсумки, та навести перелік переваг і недоліків програмного забезпечення Owasp Zap.

Таблиця 2.2

Переваги та недоліки утиліти Owasp Zap

Переваги	Недоліки
Проект із відкритим вихідним кодом за підтримки учасників.	Застарілий інтерфейс користувача, який іноді може бути незграбним і може потребувати певних налаштувань, перш ніж стане зручним.
Широкий спектр методів тестування безпеки програми, які можуть допомогти виявити потенційні вразливості.	Можливості автоматичного сканування обмежені порівняно з іншими інструментами.
Функції звітування комплексні та настроювані.	Веб-версії немає. Необхідно завантажити його у свою систему, щоб використовувати його.
Безкоштовний як для особистого, так і для комерційного використання, що робить його доступним вибором для розробників з обмеженим бюджетом.	Документація груба і складна для розуміння.

Zap має чіткий і реактивний графічний інтерфейс (GUI), хоча і досить щільний: починаючи з верхньої частини вікна, знаходиться 4 різні області. За замовчуванням під час запуску Zap відображаються лише основні вкладки. Решта

вкладок відкриваються, коли вони використовуються (наприклад, для павука та активного сканера).

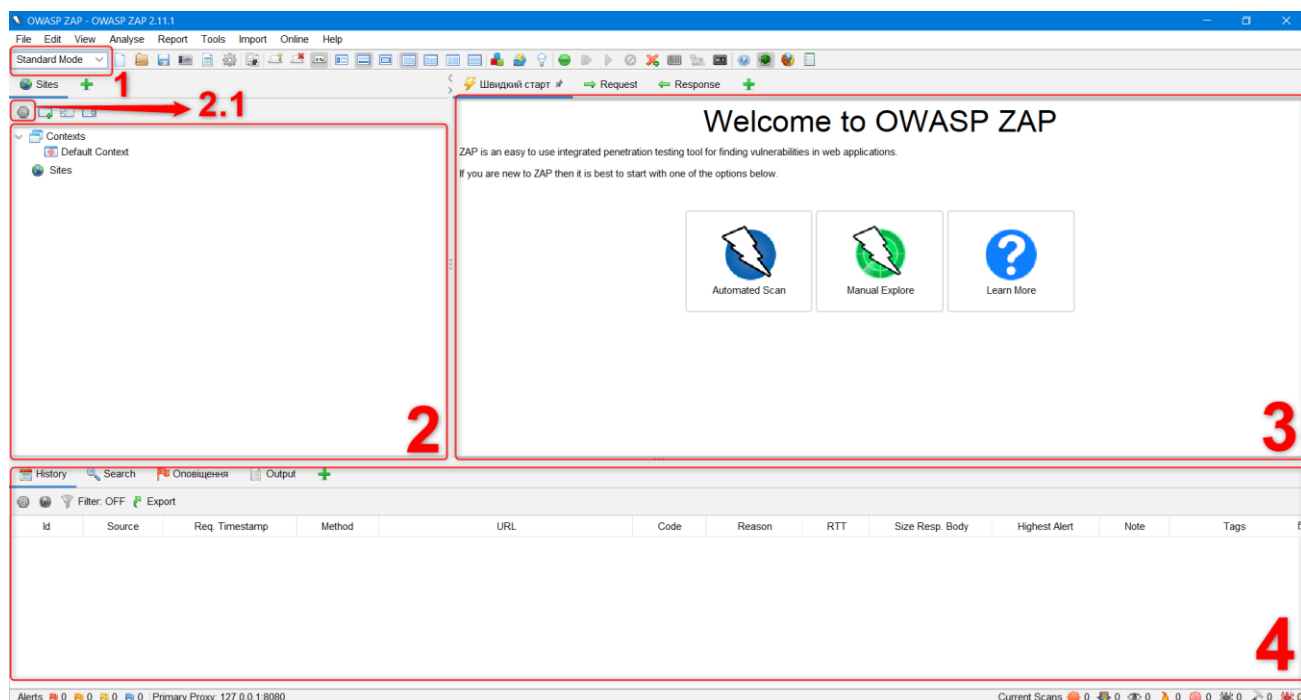


Рисунок 2.3 – Графічний інтерфейс Owasp Zap

1. Режими.

У верхньому лівому куті екрана всього 4 режими:

- стандартний режим – дозволяє робити все, що завгодно, з будь-яким веб-сайтом;
- режим атаки – активно стежить за будь-яким веб-сайтом. Нові вузли, які знаходяться в Score, активно скануються, як тільки їх буде виявлено;
- безпечний режим – вимикає всі шкідливі функції під час сканування. Тобто всі потенційно небезпечні операції заборонені;
- захищений режим: дозволяє сканувати веб-сайти в певній області. Це запобігає скануванню небажаних веб-сайтів.

Рекомендується використовувати захищений режим, щоб переконатися, що атака відбувається лише на необхідні сайти та не нанести випадково шкоди.

2. Сайти.

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		33

В даній секції будуть перераховані веб сайти, до яких буде надано доступ через Owasp Zap.

2.1 Показувати лише URL-адреси. Ця опція є досить корисною, адже після перших тестів цей розділ буде сильно засмічений. Щоб сфокусуватися на цільовому веб-сайті, необхідно поставити на нього прапор «In Score».

3. Вікно робочої області.

Дане вікно складається з наступних вкладок:

3.1 Automated scan – Zap почне сканування сайту та створить карту сайту, стан якої можна перевірити за допомогою вкладки «Павук», а потім перейде до фази атаки (вкладка «Активне сканування»). Zap продовжить сканувати веб-програму за допомогою свого павука та пасивно сканувати кожен знайдену сторінку. Потім використовуватиме активний сканер для атаки на всі виявлені сторінки, функції та параметри. Zap надає 2 павуки для сканування веб-додатків, які можна використовувати як одночасно так і по одному. Традиційний павук Zap виявляє посилання, досліджуючи HTML у відповідях веб-програми. Цей павук швидкий, але не завжди ефективний під час дослідження веб-програми AJAX, яка генерує посилання за допомогою JavaScript. Для додатків AJAX павук AJAX від Zap, швидше за все, буде ефективнішим. Цей павук досліджує веб-програму, викликаючи браузер, які потім переходять за згенерованими посиланнями. Павук AJAX повільніший, ніж традиційний павук.

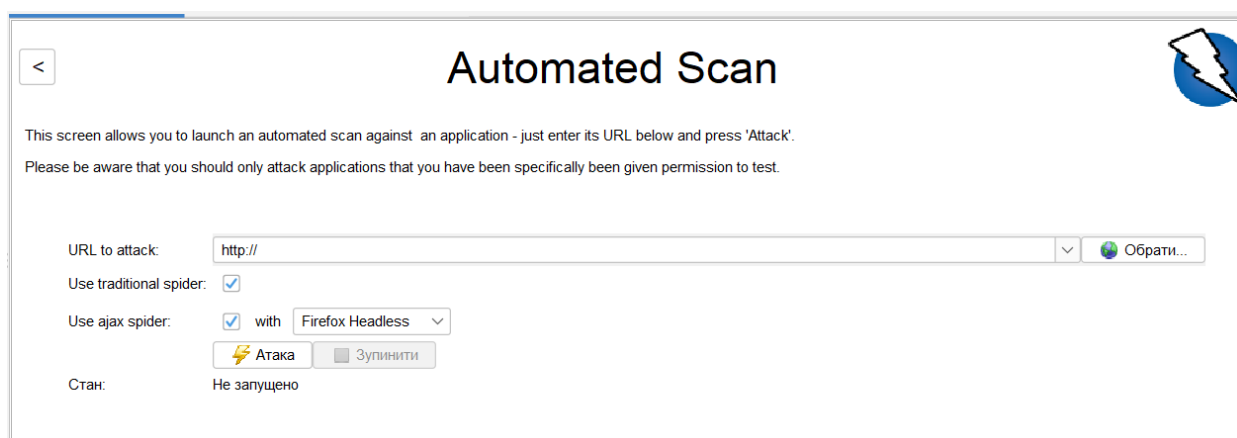


Рисунок 2.4 – Автоматичне сканування Owasp Zap

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				34
Змн.	Арк.	№ докум.	Підпис	Дата		

3.2 Manual Explore – дослідження програми вручну в Owasp Zap це процес ручного тестування програми на наявність вразливостей у безпеці. Це робиться для виявлення будь-яких потенційних ризиків безпеки, які можуть бути присутніми в програмі. Це допоможе забезпечити максимальну безпеку програми. Сканування вручну доповнює автоматичне сканування, забезпечуючи більш глибокий аналіз програми та дозволяючи керувати процесом пентесту. Автоматичне сканування може пропустити деякі вразливості, але сканування вручну може виявити пропущені проблеми. Однак дане сканування може зайняти багато часу та бути неможливим для великих програм.

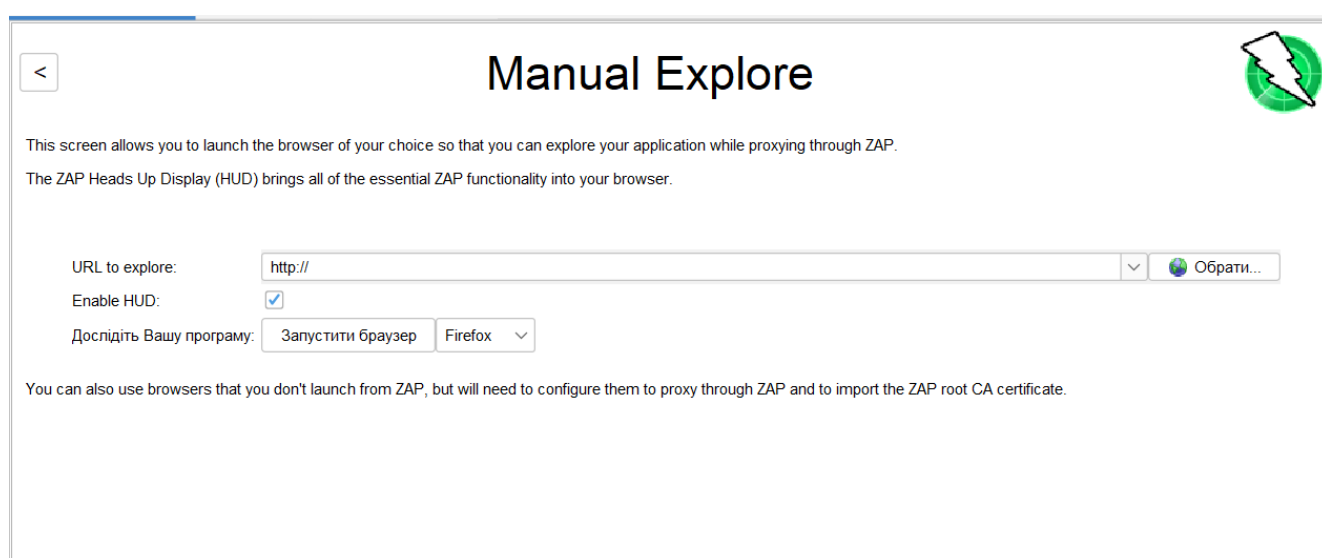


Рисунок 2.5 – Ручне тестування Owasp Zap

За замовчуванням при даному виді тестування використовується інтерфейс HUD. Якщо зняти позначку з відповідної опції перед запуском браузера, HUD буде вимкнено.

Heads Up Display (HUD) – це новий інноваційний інтерфейс, який забезпечує доступ до функцій Zap безпосередньо в браузері. Він ідеально підходить для тих, хто новачок у веб-безпеці, а також дозволяє досвідченим тестувальникам на проникнення зосередитися на функціональності додатків, надаючи при цьому ключову інформацію та функції безпеки. HUD накладається поверх цільової програми у браузері, однак дану опцію підтримують не усі браузери, лише Firefox та Chrome.

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		35

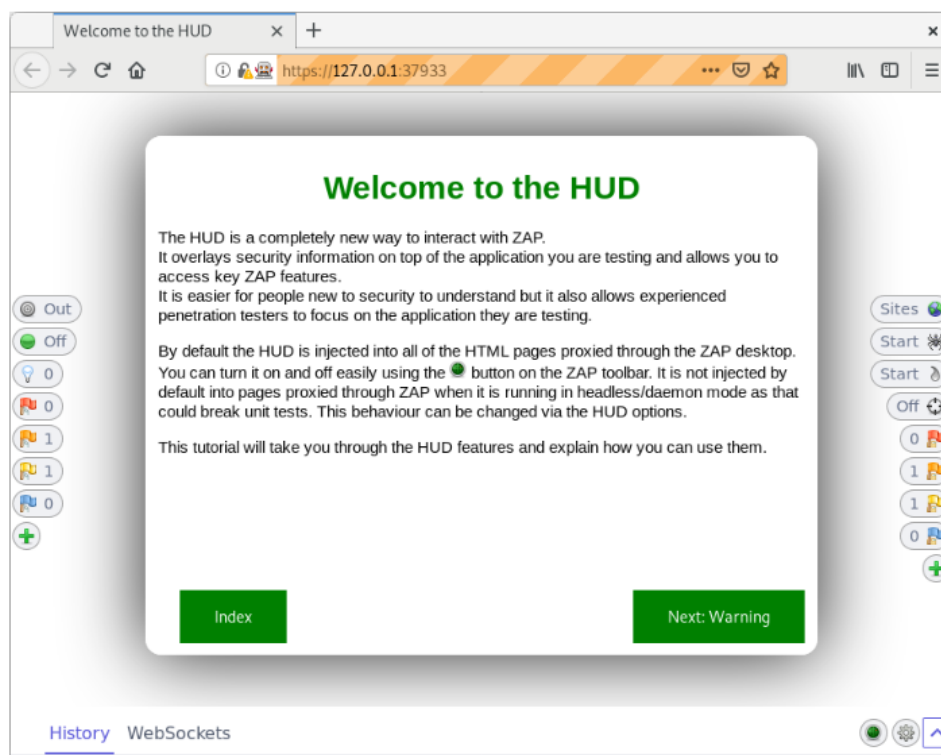


Рисунок 2.4 – Інтерфейс HUD для ручного тестування

4. Нижнє вікно відображає:

- результати тесту;
- історію запитів;
- вразливості сканування.

Також в нижньому вікні можна знайти 4 вкладки:

- history – показує список усіх запитів у порядку їх надсилання;
- search – дозволяє здійснювати пошук за регулярними виразами в URL-адресах, запитах, відповідях, заголовках;
- alerts – відображає вразливості, знайдені на перевіреному веб-сайті. Кожній знайденій вразливості присвоюється рівень критичності;
- output – виводить різні інформаційні повідомлення. Наприклад, трасування стека, помилки чи іншу інформацію про налагодження.

Найкорисніша вкладка «Alerts\Оповіщення» відображає вразливості, знайдені на веб сайті. Для того щоб вивести на екран детальну інформацію про вразливість, необхідно натиснути на одне із попереджень в даній вкладці.

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				36
Змн.	Арк.	№ докум.	Підпис	Дата		

У лівій частині нижнього вікна міститься кількість сповіщень, знайдених під час тестування, розділених на категорії ризику. Існує декілька категорій ризику:

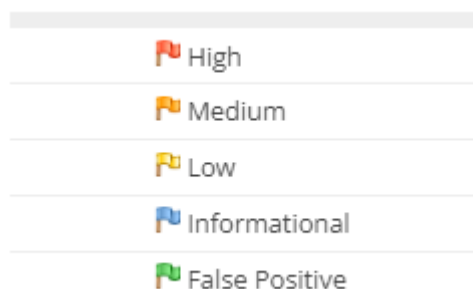


Рисунок 2.5 Категорії ризику сповіщень

High – найвищий (критичний) рівень вразливостей. Ін'єкції SQL, атаки типу path traversal, ін'єкції LDAP, Spring4Shell та інші. Він позначений червоним прапорцем.

Medium – середній рівень критичності знайдених загроз. До нього відносять Directory Browsing, HTTPOnly, XSLT-ін'єкції та інші. Він позначений помаранчевим прапорцем.

Low – низький рівень вразливостей. Тут знаходиться інформація, зв'язана з відладкою та розкриттям приватних IP-адрес. Позначено жовтим прапорцем.

Informational – рівень, який повідомляє про інформацію, що знаходиться в мережі Інтернет, а також на сервері. До нього відносяться версії веб-сервера, СУБД, бібліотек, фреймворків, а також версії мов програмування та операційних систем. Як правило, уразливості, що відносяться до цієї категорії, не можуть безпосередньо завдати шкоди системі, системі збору інформації сайту та її компонентам. Позначено синім прапорцем.

False Positive – хибні тривоги. У рідкісних випадках Zар може виявити вразливість, яка не є вразливістю. Якщо є сумніви, то програма позначає цю «вразливість» зеленим прапорцем. Однак користувач повинен перевірити чи проблема є загрозою сайту.

Отже, Owasr Zар є інструментом для сканування веб сайтів, котрий легко зрозуміти і через це користується попитом у новачків цієї справи. Але він також

вимагає певних навичок щодо режимів атаки та аналізу результатів, однак документація, яку надає OWASP, полегшує навчання.

Таким чином, Zap є не тільки інструментом для професіоналів, але також є безцінною підтримкою для вивчення нової галузі в сфері безпеки, яка розвивається.

2.3 Варіації можливих атак в утиліті Owasp Zap

Будь-який вид зловмисної діяльності, яка намагається зібрати, порушити, заборонити, погіршити або знищити ресурси інформаційної системи називається атакою. За допомогою утиліти Owasp Zap можна змодельовати певні види атак чи технік тестування програмного забезпечення. Наприклад, атаку підбору даних для аутентифікації – брут форс, чи техніку фазинг тестування.

Атака грубою силою – це метод злому, який використовує метод спроб і помилок для злому паролів, облікових даних і ключів шифрування. Це проста, але надійна тактика для отримання несанкціонованого доступу до індивідуальних облікових записів, систем і мереж організацій. Хакер пробує кілька імен користувачів і паролів, часто використовуючи комп'ютер для тестування широкого діапазону комбінацій, доки не знайде правильну інформацію для входу.

Назва «груба сила» походить від того, що зловмисники використовують надмірну силу для отримання доступу до облікових записів користувачів. Незважаючи на те, що це застарілий метод кібератак, атаки грубою силою перевірені та залишаються популярною тактикою серед хакерів. При проектуванні системи аналізують загрози з точки зору зловмисника. Злом грубою силою вимагає багато терпіння, тому що зловмиснику можуть знадобитися місяці або навіть роки, щоб успішно зламати пароль або ключ шифрування. Проте потенційні винагороди величезні.

Зловмисники грубої сили повинні докласти трохи зусиль, щоб ці схеми окупилися.

Ось як хакери отримують користь від атак грубою силою:

- отримання прибутку від реклами або збору даних про діяльність;

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				38
Змн.	Арк.	№ докум.	Підпис	Дата		

- викрадення особистих даних і цінностей;
- розповсюдження зловмисного програмного забезпечення для збоїв;
- викрадення системи для зловмисної діяльності;
- псування репутації сайту.

Сьогодні люди мають багато облікових записів і паролів. Люди, як правило, неодноразово використовують кілька простих паролів, що робить їх підданими атакам грубої сили. Крім того, повторне використання того самого пароля може надати зловмисникам доступ до кількох облікових записів.

Облікові записи електронної пошти, захищені слабкими паролями, можна підключати до додаткових облікових записів, а також використовувати для відновлення паролів. Це робить їх особливо цінними для хакерів. Крім того, якщо користувачі не змінюють пароль маршрутизатора за умовчанням, їхня локальна мережа стає вразливою до атак. Зловмисники можуть спробувати кілька простих паролів за замовчуванням і отримати доступ до всієї мережі.

Серед найпоширеніших паролів у списках грубої сили: дата народження, імена, Qwerty, 123456, ABCDef123, ABCDEF123, ABCDEF123, ABC123, пароль, Asdf, , ZXCVCBN, QAzws x, 654321, 123321, 000000, 111111, 987654321, 1q2w3e, 123qwe, qwertyuiop, gfhjkm.

Надійні паролі забезпечують кращий захист від крадіжки особистих даних, втрати даних, несанкціонованого доступу до облікових записів тощо.

Щоб захистити організацію від грубого злону паролів, необхідно використовувати надійні паролі. Паролі повинні:

- ніколи не використовувати інформацію, яку можна знайти в Інтернеті (наприклад, імена членів сім'ї);
- не використовувати короткі паролі;
- комбінувати літери, цифри та символи;
- використовувати різні логіни та паролі для кожного акаунту;
- уникати загальних шаблонів.

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		39

З точки зору адміністратора, в організації необхідно застосувати методи захисту користувачів від грубого злому паролів:

- політика блокування – заблокувати облікові записи після кількох невдалих спроб входу, а потім розблокувати їх під правами адміністратора;
- поступові затримки – заблокувати облікові записи на обмежений проміжок часу після невдалих спроб входу. Кожна спроба збільшує затримку;
- CAPTCHA – такі інструменти, як reCAPTCHA, вимагають від користувачів виконання простих завдань для входу в систему. Користувачі можуть легко виконати ці завдання, тоді як інструменти грубої сили не можуть;
- вимагання надійних паролів – застосовувати правила для користувачів, які полягають у створенні довгих та складних паролей. Також важливо встановити періодичність зміни паролей;
- двофакторна автентифікація – використовувати кілька факторів для автентифікації особи та надання доступу до облікових записів.

Фаззинг або фазз-тестування визначається як автоматичний метод тестування програмного забезпечення, який використовує широкий діапазон недійсних і неочікуваних даних як вхідні дані для пошуку недоліків у програмному забезпеченні, яке проходить тестування.

Недоліки не обов'язково мають бути вразливими місцями безпеки. Фаззинг також може виявити іншу небажану або неочікувану поведінку програмного забезпечення.

Забезпечення якості програмного забезпечення стає все більш важливим, але іноді це суперечить термінам виконання, складній програмній інженерії та залежності від іншого програмного забезпечення. Таким чином, фаззинг став частиною процедури забезпечення якості перед випуском програмного забезпечення. Фаззинг особливо корисний для виявлення потенційних вразливостей безпеки, таких як:

1. Витоки пам'яті. Витік пам'яті – це тип помилки, яка виникає, коли комп'ютерна програма неправильно керує розподілом пам'яті таким чином, що

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				40
Змн.	Арк.	№ докум.	Підпис	Дата		

пам'ять, яка більше не потрібна, не звільняється. Витоки пам'яті уповільнюють роботу системи аж до моменту, коли система або запущений процес виходять з ладу.

2. Помилки потоку керування та інші помилки виконання. Помилка потоку керування – це помилковий стрибок у виконуваний програмі, спричинений зовнішніми перешкодами. Помилка виконання – це загальний термін для будь-якої помилки, яка виникає під час виконання програми.

3. Умови перегонів. Умова змагання або небезпека змагання – це поведінка системи, де результат залежить від послідовності або часу інших неконтрольованих подій. Це стає помилкою, коли події відбуваються не в тому порядку, який задумав програміст.

Усі ці типи помилок можна використати, коли результат є більш-менш передбачуваним, і їх можна використовувати для роботи на користь зловмисників. Мета фаззингу для тестувальників полягає в тому, щоб викликати неочікувану поведінку програми та перевірити, чи призводить це до помилки, яку можна використовувати.

Комбінації для атаки можуть варіюватись, та містити наступні дані:

- числа (цілі числа зі знаком/без знаку/з плаваючою точкою);
- символи (URL-адреси, введення командного рядка);
- метадані: текст, який вводить користувач;
- чисті двійкові послідовності.

Загальний підхід до фаззингу полягає у визначенні списків «відомо небезпечних значень» для кожного типу та впровадженні їх або рекомбінацій.

Серед сучасних рішень слід розрізняти фазери білого ящика і чорного ящика. Фаззинг чорної скриньки генерує вхідні дані для цільової програми без знання її внутрішньої поведінки чи реалізації. Фазер чорної скриньки може створювати вхідні дані з нуля.

Фазери білого ящика аналізують внутрішню структуру програми, і з кожним новим запуском фаззингу вони вчаться відстежувати та максимізувати

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				41
Змн.	Арк.	№ докум.	Підпис	Дата		

покриття коду. Фазери білого ящика зазвичай використовують адаптовані алгоритми та інтелектуальне обладнання під час тестування фаз, що робить їх більш ефективними та точними у виявленні вразливостей.

Отже, за допомогою техніки тестування фазинг можна здійснити брут форс метод - перевірку на слабкість та ненадійність паролів.

Також в утиліті Owasp Zap присутня можливість запустити атаку примусового перегляду – Forced browsing.

Примусовий перегляд — це техніка, яка використовується зловмисниками для отримання доступу до веб-сторінок із обмеженим доступом або інших ресурсів шляхом маніпулювання URL-адресою. Як впливає з назви, зловмисник примусово переглядає ресурс, на який у нього немає авторизації.

Ці атаки спрямовані на файли в каталозі веб-сервера або обмежені URL-адреси, які не перевіряють авторизацію. Ці ресурси вигідні зловмисникам, якщо містять конфіденційні дані. Це може стосуватися самого веб-сайту або клієнтів сайту. Конфіденційні дані можуть включати:

- облікові дані;
- вихідний код;
- резервне копіювання файлів;
- логування;
- конфігурація.

Примусовий перегляд є типовою проблемою для веб-сайтів, які мають різні ролі користувачів, як-от звичайні користувачі та адміністратори. Кожен користувач входить з однієї сторінки, але має доступ до різних меню та параметрів. Однак, якщо сторінки, на які ведуть ці меню, не є захищеними, користувач може вгадати назву дійсної сторінки та спробувати напяму отримати доступ до її URL-адреси.

ZAP дозволяє спробувати знайти каталоги та файли за допомогою примусового перегляду.

Параметри примусового перегляду, надані інструментом Owasp Zap, такі:

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				42
Змн.	Арк.	№ докум.	Підпис	Дата		

- примусовий перегляд сайту (Forced Browse Site): використовується для ідентифікації незв'язаного вмісту в каталозі домену, виконується за замовчуванням або шляхом встановлення власного словника;
- примусовий перегляд каталогу (Forced Browse Directory) : виконує ту саму функцію, що й попередній, відмінність полягає в тому, що він ідентифікує каталог, а не весь домен;
- примусовий перегляд каталогу (і дочірніх елементів) (Forced Browse Directory and children): те саме, що й вище, за винятком того, що він також визначає вміст підкаталогів.

Висновки до другого розділу

В даному розділі було проаналізовано інструменти для проведення сканування мережі на можливі вразливості. Оцінено недоліки та переваги кожного програмного забезпечення. Розглянуто детальну інформацію про утиліту Owasr Zap та досліджено її інтерфейс. Вивчено атаки, котрі тестувальник може реалізувати через утиліту Owasr Zap, та наведено можливі варіанти захисту веб-сайту від цих атак.

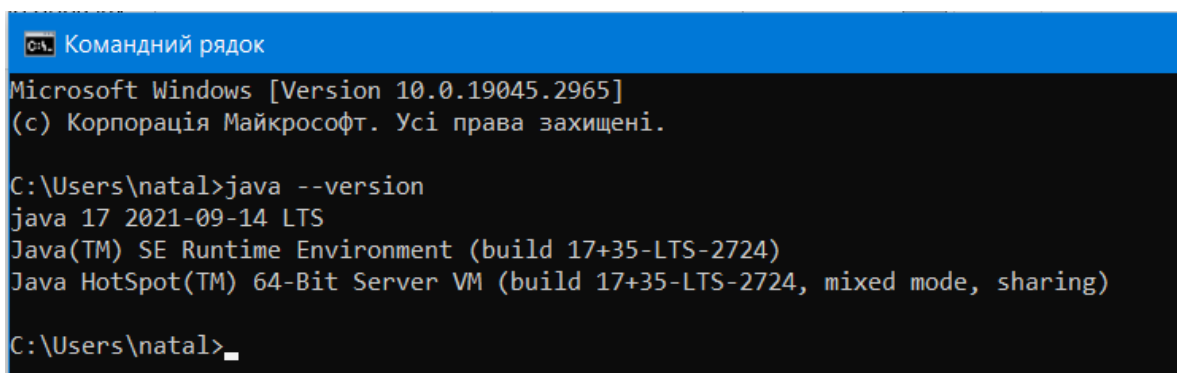
		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				43
Змн.	Арк.	№ докум.	Підпис	Дата		

РОЗДІЛ 3. НАЛАШТУВАННЯ, ПЕРЕВІРКА ФУНКЦІОНАЛУ ТА ПРОВЕДЕННЯ ТЕСТУВАННЯ ВЕБ-САЙТУ ЗА ДОПОМОГОЮ ДОДАТКУ Owasp Zар

3.1 Встановлення та налаштування програмного забезпечення

Утиліта Zар створює проксі-сервер між клієнтом та сайтом. Поки тестувальник переміщується на веб-сайті, Zар фіксує всі дії, а потім атакує сайт.

Owasp Zар є кросплатформним і для роботи вимагає лише наявності модулів Java 8+. Тому перед встановленням програмного додатку необхідно переконатись що модулі наявні на робочій станції або встановити їх. Перевірку можна виконати за допомогою командного рядка:



```
Командний рядок
Microsoft Windows [Version 10.0.19045.2965]
(c) Корпорація Майкрософт. Усі права захищені.

C:\Users\natal>java --version
java 17 2021-09-14 LTS
Java(TM) SE Runtime Environment (build 17+35-LTS-2724)
Java HotSpot(TM) 64-Bit Server VM (build 17+35-LTS-2724, mixed mode, sharing)

C:\Users\natal>
```

Рисунок 3.1 – Перевірка версії Java модулів на робочій станції

Оскільки Owasp Zар є безкоштовним інструментом з відкритим кодом, його можна завантажити у вільному доступі з офіційного сайту організації Owasp.

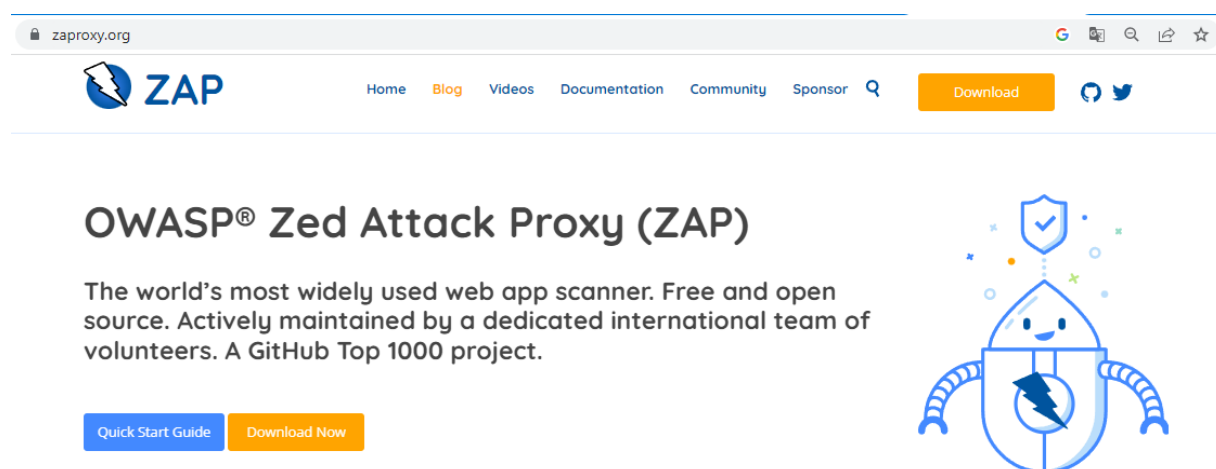


Рисунок 3.2 – Завантаження програмного додатку

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				44
Змн.	Арк.	№ докум.	Підпис	Дата		

Встановлення додатку не має особливих налаштувань, лише вибір мови графічного інтерфейсу додатку та місця розташування на комп'ютері куди буде встановлено програму.

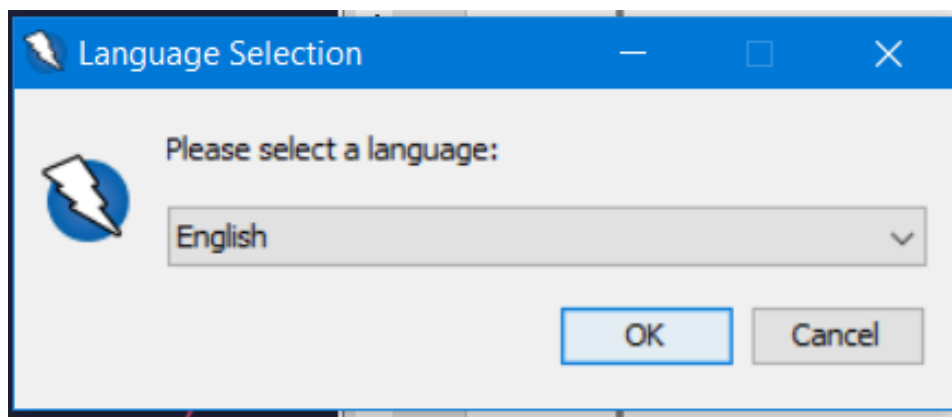


Рисунок 3.3 – Встановлення додатку Owasp Zap

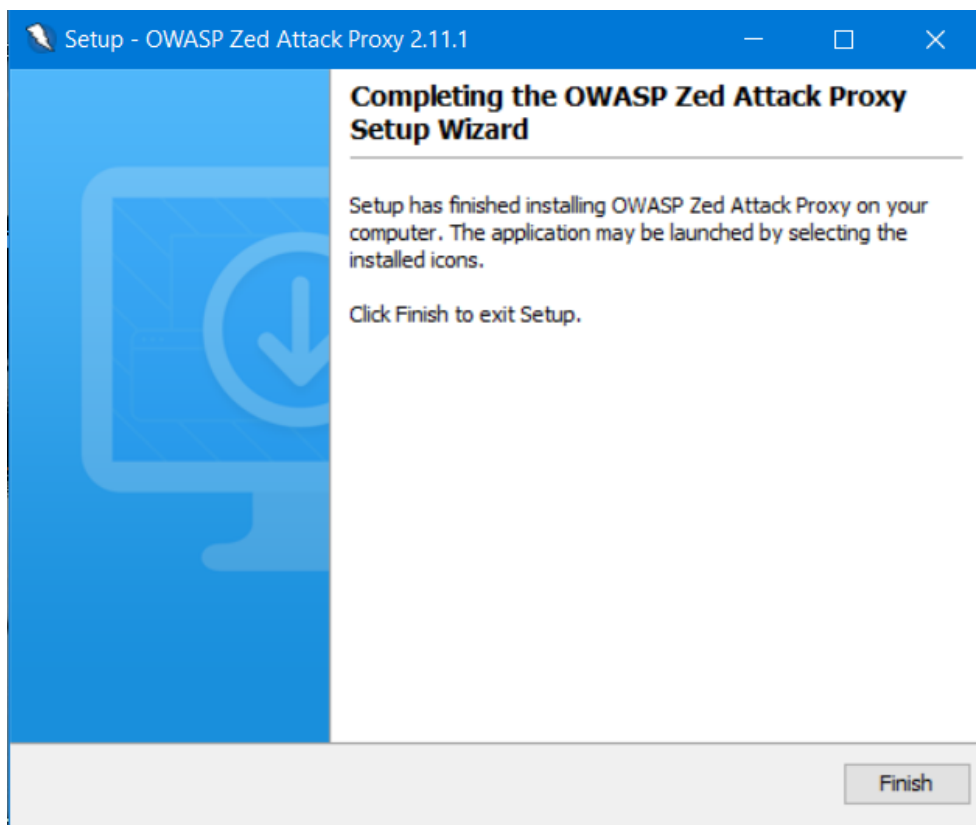


Рисунок 3.4 – Встановлення додатку Owasp Zap

Під час запуску, Zap запропонує зберегти сеанс. Якщо зберегти, то буде отримано конфігурацію свого веб-сайту або результати тестування пізніше, поки що досить значення: "No,I do not want to persist the session...".

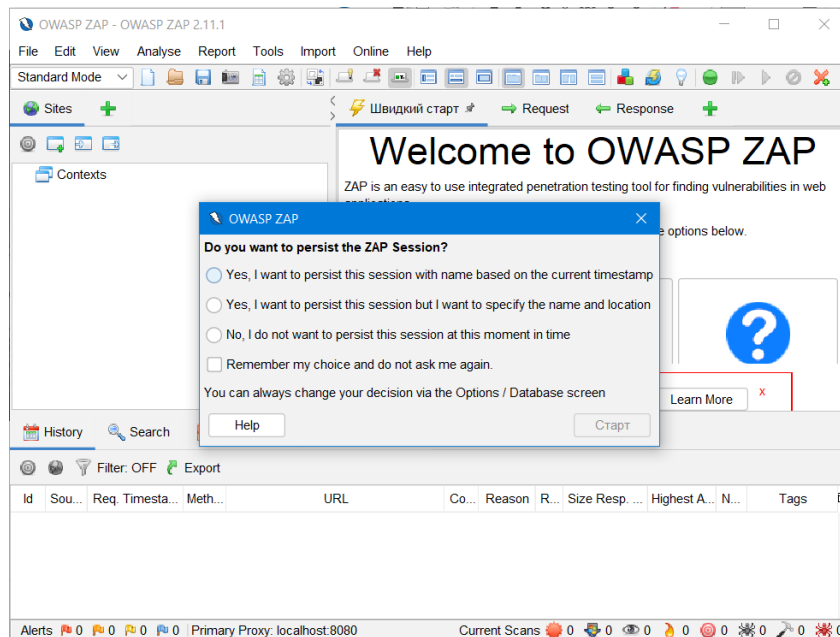


Рисунок 3.5 – Перший запуск програми

Для коректної роботи необхідно згенерувати та імпортувати SSL сертифікат. Оскільки всі запити та відповіді передаються через Zap, перевірка сертифіката не вийде для сайтів які використовують SSL (HTTPS) і з'єднання буде розірвано. Щоб цього не сталося Zap створює сертифікат SSL для кожного хоста, підписаний його власним сертифікатом центру сертифікації. Цей сертифікат створюється під час першого запуску програмного продукту і зберігається локально. Щоб використовувати проксі-сервер Zap з веб-сайтами, необхідно встановити сертифікат Zap.

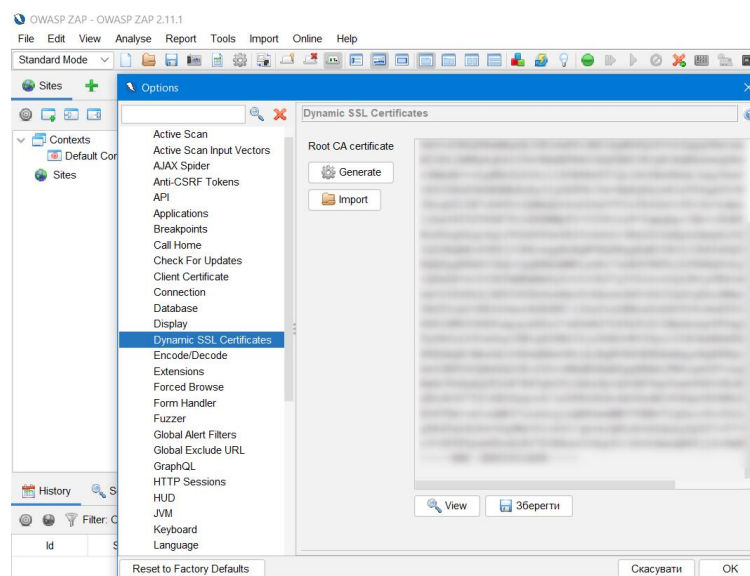


Рисунок 3.6 – Генерація SSL-сертифікату

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				46
Змн.	Арк.	№ докум.	Підпис	Дата		

Наступним кроком необхідно імпортувати згенерований сертифікат у браузер, котрий буде використовуватись для проведення тестування, наприклад Mozilla Firefox.

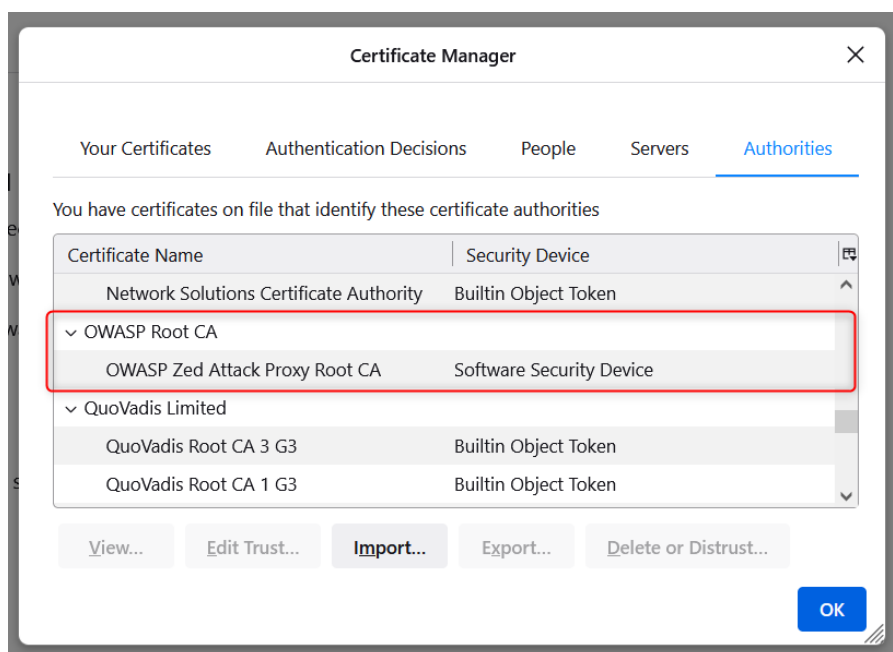


Рисунок 3.7 – Імпорт SSL-сертифікату

Також необхідно налаштувати Zar на роль проксі-сервера для локального веб-трафіку.

Проксі-сервер – проміжний сервер або комп'ютер, який є посередником між комп'ютером власника та кінцевим сервером. Проксі-сервер, як і VPN, служить для кількох цілей: підвищує рівень конфіденційності в інтернеті та допомагає обійти регіональні блокування доступу до окремих сайтів.

Проксі ідеально підходить для перехоплення відповідей на запити, тобто дозволяє бачити весь поточний трафік між браузером та веб-сервером, показуючи простим способом заголовки та тіло HTTP. Крім того, тестувальник може власноруч змінювати HTTP-трафік в обох напрямках зв'язку (між веб-сервером та браузером).

Для налаштування проксі необхідно перейти в налаштування проксі сервера на комп'ютері та змінити налаштування на наведені нижче. Локальна Ір-адреса – 127.0.0.1 та 8080 як порт.

Налаштування проксі-сервера вручну

Використовувати проксі-сервер для підключень Ethernet або Wi-Fi.
Ці настройки не застосовуються до підключень VPN.

Використовувати проксі-сервер

☒ Увімкнуто

Адреса

http://https=127.0.0.1

Порт

8080

Не використовувати проксі-сервер для адрес, що починаються з елементів, зазначених нижче. Додайте записи через крапку з комою (;).

☒ Не використовувати проксі-сервер для локальних адрес (адрес інтрамережі)

Зберегти

Рисунок 3.8 – Налаштування проксі-сервера для робочої станції

Обов'язково необхідно переконатись що налаштування програмного забезпечення відповідають налаштуванням проксі-сервера на комп'ютері.

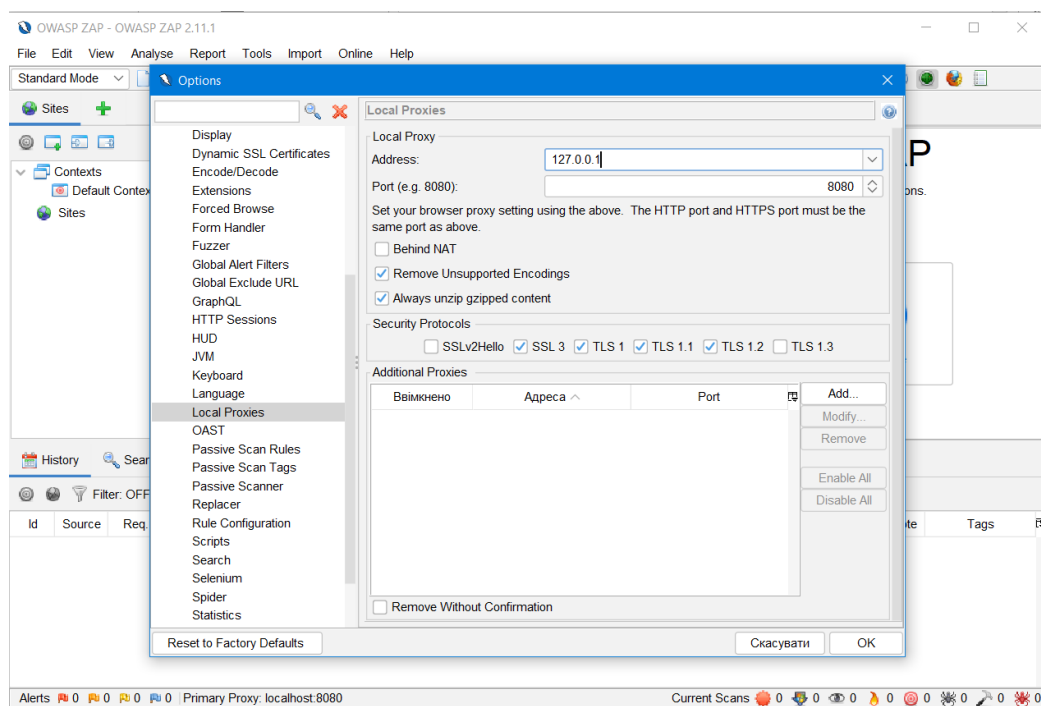


Рисунок 3.9 – Налаштування проксі сервера для програми OWASP

В налаштуваннях браузера теж необхідно змінити дані, обов'язковою умовою є однаковий порт та адреса локального хосту.

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				48
Змн.	Арк.	№ докум.	Підпис	Дата		

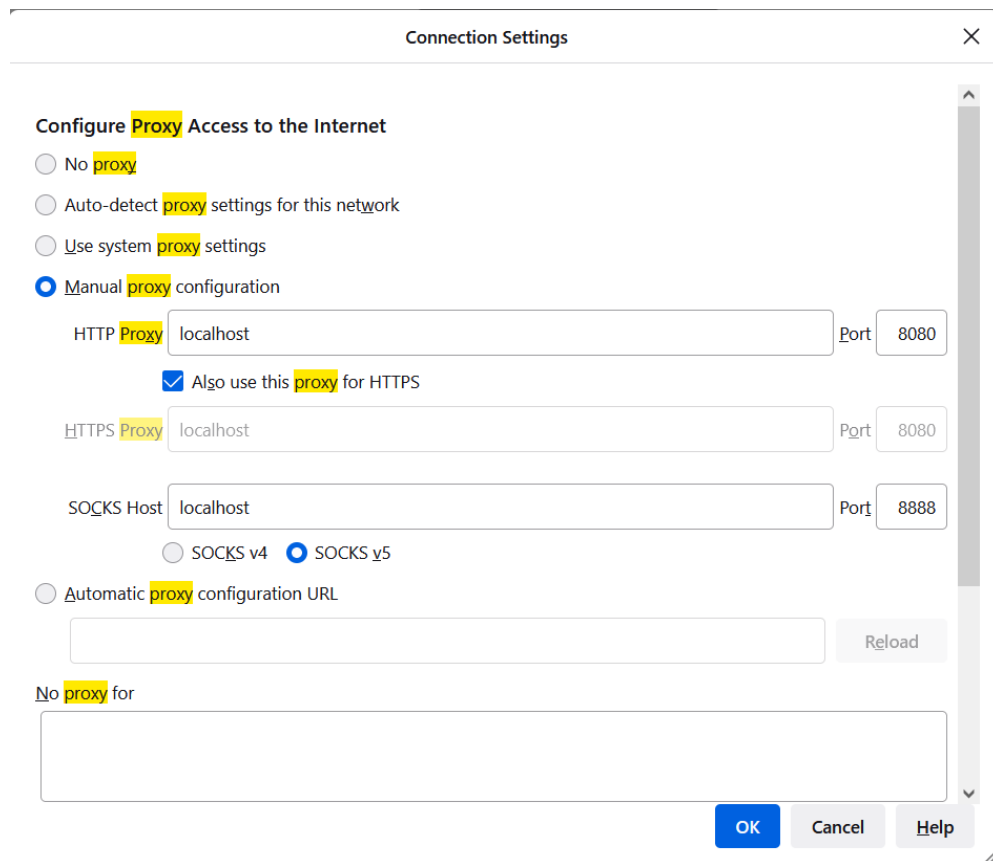


Рисунок 3.10 – Налаштування проксі-сервера для браузера

3.2 Моделювання кіберінцидентів

В програмному забезпеченні Owasp Zap можна проводити автоматичне тестування (automated) та ручне (manual). Автоматичне тестування є більш складним варіантом, оскільки скануються усі знайдені сторінки сайту. Ручне тестування займає менше ресурсів та є простішим, оскільки пентестер сам вказує сторінки які потрібно просканувати через браузер.

Для виконання атак було підібрано спеціальні веб-сайти де міститься певна вразливість. Перед початком тестування необхідно вказати цільовий сайт та додаткові налаштування для сканування:

1. «URL to attack» – задається URL сайту, який необхідно просканувати чи атакувати.
2. «Use traditional spider» – використання стандартного павука при скануванні.
3. «USE ajax spider» – використання аях «павука» при скануванні.

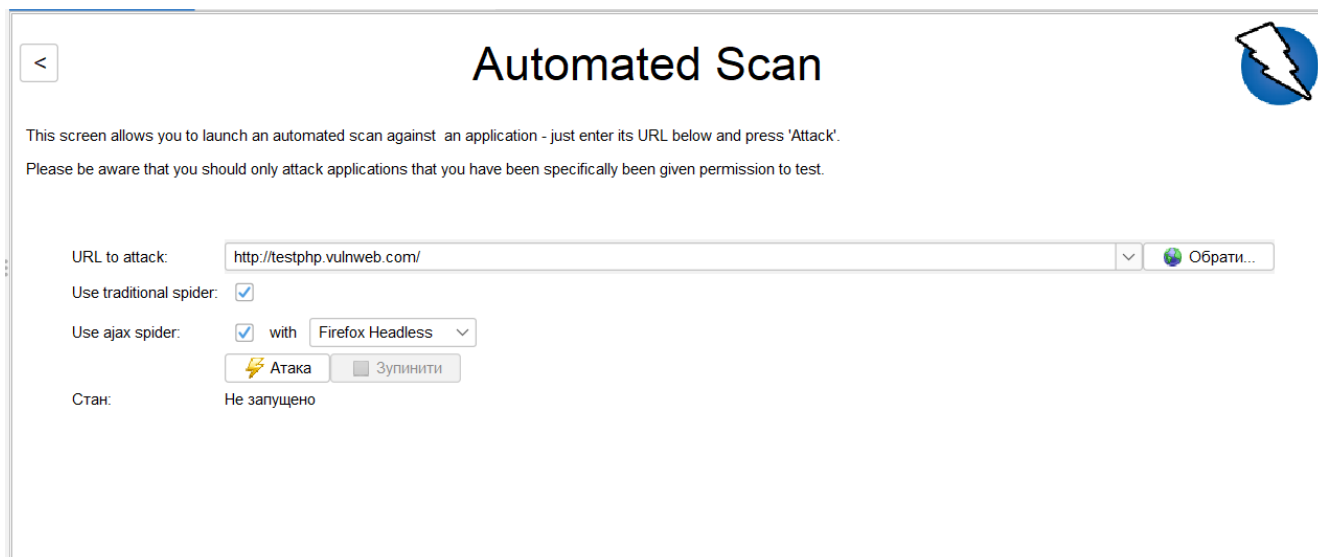


Рисунок 3.11 – Автоматичне сканування

Для того щоб почати сканування необхідно натиснути кнопку «Attack». Після цього починається процес сканування, запускається «павук», котрий шукає всі посилання, котрі мають відношення до цільового сайту. В колонці «Method» відображаються методи, котрі використовуються для отримання URL та їх статуси (колонка «Processed»).

Як правило, використовуються методи «GET» та «POST». Якщо статус «200», то відображається зелений кружок, якщо статуси «404», «500», тощо – то червоний.

Визначення статусів:

- «200» – успішний запит;
- «302» – запитуваний документ тимчасово доступний по іншому посиланню;
- «403» – доступ до ресурсу заборонений;
- «404» – дана сторінка не існує;
- «500» – внутрішня проблема серверу. Виникає коли сервер не може обробити запит.

Залежно від структури сайту та його наповнення сканування на URL може тривати тривалий час. Після того, як всі URL знайдені, запускається активне сканування.

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				50
Змн.	Арк.	№ докум.	Підпис	Дата		

Для перегляду атак, які використовуються при активному скануванні і для перегляду їх прогресу виконання необхідно перейти за позначкою «Show scan progress details»:

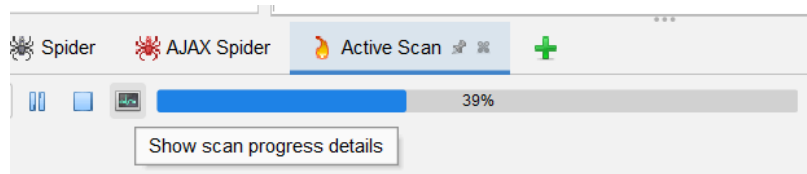


Рисунок 3.12 – Show scan progress details

Назва вразливості відображається в колонці «Analyser», прогрес сканування відображається в колонці «Progress». У стовпці «Elapsed» виводиться час, витрачений на сканування цієї вразливості. Будь яку перевірку можна зупинити натиском на синю кнопку «Scip active scanner»:

http://testphp.vulnweb.com/ Scan Progress						
Progress Response Chart						
Host: http://testphp.vulnweb.com						
	Strength	Progress	Elapsed	Reqs	Оновлення	Status
Analyser			00:00.000	0		
Plugin						
Path Traversal	Medium		00:00.000	0	0	✓
Remote File Inclusion	Medium		00:00.000	0	0	✓
Source Code Disclosure - /WEB-INF folder	Medium		00:00.730	3	0	✓
External Redirect	Medium		00:00.730	0	0	✓
Server Side Include	Medium		00:00.000	0	0	✓
Cross Site Scripting (Reflected)	Medium		00:00.000	0	0	✓
Cross Site Scripting (Persistent) - Prime	Medium		00:00.000	0	0	✓
Cross Site Scripting (Persistent) - Spider	Medium		00:00.602	1	0	✓
Cross Site Scripting (Persistent)	Medium		00:00.000	0	0	✓
SQL Injection	Medium		00:00.000	0	0	✓

Рисунок 3.13 – Прогрес автоматичного сканування

Після того як сканування буде завершено, всі вразливості будуть відображатись в розділі «Alerts». Якщо розкрити групу вразливості, то будуть відображені всі посилання, де були знайдені лазівки даного типу:



Рисунок 3.14 – Результати сканування

За результатами сканування відображається червоний прапорець біля групи з назвою «Cross Site Scripting», це означає, що на сайті присутня вразливість міжсайтового скриптингу. Для того, щоб дізнатись за яким саме посиланням

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				
Змн.	Арк.	№ докум.	Підпис	Дата		51

можна знайти дану лазівку необхідно натиснути на розділ. Також в правій частині відображається детальний опис вразливості, котрий включає в себе тип атаки, опис, спосіб усунення, ризик:



Рисунок 3.15 – Вразливість міжсайтового скриптингу

Якщо лазівку знайдено в коді, то його буде виділено. Щоб тестувальник не витрачав час на пошуки, навіть за кодом можна зрозуміти в якому саме місці на сторінці присутня лазівка і там спрацює атака. Можна відкрити дане посилання, для цього необхідно натиснути правою клавішею на групу, та обрати браузер в якому повинен відкритись веб-сайт.

На сторінці цього розділу сайту присутнє поле для введення тексту. Схоже, дане поле без валідації, тому можна ввести шкідливий код та він виконається.

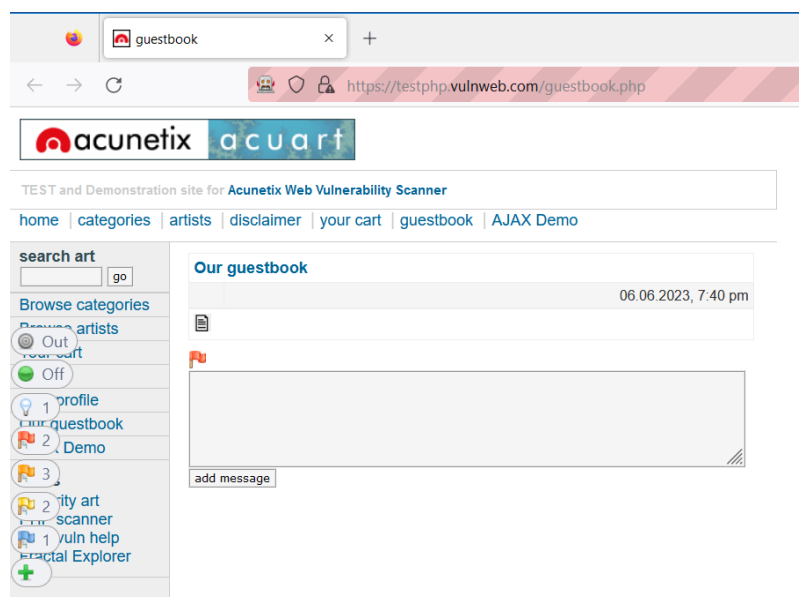


Рисунок 3.16 – Посилання з вразливістю

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				52
Змн.	Арк.	№ докум.	Підпис	Дата		

Для перевірки реагування сайту на атаку міжсайтового скриптингу, тестувальник може випробувати наведений в утиліті Owasp Zap код, скопіювати його та ввести в поле яке вказано в деталях вразливості.

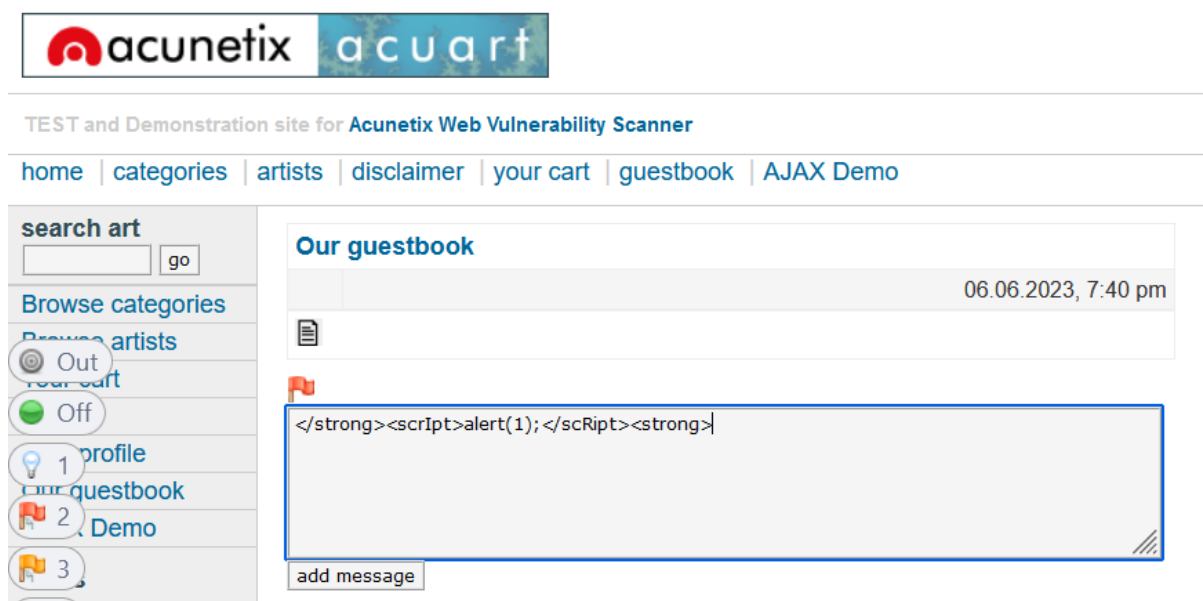
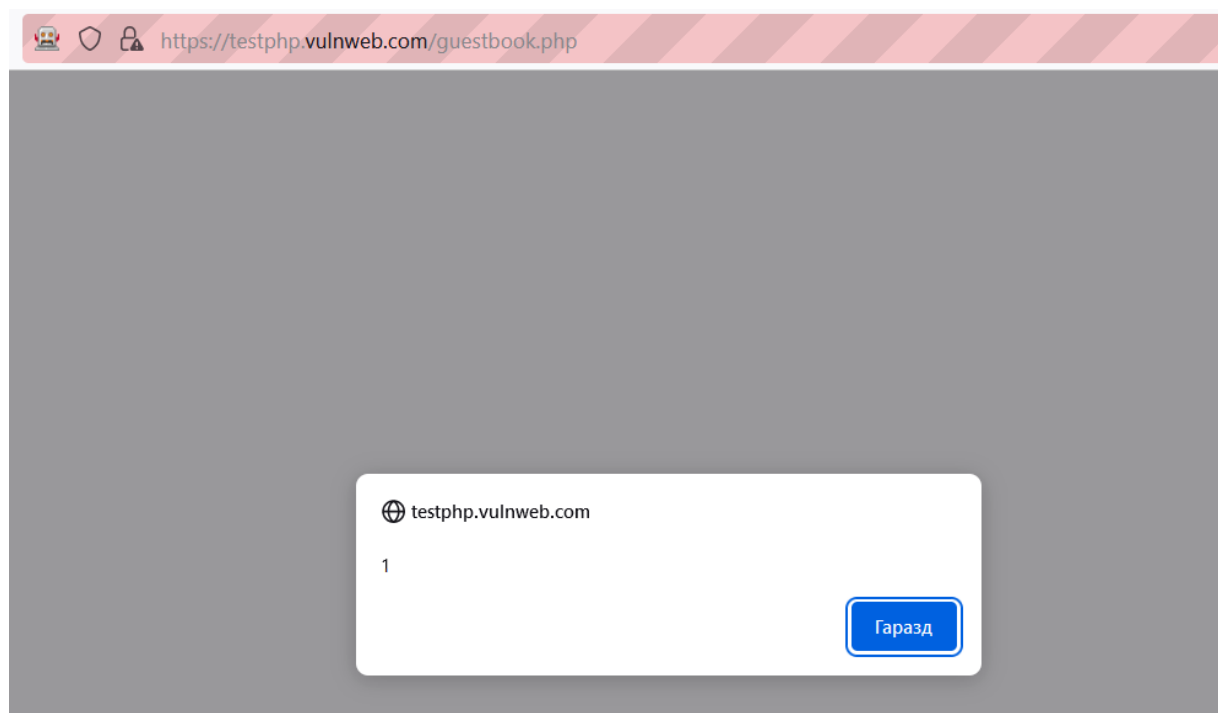


Рисунок 3.17 – Вразливість «міжсайтовий скриптинг»

Як видно, сайт сприйняв цей Javascript код та відобразив діалогове вікно з текстом «1».



Риссунок 3.18 Вразливість «міжсайтовий скриптинг»

Також в розділі «Оповіщення» присутня критична вразливість SQL-injection:

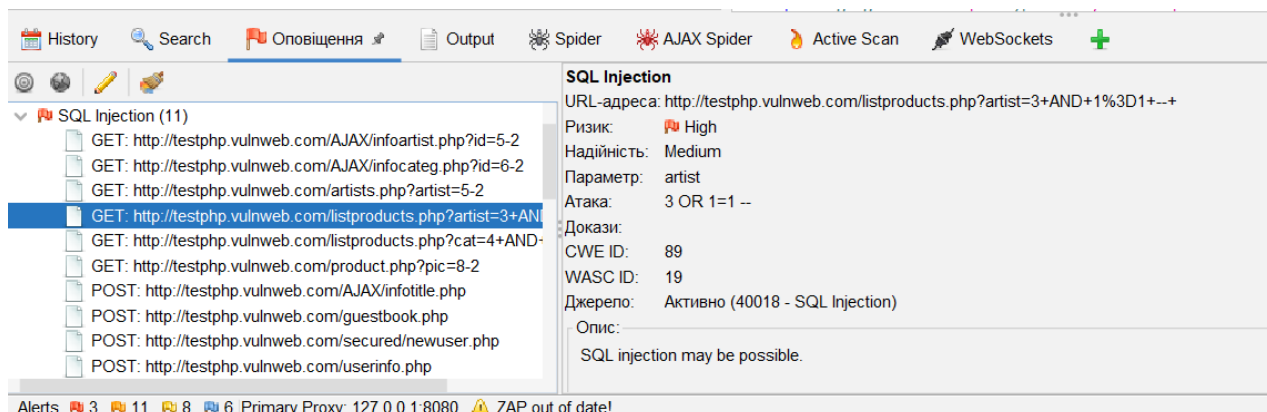


Рисунок 3.19 – Результати сканування

У Sql-ін'єкціях зазвичай використовують вираз «AND 1=1». Адже у комп'ютерних обчисленнях «1=1» завжди є істинною умовою. Отже якщо хакер планує зламати безпеку під час аутентифікації, достатньо ввести даний вираз в поле. Для прикладу посилання: «http://testphp.vulnweb.com/artists.php?artist=5-2» утотожнює запит в базу даних «SELECT data FROM artists WHERE id=3».



Рисунок 3.20 – Запит до бази даних через URL сайту

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				54
Змн.	Арк.	№ докум.	Підпис	Дата		

Результатом сканування та проведення атаки є посилання «<http://testphp.vulnweb.com/listproducts.php?artist=3+AND+1%3D1+-->», у якого запит до бази даних виглядає наступним чином – «SELECT data FROM artists WHERE id=3 and 1=1».

Одним із найпоширеніших і найпростіших методів для перевірки є використання однієї лапки «'», яка порушує запит до бази даних.

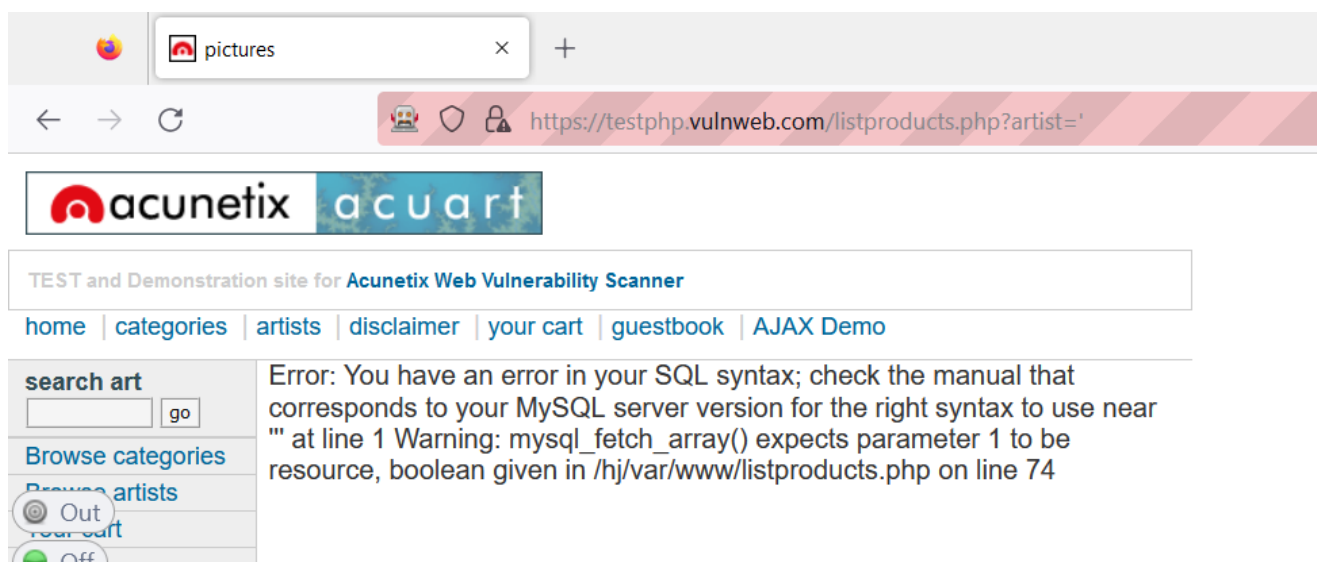


Рисунок 3.21 SQL-injection

Після відправлення запиту на сторінці з'явилося повідомлення про помилку в синтаксисі SQL запиту і це підтверджує що можна провести SQL-ін'єкцію.

Для прикладу, ціль зловмисника – увійти під обліковим записом користувача, у випадку коли відомо лише ім'я («test»), а пароль – ні. У формі для входу в обліковий запис, достатньо вписати відоме ім'я користувача, а замість паролю – істинний вираз, наприклад «' or '1'='1'». Запит до бази даних буде виглядати наступним чином:

«SELECT * FROM users WHERE name='test' and password="" or '1'='1'».

Оскільки «1=1» – це умова, яка завжди має значення «true», весь оператор «WHERE» буде істинним, незалежно від наданого імені користувача чи пароля. Тобто перевірка пароля не відбувається, можна зробити висновок що запит виглядає скоріше за все набагато простіше:

«SELECT * FROM users WHERE name='test'».

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				55
Змн.	Арк.	№ докум.	Підпис	Дата		

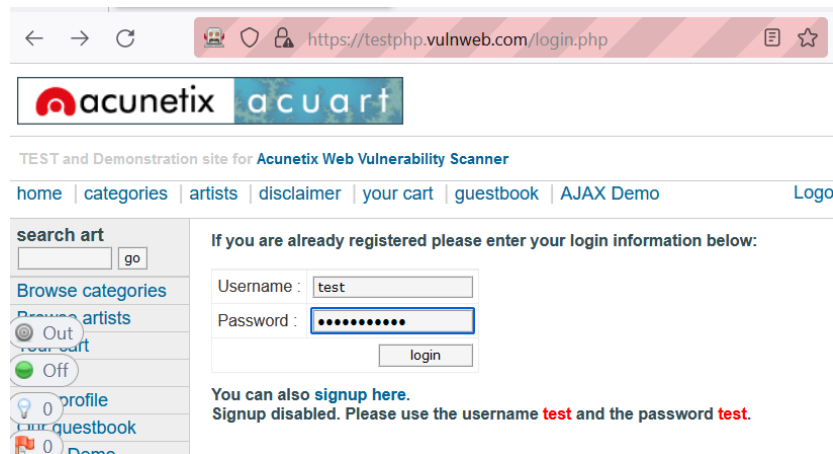


Рисунок 3.22 – SQL-injection під час авторизації

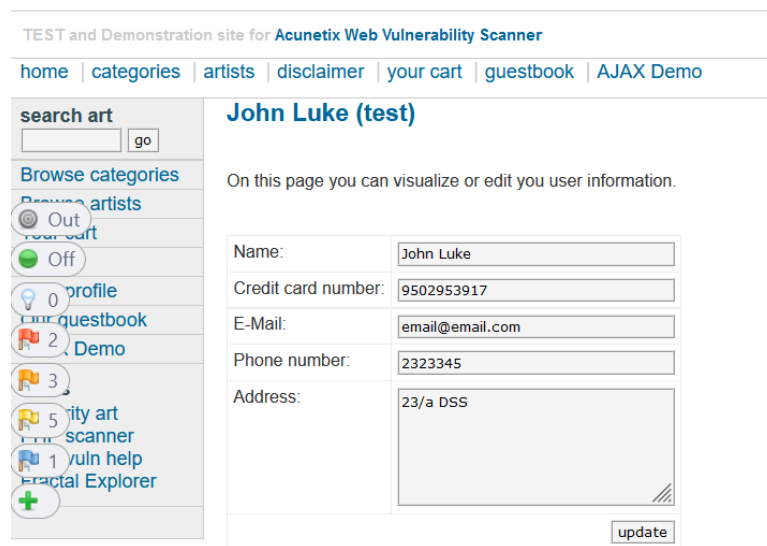


Рисунок 3.22 – SQL-injection під час авторизації

Тепер хакер має доступ до особистих даних користувача і може використати їх в своїх цілях. Тому, в утиліті Owasp Zар, Sql-ін'єкція позначена червоним прапорцем, що означає високу критичність вразливості.

Ще один спосіб як дізнатись пароль користувача, це використання атаки брут форс. Утиліта Owasp Zар надає можливість здійснити дану атаку за допомогою техніки фазингу. Різниця між брут форсом та Sql-ін'єкцією лише в тому що, під час SQL-ін'єкції хакер не дізнається пароль, він йому буде не потрібний, адже він і так отримає доступ до облікового запису. А під час брут форсу навпаки, для початку хакеру необхідно надати можливі варіації паролю, далі утиліта Zар по чергово їх підставить і визначить необхідний. Процедура генерації даних потребує досить багато часу і ресурсів. Хоча одним із варіантів,

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				56
Змн.	Арк.	№ докум.	Підпис	Дата		

це написання функцій для випадкового створення string значень і запису їх у файл. Також існують різні спеціальні утиліти для генерації паролів, наприклад Crunch чи Nahcat.

Для підбору паролю в утиліті Owasp ZAP використовується фазер.

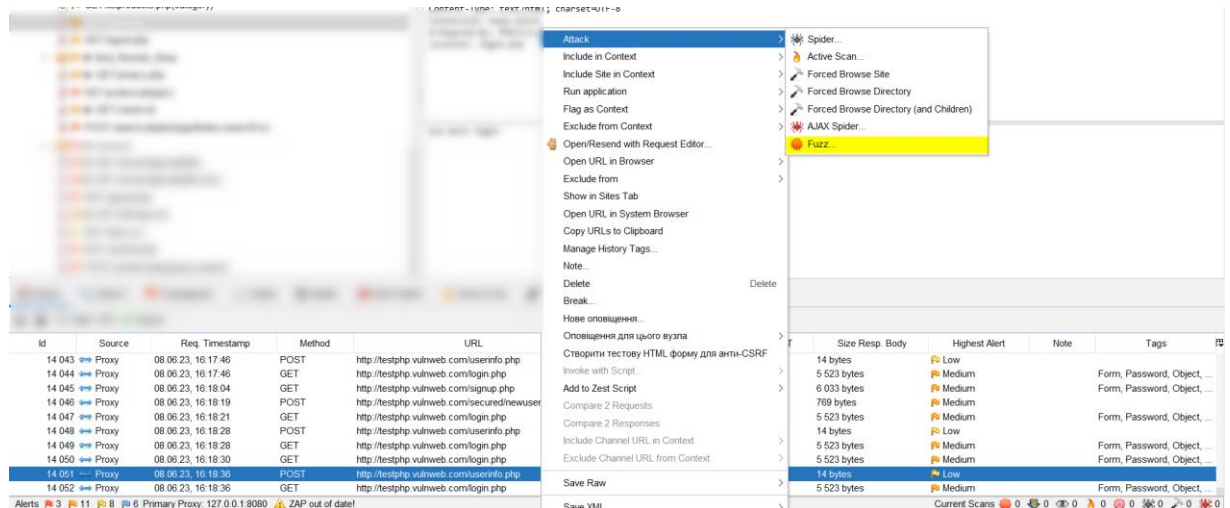


Рисунок 3.23 – Фазинг

Наступним кроком необхідно виділити область, де будуть змінюватись та підбиратись дані. В даному випадку це пароль користувача:

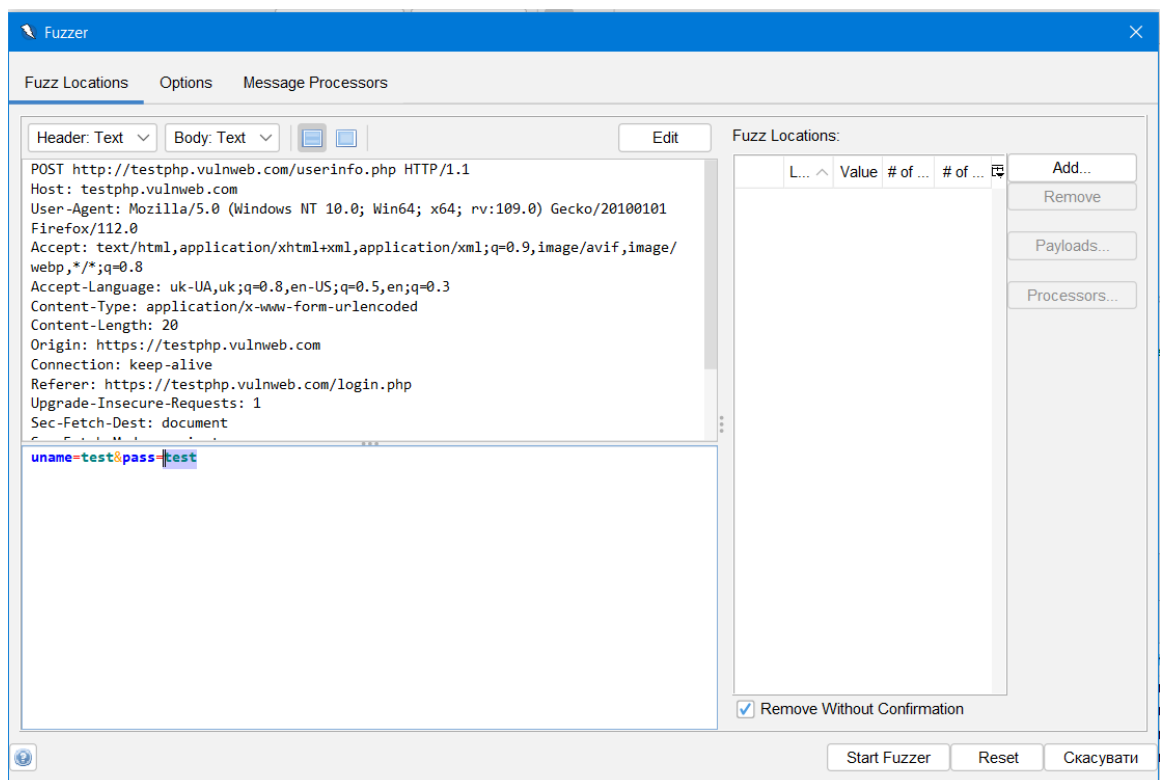


Рисунок 3.24 – Налаштування брут форс атаки

Далі необхідно додати дані, які утиліта буде підставляти у виділену область. В даному випадку варіації паролей, тому тип контенту – «string». Для цього необхідно натиснути у полі «Fuzz Locations» кнопку «Add».

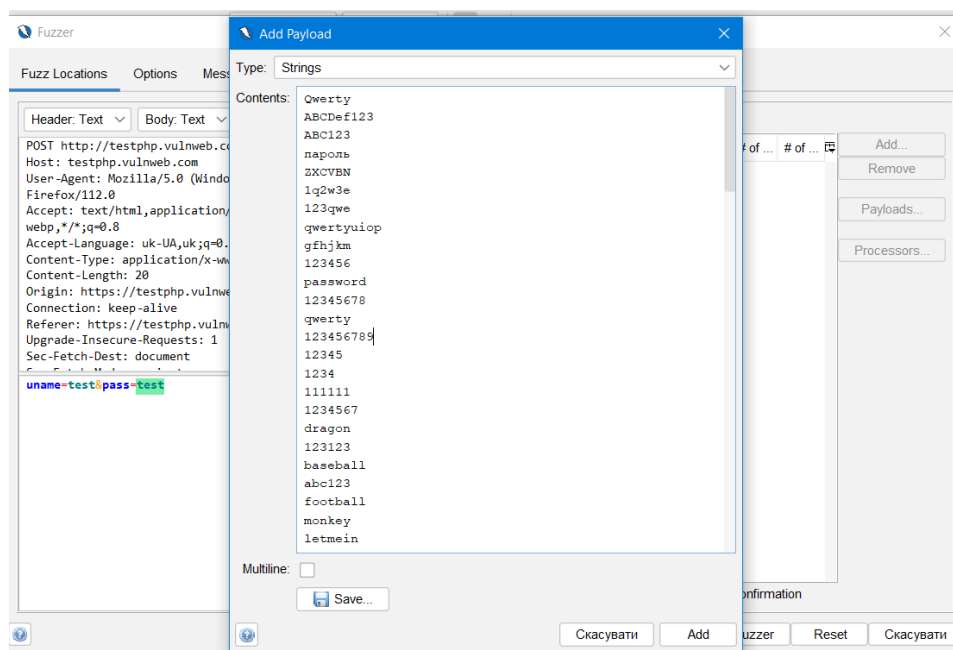


Рисунок 3.25 – Налаштування брут форс атаки

Важливо пам'ятати, що пароль буде підбиратись саме для користувача з логіном «test».

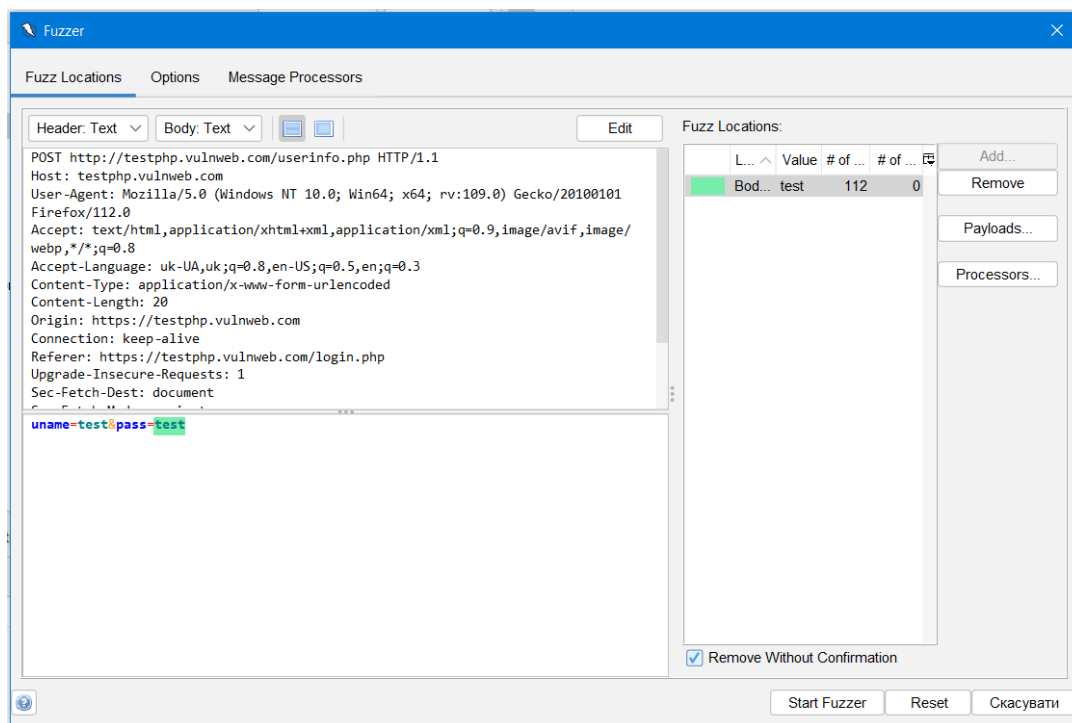


Рисунок 3.26 – Налаштування брут форс атаки

Після даних налаштувань можна запускати атаку кнопкою «Start fuzzer».

З'явиться нове вікно «Fuzzer» в якому буде відображатись прогрес виконання фазингу.

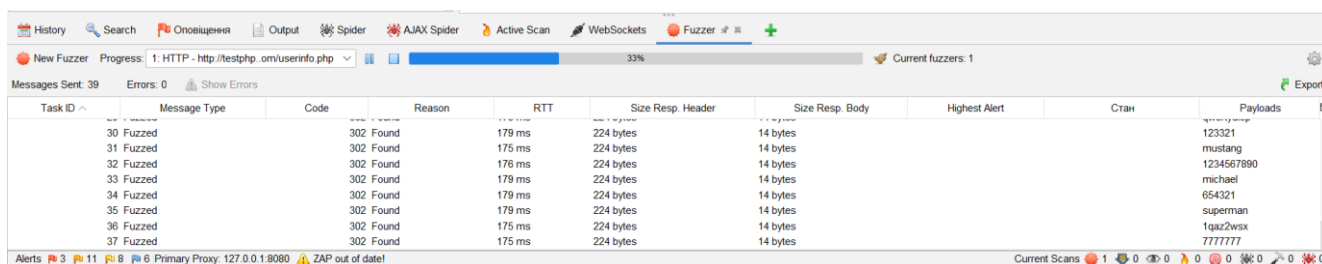


Рисунок 3.27 – Налаштування брут форс атаки

Після завершення сканування, правильно підібраний пароль, котрий використовується користувачем для авторизації, позначений жовтим прапорцем з повідомленням «Reflected». Також як видно, статус відповіді в усіх запитах повертає «302 Found» помилку, лише у дійсного паролю статус – «200 OK».

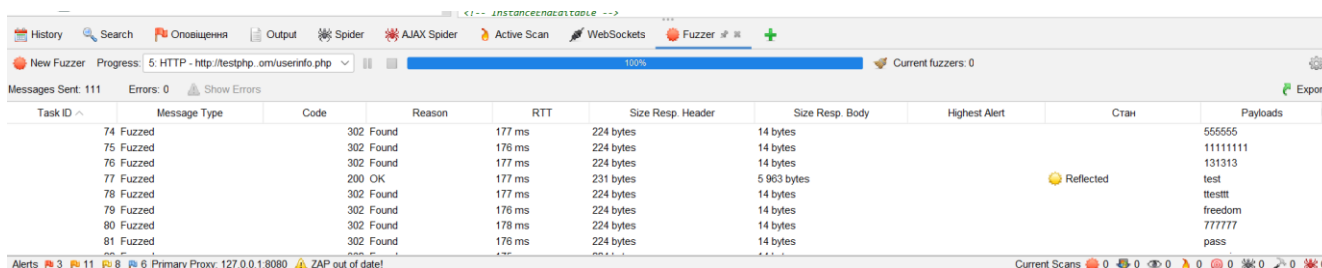


Рисунок 3.28 – Результат фазингу

На даному етапі можна вважати що сканування веб-сайту та його перевірка завершені. Наступним кроком тестувальник займається створенням звітів за результатами тесту.

3.3 Методи захисту від змодельованих атак та аналіз результатів тестування веб-сайту

Утиліта Owasp Zap дозволяє генерувати звіти за результатами сканування в різних форматах. Для того щоб створити звіт, необхідно перейти в розділ «Report» і обрати «Generate Report». У відкритому вікні буде запропоновано змінити звіт перед його створенням. В розділі «Score» можна задати базові налаштування, такі як заголовок, ім'я, шлях, куди звіт буде збережено.

Рисунок 3.29 – Створення звіту

В розділі «Template» можна обрати шаблон для збереження звіту. В даному випадку звіт буде створено за допомогою шаблону «High Level Report Sample», в якому знайдені вразливості будуть відображатись у вигляді кругової діаграми.

Рисунок 3.30 – Створення звіту

В розділі «Filter» можна обрати, які категорії ризику включати в звіт.

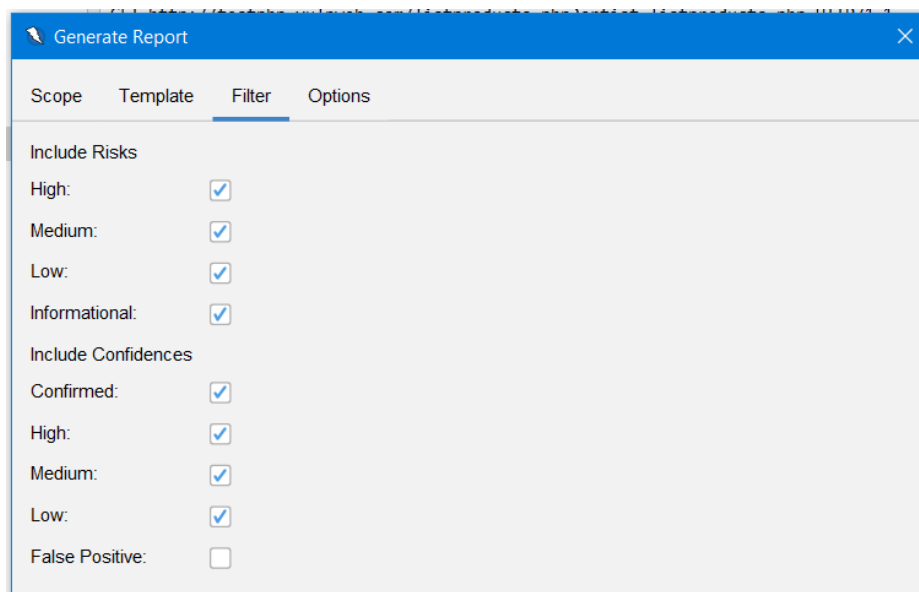


Рисунок 3.31 – Створення звіту

Після того як всі налаштування генерації звіту буде виконано, необхідно натиснути кнопку «Generate report» і звіт буде створено та розміщено за шляхом який було вказано в налаштуваннях.

У звіті відображається діграма з категоріями ризику. Прапорці виділено певним кольором:

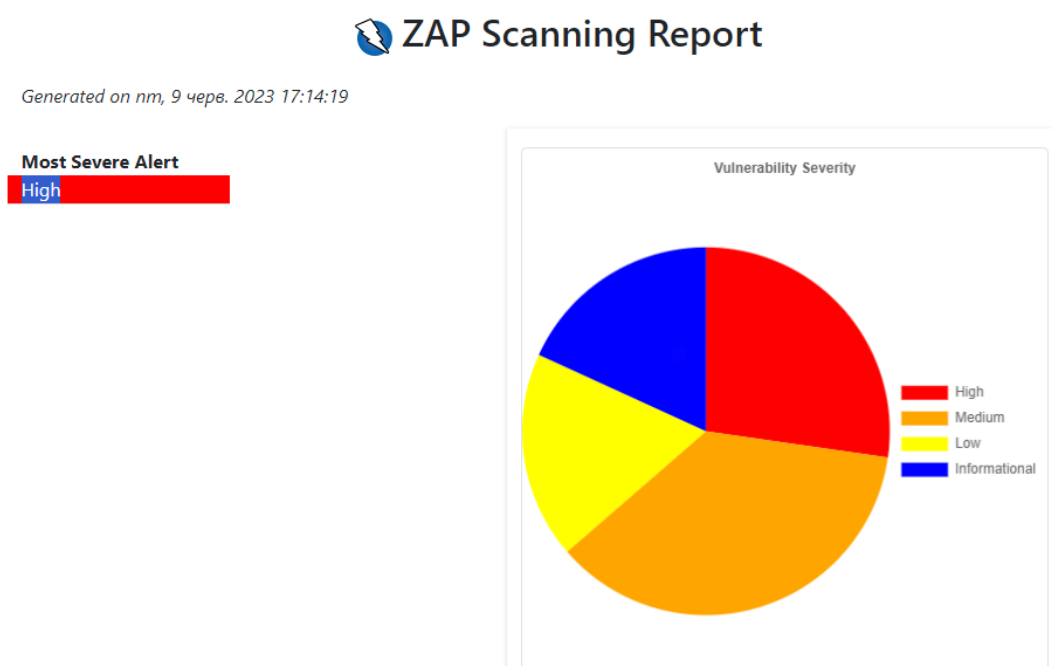


Рисунок 3.32 – Звіт сканування

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				61
Змн.	Арк.	№ докум.	Підпис	Дата		

Далі відображається таблиця, в якій наведено частоту виявлення вразливостей:

Most Common Bug

X-Content-Type-Options Header Missing (84)

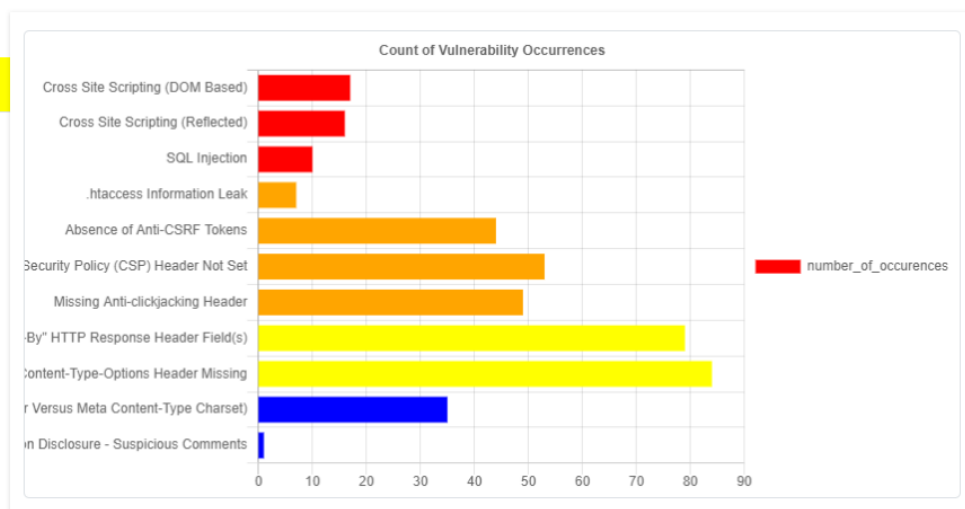


Рисунок 3.33 – Звіт сканування

Також під таблицею розміщено додатковий опис впливу на сайт кожної з вразливостей. Де користувач може детально з ними ознайомитись.

Vulnerability Impact

#	Name	Impact
1	Cross Site Scripting (DOM Based) [1] [2]	Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself is usually written in HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology.
2	Cross Site Scripting (Reflected) [1] [2]	Cross-site Scripting (XSS) is an attack technique that involves echoing attacker-supplied code into a user's browser instance. A browser instance can be a standard web browser client, or a browser object embedded in a software product such as the browser within WinAmp, an RSS reader, or an email client. The code itself is usually written in HTML/JavaScript, but may also extend to VBScript, ActiveX, Java, Flash, or any other browser-supported technology.
3	SQL Injection [1]	SQL injection may be possible.
4	.htaccess Information Leak [1]	htaccess files can be used to alter the configuration of the Apache Web Server software to enable/disable additional functionality and features that the Apache Web Server software has to offer.
5	Absence of Anti-CSRF Tokens [1] [2]	No Anti-CSRF tokens were found in a HTML submission form.
6	Content Security Policy (CSP) Header Not Set [1] [2] [3] [4] [5] [6] [7]	Content Security Policy (CSP) is an added layer of security that helps to detect and mitigate certain types of attacks, including Cross Site Scripting (XSS) and data injection attacks. These attacks are used for everything from data theft to site defacement or distribution of malware. CSP provides a set of standard HTTP headers that allow website owners to declare approved sources of content that browsers should be allowed to load on that page — covered types are JavaScript, CSS, HTML frames, fonts, images and embeddable objects such as Java applets, ActiveX, audio and video files.

Рисунок 3.34 – Звіт сканування

Усі вразливості знаходять в порядку зменшення важливості. На початку знаходяться критичні вразливості, на які варто в першу чергу звернути увагу.

Також ефективно зберігати звіт в HTML форматі, він налає більше інформації для ознайомлення з результатами сканування.

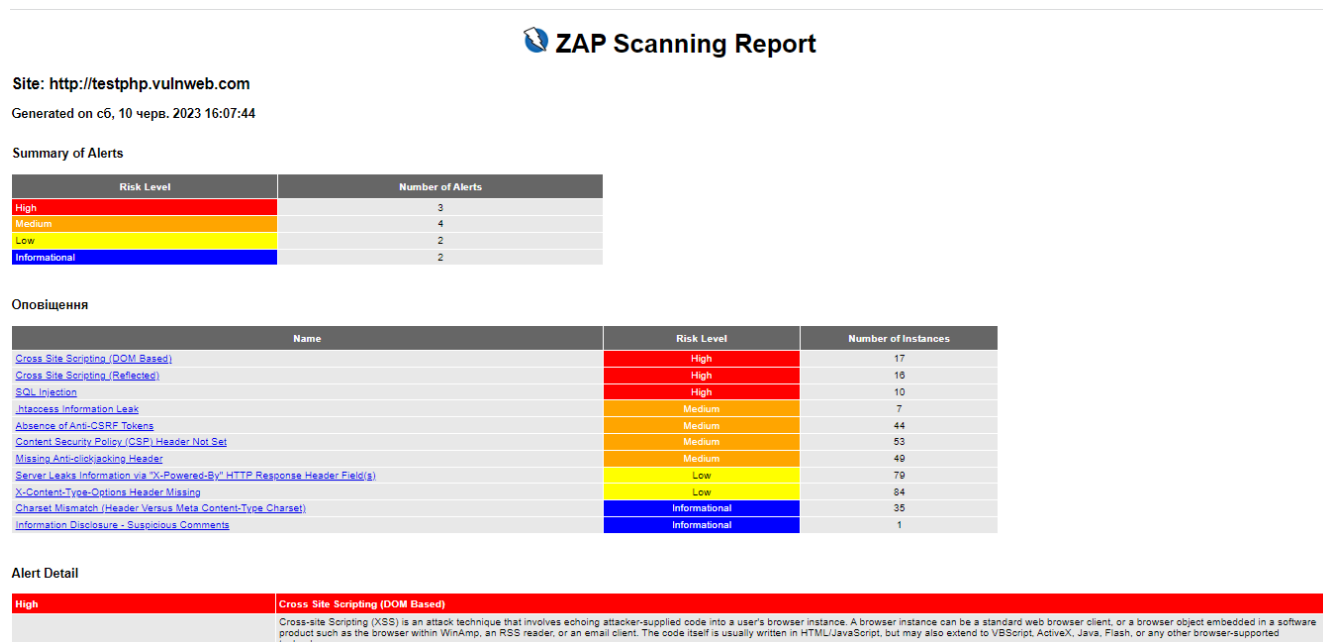


Рисунок 3.35 – Звіт сканування

Виконання технічної сторони оцінювання – це лише половина загального процесу оцінювання. Кінцевим продуктом є створення добре написаного та інформативного звіту. Звіт має бути легким для розуміння та висвітлювати всі ризики, виявлені на етапі оцінки. Звіт має бути привабливим як для виконавчого керівництва, так і для технічного персоналу. Звіт, в якійсь мірі, допомагає захистити веб-програму від останніх загроз.

Отже, створення звіту є основним кроком в проведенні пентестингу. Адже за його результатами компанія може визначити захищеність сайту чи методи для покращення його безпеки.

Є багато причин, чому люди повинні захищати свої дані. Однією з причин є те, що якщо дані не захищені. Неавторизовані люди можуть отримати до них доступ і використовувати їх. У результаті це може призвести до крадіжки особистих даних, шахрайства та інших злочинів. Інша причина полягає в тому, що якщо дані не захищені, вони можуть бути втрачені або знищені, що може спричинити значні незручності або навіть фінансові труднощі.

У разі міжсайтового сценарію є кілька кроків, які можна виконати, щоб виправити веб-сайт. Першим кроком у відновленні після міжсайтового сценарію є визначення місця розташування вразливості. Отримавши інформацію про місцезнаходження зловмисного програмного забезпечення, потрібно видалити шкідливий вміст або погані дані із бази даних. Також не завадило б перевірити решту сайту та файлових систем на наявність бекдорів. Коли відбувається компрометація, важливо змінити всі паролі та секрети програми, як тільки вразливість буде виправлено. Головним аспектом є налаштування брандмауера веб-додатків для фільтрації зловмисних запитів до веб-сайту. Вони можуть бути особливо корисними для забезпечення захисту від нових вразливостей до того, як будуть доступні виправлення.

Щоб захистити веб-сайт від міжсайтового сценарію, потрібно посилити всі засоби безпеки. Немає єдиної стратегії для запобігання атаці міжсайтового сценарію. Для захисту веб-сайту від XSS-атак необхідні деякі захисні заходи. До них входять значення дозволеного списку, обмеження HTML у вхідних даних і використання брандмауера веб-додатків (WAF).

Web Application Firewall аналізує весь доступ користувачів до критичних для бізнесу програм та захищає програми та дані від кібератак. Даний брандмауер необхідно купувати, на жаль безкоштовна версія недоступна. Для прикладу платформа безпеки та захисту веб-сайтів – Sucuri.



Рисунок 3.36 – Програмне забезпечення Sucuri

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				64
Змн.	Арк.	№ докум.	Підпис	Дата		

Sucuri автоматично створює сертифікати SSL для сервера брандмауера. Професійні та бізнес-плани можуть завантажувати власні сертифікати SSL. Після активації брандмауер перехоплює та перевіряє всі вхідні пакети, щоб переконатися, що надходять лише безпечні запити.

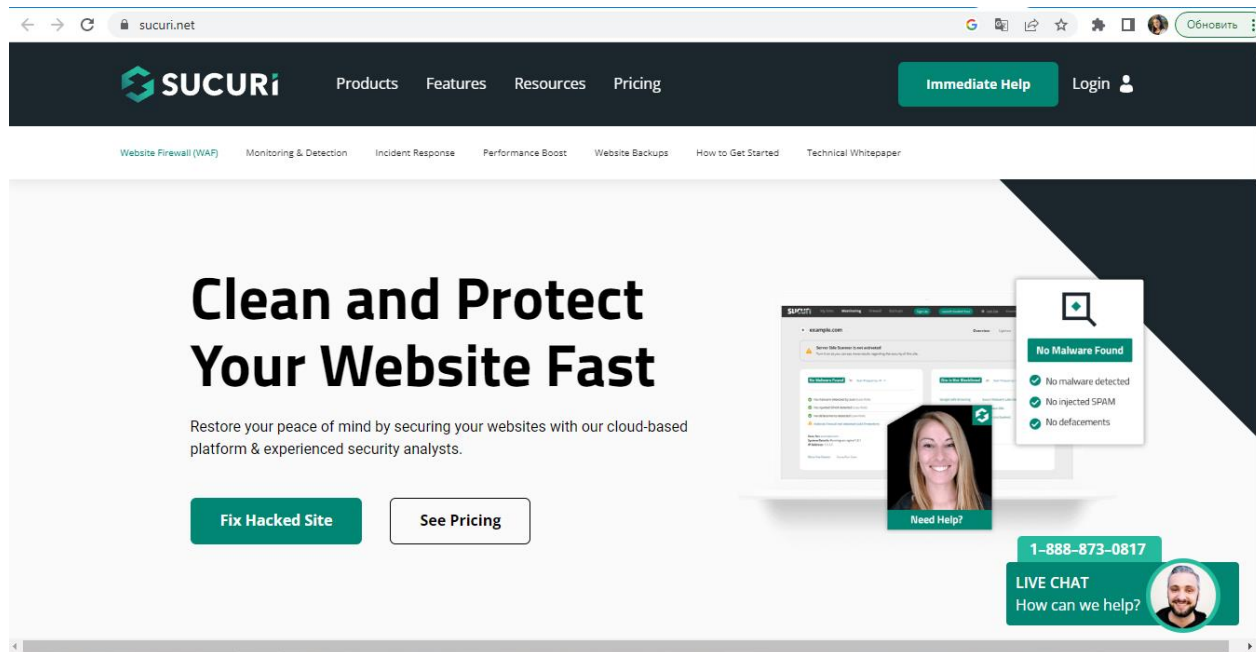


Рисунок 3.37 – Програмне забезпечення Sucuri

Для захисту від Sql-ін'єкцій краще використовувати підготовлені оператори та параметризовані запити. Параметризовані оператори гарантують, що параметри, передані в оператори SQL, обробляються безпечно.

Також важливим аспектом є фільтрування вхідних даних бази: виявлення та фільтрування зловмисного коду із введених користувачами даних. Для запобігання ненавмисних запитів до бази даних необхідно обмежити процедури та код бази даних. Також доцільно обмежити доступ до бази даних. Підтримка оновлень баз даних та моніторинг вхідних даних допоможуть запобігти атакам.

Деякі з інших методів, які використовуються для запобігання SQL-ін'єкції:

1. Хешування пароля.
2. Аутентифікація третьої сторони.
3. Брандмауер веб-додатків.
4. Придбання кращого програмного забезпечення.
5. Своєчасне оновлення пз.

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				65
Змн.	Арк.	№ докум.	Підпис	Дата		

6. Постійний моніторинг бази даних.

Наведений раніше браундмауер Sucuri допоможе підприємству в захисті веб-сайту не тільки від атаки міжсайтового скриптингу, а й від Sql-ін'єкцій.

Загальні способи запобігання атакам брут форсу включають:

- приховування сторінок входу адміністратора та клієнтів шляхом зміни їхніх імен за замовчуванням;
- застосування надійних унікальних паролів — чим довші, тим краще;
- зміна імені користувача "admin" за умовчанням на унікальне;
- обмеження частоти спроб онлайн-входу у ваш обліковий запис за допомогою;
- блокування облікового запису після досягнення заданої кількості спроб входу;
- застосування двофакторної автентифікації, CAPTCHA (наприклад, Datadome CAPTCHA) або інших форм перевірки;
- заборона багаторазових спроб входу з однієї IP-адреси;
- впровадження методів безпечного хешування паролів;
- спостереження за новими чи існуючими обліковими записами користувачів із високим рівнем активності та відсутністю покупок;
- використання автоматизованих можливостей захисту від ботів із аналізом поведінки.

Однак оператори ботів грубої сили зараз розробляють ботів, які можуть імітувати людську поведінку. Вони також використовують дуже велику кількість різних IP-адрес, що ускладнює виявлення ботів. Як наслідок, традиційні засоби безпеки на основі правил, такі як WAF, зазвичай не можуть зупинити атаки грубої сили.

Також існує програмне забезпечення для захисту серверів від атаки грубою силою – Fail2Ban. Метою Fail2Ban є моніторинг журналів загальних служб для виявлення шаблонів у помилках автентифікації. Підозрілі записи в журналі підраховуються, і коли вони досягають певної кількості, Fail2Ban надсилає

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				66
Змн.	Арк.	№ докум.	Підпис	Дата		

сповіщення електронною поштою або блокує IP-адресу зловмисника на певний період часу. Після закінчення періоду блокування IP-адреса автоматично розблокується.

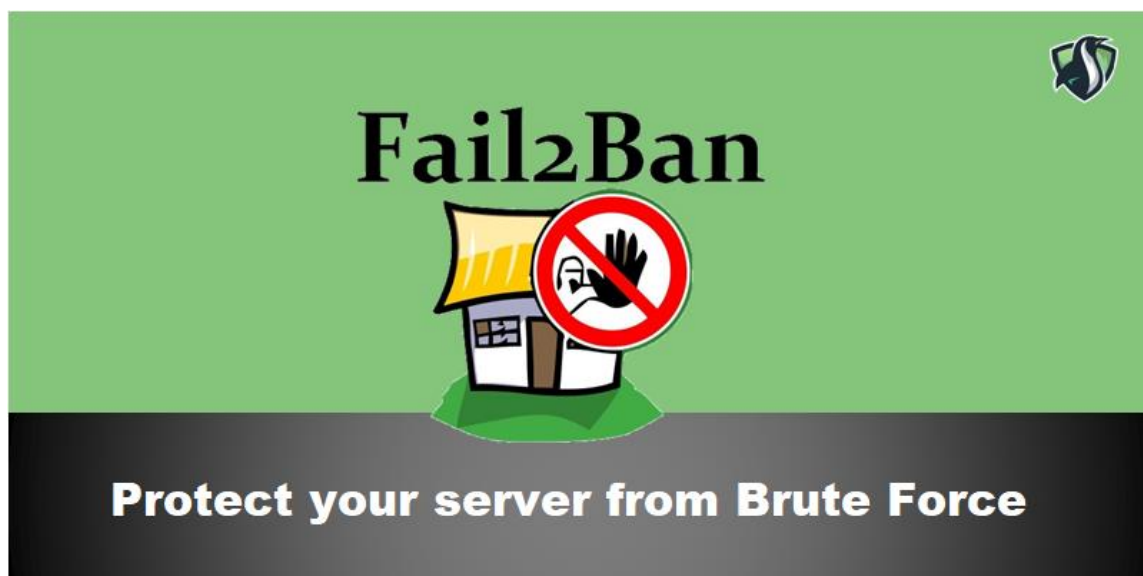


Рисунок 3.38 – Утиліта Fali2Ban

Отже, підводячи підсумки за переліченими раніше вразливостями, найкращий спосіб від них вберегтись це використовувати браундмауери і регулярно проводити сканування веб-сайту для запобігання та виявлення лазівок.

Висновок до третього розділу

В даному розділі було досліджено програмне забезпечення Owasp Zap. Встановлено утиліту на робочу станцію та налаштовано загальні параметри. Також змодельовано такі види атак як SQL-ін'єкція, міжсайтовий скриптинг та атака грубої сили, тобто брут форс. Зокрема, створено звіт за результатами сканування та перелічено рекомендації щодо захисту веб-сайту від цих атак.

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				67
Змн.	Арк.	№ докум.	Підпис	Дата		

ВИСНОВКИ

У даній кваліфікаційній роботі було проведено сканування веб-сайту на вразливості та змодельовано декілька видів атак. Під час роботи досліджено головні загрози безпеці веб-сайтів.

У першому розділі було розглянуто існуючі вразливості веб-сайтів та рекомендації щодо їх уникнення. Також описано актуальність кібербезпеки в сучасному світі та чому сканування веб-додатків на можливі лазівки в безпеці грає велику роль у їх розробці.

В другому розділі було розглянуто наявні веб-додатки для проведення пентестингу. Проаналізовано кожен можливість, які забезпечує певна утиліта. Досліджено та ознайомлено з програмним забезпеченням Owasp Zар, а також описано його переваги та недоліки. Наведено опис атак, котрі можна реалізувати за допомогою даної утиліти та перелічено рекомендації щодо попередження цих атак на веб-сайтах.

В третьому розділі було встановлено програмне забезпечення Owasp Zар та досліджено його графічний інтерфейс. Проведено сканування веб-сайту на вразливості та детально розглянуто кожен із знайдених лазівок. Також було змодельовано певні види атак. Наведено результати сканування та перераховано заходи безпеки як діяти в разі наявності лазівок в безпеці веб-сайту та заходи щоб попередити дані атаки.

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				68
Змн.	Арк.	№ докум.	Підпис	Дата		

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Як працює брут форс. URL: <https://www.makeuseof.com/forced-browsing-how-does-it-work/>
2. Що таке HTTPS. URL: <https://www.cloudflare.com/learning/ssl/what-is-https/>
3. Zap Forced Browse. URL: <https://www.cloudflare.com/learning/ssl/what-is-https/>
4. Захист від брут форс атак. URL: <https://questpro.club/administration/364-brute-force-and-ddos-protection-of-web-site/>
5. Що таке TLS. URL: <https://habr.com/ru/articles/258285/>
6. DoS and DDoS attacks. URL: <https://www.zenarmor.com/docs/network-security-tutorials/dos-vs-ddos-attacks>
7. Як створити звіт Owasp Zap. URL: <https://spy-soft.net/owasp-zap-generating-report/>
8. How to use fuzzing. URL: <https://smeesheady.medium.com/how-to-use-fuzzing-feature-in-owasp-zap-2-9-0-48df6a89bef8>
9. 8 best practice to prevent SQL-injection attacks. URL: <https://smeesheady.medium.com/how-to-use-fuzzing-feature-in-owasp-zap-2-9-0-48df6a89bef8>
10. Importance of cyber security. URL: <https://www.knowledgehut.com/blog/security/importance-of-cyber-security#importance-of-cyber-security-in-the-digital-world>
11. 10 ways to prevent cyber attacks. URL: <https://leaf-it.com/10-ways-prevent-cyber-attacks/>
12. SQL injection attack. URL: <https://brightsec.com/blog/sql-injection-attack/>
13. Exploiting SQL injection: Hands-on example. URL: <https://www.acunetix.com/blog/articles/exploiting-sql-injection-example/>
14. Коди відповідей HTTP. URL: <https://developer.mozilla.org/ru/docs/Web/HTTP/Status>

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				69
Змн.	Арк.	№ докум.	Підпис	Дата		

15. Проксі OWASP. URL: <https://blog.desdelinux.net/owasp-zed-attack-proxy/>
16. Підручник з Owasp Zap. URL: <https://uk.myservername.com/owasp-zap-tutorial-comprehensive-review-owasp-zap-tool>
17. Owasp top 10 vulnerabilities. URL: <https://www.checkpoint.com/cyber-hub/cloud-security/what-is-application-security-appsec/owasp-top-10-vulnerabilities/>
18. Security testing. URL: <https://www.geeksforgeeks.org/software-testing-security-testing-tools/>
19. How to setup Owasp. URL: <https://www.triad.co.uk/news/owasp-zap/>
20. What is fuzz testing. URL: <https://www.code-intelligence.com/what-is-fuzz-testing>
21. How ddos attacks work. URL: <https://usa.kaspersky.com/resource-center/preemptive-safety/how-does-ddos-attack-work>
22. What is clickjacking. URL: <https://www.imperva.com/learn/application-security/clickjacking/>
23. 7 benefits of using owasp zap tool for security testing. URL: <https://www.zehntech.com/7-benefits-of-using-zap-tool-for-security-testing/>
24. Penetration testing tools. URL: <https://phoenixnap.com/blog/best-penetration-testing-tools>

		Ковбан Н.С.			КБ.КР.Б-125-23-ПЗ	Арк.
		Байлюк С.М.				70
Змн.	Арк.	№ докум.	Підпис	Дата		