ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ

КАФЕДРА ІНФОРМАЦІЙНОЇ ТА КІБЕРНЕТИЧНОЇ БЕЗПЕКИ

Пояснювальна записка

до бакалаврської роботи

на тему:

«ДОСЛІДЖЕННЯ ШЛЯХІВ ТА РОЗРОБКА РЕКОМЕНДАЦІЙ ЩОДО ЗАХИСТУ WEB – ДОДАТКІВ ЗА ДОПОМОГОЮ CLOUDFLARE WAF»

Виконав студент 4 курсу, групи БСД-43
спеціальності 125 Кібербезпека
освітньо-професійної програми «Інформаційна та кібернетична безпека»
(шифр і назва спеціальності)

Касян О.В.
(прізвище та ініціали)

Керівник Марченко В.В.
(прізвище та ініціали)

Рецензент
(прізвище та ініціали)

Нормоконтролер
(прізвище та ініціали)

КИЇВ - 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ

Інститут ННІЗІ
Кафедра Інформаційної та кібернетичної безпеки
Ступінь вищої освіти Бакалавр
Спеціальність 125 Кібербезпека
Освітня програма Інформаційна та кібернетична безпека

ЗАТВЕРДЖУЮ
Завідувач кафедри ІКБ
_____ Гайдур Г.І.
«___»_____ 2023 року

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Касяну Олександру Володимировичу
(прізвище, ім'я, по батькові)

1. Тема бакалаврської роботи: Дослідження шляхів та розробка рекомендацій щодо захисту Web - додатків за допомогою Cloudflare WAF

керівник бакалаврської роботи Марченко В.В.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)
затверджені наказом закладу вищої освіти від «24 » лютого 2023 року № 26 .

2. Строк подання студентом бакалаврської роботи 29.05.2023

3. Вихідні дані до бакалаврської роботи
1) Аналіз популярних Web-додатків
2) Оцінка ефективності Cloudflare WAF
3) Розробка рекомендацій щодо використання Cloudflare WAF

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Аналіз сучасних кібератак, спрямованих на Web - додатки
2. Дослідження методів захисту web-додатків
3. Розробка рекомендацій для захисту web-додатків за допомогою Cloudflare WAF

5. Перелік графічного матеріалу
1. Тема бакалаврської роботи.
2. Об’єкт, предмет, мета та наукові завдання дослідження.
3. Інструменти для виявлення вразливостей
4. Інсталювання та налаштування WAF
5. Використання Cloudflare WAF для захисту веб-додатку
6. Висновки за результатами роботи.

6. Дата видачі завдання 24.02.2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ зп	Назва етапів бакалаврської роботи	Строк виконання етапів бакалаврської роботи	Примітка
1.	Аналіз сучасних кібератак, спрямованих на Web - додатки	12.03.2023 р.	виконано
2.	Дослідження методів захисту Web - додатків	29.03.2023 р.	виконано
3.	Інсталювання та налаштування WAF	22.04.2023 р.	виконано
4.	Розробка рекомендацій щодо захисту Web-додатків за допомогою CLOUDFLARE WAF	02.05.2023 р.	виконано
5.	Оформлення результатів дослідження.	09.05.2023 р.	виконано
6.	Підготовка доповіді до захисту.	29.05.2023 р.	виконано

Студент _____ Касян О.В.
(підпис) прізвище та ініціали

Керівник бакалаврської роботи _____ Марченко В.В.
(підпис) прізвище та ініціали

ВІДГУК РЕЦЕНЗЕНТА
на бакалаврську роботу

студента Касяна Олександра Володимировича
на тему: «Дослідження шляхів та розробка рекомендацій щодо захисту Web – додатків за допомогою Cloudflare WAF»

Актуальність: Зі зростанням важливості безпеки веб-додатків у сучасному цифровому середовищі захист веб-додатків стає все більш критичним завданням, оскільки шкідливі атаки, такі як SQL-ін'єкції, крос-сайтовий скриптінг і DDoS-атаки, стають все поширенішими і складнішими. У зв'язку з цим, важливо дослідити різні шляхи захисту веб-додатків та визначити ефективні методи, та інструменти для забезпечення їх безпеки. Cloudflare WAF є одним з таких інструментів, який надає захист від різноманітних загроз шляхом розпізнавання та блокування потенційно шкідливих запитів до веб-додатків.

Позитивні сторони:

1. Актуальність: Тема дослідження є актуальною в контексті зростаючого числа кібератак на веб-додатки та потреби в ефективних захисних заходах.
2. Використання важливого інструменту: Cloudflare WAF є популярним інструментом у сфері кібербезпеки, що робить його важливим об'єктом дослідження.
3. Емпіричне дослідження: Робота передбачає проведення емпіричного дослідження щодо ефективності Cloudflare WAF та його впливу на захист веб-додатків.
4. Розробка рекомендацій: Робота пропонує розробку конкретних рекомендацій для оптимального використання Cloudflare WAF, що може бути корисним для організацій, що прагнуть забезпечити безпеку своїх веб-додатків.

Недоліки:

1. Обмежений фокус: Робота може не охоплювати всі можливі аспекти захисту веб-додатків, а лише акцентуватися на застосуванні Cloudflare WAF.
2. Обмежена генералізація: Рекомендації, розроблені на основі дослідження, можуть не бути універсальними і вимагати додаткового налаштування для конкретних веб-додатків.

Вищезгадані зауваження не впливають на загальну позитивну оцінку бакалаврської роботи.

Висновок: робота заслуговує оцінку «відмінно», а студент Касян Олександр Володимирович гідний присвоєння кваліфікації: бакалавр з кібербезпеки за спеціалізацією Інформаційна та кібернетична безпека.

Якість роботи	
Виконано на замовлення підприємства	
Виконано за тематикою НДР	
Виконано з макетом	
Виконано з застосуванням ЕОМ та МПТ	√
Має практичну цінність	√
Проект-частина комплексної теми	

Підпис рецензента (_____)

Підпис засвідчую
Підпис особи, що засвідчує (_____)

РЕФЕРАТ

Текстова частина бакалаврської роботи: 57 сторінок, 26 рисунків, 23 джерел.

Об'єкт дослідження – Web - додатки.

Предмет дослідження - методи та засоби захисту Web-додатків з використанням Cloudflare WAF.

Мета роботи - розробити рекомендації щодо ефективного захисту Web-додатків з використанням технології Cloudflare WAF.

Методи дослідження - аналіз літератури, вивчення експлуатаційної документації, порівняльний аналіз різних методів захисту Web-додатків, вивчення можливостей та функціоналу Cloudflare WAF.

В роботі приведено огляд сучасної проблематики забезпечення безпеки Web-додатків, описано основні поняття та структури зв'язані з захистом Web-додатків.

Проаналізовано можливості застосування Cloudflare WAF в контексті захисту Web-додатків. Представлено компоненти та особливості розробки системи захисту Web-додатків з використанням Cloudflare WAF.

Досліджено різні методи та алгоритми, які можна використовувати з Cloudflare WAF для ефективного захисту Web-додатків.

Галузь використання - кібербезпека, захист Web-додатків.

ДОСЛІДЖЕННЯ, ШЛЯХИ, РОЗРОБКА, РЕКОМЕНДАЦІЇ, ЗАХИСТ, WEB-ДОДАТКИ, CLOUDFLARE WAF, КІБЕРБЕЗПЕКА, БЕЗПЕКА, АРХІТЕКТУРА, ЗАХИЩЕНІСТЬ.

ABSTRACT

Text part of the bachelor's thesis: 57 pages, 26 figures, 23 references.

Research object – Web - applications.

Research subject - methods and tools for protecting Web applications using Cloudflare WAF.

Objective of the study - to develop recommendations for effective protection of Web applications using Cloudflare WAF technology.

Research methods - literature analysis, study of operational documentation, comparative analysis of different methods for protecting Web applications, exploration of capabilities and functionality of Cloudflare WAF.

The thesis provides an overview of the current issues related to securing Web applications, describes key concepts and structures associated with the protection of Web applications.

The possibilities of applying Cloudflare WAF in the context of Web application security are analyzed. Components and features of developing a Web application protection system using Cloudflare WAF are presented.

Various methods and algorithms that can be used with Cloudflare WAF for effective Web application protection are investigated.

Field of application - cybersecurity, Web application protection.

RESEARCH, APPROACHES, DEVELOPMENT, RECOMMENDATIONS, PROTECTION, WEB APPLICATIONS, CLOUDFLARE WAF, CYBERSECURITY, SECURITY, ARCHITECTURE, SAFETY.

ЗМІСТ

Стор.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

WAF - Web Application Firewall

CDN - Content Delivery Network

API - Application Programming Interface

SQL - Structured Query Language

XSS - Cross-Site Scripting

CSRF - Cross-Site Request Forgery

OWASP - Open Web Application Security Project

DNS - Domain Name System

SSL - Secure Sockets Layer

TLS - Transport Layer Security

HTTP - Hypertext Transfer Protocol

HTTPS - Hypertext Transfer Protocol Secure

IP - Internet Protocol

TCP - Transmission Control Protocol

UDP - User Datagram Protocol

OSI - Open Systems Interconnection

DoS - Denial of Service

DDoS - Distributed Denial of Service

ACL - Access Control List

BPC - Bot Protection Control.

ВСТУП

Актуальність дослідження. У світі сучасної інформаційної технології Web-додатки відіграють дуже важливу роль, що забезпечує потреби користувачів в інтернеті. Проте, збільшення кількості Web-додатків також призводить до збільшення кількості кібератак і злочинних дій, пов'язаних із захопленням контролю над цими додатками. Web-додатки зберігають значну кількість цінної інформації, такої як особисті дані користувачів, фінансові дані, комерційна інформація тощо. Збитки, пов'язані з втратою цих даних або порушенням конфіденційності, можуть бути великими як для користувачів, так і для підприємств.

Відомо, що багато Web-додатків є вразливими до атак, які можуть призвести до викрадення даних, знищення вмісту або виконання небажаних дій, таких як розсилання спаму або крадіжка конфіденційної інформації. Ефективний захист Web-додатків допомагає запобігти несанкціонованому доступу до цієї інформації та зменшити ризик витоку даних.

З метою захисту веб-додатків від кібератак, Web Application Firewall (WAF) є ефективним засобом, що забезпечує контроль та фільтрацію трафіку відвідувачів на Web-сайті. У даній дипломній роботі буде проведено дослідження можливостей застосування Cloudflare WAF, що є однією з провідних технологій захисту веб-додатків, яка надає широкі можливості для виявлення та захисту від кібератак. Завдяки своїй популярності та широкому застосуванню, дослідження ефективного використання Cloudflare WAF стає актуальним завданням.

Об’єкт дослідження – Web - додатки.

Предмет дослідження - методи та засоби захисту Web-додатків з використанням Cloudflare WAF.

Метою роботи є розробка шляхів та рекомендацій щодо ефективного захисту Web-додатків з використанням технології Cloudflare WAF.

Для досягнення заданої мети необхідно виконати низку *завдань*:

- провести аналіз та класифікацію основних видів кібератак, які спрямовані на Web-додатки;

- проаналізувати існуючі методи та інструменти захисту Web-додатків, дослідити їх ефективність, обмеження та проблеми, що виникають при їхньому застосуванні;

- детально проаналізувати технологію Cloudflare WAF, її особливості, функціональні можливості та на основі проведеного аналізу і досліджень розробити практичні рекомендації щодо захисту Web-додатків;

- виконати експериментальне дослідження ефективності застосування Cloudflare WAF на реальному Web-додатку;

Методами дослідження у роботі є аналіз літератури, вивчення експлуатаційної документації, порівняльний аналіз різних методів захисту Web-додатків, вивчення можливостей та функціоналу Cloudflare WAF.

Практичне значення одержаних результатів: дана дипломна робота може бути корисною для IT-фахівців, які зацікавлені в захисті веб-додатків від кібератак. Результати дослідження можуть бути використані для покращення безпеки Web-додатків, а також для підвищення рівня свідомості та розуміння проблеми безпеки Web-додатків серед інших користувачів. Крім того, дана робота може бути корисною для компаній та організацій, які мають власні Web-додатки та зацікавлені в їх безпеці та захисті від кібератак.

1 АНАЛІЗ СУЧАСНИХ КІБЕРАТАК, СПРЯМОВАНИХ НА WEB ДОДАТКИ

1.1. Огляд безпеки веб-додатків

Web-додатки, як правило, містять багато конфіденційної інформації, такої як особисті дані користувачів, банківські реквізити, комерційні та бізнес-дані тощо. Ці дані можуть бути скомпрометовані через кібератаки на Web-додатки.

Кібератаки на Web-додатки - це злочинні дії, спрямовані на отримання неправомірного доступу до веб-додатків та їхніх даних. Ці атаки можуть мати різні форми та типи, зокрема:

SQL-ін'єкції: це атака, яка використовується для отримання неправомірного доступу до бази даних, що використовується в Web-додатку. Атакувач може виконувати довільний код та отримувати доступ до конфіденційної інформації, змінювати або видаляти дані.

Cross-site scripting (XSS): це атака, яка використовується для отримання доступу до конфіденційних даних користувачів, таких як логіни та паролі. Зловмисники вставляють шкідливий код у веб-сторінки, які відображаються на стороні клієнта, що може привести до виконання шкідливих дій на стороні користувача.

Cross-site request forgery (CSRF): це атака, яка використовується для виконання небажаних дій в ім'я користувача, таких як відправка електронних листів, завантаження файлів тощо. Атака зазвичай полягає в тому, що зловмисники використовують недостатньо захищені запити HTTP, щоб надіслати небажані запити на сервер.

File inclusion attacks: це атака, яка використовується для отримання доступу до конфіденійної інформації шляхом включення шкідливого коду до веб-сторінок. Атакувач може отримати доступ до конфіденційних файлів на сервері, що може призвести до витоку інформації.

Для запобігання кібератакам на Web-додатки, необхідно вживати заходів безпеки, таких як:

- Регулярне оновлення програмного забезпечення і плагінів, які використовуються в Web-додатках, для забезпечення захисту від відомих вразливостей.
- Використання сильних паролів та двофакторної автентифікації для захисту від несанкціонованого доступу до даних.
- Захист Web-додатків від SQL-ін’єкцій, XSS та CSRF атак, зокрема, шляхом валідації введення даних та екранизації виведення.
- Використання HTTPS для забезпечення безпеки передачі даних між сервером та клієнтом.
- Моніторинг Web-додатків для виявлення потенційних вразливостей та негайного відгуку на виявлені проблеми.
- Проведення регулярних аудитів безпеки Web-додатків для виявлення та виправлення вразливостей та підвищення загального рівня безпеки.

1.1.1. Види та особливості Web-додатків

Web-додатки - це програмні засоби, які забезпечують взаємодію між користувачем та веб-сервером через веб-браузер. Їх можна класифікувати за різними критеріями, такими як:

За функціональністю:

- Сайти-візитки, які містять інформацію про компанію, її продукти та послуги. Вони зазвичай мають невелику кількість сторінок та низький рівень динаміки.
- Блоги, які надають користувачам можливість публікації своїх думок та інформації про певну тему.

- Електронні комерційні сайти, які дозволяють користувачам придбати товари та послуги через Інтернет. Вони мають складну структуру та взаємодію зі системами платежів та доставки.
- Соціальні мережі, які дозволяють користувачам створювати свій профіль, ділитися інформацією та спілкуватися з іншими користувачами.

За мовою програмування:

PHP: це популярна мова програмування для Web-додатків, яка дозволяє взаємодіяти з базами даних та іншими сервісами.

Java: ця мова програмування використовується для створення великих та складних Web-додатків, які працюють на великій кількості користувачів.

Python: ця мова програмування використовується для швидкого створення прототипів Web-додатків та розробки наукових проектів.

За типом доступу до інформації:

Статичні Web-додатки: це сайти, які містять статичні сторінки, що не містять динамічної інформації.

Динамічні Web-додатки: це сайти, які містять динамічну інформацію, що змінюється залежно від взаємодії користувача з веб-сервером. Їх можна розділити на дві категорії:

Web-додатки з використанням AJAX (асинхронні JavaScript та XML): це додатки, які дозволяють користувачам здійснювати динамічні запити до сервера, не перезавантажуючи сторінку. Наприклад, можна розгорнути список товарів без перезавантаження сторінки.

Web-додатки з використанням серверних мов програмування: це додатки, які використовуються для обробки запитів на сервері та повернення динамічної інформації до користувача. Наприклад, такі додатки можуть дозволяти користувачам реєструватися, створювати профілі, додавати коментарі тощо.

За типом взаємодії з користувачем:

Desktop Web-додатки: це додатки, які запускаються на комп'ютері користувача та взаємодіють з веб-сервером через браузер. Наприклад, такі додатки можуть бути створені на основі JavaScript, HTML та CSS.

Mobile Web-додатки: це додатки, які запускаються на мобільних пристроях та взаємодіють з веб-сервером через мобільний браузер. Наприклад, такі додатки можуть бути створені на основі HTML5, CSS та JavaScript.

Progressive Web-додатки: це додатки, які мають властивості як веб-сайтів, так і нативних додатків. Вони можуть працювати в офлайн-режимі, мати доступ до камери, мікрофону, геолокації та інших функцій мобільного пристрою. Прогресивні Web-додатки можуть бути встановлені на домашній екран мобільного пристрою, щоб дозволяє користувачам отримувати доступ до них безпосередньо зі свого домашнього екрану, як до звичайних мобільних додатків. Такі додатки можуть бути створені з використанням технологій, таких як Service Workers та Web App Manifest, що дозволяють зберігати додаток в кеш-пам'яті мобільного пристрою та встановлювати його як нативний додаток. Це дає можливість працювати з додатком в офлайн-режимі, що дуже зручно для користувачів, які мають обмежений доступ до Інтернету.

Прогресивні Web-додатки також можуть бути більш безпечними, оскільки вони запускаються в контейнері браузера, що дозволяє обмежити доступ до даних користувача та захистити його від зловмисних програм. Крім того, такі додатки можуть бути більш ефективними, оскільки вони не потребують встановлення та оновлення на пристрої користувача, що дозволяє економити ресурси та час.

Узагальнюючи, Web-додатки є важливою складовою Інтернету та використовуються в різних сферах, таких як електронна комерція, соціальні мережі, банківські послуги, транспортні сервіси та інші. Вони забезпечують більш швидкий та зручний доступ до інформації та послуг для користувачів з різних куточків світу.

1.1.2. Важливість безпеки Web – додатків

В даний час Web-додатки стали необхідною частиною повсякденного життя. Вони використовуються для проведення фінансових операцій, зберігання конфіденційної інформації, обміну даними, спілкування, роботи з електронною поштою та іншими функціями. Тому безпека Web-додатків є критично важливою і необхідною[6].

Перш за все, безпека Web-додатків забезпечує захист конфіденційної інформації користувачів. Якщо Web-додатки не захищені, зловмисники можуть отримати доступ до конфіденційних даних, таких як імена користувачів, паролі, кредитні картки тощо. Це може призвести до серйозних наслідків, таких як крадіжка ідентифікаційних даних, фінансових злочинів, порушення конфіденційності та інших проблем.

Крім того, безпека Web-додатків забезпечує захист від атак з боку зловмисників. Вони можуть використовувати різні методи атак, такі як SQL-ін’єкції, Cross-Site Scripting (XSS) та інші. Якщо Web-додатки не захищені, зловмисники можуть використовувати ці атаки для внесення змін у базу даних, викрадення інформації, встановлення шкідливого програмного забезпечення та інших проблем.

Також безпека Web-додатків забезпечує надійність та стабільність роботи додатків. Якщо Web-додатки не захищені, зловмисники можуть використовувати їх для завантаження шкідливого програмного забезпечення, зміни коду та інших дій, які можуть призвести до непередбачуваних помилок та збоїв роботи системи. Це може призвести до втрати даних, падіння системи, перерв у роботі та інших проблем.

Окрім того, безпека Web-додатків є необхідною для забезпечення дотримання законодавства. Законодавство, таке як Загальний регламент про захист персональних даних (GDPR) та інші, вимагає від організацій захищати конфіденційну інформацію своїх користувачів. Якщо Web-додатки не захищені,

організації можуть порушувати ці вимоги та підлягати штрафам та іншим наслідкам.

Отже, безпека Web-додатків є критично важливою для захисту конфіденційної інформації користувачів, захисту від атак з боку зловмисників, забезпечення надійності та стабільності роботи додатків та дотримання законодавства. З цих причин безпека Web-додатків повинна бути у фокусі уваги розробників та користувачів[9].

1.1.3. Основні особливості та вимоги до захисту Web – додатків

Захист Web-додатків є важливою складовою їх розробки та експлуатації. Основні особливості та вимоги до захисту Web-додатків можна описати наступним чином:

Автентифікація та авторизація користувачів. Web-додатки повинні мати механізми автентифікації та авторизації користувачів, щоб забезпечити доступ лише авторизованим користувачам та захистити їхні дані від несанкціонованого доступу.

Захист від атак. Web-додатки повинні бути захищені від різноманітних атак, таких як SQL-ін’єкції, Cross-Site Scripting (XSS), міжсайтові скрипти та інші. Для цього розробники повинні використовувати заходи безпеки, такі як валідація введених даних, екранізація вхідних даних, обмеження прав доступу та інші.

Захист від шкідливих вкладень. Web-додатки повинні бути захищені від шкідливих вкладень, таких як відправка вірусних файлів або інших шкідливих програм на сервер додатку. Для цього розробники повинні використовувати захист від вразливостей, такий як використання безпечних мережевих протоколів, встановлення брандмауера та інші.

Резервне копіювання та відновлення даних. Web-додатки повинні мати механізми резервного копіювання та відновлення даних, щоб забезпечити їхню

безпеку та надійність. Резервне копіювання даних повинно здійснюватися регулярно, а відновлення даних повинно бути можливим у разі необхідності.

Постійне оновлення програмного забезпечення. Web-додатки повинні постійно оновлюватися, щоб бути захищеними від відомих вразливостей та бути відповідними останнім стандартам безпеки. Розробники повинні використовувати оновлення програмного забезпечення та встановлювати патчі безпеки, щоб зменшити ризик вразливостей.

Захист від злому. Web-додатки повинні бути захищені від злому та несанкціонованого доступу. Розробники повинні використовувати заходи безпеки, такі як шифрування даних та трафіку, встановлення паролів та доступу до бази даних, захист від переповнення буфера та інші.

Моніторинг та аудит безпеки. Web-додатки повинні мати механізми моніторингу та аудиту безпеки, щоб відслідковувати потенційні вразливості та незвичайну активність. Це дозволяє розробникам швидко виявляти та вирішувати проблеми безпеки.

Узагальнюючи, захист Web-додатків включає в себе захист від атак, аутентифікацію та авторизацію користувачів, захист від шкідливих вкладень, резервне копіювання та відновлення даних, постійне оновлення програмного забезпечення, захист від злому та моніторинг та аудит безпеки. Розробники повинні враховувати ці вимоги під час розробки та експлуатації Web-додатків, щоб забезпечити їх безпеку та надійність.

1.2. Загрози та вразливості Web-додатків

Загрози та вразливості Web-додатків є серйозною проблемою для розробників та користувачів, оскільки вони можуть призвести до витоку конфіденційної інформації, крадіжки особистих даних, або навіть до повного зруйнування додатку.

Основні загрози та вразливості Web-додатків можна описати наступним чином:

SQL-ін'єкції: це одна з найбільш поширених вразливостей Web-додатків, яка дозволяє злочинцям виконувати зловмисні дії на базі даних додатка, такі як видалення даних або викрадення конфіденційної інформації.

Cross-Site Scripting (XSS): ця вразливість дозволяє злочинцям вводити зловмисний код на сторінках Web-додатка, який може бути використаний для викрадення конфіденційної інформації або для запуску інших зловмисних дій[8].

Міжсайтові скрипти: ця вразливість дозволяє злочинцям виконувати скрипти на сторінках Web-додатка, які можуть бути використані для викрадення конфіденційної інформації або для запуску інших зловмисних дій.

Несанкціонований доступ: ця загроза може призвести до крадіжки особистих даних або конфіденційної інформації шляхом злому системи авторизації та автентифікації користувачів.

Безпека мережі: неправильна настройка мережевої інфраструктури може призвести до злому системи та викрадення конфіденційної інформації.

Відсутність шифрування: якщо Web-додаток не використовує шифрування, то конфіденційна інформація, яку передає користувач, може бути викрадена з мережі та використана зловмисниками.

Недостатня перевірка вхідних даних: ця вразливість дозволяє зловмисникам ввести в додаток зловмисний код або виконати зловмисну дію, що може призвести до крадіжки інформації або до виконання небажаних дій в системі.

Недостатня перевірка вихідних даних: ця вразливість дозволяє зловмисникам модифікувати вихідні дані додатка, що може призвести до небезпечних ситуацій, таких як XSS-атаки.

Недостатній захист від переповнення буфера: ця вразливість дозволяє зловмисникам вводити занадто велику кількість даних у буфер, що може призвести до переповнення буфера та виконання зловмисних дій.

Використання застарілих технологій та бібліотек: застарілі технології та бібліотеки можуть містити вразливості, які можуть бути використані зловмисниками для викрадення конфіденційної інформації або запуску інших зловмисних дій.

Досить часто зловмисники використовують комбінацію різних вразливостей та загроз, щоб отримати доступ до конфіденційної інформації, тому важливо розуміти та знати, які загрози та вразливості існують, щоб запобігти їх використанню та забезпечити безпеку Web-додатків[23].

1.2.1. Приклади вразливостей та загроз Web-додатків

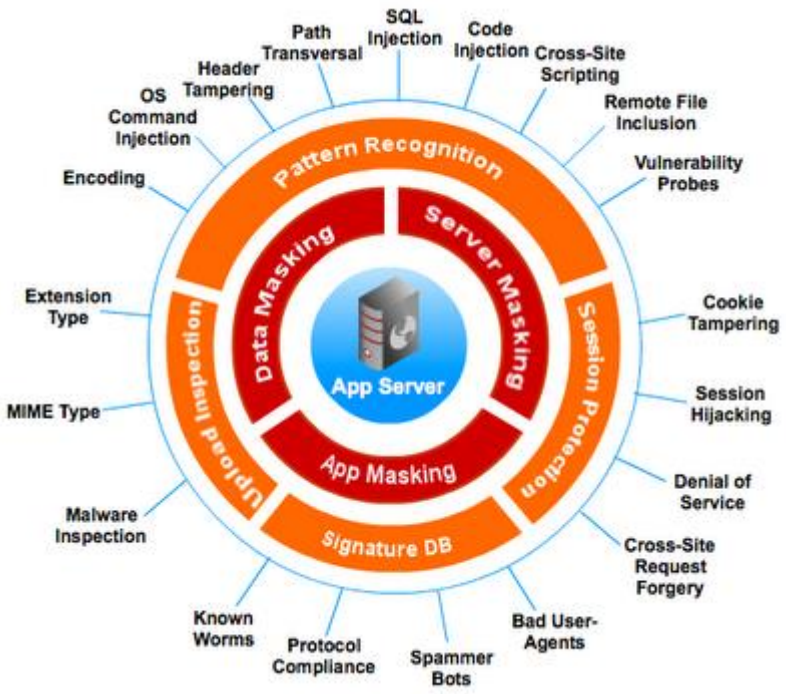

Рис. 1.1. Основні види атак на Web-додаток

OWASP (Open Web Application Security Project) - це список найбільш поширених вразливостей та загроз безпеці Web-додатків, який щорічно публікується OWASP[1]. Нижче наведено детальні описи десяти найбільш поширених вразливостей та загроз Web-додатків, які включені в цей список:

Injection (ін'єкції) - це вразливість, що дозволяє злочинцям вставляти зловмисний код в вхідні дані, які передаються в базу даних Web-додатка, з метою виконання шкідливого коду або отримання конфіденційної інформації.

Broken Authentication and Session Management (порушення автентифікації та управління сесіями) - ця вразливість дозволяє злочинцям отримати доступ до даних користувача або використовувати його обліковий запис з метою викрадення конфіденційної інформації або виконання зловмисних дій.

Cross-Site Scripting (XSS) - ця вразливість дозволяє злочинцям вводити зловмисний код на сторінках Web-додатка, який може бути використаний для викрадення конфіденційної інформації або для запуску інших зловмисних дій.

Broken Access Control (порушення контролю доступу) - ця вразливість дозволяє злочинцям отримувати доступ до конфіденційної інформації або змінювати даний Web-додаток, якщо не були встановлені відповідні механізми контролю доступу.

Security Misconfiguration (неправильна конфігурація безпеки) - ця вразливість дозволяє злочинцям виконувати зловмисні дії, якщо налаштування безпеки Web-додатка були здійснені неправильно.

Insecure Cryptographic Storage (ненадійне зберігання криптографічних даних) - ця вразливість дозволяє злочинцям отримувати доступ до конфіденційної інформації, яка була збережена у ненадійний спосіб, наприклад, з використанням незахищених алгоритмів шифрування або зберігання паролів у відкритому вигляді.

Insufficient Logging and Monitoring (недостатнє логування та моніторинг) - ця вразливість дозволяє злочинцям виконувати зловмисні дії, не залишаючи слідів у логах Web-додатка, що ускладнює виявлення атак і відновлення системи після відновлення.

Insecure Communications (небезпечне зв'язку) - ця вразливість дозволяє злочинцям перехоплювати та змінювати дані, які передаються між користувачем

та Web-додатком, що може призвести до викрадення конфіденційної інформації або виконання зловмисних дій.

Using Components with Known Vulnerabilities (використання компонентів з відомими вразливостями) - ця вразливість дозволяє злочинцям використовувати вразливості, які були виявлені у сторонніх компонентах, які використовуються в Web-додатку.

Insufficient Authorization (недостатній контроль авторизації) - ця вразливість дозволяє злочинцям отримувати доступ до конфіденційної інформації або змінювати дані в Web-додатку, якщо механізми авторизації не були встановлені правильно[22].

Cross-Site Request Forgery (CSRF) (підробка міжсайтових запитів) - ця вразливість дозволяє злочинцям виконувати зловмисні дії в ім'я авторизованого користувача, який здійснює запит на веб-сайті.

SQL Injection (внедрення SQL запитів) - ця вразливість дозволяє злочинцям виконувати зловмисні SQL запити, які можуть викрадати конфіденційну інформацію або змінювати дані в базі даних.

Broken Access Control (недієвий контроль доступу) - ця вразливість дозволяє злочинцям отримувати доступ до обмеженої інформації або функціональності, якщо механізми контролю доступу не були належним чином налаштовані.

Security Misconfiguration (неправильна конфігурація безпеки) - ця вразливість може створити кілька точок входу для злочинців, які можуть використовувати неправильну конфігурацію для зловживання або отримання доступу до конфіденційної інформації.

Unvalidated Input (невалідований ввід) - ця вразливість дозволяє злочинцям використовувати невалідований ввід, такий як некоректно введені дані або неправильний формат, для виконання зловмисних дій, таких як внедрення вредоносного коду або викрадення конфіденційної інформації.

Security through Obscurity (безпека через таємниці) - ця практика дозволяє злочинцям легко знайти слабкі місця в системі, що приховуються від розгляду.

Вона може призвести до створення ненадійних механізмів безпеки, які можуть бути легко взломані злочинцями[4].

Щоб захистити свій Web-додаток від цих вразливостей, необхідно розробити та належним чином налаштувати відповідні механізми захисту, такі як валідація вводу, належний контроль доступу та моніторинг. Крім того, необхідно регулярно проводити аудит безпеки для виявлення можливих вразливостей та вирішення їх проблем, а також оновлювати систему та програмне забезпечення з метою запобігання використання вразливостей, які вже були виправлені розробниками. Крім того, можна використовувати спеціальні інструменти для виявлення вразливостей та атак на систему, такі як відлагоджувачі та фільтри ввідних даних. Загальна мета полягає в тому, щоб забезпечити високий рівень безпеки для користувачів та даних веб-додатка.

1.2.2. Причини та наслідки виникнення вразливостей та загроз Web-додатків

Причини виникнення вразливостей та загроз Web-додатків можуть бути різноманітними. Деякі з них можуть бути пов'язані з розробкою додатку, тоді як інші можуть виникати на етапі експлуатації. Нижче перераховані деякі з найпоширеніших причин виникнення вразливостей та загроз Web-додатків:

- *Недостатня валідація вводу* - невалідований ввід може призвести до внедрення вредоносного коду на стороні сервера, що може призвести до втрати конфіденційної інформації або зловживання функціональністю додатку.
- *Недостатня автентифікація та авторизація* - неправильно налаштовані механізми автентифікації та авторизації можуть призвести до витоку конфіденційної інформації або зловживання правами доступу.
- *Недієвий контроль доступу* - неправильно налаштовані механізми контролю доступу можуть дозволити несанкціонованому користувачеві

отримати доступ до конфіденційної інформації або функціональності додатку.

- *Відсутність аудиту безпеки* - відсутність системи моніторингу та аудиту може призвести до виявлення вразливостей і загроз тільки після того, як вони вже використані злочинцем.
- *Неправильна конфігурація безпеки* - неправильно налаштовані механізми безпеки можуть створювати додаткові точки входу для злочинців, що може призвести до зловживання та витоку інформації.
- *Недостатній захист від вразливостей переповнення буфера* - відсутність захисту від вразливостей переповнення буфера може призвести до виконання небезпечного коду на стороні сервера, що може призвести до злому системи та викрадення конфіденційної інформації.
- *Недостатній захист від атак внедрення коду* - неправильно налаштовані механізми захисту від атак внедрення коду можуть дозволити злочинцеві внести вредоносний код на стороні клієнта або сервера, що може призвести до викрадення даних та виконання шкідливих дій.
- *Недостатній захист від атак міжсайтового скриптування (XSS)* - відсутність захисту від атак XSS може дозволити злочинцю внести скрипт на сторінку веб-додатку, що може призвести до викрадення конфіденційної інформації або зміни поведінки додатку.
- *Недостатній захист від атак перехоплення сесій* - неправильно налаштовані механізми захисту від атак перехоплення сесій можуть призвести до витоку даних, зміни даних та виконання дій в ім'я авторизованого користувача.
- *Відсутність захисту від атак перехоплення трафіку* - відсутність захисту від атак перехоплення трафіку може дозволити злочинцю викрадати та модифікувати дані, що передаються між сервером та клієнтом.
- *Відсутність захисту від атак на виконання коду з використанням вразливостей віртуальних машин* - відсутність захисту від таких атак

може призвести до виконання вредоносного коду на рівні віртуальної машини, що може дозволити злочинцю отримати доступ до конфіденційної інформації та виконувати шкідливі дії.

- *Недостатній захист від атак викрадення ідентифікаторів сесій* - неправильно налаштовані механізми захисту від атак викрадення ідентифікаторів сесій можуть дозволити злочинцю отримати доступ до авторизованого облікового запису, виконувати дії в ім'я користувача та викрадати конфіденційну інформацію.
- *Недостатня перевірка вхідних даних* - неправильна перевірка вхідних даних може дозволити злочинцю ввести та виконати вредоносний код на стороні сервера, що може призвести до злому системи та викрадення конфіденційної інформації.
- *Відсутність моніторингу системи на виявлення аномальної поведінки* - відсутність системи моніторингу може призвести до того, що злочинець може отримати доступ до системи та виконувати шкідливі дії, не викликаючи підозр.
- *Недостатній захист від атак фішингу* - відсутність захисту від атак фішингу може призвести до того, що злочинець може отримати доступ до облікових даних користувача та конфіденційної інформації шляхом використання соціальної інженерії.

Усі ці проблеми можуть призвести до серйозних наслідків, тому необхідно приділяти належну увагу безпеці веб-додатків та розробляти їх з урахуванням захисту від різноманітних атак[10].

1.3. Методи захисту Web-додатків

Методи захисту Web-додатків є важливою складовою для забезпечення безпеки веб-додатків. Захист веб-додатків може включати різні техніки та підходи, такі як

шифрування, перевірка введених даних, моніторинг активності, контроль доступу та інші. У дипломній роботі досліджені різні методи захисту веб-додатків та їх ефективність.

1.3.1. Стандарти безпеки Web-додатків

Стандарти безпеки Web-додатків є набором рекомендацій та правил, які визначають мінімальні вимоги до безпеки Web-додатків. Дотримання цих стандартів забезпечує високий рівень захисту Web-додатків від різноманітних атак та загроз.

Захист Web-додатків - це складний процес, і існує кілька стандартів безпеки, які можуть допомогти організаціям захистити свої додатки від різних видів атак[2].

Одним з найпопулярніших стандартів є *WASP* (Web Application Security Project), який містить рекомендації та кращі практики для захисту Web-додатків. WASP пропонує широкий спектр заходів, що включають у себе перевірку на вразливості, захист від атак, захист інфраструктури, керування доступом, криптографічний захист та інші. Однак, застосування цього стандарту не гарантує 100% безпеки.

Інший стандарт - *PCI DSS* (Payment Card Industry Data Security Standard), який спеціально створений для захисту кредитної інформації від злочинних дій. Він містить вимоги до зберігання, обробки та передачі кредитних даних. Хоча цей стандарт не орієнтований на захист Web-додатків, він забезпечує додатковий рівень захисту від злочинних дій[20].

Третій стандарт - *ISO/IEC 27001* - містить загальні вимоги до систем управління інформаційною безпекою. Цей стандарт охоплює не тільки Web-додатки, але й інші системи та процеси, що пов'язані з інформаційною безпекою. Його застосування допомагає забезпечити захист інформації від внутрішніх та зовнішніх загроз.

Ефективність цих стандартів залежить від того, наскільки добре вони застосовуються в конкретній організації. Крім того, ефективність може бути залежна від того, наскільки добре розуміються вимоги та рекомендації. На практиці, ефективність захисту Web-додатків залежить від різноманітних факторів, таких як обсяг та складність додатку, види вразливостей, які можуть бути використані для атак, а також кваліфікації та досвіду фахівців, що відповідають за безпеку. Однак, відповідне застосування стандартів може допомогти зменшити кількість можливих вразливостей та підвищити рівень безпеки.

Зокрема, застосування WASP може допомогти організаціям зменшити ризики вразливостей, а PCI DSS допоможе забезпечити захист кредитної інформації. ISO/IEC 27001 забезпечує загальні вимоги до систем управління інформаційною безпекою, які можуть бути застосовані для різних типів систем та процесів, що пов'язані з інформаційною безпекою.

Однак, важливо зазначити, що застосування стандартів не гарантує 100% безпеки. Вони повинні бути доповнені додатковими заходами, такими як аудит безпеки, моніторинг та відлагодження системи. Крім того, ефективність застосування стандартів може бути знижена, якщо вони не враховують специфіки конкретної організації та її ділової моделі. Тому, важливо ретельно аналізувати потреби та вимоги організації при виборі та застосуванні стандартів безпеки для Web-додатків.

1.3.2. Захист від вразливостей

Захист від вразливостей - це один з ключових аспектів безпеки Web-додатків. Цей процес полягає в ідентифікації та усуненні вразливостей, що можуть бути використані зловмисниками для атак на систему. Загальною практикою є використання різноманітних інструментів для сканування веб-

додатків на наявність вразливостей, таких як SQL-ін'єкції, переповнення буфера, XSS-атаки та інші[15].

Для захисту від вразливостей можна використовувати такі методи, як регулярне оновлення програмного забезпечення, використання сильних паролів, обмеження доступу до конфіденційної інформації, використання захисту від вторгнень та інших заходів. Крім того, важливо розробляти Web-додатки, використовуючи кращі практики розробки з метою запобігання вразливостям.

Захист від вразливостей є постійним процесом, оскільки зловмисники намагаються постійно знайти нові способи атак на системи. Тому, для забезпечення найвищого рівня безпеки, організації повинні постійно оновлювати свої методи та заходи для захисту від вразливостей.

1.3.3. Тестування безпеки Web-додатків

Тестування безпеки Web-додатків - це процес перевірки наявності вразливостей та ризиків безпеки в додатках, що можуть бути використані зловмисниками для отримання несанкціонованого доступу до інформації чи завдання шкоди системі.

Одним із найважливіших етапів розробки Web-додатків є проведення тестування безпеки, яке дозволяє виявити потенційні проблеми та вразливості, що можуть бути використані зловмисниками для атаки на додаток та інфраструктуру[11].

Тестування безпеки можна проводити як вручну, так і з використанням спеціалізованих інструментів. Тести вручну полягають у здійсненні спроб злому додатку та виявленні вразливостей шляхом аналізу вхідних та вихідних даних, а також поведінки системи при різних вхідних параметрах.

Спеціалізовані інструменти для тестування безпеки дозволяють проводити автоматизоване тестування, яке значно зменшує час та зусилля, необхідні для виявлення вразливостей. Ці інструменти можуть виявляти різноманітні

вразливості, такі як SQL-ін'єкції, переповнення буфера, перехоплення сесії та багато інших.

Тестування безпеки Web-додатків повинно проводитись як під час розробки, так і після випуску продукту в експлуатацію, для забезпечення постійної безпеки системи та запобігання можливим атакам. Крім того, після проведення тестування безпеки рекомендується вжити заходів для усунення виявлених вразливостей та вдосконалення захисту додатка.

Висновки до розділу

В результаті аналізу безпеки Web – додатків можна зробити висновок, що вони є підвищено вразливими до кібератак та загроз. Це обумовлено особливостями Web-технологій та можливостями, які вони надають. Виявлені вразливості можуть бути використані для різноманітних атак, таких як зламання паролів, крадіжка даних, введення змін у веб-сторінки, запуск шкідливих скриптів та інших видів кібератак.

В рамках дослідження було розглянуто основні види вразливостей та загроз, які можуть виникнути у Web – додатках, а також методи їх запобігання та захисту. До основних методів захисту від вразливостей можна віднести:

- захист від SQL-ін'єкцій,
- захист від міжсайтового скриптінгу,
- захист від переповнення буфера,
- захист від вразливостей, пов'язаних з автентифікацією та авторизацією.

Для забезпечення безпеки Web – додатків також важливо використовувати стандарти та рекомендації з безпеки, такі як OWASP, а також проводити регулярне тестування безпеки для виявлення можливих вразливостей та загроз.

Отже, забезпечення безпеки Web – додатків є важливою задачею, оскільки вони забезпечують доступ до важливої інформації та послуг. Для досягнення

високого рівня безпеки необхідно використовувати заходи захисту та стандарти безпеки, проводити регулярне тестування безпеки та відслідковувати нові вразливості та загрози[17].

2 ДОСЛІДЖЕННЯ МЕТОДІВ ЗАХИСТУ WEB – ДОДАТКІВ

2.1. Структура web-додатків

Структура web-додатків може відрізнятися в залежності від їх призначення, функціоналу та технологій, які використовуються[19]. Проте, в загальному випадку, веб-додатки можна поділити на такі компоненти:

Клієнтська сторона (Front-end) - це та частина додатку, яка відповідає за відображення інформації на стороні користувача. Вона складається з HTML-сторінок, CSS-стилів та JavaScript-скриптів. Клієнтська сторона дозволяє користувачам взаємодіяти з додатком, вводити дані, відправляти запити до сервера та отримувати результати.

Серверна сторона (Back-end) - це та частина додатку, яка відповідає за обробку запитів від клієнтів та надання відповіді. Серверна сторона складається зі скриптів та програм, написаних на мовах програмування, таких як PHP, Python, Ruby тощо. Вона також забезпечує зберігання даних та зв'язок з базою даних.

База даних - це та частина додатку, де зберігаються всі дані, необхідні для його роботи. Це можуть бути дані користувачів, товарів, послуг, замовлень тощо. База даних може бути реляційною або нереляційною, залежно від типу даних, які необхідно зберігати та обробляти.

API - це та частина додатку, яка дозволяє взаємодіяти з ним за допомогою інших додатків або сервісів. API надає доступ до функціоналу додатку та дозволяє інтегрувати його з іншими системами.

Конфігураційні файли та ресурси - це ті файли, які містять налаштування та параметри роботи дод атку, а також різноманітні ресурси, такі як зображення, аудіо- та відеофайли, які використовуються на сторінках додатку. Конфігураційні файли зазвичай зберігаються окремо від коду додатку та містять такі налаштування, як параметри зв'язку з базою даних, налаштування безпеки, налаштування електронної пошти тощо.

Управління версіями - це та система, яка використовується для зберігання та керування версіями коду додатку. Управління версіями дозволяє зберігати історію змін коду, відстежувати проблеми та відновлювати попередні версії додатку у випадку потреби.

Логіка бізнес-процесів - це та частина додатку, яка відповідає за реалізацію бізнес-логіки, тобто обробку даних та прийняття рішень на основі цих даних. Логіка бізнес-процесів використовується для розробки алгоритмів, які вирішують конкретні завдання додатку, наприклад, розрахунок ціни на товар або розсилка електронної пошти користувачам.

Управління сесіями - це та система, яка використовується для зберігання даних про користувачів, які взаємодіють з додатком, та їхніх дій на сайті. Управління сесіями дозволяє забезпечити безпеку додатку та користувачів, а також забезпечує зручну роботу з системою авторизації та аутентифікації користувачів.

Управління доступом - це та система, яка використовується для керування правами доступу користувачів до різних частин додатку. Управління доступом дозволяє обмежувати доступ користувачів до конфіденційної та приватної інформації, яка зберігається в додатку, а також обмежувати можливості користувачів у виконанні певних дій в додатку. Наприклад, можна надати доступ до перегляду списку товарів, але обмежити можливість їх редагування чи видалення.

Моніторинг та аналіз даних - це та система, яка використовується для збору, обробки та аналізу даних, що збираються в додатку. Моніторинг та аналіз даних дозволяє виявляти проблеми в додатку, аналізувати поведінку користувачів та забезпечувати оптимальну роботу додатку.

Тестування додатку - це процес перевірки та валідації роботи додатку перед його випуском. Тестування додатку може бути ручним або автоматизованим та дозволяє виявляти проблеми та помилки в роботі додатку та забезпечувати його стабільну та оптимальну роботу[5].

Конфігурація та розгортання - це та частина розробки додатку, яка відповідає за налаштування та розгортання додатку на сервері або хостингу. Конфігурація та розгортання дозволяє забезпечити оптимальну та стабільну роботу додатку на сервері та забезпечити зручний доступ користувачів до додатку через мережу Інтернет.

У даному пункті було наведено основні компоненти веб-додатку, такі як архітектура клієнт-сервер, база даних, конфігураційні файли, управління версіями, логіка бізнес-процесів, управління сесіями та управління доступом. Кожен з цих компонентів має свої важливі функції, які допомагають створювати ефективний та безпечний веб-додаток. Управління версіями дозволяє відстежувати та зберігати історію змін коду додатку, що допомагає розробникам відновлювати попередні версії у разі необхідності. Логіка бізнес-процесів забезпечує обробку даних та прийняття рішень на основі цих даних, а управління сесіями та доступом дозволяє забезпечувати безпеку додатку та користувачів.

2.1.1. Пошук типових вразливостей web-додатків

Пошук типових вразливостей веб-додатків є однією з ключових складових безпеки програмного забезпечення, оскільки веб-додатки стали дуже поширеними і важливими для бізнесу та користувачів в Інтернеті. Ця задача полягає в пошуку вразливостей, які можуть бути використані зловмисниками для злому додатку або викрадення конфіденційної інформації користувачів.

Для пошуку типових вразливостей веб-додатків використовуються спеціальні інструменти, такі як сканери вразливостей. Сканери вразливостей веб-додатків аналізують вхідні дані та взаємодію з додатком, щоб виявити потенційні вразливості. Такі інструменти можуть автоматизувати процес виявлення вразливостей та зменшити кількість помилок, які можуть бути допущені людиною.

Серед типових вразливостей, які можуть бути виявлені сканерами вразливостей, можна виділити наступні:

- *Вразливості SQL-ін'єкції*, які дозволяють зловмисникам отримати доступ до бази даних додатку і змінювати, видаляти або викрадати дані.
- *Вразливості XSS (Cross-Site Scripting),* які дозволяють зловмисникам вставляти в шкідливий код на сторінці веб-додатка, що може привести до викрадення сесійних ключів або конфіденційної інформації користувачів.
- *Вразливості CSRF (Cross-Site Request Forgery),* які дозволяють зловмисникам змінювати дані або виконувати дії на сторінці веб-додатка від імені користувача, без його згоди.

Вразливості небезпечного введення файлів, які дозволяють зловмисникам завантажувати на сервер шкідливі файли та виконувати на них зловмисний код[7].

Також, до типових вразливостей веб-додатків можна віднести вразливості, пов'язані з недостатньою аутентифікацією та авторизацією, недостатнім контролем доступу, відсутністю захисту від витоку інформації, відсутністю захисту від перехоплення даних, вразливості, пов'язані з небезпечними налаштуваннями сервера та інші.

Отже, пошук типових вразливостей веб-додатків є важливою складовою безпеки програмного забезпечення. Використання спеціальних інструментів, таких як сканери вразливостей, дозволяє автоматизувати процес виявлення вразливостей та зменшити кількість помилок, що можуть бути допущені людиною. Розуміння типових вразливостей допомагає програмістам та тестувальникам бути більш обережними та попереджувати можливі атаки на веб-додатки.

2.1.2. Аналіз інструментів для виявлення вразливостей

Інструменти для виявлення вразливостей є важливими для забезпечення безпеки веб-додатків. Нижче розглянемо кілька найпопулярніших інструментів для виявлення вразливостей веб-додатків та їх переваги та недоліки[14].

Burp Suite

Burp Suite - це один з найбільш популярних інструментів для тестування безпеки веб-додатків. Цей інструмент дозволяє виявляти різноманітні вразливості веб-додатків, такі як SQL-ін'єкції, XSS, CSRF, вразливості небезпечного введення файлів та інші.

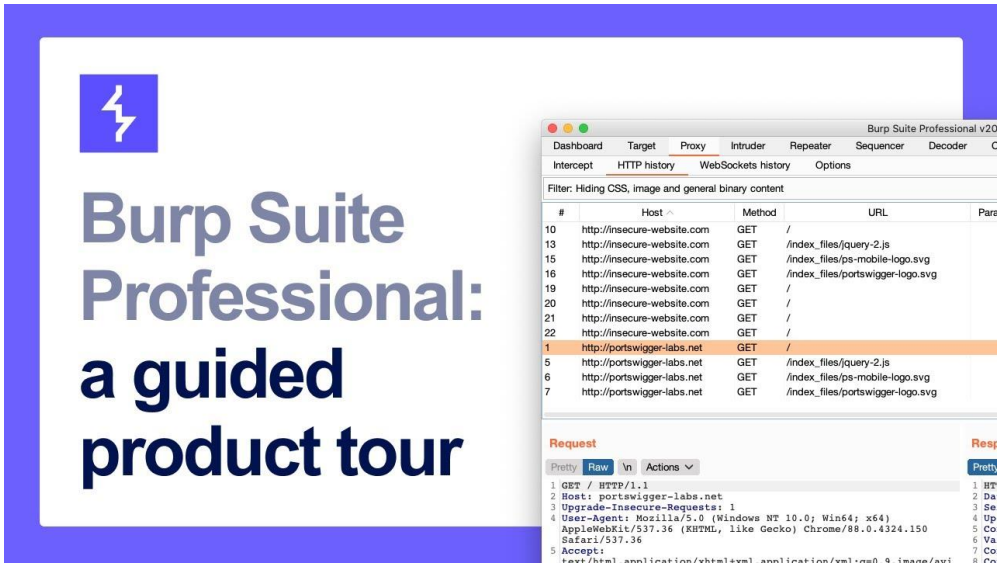

Рис. 2.1. Burp Suite

Переваги:

- Burp Suite має дуже простий та інтуїтивний інтерфейс, що робить його досить легким у використанні.
- Інструмент має можливість автоматизації деяких завдань, що дозволяє значно зменшити час, витрачений на виявлення вразливостей.
- Burp Suite дозволяє перехоплювати трафік, що дозволяє виявляти вразливості, які не можуть бути виявлені іншими інструментами.

Недоліки:

- Burp Suite має досить високу ціну, що робить його недоступним для більшості користувачів.
- Інструмент потребує певної кваліфікації для ефективного використання, тому його використання може бути складним для новачків.

OWASP ZAP

OWASP ZAP - це інструмент з відкритим вихідним кодом для тестування безпеки веб-додатків. Цей інструмент надає широкий набір функцій для виявлення різноманітних вразливостей веб-додатків.

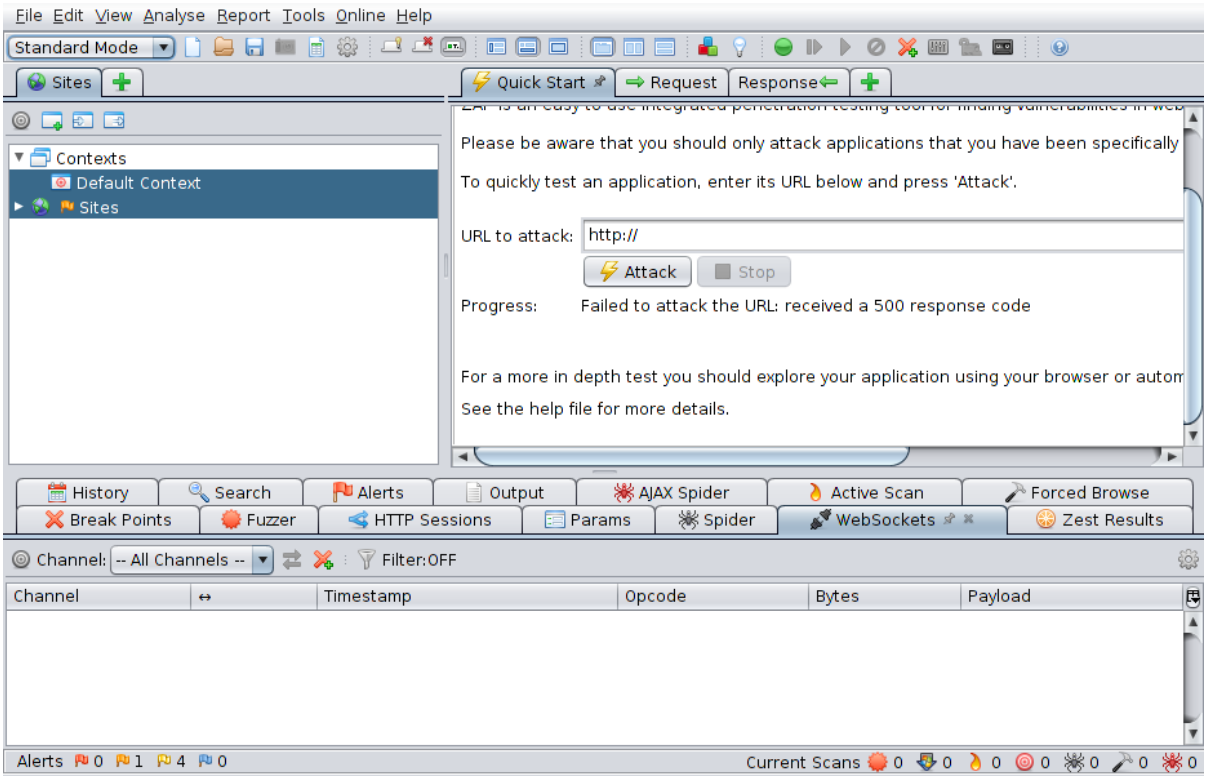

Рис. 2.2. Owasp Zap

Переваги:

- OWASP ZAP має відкритий вихідний код, що дозволяє розробникам налаштовувати та розширюватиінструмент на свій розсуд, що забезпечує більш гнучке використання та можливості.

- Інструмент має можливість сканування веб-додатків на предмет вразливостей як з певними настройками, так і з використанням автоматичного режиму сканування.
- OWASP ZAP має багато корисних функцій, які дозволяють виявляти різноманітні вразливості веб-додатків, такі як XSS, SQL-ін'єкції, CSRF, вразливості кешування та інші.

Недоліки:

- OWASP ZAP може бути менш інтуїтивним у використанні порівняно з Burp Suite, що може затримати процес тестування веб-додатків.
- Інструмент може бути менш стабільним порівняно з Burp Suite та іншими платними інструментами, що може призвести до помилок у виявленні вразливостей.

Nmap

Nmap - це інструмент для сканування портів та виявлення вразливостей мережевих пристроїв та веб-додатків. Інструмент має велику кількість настроювальних параметрів, які дозволяють виконувати різноманітні сканування та тестування безпеки.

Рис. 2.3. Утиліта Nmap

Переваги:

- Nmap є дуже потужним інструментом з великою кількістю функцій для виявлення вразливостей веб-додатків та мережевих пристроїв.
- Інструмент може працювати на різних операційних системах, що забезпечує більш гнучке використання.
- Nmap має велику спільноту користувачів, що надає доступ до різноманітних джерел знань та підтримки.

Недоліки:

- Nmap може бути складним у використанні для новачків, оскільки має велику кількість настроювальних параметрів та опціоналів.
- Інструмент може бути менш точним в порівнянні з платними аналогами, оскільки не має такої ж розширеної функціональності та бази даних вразливостей.
- Nmap може бути менш ефективним для виявлення вразливостей на веб-сайтах порівняно з іншими спеціалізованими інструментами.

У цьому аналізі були розглянуті два інструменти для тестування безпеки веб-додатків та мережевих пристроїв - *OWASP ZAP* та *Nmap*. Обидва інструменти мають свої переваги та недоліки.

OWASP ZAP має багато корисних функцій, які дозволяють виявляти різноманітні вразливості веб-додатків. Проте, інструмент може бути менш інтуїтивним у використанні та менш стабільним порівняно з платними аналогами, що може призвести до помилок у виявленні вразливостей.

Nmap є потужним інструментом з великою кількістю функцій для виявлення вразливостей веб-додатків та мережевих пристроїв. Інструмент може працювати на різних операційних системах та має велику спільноту користувачів, що надає доступ до різноманітних джерел знань та підтримки. Однак, Nmap може бути

складним у використанні для новачків через велику кількість настроювальних параметрів та опцій.

Висновуючи, обидва інструменти мають свої переваги та недоліки, і вибір між ними залежить від конкретних потреб користувача та рівня його/її досвіду у тестуванні безпеки.

2.1.3. Аналіз методів захисту Web – додатків

Web-додатки є одним з найбільш вразливих компонентів інтернет-екосистеми, оскільки вони зазвичай містять значну кількість незахищеної інформації та вразливостей, які можуть бути використані для атак. Нижче розглянуті деякі з методів захисту, які можуть бути використані для захисту веб-додатків від різних видів атак.

Використання SSL / TLS: Цей метод забезпечує шифрування даних, що передаються між браузером користувача та веб-сервером. Використання SSL / TLS знижує ризик перехоплення конфіденційної інформації, такої як паролі та фінансова інформація.

Валідація даних: Цей метод дозволяє перевірити, чи є вхідні дані коректними, перш ніж їх зберігати чи використовувати в інших операціях. Валідація даних знижує ризик SQL-ін'єкцій та інших подібних атак.

Захист від CSRF: Цей метод полягає в тому, щоб створити та перевірити токени для кожного запиту на сервер. Це запобігає атакам на внесення змін від імені авторизованого користувача.

Захист від XSS: Цей метод полягає в обмеженні можливостей внесення скриптів на стороні клієнта, які можуть бути виконані безпосередньо в браузері користувача. Захист від XSS дозволяє зменшити ризик крадіжки сесій та інші подібні атаки.

Використання різних рівнів автентифікації та авторизації: Цей метод полягає в тому, щоб вимагати різних рівнів автентифікації та авторизаціції для

різних дій та функцій веб-додатка. Наприклад, вимагати введення пароля для доступу до адміністративної панелі, або використовувати двофакторну автентифікацію для підтвердження ідентифікації користувача. Цей метод зменшує ризик несанкціонованого доступу до веб-додатку та його даних.

Редагування файлів конфігурації: Веб-додатки зазвичай містять файли конфігурації, які містять конфіденційну інформацію, таку як дані доступу до баз даних та інші налаштування[13]. Редагування цих файлів може призвести до компрометації веб-додатка та його даних. Захист від цього полягає в обмеженні доступу до файлів конфігурації та використанні механізмів автоматизованого виявлення змін в цих файлах.

Захист від DoS-атак: DoS-атаки можуть призвести до недоступності веб-додатку та втрати даних. Захист від цього полягає в використанні спеціального програмного забезпечення, яке забезпечує розподілене кешування запитів та мінімізацію відповідей на неправильні запити.

Узагальнюючи, існує багато методів захисту веб-додатків, які можуть бути використані для захисту від різних видів атак. Кожен з цих методів має свої переваги та обмеження, тому важливо використовувати комбінацію методів, щоб забезпечити максимальний рівень захисту веб-додатків.

2.2. Принцип роботи і розгортання WAF

WAF (Web Application Firewall) - це програмне забезпечення, яке призначене для захисту веб-додатків від різних видів атак, таких як SQL-ін'єкції, кросс-сайтові скрипти (XSS), атаки на міжсайтове піддавання (CSRF), брутфорс-атаки, ін'єкції коду та інших подібних вразливостей.

Основний принцип роботи WAF полягає в тому, щоб аналізувати вхідні запити, які надходять на сервер, та перевіряти їх на наявність потенційних вразливостей. Після цього WAF може відхилити або блокувати запити, які містять

небезпечні параметри, тим самим забезпечуючи захист веб-додатку від різних видів атак.

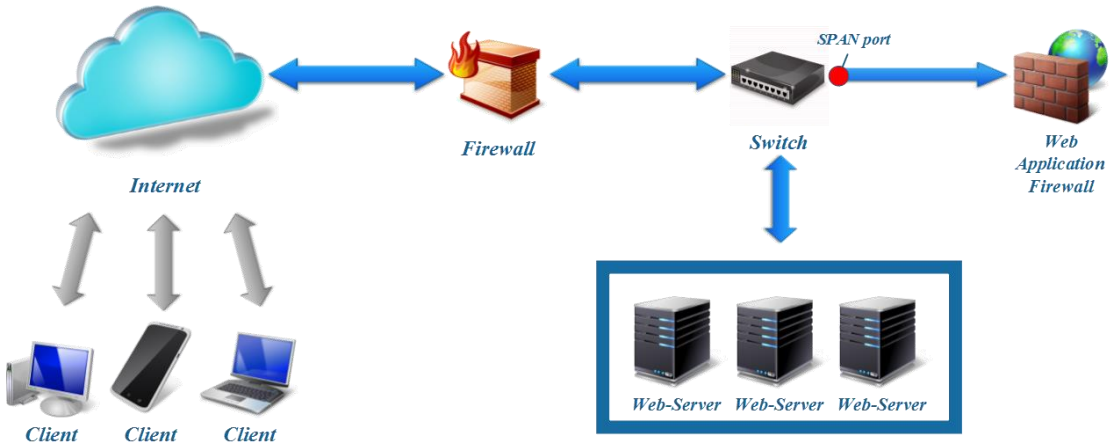

Рис. 2.4. Розгортання WAF

Розгортання WAF може здійснюватись в декілька способів. Один з найпоширеніших способів - це розгортання WAF як апаратного пристрою, який підключається між веб-сервером та Інтернетом. Цей пристрій може працювати як з точно налаштованою базою правил, так і з аналізом вхідних запитів за допомогою машинного навчання та інших методів.

Ще один спосіб розгортання WAF - це використання спеціальних модулів у веб-сервері або застосунку. Наприклад, модуль mod_security для веб-сервера Apache може виконувати функції WAF безпосередньо на веб-сервері.

Деякі WAF можуть бути розгорнуті у хмарному середовищі, що дозволяє захищати веб-додатки, які розгорнуті на хмарних платформах.

Крім того, WAF може бути розгорнутий на рівні мережі, дозволяючи захистити веб-додатки від атак, що надходять з мережі, до якої підключений веб-сервер.

Для ефективної роботи WAF потрібно правильно налаштувати та підібрати набір правил, які визначають, які запити будуть дозволені, а які - заблоковані. Це може бути складним процесом, оскільки потрібно забезпечити як максимальний захист веб-додатку, так і його нормальну функціональність.

Окрім цього, WAF потребує постійного оновлення та моніторингу, оскільки нові види атак можуть з'являтись постійно. Для цього використовуються спеціальні сервіси, які надають оновлення для WAF та моніторинг потенційних загроз.

Узагалі, WAF є важливим інструментом для захисту веб-додатків від різних видів атак, тому розгортання та налаштування WAF має велике значення для забезпечення безпеки веб-додатків.

2.2.1. Аналіз архітектури WAF та принципу його роботи

Однією з головних особливостей архітектури WAF є його розташування між веб-сервером та Інтернетом. Це дозволяє WAF аналізувати всі вхідні запити, що надходять на сервер, та перевіряти їх на наявність потенційних вразливостей.

Основний принцип роботи WAF полягає в тому, щоб аналізувати вхідні запити, які надходять на сервер, та перевіряти їх на наявність потенційних вразливостей. Для цього WAF використовує різноманітні методи, такі як перевірка синтаксису запиту, виявлення SQL-ін'єкцій, перевірка параметрів запиту на наявність небезпечних символів, визначення потенційних вразливостей, пов'язаних з використанням куків та інші методи.

Після проведення аналізу WAF може відхилити або блокувати запити, які містять небезпечні параметри, тим самим забезпечуючи захист веб-додатку від різних видів атак.

Ще однією важливою особливістю архітектури WAF є можливість застосування різних методів для виявлення потенційних атак. Для цього WAF може використовувати як точно налаштовану базу правил, так і аналіз вхідних запитів за допомогою машинного навчання та інші алгоритми штучного інтелекту.

Точно налаштована база правил є набором заздалегідь визначених правил, які визначаються на основі відомих вразливостей та типових атак. Ця база правил

забезпечує ефективний захист веб-додатку, але вона потребує регулярного оновлення та додавання нових правил[16].

Аналіз вхідних запитів за допомогою машинного навчання та інших алгоритмів штучного інтелекту є більш гнучким і ефективним методом захисту від нових та невідомих атак. Такий підхід дає можливість WAF навчитись розпізнавати нові атаки та швидко реагувати на них.

Незважаючи на те, що WAF забезпечує ефективний захист веб-додатку, він не є панацеєю від усіх видів атак[12]. Варто пам'ятати, що WAF не може забезпечити захист від вразливостей, що знаходяться в самому веб-додатку, тому що вони не пов'язані з мережевим рівнем. Тому для повноцінного захисту веб-додатку необхідно використовувати не тільки WAF, але й інші методи, такі як регулярне оновлення веб-додатку та його компонентів, перевірка на вразливості на стадії розробки та інші методи.

Узагальнюючи, WAF є важливим елементом безпеки веб-додатків, що забезпечує захист від різних видів атак на мережевому рівні. Його архітектура та принцип роботи дозволяють забезпечувати ефективний захист веб-додатку, якщо використовувати його разом з іншими методами захисту від вразливостей.

2.2.2. Реалізація безпеки при розгортанні WAF

Реалізація безпеки при розгортанні WAF є дуже важливою, оскільки вона забезпечує захист веб-додатку від різних видів атак. Існують різні методи розгортання WAF, які можуть забезпечити різний рівень безпеки.

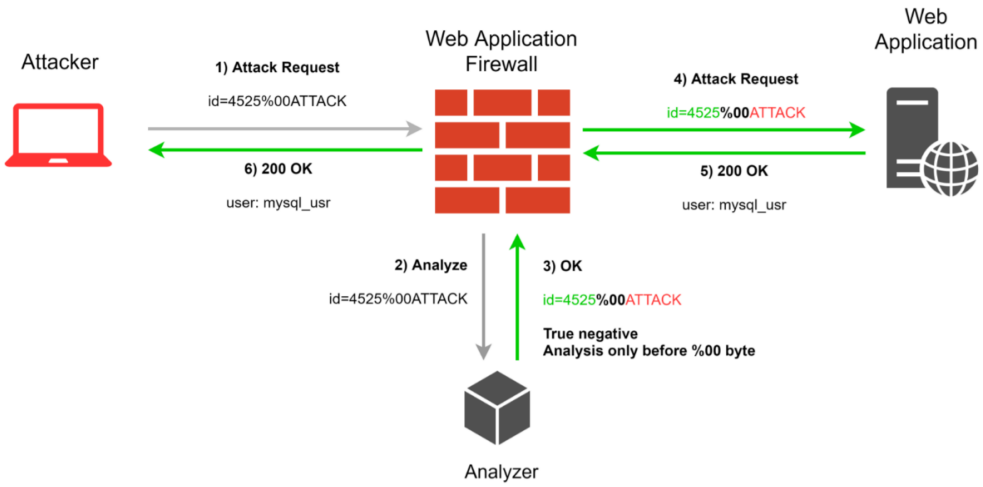

Рис. 2.5. Реалізація безпеки при розгортанні WAF

Один з найпоширеніших методів розгортання WAF є розміщення WAF перед веб-сервером. Цей метод дозволяє WAF аналізувати всі вхідні запити до веб-сервера та забезпечує високий рівень захисту. Однак, для забезпечення безпеки при розгортанні WAF перед веб-сервером потрібно виконати деякі дії.

По-перше, потрібно налаштувати WAF на відповідний рівень безпеки. Для цього можна використовувати різні методи, наприклад, встановлювати різні правила для перевірки запитів на вразливості, використовувати машинне навчання для виявлення потенційних атак, тощо.

По-друге, потрібно забезпечити захист самого WAF від атак. Для цього можна використовувати різні методи, наприклад, використовувати захист від DDoS-атак, встановлювати правила для виявлення атак на WAF, встановлювати доступ до WAF з обмеженнями тощо.

По-третє, потрібно забезпечити захист веб-сервера від атак, які можуть обійти WAF. Для цього можна використовувати різні методи, наприклад, використовувати захист від DDoS-атак, встановлювати правила для виявлення атак на веб-сервер, встановлювати обмеження на вхідні запити до веб-сервера, тощо.

По-четверте, потрібно забезпечити резервне копіювання WAF та веб-сервера для забезпечення можливості під час виникнення технічних проблем. Для цього можна використовувати різні методи, наприклад, встановлювати резервне

копіювання на різних серверах або використовувати хмарні рішення для забезпечення доступності даних.

По-п'яте, потрібно регулярно виконувати тестування на проникнення, щоб виявляти потенційні слабкі місця та уразливості системи. Для цього можна використовувати різні методи, наприклад, проводити тестування на проникнення з зовнішніх джерел або використовувати інструменти для автоматизованого тестування.

Загалом, реалізація безпеки при розгортанні WAF вимагає комплексного підходу та використання різних методів та інструментів для забезпечення максимальної захищеності системи. Для успішного розгортання WAF важливо мати достатній рівень знань та досвіду у галузі безпеки веб-додатків, а також враховувати специфіку конкретного веб-додатку та його потенційні загрози.

Висновки до розділу

У другому розділі дипломної роботи проведено дослідження методів захисту web-додатків та принципу роботи та розгортання WAF.

Було досліджено структуру web-додатків, проведено пошук типових вразливостей та аналіз інструментів для їх виявлення, а також проаналізовано методи захисту від цих вразливостей.

Також було проведено аналіз архітектури WAF та принципу його роботи, а також розглянуто реалізацію безпеки при його розгортанні, зокрема необхідність налаштування WAF на відповідний рівень безпеки, забезпечення захисту самого WAF від атак, захист веб-сервера від атак, які можуть обійти WAF, та резервного копіювання WAF та веб-сервера для забезпечення можливості відновлення в разі непередбачених ситуацій.

Отже, застосування методів захисту web-додатків та розгортання WAF є важливим кроком у забезпеченні безпеки веб-додатків, що дозволяє підвищити рівень захисту від різних видів атак та зменшити ризики вразливостей веб-додатків.

3 РОЗРОБКА РЕКОМЕНДАЦІЙ ЩОДО ЗАХИСТУ WEB – ДОДАТКІВ ЗА ДОМОГОЮ CLOUDFLARE WAF

3.1. Опис Cloudflare WAF

Cloudflare WAF (Web Application Firewall) - це захисний механізм, який застосовується для захисту веб-додатків від різноманітних загроз та атак. Cloudflare WAF є частиною Cloudflare - інтегрованої платформи захисту та прискорення веб-сайтів та додатків[3].

Cloudflare WAF базується на підписах, які містять інформацію про відомі загрози та вразливості веб-додатків. Однак, на відміну від традиційних WAF, які застосовуються на рівні додатку, Cloudflare WAF застосовується на CDN-рівні. Це означає, що Cloudflare WAF аналізує весь трафік, який проходить через CDN, що забезпечує більш ефективний та швидкий захист.

Cloudflare WAF містить багато функцій та можливостей, які дозволяють адміністраторам веб-додатків легко налаштувати та керувати захистом своїх додатків. Зокрема, Cloudflare WAF підтримує:

1. Налаштування блокування за IP-адресою;
2. Налаштування блокування за країною;
3. Налаштування блокування за URL-адресою;
4. Налаштування блокування за вмістом запиту;
5. Налаштування блокування за вмістом відповіді;

Крім того, Cloudflare WAF надає аналітичні звіти, які дозволяють адміністраторам веб-додатків зрозуміти, які атаки були спробовані та як успішно було запобіжено цим атакам[18].

Загалом, Cloudflare WAF - це потужний і ефективний інструмент захисту веб-додатків від різних загроз та атак, який забезпечує багато можливостей та

функцій для налаштування та керування захистом додатків. Будучи частиною інтегрованої платформи Cloudflare, Cloudflare WAF забезпечує швидкий та ефективний захист на CDN-рівні, що дозволяє підвищити рівень безпеки веб-додатків та зменшити ризик їх компрометації.

3.2. Інсталювання та налаштування WAF

Інсталювання та налаштування Cloudflare WAF є досить простим та зручним процесом. Нижче наведено детальну інструкцію з інсталювання та налаштування Cloudflare WAF на веб-сайті:

Реєстрація в Cloudflare

Першим кроком до інсталювання Cloudflare WAF є реєстрація в системі Cloudflare. Для цього необхідно зареєструвати акаунт на офіційному сайті Cloudflare та створити новий сайт в системі.

Крок 1: Відкрийте офіційний сайт Cloudflare, щоб зареєструвати обліковий запис.

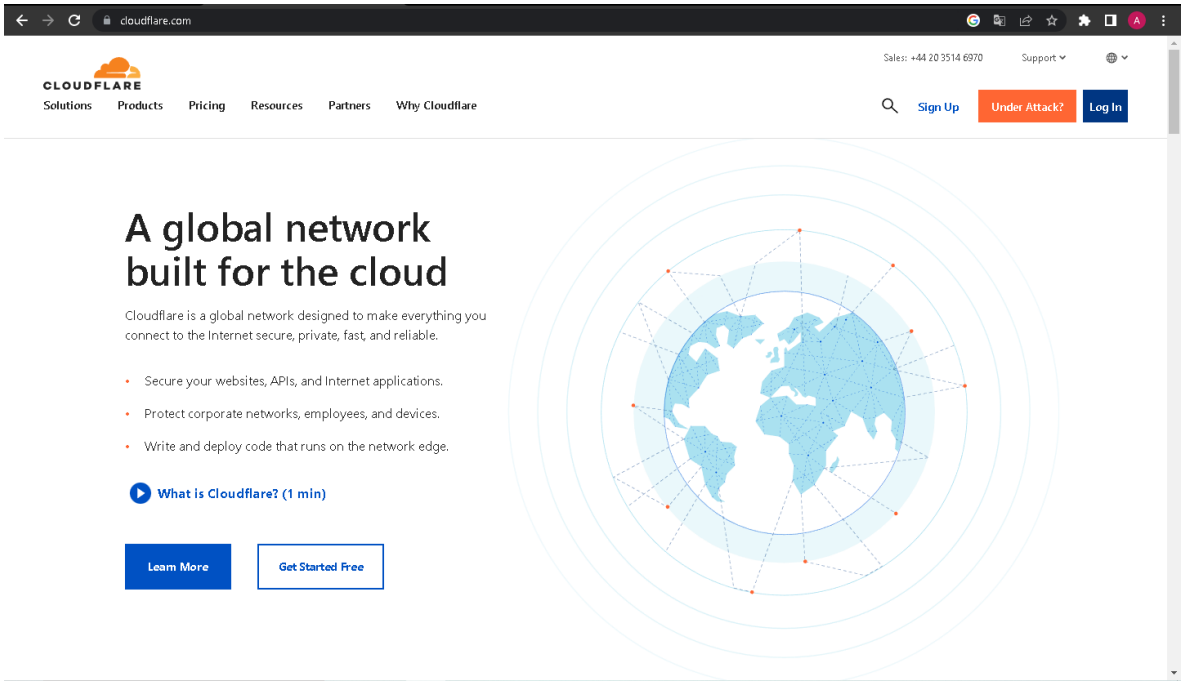

Рис. 3.1. Офіційний сайт Cloudflare

Крок 2: Натисніть на кнопку "Sign Up", щоб перейти до сторінки реєстрації.

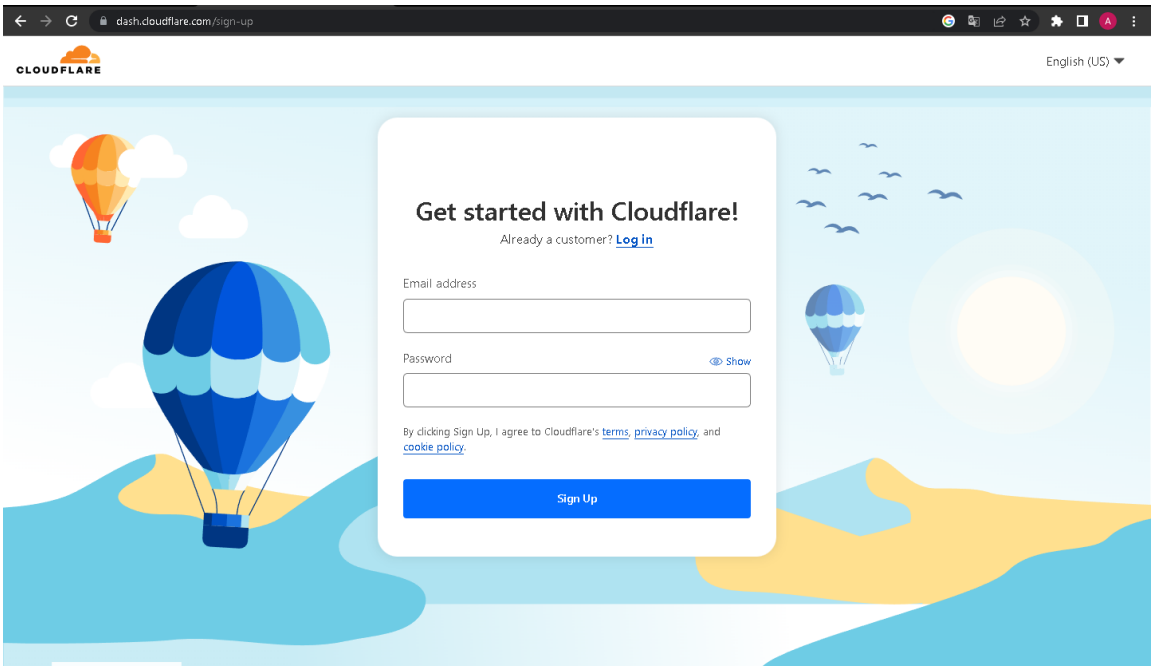

Рис. 3.2. Перехід за посиланням Sign Up

Крок 3: Введіть свою електронну пошту та пароль у відповідні поля та виберіть тарифний план, який відповідає вашим потребам. Cloudflare пропонує безкоштовний тарифний план, який підходить для більшості користувачів.

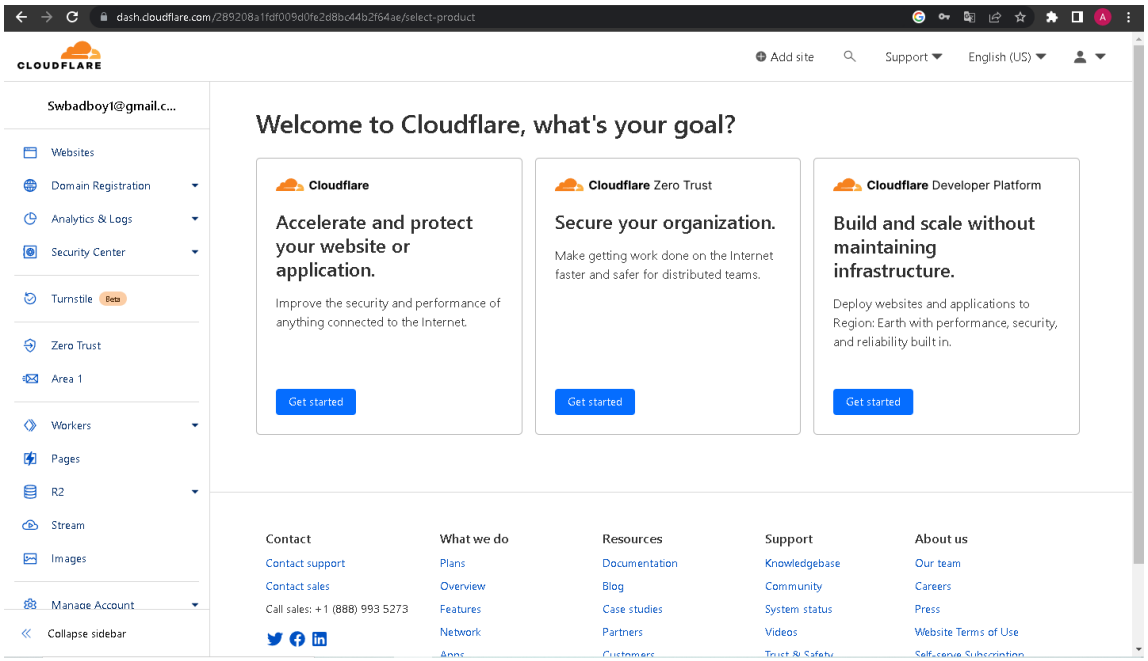

Рис. 3.3. Успішна реєстрація перехід до налаштувань

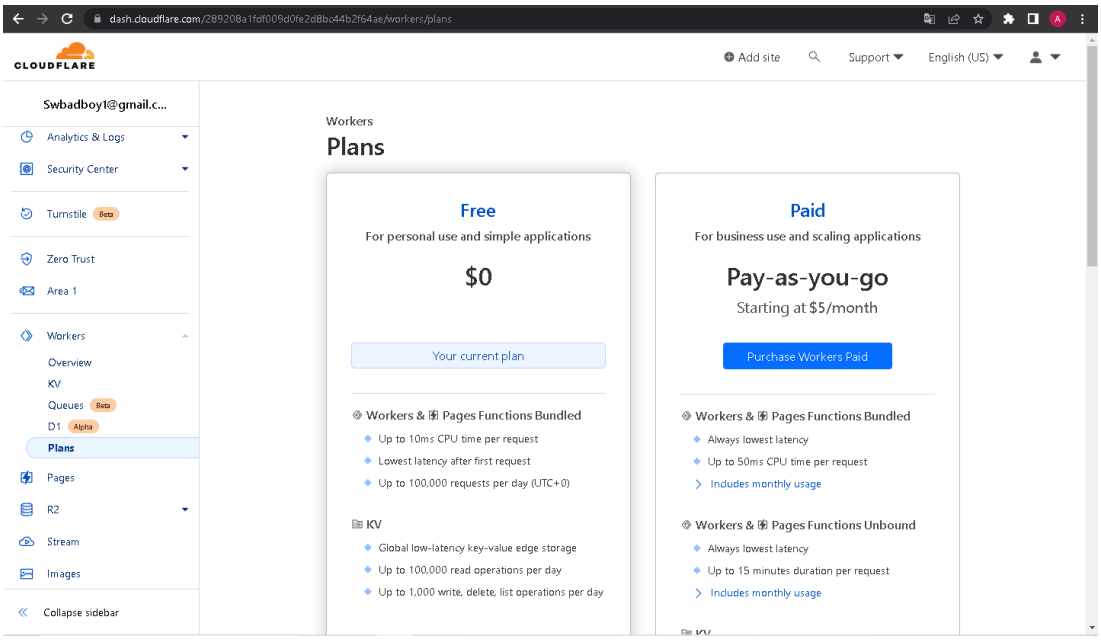

Рис. 3.4. Встановлення безкоштовного плану для демонстрації

Крок 4: Після вибору тарифного плану, натисніть кнопку "Add Site" на сторінці "Dashboard".

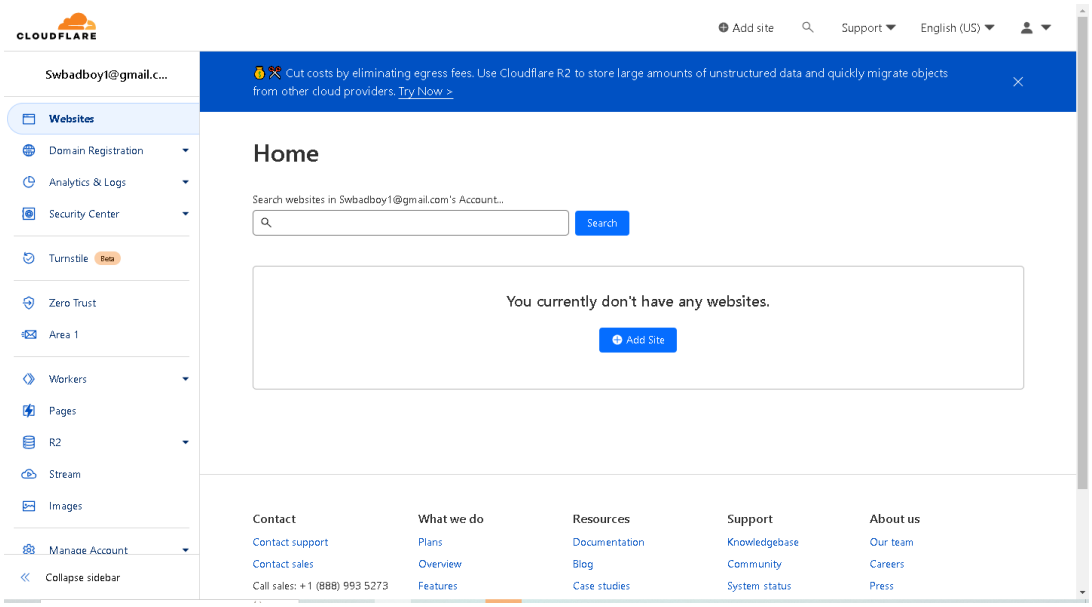

Рис. 3.5. Перехід до секції Web-Site

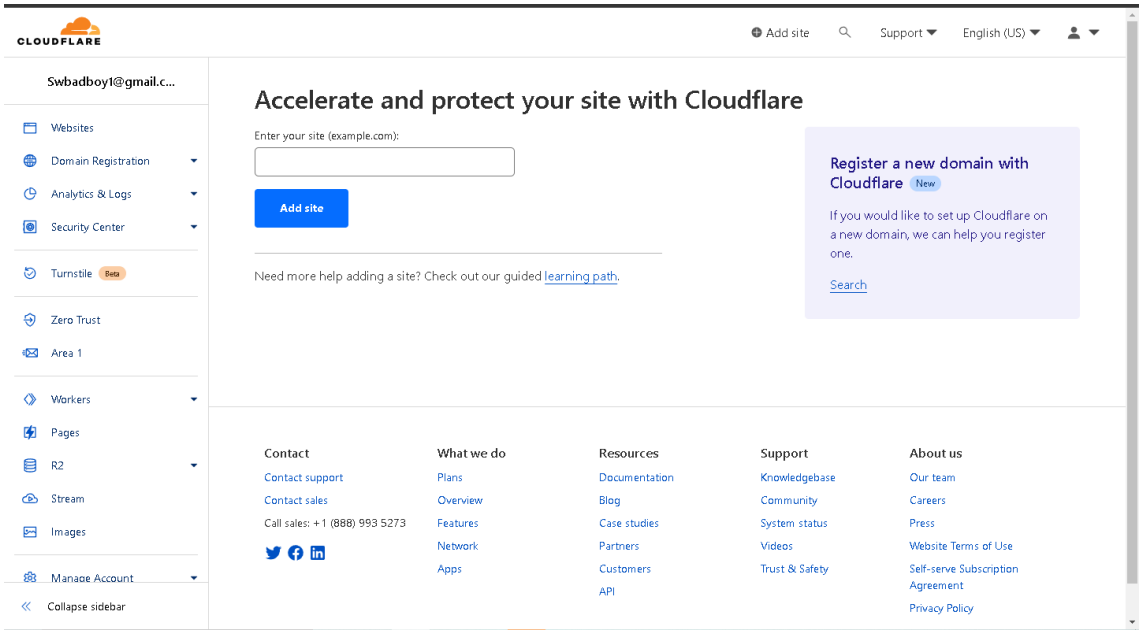

Рис. 3.6. Налаштування Add site

Крок 5: Введіть URL веб-сайту, який ви хочете захистити, та натисніть кнопку "Begin Scan". Для демонстрації буде використано веб-додаток який задиплоєно на платформі Vercel.

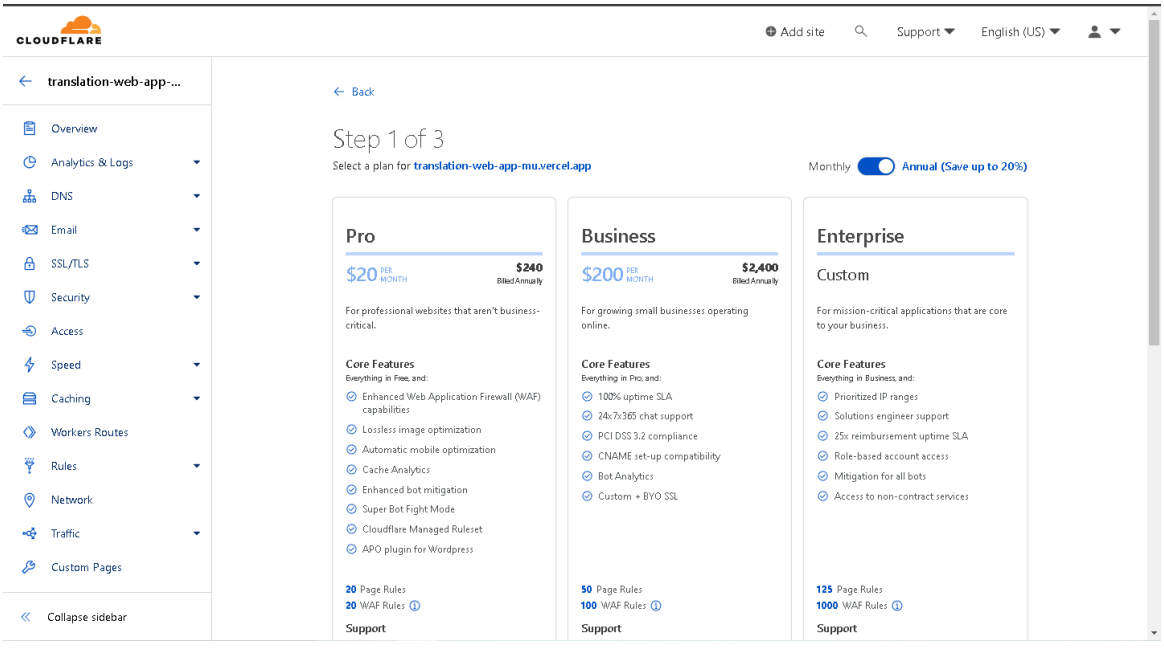

Рис. 3.7. Обрання плану після введення дійсного веб-сайту

Cloudflare, одразу пропонує декілька найпопулярніших планів, але для демонстрації використаємо безкоштовний план, який знаходиться нижче:

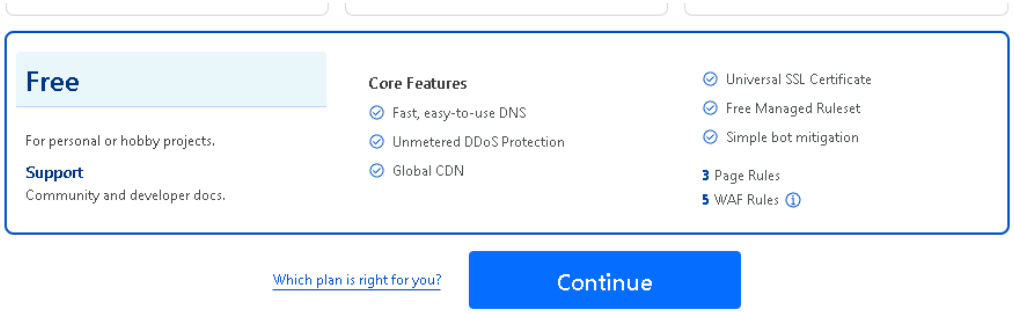

Рис. 3.8. Обрання плану

Крок 6: Cloudflare автоматично знайде потрібний домен та переведе вас до сторінки "DNS Records". На цій сторінці ви зможете перевірити та налаштувати DNS-записи для вашого сайту.

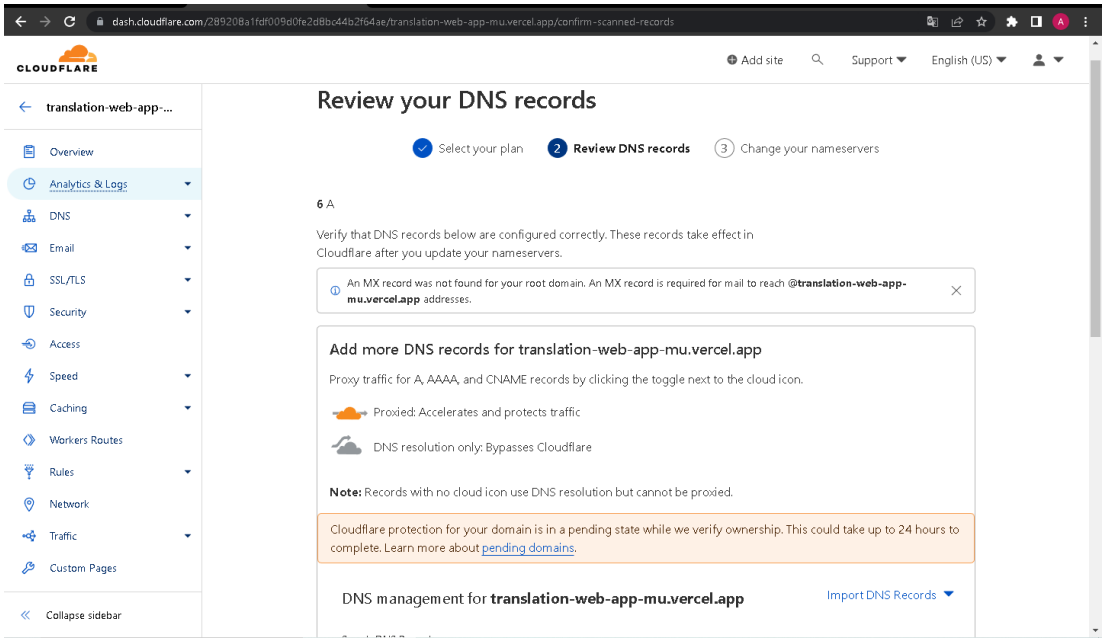

Рис. 3.9.1. Сторінка огляду та налаштування DNS

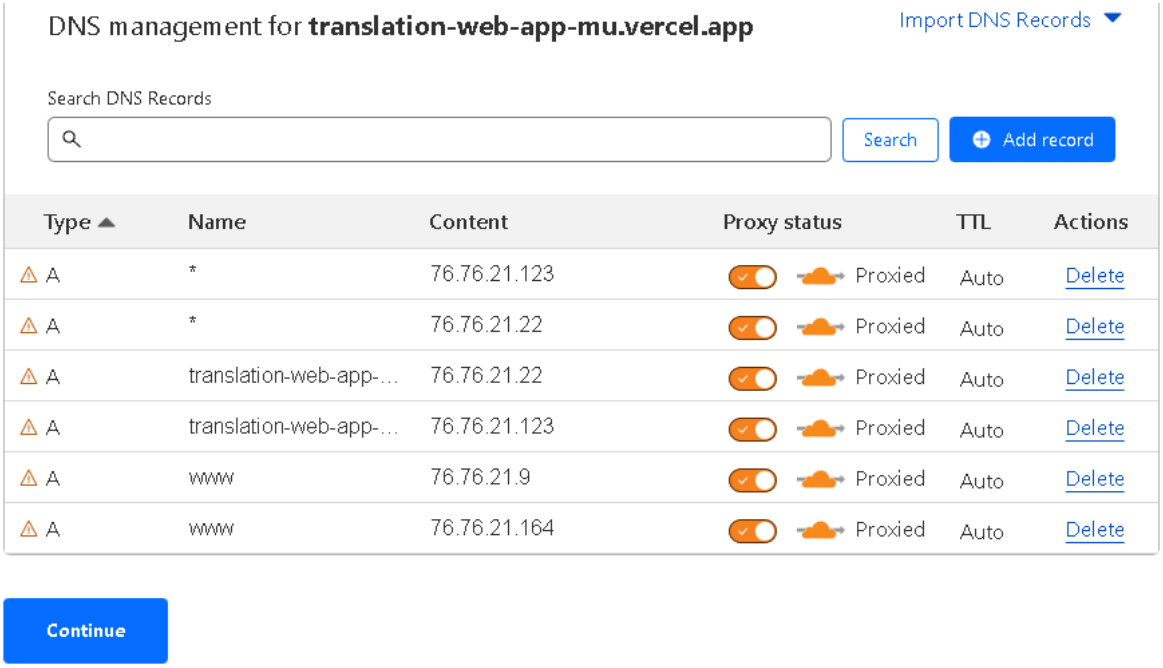

Рис. 3.9.2. Сторінка огляду та налаштування DNS

Крок 7: Після перевірки та налаштування DNS-записів, натисніть кнопку "Continue". Вам буде запропоновано вибрати тарифний план для вашого сайту.

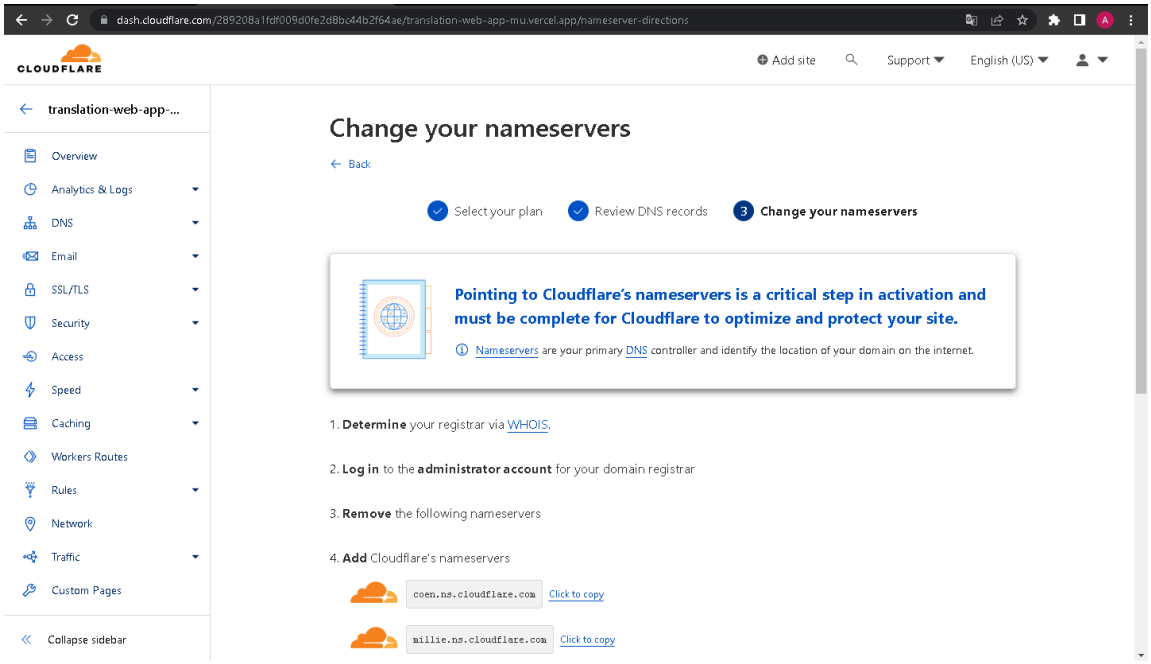

Рис. 3.10. Обов’язковий крок після DNS-records змінення nameservers

Для впровадження необхідний змін потрібно зробити ще декілька кроків, першим буде встановлення запропонованих імен до Vercel.

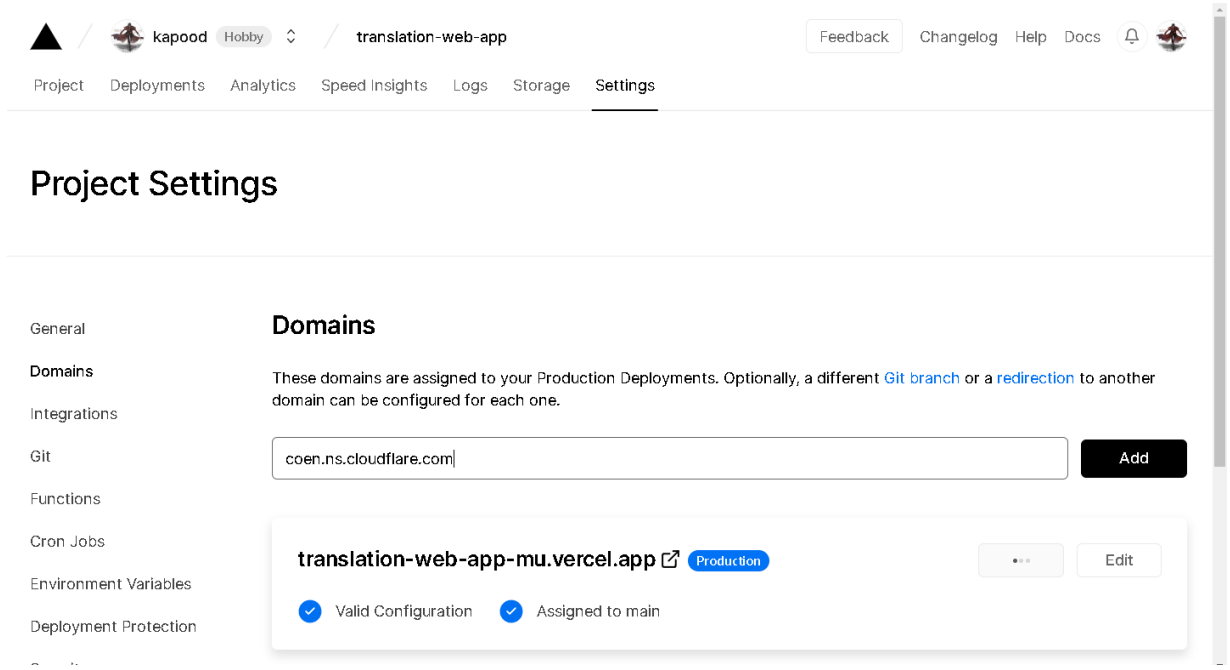

Рис. 3.12. Необхідні налаштування Dns

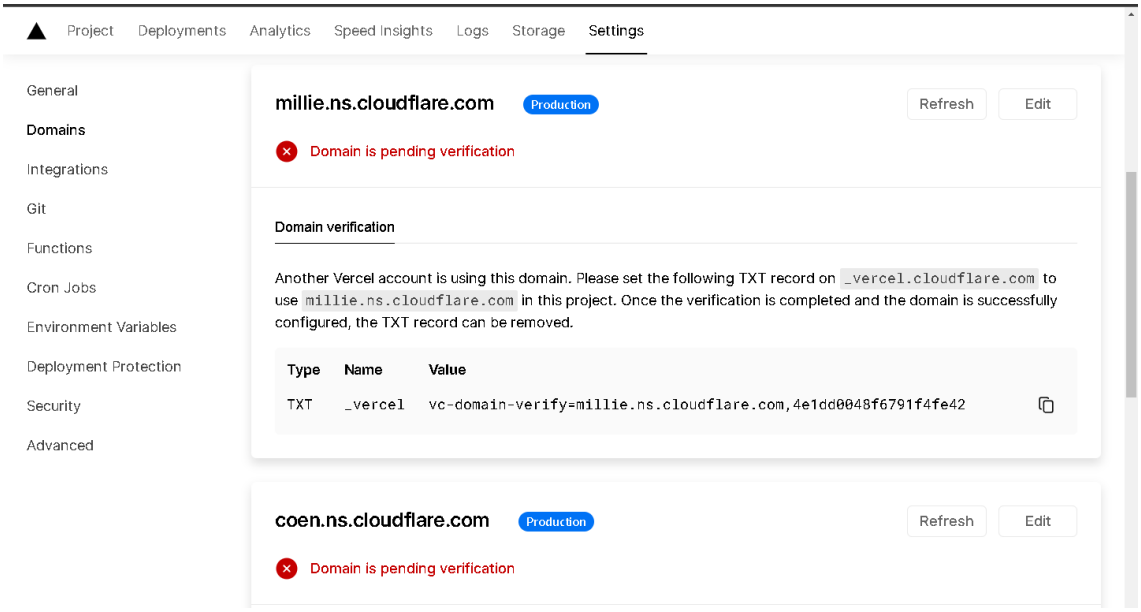

Рис. 3.13. Встановлені домени

Для інтеграції Vercel з Cloudflare необхідно встановити параметри які зазначує Vercel до домених налаштування Cloudflare, щоб підтвердити, що це дійсно ваш ресурс, верефікація буде займати деякий час.

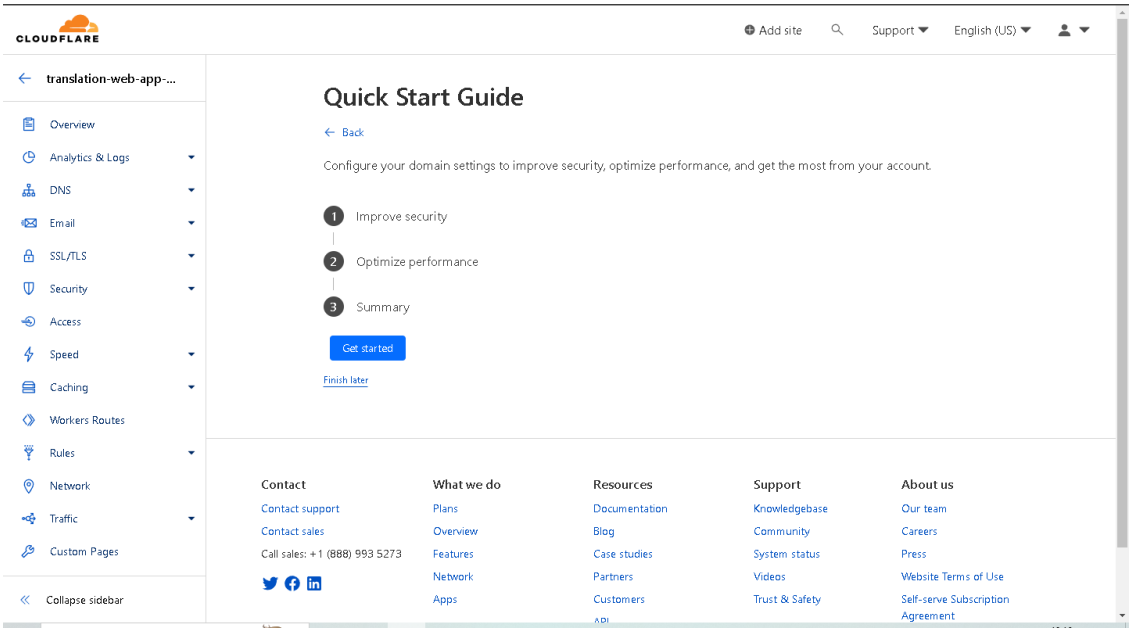

Рис. 3.14. Завершення базових налаштувань

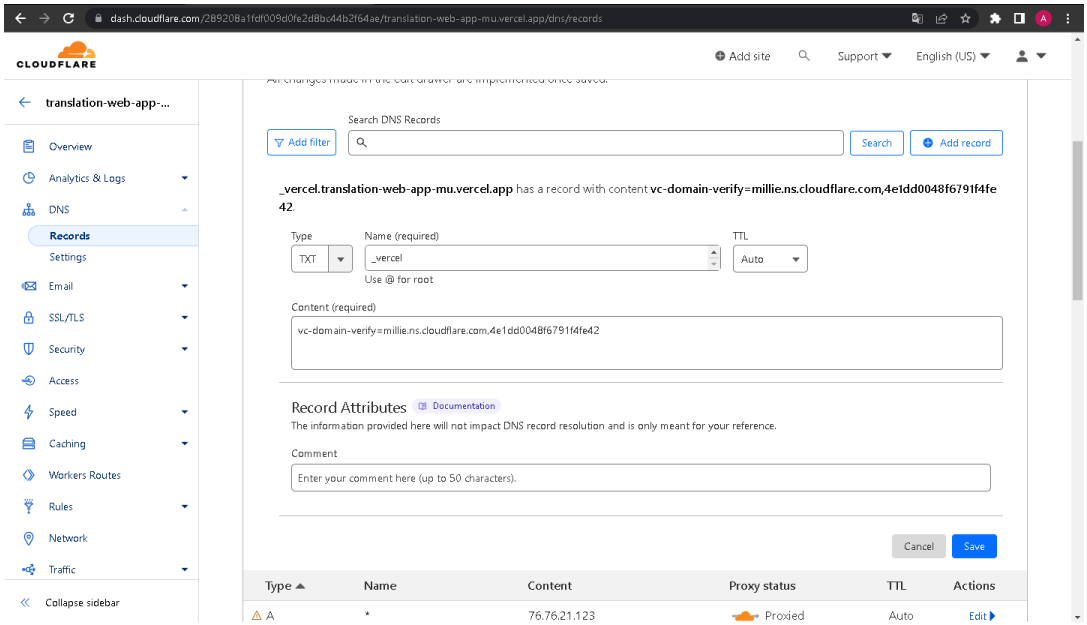

Рис. 3.15. Додавання Vercel до Dns Cloudflare

Крок 10: На сторінці "Overview" ви зможете переглянути основну інформацію про ваш сайт та його стан в системі Cloudflare. Також тут ви зможете активувати різноманітні функції, включаючи Cloudflare WAF.

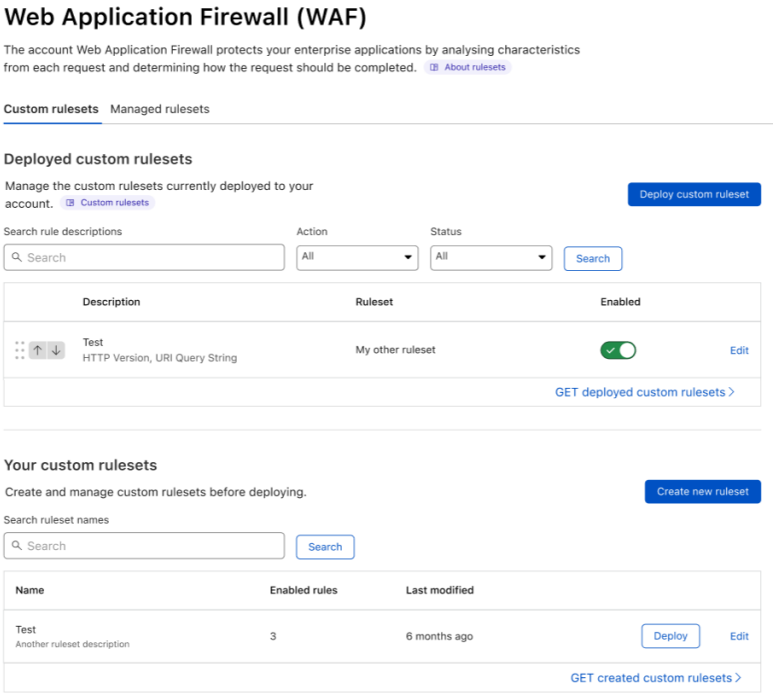

Рис. 3.16. Підключення WAF

Після виконання цих кроків ви успішно зареєструвалися в системі Cloudflare та налаштували Cloudflare WAF для свого сайту.

Загалом, інсталювання та налаштування Cloudflare WAF є важливим кроком для забезпечення безпеки веб-додатку. Після активації WAF на веб-сайті можна бути впевненим у тому, що будь-які атаки на сайт будуть відфільтровані, а необхідні заходи будуть вжиті для його захисту.

3.3. Тестування ефективності WAF

Тестування ефективності WAF (Web Application Firewall) включає в себе перевірку роботи WAF та виявлення та блокування атак на захищений веб-сайт. Це допомагає забезпечити захист від вразливостей та запобігти нападам на веб-додатки.

Для тестування ефективності WAF можна використовувати різноманітні інструменти, наприклад, відомі утиліти для сканування вразливостей веб-сайтів, такі як Nikto, Burp Suite, Acunetix та інші. Зазвичай, ці інструменти дозволяють

виконувати різноманітні види атак, такі як SQL-ін'єкції, XSS-атаки та інші, та перевіряти реакцію WAF на такі атаки.

Під час тестування ефективності WAF важливо також перевірити, чи не блокує WAF легітимний трафік до захищеного веб-сайту. Для цього можна спробувати доступитися до різних сторінок веб-сайту та виконати декілька тестових запитів, щоб переконатися, що всі запити пройшли успішно та WAF не блокує корисний трафік.

Крім того, варто звернути увагу на те, які типи атак визнає WAF, а також на те, як швидко він реагує на нові види атак та вразливості. Тому, перед використанням WAF на продуктивному середовищі, варто виконати ретельне тестування та переконатися, що WAF працює ефективно та не блокує легітимний трафік до захищеного веб-сайту.

Щоб провести тестування ефективності WAF, можна використати декілька тестових векторів, що містять типові атаки. Нижче описано кроки для проведення тестування на основі тестових векторів.

SQL-ін'єкції: ввести запити, що містять SQL-ін'єкції на сторінках, які повинні бути захищені. Наприклад, введіть SQL-запит, що містить команду SELECT, щоб отримати всі дані з таблиці. Якщо WAF працює належним чином, то він блокуватиме спробу введення SQL-ін'єкції.

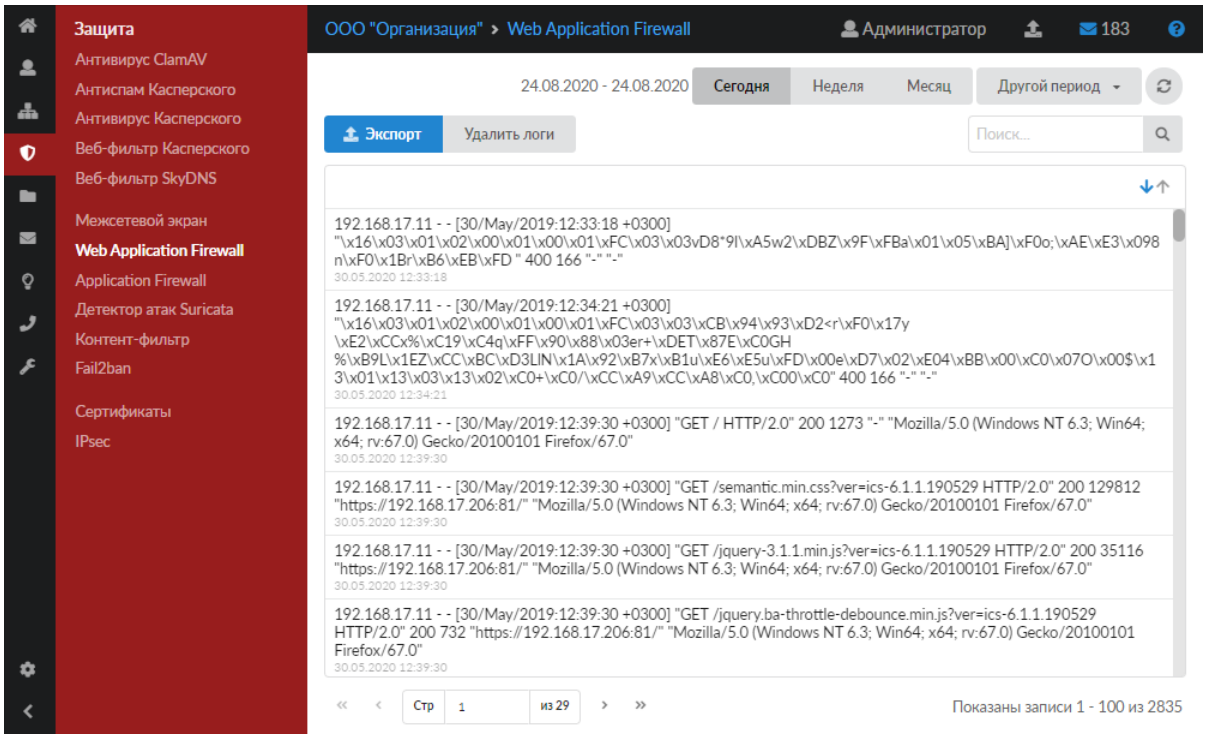

Рис. 3.17. Логи SQL – ін'єкцій атак

XSS атаки: спробуйте ввести скрипти на сторінках, які повинні бути захищені від XSS-атак. Наприклад, введіть скрипт, що змінює фоновий колір сторінки. Якщо WAF працює належним чином, то він блокуватиме спробу введення XSS-атаки.

```
XSS/98.json in URL: BLOCKED
XSS/99.json in Body: BLOCKED
XSS/99.json in Referer: BLOCKED
XSS/99.json in ARGS: BLOCKED
XSS/99.json in UA: BLOCKED
XSS/99.json in Cookie: BLOCKED
XSS/99.json in Header: BLOCKED
XSS/99.json in URL: BLOCKED


+----------+-------+
|  Status  | Count |
+----------+-------+
| BYPASSED |   0   |
| BLOCKED  |  1218 |
+----------+-------+


+---------+------+
| Payload | Zone |
+---------+------+
+---------+------+
```

Рис. 3.18. Приклад блокування XSS-атаки

Крадіжка сесій: спробуйте виконати крадіжку сесії на сторінках, які повинні бути захищені. Наприклад, введіть скрипт, що збирає дані з файлу cookie та відправляє їх на інший сервер. Якщо WAF працює належним чином, то він блокуватиме спробу крадіжки сесії[21].

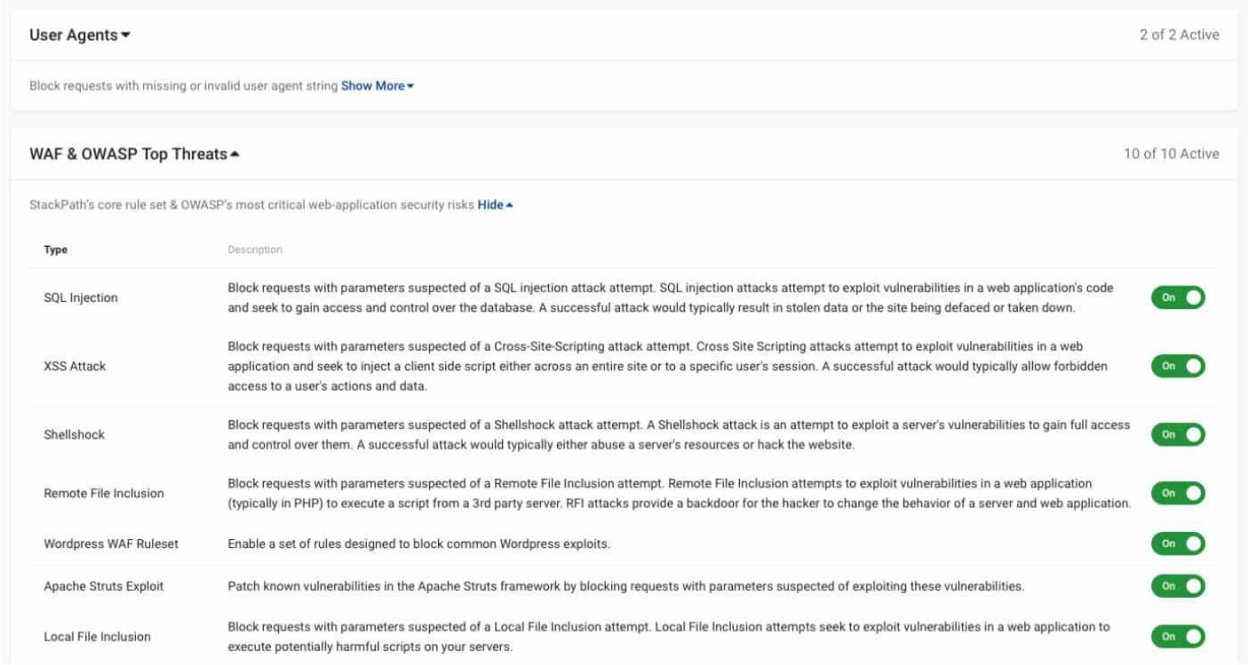

Рис. 3.19. Приклад налаштування WAF для захисту від атак

DDoS-атаки: запустіть DDoS-атаку на веб-сайт, щоб перевірити ефективність WAF в блокуванні таких атак. Наприклад, використайте тестові вектори, що містять велику кількість запитів до веб-сайту. Якщо WAF працює належним чином, то він зможе блокувати DDoS-атаки.

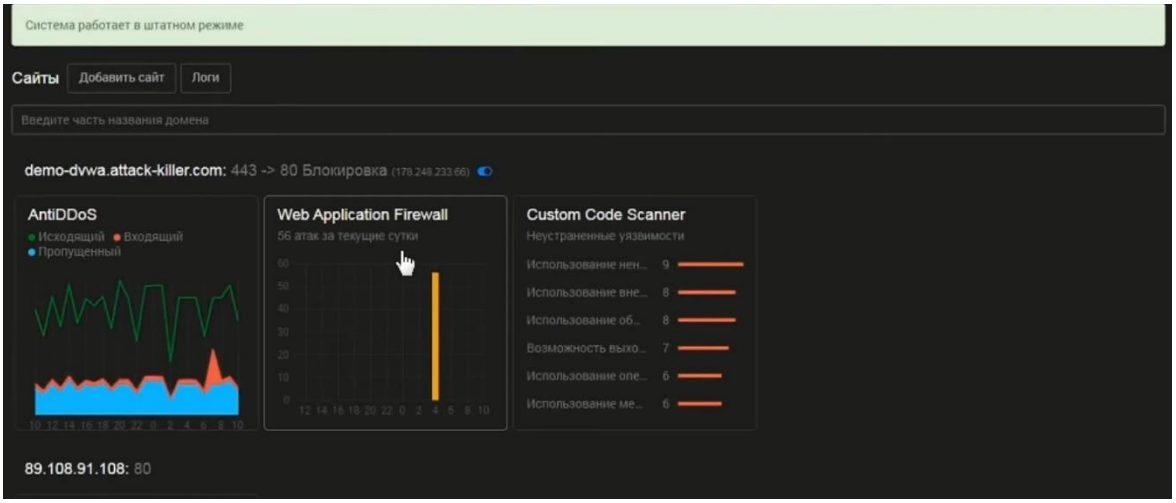

Рис. 3.20. Приклад захисту від DDoS-атаки

Злом паролів: спробуйте ввести неправильний пароль на сторінці авторизації. Якщо WAF працює належним чином, то він зможе виявити та блокувати спробу злому пароля.

Для кожної з цих атак необхідно створити тестовий вектор, який містить типову атаку. Після цього, тестовий вектор можна використовувати для проведення тестування на WAF. Наприклад, для тестування SQL-ін'єкцій, необхідно ввести SQL-запит з ін'єкцією на сторінці, яку захищає WAF. Якщо WAF працює належним чином, то він повинен блокувати спробу SQL-ін'єкції та повідомляти про це користувача або адміністратора веб-сайту.

Після проведення тестування ефективності WAF, необхідно аналізувати результати та визначати, чи ефективно працює WAF у захисті веб-сайту від атак. Якщо результати показують, що WAF не здатен ефективно захищати веб-сайт від атак, то необхідно вжити заходів для покращення безпеки веб-сайту та замінити WAF на більш ефективний рішення.

3.4. Розробка рекомендацій щодо захисту Web-додатків за допомогою CLOUDFLARE WAF

Розробка рекомендацій щодо захисту Web-додатків за допомогою Cloudflare WAF може включати такі етапи:

Аналіз поточного стану безпеки веб-додатку. Під час цього етапу необхідно визначити потенційні уразливості веб-додатку, які можуть бути використані зловмисниками для атак, а також визначити типові атаки, які можуть бути спрямовані на веб-додаток.

Вибір налаштувань Cloudflare WAF. Під час цього етапу необхідно визначити ті правила і налаштування, які будуть найбільш ефективними для захисту веб-додатку від відповідних типів атак.

Налаштування Cloudflare WAF. Після вибору налаштувань Cloudflare WAF необхідно налаштувати їх відповідно до потреб веб-додатку. Це може включати налаштування правил фільтрації трафіку, налаштування списку блокованих IP-

адрес, налаштування списку дозволених IP-адрес, налаштування списку блокованих країн, налаштування списку дозволених країн та інше.

Тестування захисту веб-додатку. Після налаштування Cloudflare WAF необхідно провести тестування захисту веб-додатку, щоб переконатися, що він дійсно захищений від відповідних типів атак. Тестування можна проводити за допомогою тестових векторів, що містять типові атаки, які можуть бути спрямовані на веб-додаток.

Розробка рекомендацій щодо підтримки безпеки веб-додатку. Після тестування захисту веб-додатку необхідно розробити рекомендації щодо підтримки безпеки веб-додатку. Це може включати такі рекомендації, як періодичне проведення аудиту безпеки веб-додатку, оновлення програмного забезпечення веб-сервера та веб-додатку до останніх версій, розробку та впровадження політик безпеки, які мають бути дотримувані при розробці та експлуатації веб-додатку, підтримку безпеки користувачів веб-додатку та інше.

Одним з головних переваг Cloudflare WAF є можливість швидко та ефективно захистити веб-додаток від відповідних типів атак. Крім того, це рішення забезпечує захист веб-додатку від атак зі сторони Інтернету, що дозволяє зменшити ризик інцидентів з безпекою. Проте, варто зазначити, що використання Cloudflare WAF не є повністю гарантованим захистом від усіх можливих атак, тому слід дотримуватися рекомендацій щодо підтримки безпеки веб-додатку та вживати інших заходів для забезпечення безпеки веб-додатку.

Кожен етап розробки рекомендацій щодо захисту веб-додатків за допомогою Cloudflare WAF має свої особливості та вимагає певної кваліфікації в галузі кібербезпеки. Розглянемо кожен етап детальніше:

Аналіз поточного стану безпеки веб-додатку

Перший етап передбачає проведення аналізу безпеки веб-додатку. На цьому етапі необхідно визначити потенційні уразливості веб-додатку, які можуть бути використані зловмисниками для атак, а також визначити типові атаки, які можуть

бути спрямовані на веб-додаток. Аналіз проводять шляхом сканування веб-додатка на наявність вразливостей, дослідження джерела даних, в якому знаходиться веб-додаток, вивчення логів атак, які вже були спрямовані на веб-додаток. На основі отриманих даних розробляється план захисту веб-додатку.

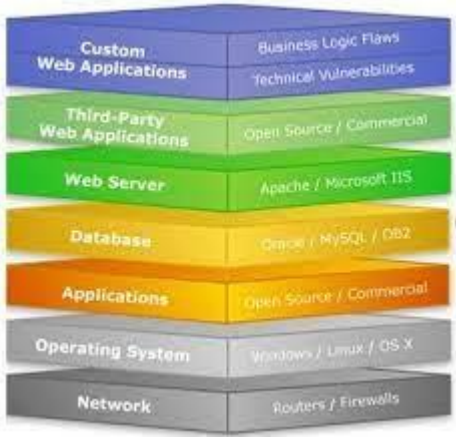

Рис. 3.21. Аналіз поточного стану

Вибір налаштувань Cloudflare WAF

На другому етапі необхідно вибрати ті налаштування Cloudflare WAF, які будуть найбільш ефективними для захисту веб-додатку від відповідних типів атак. Для цього необхідно визначити тип атак, які можуть бути спрямовані на веб-додаток та відповідні налаштування Cloudflare WAF, що можуть запобігти таким атакам. Наприклад, для захисту від SQL-ін'єкцій можна використовувати налаштування, що фільтрують запити на основі певних параметрів. Також на цьому етапі може бути визначено налаштування правил фільтрації трафіку, списку блокованих IP-адрес, списку дозволених IP-адрес, списку блокованих країн, налаштування захисту від DDoS-атак та інших типів атак.

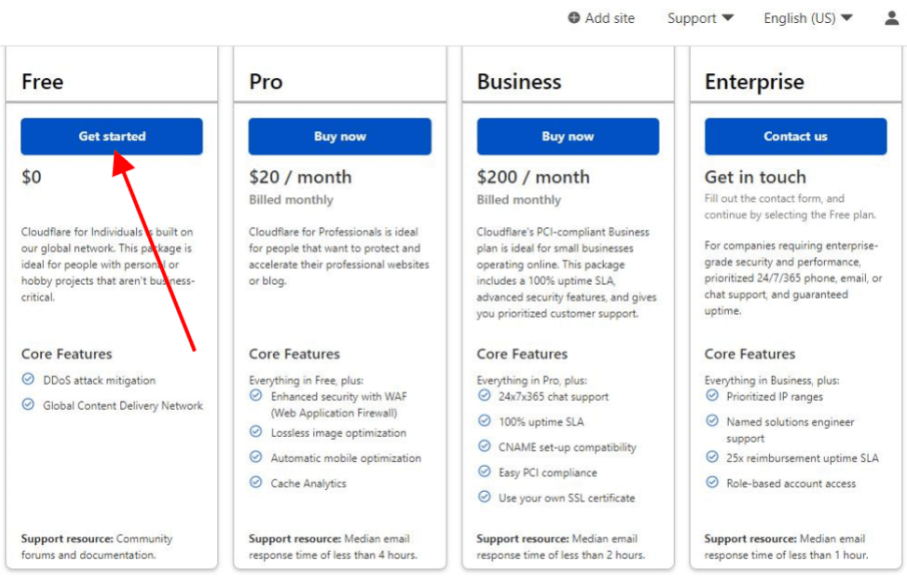

Рис. 3.22. Вибір налаштувань WAF

Налаштування Cloudflare WAF

Третій етап полягає у встановленні та налаштуванні Cloudflare WAF згідно визначених на попередньому етапі параметрів. Важливо забезпечити належну настройку параметрів WAF, адже неправильне налаштування може призвести до неправильної роботи веб-додатку або недостатнього рівня захисту. Наприклад, якщо налаштування WAF дуже строгі, то вони можуть блокувати легітимний трафік, що може призвести до проблем з доступом до веб-додатку. У той же час, якщо налаштування WAF занадто м'які, то це може дозволити зловмисникам здійснювати атаки на веб-додаток.

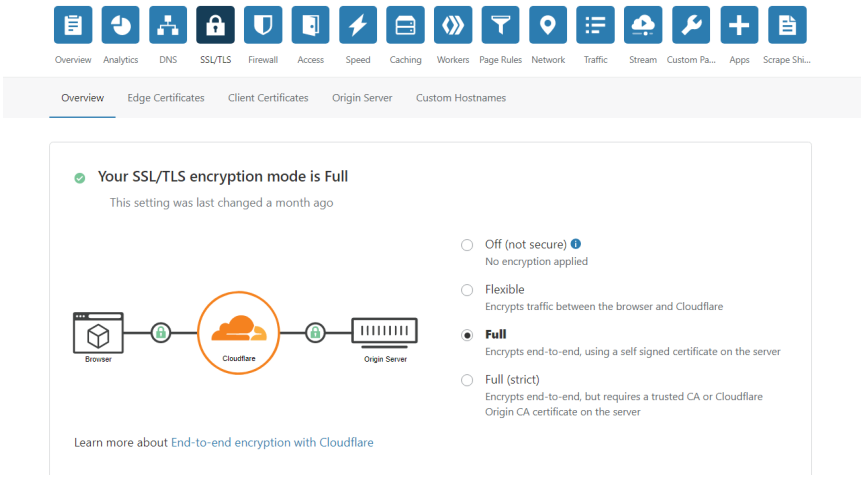

Рис. 3.23. Налаштування WAF

Тестування захисту веб-додатку

Четвертий етап передбачає проведення тестування захисту веб-додатку з використанням Cloudflare WAF. На цьому етапі необхідно перевірити, наскільки ефективним є захист веб-додатку від відповідних типів атак. Тестування можна проводити як вручну, так і автоматично з використанням спеціалізованих інструментів. В результаті тестування можуть бути виявлені проблеми у захисті веб-додатку, які необхідно виправити.

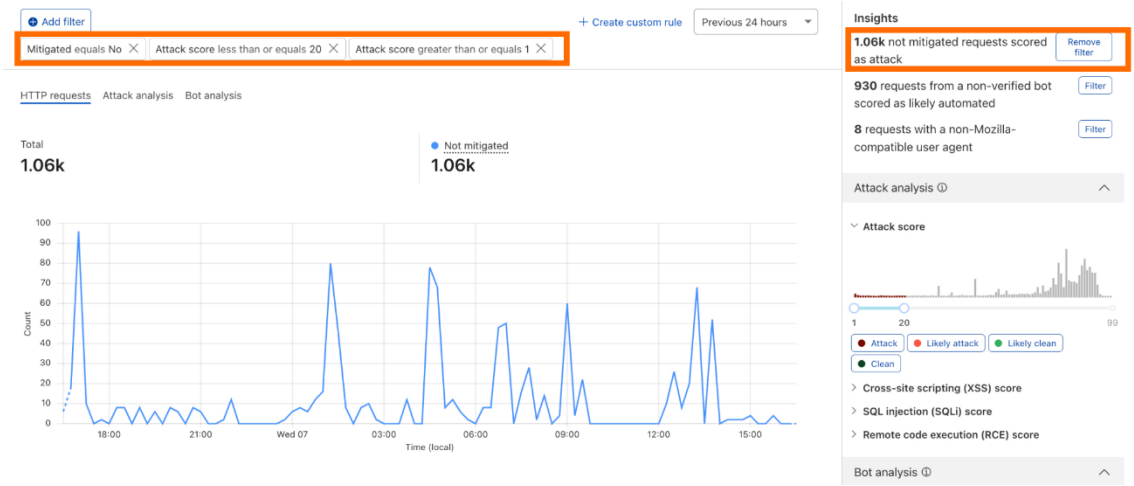

Рис. 3.24. Тестування захисту веб-додатку

Моніторинг та оновлення захисту веб-додатку

П'ятий етап передбачає постійний моніторинг та оновлення захисту веб-додатку з використанням Cloudflare WAF. На цьому етапі необхідно відслідковувати нові типи атак та зміни в захисних механізмах, які можуть виникати з часом. Також необхідно регулярно оновлювати налаштування Cloudflare WAF, щоб вони відповідали новим потенційним загрозам безпеки. Моніторинг може бути здійснюваний вручну або за допомогою автоматизованих інструментів, які можуть сповіщати про потенційні загрози безпеки в реальному часі.

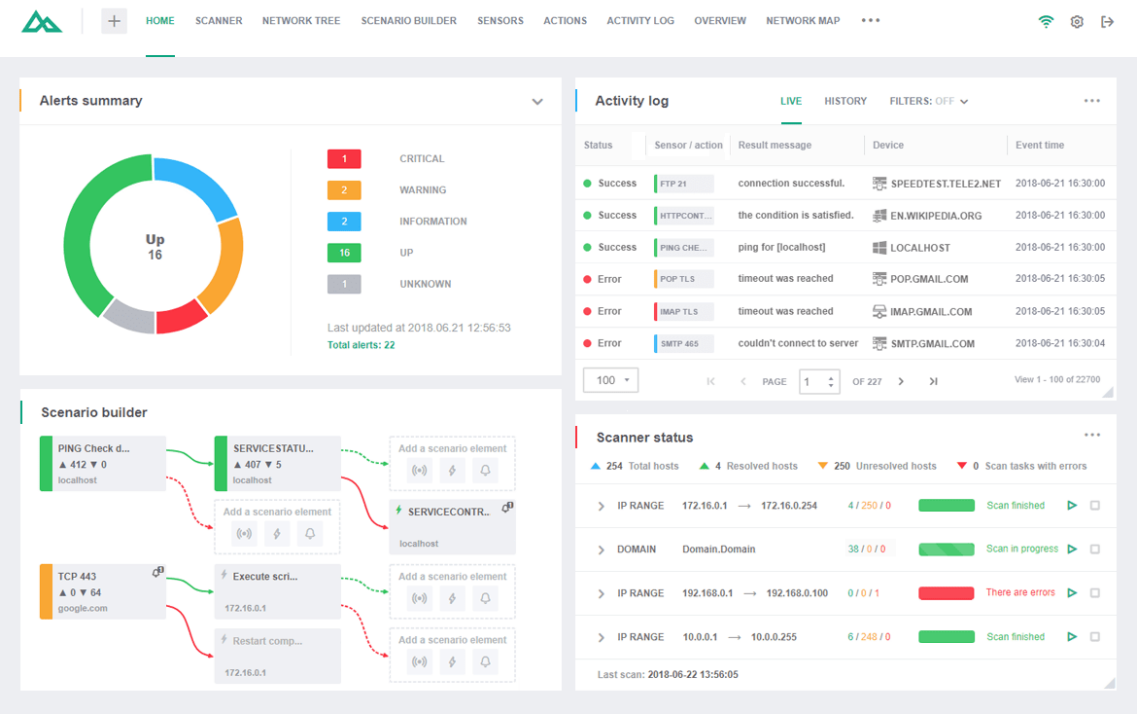

Рис. 3.25. Моніторинг та оновлення

Крім того, важливо періодично проводити аудит безпеки веб-додатку з метою виявлення нових вразливостей та забезпечення належного рівня захисту. Після виявлення нових вразливостей, необхідно вчасно вживати заходів щодо їх виправлення та запобігання можливих атак.

Узагальнюючи, захист веб-додатку від DDoS-атак та інших типів атак може бути забезпечений за допомогою Cloudflare WAF. Для досягнення належного

рівня захисту необхідно правильно визначити параметри WAF, провести тестування захисту веб-додатку, постійно моніторити та оновлювати захист веб-додатку. Також важливо періодично проводити аудит безпеки веб-додатку з метою виявлення нових вразливостей та забезпечення належного рівня захисту.

Висновки до розділу

У даному розділі було розглянуто питання захисту веб-додатків за допомогою Cloudflare WAF. Було описано принцип роботи та функціональні можливості даного захисного механізму, а також розглянуто етапи встановлення, налаштування та тестування ефективності WAF. На основі проведеного тестування було розроблено рекомендації щодо належного захисту веб-додатків за допомогою Cloudflare WAF. Основним завданням даного розділу було надання корисної інформації про можливості та принцип роботи Cloudflare WAF, а також допомогти розробити належні рекомендації щодо захисту веб-додатків з його використанням.

ВИСНОВКИ

У цій дипломній роботі було проведено аналіз сучасних кібератак, спрямованих на веб-додатки, досліджено загрози та вразливості Web-додатків, а також методи їх захисту. Було проведено дослідження методів захисту Web-додатків, зокрема аналізовано структуру веб-додатків, пошук типових вразливостей та аналіз інструментів для виявлення вразливостей. Також було розглянуто принцип роботи та розгортання WAF (Web Application Firewall).

У третьому розділі було розроблено рекомендації щодо захисту Web-додатків з використанням Cloudflare WAF. Було описано Cloudflare WAF, проведено інсталювання та налаштування WAF, а також проведено тестування ефективності захисту. На основі отриманих результатів було розроблено рекомендації щодо захисту Web-додатків з використанням Cloudflare WAF.

Отже, результати дослідження показали, що веб-додатки є вразливими для кібератак та що важливість їх безпеки надзвичайно висока. Було досліджено різноманітні методи захисту Web-додатків та проведено аналіз принципів роботи та розгортання WAF. На основі розроблених рекомендацій можна зробити висновок, що захист веб-додатків з використанням Cloudflare WAF є ефективним та забезпечує високий рівень безпеки.

ПЕРЕЛІК ПОСИЛАНЬ

1. OWASP Top 10:2021. OWASP Foundation, the Open Source Foundation for Application Security OWASP. URL: https://owasp.org/Top10/ (дата звернення: 10.03.2023).

2. Akamai. State of the Internet. Security. URL: https://www.akamai.com/our-thinking/the-state-of-the-internet/state-of-the-internet-security-reports-2020 (дата звернення: 10.03.2023).

3. Cloudflare Security. URL: https://www.cloudflare.com/security/ (дата звернення: 12.04.2023).

4. Data Breach Investigations Report. Verizon Business. URL: https://enterprise.verizon.com/resources/reports/dbir/ (дата звернення: 09.03.2023).

5. Cyber Security Resources. SANS Institute. Web Security. URL: https://www.sans.org/security-resources/ (дата звернення: 15.03.2023).

6. National Institute of Standards and Technology. Security and Privacy Controls for Information Systems and Organizations. URL: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf (дата звернення: 08.03.2023).

7. Securelist. Kaspersky's threat research and reports. URL: https://securelist.com/ (дата звернення: 17.03.2023).

8. Microsoft. Security Intelligence Report. URL: https://www.microsoft.com/security/blog/tag/security-intelligence-report/ (дата звернення: 09.03.2023).

9. Dark Reading. Cybersecurity News & Analysis. URL: https://www.darkreading.com/ (дата звернення: 07.03.2023).

10. Symantec. Internet Security Threat Report. URL: https://www.symantec.com/security-center/threat-report (дата звернення: 11.03.2023).

11. McAfee. McAfee Threats Report. URL: https://www.mcafee.com/enterprise/en-us/threat-center/threat-reports.html (дата звернення: 11.03.2023).

12. Imperva. Web Application Security. URL: https://www.imperva.com/learn/application-security/ (дата звернення: 15.03.2023).

13. Fortinet. FortiGuard Threat Intelligence Service. URL: https://www.fortinet.com/services/threat-intelligence-service.html (дата звернення: 18.03.2023).

14. FireEye. Cyber Threat Intelligence. URL: https://www.fireeye.com/services/cyber-threat-intelligence.html (дата звернення: 20.03.2023).

15. IBM. X-Force Threat Intelligence. URL: https://www.ibm.com/security/x-force (дата звернення: 12.03.2023).

16. Check Point Research. Threat Intelligence. URL: https://research.checkpoint.com/threat-intelligence/ (дата звернення: 20.03.2023).

17. Cisco. Cybersecurity Threats & Trends. URL: https://www.cisco.com/c/en/us/products/security/security-reports.html (дата звернення: 12.03.2023).

18. Palo Alto Networks. Unit 42 Threat Intelligence. URL: https://unit42.paloaltonetworks.com/threat-intelligence/ (дата звернення: 15.04.2023).

19. Trend Micro. Threat Intelligence. URL: https://www.trendmicro.com/en_us/business/products/threat-intelligence.html (дата звернення: 14.03.2023).

20. AlienVault. Unified Security Management. URL: https://www.alienvault.com/products/unified-security-management (дата звернення: 12.03.2023).

21. F5 Labs. Threat Intelligence. URL: https://f5.com/labs (дата звернення: 20.04.2023).

22. NCC Group. Cybersecurity Services. URL: https://www.nccgroup.com/us/our-services/cyber-security/ (дата звернення: 10.03.2023).

23. Gartner. Magic Quadrant for Web Application Firewalls. URL: https://www.gartner.com/reviews/market/web-application-firewalls (дата звернення: 12.03.2023).

ДОДАТОК А

Приклад коду тестових векторів

SQL-ін’єкція:

// SQL-запит, який може бути вразливий до SQL-ін’єкції

$query = "SELECT * FROM users WHERE username = '" . $_GET['username'] . "'";

// Код тестового вектора SQL-ін’єкції

http://example.com/index.php?username=' OR 1=1--

XSS-атака:

// HTML-форма, яка може бути вразливою до XSS-атак

```
<form>
  <input type="text" name="name">
  <input type="submit" value="Submit">
</form>
```

// Код тестового вектора XSS-атаки

http://example.com/index.php?name=<script>alert('XSS')</script>

DDoS-атака:

// Код тестового вектора DDoS-атаки з використанням фрагменту Python

```
import socket
import random
import time

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
bytes = random._urandom(1024)

while True:
port = 80
sock.sendto(bytes, ('<IP-адрес цілі>', port))
time.sleep(0.01)
```

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (Презентація)