ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЗАХИСТУ ІНФОРМАЦІЇ

КАФЕДРА ІНФОРМАЦІЙНОЇ ТА КІБЕРНЕТИЧНОЇ БЕЗПЕКИ

Пояснювальна записка

до магістерської роботи
на тему:

«ТЕХНОЛОГІЯ ЗАХИСТУ WEB-ДОДАТКІВ НА ОСНОВІ CLOUDFLARE WAF»

Виконав студент 6 курсу, групи БСДМ-61
спеціальності 125 Кібербезпека
освітньо-професійної програми «Інформаційна та кібернетична безпека»
(шифр і назва спеціальності)
Боднар К.С.
(прізвище та ініціали)
Керівник Марченко В.В.
(прізвище та ініціали)
Рецензент
(прізвище та ініціали)
Нормоконтролер Чумак Н.С.
(прізвище та ініціали)

КИЇВ – 2023

РЕФЕРАТ

Текстова частина магістерської роботи: 59 сторінок, 23 рисунка, 1 таблиця, 19 джерел.

Об'єкт дослідження — процес забезпечення захисту WEB-додатків від сучасних загроз на основі WAF.

Предмет дослідження — технологія захисту WEB-додатків на основі Cloudflare WAF.

Мета роботи — розробка практичних рекомендацій щодо інтеграції брандмауеру WEB-додатків на основі Cloudflare WAF.

Методи дослідження — опрацювання літератури за даною темою, аналіз експлуатаційної документації, дослідження критичних загроз безпеки WEB-додатків OWASP-10 та методів боротьби проти них, огляд сучасних рішень WAF для визначення оптимальних та ефективних підходів для інтеграції брандмауеру у WEB-додатки.

В роботі було оглянуто сучасні WEB-додатки та проаналізовано проблему захисту від загроз у WEB-додатки на базі статистичних даних. Досліджено сучасні загрози безпеки на база OWASP Top 10 та надано рекомендації протидії.

Було досліджено брандмауер WEB-додатків та проведене порівняння на основі сучасних рішень безпеки. Проведено аналіз методів захисту на основі Cloudflare WAF

На базі досліджень проведених у роботі було розроблено рекомендації щодо захисту WEB-додатку та інтеграції брандмауеру у WEB-додатки.

Галузь використання – кібербезпека WEB-додатків.

WEB-ДОДАТОК, OWASP-10, ЗАГРОЗИ, МЕТОДИ ПРОТИДІЇ, ЗАХИСТ WEB-ДОДАТКІВ, БРАНДМАУЕР, КІБЕРБЕЗПЕКА.

ABSTRACT

Master’s thesis: 59 pages, 23 figures, 1 tables, 19 sources.

Object of research — the process of ensuring the protection of WEB-applications via WAF

Subject of research — the technology of securing WEB-applications based on Cloudflare WAF

The aim of research — to develop practical recommendations regarding WEB-application security based on WAF.

Research methods — elaboration of literature on the topic, analysis of operational documentation, critical security vulnerabilities research based on OWASP-10 and methods of protection against them, modern WAF solutions review to create optimal and effective ways to integrate WAF security to WEB-applications.

In the work have been reviewed modern WEB-applications and analyzed vulnerabilities security problems based on statistics data. Modern security threats based on the OWASP Top 10 were studied and countermeasure recommendations were provided.

The firewall of WEB applications was investigated and a comparison was made based on modern security solutions. Conducted an analysis of security methods based on Cloudflare WAF.

Based on the research conducted in the work, recommendations were developed for the protection of the WEB application and the integration of the firewall in WEB applications.

Field of use – cybersecurity of corporate information system..

WEB-APPLICATION, OWASP-10, THREATHS, SECURITY METHOL , WEB-APPLICTION SECURITY, FIREWALL, CYBERSECUTITY.

ЗМІСТ

ВСТУП

Актуальність дослідження. За останні роки інтернет пройшов дуже великий шлях від середовища для науковців до всесвітнього павутиння, яке використовують мільйони бізнесів по всьому світу для надання послуг. WEB-додатки є сучасним методом взаємодії користувача з різноманітними сервісами та послугами в Інтернеті. Сучасний інтернет працює на WEB-додатках, від онлайн-банкінгу та програм для віддаленої роботи до доставки особистих розваг та електронної комерції. Не дивно, що додатки є основною мішенню для зловмисників, які використовують вразливості доступності WEB-браузеру та WEB-додатків для здійснення зловмисних дій.

Кібератаки постійно еволюціонують, і за статистикою, кількість кібератак майже подвоюється кожного року. Ось чому так важливо мати надійні засоби забезпечення захищеності критичної інформації та інфраструктури.

Брандмауер WEB-додатків є сучасним та ефективним вирішенням проблем безпеки у WEB-додатках. Розгортання WAF сумісно з WEB-додатком створює стіну, яка блокує загрози та створює надійний захист у периметрі системи. Технологія WAF постійно оновлюється та вдосконалюється, що дозволяє їй бути головним методом захисту при створенні захисту WEB-ресурсу.

Опираючись на брандмауер, можна налаштувати та побудувати надійний захист для ефективного підтримання захищеності системи на основі потужних та сучасних методів безпеки.

Об'єкт дослідження — процес забезпечення захисту WEB-додатків від сучасних загроз на основі WAF.

Предмет дослідження — технологія захисту WEB-додатків на основі Cloudflare WAF

Мета роботи — розробка практичних рекомендації щодо інтеграції брандмауеру WEB-додатків на основі Cloudflare WAF .

Наукові завдання:

- *Розглянути сучасні WEB-додатки та їх можливості.*
- *Дослідити проблеми безпеки у WEB-додатках, проаналізувати статистичні дані проблеми.*
- *Оглянути технологію забезпечення безпеки Web application firewall.*
- *Розробити рекомендації щодо забезпечення безпеки на основі рішення Cloudflare WAF.*

Практичне значення отриманих результатів дослідження полягає в можливості їх використання для оптимізації налаштування WAF при використанні у захисті WEB-ресурсу від сучасних загроз.

1 АНАЛІЗ СУЧАСНИХ WEB-ДОДАТКІВ ТА ЇХ ПРОБЛЕМ БЕЗПЕКИ

1.1 Огляд сучасних WEB-додатків та історія їх розвитку

Мільйони підприємств використовують Інтернет як економічно ефективний канал зв'язку. Це дозволяє їм обмінюватися інформацією зі своїми потенційними клієнтами і цільовим ринком та здійснювати швидкий і безпечний обмін інформацією. WEB-додатки виступають найпоширенішим методом надання послуг в всесвітньому павутинні. Будь-яка послуга, що пропонується через інтернет, за визначенням, є формою веб-додатку.

WEB-додаток — це різновид комп'ютерної програми, яка використовує WEB-браузер та WEB-технології для виконання різноманітних запитів та завдань через інтернет.

WEB-додатки включають до себе онлайн-форми, кошики для покупок, потокове відео, соціальні мережі, ігри, електронну пошту та безліч інших онлайн сервісів. Враховуючи загальну доступність, багато WEB-додатків наразі розроблюються для функцій, які раніше не потребували онлайн-доступу, таких як обробка текстів, створення електронних таблиць і редагування графіки чи відео. Таким чином, їх розповсюдженість та проблеми безпеки досягають критичних обсягів у сучасному часі[1].

Перш ніж говорити про WEB-додатки, треба зазначити різницю між WEB-додатками, які зараз охоплюють переважну частину інтернету та WEB-сайтами, які є більш популярними за визначенням. WEB-сайт і WEB-додаток обидва працюють у браузері, їм потрібен доступ до інтернету і вони використовують однакові мови програмування, проте в них є критичні відмінності. Основна відмінність між WEB-сайтом і WEB-додатком полягає в тому, що WEB-додаток — це частина програмного забезпечення, доступ до якого здійснюється за допомогою браузера, тоді як WEB-сайт — це лише набір пов'язаних WEB-сторінок з інформацією без

інтерактивних елементів. Звісно WEB-сайти можуть містить зображення, текст, аудіо та відео, проте у сучасному світі, переважна частина інтернету працює за допомогою WEB-додатків. Таким чином такі популярні WEB-портали сучасності як “Youtube”, “Twitter” або ж “Facebook” є прикладом WEB-додатків. Шлях розвитку та переходу інтернету від WEB-сайтів до WEB-додатків супроводжувався багатьма періодами розвитку [2].

Одну з перших мов програмування, на якій будувалися WEB-додатки, Perl, було розроблено в 1987 році. Це було ще до того, як інтернет став насправді популярним за межами академічних і технологічних кіл. Перші WEB-додатки були досить простими, оскільки кожна окрема веб-сторінка в інтернеті на той час була статичним документом. Інтерактивний досвід був можливий лише за допомогою онлайн-форм і кнопок на низці різних веб-сторінок, але він був обмеженим у своїй неефективності, вимагаючи від серверу надсилати абсолютно нову WEB-сторінку щоразу, коли користувач хоча б трохи маніпулював елементами інтерфейсу.

Нові можливості з’явилися в 1995 році, коли компанія комп’ютерних послуг Netscape Communications Corp. випустила JavaScript, мову програмування, яка дозволяла розробникам додавати динамічні елементи до клієнтської сторони додатку. Тепер можна було показувати або приховувати елементи WEB-сторінки, а також зберігати введені дані користувача перед надсиланням перевірки форми без звернення до сервера [1].

У 1999 році концепт WEB-додатків з’явився і в мові Java, проте, сучасного рівня інтерактивності додатки досягли тільки при виході AJAX - асинхронному JavaScript у 2005 року. Цей комплекс методів WEB-розробки дозволив програмісту створювати асинхронні WEB-додатки. Принцип асинхронності є дуже простий і водночас революційний. Цей підхід дозволив користувачеві працювати в мережі швидше та якісніше. Асинхронність полягає в тому, що WEB-додатки можуть надсилати дані на сервер і отримувати їх з нього, не заважаючи загальній роботі на WEB-сторінці. Така інновація дозволила користувачам справжню інтерактивну взаємодію з WEB-сторінками без постійного переривання та перезавантаження WEB-ресурсу, що збільшило ефективність роботи у багато разів.

Також на розвиток сучасних WEB-додатків вплинув вихід стандарту HTML5, а точніше його глобальне оновлення 2014 року. HTML використовується для написання структури WEB-сторінок, так званих API. Однієї цієї структури недостатньо, щоб зробити WEB-сторінку презентабельною та інтерактивною. Таким чином, існують допоміжні технології, такі як CSS і JavaScript, щоб доповнювати HTML і додавати до WEB-додатку сучасні функції. Історія WEB-додатків є досить складною і включає в себе багато етапів розвитку і технології. В сучасному часі WEB-додатки є дуже гнучким та зручним інструментом взаємодії з різноманітними послугами, який використовується у всьому світі щоденно [3].

Користувачі можуть отримати доступ до WEB-додатку через WEB-браузер, наприклад, Google Chrome, Mozilla Firefox або Safari з будь яких сучасних девайсів, таких як смартфон, ноутбук, планшет і навіть телевізор.

Для роботи сучасного WEB-додатку, він використовує WEB-сервер, сервер додатків і базу даних. WEB-сервери керують запитами, які надходять від користувача, тоді як сервер додатків виконує поставлене їм завдання. Для зберігання інформації використовується база даних.

Елементи WEB-додатку розділяється на дві частини. Частина, яка відповідає за роботу користувача з інтерфейсом WEB-додатку, називається front end, а частина, яка відповідає за роботу WEB-додатку з сервером – backend.

У сучасних додатках для інтерфейсу використовуються мови програмування HTML, CSS і JavaScript. Ці мови покладаються на браузер для виконання програмного коду та відповідають за роботу користувача з WEB-додатком і відображення інформації.

Програмний код для взаємодії з сервером частіше за все пишуть на мовах таких як Java або Python. Backend займається зберіганням і отриманням інформації з бази даних, щоб відправити його до front end.

Кожна сторона має взаємодіяти та ефективно працювати як єдине ціле, щоб покращити функціональність веб-сайту. Як тільки користувач відкриває WEB-сторінку, сервер надсилає певні дані браузеру у відповідь на запит користувача. Якщо бути точним, WEB-клієнт (або агент користувача) може запитувати WEB-

ресурси або більш відомі WEB-документи HTML, JSON або PDF формату через WEB-сервер. За допомогою цих мінімальних маніпуляцій з'являється потрібна інформація. Після цього починається взаємодія між користувачем і сайтом. На рисунку 1.1 представлена загальна інформація про архітектуру та роботу WEB-додатку

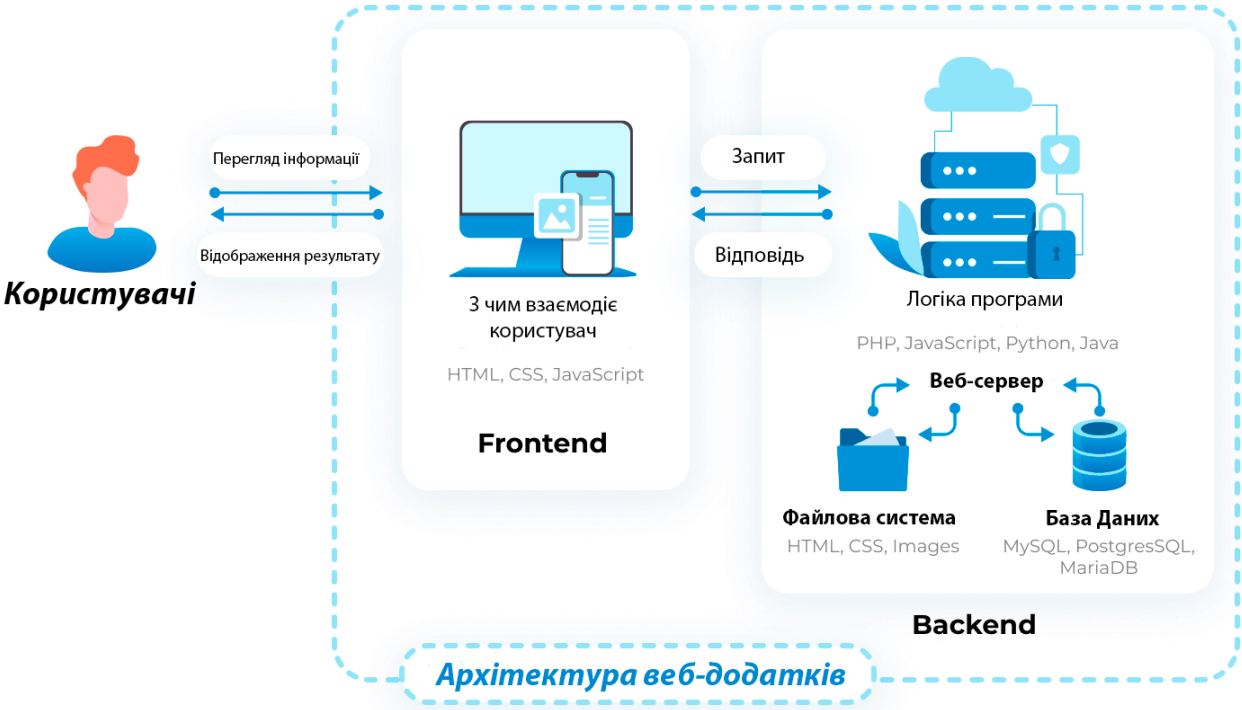


Рис 1.1 Архітектура та загальний механізм роботи WEB-додатку

Роботу типового сучасного WEB-додатку можна охарактеризувати у декількох кроках:

1. Користувач створює запит до WEB-сервера через WEB-браузер за допомогою інтерфейсу додатку.
2. WEB-сервер надсилає запит серверу WEB-додатку.
3. Сервер WEB-додатків виконує поставлене завдання і якщо цього потребує ситуація, звертається за інформацією до бази даних, після чого надсилає результати необхідних операцій до WEB-серверу.
4. WEB-сервер передає потрібну інформацію клієнту, після чого він бачить результат роботи за допомогою інтерфейсу у WEB-браузері.

1.2 Структура WEB-додатків

1.2.1 Архітектура роботи WEB-додатку

В цій частині роботи буде розглянуто побудову сучасного WEB-додатку для більш просунутого розуміння його роботи. Сучасні WEB-додатки та інформаційні системи досягли такого рівня розвитку, що застосовуваний до них термін “архітектура” вже не здається недоречним.

Для детального розгляду роботи WEB-додатку, на рисунку 1.2 представлена схема процесу взаємодії браузера користувача з іншими необхідними компонентами при роботі з WEB-додатком в інтернеті.

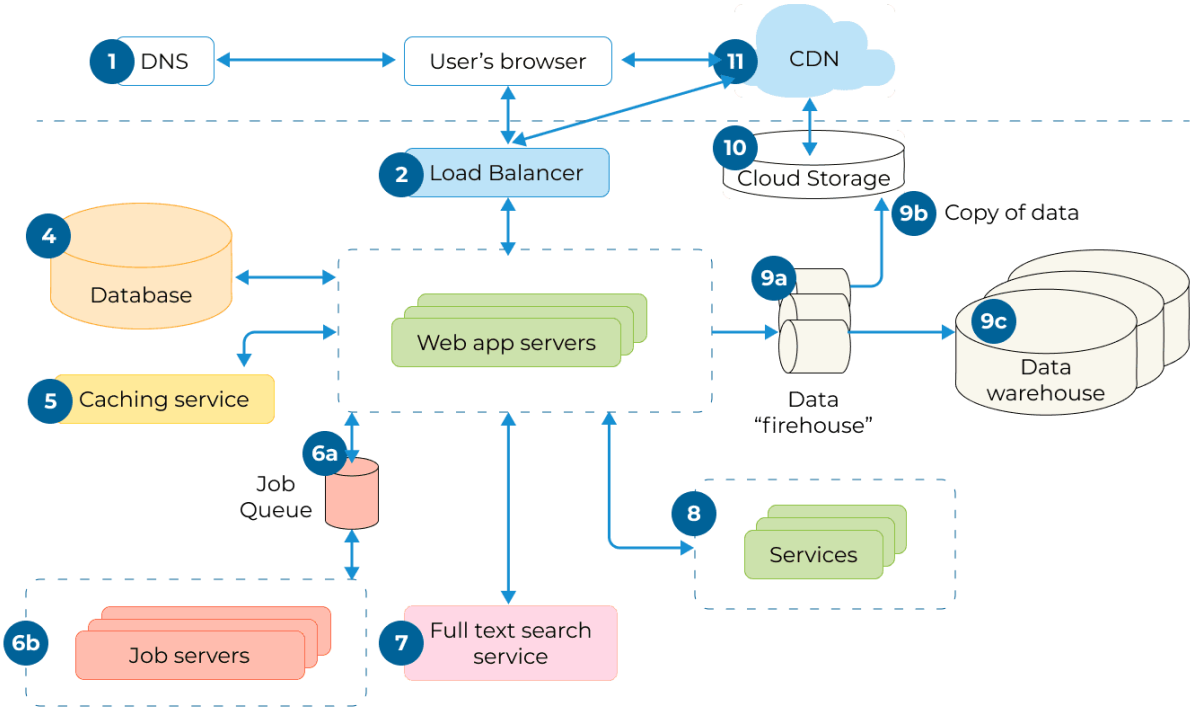


Рис 1.2 Схема процесу роботи запитів користувач-сервер

Компоненти архітектури веб-додатків [4]:

Система доменних імен (DNS)

DNS або система доменних імен є основною системою, яка допомагає шукати ім’я домену та IP-адресу для розміщення WEB-сторінки в інтернеті, і таким чином,

певний сервер отримує запит, надісланий користувачем. Можна сказати, що DNS схожий на телефонну книгу, але для WEB-сайтів в Інтернеті.

Балансувальник навантаження (Load balancer)

Основною метою балансування навантаження є запобігання перевантаженню сервера та підтримка належного рівня активності на кожному сервері. Балансуючи вхідні запити та потужність ресурсів, балансувальник навантаження створює високодоступну та надійну систему.

Балансир навантаження зазвичай розміщуються між внутрішніми серверами та брандмауером. Використання балансира навантаження, який розподіляє клієнтські запити на доступні сервери, дозволяє обробляти умови високого трафіку без простою. Це також забезпечує кращу доступність і швидкість роботи.

Сервери WEB-додатків (Web apps servers)

Цей компонент обробляє запит користувача та надсилає документи у форматі JSON назад у браузер. Для виконання запиту WEB-сервер зазвичай звертаються до внутрішньої інфраструктури додатку, а саме до бази даних, кеш-серверу, черги завдань, тощо.

База даних (Database)

База даних зберігає критичну інформацію WEB-додатку та надає інструменти для організації, додавання, пошуку, оновлення, видалення та виконання різноманітних дій з даними.

Кешування (Caching service)

Кешування — це процес зберігання копій файлів у кеші або місці тимчасового зберігання, щоб до них можна було швидше отримати доступ. Технічно кеш — це будь-яке місце тимчасового зберігання копій файлів або даних, але цей термін часто використовується стосовно інтернет-технологій. WEB-браузери кешують файли HTML, JavaScript і зображення, щоб швидше завантажувати WEB-сторінки, тоді як DNS-сервери кешують записи DNS для швидшого пошуку, а сервери CDN кешують вміст для зменшення затримки.

Черга завдань (Job Queue)

Хоча черга завдань не є обов’язковим атрибутом у WEB-додатках, нині все більше компанії імплементують це поняття не тільки при розробці WEB-додатків, а і при побудові архітектурі у багатьох типах програмного забезпечення.

Черга завдань — це структура даних, яка підтримується програмним забезпеченням планувальника завдань і містить завдання для виконання. Вона дозволяє користувачеві WEB-додатку ініціювати, тобто, поставити в чергу, довгострокову операцію, зберігаючи ресурси для подальшої роботи додатку.

Служба повнотекстового пошуку (Full-text search service)

Багато WEB-додатків підтримують функцію пошуку за текстом або так званий запит, а потім додаток надсилає користувачеві найбільш релевантні результати. Ця технологія називається сервісом повнотекстового пошуку. За допомогою ключових слів він шукає потрібні дані серед величезної кількості документів.

Services (Послуги)

WEB-сервіси — це системи обміну інформацією на основі XML, які використовують Інтернет для прямої взаємодії між додатками. Ці системи можуть включати програми, об'єкти, повідомлення або документи. Сервіси — це набір відкритих протоколів і стандартів, які використовуються для обміну даними між програмами або системами. Коли WEB-додаток досягає певного рівня, служби створюються у вигляді окремих додатків, які взаємодіють між собою.

Сховище даних(Data warehouse)

Сховище даних — це центральний репозиторій інформації, яку можна аналізувати для прийняття більш обгрунтованих рішень. Дані надходять у сховище з транзакційних систем, реляційних баз даних та інших джерел, як правило, з певною періодичністю. Бізнес-аналітики, фахівці з роботи з даними та особи, відповідальні за прийняття рішень, отримують доступ до даних за допомогою інструментів бізнес-аналітики, SQL-клієнтів та інших програм для аналітики

Практично кожен сучасний додаток постійно взаємодіє з даними. Для ефективного збору, зберігання та аналізу цих даних виконується декілька єтапів:

1. Дані надсилаються до сервісу data firehose, який збирає, перетворює та доставляє потокові дані до інших сервісів збору даних.
2. Необроблені та структуровані дані надсилаються в хмарне сховище, а оброблені та структуровані дані попадають до сховища даних для подальшого аналізу.

Ця модель включає до себе онлайн-зберігання та обмін даних через Інтернет.

Сховище даних можна використовувати для зберігання різноманітних файлів різних типів, таких як відео, фотографії, тощо.

Мережа доставки контенту (CDN)

Мережа доставки контенту (CDN) відноситься до територіально розподіленої групи серверів, які працюють разом, щоб забезпечити швидку доставку Інтернет-вмісту.

CDN дозволяє швидко передавати ресурси, необхідні для завантаження контенту, включаючи сторінки HTML, файли JavaScript, таблиці стилів(CSS), зображення та відео. Популярність послуг CDN продовжує зростати і сьогодні більшість веб-трафіку обслуговується через CDN, включаючи трафік із таких основних популярних, як Facebook, Netflix і Amazon.

Правильно налаштований CDN також може допомогти захистити WEB-сайти від деяких поширених зловмисних атак, таких як DDOS.

1.2.2 Типи будови WEB-додатку

Буває також декілька популярних типів побудови архітектури сучасного WEB-додатку. Як вже говорилося раніше, додаток поділяється на дві частини front end та back end і для кожної частини існують свої нюанси та типи роботи [5].

У Front end частині можна виділити два популярні типи архітектури: SPA та SSR

Single-Page Application

Такий тип додатків спрямований на полегшення роботи програмістів. Щоб виконати дію на сайті, користувачу не потрібно щоразу завантажувати абсолютно

нові сторінки при переході у інший розділ WEB-додатку. Натомість контент надається та оноволюється на поточній сторінці користувача.

Цей тип архітектури WEB-дизайну створено таким чином, щоб WEB-сторінки надсилає запит тільки для отриманні самих важливих даних та контенту. Таким чином, SPA запобігають перериванням активності користувача, щоб підвищити інтуїтивно зрозумілий та інтерактивний досвід користування додатком.

Server-Side Rendered Application

SSR — це процес рендеру WEB-сторінки на сервері для того, щоб віддати вже готовий результат до браузеру користувача. За допомогою цього інструменту можна швидко доставляти найважливіші ресурси, таким чином прискорюючи швидкість браузера, рендеринг сторінки користувачу займає менше часу та потребує менше ресурсів.

Коли додаток працює на архітектурі SSR, сервер компілює всі дані та подає новий HTML-документ за кожним запитом. І коли браузер отримує CSS, він може малювати інтерфейс користувача, і немає необхідності чекати поки завантажиться JavaScript. Таким чином сторінка завантажується швидше.

На back end частині додатку теж є два популярних типи архітектури: Microservices і Serverless Architecture:

Microservices

Мікросервіси є одним із видів SOA (сервісно-орієнтованої архітектури). Загалом, мікросервіси мають справу з невеликими та легкими завданнями та виконують одну функцію. Цей тип дизайну WEB-архітектури є ефективним і продуктивним. З його допомогою розробники можуть заощадити багато часу

Оскільки компоненти мікросервісів не обов’язково створюються за допомогою однієї мови програмування, вони не є взаємозалежними. Це означає, що розробники можуть вільно вибирати технологію, яка їм більше подобається, що в результаті робить розробку архітектури мікросервісів швидшою та легшою.

Serverless architecture

З назви можна здогадатися, що безсерверна архітектура — це та, у якій відсутні сервери. Ну, це не так. Насправді це означає, що налаштування та

адміністрування серверів, на яких працює програмне забезпечення для їх підтримки, більше не є необхідністю. Вся інфраструктура у такому методі підтримується сторонніми провайдерами. Сторонній постачальник надає аутсорсинговий сервер і керування інфраструктурою.

У цьому типі архітектури WEB-додатків розробник консультується зі стороннім постачальником послуг хмарної інфраструктури щодо аутсорсингу сервера, а також управління інфраструктурою.

Цей підхід зараз набуває популярність, оскільки він є дуже швидким та простим у користуванні та налаштуванні.

1.3 Сучасний стан проблеми забезпечення безпеки у WEB-додатках оснований на статистичних даних

Кібератаки спрямовані на руйнування роботи WEB-додатків є реальністю й трапляються кожен у самих на перш погляд захищених системах та завдають колосальних збитків.

Уразливості зазвичай відносяться до недоліку програмного коду, який дозволяє зловмиснику отримати доступ до мережі або системи. Уразливості роблять компанії та окремих осіб відкритими для низки загроз, включаючи вживлення зловмисного програмного забезпечення до комп’ютеру жертви або захоплення облікового запису користувача.

Безпека WEB-додатків – це постійна боротьба проти постійно прогресуючих кіберзагроз, які постійно розвиваються, і ставки цієї проблеми є високими. Для підприємств, які все більше покладаються на їхні WEB-ресурси для залучення та спілкування зі своїми клієнтами, наслідки порушення безпеки може бути дорогими та руйнівними.

Як видно на рисунку 1.3, за статистикою positive technologies 2020 року, 84 відсотки компаній мають доволі критичні вразливості безпеки. Уразливості

високого ризику, які мають загальнодоступний експлоіт, присутні в 58 відсотках компаній.

Рис 1.3. Відсоток вразливостей у WEB-додатках

Це означає що приблизно в 9 із 10 WEB-додатків користувачі можуть бути атаковані хакерами. Деяким звичайним типом атак є переспрямування користувачів на інший небезпечний ресурс, фішингові атаки для викрадення облікових даних і зараження комп'ютерів шкідливим, рекламним і шпигунським програмним забезпеченням.

Одним з критичних моментів проблеми виступає те, що багато малих та середніх підприємств досі не сприймають безпеку своїх WEB-додатків серйозно. Відповідно до звіту "State of Website Security and Threat Report" від Sectigo, SiteLock 2021 року, 48% підприємств констатують, що вони є занадто незначними, щоб бути цілями кібератак. Ця надмірна самовпевненість нерідко призводить до прямих збитків та очорнення репутації бізнесу, на відновлення якої можуть піти роки.

Справа в тому, що WEB-сайти малого та середнього бізнесу є основною ціллю для кіберзлочинців через їх слабку конструкцію безпеки. Жодна компанія не застрахована від постійно зростаючого обсягу атак і широких, різноманітних загрози, що постійно розвиваються.

За даними Sitelock з понад 14 мільйонів WEB-додатків, домінуючими проблемами безпеки на сьогодні виступають наступні пункти[5]:

• Обсяг загроз подвоївся у 2021 році порівняно з 2020 роком. Така ж тенденція зараз і у 2022 році оскільки база даних NVD містить 8051 вразливість, опубліковану в першому кварталі 2022 року. Це приблизно на 25% більше, ніж за той самий період минулого року.

• Основним джерелом збільшення обсягу атак є шкідливі боти. WEB-ресурси отримують в 5,5 разів більше трафіку від ботів, ніж від людей, причому понад 60% цих ботів є зловмисними.

• Шкідливе програмне забезпечення залишається значною проблемою для малого та середнього бізнесу. 18% WEB-додатків заражені загрозами критичного рівня, такими як бекдори та зловмисні модифікації коду програми, а додатки інфіковані загрозами високого рівня небезпеки майже подвоїлися у 2021 році до 26% рівня зараження.

• WEB-додатки на основі популярних платформ управління контентом WEB-ресурсів (CMS), таких як WordPress і Drupal експлуатується із загрозливою швидкістю.

• Пошукові системи часто пропускають зараження зловмисним програмним забезпеченням, у 92% випадків заражені зловмисним програмним забезпеченням WEB-додатки не позначаються пошуковою системою небезпечними та не заносяться в чорний список.

З розвитком технологій постійно з’являються нові вектори атак. Зараз, як ніколи, потрібні комплексні інструменти безпеки. Раніше захист кінцевих точок пристроїв та мережева безпека були найкращими заходами захисту. Проте, після виходу на ринок мобільних та хмарних технології, які значно знизили ефективність

безпеки мережі та, по суті, зруйнували підхід до захисту, орієнтований на периметр, на який так покладалися.

95% WEB-ресурсів написані на HTML5 та JavaScript – мовах програмування, які дуже легко редагувати та компрометувати, чим користуються зловмисники. Таким чином, WEB-додатки та API є дуже вразливими до кібератак на стороні клієнта. За даними Symantec, кожен місяць близько 4800 WEB-додатків перестають працювати шляхом розкрадання та повної компрометації.

Зараз організації все більше покладаються на інтерфейси прикладного програмування (API) для стимулювання інновацій, швидкості розробки та надання нових можливостей монетизації. Згідно зі звітом Open Web Application Security OWASP 2019, року "За своєю природою API розкривають логіку програми та конфіденційні дані, такі як інформація, що ідентифікує особу, і через це API все частіше стають ціллю для нападників". Переміщення важливих компонентів на клієнтську сторону програм, тобто поза захистом традиційної мережевої безпеки створює масову нову поверхню для атак, збільшуючи ризик таких атак, як formjacking, підробка об'єктної моделі документа (DOM tampering), зловживання сеансом (session abuse), атаки спрямовані на те, щоб замінити інтерфейс справжньої програми на телефоні та вкрасти дані (overlay attacks) та зловживання API [6].

Іншою критичною стороною проблеми, як вже зазначалося раніше, є боти. Вони і є основним джерелом подвоєння обсягу атак. Фактично, переважна більшість атак здійснюються автоматизованими ботами, а не людьми. Компанії стикаються зі стрімким зростанням обсягу трафіку ботів, який у середньому складає 2306 відвідувань на тиждень, згідно з аналізом трафіку Sitelock. Хоча існують законні причини для ботів відвідувати ресурси, такі як сканери пошукових систем і сканування авторських прав, боти також використовуються для різноманітних зловмисних цілей. Шкідливі боти можуть програмно відвідувати веб-сайти та виявляти вразливі місця в коді веб-сайту для здійснення масштабних атак. До цього списку можут відноситися атаки від DDoS до сканування на бекдор вразливості для вживлення зловмисного програмного забезпечення. Такий автоматизований підхід до атак насправді є дуже дешевий і простий у застосуванні

для хакерів, що визиває стрімкий зріст його популярності, таким чином, розповсюджені ботнети можуть коштувати лише 5 доларів США на день. З проблемою ботів краще за все як раз справляється фаервол, з яким далі буде детально ознайомлено в роботі, тому що він може блокувати поганих ботів, при цьому, не чіпаючи хороших.

Оскільки боти виявляють багато вразливих місць, це призводить до бекдорів, та проникнення зловмисного програмного забезпечення. Основною небезпекою є те, що типи шкідливих програм, які вживлюється на веб-сайти, постійно змінюються та прогресують. Наприклад, на початку пандемії COVID-19 спостерігався різкий зріст атак SQL-ін'єкцій на веб-сайти клієнтів, так як цей вектор атаки виявився ефективним проти компаній, які намагаються налагодити бізнес з нуля або розширюють свої онлайн-пропозиції.

Як тільки WEB-додаток зазнав зараження, кількість спроб руйнування роботи через різноманітні автоматизовані вектори атак лише збільшується та є постійним викликом для спеціалістів кібербезпеки. Як показує статистика SiteLock:

- 36% заражених WEB-додатків мали filehacker. Цей тип модифікації файлів дозволяє хакерам змінити або викрасти файли.
- 32% заражених WEB-додатків мали backdoor. Цей тип шкідливих програм надає хакерам несанкціонований доступ безпосередньо до файлів додатку та в багатьох випадках, доступ до бази даних, що може призвести до модифікації даних програми або бази даних, видалення та викрадення інформації, в тому числі ідентифікаційних даних клієнтів.
- 28% заражених додатків мали shall scipt атаку. При такій атаці кіберзлочинець вживляє шкідливий файл у каталог цільового WEB-сервера, а потім запускає цей файл зі свого WEB-браузера, що може призвести до взяття під контроль WEB-додатку хакерами.
- 28% заражених WEB-додатків мали malicious eval request атаку. Цей тип атаки є типом ін'єкції та може дозволити хакерам виконати небажаний код у WEB-додатку

- 6% заражених WEB-додатків мали фішингову атаку. Така атака проявляється у людському факторі. Шахрайські електронні листи чи інші форми взаємодії з працівниками використовуються для оману людини, що у подальшому допомагає отримати несанкціонований доступ до системи [6].

Ці дані підкреслюють важливість надійної стратегії безпеки WEB-додатків, яка буде захищати від усіх типів загроз, які застосовуються проти ресурсу, а також ефективне усунення випадків ін'єкції зловмисного програмного забезпечення. Також рішення мають бути автоматизовані, щоб малі і середні бізнеси могли зосередитися на веденні свого бізнесу не турбуючись про те, щоб стати наступною жертвою кібератаки.

Для захисту від зростаючого обсягу загроз малому та середньому бізнесу потрібно застосовувати цілісний підхід забезпечення безпеки, яка включає впровадження кількох рівнів автоматизованого захисту. Ці рівні повинні включати постійний захист від зловмисного програмного забезпечення, DDoS, внесення вірусів у чорний список, а також автоматизовані патчі для уникнення вразливостей та резервні копіювання WEB-додатків.

Такий перелік методів захисту включає в себе сучасний брандумауер, налаштування якого є основною захисту будь якого WEB-додатку. Детальніше побудову захисту WEB-додатків буде розглянуто у другій частині роботи

Висновки до розділу 1:

В даному розділі було розглянуто сучасні WEB-додатки та їх можливості. На основі цих даних було виявлено що безпека WEB-додатків стала складнішою, оскільки все частіше використовуються інтерфейси прикладного програмування (API), які часто стають мішенями хакерів та значно збільшують поверхню для кібератак.

Проаналізовано проблему загроз у WEB-додатках на основі статистичних даних. Після дослідження можна сказати, що постійно відбувається натиск нових WEB-атак на організації, а їх кількість постійно зростає. Проблеми атак ботів та

зловмисного програмного забезпечення також залишаються серйозними проблемами безпеки сьогодні.

Більшість компаній не надають проблемі захисту належної уваги, що полегшує зловмисникам створення серйозних збоїв, крадіжку або підробки особистої інформації.

2 ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ У СУЧАСНОМУ WEB-ДОДАТКУ

2.1 Аналіз ключових загроз безпеки та методів протидії на основі OWASP Top 10

WEB-додатки надають ті самі функції, що й додатки для настільних комп'ютерів або мобільні додатки, але зі зручністю доступності браузера. Їх також легше розповсюджувати на різних платформах, що підвищує здатність організації створювати більшу базу користувачів. На жаль, все ціла доступність також створюють шлюзи для зловмисників, які можуть зламати базу даних або клієнтські системи.

Багато експертів кібербезпеки раніше працювали за принципом "безпека через невідомість": вони вважали, що хакери не будуть копати достатньо глибоко, щоб використовувати вразливості в коді. Це виявилося помилкою. Зараз дуже важливо, щоб компанії змінили своє мислення та взяли на себе відповідальність за безпеку ресурсів, перш ніж їх випускати[7].

Open Web Application Security Project (OWASP) – це галузева некомерційна організація, яка займається просуванням безпеки в Інтернеті та аналізом сучасних загроз. Кожні кілька років вони створюють оновлений список 10 найпопулярніших уразливостей веб-додатків. За допомогою цього списка можно побачити тенденцію сучасних загроз та побудувати модель безпеки.

Повний список складено з допомогою різноманітних експертних джерел, таких як, консультанти з безпеки, постачальники засобів безпеки та відділи кібербезпеки з компаній і організацій будь-якого розміру. Цей лист визнають важливим посібником із найкращих практик безпеки веб-додатків.

Останне оновлення OWASP було у 2021 та включало наступні загрози [8]:

1. Broken Access Control
2. Cryptographic Failures
3. Injection
4. Insecure Design

5. Security Misconfiguration
6. Vulnerable and Outdated Components
7. Identification and Authentication Failures
8. Software and Data Integrity Failures
9. Security Logging and Monitoring Failures
10. Server-Side Request Forgery

Надалі буде досліджено кожну з 10 найпопулярніших вразливостей OWASP, щоб краще зрозуміти їхній вплив на безпеку, і як їх можна уникнути. Чим вище знаходиться загроза у списку, тим вона небезпечніша.

Broken access control

Контроль безпеки WEB-додатку повинен надавати доступ користувачу лише до тих сторінок або розділів, до яких він звертається та має доступ. Наприклад, адміністратори електронної комерції повинні мати можливість додавати нові посилання або рекламні акції. Ці функції не повинні бути доступні для інших типів відвідувачів.

Порушений контроль доступу може надати користувачу доступ до панелей адміністратора, серверів, баз даних та інших критично важливих елементів програми. Також ця загроза може навіть використовуватися для перенаправлення браузерів користувачів на інші цільові URL-адреси.

Поширені загрози, які призводять до порушення контролю доступу включають:

- Порушення принципу найменших привілеїв або заборони за замовчуванням
- Обхід перевірок контролю доступу шляхом зміни URL-адреси
- Незахищені прямі посилання на конфіденційні дані користувача методом підміни унікального ідентифікатору
- Доступ до API з некоректно налаштованим контролем доступів для команд POST, PUT і DELETE.
- Підвищення привілеїв користувачів
- Маніпуляція метаданими, а саме, повторне відтворення чи підробка токену керування доступом JSON Web Token (JWT) або маніпуляція файлами cookie

- Неправильна конфігурація CORS, що дозволяє отримати доступ до API з неавторизованих/ненадійних джерел.
- Можливість примусового перегляду автентифікованих сторінок як неавтентифікований користувач

Уразливість порушеного контролю доступу можна усунути кількома способами:

- Застосування найменш привілейованого підходу, щоб кожній ролі надавався найнижчий рівень доступу по стандарту, необхідний для виконання її завдань.
- Видалення неактивних та непотрібних аккаунтів із системи.
- Аудит активності WEB-додатку
- Підтримання ефективності серверів, методом вимикання непотрібних служб та сервісів.
- Якщо є кілька точок доступу в одного користувача, вимикати ті, які в даний момент не використовуються.

Cryptographic Failures

Критичні дані WEB-додатку, що передаються та залишаються у системі, такі як паролі, номери кредитних карток, медичні записи, особиста інформація та комерційна таємниця завжди повинні перебувати у системі в зашифрованому вигляді для їх безпеки.

Інциденти безпеки пов’язані з криптографією можуть призводити до розкриття конфіденційних даних, а у деяких випадках також можуть скомпрометувати систему, таким чином наслідки цієї вразливості вважаються одним із найбільш критичних ризиків для безпеки як для організацій, так і для бізнес-користувачів.

До недоліків безпеки, які зазвичай призводять до збоїв у криптографії, належать:

- Передача секретних даних у вигляді відкритого тексту
- Використання старого/менш безпечного алгоритму
- Використання жорстко закодованого пароля у конфігураційних файлах
- Неправильне керування криптографічним ключем

- Відсутнє шифрування
- Небезпечна реалізація перевірки сертифіката
- Використання застарілих хеш-функцій
- Наявність конфіденційних даних у системі керування джерелами
- Використання незахищених векторів ініціалізації
- Використання паролів як криптографічних ключів без функції виведення ключа на основі пароля

До методів уникнення криптографічних збоїв відносяться:

- Вимкнення автозаповнення у формах які збирають дані.
- Мінімізація розміру поверхні з критичними даними.
- Шифрування критичних даних під час простою та транзиту.
- Використання найсучасніших методів шифрування.
- Вимкнення хешування у формах для збору даних.
- Використання стійких та адаптивних функції хешування під час збереження паролів.

Injection

Ін’єкційні атаки належать до широкого класу векторів атак. Під час ін’єкційної атаки зловмисник надає програмі зловмисні дані, проте вони обробляються інтерпретатором як частина звичайної команди або запиту. У свою чергу, це змінює виконання цієї програми та змушує її виконати те, для чого вона не була розроблена, наприклад, генерувати команди які не повинні були виконуватися або отримати доступ до даних без належної автентифікації.

WEB-додаток вразливий до такого типу атак коли:

- Дані, надані користувачем, не перевіряються та фільтруються додатком.
- Динамічні запити або несанкціановані запити використовуються безпосередньо в інтерпретаторі.
- Несанкціновані дані використовуються безпосередньо або об’єднуються з звичайними даним.

Ін’єкційні атаки можна запобігти будь-якою комбінацією таких підходів:

- Відокремлення команди додатку від даних, щоб уникнути атак, які замінюють дані ненавмисним виконанням команд.
- Кодування запитів SQL із параметрами, замість структурування команди лише з даних, які були введені користувачем.
- Повне виключення інтерпретатору за допомогою безпечного API.
- Впровадження перевірки валідності даних на стороні сервера, а також системи виявлення вторгнень, яка виявляє підозрілу поведінку на стороні клієнта

Insecure Design

Незахищений дизайн — це широкий термін, який охоплює низку недоліків і визначається як “відсутній або неефективний дизайн контролю доступу”.

За своєю суттю незахищений дизайн полягає у відсутності засобів контролю безпеки, інтегрованих у програму протягом усього циклу розробки. Це може мати різноманітні глибокі наслідки для безпеки, оскільки сама програма розроблена не з урахуванням безпеки

Побудова WEB-додатку без урахування безпеки залишає фундаментальну конструкцію та основу програми незахищеними, відкриваючи двері для безлічі дірок у безпеці, що часто призводить до розкриття конфіденційної інформації або повного зламу програми.

Одним із найважливіших факторів, які сприяють незахищеному дизайну, є відсутність профілювання бізнес-ризиків, властивого програмному забезпеченню чи системі, що розробляється, і, отже, нездатність визначити, який рівень безпеки дизайну потрібен WEB-додатку.

Для уникнення незахищеного дизайну слід використовувати наступні техніки безпеки при розробці WEB-додатку:

- Використання безпечного життєвого циклу розробки створений професіоналами із AppSec.
- Використання бібліотеку безпечних шаблонів проектування або готових до використання компонентів.
- Використання моделювання загроз для важливої інфраструктури WEB-додатку, а саме автентифікації, контролю доступ і бізнес-логіки додатку.

- Історія дій користувачів має містити елементи захисту та керування.
- Інтеграція методів перевірки валідності даних у на кожному рівні додатку (від front end до back end).
- Розділення рівнів додатку на системний і мережевий залежно від вимог захисту,
- Обмеження споживання ресурсів користувачів та служб.

Security Misconfiguration

Неправильна конфігурація безпеки, як видно з назви, визначається невдачею реалізації засобів безпеки для сервера, системи чи веб-програми або реалізація цих засобів з помилкою. Ця загроза безпеці являється суто людською помилкою, оскільки програмне забезпечення це лише інструмент, впровадження якого у систему, повністю виконую людина.

Програма може бути вразливою до такого типа загрози, якщо:

- Відсутні відповідні елементи безпеки в будь-якій частині стеку програм або неправильно налаштовані контролі доступу для хмарних служб.
- Увімкнені або встановлені зайві функції (наприклад, непотрібні порти, служби, сторінки, облікові записи або привілеї).
- Облікові записи за замовчуванням і їхні паролі не змінюються та залишаються активними
- Обробник помилок відображає користувачам трасування стека технологій або інші надто інформативні повідомлення про помилки.
- Для оновлених систем найновіші функції безпеки вимкнено або неналаштовано безпечно.
- Параметри безпеки на серверах додатків, бібліотеках або базах даних не мають правильних значень та налаштувань безпеки.
- Програмне забезпечення є застарілим або вразливим
- HTTP заголовки є неправильно конфігурованими

Vulnerable and Outdated Components

Сучасні WEB-додатки часто включають до себе компоненти з відкритим кодом, такі як бібліотеки та фреймворки. Будь-який компонент із відомою вразливістю стає слабкою частиною, яка може вплинути на безпеку всієї програми. Хоча використання компонентів з відкритим вихідним кодом та відомими вразливими місцями займає низьке місце з точки зору серйозності проблеми безпеки, воно займає перше місце в рейтингу OWASP Top 10 за тим, як часто уразливість була основною причиною фактичного порушення даних.

Такий тип загроз представляє небезпеку, якщо:

- В WEB-додатку є компоненти з невідомими або не обновленими версіями.
- Програмне забезпечення додатку є вразливим, не підтримується або застаріло.
- Сканування на наявність вразливостей не проводиться регулярно
- Не проводятся перевірки на сумісність оновлених, нових або виправлених бібліотек.
- Компоненти мають неправильну конфігурацію безпеки

Найефективнішим захистом від цієї вразливості є сканування всіх компонентів коду на відомі вразливості та виконання патча безпеки чи іншого засобу у разі виявлення небезпеки. Існує кілька найкращих практик, які підвищують ефективність цієї лінії захисту:

- Усі компоненти, інтегровані у фреймворки компанії, мають бути під керуванням конфігурації.
- Сканер повинен мати можливість автоматично виявляти всі компоненти, які потрібно контролювати.
- Сканування слід проводити за базою даних уразливостей, збагаченою даними аналізу загроз.
- Менеджмент патчів для визначення, тестування та розгортання правильного виправлення мають бути максимально автоматизованими, щоб звести до мінімуму операційний ризик, пов’язаний із встановленням виправлень.

Identification and Authentication Failures

Помилки ідентифікації та автентифікації можуть виникати, коли функції, пов’язані з ідентифікацією користувача, автентифікацією або керуванням сеансом не реалізовані або не захищені належним чином у додатку. Зловмисники можуть скористатися помилками ідентифікації та автентифікації, зламавши паролі, ключі, токени сеансу, або використати інші недоліки реалізації, щоб тимчасово або назавжди отримати доступ до конфіденційних даних інших користувачів.

Проблеми автентифікації виникають, якщо додаток:

- Незахищений від атак грубої сили або інших автоматизовані атаки.
- Дозволяє стандартні, слабкі або добре відомі паролі
- Використовує слабкі або неефективні процеси відновлення облікових даних і забутих паролів
- Використовує слабко хешовані сховища даних паролів
- Має неефективну багатофакторна автентифікація або не використовує її.
- Розкриває ідентифікатор сеансу в URL-адресі.
- Повторно використовує ідентифікатор сеансу після успішного входу.

Основні найкращі практичні рекомендації OWASP щодо пом’якшення вразливості порушеної автентифікації:

- Запровадження багатофакторної автентифікації.
- Уникнення створення нових облікових записів з даними за замовчуванням, особливо для користувачів із правами адміністратора.
- Використання надійних паролей.
- Блокування аккаунту якщо користувач більше 3 раз ввів неправильний пароль.
- Використання безпечного менеджеру сеансів, який генерує випадкові ідентифікатори сеансів з обмеженим часом.
- Уникнення відображення унікальних ідентифікаторів сеансу в URL-адресах.

Software and Data Integrity Failures

Порушення цілісності програмного забезпечення та даних відносится до коду та інфраструктури додатку, які не захищають від порушень цілісності даних.

Така небезпека може виникати коли програма, використовує плагіни, бібліотеки або модулі з ненадійних джерел, сховищ або мереж доставки вмісту (CDN). Неавторизований доступ, зловмисний код або компрометація системи – усе це може бути ризиком незахищеного процесу CI/CD.

Також, багато частин програмного забезпечення тепер мають можливості автоматичного оновлення, які дозволяють отримувати оновлення без необхідної перевірки цілісності даних та застосовувати їх до програм, які раніше були довіреними. Зловмисники потенційно можуть поширювати та запускати власні оновлення в усіх системах із цією функціональністю.

Для уникнення такого типи загроз слід використовувати наступні практики:

- Використання цифрових підписів
- Методичне виконання процедур перегляду коду та змін конфігурації.
- Використання тільки надійних бібліотек та залежностей, при роботі з npm або Maven.
- Уникнення передачі непідписаних або незашифрованих серіалізованих даних ненадійним клієнтам без певної перевірки цілісності даних або цифрового підпису для виявлення зловмисних змін.

Security Logging and Monitoring Failures

Збої роботи в системі логів та моніторингу часто є чинником серйозних інцидентів безпеки. Відсутність належного протоколювання, моніторингу або звітування про події безпеки, таких як спроби входу до системи, ускладнює виявлення підозрілої поведінки та значно підвищує ймовірність того, що зловмисник зможе успішно використати уразливості додатку.

Хакери можуть перевіряти програму або компоненти програмного забезпечення на відомі вразливості протягом певного періоду перед початком активних дій. Дозволяючи таким зондам продовжувати залишатися непоміченими, підвищується ймовірність того, що зловмисник зрештою знайде вразливе місце та

успішно використає недолік. Дослідження показують, що час від початку атаки до її виявлення може зайняти приблизно до 200 днів. Це вікно дає кіберзлодіям достатньо часу для втручання в роботу серверів, пошкодження баз даних, викрадення конфіденційної інформації та встановлення шкідливого коду.

Риск такої небезпеки у WEB-додатку є дуже високим якщо:

- Не реєструються важливі події, такі як невдалі входи та транзакції великих сум
- Логи і API не перевіряються на наявність підозрілої активності.
- Журнали логів зберігаються лише локально.
- Тестування на проникнення та сканування інструментами динамічного тестування безпеки додатків (DAST) не викликають попереджень безпеки.
- Програма не може виявляти активні атаки та сповіщати про них в режимі реального часу.

Для швидкого виявлення підозрілих дії та спроб неавторизованого доступу слід використовувати готове програмне забезпечення для ведення логів та аудиту додатку. При використанні такого програмного забезпечення, навіть якщо виявлена атака виявилася невдалою, логи та моніторинг нададуть безцінні інструменти для аналізу джерела та вектора атаки та вивчення того, як політику безпеки та елементи керування можна посилити для запобігання вторгненням.

Server-side request forgery

Підробка запитів на стороні сервера (SSRF) — це недолік WEB-безпеки, який дозволяє зловмиснику змусити код на стороні сервера надсилати запити HTTP до будь-якого домену, який вибере зловмисник. Коли WEB-додаток отримує ресурс без перевірки наданої користувачем URL-адреси, виникає помилка SSRF. Навіть якщо додаток захищений брандмауером, VPN або іншим списком контролю доступу до мережі, зловмисник може змусити її надіслати підроблений запит у несподіване місце. Оскільки сучасні додатки надають користувачам зручні та сучасні можливості, отримання URL-адреси стає звичайним сценарієм. Також, серйозність SSRF атак стає вищою через хмарні сервіси та складність архітектури додатку. Наслідком цього є стрімкий ріст SSRF атак.

Для уникнення таких типів атак можна приймати міри захисту на мережевому рівні та рівні самого WEB-додатку:

- На мережевому рівні

- Розділення функції віддаленого доступу до ресурсів у окремі мережі, для зменшення впливу SSRF.
- Застосування політики брандмауера «відмови доступу за замовчуванням»

- На рівні додатку

- Фільтрація усіх вхідних даних, наданих клієнтом.
- Оброблення запитів перед надання результатів клієнтам.
- Вимкнення перенаправлення HTTP запитів [8, 9].

Раніше OWASP Top 10 ніколи не створювався як основа для побудови захисту WEB-додатку, однак багатьом організаціям, які тільки починають шлях до безпеки додатків, важливо почати з чогось. OWASP Top 10 2021 є хорошим початком для базового контрольного переліку заходів безпеки, але це не є панацея від усіх загроз, яким слід протидіяти у сучасному WEB-додатку.

Останній список OWASP Top 10 містить більше ризиків, пов’язаних з контролем доступу та конфігурацією, ніж будь-коли раніше. Порушений контроль доступу та криптографічні збої займали дві перші позиції в списку у 2021 році. Супутні проблеми, такі як неправильна конфігурація безпеки та застарілі компоненти, також стали більш помітними.

2.2 Огляд технології WAF

2.2.1 Розвиток технології та різниця між WAF і звичайним брандмауером

У сучасну епоху просунутих кібератак і цифрових інновацій для компаній життєво важливо розуміти загрози безпеки і якими методами можна захиститися від руйнівних наслідків кібератак.

Брандмауер з самого початку зародження популярності всесвітнього павутиння до наших днів виступає основним методом захисту у мережі. Брандмауер працює методом фільтрації даних, що надходять до мережу. Він

аналізує дані, перевіряючи адресу відправника, програмне забезпечення, для якої ці дані призначені, і вміст даних. Поєднуючи ці частини, брандмауер визначає, який трафік є шкідливим, а який ні, і відповідно до отриманої інформації відкриває або закриває мережевий шлюз.

Перше згадування цієї технології приходиться на кінець 1980-х років разом з стрімким зростанням популярності та використання Інтернету. Ранні брандмауери були розроблені з простою технологією фільтрації пакетів, яка була першою технологією моніторингу зовнішніх з'єднань перед тим, як дозволити їм з'єднатися з мережевими пристроями. Ця технологія заклала основу для майбутньої еволюції брандмауера [9].

Традиційно підприємства захищали свої дані та користувачів за допомогою мережевих брандмауерів, яким бракує гнучкості для захисту від сучасних загроз безпеки. Проте, зі стрімким ростом методології роботи "Bring your own device", загальнодоступних хмарних середовищ та програмного забезпечення як послуги (SaaS) з'явилася необхідність у впровадженні брандмауеру WEB-додатків (WAF) до стратегії безпеки підприємств. Такий брандмауер посилює захист від кібератак на WEB-додатки, знаходяться на віддаленому сервері та працюють через інтернет за допомогою інтерфейсу WEB-браузера, який є привабливою ціллю для хакерів.

Звичайний мережевий брандмауер захищає захищену локальну мережу від несанкціонованого доступу, щоб запобігти ризику атак. Його основна мета — відокремити захищену зону від менш безпечної зони та контролювати зв'язок між ними. Без нього будь-який комп'ютер із загальнодоступною адресою Інтернет-протоколу (IP) доступний за межами мережі та потенційно загрожує атаці

WAF захищає WEB-додатки, орієнтуючись на трафік протоколу передачі гіпертексту (HTTP). Такий підхід відрізняється від стандартного брандмауера, який створює бар'єр між зовнішнім і внутрішнім мережевим трафіком.

WAF знаходиться між зовнішніми користувачами та самими WEB-додатками для аналізу всього HTTP-зв'язку. Потім він виявляє та блокує зловмисні запити до того, як вони досягнуть користувачів або ресурсів. WAF захищають критично важливі для бізнесу додатки та веб-сервери від загроз нульового дня та

інших атак на рівні додатків. Це стає дедалі важливішим у міру того, як компанії виходять на нові цифрові ініціативи, що робить нові веб-додатки та інтерфейси програмування додатків (API) уразливими до атак.

Основна технічна відмінність між брандмауером на рівні додатків та брандмауером мережевого рівня полягає в рівні безпеки, на якому вони працюють. Вони визначаються моделлю взаємозв'язку відкритих систем (OSI), яка характеризує та стандартизує функції зв'язку в телекомунікаційних і обчислювальних системах.

Брандмауери зазвичай зосереджені на мережевому та транспортному рівнях 3 і 4 OSI. Рівень 3 зосереджується на мережевих пакетах і їх передачі через вузли мережі. Рівень 4 стосується передачі даних від джерела до місця призначення.

Брандмауер додатків захищають атаки на 7 рівні моделі OSI, який є рівнем додатків. Рівень 7 зосереджений на взаємодії користувача з додатком та її інтерфейсом для роботи в мережі [10].

WAF захищає WEB-додаток, методом фільтрування, відстеження та блокування зловмисного HTTP трафіку, що надходить до WEB-додатку на постійній основі і запобігає несанкціонованих дій з конфіденційними даними. WAF знаходиться між додатком і клієнтом користувача, фільтруючи HTTP-трафік, який надходить та відправляється WEB-службами [11].

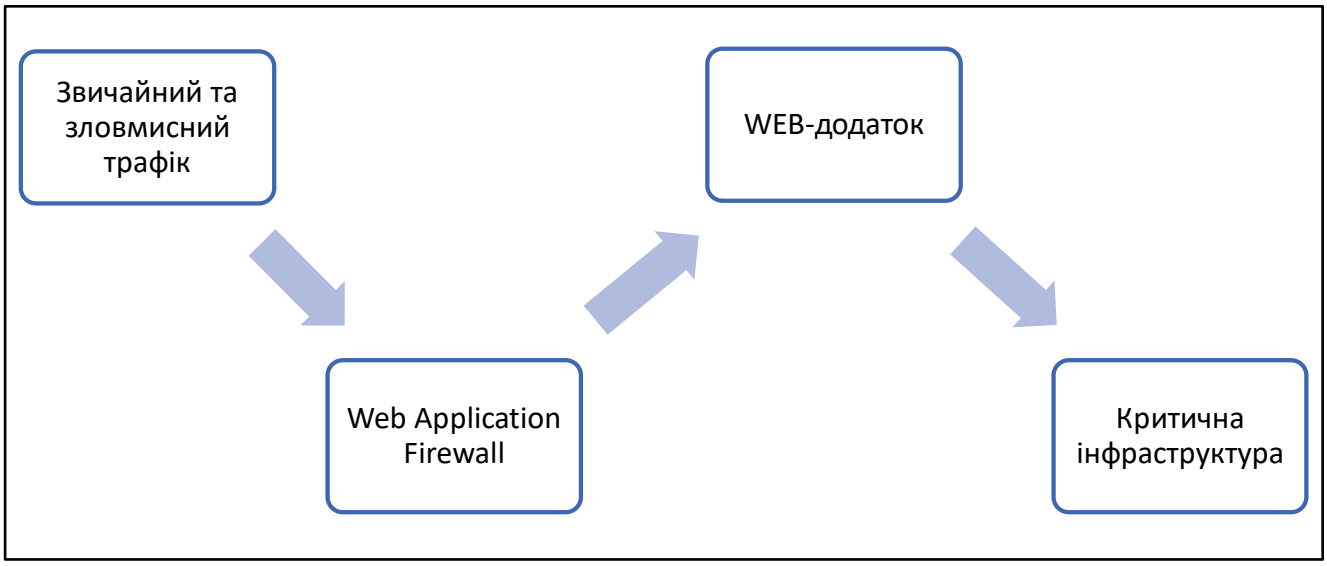


Рис. 2.2. Процес передачі даних через WAF

Брандмауери WEB-додатків були створені та стали популярними в другій половині 1990-х років, коли популярність WEB-додатків зростала з великими темпами і з’явилися перші серйозні проблеми кібератак. Ранні версії WAF захищали додатки від запитів з неправильними символами, але з тих пір вони еволюціонували, щоб мати можливості захисту від багатьох загроз і вразливостей кібербезпеки, з якими стикаються WEB-додатки.

2.2.2 Огляд технології WAF

Основні функції WAF працюють шляхом дотримання набору політик та правил, які допомагають визначити, який трафік є шкідливим, а який безпечним. Ці політики, дозволяють розгортати захист від атак, таких як XSS та SQL ін’єкції, порушення контролю доступу, а також багатьох інших найпоширеніших загроз безпеки з переліку Open Web Application Security Project [12].

Набір політик фаерволу складається на основі моделей безпеки. Основними моделями безпеки у типічному WAF виступають негативна та позитивна моделі безпеки [12]:

- Негативна модель безпеки

Така модель дозволяє пропускати весь трафік, якщо він не відповідає визначеним правилам та політикам. Якщо трафік відповідає установленим політикам – він блокується.

Негативна модель безпеки зазвичай реалізується за допомогою підходу на основі підписів, тобто унікальних прикметах загрози. Такий підхід спрямований на ідентифікацію зловмисних вхідних даних за допомогою методів зіставлення шаблонів, що дозволяє блокувати популярні загрози безпеки. Такий підхід є дуже простим в налаштуванні та використанні проте є дуже ненадійним і не захищає від нових загроз безпеки

- Позитивна модель безпеки

Позитивна модель є протилежною до негативною, тобто дозволяє пропускати трафік тільки, який відповідає визначеним правилам та політикам. Цей підхід є дуже складним у реалізації та вимагає досвідчених експертів та постійної

підтримки для коректної роботи. Хоча позитивна модель є тяжкою у реалізації, вона також має високу ефективність, оскільки увесь трафіку блокується по стандарту.

- Гібридна модель безпеки

Оскільки і позитивна, і негативна моделі мають свої мінуси та плюси, впровадження комбінації обох моделей у багатьох випадках може дати найкращий результат.

Такий тип моделі дозволяє пропускати трафік, який був спеціально дозволений, і блокує трафік, який був заблокований

Існує також три основних способу реалізації WAF [13]:

- Cloud-based WAF

Хмарний WAF — це доступний і простий у реалізації варіант розгортання WAF який можна швидко розгорнути за допомогою ресурсів, які надає стороння компанія-постачальник фаєрволу. Такий тип WAF зазвичай заснований на підписці та має мінімальні початкові витрати. Хмарні WAF мають доступ до бази даних сучасних загроз, яка постійно оновлюється. Такі WAF також можуть пропонувати служби керування, які допоможуть визначити правила безпеки та реагувати на атаки, коли вони відбуваються.

В останні роки хмарні WAF стали переважаючим типом розгортання для більшості організацій у всьому світі, оскільки вони здатні забезпечувати високий рівень безпеки з мінімальними початковими інвестиціями та без потреби у великому внутрішньому досвіді розгортання безпеки.

- Hardware-based WAF

Апаратний(Мережевий) WAF інсталюється локально в мережі. Такий тип розгортання є найдорожчий з усіх трьох типів форми, оскільки для такі WAF вимагають команду обслуговування і підтримки працездатності та місця для зберігання. Основне призначення такого підходу це мінімізація затримок та максимізація безпеки.

Апаратні WAF зазвичай використовуються великими організаціями, які мають бюджет і команду для керування локальними пристроями та IT-

інфраструктурою. Крім того, організації використовують апаратні WAF, коли швидкість і продуктивність додатків є критичними, або під час запуску конфіденційних програм у локальних середовищах, таких як державні установи, агентства національної безпеки, оборонна промисловість тощо.

Останніми роками WAF на основі апаратного забезпечення дедалі більше застарівають, оскільки затрата такої великої кількості ресурсів та грошей просто не коштує того, тим паче при можливості розгортання хмарного WAF.

- Software-based WAF

Програмний WAF є альтернативою апаратного. WAF розгорнутий на основі програмного забезпечення працює завдяки віртуальному пристрою або агенту та може розгортатися локально, у приватній хмарі або в загальнодоступній хмарі.

WAF на основі програмного забезпечення зазвичай використовують організації з додатками, розміщеними в приватних або публічних хмарних центрах обробки даних. Також такий тип розгортання може бути популярними серед організацій, які не мають бюджету та можливостей підтримувати апаратні WAF, але все одно хочуть підтримувати власний WAF без посередника.

Таким чином можна зробити висновок, що використання програмного та апаратного WAF все більш застарівають, коли використання хмарного WAF є найкращим вибором для забезпечення безпеки у WEB-додатках сьогодні.

Основна перевага таких брандмауерів порівняно з локальним полягає в тому, що віртуальне посилення безпеки та встановлення виправлень набагато простіше і ефективніше в хмарному середовищі. Такий захист є досить ефективний, навіть без оновлення системи менеджменту контенту(CMS), плагінів, та іншого програмне забезпечення WAF до останньої безпечної версії. Хмарні брандмауери також не перевантажують ресурси додатку або серверу, оскільки локальні брандмауери обмежені доступністю ресурсів сервера постачальника програми.

Для впровадження хмарного WAF зазвичай є простішим і доступнішим процесом, ніж налаштування локального рішення. Щоб почати, вам потрібно зареєструватися в службі WAF і налаштувати її для роботи з вашою WEB-програмою.

Сучасні хмарні WAF надають елементи керування безпекою та можливості, подібні до локальних WAF. До них належать [14, 15]:

- База даних шаблонів та сигнатур загроз

Сигнатури загроз — це шаблони, які можуть вичислити зловмисний трафік, аномальні відповіді сервера та відомі шкідливі IP-адреси. Раніше WAF покладалися переважно на бази даних шаблонів атак, які були менш ефективними проти нових або невідомих атак.

- Профілювання додатку

Профілювання передбачає аналіз структури програми, включаючи типові запити, URL-адреси, значення та дозволені типи даних. Ця функція дозволяє WAF ідентифікувати та блокувати потенціально небезпечні загрози

- Аналіз трафіку за допомогою AI

Алгоритми штучного інтелекту дозволяють аналізувати поведінку моделей трафіку, використовуючи базові показники поведінки для різних типів трафіку. Цей метод дозволяє виявляти небезпечні аномалії роботи, які не відповідають відомим шкідливим шаблонам або не були виявлені попередніми методами.

- Механізм кореляції

Такий механізм аналізує вхідний трафік і сортує його за відомими сигнатурами атак, профілюванням додатків, аналізом штучного інтелекту та спеціальними правилами, щоб визначити, чи слід його блокувати.

- Кастомізація

Спеціалісти безпеки можуть створювати свої, унікальні правила безпеки, що застосовуються до трафіку програми. Це дозволяє організаціям налаштовувати поведінку WAF відповідно до своїх потреб і запобігати блокуванню звичайного трафіку.

- Платформи захисту від DDoS

У багатьох сучасних WAF є можливість інтегрування хмарної платформи, яка захищає від DDoS атак. Якщо WAF виявляє DDoS-атаку, він може перенаправити трафік на платформу захисту від DDoS, яка може впоратися з великою кількістю запитів.

- SSL/TLS шифрування

Шифрування SSL/TLS можна використовувати для шифрування трафіку у WEB-додатку. Цей процес захищає дані під час проходження трафіку до WEB-додатку та при виході з нього від перехоплення та читання зловмисниками.

2.3 Порівняння сучасних рішень безпеки WAF

Вибір сучасного рішень WAF залежать від конкретних потреб і вимог компанії. Якщо потрібен базовий, проте надійний рівень захисту, може бути достатньо безкоштовного рішення, такого як Cloudflare. Для середнього бізнесу підійдуть рішення з більш розширеним захистом, кращим вибором з цієї категорії можуть стати комплексне рішення безпеки, наприклад Imperva Incapsula або Prophaze. Зрештою, важливо порівняти характеристики та ціни, щоб знайти найкращий варіант для організації.

Для розгляду було взято три сучасних рішення WAF — CloudFlare, AWS WAF та Prophaze [16]:

Cloudflare є брандмауером, який дуже успішно захищає WEB-хости від DDoS та інших популярних атак. Цей сервіс має безліч функціоналу для розширення захисту. Служба брандмауера Cloudflare поєднує зворотний проксі-сервер із мережею доставки вмісту, надаючи ряд додаткових функцій безпеки та оптимізації. Cloudflare — це надзвичайно потужний брандмауер із чудовими функціями безпеки, ефективною оптимізацією WEB-додатку, швидкою глобальною мережею та інтуїтивно зрозумілим дизайном програми. Головним плюсом цього рішення є його доступність та безкоштовна версія, також Cloudflare є перевірений часом і доволі безпечний, що робить його гарним вибором для захисту WEB-додатку. Сервіс є дуже привабливим варіантом для нових або малих бізнесів і якість цього сервісу дуже важко перевершити.

Prophaze WAF — це хмарний проксі-сервер, який діє як брандмауер WEB-додатків. Основною перевагою Prophaze є функції штучного інтелекту, які вдосконалюють правила виявлення загроз шляхом коригування базової лінії

стандартної поведінки. Ця функція допомагає зменшити кількість помилкових загроз і надати відвідувачам сайту безперебійний доступ до контенту. При застосуванні до трафіку, система аналітики поведінки Prophaze налаштує обробку конкретних типів трафіку та переосмислить визначення зловмисного трафіку.

Сама система Prophaze працює з контейнерами Kubernetes і також може відстежувати продуктивність і безпеку дій Kubernetes у системі, а також виконувати традиційне виявлення активності хакерів.

Prophaze є хорошим вибором для підприємств, які хочуть самостійно керувати своїми WAF, але не мають високоякісного досвіду безпеки та спеціалістів, щоб точно визначити політику безпеки. Оскільки ці політики з часом коригуються за допомогою аналізу поведінки WAF, помилки, допущені у визначенні політик безпеки, зрештою будуть виправлені.

Брандмауер веб-додатків Amazon AWS — це надійне та ефективне рішення для безпеки WEB-додатків. AWS WAF надає кілька політик захисту, керованих за допомогою AWS, і також декілька популярних політик сторонніх служб для більш ефективного захисту. Його легко налаштувати та інтегрувати з CloudFront, балансир навантаження або шлюзом API. Основним плюсом рішення є ефективність з фінансової точки зору, оскільки треба платити тільки за послуги, якими користується компанія.

AWS WAF є надійним та перевіреним рішенням з багатим функціоналом та простим налаштуванням, однак AWS WAF доступний лише для клієнтів, які використовують WEB-сервіси компанії Amazon. Таким чином, це рішення є дуже ефективним для підприємств, які хочуть використовувати рішення AWS як основу для бізнесу в інтернеті. При такому підході WEB-додаток буде мати дуже високий рівень захисту, тому що AWS WAF добре інтегруєтеся та є оптимізований спеціально для WEB-сервісів WAF.

Таблиця 2.1

Порівняння сучасних рішень безпеки WAF для захисту WEB-додатку

WAF	Cloudflare	AWS WAF	Prophaze
Ціна	Безкоштовна версія; Про: 20$ на місяць; Бізнес: 200$ на місяць;	Web ACL: 5$ на місяць; Правило: 1,00$ на місяць Запит: 0,60$ за 1 мільйон запитів.	Безкоштовна пробна версія; Від 299 $ на місяць
Найкраще підходить для	Особистого використання, малого і середнього бізнесу, а також підприємства високого рівня.	Можливість масштабування для підприємства будь-якого розміру, якщо вони є клієнтами AWS	Середніх та великих підприємств; WEB-додатків на основі публічной хмарної середи (AWS/Azure/GCP), приватної хмарної середи
Типи захисту	OWASP Top 10; Захист ключових портів; DDoS атаки; Блокування загроз на основі чорних списків, HTTP запитів та аналізу трафіку.	OWASP Top 10; DDoS атаки; Кастимізовані правила та політики безпеки;	OWASP Top 10; Захист від ботів; Пом'якшення DDoS атак; Виявлення та блокування загроз на основі поведінки трафіку;

Продовження таблиці 2.1

Особливості та переваги	Ведення журналів логів і звітності; Відстеження проблем безпеки; Моніторинг проблем безпеки; Аналітика безпеки;	Гнучкий захист від WEB-атак; Простота розгортання та обслуговування; Ефективність у плані коштів; Безпека при розробці;	Захист від загроз на основі штучного інтелекту; Пом'якшення бот-нетів; Інформаційна панель у реальному часі;

Таким чином, розглянувши декілька лідерів ринку технології WAF, можна сказати, що хоча Prophaze і є ефективним брандмауером з просунутим штучним інтелектом, таке рішення більше підходить для середніх та великих підприємств на основні хмарних технології додатків у зв’язку з вартістю та технологіями, які потребують команди спеціалістів безпеки, коли Cloudflare є більш розповсюдженим та доступним рішенням як для підприємств, так і для особистого використання.

AWS є гарним рішенням як з фінансової точки зору, так і з точки зору ефективності роботи та безпеки, проте використовувати його можуть тільки клієнти системи AWS. Таким чином, це рішення є найкращим для клієнтів системи AWS, або підприємств, які хочуть розгорнути бізнес на інфраструктурі AWS.

Незалежно від того, яке за розміром є підприємство, брандмауер при утриманні WEB-додатків є головним пріоритетом безпеки. Жоден бізнес чи власник WEB-додатку не може дозволити собі втратити конфіденційні дані, активи WEB очорнення -сайту та дані про фінансові операції.

Рішення щодо вибору брандмауера WEB-додатку не таке просте, як здається, і кожне рішення рекомендується приймати згідно з ситуацією. Перед вибором рекомендується детально оцінити функції кожного рішення та скористатися безкоштовними пробними версіями, перш ніж купувати будь-яке конкретне.

Висновки до розділу 2

В даному розділі було досліджено критичні загрози безпеки WEB-додатків на основі OWASP Top 10. Виявлено, що основними загрозами безпеки на сьогодні є порушений контроль доступу та криптографічні збої. Супутні проблеми, такі як неправильна конфігурація безпеки та застарілі компоненти, також стали більш значущими.

Був проведений аналіз технології WAF, його еволюція та порівняння зі звичайним брандмауером. Виявлено, що найкращий варіант розгортання WAF на сьогодні є хмарний, коли мережеві WAF є більш складні у експлуатуванні та менш ефективні в плані коштів.

Також, було виокремлено рішення WAF та проведений їх аналіз та порівняння.

Згідно з темою проекту підкреслено можливості Cloudflare WAF як одного з лідерів ринку та кращих рішень для забезпечення безпеки у WEB-додатках.

3 ПРОЦЕС ІНТЕГРАЦІЇ ТА НАЛАШТУВАННЯ WAF

3.1 Можливості технології Cloudflare WAF

Брандмауер Cloudflare WAF є сучасним рішенням безпеки WEB-додатків і APl. До можливостей цього рішення відносится запобігання DDoS атакам, блокування ботів, виявлення аномалії і шкідливого трафіку та моніторинг периметру роботи браузеру. Ключовими функціями рішення Cloudflare WAF є [17]:

- Рівневий захист за допомогою наборів правил WAF

Зупинення зловмисного навантаження компонентів за допомогою наборів правил:

- Правила, керовані Cloudflare
- Набори правил сторонніх розробників (OWASP Top 10)
- Кастомні набори правил для зупинки будь-яких атак

- WAF ML.

Виявлення шкідливого трафіку на основі машинного навчання та штучного інтелекту

- Оновлені правила для захисту від атак нульового дня.

Постійне оновлення правил від атак нульового дня командою Cloudflare без потреби встановлення патчів або оновлень

- Унікальний набір правил Cloudflare для основних CMS та електронної комерції.

Отримання захисту WEB-додатків при налаштуванні додатків на популярних платформах, таких як: WordPress, Joomla, Plone, Drupal, Magneto, IIS тощо.

- Просунуте обмеження кількості запитів.

Захист від DDoS і BruteForce атак шляхом обмеження швидкості окремих IP-адрес або за допомогою HTTP заголовків, ASN або країни.

- База даних репутації IP адресів.

Блокування підключення зі зловмисних IP-адрес за допомогою штучного інтелекту у режимі реального часу з базою даних понад 1 млрд. унікальних адресів

- Запобігання втраті даних.

Блокування відповідей, що містять конфіденційні дані, наприклад, ідентифікаційну та фінансову інформацію, кредитні картки або ключі API.

- Перевірка незахищених облікових даних.

  Виявляння BruteForce атак з викраденими обліковими даними

- SSL/TLS.

  Гнучке налаштування трафіку SSL у додатку.

Cloudflare є потужним інструментом безпеки, який вже при інсталяціє надає низку автоматично налаштованих можливостей та засобів безпеки, до яких входить налаштований брандмауер на основі набору правил Cloudflare, протекція від DDoS атак та інше. Все це робить Cloudflare лідером серед засобів надання безпеки на ринку та прекрасним вибором для захисту як для маленьких та середніх, так і для корпоративних мереж

3.2 Інсталяція та налаштування Cloudflare WAF

Перед початком інтеграції WAF треба зареєструватися на офіційному сайті Cloudflare та зробити персональний аккаунт. Сервіс Cloudflare надає безліч корисних функції для утримання WEB-сайтів та WEB-додатків. До основних можливостей сервісу відносяться:

- Мережа доставки контенту
- Система доменних імен
- Балансир навантаження
- Прискорення мобільних сторінок
- Можливості кешування
- Потокове відео
- DDoS-захист
- WAF
- Підтримка SSL/TLS
- DNSSEC

- Аналітика
- Реєстратор доменів
- Робітники (для розробників)

Для початку роботи з WAF треба зайти на сервіс та додати WEB-ресурс до створеного аккаунту. Зробити це можна у вкладці Websites або на кнопці Add site, яка знаходиться у шапці сервісу. Для демонстрації було взято тестовий ресурс.

Першим кроком встановлення WAF є вибір плану підписки. Для початку можна взяти безкоштовний план для тестування сервісу, а після чого можна змінити план пізніше.

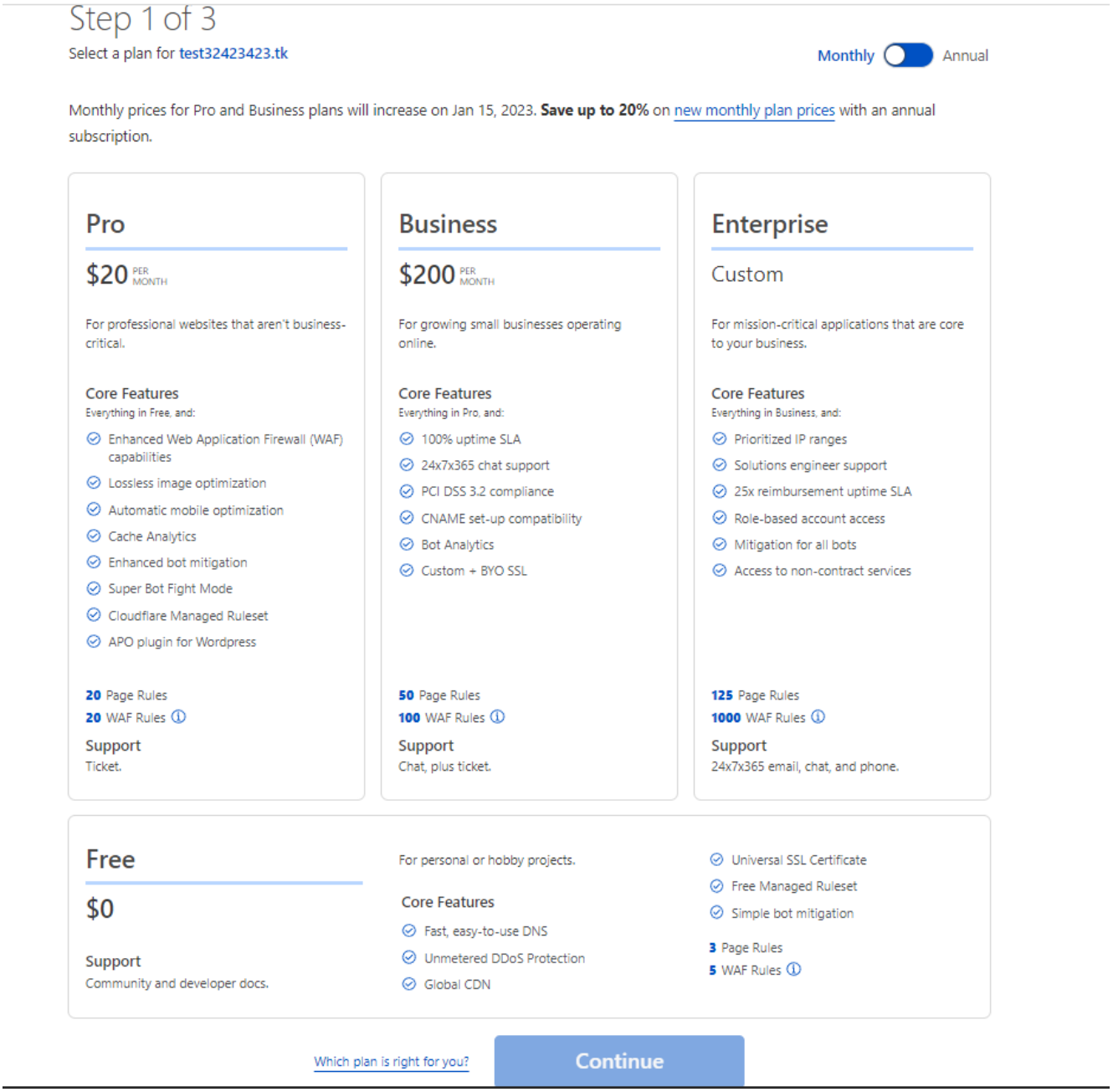


Рис 3.1 Вибір плану підписки

Наступним кроком треба налаштувати DNS сервер для підключення ресурсу до Cloudflare. Сервіс просканує WEB-ресурс та виявить усі DNS підключені до ресурсу, якщо такі є. Для роботи Cloudflare надає унікальні для кожного користувача доменні імена для роботи сервісу.

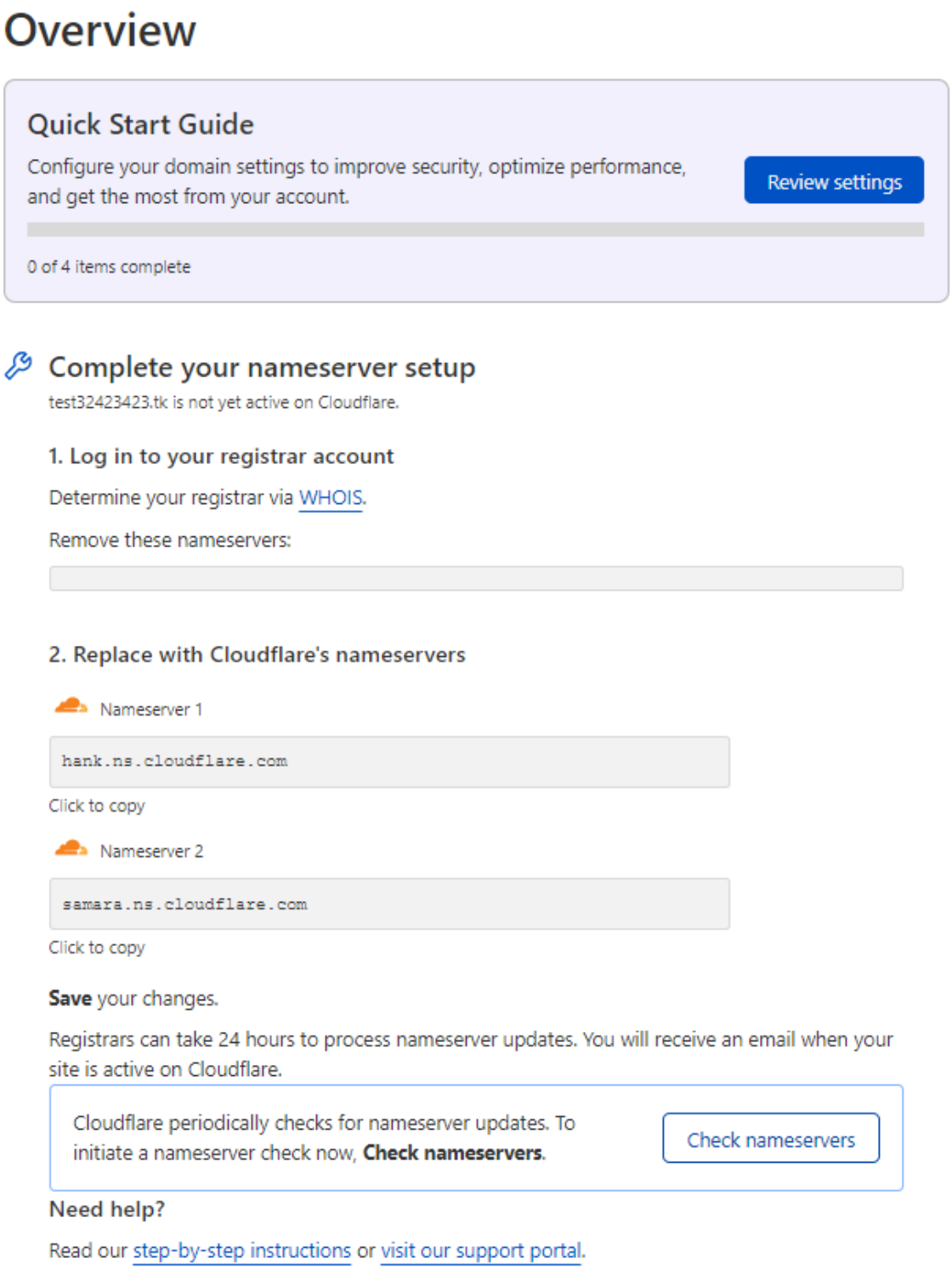


Рис 3.2 Налаштування DNS Cloudflare

Потрібно встановити ці унікальні доменні імена у системі, яка надає хостінг для ресурсу, який потрібно захистити. Існує безліч ресурсів хостінгу, проте на кожному процедура буде майже однакова. Треба зайти у налаштування та вибрати кастомні імена серверів.

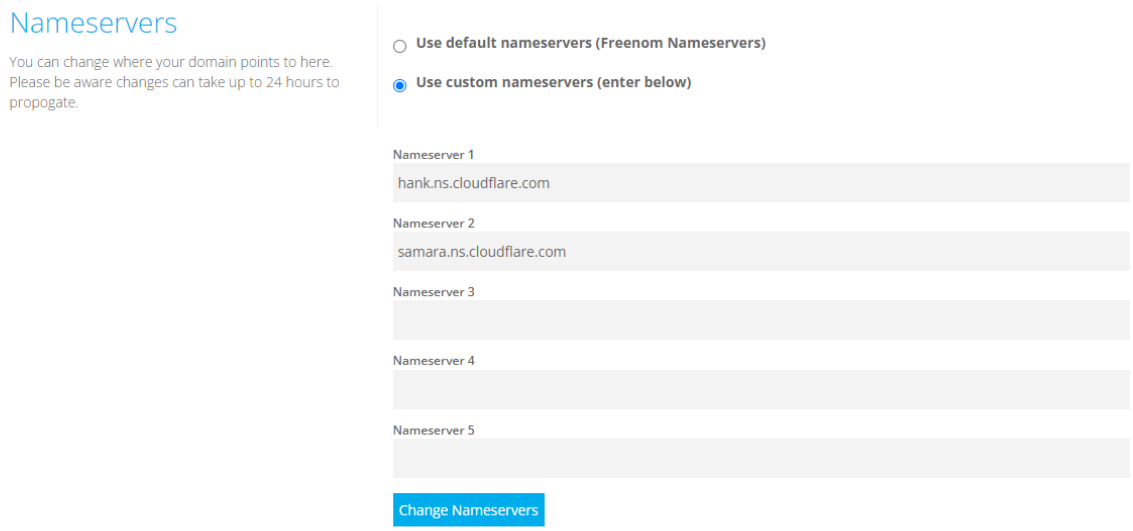


Рис 3.3 Встановлення кастомних DNS Cloudflare

Процес зміни доменних імен може зайняти до 24 годин, після чого WEB-додаток активується в системі Cloudflare.

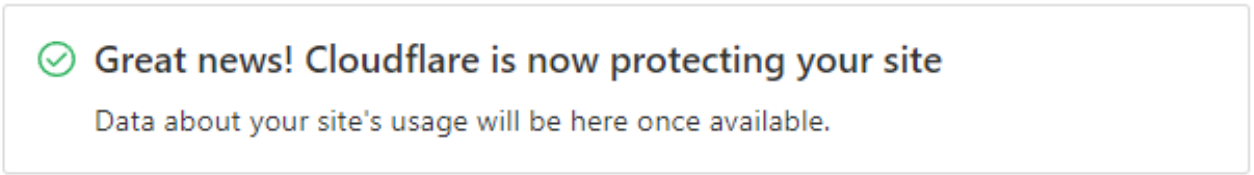


Рис 3.4 Успішна активація Cloudflare для WEB-ресурсу

Після активації Cloudflare запропонує швидке керівництво для початку роботи. Сервіс пропонує три кроки для початку роботи, а саме:

1. Підвищення безпеки
2. Оптимізація роботи коду WEB-додатку
3. Резюме внесених змін

Для початку треба увімкнути Автоматичні перезаписи HTTPS для виправлення помилок при переході з HTTP (Рис. 3.5), після чого ввімкнути використання HTTPS всіх сторінок WEB-ресурсу для шифрування даних (Рис. 3.6). Це дозволить уникнути багатьох вразливостей при передачі даних між сторінками ресурсу.

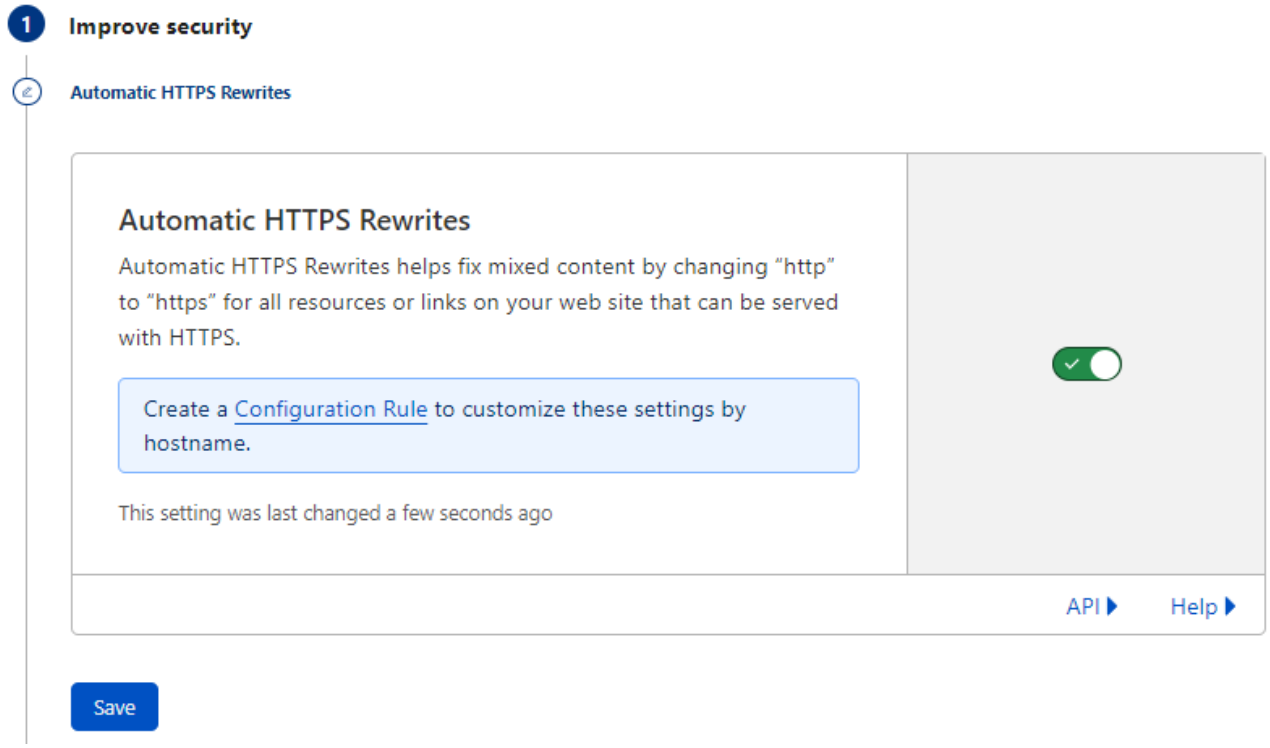


Рис. 3.5 Увімкнення автоматичних перезаписів HTTPS

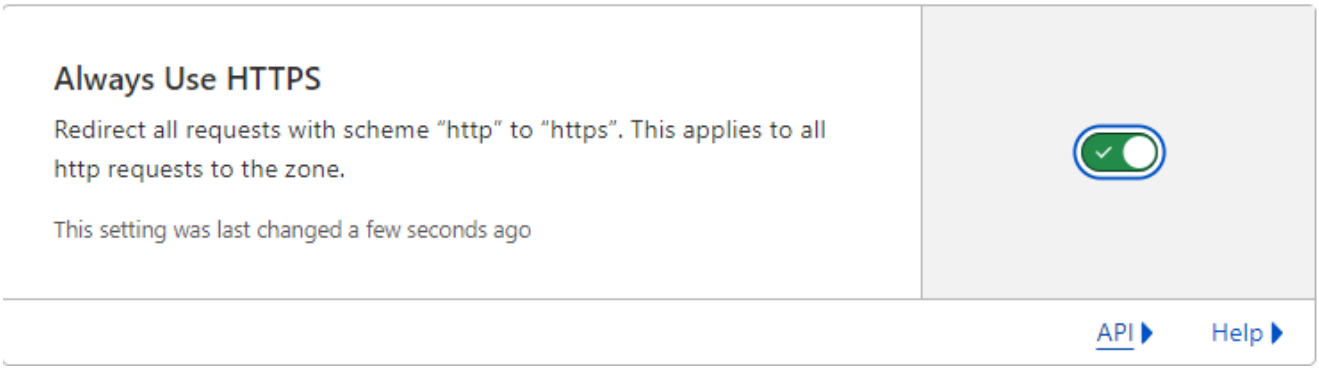


Рис. 3.6 Увімкнення функції постійного використання HTTPS

Далі будуть запропоновані функції оптимізації коду WEB-додатку, які можна вмикати за бажанням, оскільки вони не мають відношення до безпеки, після чого можна передивитись налаштовані функції у пункті Summary.

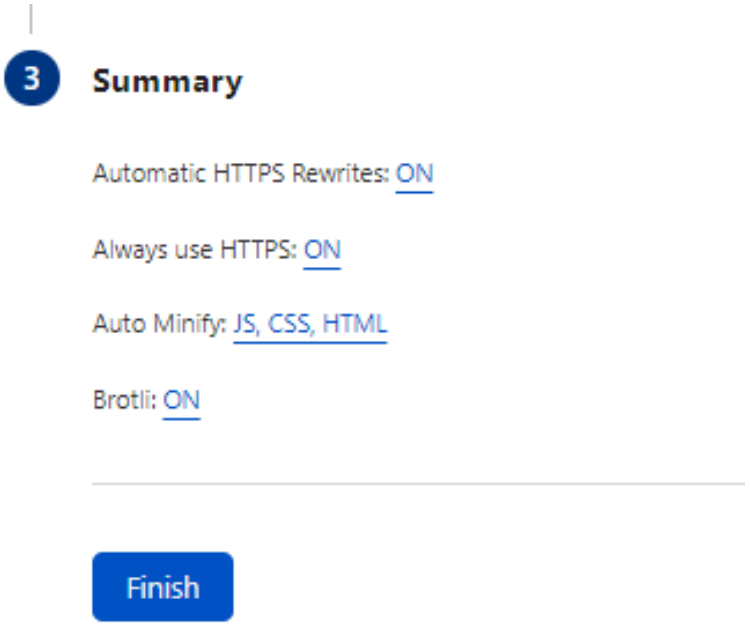


Рис. 3.7 Пункт “Summary”

На цьому етапі базові налаштування майже закінчені, проте є декілька простих кроків для швидкого підвищення безпеки WEB-додатку.

Важливим кроком є посилення шифрування. Для цього треба перейти до вкладки “SSL/TLS” на панелі управління. Тут знаходиться контроль шифрування та протоколів для шифрування. На панелі “Overview” слід виставити повний строгий режим шифрування. Такий режим слід використовувати завжди, якщо компанія, яка використовує сервіс не має своїх безпечних протоколів шифрування. Цей режим встановлює HTTPS з’єднання не тільки між користувачем і Cloudflare, а і між Cloudflare та сервером WEB-додатку, а також перевіряє достовірність сервера, що дозволяє уникнути зловмисного перехвату трафіку та використання хакерами їхнього протоколу для оману системи.

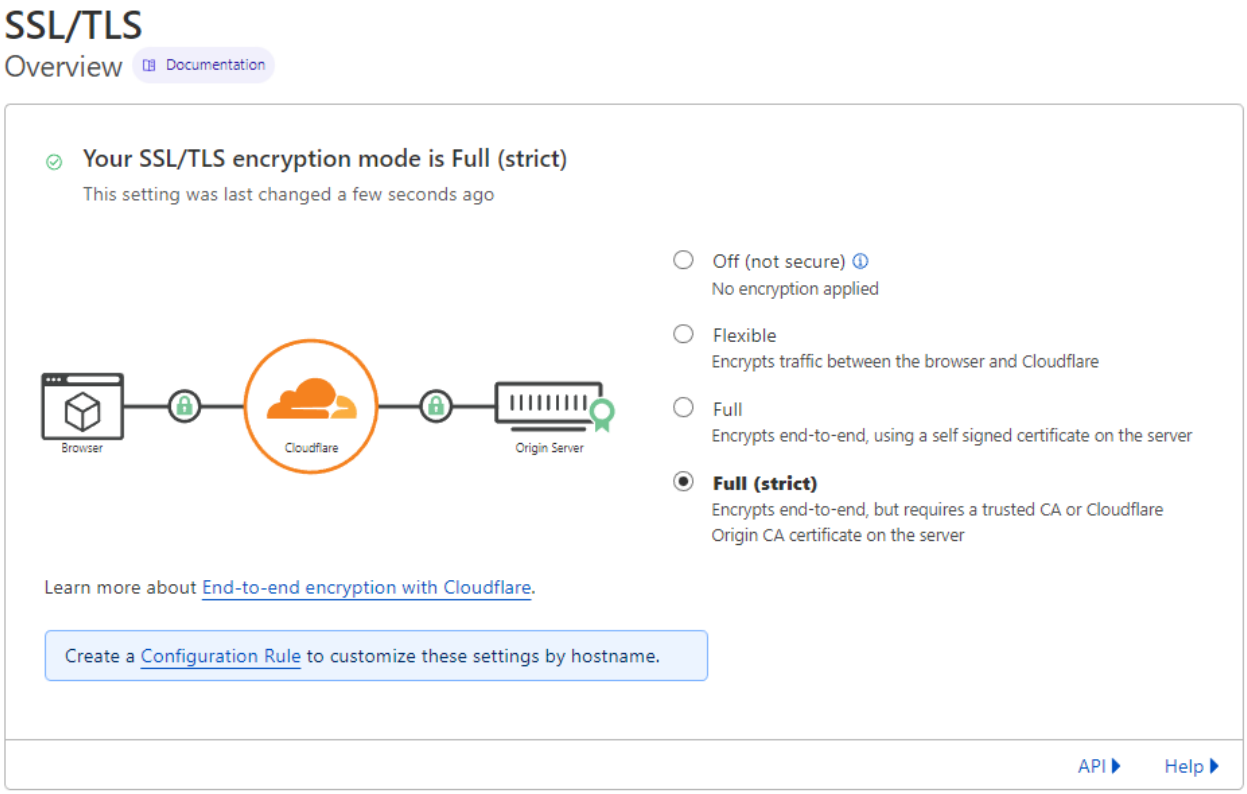


Рис. 3.8 Налаштування режиму шифрування

Як вже було висвітлено раніше, у частині проблематики безпеки, боти, або атаки ботнетів є одною з ключових загроз безпеки WEB-ресурсів на сьогоднішній день, тому в Cloudflare є спеціальна функція боротьби проти ботів

Дії режиму боротьби проти ботів цілком залежать від плану підписки. При безкоштовній версії є можливість блокування ботів тільки з хмарних систем ASN та браузерів, які не надають заголовків при HTTP запитах.

Система виявлення ботів працює за допомогою двигунів виявлення ботів, при безкоштовній та про версії це двигун евристики та механізм Javascript detections.

Двигун евристики обробляє усі запити та проводить низку перевірок для ідентифікації автоматичного трафіку, зіставляючи з базою даних зловмисних шаблонів

Механізм JavaScript Detections (JSD) виявляє браузери які не надають заголовків при HTTP запитах та інші зловмисні прикмети. Цей механізм виконує легку, непомітну ін’єкцію JavaScript коду на стороні клієнта будь-якого запиту,

дотримуючись суворих стандартів конфіденційності Cloudflare. Механізм JSD блокує, інформує про загрозу або передає запити іншим механізмам.

Для бізнес версії доступне виявлення ботів на основі штучного інтелекту та машинного навчання. Такий механізм відповідає за більшість усіх знаходжень зловмисного трафіку, як людського, так і ботів. Такий підхід використовує глобальну мережу Cloudflare, яка щодня обробляє мільярди запитів, щоб ідентифікувати будь який небезпечний трафік. Таким чином, механізм машинного навчання постійно вдосконалюється та стає точнішим і адаптується до нових загроз.

Для корпоративних версій також доступний двигун виявлення аномалій. Такий механізм працює методом дослідження базового рівня трафіку WEB-ресурсу та на основні штучного інтелекту виявляє не неприродній або аномальний трафік. Такий метод є дуже потужним та часто призводить до хибних виявлень небезпеки, тому його слід використовувати тільки в критичній інфраструктурі під постійним наглядом спеціалістів кібербезпеки [18].

Щоб ввімкнути двигун боротьби проти ботів, треба перейти у вкладку “Security” на панель “Bots” та включити “Bot Fight Mode”

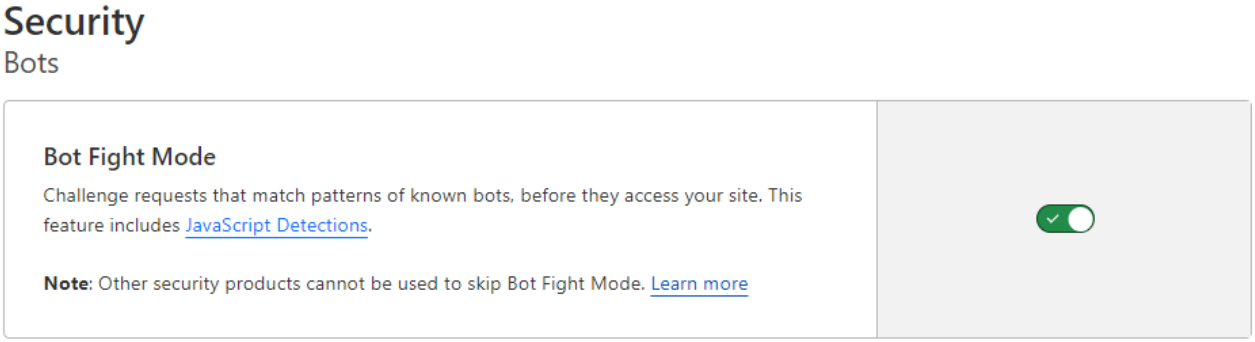


Рис. 3.9 Включення режиму боротьби проти ботів

Cloudflare є сучасним та ефективним програмним забезпеченням, над яким працюють багато професіоналів у сфері кібербезпеки, тому Cloudflare є вже оптимізованим на захист ресурсів при підключення до системи. Однак Інтернет є дуже великим і на кожну низку бізнесу є свої види загроз, тому для підприємств різних сфер та розмірів потрібні відповідні заходи безпеки. Для цього існують так

звані правила та шаблони безпеки. Cloudflare надає різні шаблони правил безпеки згідно до плану. У безкоштовному плані по стандарту налаштований безкоштовний керований набір правил Cloudflare для брандмауеру. Такий план безпеки розроблено, щоб забезпечити пом'якшення вразливостей високого та широкого впливу на безпеку. Правила безпечні для розгортання в більшості програм.

Для більш просунутих планів Cloudflare надає кастомні керовані правила безпеки. Створений командою безпеки Cloudflare, цей набір правил забезпечує швидкий і ефективний захист для різних типів WEB-додатків. Набір правил постійно оновлюється, щоб охопити нові вразливості та зменшити помилкові спрацьовування безпеки. Cloudflare надає набори правила, відповідно до стеку технології, наприклад є спеціальний набір правил для користувачів wordpress.

Cloudflare має свій стек правил для загроз OWASP Top 10. Набір правил Cloudflare OWASP — це реалізація базового набору правил OWASP ModSecurity від Cloudflare. Cloudflare регулярно відстежує наявність оновлень від OWASP на основі останньої версії, доступної в офіційному сховищі коду.

Ці правила розроблено як єдине ціле для обчислення оцінки загрози та виконання дії на основі цієї оцінки. Коли правило відповідає запиту, оцінка загрози збільшується відповідно до оцінки правила. Якщо оцінка загрози перевищує налаштований поріг, Cloudflare виконує дії для уникнення небезпеки.

Поріг спрацьовування кожного правила налаштовується так званими рівнями параної, від 1 до 4 рівня. Більш високі рівні параної забезпечують підвищений захист, але можуть призвести до блокування більшої кількості звичайного трафіку через помилкові спрацьовування.

Також є керований набір правил Cloudflare Exposed Credentials Check — це набір попередньо налаштованих правил для відомих програм CMS, які виконують пошук у загальнодоступній базі даних викрадених облікових даних [19].

Оптимальним рішенням буде використання кастомного набору правил Cloudflare та Cloudflare OWASP і їх коригування, дивлячись на потреби конкретного ресурсу.

Для налаштування правил безпеки треба перейти до вкладки “Security” після чого на сторінку “WAF”, а потім до вкладки “Managed rules”, де можна вмикати та вимикати набори правил, які надає Cloudflare та налаштовувати їх.

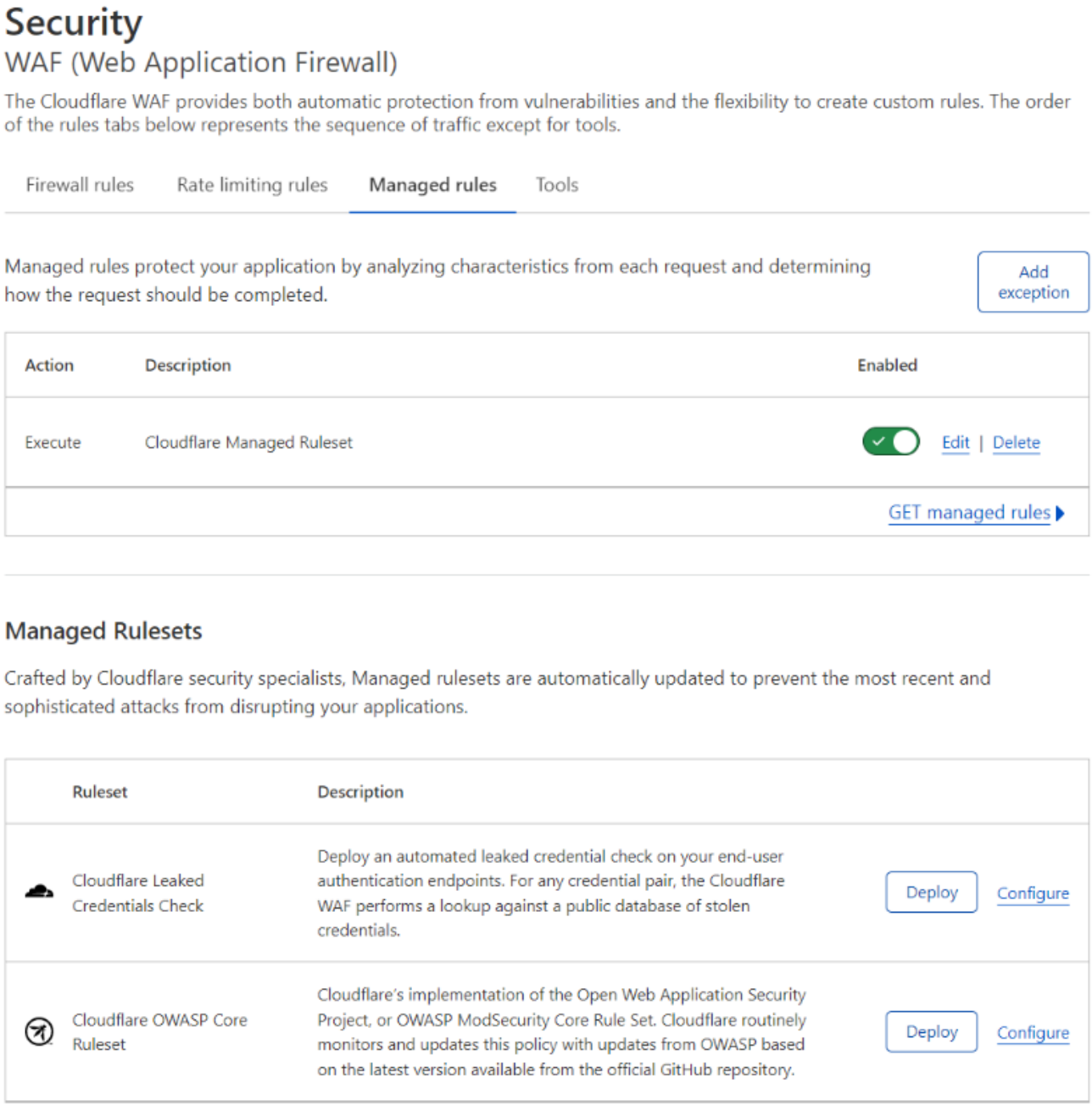


Рис. 3.10 Налаштування набору правил Cloudflare

Для додавання нового правила WAF треба перейти до вкладки “Security” після чого на сторінку “WAF”, а потім до вкладки “Firewall rules”. У цій вкладці можна переглянути правила, які зараз діють, або додати нові. Кількість правил WAF також залежить від рівню підписки. У безкоштовній версії доступно 5

кастомних правил безпеки. Щоб додати нове правило, треба натиснути на кнопку “Create firewall rule”.

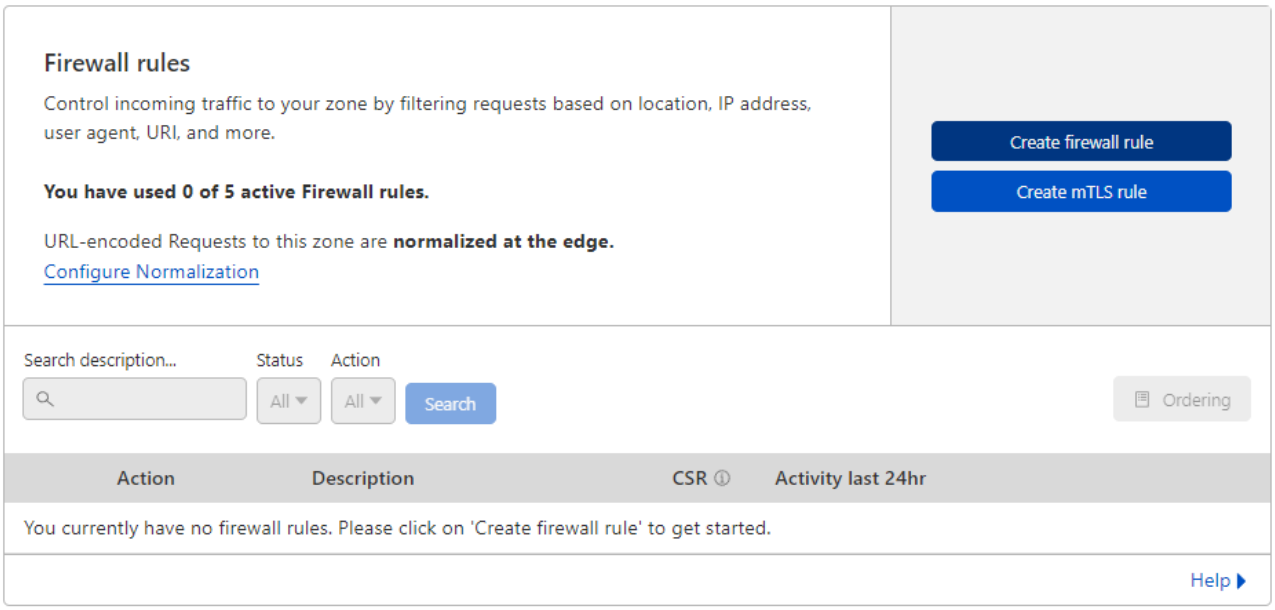


Рис. 3.11 Налаштування правил брандмауеру.

Для того, щоб створити правило, треба вибрати певний параметр, по якому правило буде працювати. Цим параметром може бути певна країна, IP адреса, cookie файли, url, тощо. Також, треба вибрати оператор та значення параметру. Після чого вказати дію яку треба застосовувати до даного параметру, наприклад, заблокувати, дозволити чи надати певну легку задачу або капчу для доступу до ресурсу. Якщо WEB-сервіс працює тільки в Україні, можна створити правило, яке буде надавати певну задачу для входу на сайт, якщо клієнт не є з України, для уникнення небажаного трафіку. Для цього в параметрі треба указати “country”, в операторі поставити “does not equal”, а значення “Ukraine”. Дію слід обрати “Managed challenge” або “Block” для повного знешкодження трафіку з інших країн.

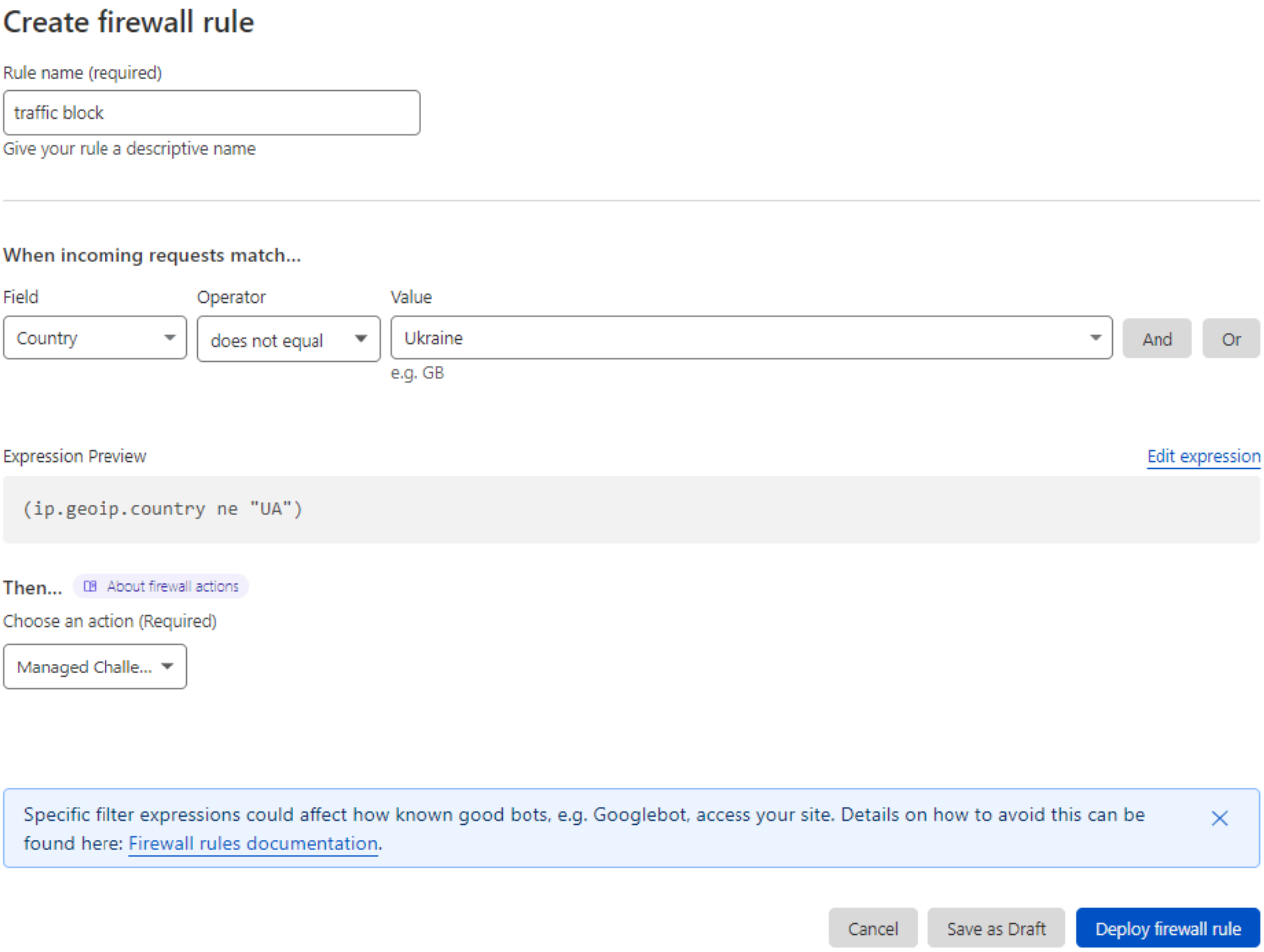


Рис. 3.12 Створення кастомного правила для брандмауеру.

Далі, для підвищення захисту, гарною практикою буде блокування посилання на захищені зони WEB-додатку, особливого, якщо він побудований на основі такого популярного сервісу, як WordPress. Для цього треба створити правило, яке буде блокувати трафік до популярних посилань, наприклад, експлоіту xmlrpc у WordPress. Проте, у такому випадку для доступу до цих ресурсів треба також створити правило, яке буде дозволяти дозвіл з певних Ip-адресів для роботи у цих ресурсах.

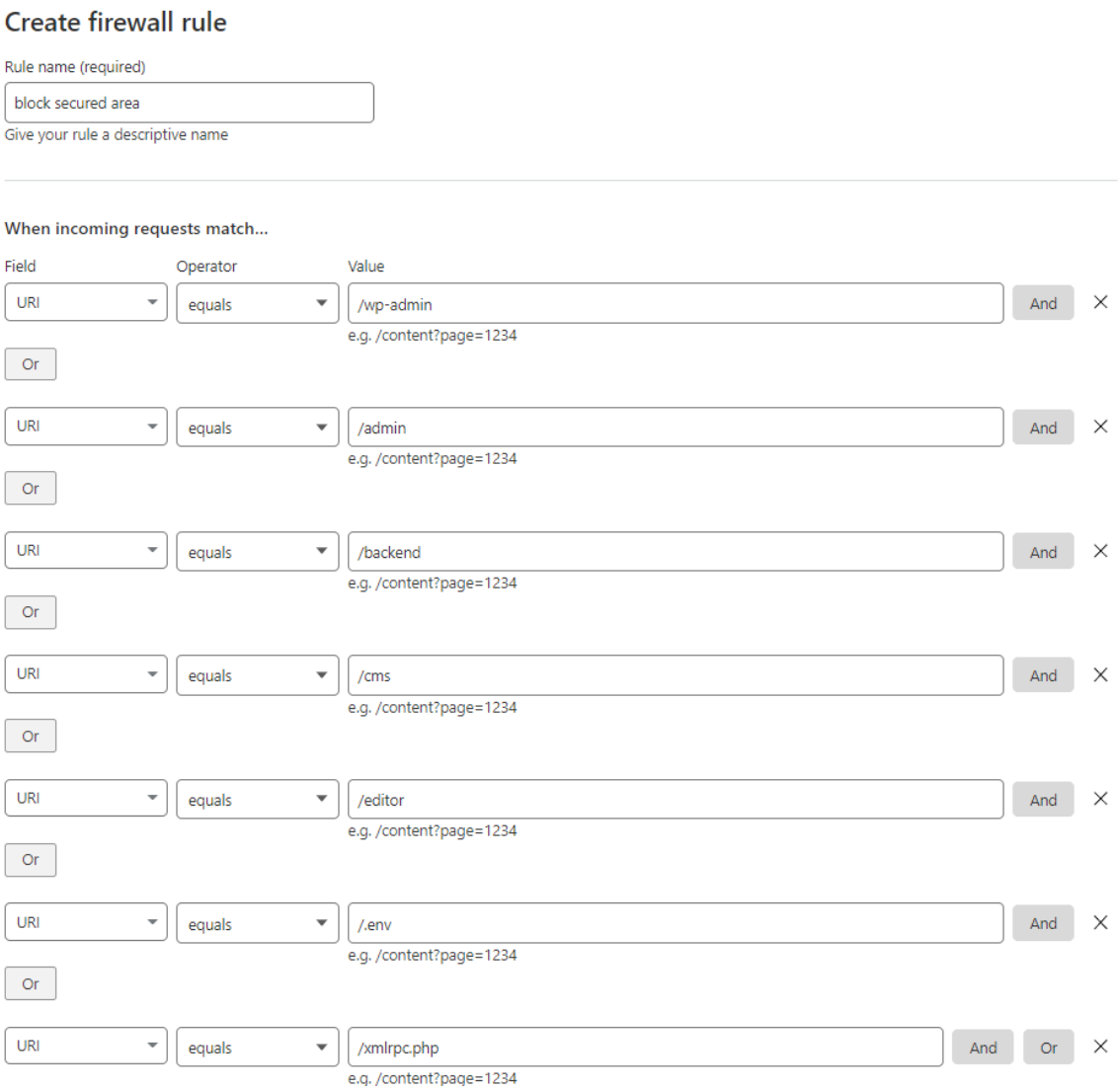


Рис. 3.13 Створення правила для блокування захищеної зони

Таким чином при спробі увійти до захищеного домену буде видана помилка:

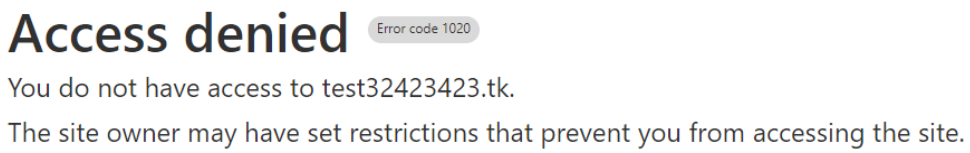


Рис. 3.14 Помилка при спробі входу до захищеного ресурсу

Закінчивши створювати правила, їх можна подивитись на панелі WAF, де можна також побачити статистику спрацювань та редагувати створені правила.

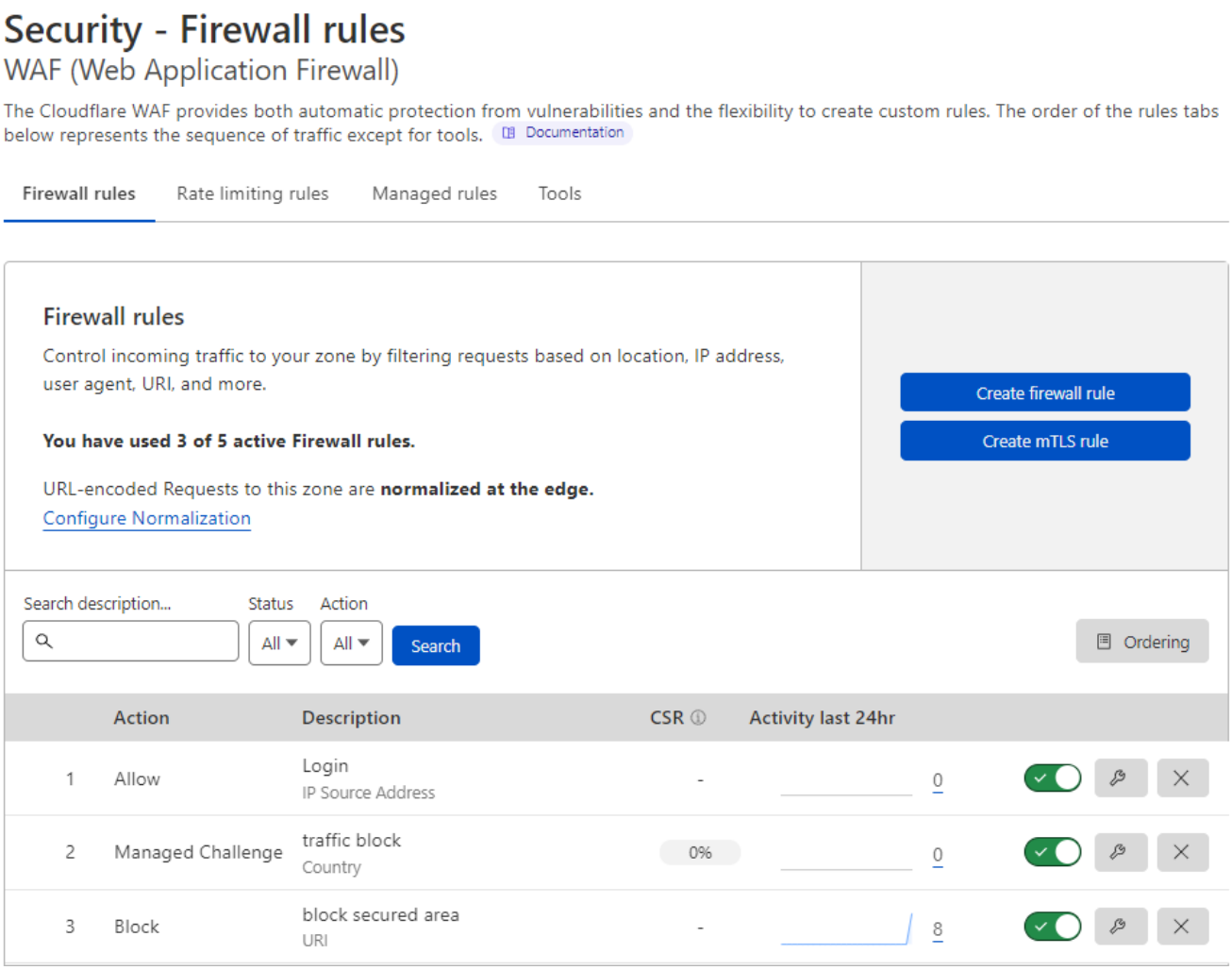


Рис. 3.15 Перелік правил безпеки

Відповідно до правил WAF, Cloudflare надає також правила обмеження запитів для боротьби с ботами та DDoS атаками. Ці правила створюються відповідно до правил Брандмауеру у розділі “WAF” та вкладці “Rate limiting rules”. У безкоштовній версії доступно тільки одне таке правило.

Можна створити правило для блокування перевірених ботів після трьох спроб підключення для уникнення DDoS та Bruteforce атак.

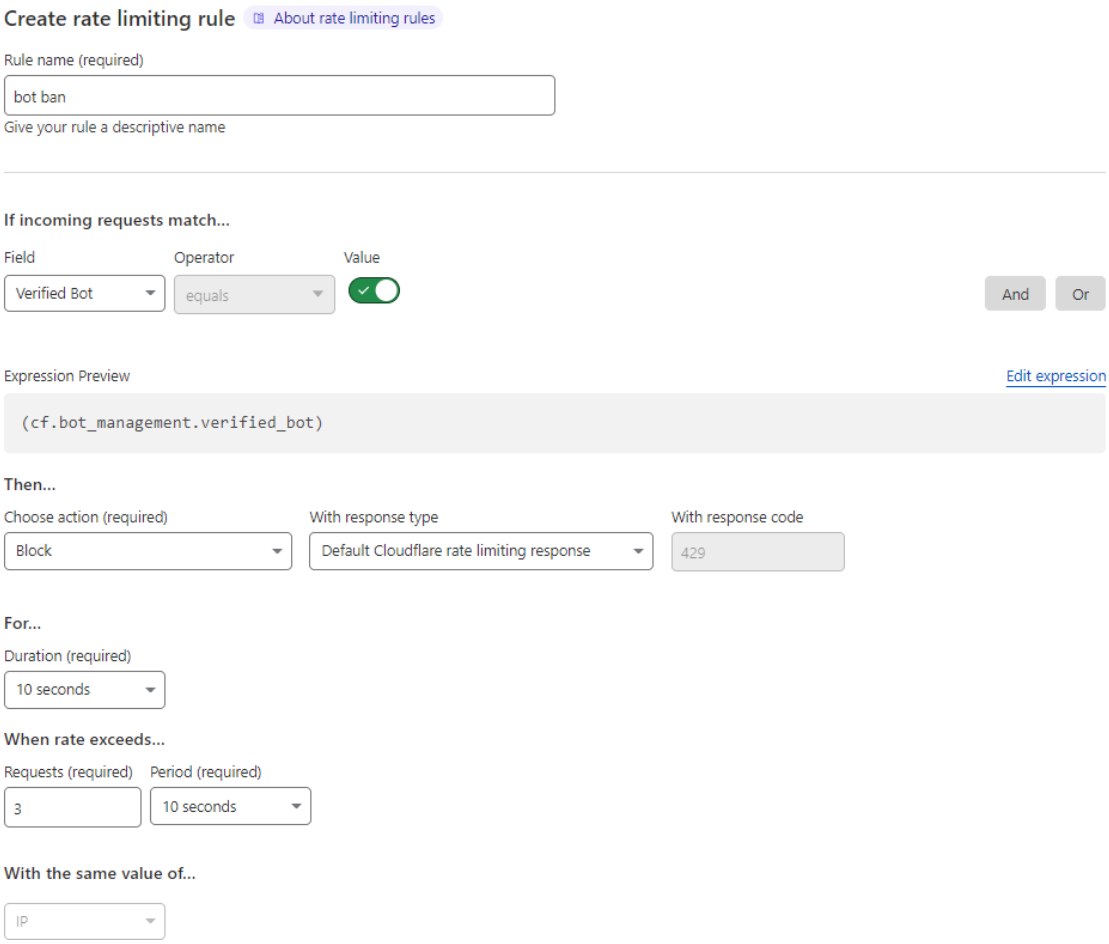


Рис. 3.16 Створення правила обмеження запитів.

Подивитись повну історію спрацювань правил безпеки можна у вкладці “Security” на панелі “Events”

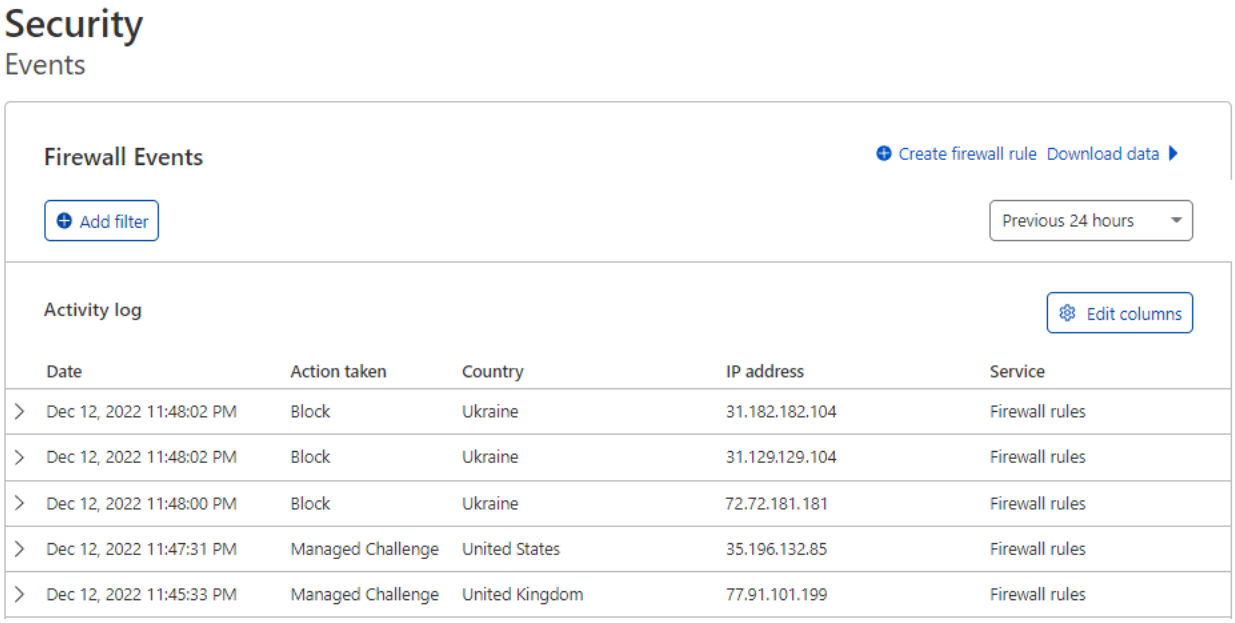
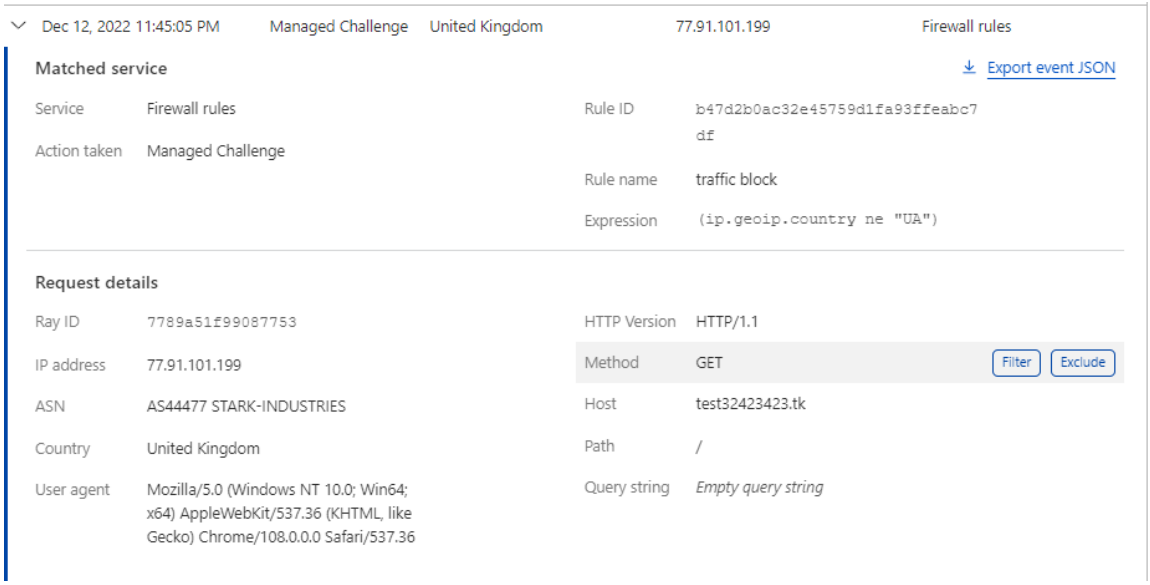


Рис. 3.17 Історія спрацювань правил безпеки

Натиснувши на івент, є можливість перегляду інциденту для отримання детальної інформації та можливого коригування правил безпеки.



3.18 Перегляд інциденту безпеки.

Після налаштування безпеки у Cloudflare, також можна розглянути важливу функцію, яку надає ресурс. Cloudflare має режим який називається “Під час атаки”. При включенні такий режим надає усім користувачам ресурсу графічну, інтерактивну задачу засновану на Javascript. Цей режим є ключовим методом безпеки у разі ботнет або DDoS атак на ресурс. Ввімкнути його можна на головній вкладці “Overview” у розділі “Quick Actions”.

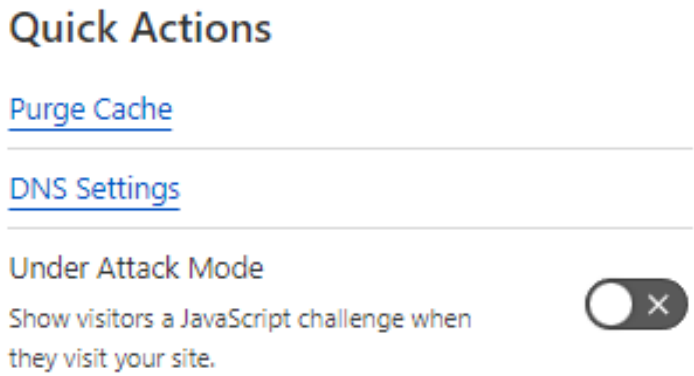


Рис 3.19 Режим захисту “Під час атаки”

3.3 Рекомендації захисту WEB-додатку на основі Cloudflare WAF

Для ефективного використання Cloudflare треба спочатку визначитись з планом підписки. Кожен план надає свої можливості та переваги, але треба враховувати бюджет, який можна витратити на брандмауер. Для персональних проектів та переважно невеликих бізнес-проектів підійде безкоштовна або професійна версія. Безкоштовна версія надає базовий автоматично налаштований набір правил для боротьби від масштабних та самих критичних загроз безпеки та стандартну протекцію від ботів та DDoS атак, а якщо ресурс є більш уразливий до загроз, слід купити про версію для застосування набору правил Cloudflare для захисту від загроз OWASP TOP-10. Для повноцінних бізнес-проектів та більш просунутої команди кіберспеціалістів слід використовувати версію для бізнесу або корпоративну. Бізнес версія надає цілодобову підтримку, значно кращі методи боротьби проти ботів та DDoS атак з використанням штучного інтелекту та 100 кастомних правил брандмауеру для впровадження усіх необхідних політик безпеки. Корпоративна версія містить усе для розгортання захисту в критичній інфраструктурі та багаторівневій системі WEB-ресурсів і потребує команди спеціалістів для підтримання захисту. Такий захист дозволяє виявляти будь яку аномальну поведінку трафіку у WEB-додатку та блокувати до 100 відсотків усіх ботів та загроз безпеки.

Хоча Cloudflare є дуже розумним інструментом, який автоматично налаштовується при впровадженні WEB-додатку, для більш ефективного захисту потрібно зробити декілька кроків налаштувань для забезпечення захищеності ресурсу.

Перш за все треба впровадити найвищий рівень шифрування для захисту інформації при вході та виході даних з системи. Для цього треба включити постійне використання HTTPS протоколу замість незахищеного HTTP, а також поставити повний строгий режим шифрування SSL/TLS. Такий режим дозволяє встановити

HTTPS з’єднання не тільки між користувачем і Cloudflare, а і між Cloudflare та сервером WEB-додатку.

Наступним кроком налаштування є включення режиму боротьби проти ботів, а також встановлення правил обмеження запитів трафіку для уникнення ботів та DDoS атак.

Також, дивлячись на специфіку WEB-додатку та підприємства слід налаштувати кастомні правила Cloudflare. Ефективною практикою при встановленні правил є блокування усього трафіку з країн або континентів, на які не врахований бізнес, а також блокування запитів до будь якої критичної інфраструктури додатку.

Висновки до розділу 3

У даному розділі було досліджено основні функції сервісу Cloudflare, а також проведено інтегрування даного рішення до WEB-додатку та налаштування для забезпечення безпеки.

Був проведений аналіз різних можливостей та функції на основі різних категорії підписок даного рішення, після чого виведено конкретні рекомендації використання кожного рішення для конкретної групи підприємств.

При процесі налаштування було розглянуто основні методи та можливості даного рішення для підтримання та забезпечення безпеки. При налаштуванні були виведені та проведені основні кроки для оптимізації даного рішення і виведення найкращих налаштувань безпеки.

Розроблено рекомендації щодо оптимізації Cloudflare для забезпечення максимальної безпеки WEB-додатку.

ВИСНОВКИ

В ході магістерської роботі був проаналізований процес забезпечення захисту у WEB-додатку за допомогою брандмауеру Cloudflare WAF, на основі чого було розроблено оптимальні налаштування та рекомендації щодо інтеграції рішення у WEB-додатки.

Для досягнення цілі були проведені наступні кроки:

1. Досліджено сучасні WEB-додатки, їх типи та історія розвитку для кращого розуміння теми

2. Наведено основні проблеми безпеки WEB-додатків базовані на статистичних даних.

3. Розглянуто сучасні критичні загрози на основі переліку OWASP – 10 та методи протидії.

4. Досліджено технологію WAF, її розвиток та зроблено порівняння з звичайним брандмауером.

5. Виконано порівняльний аналіз трьох сучасних рішень WAF для кращого розуміння ринку технології та її можливостей.

6. Проведено налаштування та оптимізація рішення Cloudflare WAF.

7. Виведено рекомендації щодо використання Cloudflare WAF для забезпечення максимальної безпеки WEB-додатку.

ПЕРЕЛІК ПОСИЛАНЬ

1. Web application [Електронний ресурс] – Режим доступу до ресурсу: https://www.britannica.com/topic/Web-application.

2. Difference between website and web application [Електронний ресурс] – Режим доступу до ресурсу: https://byjusexamprep.com/difference-between-website-and-web-application-i.

3. History of web application development [Електронний ресурс] – Режим доступу до ресурсу: https://www.devsaran.com/blog/history-web-application-development.

4. Web application architecture [Електронний ресурс] – Режим доступу до ресурсу: https://litslink.com/blog/web-application-architecture.

5. Sitelock Website Security report [Електронний ресурс] – Режим доступу до ресурсу: https://s3.us-east-1.amazonaws.com/sectigo-sites-web/global/uploads/2022-SiteLock-Website-Security-Report-FINAL.pdf.

6. What is Web Application Security? [Електронний ресурс] – Режим доступу до ресурсу: https://digital.ai/glossary/what-is-web-application-security/.

7. OWASP Top 10 Vulnerabilities [Електронний ресурс] – Режим доступу до ресурсу: https://snyk.io/learn/owasp-top-10-vulnerabilities/.

8. OWASP Top 10 - 2021 [Електронний ресурс] – Режим доступу до ресурсу: https://owasp.org/Top10/.

9. What Is a Firewall and Why Do You Need One? [Електронний ресурс] – Режим доступу до ресурсу: https://www.avast.com/c-what-is-a-firewall.

10. WAF vs. Firewall: Web Application & Network Firewalls [Електронний ресурс] – Режим доступу до ресурсу: https://www.fortinet.com/resources/cyberglossary/waf-vs-firewall.

11. What is a Web Application Firewall (WAF)? [Електронний ресурс] – Режим доступу до ресурсу: https://www.f5.com/glossary/web-application-firewall-waf.

12. Signature-based and Machine-Learning-based Web Application Firewalls: A Short Survey [Електронний ресурс] – Режим доступу до ресурсу: https://www.sciencedirect.com/science/article/pii/S1877050921012308.

13. 3 Types of WAF: Cloud, Hardware and Software [Електронний ресурс] – Режим доступу до ресурсу: https://www.radware.com/cyberpedia/application-security/3-types-of-waf/.

14. How to Improve Cloud Security With a Web Application Firewall (WAF)? [Електронний ресурс] – Режим доступу до ресурсу: https://www.netguru.com/blog/cloud-security-waf.

15. What Is WAF [Електронний ресурс] – Режим доступу до ресурсу: https://www.imperva.com/learn/application-security/what-is-web-application-firewall-waf/.

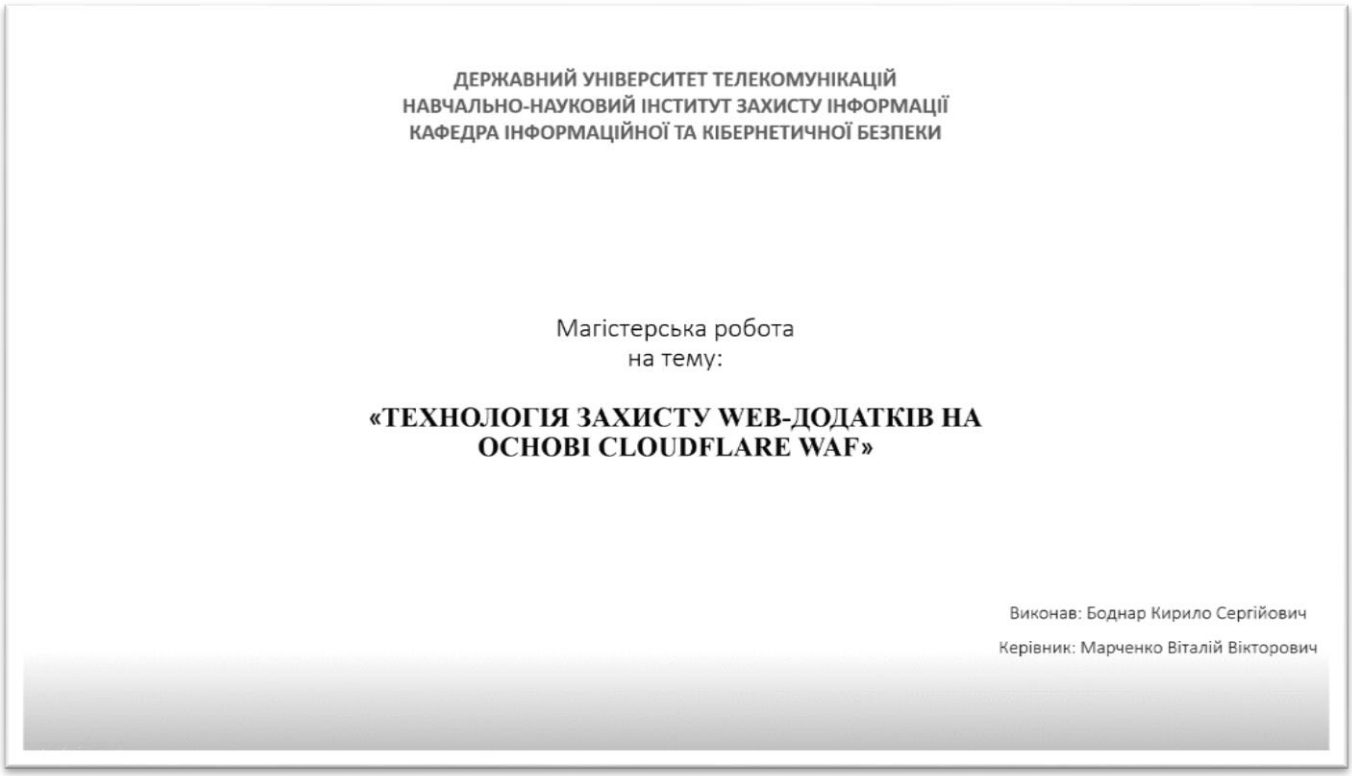
16. BEST Web Application Firewalls (WAF) Vendors In 2022 [Електронний ресурс] – Режим доступу до ресурсу: https://www.softwaretestinghelp.com/web-application-firewall-waf/.

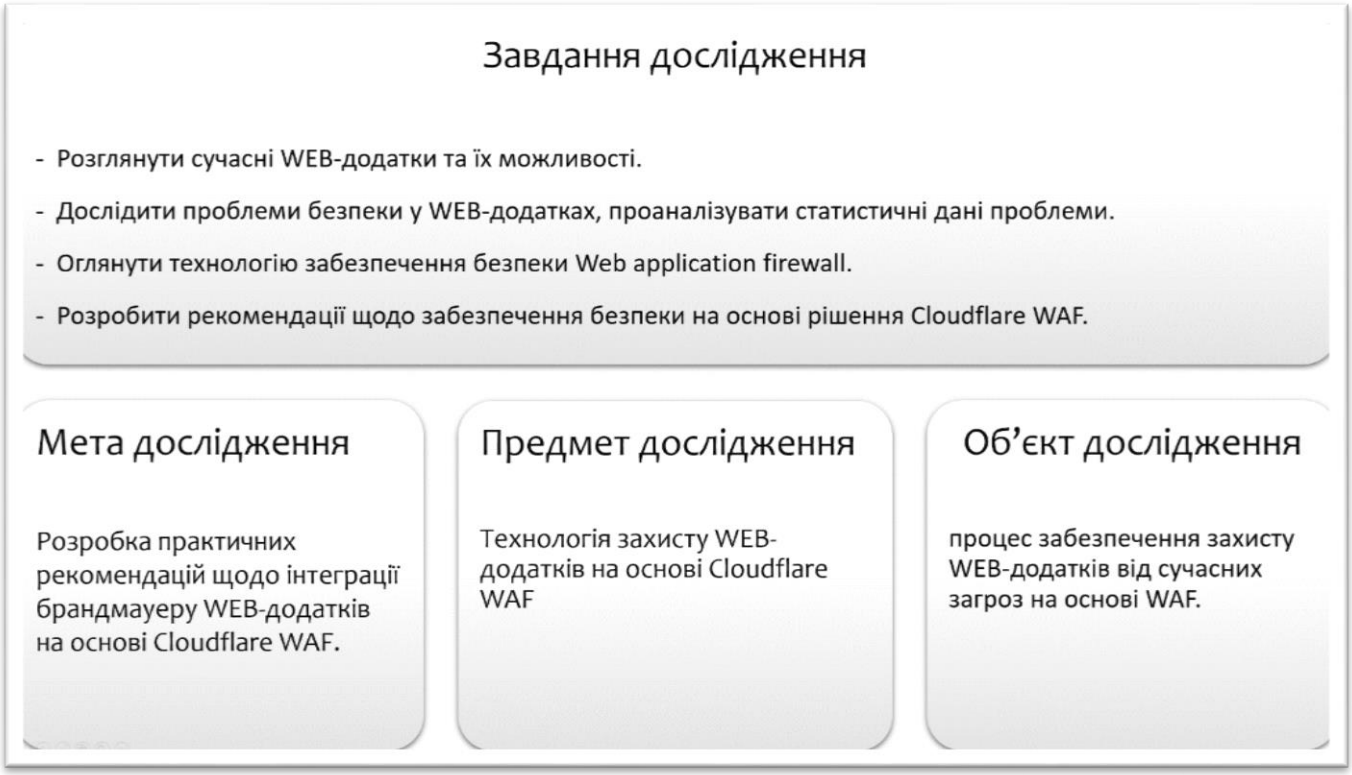
17. A WAF for Modern Application Security [Електронний ресурс] – Режим доступу до ресурсу: https://cf-assets.www.cloudflare.com/slt3lc6tev37/3bmcdEbQ3WQBBMG7Ui8xuY/c1a1e2c41b6e3ce640542afa7d9dcba7/Datasheet_A-WAF-for-Modern-Application-Security.pdf.

18. Cloudflare bot solutions [Електронний ресурс] – Режим доступу до ресурсу: https://developers.cloudflare.com/bots/.

19. Managed Rulesets [Електронний ресурс] – Режим доступу до ресурсу: https://developers.cloudflare.com/waf/managed-rulesets/.

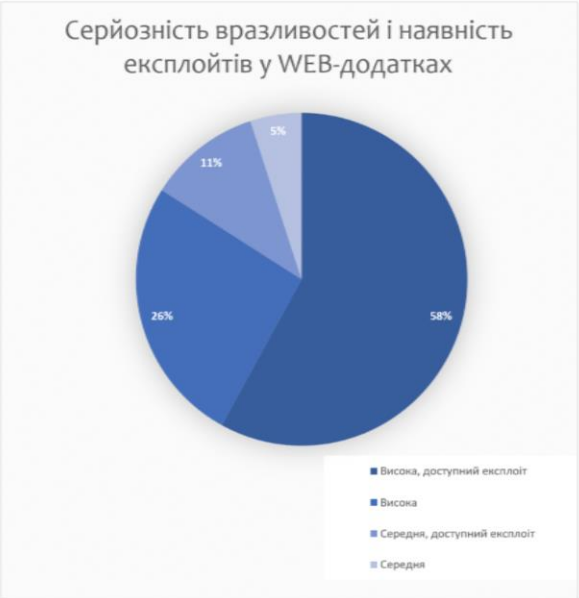
ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ (ПРЕЗЕНТАЦІЯ)





Проблематика безпеки у WEB-додатках

- Обсяг загроз подвоївся у 2021 році порівняно з 2020 роком. Така ж тенденція зараз і у 2022 році
- Основним джерелом збільшення обсягу атак є шкідливі боти. WEB-ресурси отримують в 5,5 разів більше трафіку від ботів, ніж від людей, причому понад 60% цих ботів є зловмисними.
- Шкідливе програмне забезпечення залишається значною проблемою для малого та середнього бізнесу. 18% WEB-додатків заражені загрозами критичного рівня.
- WEB-додатки на основі популярних платформ управління контентом WEB-ресурсів (CMS), таких як WordPress і Drupal експлуатується із загрозливою швидкістю.
- Пошукові системи часто пропускають зараження зловмисним програмним забезпеченням, у 92% випадків заражені зловмисним програмним забезпеченням WEB-додатки не позначаються пошуковою системою небезпечними та не заносяться в чорний список.



3

Технологія Web Application Firewall

WAF захищає веб-додаток методом фільтрації та блокування зловмисного HTTP трафіку. Такий захист працює на 7 рівні моделі OSI, коли звичайний брандмауер працює на 3 та 4 рівнях моделі OSI.

Моделі безпеки

- Негативна модель безпеки
- Позитивна модель безпеки
- Гібридна модель безпеки

Способи реалізації

- Хмарний
- Мережевий (Апаратний)
- Програмний

4

Основні функції рішення Cloudflare

- Рівневий захист за допомогою наборів правил WAF
- Кастомні правила з легким налаштуванням
- WAF ML.
- Оновлені правила для захисту від атак нульового дня.
- Унікальний набір правил Cloudflare для основних CMS та електронної комерції.
- Просунуте обмеження кількості запитів.
- База даних репутації IP адресів.
- Запобігання втраті даних.
- Перевірка незахищених облікових даних.
- SSL/TLS шифрування.
- Захист від DDoS атак.

5

Плани підписки Cloudflare

- Безкоштовний

Для Персональних проектів або ресурсів.

- Про

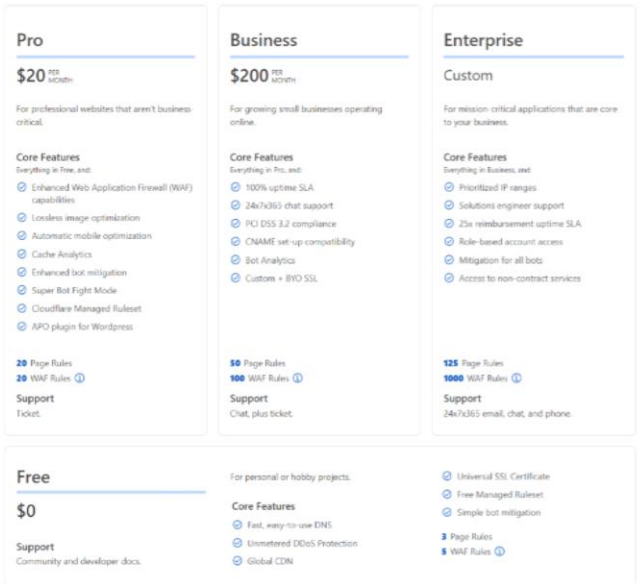
Для професійних WEB-додатків в яких нема критичної інфраструктури.

- Бізнес

Для маленьких та середніх бізнесів, які повноцінно працюють онлайн.

- Корпораційний

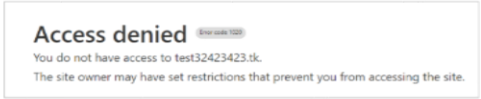
Для WEB-додатків та бізнесів, які вимагають найвищий рівень безпеки, продуктивність та потребують постійної підтримки від команди Cloudflare.



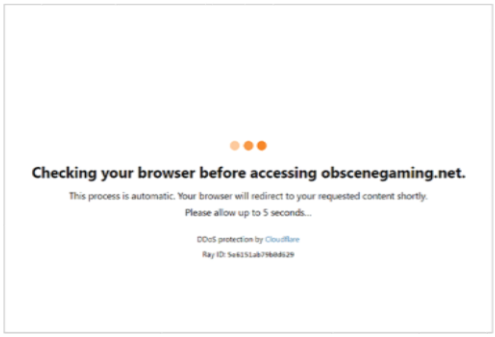
6

Налаштування та робота з рішенням Cloudflare

- Підключення DNS.
- Постійне використання HTTPS.
- Повне (строге) використання SSL/TLS.
- Режим боротьби проти ботів.
- Встановлення наборів правил від Cloudflare.
  - Cloudflare managed ruleset
  - Owasp Top 10
  - Credential check
  - Customized rules for CMS
- Створення нових правил безпеки.



Блокування трафіку



Режим “Під час атаки”

7

Рекомендації

- Визначення доцільного плану підписки Cloudflare базуючись на конкретному випадку.
- Забезпечення захищеного шифрування критичних даних
- Посилення захисту від ботів.
- Використання наборів правил для ефективної роботи WAF
- Моніторинг небезпек для своєчасного виявлення критичної ситуації.
- Оптимальне налаштування рішення Cloudflare за допомогою правил та інших практик безпеки згідно з ситуацією та важливістю даних.

8

Висновки

1. Досліджено сучасні WEB-додатки, їх типи та історія розвитку для кращого розуміння теми
2. Наведено основні проблеми безпеки WEB-додатків базовані на статистичних даних.
3. Розглянуто сучасні критичні загрози на основі переліку OWASP – 10 та методи протидії.
4. Досліджено технологію WAF, її розвиток та зроблено порівняння з звичайним брандмауером.
5. Виконано порівняльний аналіз трьох сучасних рішень WAF для кращого розуміння ринку технології та її можливостей.
6. Проведено налаштування та оптимізація рішення Cloudflare WAF.
7. Виведено рекомендації щодо використання Cloudflare WAF для забезпечення максимальної безпеки WEB-додатку.

9

Дякую за увагу!

10