ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ

КАФЕДРА ІНФОРМАЦІЙНОЇ ТА КІБЕРНЕТИЧНОЇ БЕЗПЕКИ

КВАЛІФІКАЦІЙНА РОБОТА

на тему:

«Дослідження шляхів та розробка рекомендацій щодо захищеності web-додатків»

зі спеціальності *125 Кібербезпека*
(код, найменування спеціальності)

освітньо-професійної програми *Інформаційна та кібернетична безпека*
(назва програми)

Кваліфікаційна робота містить результати власних досліджень. Використання ідей, результатів і текстів інших авторів мають посилання на відповідне джерело

_____ ЛЮЛЬЧУК Нікіта

Виконав: здобувач вищої освіти групи БСД-42
Люльчук Нікіта
(прізвище, ім'я)

Керівник ДОРОХІН Орест
(науковий ступінь, вчене звання, прізвище, ім'я)

Рецензент _____
(науковий ступінь, вчене звання, прізвище, ім'я)

Київ 2024

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ

Кафедра Інформаційної та кібернетичної безпеки
Ступінь вищої освіти Бакалавр
Спеціальність 125 Кібербезпека
Освітньо-професійна програма Інформаційна та кібернетична безпека

ЗАТВЕРДЖУЮ
Завідувач кафедри ІКБ
_____ Галина ГАЙДУР
“___” _____ 2024 року

З А В Д А Н Н Я
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Люльчуку Нікіті
(прізвище, ім’я)

1. Тема кваліфікаційної роботи: «Дослідження шляхів та розробка рекомендацій щодо захисту web-додатків»

керівник кваліфікаційної роботи Дорохін Орест
(прізвище, ім’я, науковий ступінь, вчене звання)

затверджені наказом Державного університету інформаційно-комунікаційних технологій від «___» _____ 2024 року № ____.

2. Строк подання здобувачем вищої освіти кваліфікаційної роботи 31.05.2024 р.

3. Вихідні дані до кваліфікаційної роботи
Рішення на основі рекомендацій
Наукова та технічна література, експлуатаційна документація, нормативні документи, міжнародні стандарти

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
1. Дослідження проблематики загроз
2. Аналіз існуючих рішень, методів та засобів захисту web-додатків
3.Розробка рекомендацій щодо захисту web-додатків.

5. Перелік графічного матеріалу
Презентація PowerPoint.

6. Дата видачі завдання 22.03.2024 р.

КАЛЕНДАРНИЙ ПЛАН

№ зп	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1.	Визначення актуальності загрози web-додаткам	27.03.2024 р.	
2.	Аналіз існуючих загроз для web-додатків	17.04.2024 р.	
3.	Аналіз існуючих методів виявлення загроз	23.04.2024 р.	
4.	Дослідження існуючих інструментів для захисту web-додатків, з використанням Cloudflare WAF	04.05.2024 р.	
5.	Розроблення рекомендацій щодо застосування Cloudflare WAF для захисту web-додатків	19.05.2024 р.	
6.	Розроблення загальних рекомендацій щодо захисту web-додатків	22.05.2024 р.	
7.	Оформлення результатів дослідження.	25.05.2024 р.	
8.	Підготовка доповіді до захисту.	31.05.2024 р.	

Здобувач вищої освіти _____ Нікіта Люльчук
(підпис) *(ім'я, прізвище)*

Керівник кваліфікаційної роботи _____ Орест Дорохін
(підпис) *(ім'я, прізвище)*

ВІДГУК РЕЦЕНЗЕНТА

здобувача Люльчука Нікіти
на тему: «Дослідження шляхів та розробка рекомендацій щодо захисту web-додатків»

Актуальність: Захист web-додатків є критично важливим завданням для будь-якої організації, яка використовує інтернет-технології для надання своїх послуг. Зростаюча кількість кібератак на web-додатки робить цю тему надзвичайно актуальною. Важливість розробки ефективних заходів захисту web-додатків важко переоцінити, оскільки вони забезпечують безпеку конфіденційної інформації, безперервність бізнес-процесів і довіру користувачів.

Позитивні сторони:

1. Робота чітко визначає основні проблеми, пов'язані із захистом web-додатків, і формулює мету та завдання дослідження, що свідчить про глибоке розуміння теми.
2. У дослідженні представлені сучасні методи захисту web-додатків, включаючи використання рішень на базі Cloudflare WAF, що забезпечують високий рівень безпеки.
3. Запропоновані рекомендації щодо покращення захисту web-додатків є практичними та можуть бути застосовані в реальних умовах. Робота базується на реальних прикладах кіберінцидентів, що робить її особливо цінною для практиків.
4. Текст роботи викладено чітко та логічно. Висновки сформульовано ясно та змістовно, що полегшує їхнє розуміння та подальше застосування. Графічні матеріали оформлено якісно, що сприяє кращому сприйняттю інформації.

Недоліки:

1. У роботі бажано було б більш детально розглянути конкретні приклади кібератак на web-додатки та їхній аналіз, включаючи розбір зловмисного коду та скриптів, використаних атакувальниками. Це дозволило б поглибити розуміння механізмів атак і запропонувати ще більш ефективні засоби захисту.

Відзначене зауваження не впливає на загальну позитивну оцінку кваліфікаційної роботи.

Висновок: Враховуючи недоліки, кваліфікаційна робота заслуговує оцінку **"добре"**, а здобувач(ка) **ЛЮЛЬЧУК Нікіта** – присвоєння кваліфікації бакалавр з кібербезпеки за спеціалізацією Інформаційна та кібернетична безпека.

Рецензент:

_____	_____	_____
(науковий ступінь, вчене звання)	*(підпис)*	*(ім'я, прізвище)*

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ

ПОДАННЯ
ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ
ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ

на здобуття освітнього ступеня бакалавра

Направляється здобувач Люльчук Нікіта до захисту кваліфікаційної роботи
(прізвище, ім'я)

спеціальності 125 Кібербезпека
освітньо-професійної програми Інформаційна та кібернетична безпека
(шифр і назва спеціальності)

на тему: «Дослідження шляхів та розробка рекомендацій щодо захисту web-додатків»

Кваліфікаційна робота і рецензія додаються.

Директор інституту _____ Віталій САВЧЕНКО
(підпис) *(ім'я, прізвище)*

Висновок керівника кваліфікаційної роботи

Здобувач ***ЛЮЛЬЧУК Нікіта*** обрав тему роботи, метою якої було дослідити методи та засоби розробки шляхів щодо захисту web-додатків. Перелік використаних джерел свідчить про вміння здобувачем розбиратись в наукових питаннях та застосовувати їх при дослідженнях. Під час виконання кваліфікаційної роботи ***ЛЮЛЬЧУК Нікіта*** показав добру теоретичну та практичну підготовку, вміння самостійно вирішувати питання і робити висновки. Роботу виконував сумлінно, акуратно та вчасно за планом.

Все це дозволяє оцінити виконану кваліфікаційну роботу здобувача **Люльчука Нікіти** на оцінку **"добре"** та присвоїти йому кваліфікацію бакалавр з кібербезпеки за спеціалізацією Інформаційна та кібернетична безпека.

Керівник кваліфікаційної роботи _____ Орест ДОРОХІН
(підпис) *(ім'я, прізвище)*
"____" _____ 2024 року

Висновок кафедри про кваліфікаційну роботу

Кваліфікаційна робота розглянута. Здобувач **Люльчук Нікіта** допускається до захисту даної кваліфікаційної роботи в Екзаменаційній комісії.

Завідувач кафедри Інформаційної та кібернетичної безпеки
(назва)

_____ Галина ГАЙДУР
(підпис) *(ім'я, прізвище)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи 42 сторінки, 22 рисунки, 4 таблиці

Об'єкт дослідження – процес захисту web-додатків.

Предмет дослідження – методи та засоби підвищення захищеності web-додатків.

Мета роботи - дослідити сучасні методи захисту web-додатків та розробити рекомендації для підвищення їхньої захищеності.

Методи дослідження – опрацювання літератури за даною темою, аналіз експлуатаційної документації, міжнародних стандартів та їх порівняння.

Забезпечення захищеності web-додатків є актуальною та важливою проблемою в сучасному світі, оскільки велика кількість організацій та підприємств стикаються з постійно зростаючими кіберзагрозами та потребами в ефективному захисті своїх інформаційних систем.

В роботі досліджено проблему захищеності web-додатків. Проведено аналіз основних загроз для web-додатків. Проаналізовано сучасні методи та засоби захисту web-додатків.

На основі досліджень, проведених у роботі, запропоновано комплекс заходів для підвищення захищеності web-додатків. Розроблено рекомендації щодо впровадження цих заходів

Галузь використання – кібербезпека інформаційних ресурсів організації.

WEB-ДОДАТКИ, БЕЗПЕКА WEB-ДОДАТКІВ, ТЕХНОЛОГІЇ ЗАХИСТУ WEB-ДОДАТКІВ

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

XSS – Cross-Site Scripting

SQL – Structured Query Language

CSRF – Cross-Site Request Forgery

URL – Uniform Resource Locator

ORM – Object-Relational Mapping

CSP – Content Security Policy

SIEM – Security Information and Event Management

WAF – Web Application Firewall

HTTPS – Hypertext Transfer Protocol Secure

SSL/TLS – Secure Sockets Layer / Transport Layer Security

OAuth – Open Authorization

JWT – JSON Web Token

HTTP – Hypertext Transfer Protocol

IDS/IPS – Intrusion Detection System / Intrusion Prevention System

2FA – Two-Factor Authentication

MITRE – The MITRE Corporation

SANS – SysAdmin, Audit, Network, and Security Institute

ВСТУП

Актуальність дослідження. Забезпечення захищеності web-додатків є надзвичайно важливою проблемою в сучасному цифровому світі. Організації та підприємства стикаються зі зростаючими кіберзагрозами та постійно зростаючими потребами в надійному захисті своїх інформаційних систем. Водночас, швидкий розвиток технологій та поява нових вразливостей ускладнюють завдання захисту web-додатків.

Забезпечення безпеки web-додатків є ключовим завданням для організацій та розробників, які прагнуть захистити конфіденційні дані користувачів та забезпечити надійність функціонування своїх систем. Враховуючи цей факт, розробка ефективних рекомендацій щодо підвищення захищеності web-додатків стає особливо актуальною.

До забезпечення захищеності web-додатків ставляться численні вимоги щодо ефективності, надійності та відповідності сучасним стандартам безпеки. Розробка рекомендацій для захисту web-додатків є важливим завданням, яке допоможе забезпечити високий рівень захисту та задовольнити потреби користувачів.

Методи дослідження, такі як опрацювання літератури, аналіз експлуатаційної документації та міжнародних стандартів, дозволяють отримати об'єктивну інформацію про можливі загрози та методи захисту web-додатків, сприяючи ефективному прийняттю рішень при розробці рекомендацій щодо їхньої захищеності.

Дане дослідження має значне практичне значення для вирішення актуальних проблем у сфері кібербезпеки. Дослідження допоможуть організаціям підвищити захищеність своїх web-додатків та забезпечити їхню надійність в умовах зростаючих кіберзагроз.

Об’єкт дослідження – захищеність web-додатків.

Предмет дослідження – методи та засоби підвищення захищеності web-додатків.

Мета роботи – дослідити сучасні методи захисту web-додатків та розробити рекомендації для підвищення їхньої захищеності.

Наукові завдання:

дослідити сутність проблеми захищеності web-додатків;

проаналізувати сучасні методи та засоби захисту web-додатків;

проаналізувати існуючий стан безпеки web-додатків;

запропонувати варіанти підвищення захищеності web-додатків;

розробити рекомендації щодо впровадження заходів захисту.

Методи дослідження – опрацювання літератури за даною темою, аналіз експлуатаційної документації, міжнародних стандартів та їх порівняння.

Практичне значення одержаних результатів: розроблено рекомендації щодо підвищення захищеності web-додатків, що дозволить забезпечити високий рівень їхньої безпеки та надійності.

1 АНАЛІЗ ЗАГРОЗ БЕЗПЕКИ WEB-ДОДАТКІВ

1.1 Аналіз роботи web-додатків

Досягнення в галузі розробки web-додатків у поєднанні з бізнес-середовищем, що змінюється, означають, що web-додатки стають все більш поширеними в корпоративній, державній та урядовій сферах послуг.

Web-додатки можуть забезпечити зручність та ефективність у сфері послуг, вони також можуть створювати низку нових загроз безпеці, які потенційно можуть становити значні ризики для інфраструктури інформаційних технологій організації, якщо їх не захищати належним чином.

Організації залежать від заходів безпеки для захисту IT-інфраструктури. Проте традиційні заходи безпеки та технології можуть бути недостатніми для захисту мережі від нових загроз, оскільки атаки тепер спеціально націлені на недоліки безпеки в дизайні web-додатків.

Нові заходи безпеки, мають бути реалізовані поряд із розробкою web-додатків. Щоб усунути загрози, пов'язані з web-додатками, важливо розуміти вразливості, які зазвичай зустрічаються у web-додатках і процес розробки web-додатків.

Необхідно знати, як працюють web-додатки, для того щоб їх можна захистити. Розуміння механізмів їх функціонування дозволяє не тільки ефективно використовувати ці технології, але й виявляти потенційні вразливості, що можуть бути використані зловмисниками, і розробляти ефективні методи захисту.

Web-додатки складаються з клієнтської та серверної частин, кожна з яких виконує свої функції і має свої особливості, що потребують окремого аналізу.

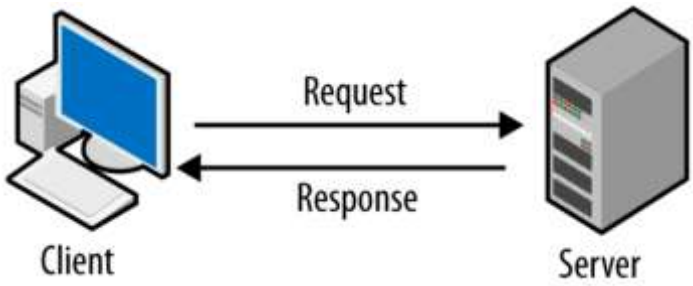


Рис. 1.1. Архітектура клієнт-сервер

Клієнтська частина web-додатка – це те, що бачить і з чим взаємодіє користувач у своєму браузері. Це інтерфейс користувача, створений за допомогою таких мов розмітки та стилів, як HTML (HyperText Markup Language) і CSS (Cascading Style Sheets), а також мов програмування, таких як JavaScript. HTML відповідає за структуру сторінок, CSS забезпечує їх стильове оформлення, а JavaScript додає динамічність і інтерактивність, роблячи додаток живим і функціональним. JavaScript дозволяє змінювати контент веб-сторінки без перезавантаження, створювати інтерактивні елементи, такі як форми, кнопки, анімації, та реагувати на дії користувача в реальному часі.

JavaScript відкриває широкі можливості для web-додатків, дозволяючи реалізовувати складні функціональні можливості, включаючи валідацію даних, асинхронні запити до сервера через AJAX, динамічне оновлення контенту. Web-додатки часто використовуються різні бібліотеки та фреймворки JavaScript, такі як React, Angular, Vue.js, які значно спрощують і прискорюють розробку складних інтерфейсів.

Серверна частина web-додатка виконує всі основні обчислення та обробку даних. Вона обробляє запити від клієнта, взаємодіє з базою даних і повертає відповідні відповіді клієнту. Сервер може бути написаний на різних мовах програмування, таких як PHP, Python, Java або Node.js.

Сервер виконує такі функції: обробка HTTP-запитів, виконання бізнес-логіки додатка, взаємодія з базою даних для збереження та отримання інформації, та формування HTML-сторінок або даних у форматі JSON чи XML у відповідь на запити клієнтів.

Серверна частина також відповідає за забезпечення безпеки даних та користувачів, що взаємодіють з додатком.

Взаємодія клієнта та сервера у web-додатках базується на протоколі HTTP (Hypertext Transfer Protocol). HTTP протокол, визначає правила обміну даними між клієнтом і сервером. Клієнт відправляє HTTP-запит до сервера, сервер обробляє запит і відправляє відповідь назад клієнту. Цей процес може бути синхронним або асинхронним, залежно від специфіки web-додатка.

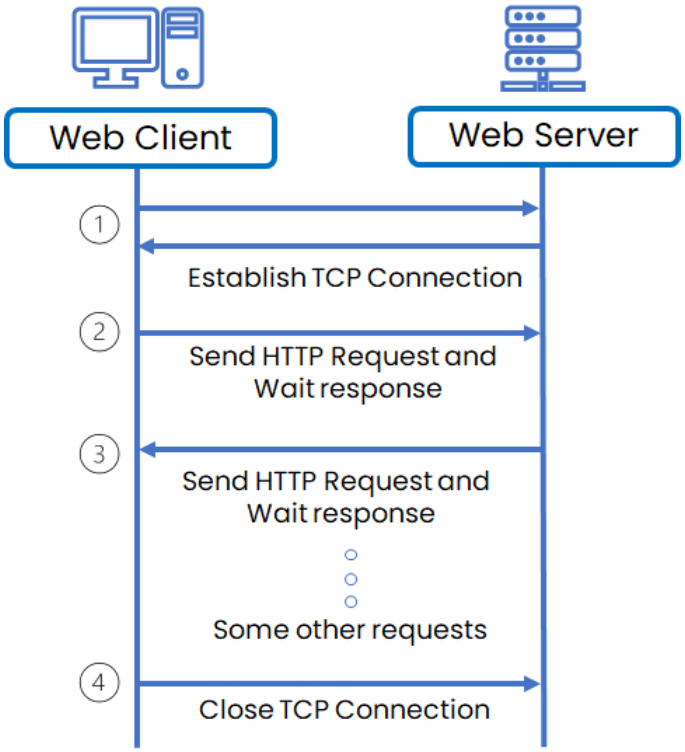


Рис. 1.2 Принцип роботи HTTP протоколу

Роль у взаємодії клієнта та сервера відіграють API (Application Programming Interface), які дозволяють різним програмним компонентам взаємодіяти між собою. RESTful API (Representational State Transfer) є найбільш поширеним способом реалізації API для web-додатків. RESTful API дотримується принципів REST, використовуючи стандартні HTTP методи, такі як GET, POST, PUT та DELETE, для обміну ресурсами, та дозволяє створювати прості та ефективні інтерфейси, які легко інтегрувати з іншими системами. Кожен ресурс у RESTful API представлений унікальним URL, що спрощує навігацію та управління даними.

Рис. 1.3. API яке взаємодіє з клієнтською частиною

Основними перевагами web-додатків є їх доступність та кросплатформеність. Вони не потребують встановлення на пристроях користувачів, легко оновлюються та можуть працювати на будь-якому пристрої з браузером і доступом до Інтернету. Переваги які були перераховані раніше роблять web-додатки ідеальними для використання у мобільних умовах, дозволяючи користувачам працювати з будь-якого місця і в будь-який час.

1.2 Аналіз найпоширеніших атак на web-додатки

Web-додатки є дуже популярними і ними можуть користуватися мільйони користувачів, необхідно знати які можуть виникнути вразливості при недотриманні рекомендацій по їх захисту.

Вразливості у web-додатку – це помилки у коді, які були допущені на стадії розробки. Ці вразливості є потенційними точками входу для зловмисників, які можуть використовувати їх для компрометації системи. За допомогою таких вразливостей, зловмисники можуть отримати конфіденційні дані користувачів додатку, зокрема паролі, фінансову інформацію та особисті дані, що може мати серйозні наслідки для приватності та безпеки користувачів. Крім того, зловмисники можуть спробувати нашкодити певному підприємству шляхом видалення або модифікації бази даних клієнтів, що може призвести до значних фінансових втрат та пошкодження репутації компанії.

Доволі часто доступ до web-додатку зловмисники отримують через погану обробку даних, що надходять від користувача. Наприклад, якщо введені дані не перевіряються належним чином, зловмисник може передати шкідливий код через поле вводу, і цей код буде виконуватися на сервері або у браузері користувача. Це може призвести до критичних наслідків, включаючи крадіжку даних, несанкціонований доступ до системи або навіть повну компрометацію додатку.

Існує багато загроз для web-додатків, тому розглянемо кожну атаку більш детальніше, щоб краще з ними ознайомитись .

SQL ін'єкції

Одна з найпоширеніших та найнебезпечніших вразливостей web-додатків. Вона виникає, коли зловмисники вводять шкідливий SQL-код у запити до бази даних через слабкі місця у введенні даних у web-додатку. Використовуючи різноманітні методи, зловмисники можуть впроваджувати шкідливий код через веб-форми, URL-параметри або навіть HTTP-заголовки. Вразливі місця у web-додатку, які не забезпечують належної перевірки введених даних, відкривають доступ зловмисникам до бази даних, що може призвести до витоку конфіденційної інформації, модифікації або видалення даних.

Однією з головних небезпек SQL ін'єкцій є можливість витягування чутливих даних. Зловмисники можуть отримати доступ до паролів користувачів, фінансової інформації або особистих даних, що може мати серйозні наслідки для безпеки користувачів. Наприклад, зловмисники можуть змінювати дані в базі, порушуючи роботу додатку, або видаляти цілі таблиці, що може паралізувати діяльність компанії. У деяких випадках, SQL ін'єкції можуть дозволити зловмисникам виконувати довільні команди на сервері бази даних, отримуючи повний контроль над системою.

Основні методи, що використовуються зловмисниками для впровадження шкідливого коду, включають введення через веб-форми або URL-параметри. Наприклад, зловмисники можуть вводити шкідливий код у поля вводу форми входу або реєстрації. Якщо ці дані не перевіряються належним чином, сервер виконає шкідливий код. Аналогічно, додавання шкідливого SQL-коду до URL-параметрів,

таких як User-Agent або Referer, може призвести до виконання коду на сервері бази даних.

Кросс-сайтовий скриптинг (XSS)

XSS-атаки включають різні методи впровадження шкідливого коду які потім застосовуються в браузері користувача.

XSS дозволяє зловмисникам впроваджувати клієнтські сценарії у загальнодоступні веб-сторінки і в багатьох випадках може використовуватися зловмисниками для обходу контролю доступу. Це робиться шляхом обману змусити браузер приймати дані з ненадійного джерела, і це зазвичай відбувається, коли зловмисники використовують знайомий код (наприклад JavaScript). Успішна атака XSS може дозволити зловмиснику отримати доступ до сесійних файлів cookie, перенаправляти користувачів до шкідливих сайтів. Шкідливий скрипт може зберігатися на сервері і буде відображатися всім користувачам, які будуть відкривати скомпроментовану сторінку.

Web-додатки, які дозволяють користувачеві вводити дані без повного контролю над виведенням, можуть сильно постраждати від XSS-атак. Якщо XSS-атака успішна, зловмисники можуть завдати серйозної шкоди web-додатку.

Міжсайтова підробка запитів

Міжсайтова підробка запитів, або CSRF (Cross-Site Request Forgery), є вишуканим методом маніпуляції, що експлуатує авторизацію web-додатків. Зловмисники можуть створювати підроблені HTTP-запити, які будуть відправлятися від імені авторизованого користувача, якщо той використовує недостатньо захищений додаток. Ця вразливість дозволяє зловмисникам виконувати небажані дії на сервері, вводячи в оману web-додаток, ніби ці дії ініційовані автентичним користувачем.

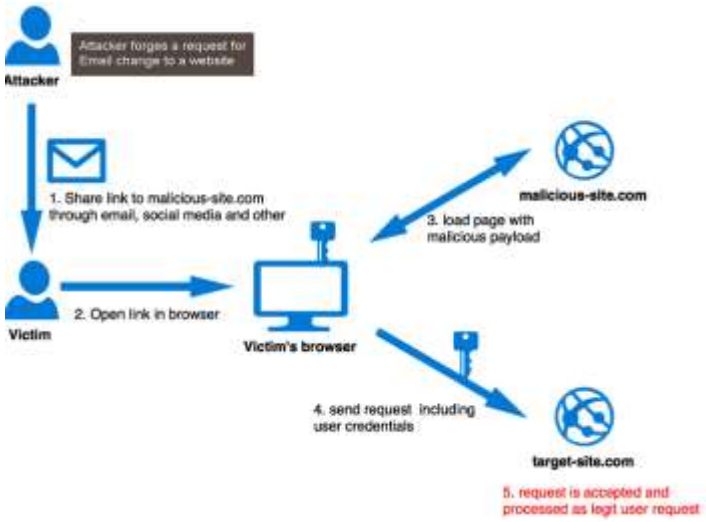


Рис 1.4. Принцип роботи CSRF

Якщо користувач вже увійшов до свого облікового запису в іншому додатку, зловмисник може скористатися цим для виконання небажаних дій. Наприклад, зловмисник може відправити запит на видалення профілю користувача, використовуючи авторизаційні куки цього користувача. Таким чином, дії виконуються від імені користувача без його відома або згоди.

CSRF-атаки можуть бути надзвичайно небезпечними, оскільки вони дозволяють зловмисникам впливати на захищені дії користувачів, такі як зміна налаштувань, здійснення транзакцій або навіть видалення облікових записів. Відсутність належних механізмів захисту робить додаток вразливим до таких атак, створюючи серйозні загрози для безпеки користувачів та цілісності даних.

Відмова в обслуговуванні (DoS) та розподілена відмова в обслуговуванні (DDoS)

Ці дві атаки спрямовані на відключення web-додатку, щоб зробити його недоступним для користування звичайними користувачами. DoS використовує відправку великої кількості запитів, щоб перегрузити сервер и зробити його недоступним для використання.

Відмінно від DoS атак, DDoS буде використовувати ботнети або велику кількість комп'ютерів для координованого нападу на сервер та виведення його зі строю.

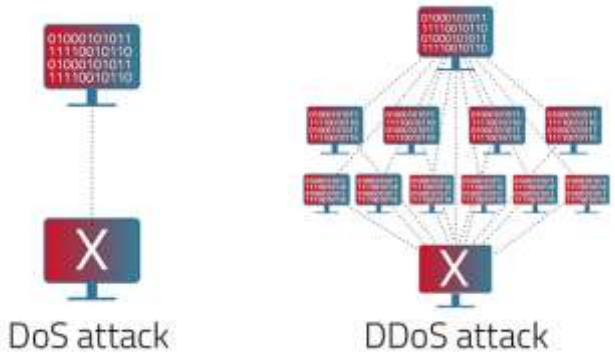


Рис. 1.5. Різниця між DoS та DDoS

Неправильна автентифікація та керування сессіями.

Якщо функціональні можливості програми неправильно реалізовані, злочинець може зламати або скомпрометувати паролі, ідентифікатори сеансів та інші вразливості, використовуючи вкрадені облікові дані. Сеанси повинні бути унікальними для кожного окремого користувача, і без будь-якого необхідного керування сеансами зловмисник може замаскуватися під користувача та вкрасти токени та паролі, щоб отримати доступ до нього.

Окрім основних технічних загроз які зустрічаються найчастіше, інснують також загрози які можуть виникнути за рахунок людського фактору.

Наприклад, використання при розробці старої версії певної бібліотеки або фреймворку який не оновлюється, можуть мати в собі велику кількість не вирішених вразливостей, якими можуть скористатися зловмисники та нашкодити web-додатку. Ще одним видом загрози безпеці web-додатку може стати використання, неофіційних або непопулярних залежностей які будуть використовуватися при розробці web-додатку, вони також можуть налічувати багато невирішених проблем з безпекою у коді.

Таблиця 1.1

Існуючі загрози

Тип вразливості	**Наслідки**
SQL ін'єкції	Отримання доступу до конфіденційної інформації, видалення або зміна даних.
Кросс-сайтовий скриптинг (XSS)	Отримання доступу до сесійних файлів cookie, перенаправлення користувачів на шкідливі сайти.
Міжсайтова підробка запиту (CSRF)	Виконання дій від імені користувача без його відома, наприклад, видалення профілю користувача.
DoS та DDoS	Недоступність web-додатку через перегрузку сервера.
Неправильна автентифікація та керування сесіями	Отримання доступу до акаунтів користувачів, викрадення токенів та паролів
Використання старих версій бібліотек або фреймворків	Зловмисники можуть скористатися відомими вразливостями для нанесення шкоди web-додатку.
Використання неофіційних або непопулярних залежностей	Web-додаток може бути вразливим до атак через невирішені проблеми з безпекою у коді залежностей.

1.3 Аналіз методів та інструментів для виявлення уразливостей

Вразлива система web-додатку може бути використана зловмисниками і зловмисник може отримати контроль над системою та пошкодити її, почати нові атаки або отримати певну привілейовану інформацію, яку він може використовувати для власної вигоди. Зважаючи на це, важливо знати різні запобігання та виявлення, щоб спробувати уникнути присутності певних

вразливостей в остаточної системи web-додатку, а потім зменшити можливість атак і дорогі збитки.

Щоб забезпечення безпеку web-додатків необхідно своєчасно виявляти та усувати вразливості. Цей процес здійснюється за допомогою різних методів та інструментів, які доповнюють один одного і забезпечують комплексний підхід до захисту.

Серед основных методів та інструментів, які можуть використовуватися, можна виділити наступні:

- Використання сканерів безпеки для пошуку вразливостей у web-додатках
- Аналіз коду web-додатка, на виявлення вразливостей ще на етапі розробки
- Пенетраційні тести
- Логування та моніторинг web-додатку

Для виявлення вразливостей у web-додатку використовуються сканери безпеки. Ці інструменти автоматизують процес аналізу web-сторінок та їх компонентів, зосереджуючись на пошуку відомих вразливостей.

За допомогою цих інструментів можна не тільки сканувати web-додаток, але й генерувати детальні звіти, які можуть містити інформацію про знайдені проблеми та рекомендації щодо їх усунення. Така автоматизація дозволяє знизити навантаження на команду розробників та підвищити ефективність процесу забезпечення безпеки.

OWASP ZAP один із найпопулярніших і доступних варіантів для виявлення вразливостей у web-додатках. OWASP ZAP здатний виявляти такі вразливості, як XSS, SQL ін'єкції та інші. ZAP може формувати докладні звіти, що містять інформацію про виявлені проблеми та рекомендації щодо їх усунення. Альтернативою до ZAP є Acunetix, він підтримує автоматичне сканування web-додатків, і може виявляти велику кількість вразливостей, включаючи XSS, SQL ін'єкції та CSRF. Також, надає інтеграції з системами управління вразливостями, що дозволяє управляти процесом усунення вразливостей.

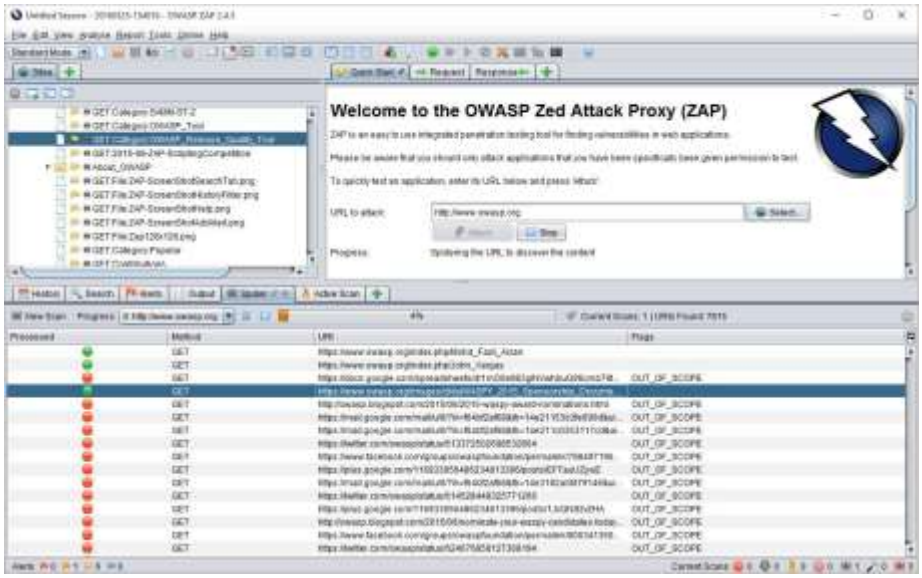


Рис. 1.6. OWASP ZAP

Аналіз вихідного коду, або статичний аналіз коду, є ще одним важливим методом, який дозволяє виявити вразливості на ранніх етапах розробки. Цей метод забезпечує можливість виявлення та усунення проблем ще до того, як код буде розгорнутий у виробничому середовищі, що значно знижує ризик експлуатації вразливостей у готовому продукті.

Наприклад, SonarQube підтримує понад 25 мов програмування та надає можливість аналізувати код на наявність різних типів уразливостей, таких як SQL ін'єкції, XSS, відсутність підтвердження входу, використання застарілих або небезпечних бібліотек. SonarQube генерує докладні звіти, які допомагають розробникам зрозуміти, де проблемні місця в коді та способи їх усунення.

Моніторинг та логування є невід'ємною частиною процесу забезпечення безпеки. Інструменти моніторингу, такі як ELK Stack (Elasticsearch, Logstash, Kibana), збирають та аналізують журнали подій серверів, додатків і мережевих пристроїв. ELK Stack (Elasticsearch, Logstash, Kibana) є популярним набором інструментів для логування та моніторингу. Elasticsearch забезпечує зберігання та пошук даних, Logstash займається збором та обробкою логів, а Kibana надає можливості для візуалізації даних і створення дашбордів. ELK Stack дозволяє виявляти несанкціонований доступ, спроби SQL-ін'єкцій, DDoS-атаки та інші загрози в реальному часі. Завдяки цьому адміністратори можуть швидко реагувати

на інциденти та мінімізувати можливі збитки, забезпечуючи безперервну роботу web-додатків.

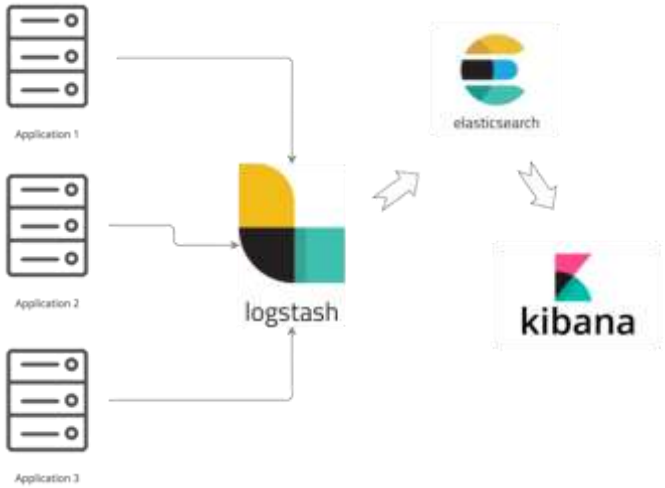


Рис. 1.7. ELK Stack

Тестування на проникнення, є ключовим елементом у процесі забезпечення безпеки web-додатків. Цей метод полягає у симуляції атаки на web-додаток з метою виявлення вразливостей та оцінки їх серйозності. Пенетраційні тестери, або пентестери, використовують як автоматизовані інструменти, так і ручні методи для виявлення слабких.

Автоматизовані інструменти можуть швидко сканувати велику кількість кодів та конфігурацій, що дозволяє швидко виявляти загальні вразливості, такі як SQL-ін'єкції або міжсайтовий скриптінг (XSS). Такі інструменти можуть забезпечити початковий рівень захисту та допомогти зосередитися на більш складних та менш очевидних проблемах. Пентестери можуть виконувати складні атаки, моделюючи дії реальних зловмисників, що дозволяє перевірити, як система поводиться в умовах справжньої атаки.

Таблиця 1.2

Методи для пошуку вразливостей

Метод	**Опис**
Використання сканерів безпеки для пошуку вразливостей у web-додатках	Використання автоматизованих інструментів для перевірки web-додатків на наявність вразливостей.
Аналіз коду web-додатка, на виявлення вразливостей ще на етапі розробки	Перевірка вихідного коду на наявність потенційних вразливостей шляхом статичного або динамічного аналізу.
Пенетраційні тести	Ручне або автоматизоване тестування web-додатків з метою виявлення вразливостей шляхом імітації реальних атак.
Логування та моніторинг web-додатку	Збір та аналіз логів web-додатку для виявлення аномальної активності та можливих атак.

У web-додатків існують різні види вразливостей. Кожна атака яка виникає через вразливість у певної частини web-додатку, має свою особливість та методи протидії. Загалом атаки на web-додатки здійснюються через неправильний підхід до розробки, ігнорування рекомендацій, використання небезпечних бібліотек або через неправильне налаштування проекту.

Великий обсяг інструментів та методів дозволяє виконувати пошук вразливостей, а також допомогає виправляти загрози ще на етапі розробки web-додатків. Тому їх використання має бути обов'язковим для забезпечення захисту та стабільності роботи web-додатків.

2 АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ ЗАХИСТУ WEB-ДОДАТКІВ НА БАЗІ РІШЕННЯ CLOUDFLARE WAF

2.1 Основні функції рішення Cloudflare WAF

Для ефективного захисту web-додатків використовуються різні інструменти та технології, які допомагають виявляти та запобігати атакам.

Одним із таких інструментів є веб-аплікаційні фаєрволи (WAF). Вони є потужним засобом захисту web-додатків від різних атак, які аналізують HTTP-запити та відповіді, застосовуючи правила для виявлення і блокування шкідливого трафіку. Ці фаєрволи діють як бар'єр між web-додатком та Інтернетом, забезпечуючи безпеку та цілісність даних.

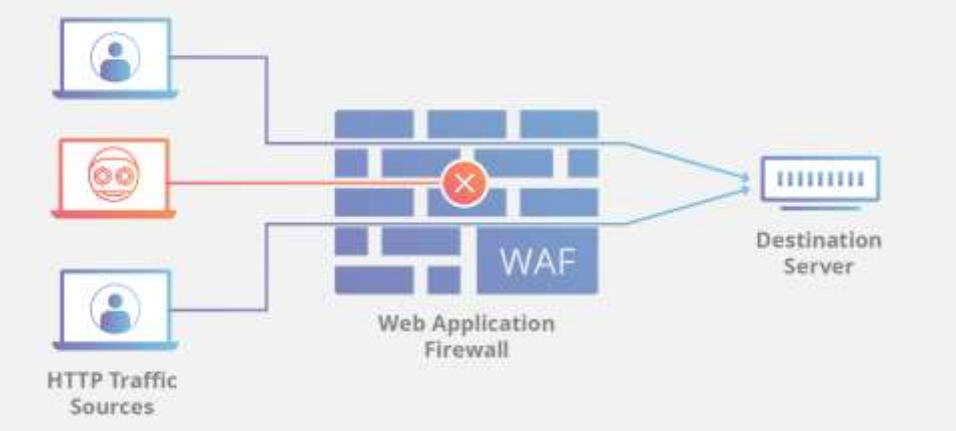


Рис. 2.1. Принцип роботи WAF

Cloudflare WAF - це потужний інструмент, який захищає web-додатки від різноманітних загроз в Інтернеті. Web-додатки часто піддаються атакам, які можуть призвести до компрометації даних, порушення роботи систем та втрати довіри користувачів. Cloudflare WAF (Web Application Firewall) створений спеціально для того, щоб запобігати таким атакам і забезпечувати безпеку web-додатків. Він надає багаторівневий захист, який охоплює широкий спектр потенційних загроз, включаючи SQL-ін'єкції, XSS-атаки, DDoS-атаки та багато інших.

Як частина хмарної платформи Cloudflare, цей WAF інтегрований у глобальну мережу доставки контенту (CDN). Cloudflare CDN забезпечує швидку і надійну доставку контенту користувачам по всьому світу, зменшуючи затримки і підвищуючи продуктивність web-додатків. Інтеграція WAF з CDN дозволяє ефективно фільтрувати і блокувати шкідливий трафік ще до того, як він досягне серверів кінцевих користувачів. Це означає, що потенційні загрози виявляються і нейтралізуються на периферії мережі, що значно знижує ризик успішних атак.

Cloudflare WAF використовує передові алгоритми і методи для аналізу трафіку, виявлення підозрілої активності і блокування шкідливих запитів. Кожен запит до web-додатка проходить через систему перевірки, яка автоматично ідентифікує потенційні загрози на основі заздалегідь визначених правил безпеки. Ці правила постійно оновлюються командою безпеки Cloudflare, яка аналізує нові загрози і впроваджує необхідні зміни для забезпечення актуального захисту.

Крім того, Cloudflare WAF пропонує можливість налаштування користувацьких правил безпеки, що дозволяє адміністраторам адаптувати систему під специфічні потреби кожного web-додатка. Це дає можливість точно визначати, який трафік має бути дозволений, а який заблокований, що підвищує ефективність захисту і забезпечує максимальну безпеку.

Важливою перевагою Cloudflare WAF є його здатність забезпечувати високий рівень захисту без впливу на продуктивність web-додатків. Завдяки інтеграції з CDN, користувачі можуть насолоджуватися швидким і безпечним доступом до контенту, навіть під час масових атак. Це досягається завдяки розподіленій інфраструктурі Cloudflare, яка може ефективно обробляти великий обсяг трафіку і забезпечувати безперебійну роботу web-додатків.

Оновлення правил є одною із дуже користних переваг Cloudflare WAF. Команда безпеки Cloudflare постійно аналізує нові загрози і швидко впроваджує оновлення для своїх правил. Це означає, що користувач завжди матимете актуальний захист без необхідності вручну оновлювати налаштування WAF. Цей процес відбувається автоматично і прозоро для користувачів.

Однією з основних функцій Cloudflare WAF є фільтрація HTTP-запитів. Кожен запит, що надходить до вашого web-додатка, проходить ретельний аналіз. Ця система автоматично ідентифікує та блокує потенційно небезпечні запити, базуючись на заздалегідь визначених правилах безпеки. Такий підхід дозволяє негайно реагувати на загрози і попереджувати проникнення шкідливих даних у вашу систему.

Особливе місце серед функцій Cloudflare WAF займає захист від SQL-ін'єкцій. SQL-ін'єкції є однією з найпоширеніших і водночас небезпечних атак на web-додатки. Вони можуть спричинити витік конфіденційної інформації або навіть повну компрометацію бази даних.

Захист від атак XSS (Cross-Site Scripting) також є однією з ключових функцій Cloudflare WAF. Ці атаки дозволяють зловмисникам впроваджувати шкідливі скрипти у веб-сторінки, що можуть використовуватися для крадіжки даних користувачів або інших зловмисних дій. Cloudflare WAF виявляє і блокує такі спроби, тим самим забезпечуючи безпеку ваших користувачів і їхніх даних.

Cloudflare WAF застосовує складні алгоритми для виявлення і блокування SQL-ін'єкцій та XSS. Одним із основних методів захисту є фільтрація вхідних даних. Всі запити, що надходять до web-додатка, проходять через систему фільтрації, яка аналізує кожен запит на предмет потенційно шкідливого коду. Система автоматично розпізнає шаблони, характерні для ін'єкцій, і блокує їх до того, як вони будуть виконані.

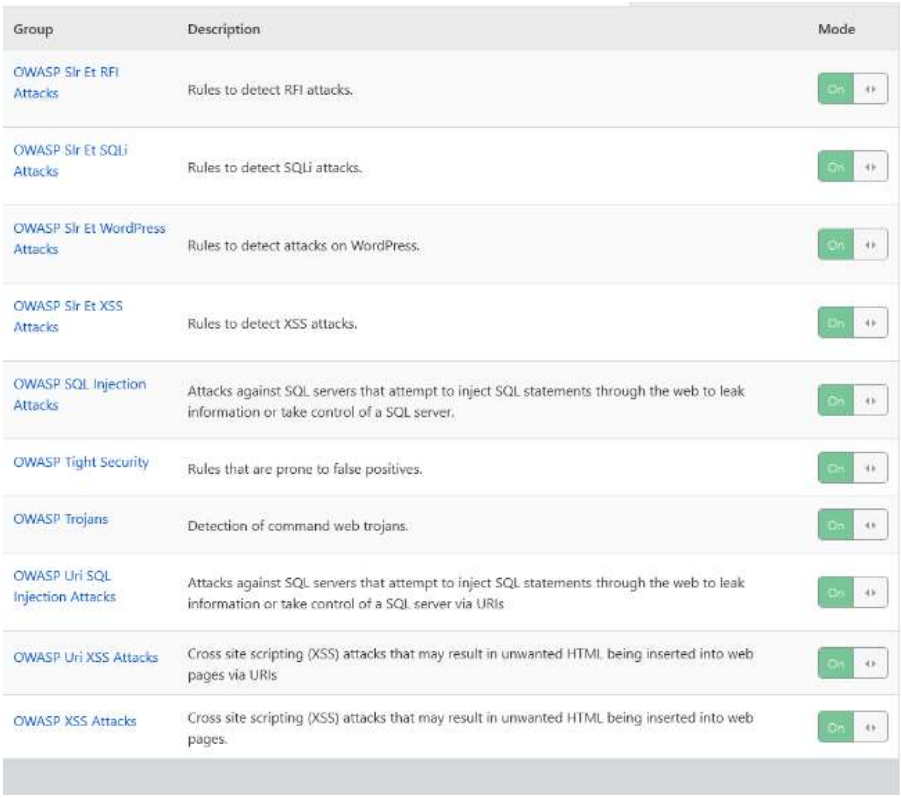
Найбільш потужна функція Cloudflare WAF є захист від DDoS-атак (Distributed Denial of Service). Ці атаки спрямовані на перевантаження web-додатків величезним обсягом запитів, що може зробити їх недоступними для користувачів. Cloudflare WAF здатний виявляти та нейтралізувати такі атаки, зберігаючи стабільність і доступність ваших веб-ресурсів навіть під час масових атак.

Рис. 2.2. Огляд доступних правил для захисту від DDoS

Важливою складовою захисту є використання IP-репутації. Cloudflare WAF використовує глобальну базу даних IP-адрес з поганою репутацією для автоматичного блокування підозрілих запитів. Це значно знижує ризик атак з відомих шкідливих джерел, забезпечуючи додатковий рівень захисту.

Окрему увагу слід приділити захисту від шкідливих ботів, яку також забезпечує Cloudflare WAF. Шкідливі боти можуть здійснювати різноманітні атаки на web-додатки або використовуватися для автоматичного збирання інформації. Cloudflare WAF застосовує складні алгоритми для виявлення та блокування таких ботів, що допомагає захистити ваш web-додаток від автоматизованих загроз.

Крім стандартних функцій, Cloudflare WAF надає можливість створення та налаштування індивідуальних правил безпеки. Це дозволяє адміністраторам адаптувати систему під специфічні потреби кожного web-додатка, забезпечуючи максимальну ефективність захисту.



Group	Description	Mode
OWASP Slr Et RFI Attacks	Rules to detect RFI attacks.	On
OWASP Slr Et SQLi Attacks	Rules to detect SQLi attacks.	On
OWASP Slr Et WordPress Attacks	Rules to detect attacks on WordPress.	On
OWASP Slr Et XSS Attacks	Rules to detect XSS attacks.	On
OWASP SQL Injection Attacks	Attacks against SQL servers that attempt to inject SQL statements through the web to leak information or take control of a SQL server.	On
OWASP Tight Security	Rules that are prone to false positives.	On
OWASP Trojans	Detection of command web trojans.	On
OWASP Uri SQL Injection Attacks	Attacks against SQL servers that attempt to inject SQL statements through the web to leak information or take control of a SQL server via URIs	On
OWASP Uri XSS Attacks	Cross site scripting (XSS) attacks that may result in unwanted HTML being inserted into web pages via URIs	On
OWASP XSS Attacks	Cross site scripting (XSS) attacks that may result in unwanted HTML being inserted into web pages.	On

Рис. 2.3. Набори правил для захисту від вразливостей

Cloudflare WAF дозволяє створювати користувацькі правила для фільтрації трафіку, що є однією з його надзвичайно корисних функцій. У сучасному світі кожна організація стикається з унікальними викликами та загрозами, які можуть відрізнятися залежно від галузі, розміру компанії, географічного розташування та багатьох інших факторів. Саме тому можливість налаштування власних правил для фільтрації трафіку є незамінною для багатьох підприємств.

Для організацій з унікальними вимогами до безпеки або специфічними загрозами, які вони регулярно зустрічають, ця функція стає надзвичайно важливою. Наприклад, деякі компанії можуть працювати з конфіденційними даними клієнтів, такими як фінансові або медичні записи, що потребує додаткового рівня захисту. Інші організації можуть мати справу з підвищеною активністю ботів або іншими типами шкідливого трафіку, які вимагають спеціальних заходів для забезпечення безпеки.

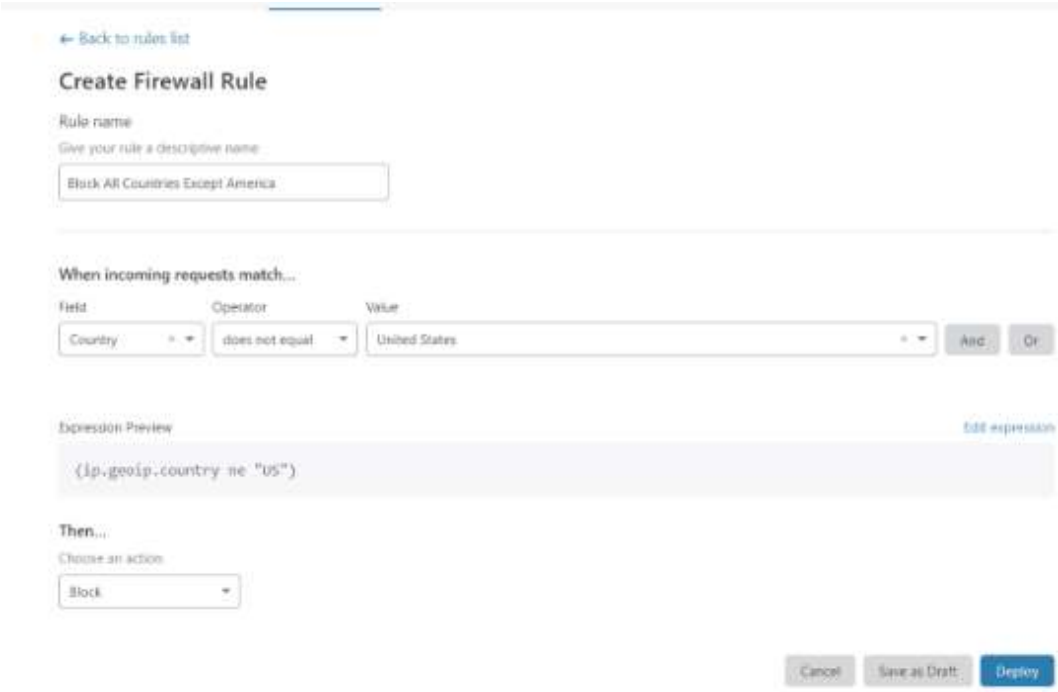


Рис. 2.4. Створення користувацького правила

Можливість легко налаштувати користувацькі правила під конкретні потреби робить захист ще більш персоналізованим і ефективним. Це дозволяє адміністраторам точно визначати, який трафік повинен бути дозволений, а який заблокований, залежно від специфіки їхнього бізнесу. Наприклад, можна налаштувати правила для блокування трафіку з певних географічних регіонів, де зафіксована підвищена активність зловмисників, або встановити спеціальні правила для виявлення і блокування відомих атак на конкретні web-додатки.

В Cloudflare WAF існує інтеграція з іншими продуктами Cloudflare, що дозволяє створити єдину платформу для захисту вашого web-додатка. Ви можете використовувати Cloudflare WAF разом з іншими рішеннями, такими як Cloudflare CDN, Cloudflare Load Balancing та Cloudflare Workers, для створення комплексної стратегії безпеки і оптимізації продуктивності вашого web-додатка.

Також Cloudflare WAF може запропонувати розширені можливості моніторингу та звітності, що дозволяє адміністраторам отримувати детальну інформацію про стан безпеки їх web-додатків. Це включає в себе реальні описи атак, звіти про вразливості та аналітику трафіку.

Таблиця 2.1

Ключові функції Cloudflare WAF

Функція	**Опис**	**Переваги**
Фільтрація HTTP-запитів	Аналізує кожен HTTP-запит, що надходить до web-додатка, для виявлення та блокування потенційно небезпечних запитів.	Миттєва реакція на загрози, запобігання проникненню шкідливих даних.
Захист від SQL-ін'єкцій	Використовує складні алгоритми для виявлення та блокування спроб впровадження шкідливих SQL-запитів.	Захист від витоку конфіденційної інформації та компрометації бази даних.
Захист від XSS-атак (Cross-Site Scripting)	Виявляє та блокує спроби впровадження шкідливих скриптів у веб-сторінки	Запобігання крадіжці даних користувачів та іншим зловмисним діям.
Захист від DDoS-атак (Distributed Denial of Service)	Виявляє та нейтралізує атаки, спрямовані на перевантаження web-додатків великим обсягом запитів.	Підтримання стабільності та доступності веб-ресурсів навіть під час масових атак.
Використання IP-репутації	Застосовує глобальну базу даних IP-адрес з поганою репутацією для автоматичного	Зниження ризику атак з відомих шкідливих джерел, додатковий рівень захисту.

	блокування підозрілих запитів.	
Захист від шкідливих ботів	Використовує складні алгоритми для виявлення та блокування шкідливих ботів.	Запобігання автоматизованим атакам та зниження навантаження на сервери.
Гнучкі правила безпеки	Дозволяє створювати та налаштовувати індивідуальні правила для специфічних потреб web-додатків.	Адаптація системи безпеки до унікальних загроз, максимальна ефективність захисту.

2.2 Інструменти та технології для захисту web-додатків на базі рішення Cloudflare WAF

Cloudflare WAF пропонує інноваційний підхід до забезпечення захисту web-додатків, використовуючи передові інструменти та технології.

Cloudflare WAF, як частина комплексного рішення з безпеки, надає потужні засоби для захисту web-додатків від різноманітних загроз, особливе місце займають інтелектуальні механізми виявлення та запобігання загрозам. Використовуючи машинне навчання та штучний інтелект, Cloudflare WAF здатний аналізувати великий обсяг трафіку в режимі реального часу, ідентифікуючи аномалії та потенційні загрози. Це дозволяє швидко реагувати на нові методи атак і впроваджувати відповідні заходи для їх нейтралізації.

Панель управління Cloudflare. Представляє собою центральний елемент, який дозволяє адміністраторам легко налаштовувати та керувати безпекою своїх web-додатків. Інтерфейс панелі простий у використанні та забезпечує доступ до всіх необхідних функцій, таких як моніторинг активності, налаштування правил безпеки та отримання звітів про стан захисту в режимі реального часу. Це дозволяє

швидко реагувати на потенційні загрози та ефективно управляти безпекою web-додатка.

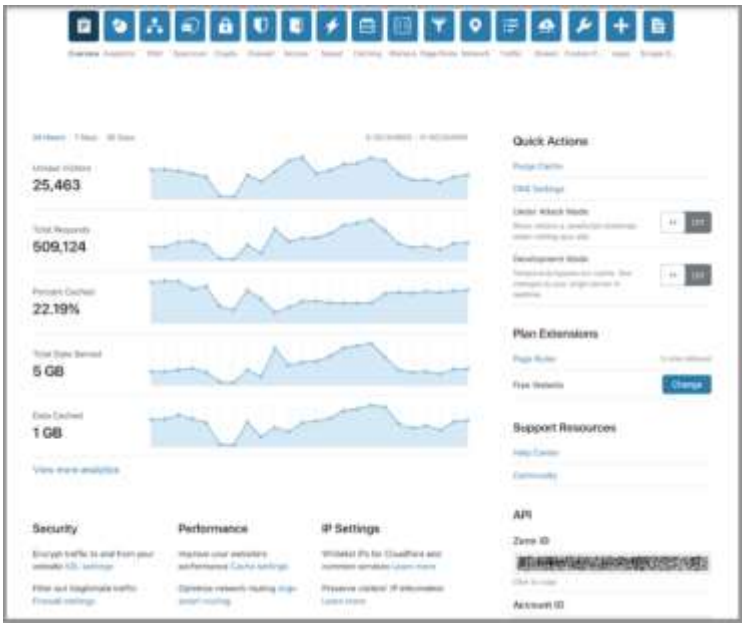


Рис. 2.5. Панель управління Cloudflare

Машинне навчання та штучний інтелект. Однією з передових технологій, що використовується Cloudflare WAF, є машинне навчання. Завдяки машинному навчанню система може аналізувати великі обсяги даних, виявляючи аномальні патерни, що можуть свідчити про потенційні атаки.

Алгоритми здатні постійно вдосконалюватися, адаптуючись до нових загроз і вдосконалюючи свої методи виявлення шкідливої активності. Штучний інтелект, який також застосовується в Cloudflare WAF, дозволяє системі автоматично адаптуватися до нових видів загроз і оновлювати правила безпеки, забезпечуючи постійний захист вашого web-додатка.

Інтелектуальна система збору даних про загрози. Cloudflare WAF використовує систему збору даних про загрози з різноманітних джерел для покращення здатності виявлення та блокування загроз у реальному часі. Система використовує різноманітні джерела для збору даних про загрози, що дозволяє отримувати актуальну інформацію про нові методи атак і вразливості. Завдяки цьому Cloudflare WAF може забезпечувати найвищий рівень захисту для web-додатків, які використовують його рішення.

Cloudflare дозволяє бути на крок попереду потенційних атак, забезпечуючи захист вашого web-додатка від найновіших загроз. Завдяки інтелектуальній системі збору даних про загрози, Cloudflare має змогу швидко ідентифікувати нові методи атак і впроваджувати відповідні заходи для їх нейтралізації. Це означає, що web-додаток завжди знаходиться під захистом найсучасніших технологій безпеки, навіть коли зловмисники використовують нові та складніші методи атак.

Зібрані дані аналізуються та використовуються для постійного оновлення правил безпеки, що дозволяє швидко реагувати на нові загрози. Аналітики та автоматизовані системи Cloudflare постійно обробляють величезні обсяги даних, виявляючи аномальні патерни і потенційні загрози. Ця інформація негайно використовується для оновлення правил безпеки Cloudflare WAF, що забезпечує актуальний захист без необхідності вручного втручання з боку користувачів.

Автоматичне оновлення правил безпеки. Cloudflare WAF регулярно оновлює свої набори правил безпеки на основі новітніх даних про загрози.

Ваш web-додаток завжди буде захищений від нових видів атак, навіть якщо ви не маєте можливості постійно слідкувати за оновленнями. Автоматичні оновлення гарантують, що ваша система безпеки завжди актуальна і відповідає сучасним загрозам.

Обмеження швидкості запитів. Функція обмеження швидкості запитів дозволяє контролювати кількість запитів від певного IP-адреси або користувача за визначений період часу. Це дуже корисний інструмент для запобігання зловмисним атакам, таким як brute force атаки, де зловмисники намагаються вгадати паролі шляхом багаторазових спроб. Обмеження швидкості запитів допомагає знизити ризик надмірного використання ресурсів, що може призвести до уповільнення роботи або навіть падіння web-додатка. Це особливо важливо для запобігання DDoS-атакам (Distributed Denial of Service), які спрямовані на перевантаження сервера великою кількістю одночасних запитів. Застосування цієї функції дозволяє забезпечити стійкість web-додатка до таких атак і підтримувати стабільну роботу сервера.

Функція обмеження швидкості запитів може бути налаштована відповідно до специфічних потреб та вимог кожного web-додатка. Адміністратори можуть встановлювати різні ліміти для різних типів запитів, користувачів або IP-адрес, що дозволяє забезпечити максимальну гнучкість і ефективність управління трафіком. Завдяки цьому, web-додаток залишається захищеним від зловмисних атак і перевантажень, що забезпечує його стабільну і безперебійну роботу.

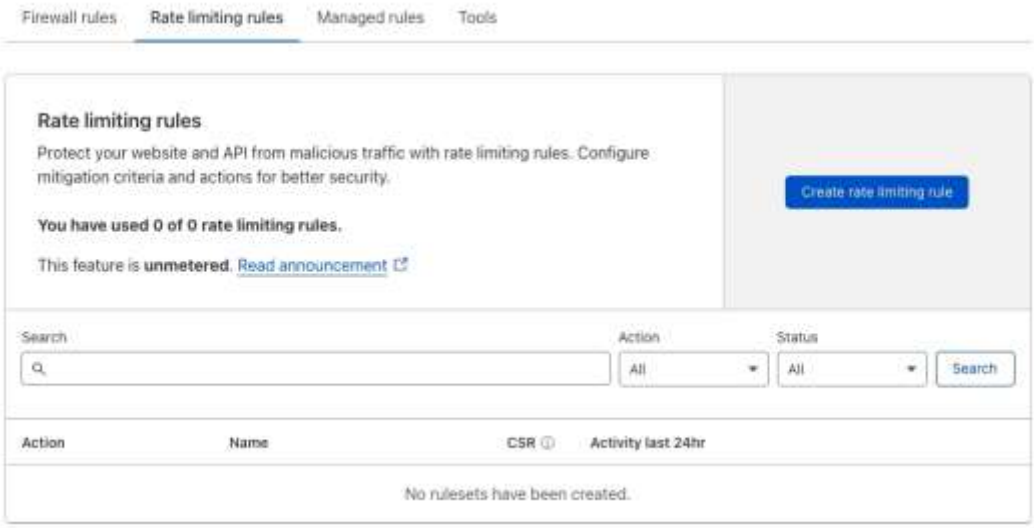


Рис. 2.6. Встановлювання обмеження запитів

Захист API. API Shield забезпечує безпеку інтерфейсів прикладного програмування від зловживань та несанкціонованого доступу. API є важливою частиною багатьох сучасних web-додатків, і їх захист є критично важливим. API Shield дозволяє захистити ваші API від атак, забезпечуючи безпечну взаємодію між системами.

API Shield забезпечує аутентифікацію запитів. Кожен запит до API повинен бути належним чином аутентифікований, щоб гарантувати, що він надходить від довіреного джерела. Аутентифікація може здійснюватися за допомогою різних методів, включаючи API-ключі, токени OAuth або інші механізми, які дозволяють ідентифікувати і перевіряти законність запитів. Такий підхід значно знижує ризик несанкціонованого доступу до вашого API.

Обмеження доступу є важливим компонентом API Shield. Обмеження дозволяє контролювати, хто і що може робити за допомогою API. Можна

налаштувати політики доступу, які обмежують можливості користувачів або систем, що використовують API, на основі їх ролей або інших критеріїв, що створює додатковий рівень безпеки, оскільки навіть у разі компрометації облікового запису зловмисники не зможуть виконати небажані дії.

API Shield надає інструменти для детального моніторингу та аналізу всіх запитів до вашого API. Це дозволяє виявляти підозрілу активність і оперативно реагувати на потенційні загрози. За допомогою моніторингу можна отримувати звіти і сповіщення про несанкціоновані спроби доступу, аномальні запити або інші ознаки можливих атак. Моніторинг активності допомагає підтримувати високий рівень безпеки і забезпечує прозорість роботи вашого API.

API Shield також включає можливості для обмеження швидкості запитів, що допомагає запобігти зловживанням, таким як DDoS-атаки або brute force атаки. Обмеження швидкості дозволяє контролювати кількість запитів, які можуть бути надіслані до вашого API за певний проміжок часу, забезпечуючи стабільну роботу системи навіть під час пікових навантажень.

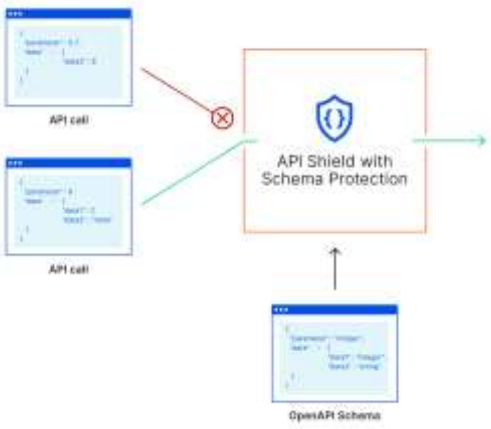


Рис. 2.7. Захист API

Захист DNS-запитів. DNS Security Extensions (DNSSEC) забезпечує захист DNS-запитів, запобігаючи атакам типу "man-in-the-middle" та підробленню DNS-відповідей. Це забезпечує цілісність та безпеку вашого доменного імені, запобігаючи перенаправленню користувачів на шкідливі сайти. DNSSEC є

важливим компонентом безпеки, який допомагає забезпечити правильну маршрутизацію трафіку і запобігти перенаправленню на шкідливі сайти. Важливо зазначити, що впровадження DNSSEC вимагає підтримки з боку як провайдерів DNS, так і власників доменних імен. Це включає налаштування цифрових підписів для зон DNS і перевірку підписів під час обробки запитів. Впровадження DNSSEC може бути складним, але воно забезпечує значний рівень захисту для DNS-запитів.

DNSSEC використовує криптографічні підписи для забезпечення цілісності та автентичності даних DNS. Кожен DNS-запит і відповідь підписуються цифровим підписом, що гарантує, що дані не були змінені або підроблені під час передачі, що забезпечує цілісність та безпеку доменного імені, запобігаючи перенаправленню користувачів на шкідливі сайти.

Cloudflare пропонує підтримку DNSSEC як частину своїх послуг, роблячи цей процес більш доступним і простим для адміністраторів web-додатків. Використання DNSSEC з Cloudflare дозволяє автоматично налаштовувати та підтримувати цифрові підписи для ваших доменів, забезпечуючи постійний захист без додаткових зусиль. Це спрощує процес впровадження та управління DNSSEC, забезпечуючи високий рівень безпеки для web-додатка.

Таблиця 2.2

Інструменти та технології у Cloudflare WAF

Інструмент/Технологія	**Опис**
Інтуїтивно зрозуміла панель управління	Центральний елемент для налаштування та управління безпекою web-додатків.
Машинне навчання та штучний інтелект	Використання передових алгоритмів для виявлення нових та еволюціонуючих загроз.

Інтелектуальна система збору даних про загрози	Система збору та аналізу даних про загрози з різних джерел по всьому світу.
Автоматичні оновлення правил безпеки	Регулярне оновлення наборів правил безпеки на основі новітніх даних про загрози.
Обмеження швидкості запитів	Контроль кількості запитів від певного IP-адреси або користувача за визначений період часу.
Захист API	Безпека інтерфейсів прикладного програмування від зловживань та несанкціонованого доступу.
Захист DNS-запитів	Захист DNS-запитів від атак типу "man-in-the-middle" та підроблення DNS-відповідей.
Гнучкі правила безпеки	Можливість створення та налаштування індивідуальних правил безпеки для фільтрації запитів.

Cloudflare WAF ефективно захищає web-додатки, надаючи комплексні функції, такі як фільтрація HTTP-запитів, захист від SQL-ін'єкцій, XSS-атак та DDoS-атак, використання IP-репутації та гнучкі правила безпеки.

Також він предоставляє великий вибір технологій, та інструментів, для забезпечення захисту сучасних web-додатків. Це дозволяє швидко реагувати на нові методи атак і впроваджувати відповідні заходи для їх нейтралізації, забезпечуючи їхню стабільну та безпечну роботу.

3 РОЗРОБКА РЕКОМЕНДАЦІЙ ЩОДО ПІДВИЩЕННЯ ЗАХИСТУ WEB-ДОДАТКІВ НА БАЗІ РІШЕННЯ CLOUDFLARE WAF

3.1 Рекомендації по покращенню безпеки web-додатків на базі рішення Cloudflare WAF

Використання рішення Cloudflare WAF (Web Application Firewall) дозволяє значно підвищити рівень безпеки web-додатків.

Рекомендується активувати основні захисні правила WAF. Cloudflare WAF пропонує набір готових правил для захисту від найпоширеніших атак, таких як SQL-ін'єкції, XSS (міжсайтовий скриптінг) та CSRF (міжсайтові запити підробки). Ці правила забезпечують базовий рівень захисту від відомих вразливостей і є фундаментом для побудови надійної системи безпеки. Основні захисні правила також охоплюють механізми захисту від інших типів атак, таких як вразливості у файлах cookie, атаки на сеанси користувачів, а також захист від маніпуляцій з параметрами URL. Додатково, правила включають блокування доступу до небезпечних або підозрілих IP-адрес, які можуть бути джерелами шкідливої активності. Завдяки постійним оновленням та автоматичному виявленню нових загроз, Cloudflare WAF забезпечує актуальність захисних правил, що дозволяє оперативно реагувати на нові виклики у сфері кібербезпеки. Активація цих правил є першим і найважливішим кроком у процесі захисту web-додатків, забезпечуючи базовий, але ефективний рівень безпеки від більшості відомих атак.

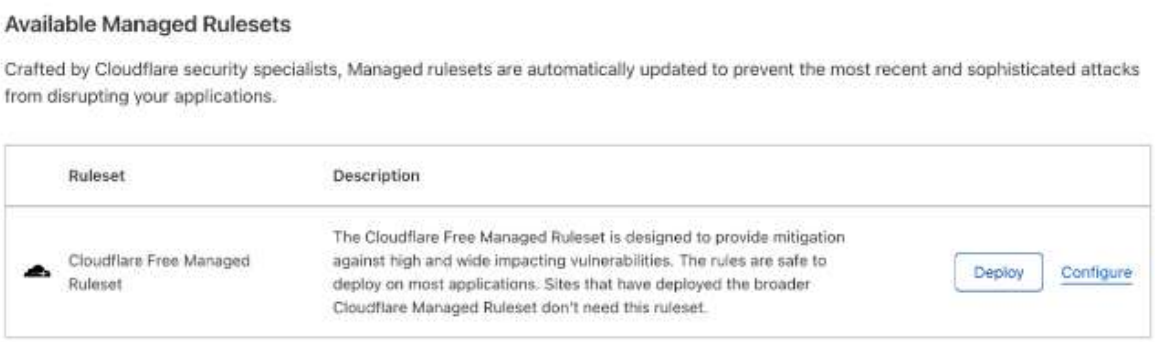


Рис. 3.1. Доступні основні захисні правила

Необхідно налаштувати користувацькі правила WAF відповідно до потреб web-додатку. Cloudflare дозволяє створювати індивідуальні правила для блокування або дозволу певних типів запитів на основі різних параметрів, таких як IP-адреса, геолокація, ідентифікатори запитів. Це дозволяє зосередити захист саме на тих аспектах, які є найбільш важливими для вашої системи. Основні захисні правила Cloudflare WAF включають також захист від інших поширених атак, таких як brute force атаки, спроби експлуатації відомих вразливостей у web-додатках та інші типи шкідливої активності. Використання цих правил допомагає створити міцний базис для безпеки web-додатку, що дозволяє захистити його від найбільш розповсюджених та небезпечних кіберзагроз.

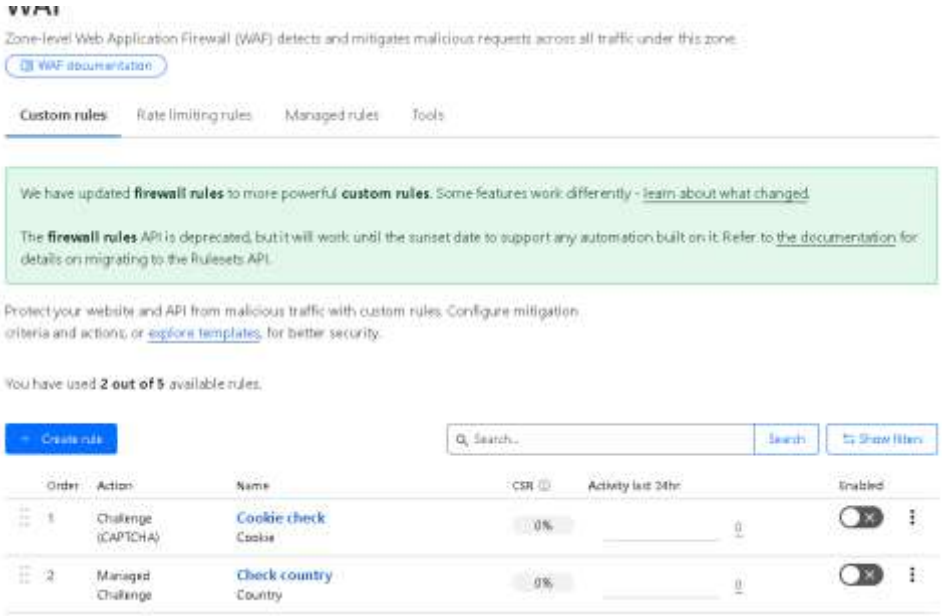


Рис. 3.2. Вже створені користувацькі правила

Важливо своєчасно оновлювати правила та конфігурації Cloudflare WAF, а також проводити регулярні тести безпеки, включаючи тестування на проникнення для виявлення та усунення потенційних вразливостей.

Інтеграція з іншими інструментами безпеки дозволяє забезпечити комплексний підхід до захисту. Наприклад, інтеграція Cloudflare WAF з системами виявлення вторгнень (IDS) або системами управління подіями безпеки (SIEM)

дозволяє об'єднати дані з різних джерел і отримати більш повну картину про стан безпеки вашого web-додатку.

Cloudflare WAF може бути ефективно інтегрований з іншими рішеннями від Cloudflare, такими як захист від DDoS атак, SSL/TLS шифрування та автоматичне обмеження швидкості. Використання цих інструментів разом створює багатошарове захисне середовище, яке значно підвищує загальний рівень безпеки.

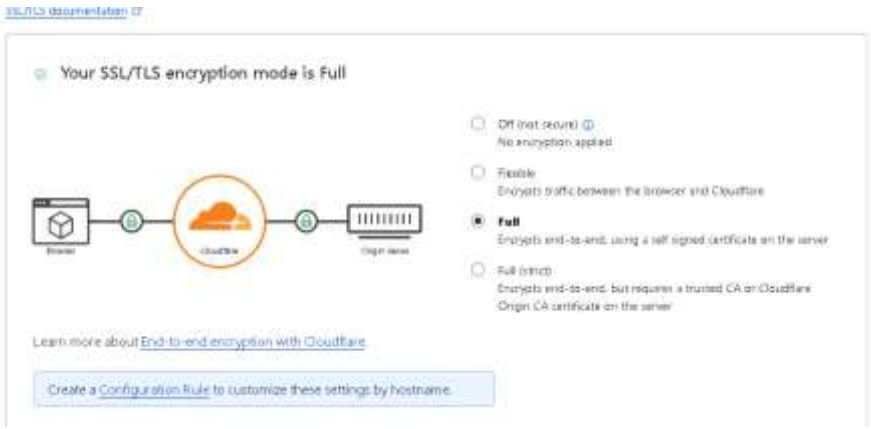


Рис. 3.3. Включення шифрування

Інтеграція з інструментами управління доступом та автентифікації дозволяє забезпечити контроль доступу до чутливих даних та ресурсів, що додатково підвищує загальний рівень безпеки. Використання двофакторної автентифікації (2FA), яка додає додатковий рівень захисту, вимагаючи від користувачів введення додаткового коду, отриманого на мобільний пристрій або через електронну пошту.

Управління ролями та дозволами користувачів дозволяє чітко визначити, які дії можуть виконувати різні категорії користувачів, що мінімізує ризики несанкціонованого доступу до критично важливих даних.

Додатково, впровадження сучасних протоколів автентифікації, таких як OAuth і SAML, сприяє безпечному управлінню доступом та автентифікацією, забезпечуючи єдину точку входу та спрощуючи процеси авторизації. Також це знижує ризик компрометації облікових даних, оскільки користувачам не потрібно запам'ятовувати кілька паролів для різних систем.

Інтеграція з інструментами управління доступом включає моніторинг активності користувачів та ведення журналу подій. Це дозволяє виявляти та

реагувати на підозрілу активність, наприклад, несанкціоновані спроби входу або доступ до конфіденційних даних. Автоматичне повідомлення адміністраторів про такі інциденти допомагає своєчасно реагувати на потенційні загрози та вживати необхідних заходів для їх нейтралізації.

3.2 Рекомендації щодо застосування методів та засобів захисту web-додатків

Забезпечення безпеки web-додатків вимагає всебічного підходу, що охоплює різні аспекти розробки та впровадження. Важливо інтегрувати заходи безпеки на кожному етапі створення web-додатку, починаючи від планування і закінчуючи підтримкою та оновленням.

Шифрування даних відіграє важливу роль у забезпеченні конфіденційності та цілісності інформації. Використання сучасних протоколів, таких як TLS 1.3, забезпечує захист даних під час їх передачі між клієнтом і сервером. TLS 1.3 усуває деякі відомі вразливості попередніх версій і покращує продуктивність завдяки оптимізації процесу встановлення з'єднання. Шифрування даних на рівні зберігання за допомогою алгоритмів AES (Advanced Encryption Standard) гарантує, що навіть у разі компрометації фізичних носіїв, дані залишаться захищеними. Використання SHA-256 для хешування паролів забезпечує їх надійне зберігання, що робить неможливим їх відновлення у разі витоку бази даних.

Для додаткового захисту варто використовувати апаратні модулі безпеки (HSM) для зберігання криптографічних ключів. HSM забезпечують надійне зберігання ключів та запобігають їх несанкціонованому використанню. Регулярне оновлення алгоритмів шифрування та ключів є важливим для підтримки актуальності захисту. Наприклад, заміна ключів шифрування кожні кілька місяців дозволяє зменшити ризик компрометації у разі витоку ключів.

Використання інструментів логування наприклад таких як Sentry може стати важливим для забезпечення безпеки та стабільної роботи web-додатків. Sentry – це потужна платформа для моніторингу помилок і логування, яка допомагає

відстежувати, фіксувати та реагувати на різноманітні помилки у їхніх додатках у реальному часі.

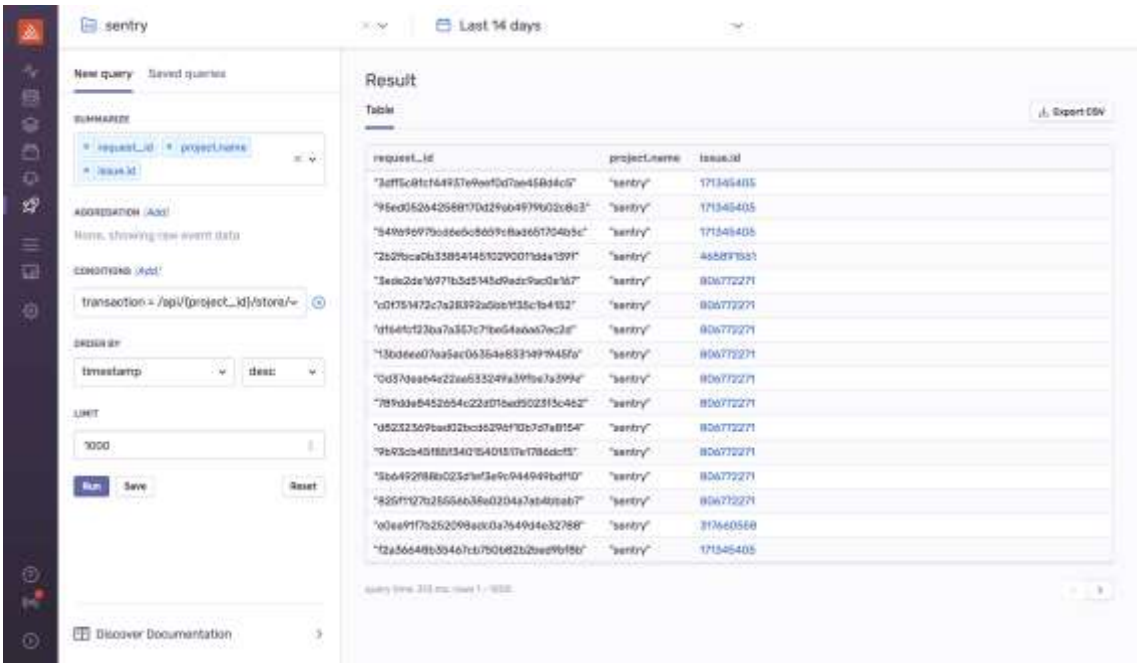


Рис. 3.4. Логи Sentry

Sentry забезпечує автоматичне виявлення помилок у web-додатках. Виявлені помилки усуваються та логуються, що дозволяє отримувати інформацію про збої та неполадки web-додатків. Такий підхід важливий для швидкого реагування на критичні проблеми та мінімізації негативних наслідків.

Кожна зафіксована помилка включає детальну інформацію, таку як стек викликів, контекст виконання, середовище, у якому сталася помилка, та інші важливі дані. Детальна інформація допомагає розробникам швидко зрозуміти причину помилки і знайти ефективне рішення.

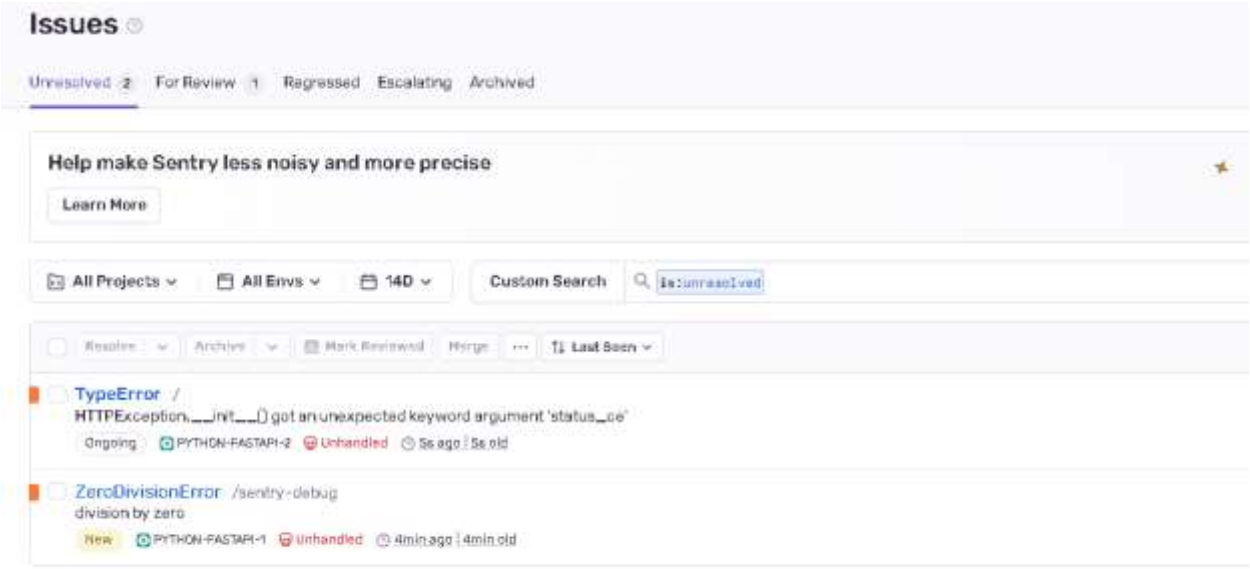


Рис. 3.5. Огляд помилок

Sentry також агрегує та аналізує помилки, виявляючи повторювані проблеми. Це дозволяє розробникам визначати найбільш поширені та критичні помилки, що потребують негайної уваги.

Логи, зібрані Sentry, можуть бути використані для проведення аудиту та розслідування інцидентів безпеки. Це допомагає виявити вразливості, які були використані зловмисниками, та забезпечити їхнє усунення для запобігання повторним атакам. Аналіз помилок дозволяє виявляти слабкі місця у коді та архітектурі додатка, сприяючи постійному вдосконаленню якості коду та підвищенню стійкості системи до потенційних атак.

Регулярне оновлення програмного забезпечення є критично важливим для підтримання безпеки web-додатків. Оновлення повинні включати не тільки операційну систему та серверне програмне забезпечення, але й усі компоненти web-додатку, включаючи бібліотеки та фреймворки. Використання автоматизованих систем управління оновленнями, таких як пакетні менеджери, дозволяє забезпечити своєчасне встановлення останніх версій програмного забезпечення. Це знижує ризик експлуатації відомих вразливостей, які можуть бути використані зловмисниками для атаки на додаток. Варто налаштувати систему

моніторингу для відстеження стану оновлень та своєчасного інформування про необхідність їх встановлення.

Інструменти для контролю доступу, такі як OAuth та JWT, є важливими для захисту web-додатків. OAuth використовує токени доступу, які можуть бути обмежені за часом дії та обсягом доступу.

OAuth, або Open Authorization, є стандартом для авторизації, який дозволяє стороннім додаткам отримувати обмежений доступ до ресурсів користувачів без необхідності розкривати їх паролі. Це надзвичайно важлива функція, адже вона дозволяє зберігати конфіденційність паролів, мінімізуючи ризики компрометації облікових записів. В основі OAuth лежить концепція використання токенів доступу, які можуть бути обмежені за часом дії та обсягом доступу. Це означає, що сторонні додатки отримують лише той рівень доступу, який їм необхідний, і на обмежений період, що значно підвищує рівень безпеки. Однією з головних переваг OAuth є його здатність працювати з різними платформами і сервісами, забезпечуючи єдиний механізм авторизації. Це робить його надзвичайно гнучким і зручним у використанні для розробників.

JWT (JSON Web Tokens) є ще одним важливим інструментом для забезпечення безпеки web-додатків. JWT – це популярний формат токенів, який забезпечує безпечний обмін інформацією між клієнтом та сервером. Токени JWT складаються з трьох частин: заголовка, корисного навантаження і підпису. Заголовок містить інформацію про тип токена і алгоритм шифрування, корисне навантаження містить дані, які потрібно передати, а підпис гарантує цілісність токена і підтверджує його автентичність.

Рис. 3.6. Декодування JWT токена

Однією з головних переваг JWT є те, що вони самодостатні. Вся необхідна інформація для аутентифікації і авторизації міститься всередині токена, що дозволяє серверам швидко і ефективно обробляти запити без необхідності звертатися до бази даних або іншого сховища. Це значно підвищує продуктивність і спрощує архітектуру додатку.

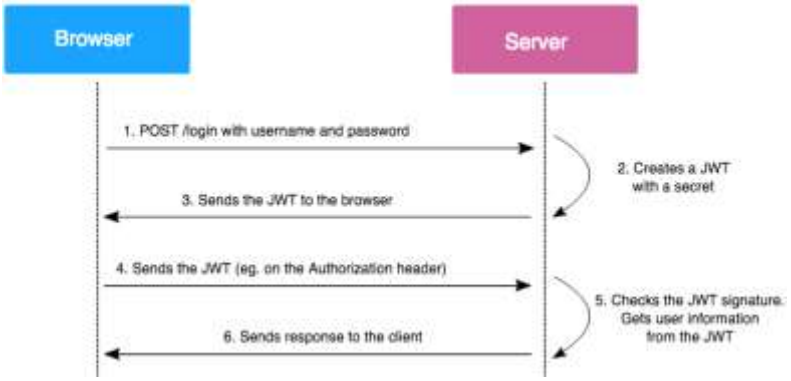


Рис. 3.7. Аутентифікація та авторизація за допомогою JWT токена

JWT використовуються для аутентифікації та авторизації, забезпечуючи простоту інтеграції та високу продуктивність. Після успішної аутентифікації користувача сервер генерує JWT, який містить інформацію про користувача і права доступу. Цей токен потім передається клієнту, який використовує його для доступу до захищених ресурсів. Завдяки підпису, сервер може легко перевірити автентичність токена і впевнитися, що він не був змінений.

Об'єктно-реляційне відображення (ORM) є технологією, що дозволяє розробникам взаємодіяти з базами даних за допомогою об'єктно-орієнтованого підходу. Це значно спрощує процес розробки і забезпечує вищий рівень безпеки. ORM дозволяє працювати з базами даних, використовуючи об'єкти замість написання складних SQL-запитів вручну.

SQLAlchemy є одним з найпотужніших інструментів ORM для Python, який забезпечує зручний і безпечний інтерфейс для роботи з базами даних. Однією з головних переваг SQLAlchemy є захист від SQL-ін'єкцій. SQL-ін'єкції є однією з найпоширеніших атак на web-додатки, що дозволяє зловмисникам виконувати шкідливі SQL-запити, змінювати дані або отримувати доступ до конфіденційної інформації. SQLAlchemy автоматично запобігає SQL-ін'єкціям, використовуючи параметризовані запити. Це означає, що всі введені користувачем дані автоматично екрануються і передаються в запити як параметри, що унеможливлює ін'єкцію шкідливого коду.

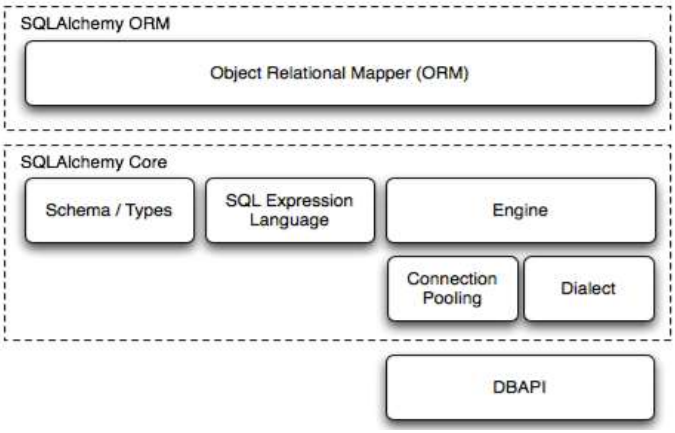


Рис. 3.8. Архітектура SQLAlchemy

SQLAlchemy дозволяє реалізовувати бізнес-логіку на рівні ORM. Це зменшує ймовірність помилок при написанні SQL-запитів вручну та сприяє зниженню кількості вразливостей у коді. Валідація даних є ще одним важливим аспектом, який забезпечується SQLAlchemy. Перед збереженням даних у базі даних можна легко додати валідацію, щоб переконатися, що дані відповідають очікуваному формату і не містять шкідливих елементів. Це значно підвищує безпеку web-додатків, оскільки знижує ризик введення некоректних або шкідливих даних.

Логування та аудит дій у системі також підтримуються SQLAlchemy. Це дозволяє вести облік всіх виконуваних запитів і виявляти підозрілу активність у базі даних. Логування SQL-запитів для подальшого аналізу допомагає виявляти аномалії та потенційні загрози, що сприяє оперативному реагуванню на інциденти безпеки.

Використовуючи інструменти, які пропонує нам Cloudflare WAF, можна значно покращити безпеку web-додатків. Cloudflare WAF забезпечує захист від відомих різноманітних загроз, завдяки вбудованим захисним правилам

Також за допомогою загальних рекомендацій, створення web-додатків стає набагато простішим. Впровадження шифрування даних, регулярне оновлення програмного забезпечення, інтегрування ORM та використання JWT токенів значно підвищують рівень безпеки web-додатку.

ВИСНОВОК

Були досліджені та проаналізовані види загроз для bweb-додатків, а також методи їх виявлення.

Проаналізовано ключові аспекти забезпечення безпеки web-додатків, включаючи захист баз даних, аутентифікацію та авторизацію користувачів, шифрування даних, регулярне оновлення програмного забезпечення та використання веб-аплікаційних фаєрволів (WAF). Особливу увагу приділено захисту від SQL-ін'єкцій, XSS (Cross-Site Scripting), CSRF (Cross-Site Request Forgery) та інших поширених вразливостей.

Розглянуто основні функції Cloudflare WAF, та був досліджений його інструментарій який можна використовувати для захисту web-додатку.

Розроблені рекомендації на основі Cloudflare WAF дозволяють створити надійні та захищені web-додатки, що відповідають сучасним вимогам безпеки та забезпечують конфіденційність, цілісність та доступність даних користувачів.

Також були розглянуті загальні рекомендації щодо забезпечення безпеки web-додатків, включаючи захист баз даних, аутентифікацію та авторизацію користувачів, шифрування даних, регулярне оновлення програмного забезпечення та логування. Впровадження загальних рекомендацій сприяє мінімізації ризиків кіберзагроз та забезпечує більш стабільну та надійну роботу web-додатків.

Запропоновані рекомендації включають інтеграцію заходів безпеки на всіх етапах життєвого циклу web-додатків.

Розроблені рекомендації дозволяють створити надійні та захищені web-додатки, що відповідають сучасним вимогам безпеки та забезпечують конфіденційність, цілісність та доступність даних користувачів. Впровадження цих рекомендацій сприяє мінімізації ризиків кіберзагроз та забезпечує стабільну та надійну роботу інформаційних систем.

ПЕРЕЛІК ПОСИЛАНЬ

1. NIST Special Publication 800-95: Guide to Secure Web Services веб-сайт. URL https://csrc.nist.gov/pubs/sp/800/95/final (дата звернення: 20.04.2024);

2. NIST Special Publication 800-63B: Digital Identity Guidelines веб сайт URL https://pages.nist.gov/800-63-3/sp800-63b.html (дата звернення: 20.04.2024);

3. Microsoft SDL Practices веб сайт URL https://www.microsoft.com/en-us/securityengineering/sdl/practices (дата звернення: 23.04.2024);

4. Mozilla Developer Network (MDN) Security веб сайт URL https://developer.mozilla.org/en-US/docs/Web/Security (дата звернення: 23.04.2024);

5. Mozilla Developer Network (MDN) Content Security Policy (CSP) веб сайт URL https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP (дата звернення: 27.04.2024);

6. JWT introduction веб сайт URL https://jwt.io/introduction (дата звернення: 27.04.2024);

7. CWE/Sans Top 25 веб сайт URL https://www.sans.org/top25-software-errors/ (дата звернення: 27.04.2024);

8. OWASP Top 10 веб сайт URL https://owasp.org/www-project-top-ten/ (дата звернення: 27.04.2024);

9. SQLAlchemy ORM вебсайт URL https://docs.sqlalchemy.org/en/20/intro.html (дата звернення: 03.05.2024);

10. Cloudflare WAF веб сайт URL https://www.cloudflare.com/application-services/products/waf/ (дата звернення: 03.05.2024);

11. Bryan Sullivan, Vincent Liu. Web Application Security, A Beginner's Guide – 352 p.