ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА КОМП'ЮТЕРНОЇ ІНЖЕНЕРІЇ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «Хмарні сервіси для кращого захисту серверів»

на здобуття освітнього ступеня бакалавра
зі спеціальності 123 – Комп'ютерна інженерія
(код, найменування спеціальності)
освітньо-професійної програми Комп'ютерна інженерія
(назва)

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

_____	*Анастасія ЄМЧЕНКО*
(підпис)	*Ім'я, ПРІЗВИЩЕ здобувача*

Виконала: здобувачка вищої освіти гр. КІД-41
Анастасія ЄМЧЕНКО
Ім'я, ПРІЗВИЩЕ
Керівник: Ігор БУЧЕНКО
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ
Рецензент: _____
науковий ступінь, вчене звання Ім'я, ПРІЗВИЩЕ

Київ 2025

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій

Кафедра Комп'ютерної інженерії

Ступінь вищої освіти бакалавр

Спеціальність 123 – Комп'ютерна інженерія

Освітньо-професійна програма Комп'ютерна інженерія

ЗАТВЕРДЖУЮ
Завідувач кафедрою Наталія ЛАЩЕВСЬКА
_____Ім'я, ПРІЗВИЩЕ
«_____» _____ 2025р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Ємченко Анастасія Володимирівна
(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Хмарні сервіси для кращого захисту серверів керівник кваліфікаційної роботи Ігор БУЧЕНКО,
(Ім'я, ПРІЗВИЩЕ, науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «27» лютого 2025р. № 36
2. Строк подання кваліфікаційної роботи «5» червня 2025р.
3. Вихідні дані до кваліфікаційної роботи: науково-технічна література; хмарні сервіси: Cloudflare та AWS Shield; види атак на сервери та принцип їх роботи
4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)
   1. Загрози для серверів у хмарному середовищі
   2. Огляд хмарних сервісів захисту
   3. Практична реалізація захисту сервера з використанням Cloudflare
5. Перелік ілюстративного матеріалу: *презентація*
6. Дата видачі завдання «27» лютого 2025р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Збір даних	01.03-22.03.2025	Виконано
2	Обробка матеріалу	22.03-07.04.2025	Виконано
3	Розробка теоретичної частини	09.04-12.04.2025	Виконано
4	Розробка практичної частини	14.04-20.04.2025	Виконано
5	Висновки за результатами аналізу	21.04-06.05.2025	Виконано
6	Оформлення пояснювальної записки	08.05-10.05.2025	Виконано
7	Розробка презентації	20.05-25.05.2025	Виконано
8	Передзахист	27.05-29.05.2025	Виконано
9	Здача в деканат	02.06-05.06.2025	Виконано

Здобувачка вищої освіти _____ Анастасія ЄМЧЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник кваліфікаційної роботи _____ Ігор БУЧЕНКО
(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня бакалавра: 65 стор., 23 рис., 2 табл., 30 джерел.

Мета роботи – проведення порівняльного аналізу хмарних сервісів Cloudflare та AWS Shield за низкою технічних, функціональних та організаційних критеріїв з метою обґрунтування вибору найбільш доцільного рішення для захисту вебресурсу.

Об'єкт дослідження – дослідження хмарних технологій захисту вебресурсів.

Предмет дослідження – засоби забезпечення безпеки вебінфраструктури з використанням сучасних хмарних сервісів

Короткий зміст роботи: Робота присвячена дослідженню та порівняльному аналізу хмарних сервісів безпеки Cloudflare та AWS Shield для захисту вебресурсів від сучасних кіберзагроз. Визначено актуальність використання таких сервісів та сформульовано критерії їх оцінки. Проведено порівняння Cloudflare та AWS Shield за цими критеріями, а також здійснено практичне дослідження застосування Cloudflare, оцінено його зручність, ефективність базового захисту та можливості масштабування безпеки. Надано висновки щодо доцільності використання Cloudflare та його порівняння з AWS Shield для оптимального вибору рішення.

КЛЮЧОВІ СЛОВА: ХМАРНІ СЕРВІСИ, КІБЕРАТАКА, СЕРВЕР, DENIAL OF SERVICE (DDOS), CLOUDFLARE, AWS SHIELD, FIREWALL.

ЗМІСТ

ВСТУП…… 8
1 ЗАГРОЗИ ДЛЯ СЕРВЕРІВ У ХМАРНОМУ СЕРЕДОВИЩІ…… 10
1.1 Сервери та їх класифікація …… 10
1.2 Основні види атак на сервери …… 15
1.3 Методи захисту серверів…… 23
1.4 Значення хмарних сервісів у забезпеченні безпеки…… 39
2 ОГЛЯД ХМАРНИХ СЕРВІСІВ ЗАХИСТУ…… 42
2.1 Архітектура та принципи роботи Cloudflare…… 42
2.2 AWS Shield: Огляд типів, інтеграції та рівнів захисту…… 51
2.3 Порівняння Cloudflare та AWS Shield…… 55
2.4 Вибір оптимального сервісу на основі аналізу…… 59
3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ЗАХИСТУ СЕРВЕРА З ВИКОРИСТАННЯМ CLOUDFLARE…… 61
3.1 Розгортання тестового вебресурсу…… 61
3.2 Підключення вебресурсу до Cloudflare та налаштування захисту... 63
3.3 Імітація загроз та аналіз захисту…… 67
3.4 Оцінка результатів…… 71
ВИСНОВКИ…… 73
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ…… 74
Додаток А. HTML-КОД ДЛЯ ДЕМОНСТРАЦІЙНОЇ ВЕБСТОРІНКИ…… 77
Додаток Б. ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ…… 78

ВСТУП

Стрімкий розвиток цифрових технологій, активна діджиталізація бізнесу та постійне зростання обсягів вебтрафіку обумовили необхідність ефективного захисту вебресурсів. Сучасні інформаційні системи є надзвичайно вразливими до кіберзагроз, серед яких особливу небезпеку становлять атаки типу Distributed Denial of Service (DDoS), автоматизовані сканування, зловмисні боти та інші види зовнішнього втручання. У зв'язку з цим забезпечення доступності, безперервності та захищеності вебресурсів набуває пріоритетного значення на етапі проєктування та розгортання інформаційних систем.

Одним із напрямів ефективного вирішення зазначених проблем є використання хмарних сервісів безпеки, які надають можливість централізовано та автоматизовано реалізувати захист на рівні мережі, застосунку та інфраструктури. Такі сервіси дозволяють не лише виявляти й блокувати шкідливу активність, а й забезпечувати балансування навантаження, оптимізацію трафіку та прискорення доступу до контенту. До найбільш поширених рішень у цій сфері належать Cloudflare та AWS Shield, які суттєво відрізняються за архітектурною побудовою, функціональними можливостями, умовами інтеграції та ціноутворенням.

Актуальність теми кваліфікаційної роботи обумовлена необхідністю здійснення комплексної порівняльної оцінки таких сервісів для вибору оптимального рішення відповідно до умов конкретного застосування. Адже хибний вибір технологічного засобу безпеки може призвести до перевитрати ресурсів, неефективного захисту або надлишкової складності адміністрування.

У рамках дослідження проаналізовано сучасні загрози безпеці вебзастосунків і мережевої інфраструктури, виявлено актуальні вимоги до систем захисту у хмарному середовищі, сформульовано систему критеріїв оцінки хмарних сервісів, а також здійснено систематизоване порівняння Cloudflare та AWS Shield за зазначеними параметрами.

На основі результатів аналізу було визначено, яке з рішень є більш придатним для впровадження в межах даного проєкту з урахуванням доступності, масштабованості, функціональної повноти, вартості впровадження та зручності експлуатації.

Практична цінність роботи полягає в можливості застосування отриманих результатів для реального проєктування системи захисту інформаційних ресурсів з опорою на доступні комерційні рішення без потреби у власній фізичній інфраструктурі.

1 ЗАГРОЗИ ДЛЯ СЕРВЕРІВ У ХМАРНОМУ СЕРЕДОВИЩІ

1.1 Сервери та їх класифікація

Будь-яка IT-інфраструктура складається із серверів, подібно до того, як цеглини утворюють будинок. Сервери мають різні конфігурації та служать різним цілям. Наприклад, деякі перетворюють запити на шляхи до певних IP-адрес, тоді як інші зберігають дані. Ролі сервера настільки різні, що спочатку незрозуміло, чому ці компоненти мають однакові назви.

Сервер визначається як високопродуктивна обчислювальна платформа, оптимізована для виконання завдань з обробки, зберігання та адміністрування даних, а також керування мережевими пристроями та системами. Ця комплексна комп'ютерна архітектура забезпечує надання різноманітних ресурсів мережевим елементам, зокрема для відображення вебсторінок та здійснення обміну електронною кореспонденцією. В якості сервера може виступати спеціалізоване апаратне забезпечення, відповідне програмне забезпечення або віртуалізоване середовище, що володіє необхідними функціональними можливостями. Проте, функціональність сервера значно перевищує можливості типового персонального комп'ютера, дозволяючи йому ефективно обробляти множинні одночасні запити користувачів, забезпечувати хостинг вебсайтів зі значним інформаційним наповненням, конфігурувати спільні дискові ресурси для мережевих пристроїв та виконувати інтенсивні обчислювальні завдання, такі як управління транзакціями баз даних, що вимагає значної обчислювальної потужності. Комунікація в мережі здійснюється за моделлю "клієнт-сервер", де сервер (хост-пристрій) виступає постачальником ресурсів для клієнтських пристроїв [1].

Серверні комп'ютери є складними системами, що включають чотири основні компоненти, які забезпечують їхню здатність виконувати код, обмінюватися даними та зберігати їх протягом різних періодів часу. Ці компоненти – процесор,

оперативна пам'ять, сховище та пропускна здатність – хоча й присутні в багатьох сучасних електронних пристроях, у серверних машинах характеризуються значно вищою продуктивністю та надійністю.

Центральний процесор (CPU) є "серцем" будь-якого сервера, відповідаючи за виконання обчислень. Сучасні сервери оснащуються багатоядерними процесорами, здатними паралельно обробляти тисячі інструкцій.

Оперативна пам'ять (RAM) є енергозалежним блоком пам'яті, що використовується для тимчасового зберігання активних даних та машинного коду, забезпечуючи швидкий доступ до них для процесора. Підсистема зберігання, представлена традиційними жорсткими дисками (HDD) або твердотільними накопичувачами (SSD), використовується для довготривалого зберігання операційної системи, програмних бібліотек, баз даних, файлів вебсайтів та іншої необхідної інформації.

Пропускна здатність мережевого інтерфейсу визначає обсяг даних, які сервер може передавати та отримувати за одиницю часу, що є критично важливим для ефективного обслуговування численних користувачів та їхніх запитів. Для підвищення відмовостійкості багато серверів використовують технологію RAID, що об'єднує кілька фізичних дисків в єдину логічну одиницю з метою резервування даних та підвищення продуктивності.

Робота сервера ґрунтується на моделі "клієнт-сервер" (рис. 1.1). Коли користувач вводить URL-адресу вебсайту в браузері, цей запит проходить кілька етапів. Браузер спочатку ідентифікує протокол (HTTP або HTTPS), який визначає спосіб передачі даних між браузером та сервером. Потім визначається ім'я сервера, яке за допомогою системи доменних імен (DNS) перетворюється на унікальну IP-адресу сервера, де розміщено запитуваний вебсайт. Нарешті, визначається назва конкретного файлу або ресурсу (HTML, CSS, зображення, скрипти тощо), який браузер запитує у сервера для відображення вебсторінки.

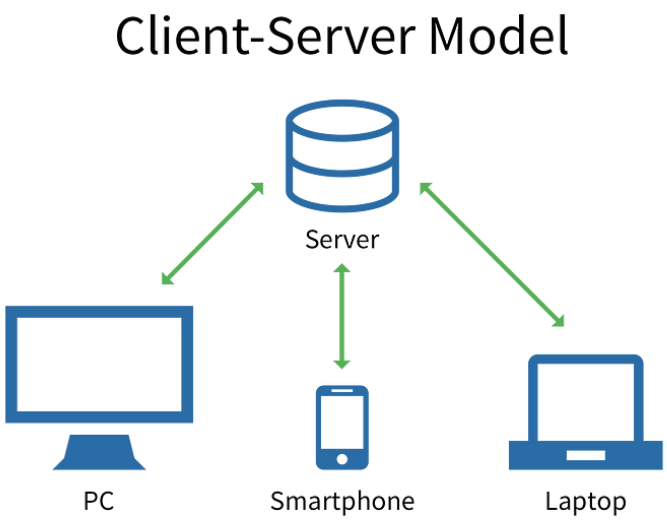


Рисунок 1.1 − Принцип роботи моделі "клієнт-сервер"[2]

Етапи роботи сервера:

1) браузер надсилає запит на доменне ім’я серверу DNS;

2) DNS-сервер визначає IP-адресу, пов’язану із запитуваним доменом;

3) після отримання IP-адреси браузер пересилає запит користувача на цільовий сервер;

4) сервер збирає дані сайту та динамічні елементи, необхідні для відображення вебсайту з боку користувача, і надсилає їх назад користувачеві у відповідь на його запит.

Типи серверів. Сьогодні використовуються сервери різних типів, кожен з яких оптимізований для певних завдань. Багато серверів є універсальними і можуть виконувати кілька функцій одночасно, таких як розміщення вебсайтів, обробка електронної кореспонденції та забезпечення безпеки внутрішніх мереж. На противагу їм існують виділені сервери, які спеціалізуються на одній конкретній функції, наприклад, керування базами даних або обробка завдань друку. Отже, основні типи серверів включають в себе:

– *вебсервери.* Вони є ключовими елементами для функціонування вебсайтів та онлайн-програм, забезпечуючи їхню роботу в браузерах, таких як Chrome або Firefox. Їхнє завдання полягає у зберіганні, обробці та доставці вмісту вебсторінок користувачам. Для забезпечення ефективного обміну інформацією між

мережевими пристроями вебсервери використовують такі протоколи, як HTTP, FTP та SMTP. У мережі кожна машина, на якій розміщено вебсайт, повинна мати встановлене програмне забезпечення вебсервера для оптимізованого керування доступом користувачів до розміщених файлів. Серед відомих прикладів вебсерверів можна назвати Apache HTTP Server, Microsoft Internet Information Services (IIS) та Sun Java System Web Server від Sun Microsystem;

– *поштові сервери.* Вони відповідають за зберігання та організацію електронної пошти для користувачів. Вони використовують різні протоколи для надсилання (наприклад, SMTP) та отримання (наприклад, POP3) імейлів. Подібно до поштового відділення, сервер зберігає та сортує листи перед їхньою доставкою. Коли користувач відправляє імейл, він фактично просить поштовий сервер доставити його адресату. Для цього поштові сервери використовують спеціалізовані програмні компоненти, що забезпечують надсилання, отримання та пересилання електронних листів;

– *сервери застосунків.* Забезпечують інфраструктуру для розгортання, виконання та керування вебдодатками, незалежно від їхнього призначення. Вони обробляють програмний код (наприклад, PHP, Java, .Net), необхідний для їхньої роботи. Подібно до вебсерверів, сервери застосунків використовують протокол HTTP для комунікації з клієнтами та відповідають на їхні запити. Однак, на відміну від вебсерверів, які часто використовуються для відображення статичного контенту, сервери застосунків орієнтовані на генерацію динамічних вебсайтів (таких як онлайн-магазини або блоги), використовуючи для цього пули об'єктів, служби обміну повідомленнями та інші механізми. Варто зазначити, що в деяких випадках сервер застосунків може містити в собі компонент вебсервера, що дозволяє йому виконувати і його функції;

– *сервери бази даних.* Основна функція серверів баз даних полягає в наданні послуг управління базами даних для клієнтських машин. Користувачі можуть взаємодіяти з даними, виконуючи запити, такі як SQL-запити, для доступу, редагування, збереження та отримання інформації з бази даних. Ці сервери відіграють ключову роль у забезпеченні безпеки та можливості відновлення систем

керування базами даних (СУБД). Крім того, вони використовуються для зберігання різних цифрових файлів, включаючи програмні та не пов'язані безпосередньо з базами даних. Великі організації покладаються на сервери баз даних для захисту своєї конфіденційної та цінної корпоративної інформації. До поширених прикладів належать Microsoft SQL, MySQL, MongoDB та SQLite;

– *DNS-сервери.* Це сервери доменних імен. Вони розпізнають імена серверів, які знаходяться в мережі. DNS-сервери є невіддільною частиною Інтернету, оскільки вони перекладають зрозумілі користувачам URL-адреси (наприклад, www.netforce.company) на відповідні IP-адреси;

– *проксі-сервери.* Вони виконують роль посередника в комунікації між локальними мережами та зовнішнім світом. Вони слугують точкою взаємодії, приймаючи запити від внутрішньої мережі та перенаправляючи їх, використовуючи власну IP-адресу. Однією з важливих функцій проксі-серверів є кешування часто запитуваних вебсторінок, що дозволяє значно прискорити їхнє повторне завантаження та оптимізувати використання мережевої пропускної здатності. Крім того, проксі-сервери здійснюють фільтрацію мережевого трафіку, забезпечують постійну доступність, що є важливим для розподілу мережевого навантаження, а також підвищують конфіденційність користувачів, замінюючи їхні оригінальні IP-адреси на власні;

– *сервери DHCP (протоколу динамічної конфігурації хоста).* Автоматично розподіляють IP-адреси між клієнтськими пристроями в мережі. Такий підхід мінімізує ризик помилок, які можуть виникнути при ручному налаштуванні, зокрема помилок введення або конфліктів IP-адрес. DHCP-сервери також забезпечують гнучке керування конфігурацією мережевих пристроїв, дозволяючи оновлювати застарілі IP-адреси та ефективно адмініструвати IP-адреси пристроїв, що змінюють своє розташування. Завдяки здатності одночасно обробляти численні запити від клієнтів, DHCP-сервери відіграють ключову роль в ефективному адмініструванні мережі;

– *файлові сервери.* Вони функціонують як централізовані сховища даних, надаючи спільний доступ до дисків для комп'ютерів у мережі. На цих дисках

зберігаються різноманітні типи файлів, такі як текстові документи, графічні зображення, аудіо- та відеоматеріали. Основна роль файлових серверів полягає у забезпеченні зберігання даних, а не у виконанні програмного коду чи обчислювальних операцій. Залежно від мережевого середовища, для доступу до файлів використовуються різні протоколи: в Інтернеті – FTP та HTTP, а в локальних мережах (наприклад, в офісах або навчальних закладах) – SMB або NFS;

– *ігрові сервери.* Ці сервери основою для онлайн-геймінгу, об'єднуючи гравців у таких багатокористувацьких іграх, як Counter-Strike та Minecraft. Вони керують даними гравців та ігровим процесом, синхронізуючи дії у віртуальному світі. Популярні хостинг-провайдери ігрових серверів, такі як ScalaCube, HostHavoc та Cloudzy, надають в оренду сервери для ігор на кшталт ARK та Rust;

– *сервери друку.* Вони виконують важливу функцію з'єднання принтерів та схожих пристроїв з користувачами в мережі. Вони приймають відправлені на друк завдання та ставлять їх у чергу для послідовного виконання, оскільки принтери часто не можуть обробити велику кількість запитів одночасно. Для взаємодії з принтерами та клієнтами ці сервери використовують різноманітні протоколи, включаючи протокол Інтернет-друку (IPP), протокол демона лінійного принтера (LPD) та JetDirect. На ринку серверів друку відомі такі бренди, як IOGEAR та HP.

1.2 Основні види атак на сервери

У сучасному світі інформаційні технології стали невід’ємною частиною всіх сфер життя. Сервери відіграють ключову роль у функціонуванні цифрової інфраструктури. Вони забезпечують збереження та обробку даних, підтримку вебсайтів, електронної пошти, хмарних сервісів, фінансових операцій та багатьох інших критично важливих функцій. Саме через свою значущість, сервери часто стають об’єктом атак з боку зловмисників.

Кібератаки на сервери можуть мати різні цілі: отримання несанкціонованого

доступу, викрадення конфіденційної інформації, порушення нормальної роботи систем або нанесення фінансової шкоди компаніям та користувачам. Види атак значно відрізняються за своєю природою: від примітивних спроб злому паролів до складних багатоетапних атак, що використовують соціальну інженерію, вразливості програмного забезпечення та ботнети.

Однією з найбільш поширених загроз є DDoS-атаки (Distributed Denial of Service), які спрямовані на перевантаження серверів, що призводить до їх недоступності для легітимних користувачів. Також значну небезпеку становлять SQL-ін’єкції, атаки типу «людина посередині» (MITM), експлойти вразливостей, атаки на протоколи зв’язку та злам облікових даних.

З огляду на зростаючу кількість загроз і розвиток технологій кіберзлочинців, забезпечення безпеки серверів є важливим пріоритетом для компаній та організацій. Ефективний захист включає використання сучасних засобів безпеки, таких як брандмауери, системи виявлення вторгнень (IDS/IPS), шифрування даних, регулярне оновлення програмного забезпечення та навчання персоналу [3].

DoS та DDoS-атаки. Перш за все, варто розібратися, що таке DoS та DDoS атаки, а також у чому полягає їхня різниця. Незважаючи на схожість абревіатур, насправді різниця між ними значно суттєвіша, ніж може здатися на перший погляд. Основна відмінність між DoS та DDoS атаками полягає у джерелі запитів (рис 1.2). У DoS атаці використовується одна IP-адреса, тоді як DDoS атака залучає безліч IP-адрес заражених комп'ютерів, об'єднаних у бот-мережу. DoS (Denial-of-Service) є мережевою атакою, метою якої є перевантаження цільового комп'ютера або мережевого обладнання великою кількістю запитів, що призводить до відмови в обслуговуванні та недоступності онлайн-сервісів на певний період часу.

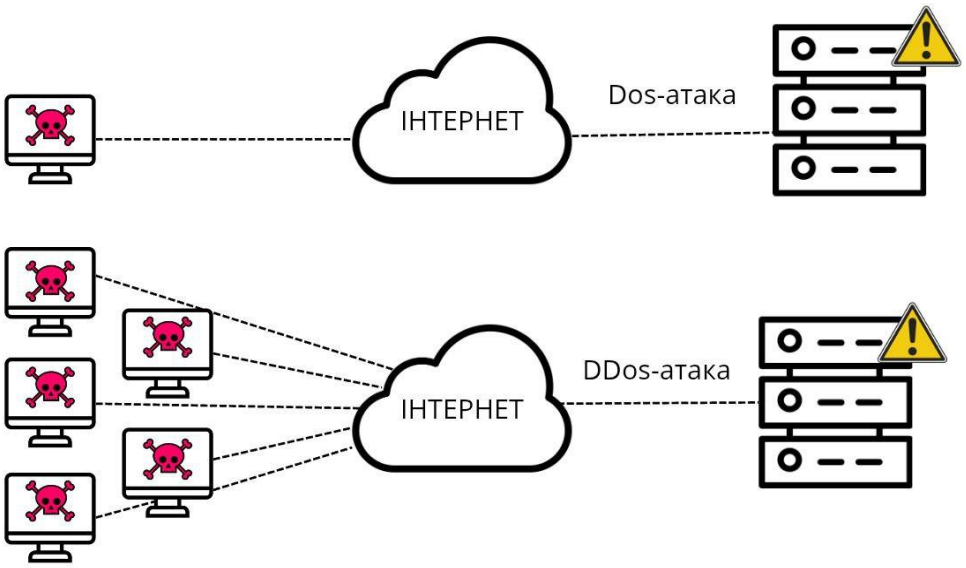


Рисунок 1.2 –Різниця між DoS та DDoS-атакою

Існує два основних типи DoS атак:

– генерація надлишкового трафіку (Overwhelming Quantity of Traffic), що характеризується інтенсивною передачею значного обсягу даних, який перевищує можливості обробки атакованих мереж та хостів. Це призводить до збільшення часу затримки передачі даних та відповіді, відмови в обслуговуванні або нештатного завершення сеансу;

– надсилання пакетів зловмисного формату (Maliciously Formatted Packets), що полягає у передачі великої кількості некоректно сформованих пакетів до цільового хоста або застосунку, що унеможливлює їх належну обробку. Зокрема, атакуючий може навмисно порушувати стандартний алгоритм функціонування сервера, відправляючи пакети, які не піддаються ідентифікації програмним забезпеченням, або передаючи пакети з помилковою структурою, що може спричинити сповільнення продуктивності або відмову приймаючого пристрою.

DDoS-атака (розподілена відмова в обслуговуванні) є аналогом DoS-атаки, але відрізняється тим, що атакуючий трафік надходить з багатьох скоординованих пристроїв. Основний принцип полягає в тому, щоб створити настільки великий потік запитів до цільового ресурсу, щоб його канал зв'язку не міг впоратися з навантаженням. В результаті легітимні запити користувачів не обробляються через надмірну кількість запитів, що призводить до значних затримок або повної

недоступності вебресурсу.

Комп'ютери, інфіковані троянськими програмами та залучені до DDoS-атаки, називаються "зомбі", а їх об'єднана мережа – ботнетом. Типова DDoS-атака розгортається наступним чином: спочатку відбувається пошук слабких місць у мережі, потім ці вузли захоплюються і на них встановлюється шкідливе ПЗ, після чого "комп'ютерам-зомбі" надсилаються команди для атаки на конкретний вебсервер.

Існує безліч видів DDoS-атак, які можна класифікувати за різними критеріями, зокрема за рівнем моделі OSI, на який вони спрямовані, або за методом впливу. Ось основні категорії та деякі поширені приклади:

За рівнем моделі OSI:

а) *атаки на рівні мережі (рівень 3) та транспортному рівні (рівень 4).* Ці атаки спрямовані на перевантаження пропускної здатності мережі або ресурсів сервера великим обсягом трафіку:

1) *UDP-флуд (UDP Flood).* Затоплення сервера великою кількістю UDP-пакетів на випадкові порти. Оскільки UDP не передбачає встановлення з'єднання, сервер витрачає ресурси на спроби обробити кожен пакет і відповісти на неіснуючі адреси;

2) *SYN-флуд (SYN Flood).* Експлуатує процес встановлення TCP-з'єднання ("рукостискання"). Атакуючий надсилає велику кількість SYN-пакетів на сервер, але не завершує встановлення з'єднання, залишаючи безліч напіввідкритих з'єднань і вичерпуючи ресурси сервера;

3) *ICMP-флуд (ICMP Flood) або Ping of Death.* Затоплення цілі великою кількістю ICMP-пакетів (ping-запитів). "Ping of Death" є застарілою атакою, що використовує фрагментовані ICMP-пакети, розмір яких після збірки перевищує максимально допустимий, що може призвести до збою системи;

4) *DNS-ампліфікація (DNS Amplification).* Атакуючий надсилає невеликі запити на публічні DNS-сервери, підміняючи IP-адресу відправника

на IP-адресу жертви. DNS-сервери відповідають значно більшими за розміром пакетами на IP-адресу жертви, створюючи великий обсяг трафіку;

5) *NTP-ампліфікація (NTP Amplification).* Аналогічно DNS-ампліфікації, використовує публічні NTP-сервери для посилення трафіку;

6) *Smurf Attack.* Застаріла атака, що використовує широкомовні ping-запити з підробленою IP-адресою жертви, щоб змусити безліч хостів в мережі відповісти жертві;

7) *Fraggle Attack*. Варіація Smurf-атаки, що використовує UDP-пакети замість ICMP.

б) *атаки на рівні додатків (рівень 7).* Ці атаки спрямовані на вичерпання ресурсів конкретних програм або сервісів на сервері. Вони часто імітують легітимний трафік, що ускладнює їх виявлення та блокування:

1) *HTTP-флуд (HTTP Flood).* Затоплення вебсервера великою кількістю HTTP-запитів (GET або POST) з метою перевантаження його обчислювальних ресурсів;

2) *Slowloris.* Атака, яка намагається вичерпати пул з'єднань вебсервера, надсилаючи часткові HTTP-запити та утримуючи з'єднання відкритими якомога довше;

3) *R-U-Dead-Yet (RUDY)*. Повільна атака типу "POST", яка надсилає HTTP-запити з повільною передачею даних тіла запиту, щоб зайняти з'єднання сервера на тривалий час;

4) *SQL-ін'єкції (SQL Injection) та XSS (Cross-site Scripting) як вектори DDoS.* Хоча це в першу чергу атаки на безпеку вебдодатків, вони можуть бути використані для спричинення відмови в обслуговуванні, наприклад, шляхом виконання ресурсоємних запитів до бази даних або перевантаження браузерів користувачів шкідливим кодом;

5) *атаки на API* спрямовані на перевантаження API вебсервісів великою кількістю запитів.

За механізмом дії:

– *об'ємні атаки (Volumetric Attacks).* Вони націлені на перевантаження пропускної здатності мережі жертви великим обсягом трафіку. Приклади включають UDP-флуд, ICMP-флуд та ампліфікаційні атаки (DNS, NTP);

– *протокольні атаки (Protocol Attacks).* Експлуатують вразливості в протоколах мережевого рівня (рівні 3 та 4) для вичерпання ресурсів сервера або мережевого обладнання. Приклади включають SYN-флуд та Ping of Death;

– *атаки на рівні додатків (Application Layer Attacks).* Націлені на конкретні вебдодатки або сервіси та намагаються вичерпати їх ресурси. Приклади включають HTTP-флуд, Slowloris та RUDY.

Важливо зазначити, що зловмисники часто комбінують різні види DDoS-атак для підвищення їх ефективності та ускладнення захисту.

SQL-ін'єкція. SQL-ін'єкція являє собою вектор атаки на вебресурси, що взаємодіють з базами даних, шляхом впровадження неочікуваного SQL-синтаксису в запити. Експлуатуючи недостатню перевірку вхідних даних, зловмисник може виконувати довільні SQL-інструкції, що, залежно від особливостей СУБД, може призвести до несанкціонованого доступу до даних (читання, модифікація, видалення), файлової системи сервера та виконання операційної системи. Типовим сценарієм є використання GET- або POST-параметрів без належної екранізації спеціальних символів при формуванні SQL-запитів. Заходи протидії включають обов'язкову фільтрацію потенційно небезпечних символів, таких як лапки, та явне приведення числових значень до відповідного типу. Рекомендованою практикою також є застосування спеціалізованих бібліотек для побудови SQL-запитів, які забезпечують параметризацію запитів та запобігають інтерпретації даних як коду.

XSS-атака. Міжсайтовий сценарій (XSS) – це кібератака, яка використовує вразливі місця вебсайту для впровадження коду сценарію на стороні клієнта (рис. 1.3). Зловмисники використовують ці вразливості, щоб вставити шкідливі сценарії JavaScript у сховище даних вебсайту. Наступного разу, коли користувач запитує ці дані, браузер жертви виконає вбудований шкідливий код. Це може призвести до викрадених файлів cookie браузера, які необхідні для перехоплення сеансів

користувачів. Компрометація сеансу надає зловмиснику можливості для подальшої експлуатації, включаючи отримання конфіденційної інформації мережі та потенційний контроль над системою кінцевого користувача. У корпоративному середовищі атаки XSS становлять значну загрозу, оскільки успішне використання може скомпрометувати всю мережеву інфраструктуру [4].

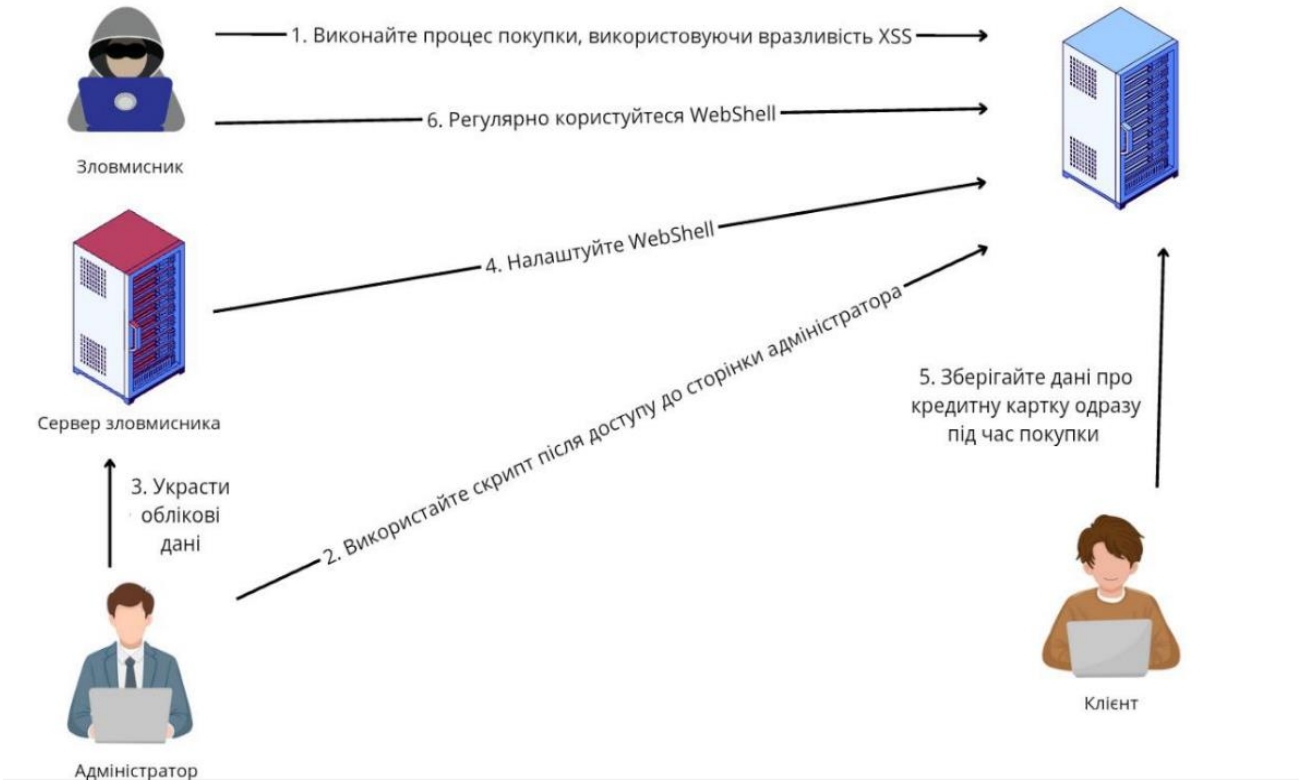


Рисунок 1.3 – Принцип роботи XSS-атаки

CSRF-атака. CSRF, або міжсайтове підроблення запиту – це вид атаки, що полягає в обмані користувача з метою відправлення шкідливого запиту до вебсервера. Атака використовує вже встановлене з'єднання та права жертви для виконання небажаних дій (за винятком особливого випадку – CSRF для входу). Оскільки браузер автоматично додає до запитів файли cookie сесії та інші облікові дані, вебсервер не може розрізнити справжній запит від підробленого, якщо користувач авторизований.

Цей тип атак спрямований на функції, які змінюють стан на сервері, наприклад, оновлення контактних даних, паролів або ініціювання транзакцій.

Зловмисник не отримує прямої вигоди від відповіді сервера, але змушує жертву несвідомо виконати шкідливу дію.

Окремий сценарій – login CSRF – дозволяє зловмиснику отримати доступ до особистих даних жертви. Він змушує неавторизованого користувача увійти до облікового запису, контрольованого зловмисником. Якщо жертва не помічає підміни, вона може ввести конфіденційну інформацію, наприклад, платіжні дані, які потім стануть доступні зловмиснику.

Buffer Overflow. Буфери є областями пам'яті, призначеними для тимчасового зберігання даних під час їхньої обробки. Переповнення буфера, або вихід за його межі, відбувається, коли програма намагається записати в буфер більше даних, ніж він може вмістити (рис. 1.4). Це призводить до того, що надлишкові дані перезаписують сусідні комірки пам'яті. Розглянемо приклад: якщо буфер для збереження імені користувача та пароля має розмір 8 байт, а користувач вводить 10 байт інформації, останні 2 байти будуть записані за межами буфера, потенційно пошкоджуючи інші дані в пам'яті [5].

Проблеми переповнення буфера можуть виникати в різноманітному програмному забезпеченні. Зазвичай їх спричиняють неправильно оброблені вхідні дані або недостатній обсяг виділеної пам'яті для буфера. Коли відбувається перезапис виконуваного коду, це може призвести до нестабільної поведінки програми, помилок при зверненні до пам'яті або її раптового закриття.

Зловмисники експлуатують переповнення буфера, змінюючи вміст пам'яті програми. Перезаписуючи певні ділянки, вони можуть змінювати логіку виконання програми, що призводить до пошкодження файлів або витоку конфіденційних даних. Наприклад, хакер може впровадити шкідливий код, щоб отримати несанкціонований доступ до інформаційних систем.

Маючи знання про організацію пам'яті програми, зловмисник може навмисно передати надмірні дані, викликаючи переповнення буфера та перезапис областей, де зберігається виконуваний код. Замінюючи цей код власним, він отримує контроль над пр ограмою. Зокрема, є можливість перезаписати вказівники пам'яті, спрямовуючи їх на свій шкідливий код (експлойт).



Рисунок 1.4 – Схематичне зображення переповнення буфера

1.3 Методи захисту серверів

Забезпечення безпеки сервера є критично важливим і передбачає впровадження ряду політик та методів для протидії різноманітним загрозам, від DDoS та брутфорс-атак до ненавмисних помилок або зловмисних дій користувачів. Для досягнення цієї мети можуть застосовуватися такі заходи, як налаштування та адміністрування брандмауерів, використання складних паролів і надійних протоколів автентифікації, встановлення антивірусних програм, регулярне створення резервних копій даних для запобігання їх втраті, а також розробка процедур безпеки та навчальних програм для співробітників.

Сервери відіграють дуже важливу роль в обробці та збереженні цінної для бізнесу інформації. Захист серверів від зовнішніх загроз має важливе значення для забезпечення:

– *цілісності.* Безпека серверів гарантує точність і повноту даних, що зберігаються на сервері, запобігаючи несанкціонованому доступу та випадковим змінам;

– *доступності.* Системні адміністратори повинні забезпечити безперебійний доступ серверів і сервісів для авторизованих користувачів без перебоїв;

– *конфіденційності.* Ключовою метою захисту серверів є запобігання несанкціонованому доступу до чутливих даних та їх зловмисному використанню;

– *репутації.* Компрометація безпеки може призвести до втрати даних або простою сервісів, завдаючи шкоди іміджу та спричиняючи фінансові втрати;

– *відповідності вимогам*. Законодавство часто зобов'язує компанії забезпечувати захист конфіденційних даних, що зберігаються на їхніх серверах, а недотримання цих вимог може призвести до значних штрафів.

Оновлення ПЗ. Забезпечення кібербезпеки серверної інфраструктури є багатоаспектним завданням, одним з критично важливих елементів якого виступає систематичне оновлення програмного забезпечення. В умовах ескалації складності та витонченості кіберзагроз, своєчасне впровадження оновлень та коригуючих патчів набуває статусу імперативу для підтримки цілісності, конфіденційності та доступності критично важливих інформаційних систем.

Розробники програмних продуктів здійснюють безперервний процес ідентифікації та елімінації потенційних вразливостей, що можуть бути присутні в кодовій базі їхніх розробок. Дані дефекти, що виникають внаслідок помилок програмування або архітектурних недоліків, можуть бути експлуатовані зловмисниками для несанкціонованого проникнення до серверних ресурсів, ініціювання виконання довільного коду, неправомірного отримання чутливої інформації або деструкції функціональності критично важливих сервісів. У цьому контексті, оновлення програмного забезпечення, що містять верифіковані виправлення безпеки, є першочерговим засобом нейтралізації виявлених векторів атак та мінімізації поверхні вразливості серверної інфраструктури. Ігнорування процедур оновлення експонує систему до відомих експлойтів, значно підвищуючи ймовірність успішної кібератаки.

Окрім аспектів безпеки, регулярне оновлення програмного забезпечення є вагомим фактором оптимізації продуктивності та забезпечення стабільності серверних платформ. Розробницькі колективи постійно працюють над удосконаленням алгоритмів, рефакторингом коду та виправленням виявлених еррорів, що можуть негативно впливати на ефективність використання системних

ресурсів та загальну надійність функціонування серверів. Впровадження оновлень сприяє підтримці оптимального стану операційного середовища, забезпечуючи безперебійність надання сервісів та раціональне використання обчислювальних потужностей [6].

Принцип сумісності також є значущим аргументом на користь систематичного оновлення програмного забезпечення. Еволюція інформаційних технологій супроводжується виходом нових версій операційних систем, допоміжного програмного забезпечення та апаратних компонентів. Регулярне оновлення забезпечує належну інтероперабельність між різнорідними елементами інфраструктури, запобігаючи виникненню конфліктів та забезпечуючи стабільну взаємодію. Відсутність своєчасних оновлень може призвести до проблем при інтеграції нових технологічних рішень або до деградації функціональності існуючих сервісів через порушення сумісності.

У низці регуляторних доменів існують чітко регламентовані вимоги щодо забезпечення безпеки оброблюваних даних, що включають положення про підтримку актуальності використовуваного програмного забезпечення. Невиконання зазначених вимог може призвести до значних фінансових санкцій та юридичної відповідальності. Систематичне оновлення серверного програмного забезпечення є необхідною умовою забезпечення відповідності нормативно-правовим актам та мінімізації потенційних ризиків, пов'язаних з недотриманням регуляторних стандартів.

Ризики, асоційовані з ігноруванням процедур оновлення серверного програмного забезпечення, є статистично значущими та потенційно деструктивними. До них належать підвищена ймовірність успішних кібератак, загроза компрометації та втрати критично важливих даних, можливість виникнення перебоїв у наданні ключових сервісів, зниження продуктивності обчислювальних ресурсів, проблеми інтеграції з новими технологіями, репутаційні втрати та потенційні юридичні наслідки. У сучасній інформаційній екосистемі, де дані є стратегічним активом, нехтування своєчасним оновленням програмного забезпечення на серверних платформах є неприпустимим фактором ризику, що

може мати серйозні та довгострокові негативні наслідки для функціонування бізнес-процесів.

Для ефективного управління процесом оновлення програмного забезпечення на серверних ресурсах рекомендується розробка та імплементація формалізованого плану, що включає періодичну верифікацію наявності нових версій, використання автоматизованих систем керування оновленнями, проведення ретельного тестування оновлень у некритичних середовищах перед їхнім розгортанням у виробничому контурі, пріоритетне встановлення критичних оновлень безпеки, ведення деталізованого обліку встановлених оновлень та обов'язкове створення резервних копій системи перед застосуванням значних оновлень.

Фаєрвол (Firewall). Мережевий екран визначається як система або комбінація систем, що забезпечують сегментацію мережі на дві або більше дискретних зон та застосування комплексу правил, які регламентують умови транзиту мережевих пакетів між цими зонами (рис.1.5). Типовим сценарієм є розмежування між корпоративною локальною мережею та публічною мережею Інтернет, проте подібна сегментація може бути реалізована і в межах внутрішньої мережевої інфраструктури підприємства. Брандмауер функціонує як контрольна точка, через яку проходить весь мережевий трафік. Для кожного окремого пакета мережевий екран приймає бінарне рішення щодо його подальшої маршрутизації (пропустити або відхилити), керуючись попередньо сконфігурованим набором правил. Детальний опис структури цих правил та використовуваних параметрів буде представлено в наступних розділах [7].

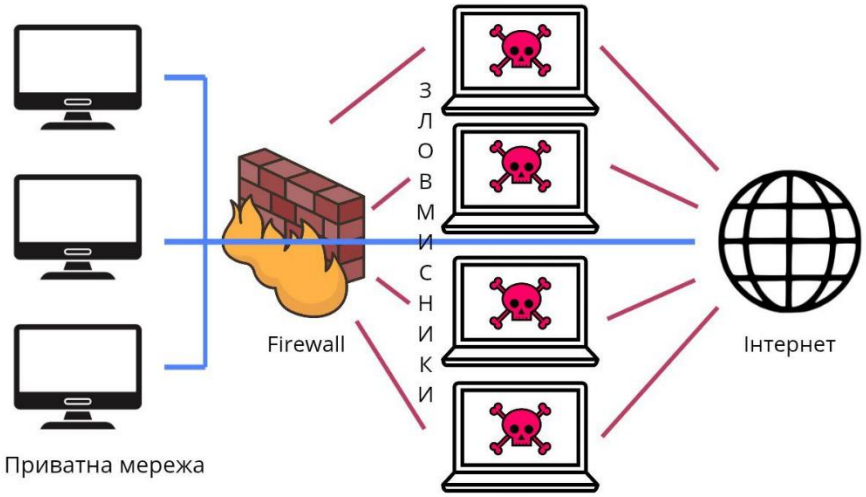


Рисунок 1.5 – Схема мережі з використанням брандмауера

Типовими операційними платформами для розгортання фаєрволів є UNIX-подібні системи (BSDI, SunOS, AIX, IRIX тощо), з меншою частотою використовуються DOS, VMS, WNT та Windows NT. Серед апаратних платформ превалюють INTEL, Sun SPARC, RS6000, Alpha, HP PA-RISC, а також процесори на базі RISC-архітектури R4400-R5000. Стосовно мережевих інтерфейсів, більшість брандмауерів підтримують Ethernet, FDDI, Token Ring, стандарти 100Base-T та 100VG-AnyLan, а також різноманітні послідовні інтерфейси. Мінімальні вимоги до оперативної пам'яті становлять зазвичай 32 МБ, а до дискового простору – 500 МБ, проте ці параметри можуть варіюватися залежно від розміру захищеного мережевого сегмента.

З метою підвищення рівня безпеки операційної системи, на якій функціонує фаєрвол, здійснюється модифікація ядра ОС та відповідних конфігураційних файлів. Принциповою вимогою є відсутність будь-яких користувацьких облікових записів на самому мережевому екрані (для мінімізації потенційних векторів атак), за винятком облікового запису адміністратора. Деякі брандмауери функціонують в однокористувацькому режимі. Більшість фаєрволів інтегрують систему верифікації цілісності виконуваних кодів, що передбачає порівняння контрольних сум програмних файлів із захищеними еталонними значеннями під час запуску для виявлення несанкціонованих змін.

Реалізація міжмережевого екрану може бути апаратною (у вигляді окремого

маршрутизатора) або програмною (у складі операційної системи персонального комп'ютера чи проксі-сервера). Прості апаратні або програмні фаєрволи початкового рівня можуть мати обмежену функціональність щодо гнучкого налаштування правил фільтрації пакетів та трансляції мережевих адрес (Network Address Translation, NAT) для вхідного та вихідного трафіку.

За критерієм відстеження стану активних мережевих з'єднань фаєрволи класифікуються на:

– *stateless-фаєрволи.* Вони здійснюють пакетну фільтрацію, базуючись виключно на статично визначених правилах, без аналізу контексту поточних з'єднань (наприклад, TCP-сесій);

– *stateful-фаєрволи.* Відповідають за забезпечення фільтрації з урахуванням стану з'єднань, аналізуючи послідовність пакетів та їх відповідність логіці роботи протоколів і застосунків. Цей тип фаєрволів демонструє вищу ефективність у протидії атакам типу "відмова в обслуговуванні" (DoS) та експлуатації вразливостей на рівні мережевих протоколів.

Для забезпечення різноманітних вимог безпеки існують три основні архітектурні типи брандмауерів: мережевого рівня, прикладного рівня та рівня сеансів. Кожен з цих типів базується на відмінних принципах контролю мережевого трафіку:

– фаєрволи мережевого рівня, що зазвичай реалізуються у вигляді екрануючих маршрутизаторів, здійснюють контроль трафіку на рівнях 3 та 4 моделі OSI, аналізуючи заголовки пакетів. Обмеженням даного підходу є відсутність контролю над даними вищих рівнів. Крім того, більшість пакетних фільтрів не надають вбудованих засобів аудиту та генерації сигналів тривоги щодо безпекових інцидентів;

– фаєрволи прикладного рівня, також відомі як проксі-сервери, забезпечують високий рівень ізоляції між внутрішньою мережею та зовнішнім середовищем (Інтернетом). Однак глибока інспекція пакетів на рівні застосунків може призводити до зниження пропускної здатності мережі, що зумовлює необхідність використання високопродуктивного обладнання для реалізації проксі-

функціональності;

– фаєрволи рівня сеансів поділяють архітектурну концепцію проксі-сервера з брандмауерами прикладного рівня. Ключова відмінність полягає в тому, що фаєрволи прикладного рівня потребують спеціалізованих проксі-застосунків для кожного сервісу (наприклад, FTP, HTTP), тоді як фаєрволи рівня сеансів є більш протоколо-незалежними та здатні обробляти широкий спектр мережевих протоколів.

Усі брандмауери можуть бути класифіковані як пакетні фільтри, шлюзи прикладного рівня та шлюзи рівня сеансів, і гібридні рішення можуть поєднувати функціональність кількох з цих типів.

Захист SSH, авторизація за ключами. SSH (Secure Shell) – це основний спосіб віддаленого доступу до сервера, і його захист є важливим кроком у забезпеченні безпеки (рис. 1.6). Щоб захистити доступ, важливо:

– *використовувати ключі для авторизації.* Один із найефективніших способів захисту SSH – це використання ключової пари замість пароля. Ключі, як правило, довші та складніші за паролі, що робить їх більш стійкими до брутфорсу (атак методом перебору). Важливо зберігати приватний ключ у безпечному місці і ніколи не передавати його незахищеними каналами;

– *зміна стандартного порту SSH.* За замовчуванням SSH використовує порт 22. Змінюючи його на незвичний порт, ви можете зменшити кількість спроб несанкціонованого доступу;

– *обмеження доступу.* Доступ до SSH слід обмежити тільки тими IP-адресами, які дійсно потребують віддаленого доступу. Це можна зробити за допомогою файлу конфігурації SSH або за допомогою системи міжмережевого екрану;

– *заборонити вхід під root.* Безпосередній вхід у систему з root-правами через SSH – погана практика. Краще створити окремого користувача з правами sudo і використовувати його для адміністративних завдань [8].

Правильне налаштування і захист SSH може значно підвищити безпеку вашого сервера, зменшивши ризик несанкціонованого доступу.



Рисунок 1.6 – Схема роботи SSH

Антивірус та антишпигунське ПЗ. Для забезпечення надійного захисту серверів, які є критично важливими елементами будь-якої сучасної інформаційної інфраструктури, від різноманітних кіберзагроз, зокрема вірусів та шпигунського програмного забезпечення, необхідне впровадження комплексних рішень. Ці рішення повинні поєднувати в собі функціональність як традиційного антивірусного програмного забезпечення, так і спеціалізованих інструментів для виявлення та нейтралізації шпигунського ПЗ [9].

Антивірусне програмне забезпечення, розроблене для серверних операційних систем, відіграє ключову роль у виявленні, блокуванні та видаленні шкідливих програм. До цього спектру загроз належать віруси, комп'ютерні черв'яки, троянські програми, програми-вимагачі та інші форми зловмисного програмного забезпечення, здатні порушити стабільність роботи сервера, призвести до втрати або пошкодження критично важливих даних, а також використовувати обчислювальні ресурси сервера в злочинних цілях, наприклад, для розсилки спаму або участі в DDoS-атаках. Основні функціональні можливості антивірусних рішень для серверів включають сканування файлової системи в режимі реального часу, що дозволяє виявляти загрози безпосередньо в момент їхньої появи або спроби проникнення, а також планове сканування всієї файлової системи або окремих її частин за заданим розкладом. Виявлення шкідливого коду здійснюється за допомогою аналізу сигнатур, що являють собою унікальні

послідовності байтів, характерні для відомих вірусів, а також за допомогою поведінкового аналізу, який дозволяє ідентифікувати підозрілу активність, навіть якщо сигнатура загрози ще не внесена до антивірусних баз. У разі виявлення заражених файлів, антивірусне програмне забезпечення намагається їх вилікувати, видаляючи шкідливий код, а якщо це неможливо, інфіковані об'єкти поміщаються в карантин – ізольовану область, де вони не можуть завдати шкоди системі, або ж видаляються. Для корпоративних середовищ особливо важливими є функції централізованого управління, які дозволяють адміністраторам контролювати стан захисту всіх серверів з єдиної консолі, отримувати звіти про виявлені загрози та застосовувати уніфіковані політики безпеки.

Паралельно з антивірусним захистом, важливу роль у забезпеченні безпеки серверів відіграє антишпигунське програмне забезпечення. Його основне завдання полягає у виявленні та видаленні шпигунських програм, які можуть таємно збирати різноманітну інформацію про діяльність системи та користувачів, перехоплювати введені дані, такі як облікові записи та паролі, відстежувати історію вебсерфінгу та іншу конфіденційну інформацію з подальшою передачею її зловмисникам. Хоча сервери зазвичай не використовуються для безпосередньої роботи користувачів в інтернеті, шпигунське ПЗ може проникнути на них через вразливості в іншому встановленому програмному забезпеченні або в результаті несанкціонованого доступу. Антишпигунські програми здійснюють сканування системи на предмет наявності відомих шпигунських програм, а також ведуть моніторинг активних процесів та системного реєстру для виявлення підозрілої активності, характерної для шпигунського ПЗ. У разі виявлення таких програм, вони блокуються та видаляються, запобігаючи таким чином потенційному витоку важливої інформації, яка може зберігатися або оброблятися на сервері.

Для забезпечення максимально ефективного захисту серверів рекомендується використовувати комплексні рішення, які об'єднують в собі функціональність антивіруса та антишпигунського програмного забезпечення. Крім того, такі комплексні рішення часто включають додаткові рівні захисту, такі як брандмауер, який контролює мережевий трафік та блокує несанкціоновані

спроби доступу до сервера, системи виявлення та запобігання вторгненням, що аналізують мережеву активність на предмет підозрілих шаблонів та блокують відомі атаки, а також засоби захисту від руткітів – прихованих шкідливих програм, які можуть маскувати іншу зловмисну діяльність.

При виборі антивірусного та антишпигунського програмного забезпечення для серверів слід керуватися кількома ключовими критеріями. Одним з найважливіших є продуктивність – обране рішення не повинно істотно впливати на швидкість та стабільність роботи сервера. Також необхідно враховувати надійність виявлення загроз, тобто здатність програмного забезпечення ефективно ідентифікувати як відомі, так і нові, ще не задокументовані загрози. Для корпоративних середовищ критично важливою є наявність централізованої системи управління, яка спрощує адміністрування захисту великої кількості серверів. Обов'язковою є підтримка серверних операційних систем, що використовуються в інфраструктурі (наприклад, різних дистрибутивів Linux або Microsoft Windows Server). І, нарешті, надзвичайно важливим є регулярне автоматичне оновлення антивірусних баз та самого програмного забезпечення, оскільки нові загрози з'являються постійно, і тільки актуальні бази даних можуть забезпечити належний рівень захисту.

Моніторинг та логування. Для ефективного захисту серверів необхідно здійснювати безперервний моніторинг їхнього стану та активності. Це передбачає постійне відстеження ключових показників продуктивності, таких як завантаження процесора, використання пам'яті та дискового простору, оскільки різкі зміни можуть свідчити про атаку або зловмисну діяльність. Важливо також контролювати доступність критично важливих сервісів, таких як вебсервери та бази даних, щоб оперативно виявляти їхню непрацездатність, спричинену як технічними проблемами, так і діями зловмисників [10].

Ключовим елементом моніторингу безпеки є ретельний аналіз журналів подій, включаючи системні журнали, журнали безпеки та журнали застосунків, для виявлення спроб несанкціонованого доступу, підозрілих дій користувачів або спрацювань систем виявлення вторгнень. Аналіз мережевого трафіку також є

важливим для виявлення аномальних з'єднань або незвично великих обсягів переданих даних, що можуть вказувати на атаку або витік інформації.

Крім того, для забезпечення цілісності системи необхідно контролювати важливі файли на предмет несанкціонованих змін. У випадку виявлення будь-якої підозрілої активності або відхилень від нормальних показників, повинна спрацьовувати система оповіщень, яка своєчасно інформує адміністраторів для вжиття необхідних заходів. Таким чином, постійний моніторинг є активною формою захисту, що дозволяє не лише реагувати на вже виниклі загрози, але й запобігати їхньому розвитку.

Лог-файли (англ. Log files) є структурованими текстовими файлами, що містять хронологізований журнал подій, пов'язаних з функціонуванням інформаційної системи. Ці файли можуть бути локалізовані на персональному комп'ютері або серверному обладнанні та включати в себе дані про операції, ініційовані програмними процесами або користувачами, значення системних параметрів та поточний стан системи в процесі її виконання. Додатково, лог-файли можуть містити інформацію про зафіксовані помилки в роботі системи. Процес генерації та збереження записів у лог-файлі визначається як логування.

Першочерговим призначенням логування є генерація інформації, що забезпечує можливість аналізу аномалій та дефектів у функціонуванні програмних систем. Зокрема, в контексті інтегрованих систем, при виникненні помилки на певному етапі виконання, саме логи надають змогу ідентифікувати джерело її виникнення, сприяючи локалізації та подальшій оперативній корекції. Разом з тим, слід зазначити, що не всі операції потребують фіксації в логах. Однак, існує ряд процесів, що є обов'язковими для логування, серед яких: ініціація та завершення виконання програми, транзиції стану програми в рамках бізнес-процесів, запити до систем керування базами даних у форматі SQL, а також параметри, пов'язані з безпекою, під час процедур реєстрації користувачів, аутентифікації та відновлення облікових даних.

Сегментація мережі. Захист серверів шляхом сегментації мережі передбачає поділ мережі на ізольовані сегменти з метою мінімізації ризиків. Це дозволяє

обмежити зону ураження у випадку кібератаки, контролювати доступ до критично важливих серверів та застосовувати індивідуальні політики безпеки до різних частин мережі.

Сегментація може бути реалізована фізично, шляхом використання окремого мережевого обладнання для кожного сегмента, або логічно, за допомогою віртуальних локальних мереж (VLAN) в межах одного фізичного пристрою. Більш гранулярний підхід – мікросегментація. Вона дозволяє ізолювати окремі віртуальні машини або навіть застосунки. Ключову роль у контролі трафіку між сегментами відіграють брандмауери.

Для ефективної сегментації необхідно визначити найважливіші сервери та дані, розробити чітку структуру сегментів з правилами їхньої взаємодії, застосовувати принцип найменших привілеїв при наданні доступу та регулярно переглядати й оновлювати правила сегментації. Ретельно спланована та впроваджена сегментація значно підвищує рівень безпеки серверної інфраструктури.

Системи виявлення вторгнень (IDS/IPS). Системи виявлення вторгнень (IDS) та системи запобігання вторгненням (IPS) є ключовими елементами для забезпечення належного моніторингу потенційних загроз безпеці. Незважаючи на їхню спільну спрямованість на захист від несанкціонованого проникнення, вони демонструють принципово різні підходи до контролю та моніторингу мережевого трафіку. Детальне обговорення цих відмінностей буде представлено нижче [11].

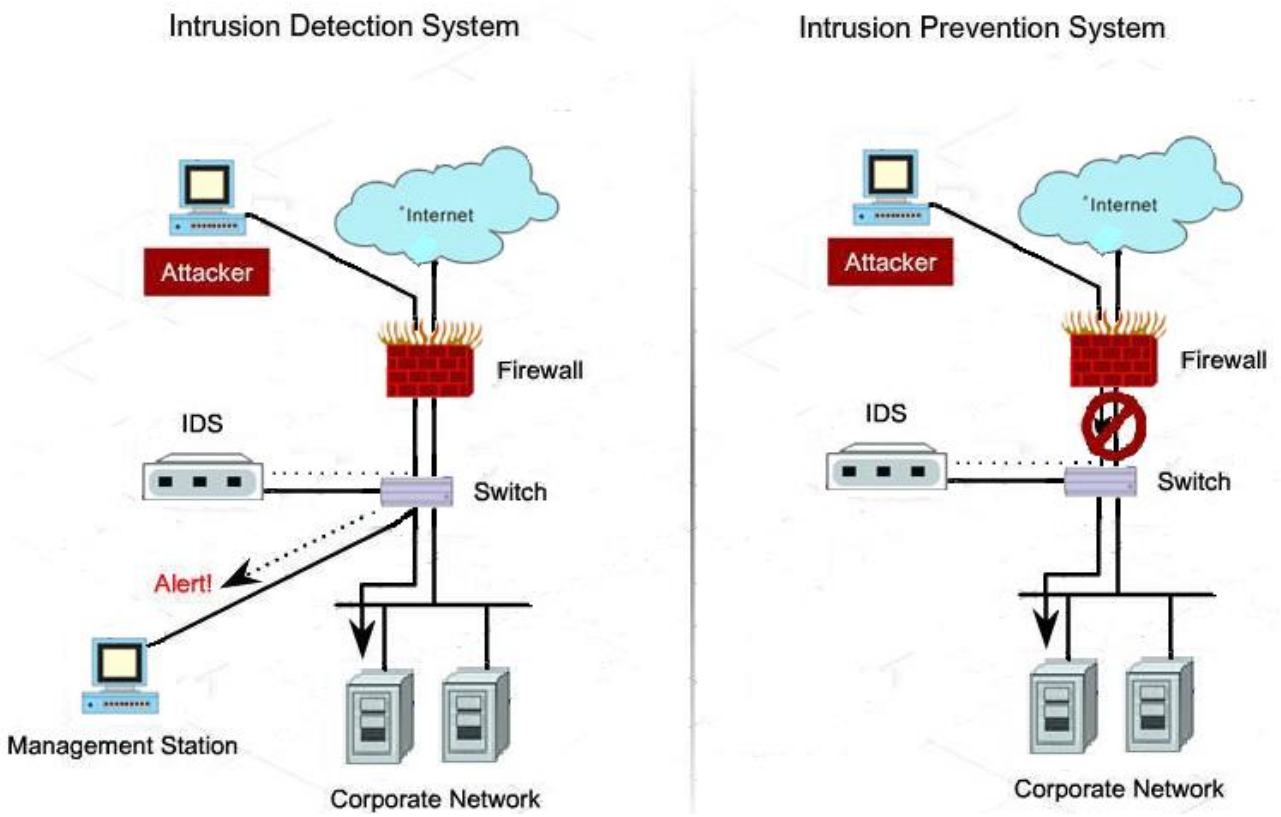


Рисунок 1.7 – Системи виявлення і запобігання вторгнень

Система виявлення вторгнень (IDS) – це програмний або апаратний засіб, призначений для моніторингу онлайн-загроз у мережі або окремій системі. При виявленні незвичної активності, IDS повідомляє про це адміністратора або передає дані в систему SIEM для подальшого аналізу. IDS може функціонувати трьома способами: виявляти аномалії, аналізуючи мережеву та системну активність і класифікуючи її як нормальну чи аномальну (цей метод, що часто використовує машинне навчання, добре виявляє нові загрози, але може давати хибні спрацьовування); здійснювати виявлення на основі сигнатур, шукаючи відомі шаблони атак (ефективно проти відомих загроз, але не проти нових); та використовувати виявлення на основі репутації, де оцінка репутації є ключовим фактором. Існують два основні типи IDS: NIDS, що аналізує вхідний та вихідний мережевий трафік, порівнюючи його з базою відомих атак для виявлення підозрілої активності; та HIDS, що контролює операційну систему на окремих хостах, відстежуючи пакети та повідомляючи про зміни критичних файлів.

IPS, або система запобігання вторгненням, виявляє загрозливу активність,

сповіщає про неї та намагається її блокувати. Зазвичай розташована за брандмауером, IPS ефективна у виявленні проблем безпеки, кіберзлочинців та загроз для документів. IPS запобігає атакам, модифікуючи їхній вміст, перенастроюючи брандмауери тощо. Деякі вважають IPS розширеною версією IDS через схожі функції моніторингу мережі. Методи роботи IPS включають: моніторинг протоколів зі збереженням стану (порівняння активності з узагальненими правилами для виявлення відхилень), моніторинг на основі сигнатур (пошук відомих шаблонів у мережевих пакетах) та статистичний моніторинг (порівняння поточної активності з попередньо визначеною "нормальною" базовою лінією, що може призводити до хибнопозитивних спрацювань при неправильній конфігурації). Після виявлення загрози IPS може заблокувати порушника, завершити TCP-сесію, видалити шкідливий контент або перенастроїти брандмауер для запобігання майбутнім атакам.

Резервне копіювання. Резервне копіювання для захисту серверів передбачає регулярне створення копій усіх важливих даних, конфігурацій та системних станів сервера. Ці резервні копії зберігаються в окремому, безпечному місці, відмінному від основного сервера. У випадку будь-якої непередбачуваної події, що призводить до втрати даних (наприклад, апаратний збій, кібератака, помилка програмного забезпечення або людський фактор), резервні копії можуть бути використані для відновлення сервера до попереднього працездатного стану, мінімізуючи час простою та запобігаючи втраті критично важливої інформації. Ефективна стратегія резервного копіювання включає визначення частоти створення копій, вибір відповідних методів резервування та тестування процедур відновлення.

Обмеження прав доступу. Обмеження прав доступу є ключовим принципом захисту серверів, що передбачає надання користувачам і процесам лише тих дозволів, які необхідні для виконання їхніх безпосередніх завдань. Такий підхід мінімізує потенційну шкоду від випадкових помилок, дій зловмисників (як внутрішніх, так і зовнішніх) та обмежує наслідки успішних кібератак.

Реалізація цього принципу включає кілька етапів: ідентифікацію користувачів, їхню аутентифікацію для підтвердження особистості та авторизацію,

яка визначає, які дії їм дозволено виконувати. Ефективним є використання ролей та груп для управління правами, а також застосування принципу "заборонено за замовчуванням", коли доступ надається лише явно. Регулярний перегляд та оновлення прав доступу є важливим для підтримки актуального рівня безпеки. Обмеження прав доступу значно зменшує поверхню атаки та обмежує радіус ураження у випадку компрометації системи.

Шифрування. Шифрування для захисту серверів є ключовим заходом безпеки, спрямованим на перетворення даних у нечитабельний формат, відомий як шифротекст. Лише авторизовані користувачі або системи, що володіють відповідним ключем дешифрування, можуть відновити оригінальні дані. Застосування шифрування на серверах є необхідним для забезпечення конфіденційності чутливої інформації, цілісності даних під час зберігання та передачі, а також для підтвердження автентичності джерела даних.

Існує два основних напрямки застосування шифрування на серверах: захист даних під час передачі та захист даних під час зберігання. Для захисту даних, що передаються між серверами та клієнтами (наприклад, веббраузерами), використовуються протоколи TLS/SSL, які забезпечують безпечне HTTPS-з'єднання. Для безпечного віддаленого адміністрування серверів застосовується протокол SSH, що шифрує команди та передачу файлів. Віртуальні приватні мережі (VPN) створюють зашифровані тунелі для захисту всього мережевого трафіку, а протокол IPsec забезпечує безпеку IP-комунікацій на рівні мережевих пакетів.

Для захисту даних, що зберігаються на серверах, використовуються різні методи. Повне шифрування диска (FDE) кодує весь вміст носіїв інформації. Шифрування на рівні файлової системи дозволяє вибірково шифрувати окремі файли або каталоги. Шифрування баз даних може застосовуватися як до цілих файлів баз даних, так і до окремих полів або записів. Для підвищення безпеки криптографічних операцій та зберігання ключів можуть використовуватися апаратні модулі, такі як TPM або HSM.

Застосування шифрування на серверах має численні переваги, включаючи захист від перехоплення даних, запобігання несанкціонованому доступу до

збереженої інформації, зниження ризиків витоку даних у випадку інцидентів безпеки, підвищення довіри користувачів та забезпечення відповідності нормативним вимогам. Однак існують і певні ризики та складності, такі як можливість втрати ключів шифрування, потенційний вплив на продуктивність сервера, складність налаштування та управління, а також ризики, пов'язані з вразливостями в алгоритмах шифрування.

Враховуючи зростаючу кількість кіберзагроз та вимоги до захисту конфідційної інформації, шифрування є невід'ємним елементом комплексної стратегії безпеки серверів. Правильне планування, вибір надійних методів шифрування та ефективне управління ключами є критично важливими для забезпечення високого рівня захисту без істотного впливу на працездатність системи.

1.4 Значення хмарних сервісів у забезпеченні безпеки

Хмарні сервіси набувають дедалі більшого значення у забезпеченні безпеки інформаційних систем, пропонуючи широкий спектр переваг та можливостей для захисту даних, інфраструктури та застосунків. В умовах зростання кількості та складності кіберзагроз, а також посилення вимог щодо відповідності нормативним стандартам, хмарні платформи стають ключовим елементом стратегії безпеки для багатьох організацій.

У сфері хмарних обчислень забезпечення безпеки є спільною відповідальністю між постачальниками хмарних послуг та їхніми клієнтами, причому розподіл обов'язків залежить від конкретного типу пропонованих послуг.

Загальнодоступні хмарні середовища, що знаходяться під повним контролем провайдерів і використовуються одночасно багатьма незалежними клієнтами, передбачають чіткий розподіл відповідальності.

Приватні хмарні середовища можуть управлятися як сторонніми постачальниками, так і розміщуватися безпосередньо в центрах обробки даних клієнта; в обох випадках доступ до серверних ресурсів має лише одна організація, що виключає необхідність спільного використання інфраструктури з іншими компаніями.

Гібридні хмарні середовища являють собою інтеграцію локальних центрів обробки даних з хмарними ресурсами сторонніх постачальників.

Багатохмарні середовища характеризуються використанням послуг щонайменше двох різних хмарних провайдерів.

Незалежно від використовуваного типу хмарного середовища або їхньої комбінації, концепція безпеки в хмарі спрямована на всебічний захист як фізичної інфраструктури, включаючи мережеве обладнання (маршрутизатори) та системи електропостачання, так і цифрових активів, таких як дані, пристрої зберігання, сервери, програмні застосунки, операційні системи, а також усього програмного та апаратного забезпечення.

Організації змушені використовувати хмарні технології для підтримки конкурентоздатності, прискорення розробки, спрощення доступу до сервісів для співробітників і клієнтів, а також для захисту даних і систем від таких загроз: компрометовані облікові записи (часто через фішинг), вразливості апаратного та програмного забезпечення (необхідність їхнього своєчасного оновлення), внутрішні загрози (людський фактор, неправильні конфігурації, переходи за небезпечними посиланнями, переміщення даних), недостатня видимість хмарних ресурсів (ускладнює виявлення та реагування на загрози), відсутність пріоритезації ризиків (перевантаження рекомендаціями, важливість фокусування на ключовому), ризиковані хмарні дозволи (збільшення векторів атак через нові сервіси та ідентичності), нові загрози (постійна еволюція ризиків), відсутність інтеграції розробки та безпеки (необхідність співпраці для виявлення проблем на ранніх етапах).

Існує чотири ключові сфери безпеки в хмарних обчисленнях, за які відповідальність розподіляється між постачальниками хмарних послуг та їхніми клієнтами.

Перша – це обмеження доступу, що є критично важливим з огляду на доступність усіх хмарних ресурсів через інтернет; необхідно забезпечити, щоб лише авторизовані користувачі мали доступ до необхідних інструментів протягом чітко визначеного періоду часу.

Друга – захист даних, що вимагає від організацій розуміння розташування своїх ресурсів та застосування відповідних контролів для забезпечення безпеки як самих даних, так і інфраструктури, на якій вони розміщені.

Третя – відновлення даних, оскільки у випадку порушення безпеки наявність надійного рішення для резервного копіювання та детального плану відновлення даних є абсолютно необхідною.

Четверта – план реагування на інциденти, який має бути розроблений організаціями для мінімізації наслідків атак та запобігання їхньому поширенню на інші системи [12].

Окрім цих основних аспектів, важливу роль відіграють такі підходи: Shift Left Security, що передбачає тісну співпрацю команд безпеки та розробки на всіх етапах створення хмарних застосунків для забезпечення їхнього захисту, починаючи з рівня коду; комплексна видимість захищеності DevOps, що забезпечує усунення "сліпих зон" за допомогою єдиного рішення для отримання аналітичної інформації про безпеку DevOps на різних платформах; та стратегічна зосередженість команди безпеки на виявленні та протидії новим загрозам шляхом оптимізації конфігурацій хмарних ресурсів безпосередньо на рівні коду, що дозволяє зменшити кількість потенційних проблем із безпекою в робочих середовищах.

2 ОГЛЯД ХМАРНИХ СЕРВІСІВ ЗАХИСТУ

2.1 Архітектура та принципи роботи Cloudflare

Cloudflare – це глобальна хмарна платформа, яка функціонує як зворотний проксі між кінцевими користувачами та вебсерверами. Її інфраструктура охоплює понад 335 міст у всьому світі, забезпечуючи низьку затримку та високу доступність.

Основні принципи роботи Cloudflare:

– *anycast-мережа.* Cloudflare використовує глобальну Anycast-інфраструктуру, яка дозволяє направляти запити користувачів до найближчого доступного периферійного вузла (edge-локації). Це не лише зменшує затримки, а й забезпечує більш рівномірне розподілення навантаження між дата-центрами. Anycast також сприяє підвищенню стійкості до відмов і зменшенню часу відгуку, оскільки в разі недоступності одного вузла запити автоматично перенаправляються до іншого. Завдяки цьому, Cloudflare може ефективно обслуговувати запити з будь-якої точки світу, мінімізуючи вплив відстані до серверів;

– *глобальна система кешування.* В основі роботи лежить складна система кешування, що функціонує у всіх його edge-серверах. Вона дозволяє зберігати статичний контент (зображення, CSS, JavaScript, HTML тощо) максимально близько до кінцевого користувача. Це значно знижує кількість звернень до оригінального (origin) сервера, зменшує його навантаження та прискорює доставку контенту. Додатково, кешуючі механізми можуть адаптуватися до популярності запитів, забезпечуючи ефективне використання ресурсу з урахуванням попиту. Завдяки цьому підхід до кешування є не лише розподіленим, але й динамічним;

– *інтеграція безпеки та продуктивності.* Однією з ключових переваг Cloudflare є об’єднання інструментів забезпечення безпеки та оптимізації

продуктивності в межах єдиної платформи, зокрема, функції Web Application Firewall (WAF), захисту від DDoS-атак, контролю бот-трафіку та шифрування трафіку (TLS) інтегровані безпосередньо в мережеву інфраструктуру, через яку проходить весь трафік. Паралельно працюють засоби прискорення, такі як CDN, кешування, мінімізація запитів та оптимізація маршрутів (наприклад, через Argo Smart Routing). Це дозволяє Cloudflare не лише блокувати шкідливі запити, а й забезпечувати швидке та стабільне з'єднання для легітимного трафіку.

Архітектура Cloudflare побудована за принципами децентралізації та високої доступності, що дозволяє платформі забезпечувати стабільну роботу вебресурсів навіть у складних мережевих умовах. Основу цієї архітектури становить глобальна мережа з більш ніж 200 дата-центрів, розташованих у стратегічних точках по всьому світу (рис. 2.1). Завдяки цьому Cloudflare мінімізує затримки при доставці контенту та забезпечує швидке реагування на запити з будь-якої точки планети [13].

Головною технічною особливістю архітектури є використання Anycast – технології маршрутизації, при якій один і той самий IP-адрес може бути доступний з кількох фізично різних локацій. Усі запити автоматично перенаправляються до найближчого edge-вузла (тобто найближчого до користувача дата-центру), що дозволяє мінімізувати шлях трафіку, скоротити час відповіді й ефективно балансувати навантаження. Навіть у разі виходу з ладу окремого вузла або регіонального збою, запити миттєво маршрутизуються до найближчого активного центру обробки без втрати працездатності.

Ще одним важливим аспектом є внутрішні алгоритми маршрутизації Cloudflare. Вони враховують не лише географічну відстань до користувача, а й поточний стан мережі, завантаження серверів, мережеві затримки, втрати пакетів та інші фактори. Усе це дозволяє динамічно обирати найефективніший маршрут доставки даних у режимі реального часу.

Кожен edge-вузол є самодостатнім: він здатен приймати, обробляти, кешувати запити, фільтрувати шкідливий трафік, застосовувати політики безпеки та виконувати частину логіки додатків. Завдяки цьому система не залежить від

централізованої обробки, що знижує ризик "вузьких місць" і дозволяє масштабувати інфраструктуру горизонтально.

Архітектура Cloudflare умовно поділяється на кілька основних структурних компонентів:

– *edge-сервери (периферійні вузли)*. Фізичні або віртуальні сервери, які розташовані у периферійних точках мережі. Саме вони приймають вхідні HTTP/HTTPS-запити користувачів. Edge-сервери здатні кешувати статичний контент, знижуючи навантаження на оригінальні сервери, виконувати аналіз і фільтрацію запитів за допомогою WAF (вебаплікаційного фаєрволу), а також захищати ресурси від DDoS-атак. На цьому рівні також реалізовано підтримку SSL/TLS-шифрування та різноманітних політик безпеки. Control Plane, що керує розподілом налаштувань, аналітикою, політиками маршрутизації і синхронізує їх між дата-центрами;

– *control Plane*. Централізований рівень керування, який відповідає за поширення налаштувань між усіма вузлами мережі. Через нього здійснюється розгортання конфігурацій, оновлення політик безпеки, контроль за сертифікатами, аналітика та моніторинг. Control Plane підтримує стан узгодженості між edge-серверами та забезпечує їхню синхронізовану роботу в режимі реального часу. Саме цей компонент дає змогу адміністраторам керувати сервісом через вебінтерфейс або API;

– *Argo Smart Routing*. Окрема підсистема маршрутизації, яка використовує інтелектуальний аналіз стану глобального інтернету. Вона динамічно обирає найкращий маршрут між вузлами Cloudflare та джерелом контенту на основі таких метрик, як затримка, пропускна здатність, відмови та нестабільність з'єднання. Це особливо важливо при взаємодії з бекенд-серверами, розміщеними у хмарних середовищах або на власній інфраструктурі клієнта;

– *Cloudflare Workers*. Середовище виконання JavaScript-коду без потреби у традиційному сервері. Користувачі можуть розгортати власну бізнес-логіку безпосередньо на edge-серверах Cloudflare. Це забезпечує мінімальні затримки та масштабованість без складних налаштувань. Workers використовуються для

обробки запитів у реальному часі, A/B-тестування, переписування URL-адрес, валідації запитів тощо;

– *механізми самовідновлення та моніторингу*. Cloudflare інтегрує систему постійного моніторингу стану своїх дата-центрів. У разі виявлення збоїв або деградації сервісу певного вузла, трафік миттєво перенаправляється до альтернативного маршруту. Це досягається за допомогою автоматичних health-checks і відмовостійких механізмів балансування [14].

Загалом архітектура Cloudflare – це приклад сучасного підходу до побудови розподілених мережевих систем, де висока доступність, безпека та продуктивність досягаються через розумне поєднання хмарних технологій, автоматизації та глобального масштабування.

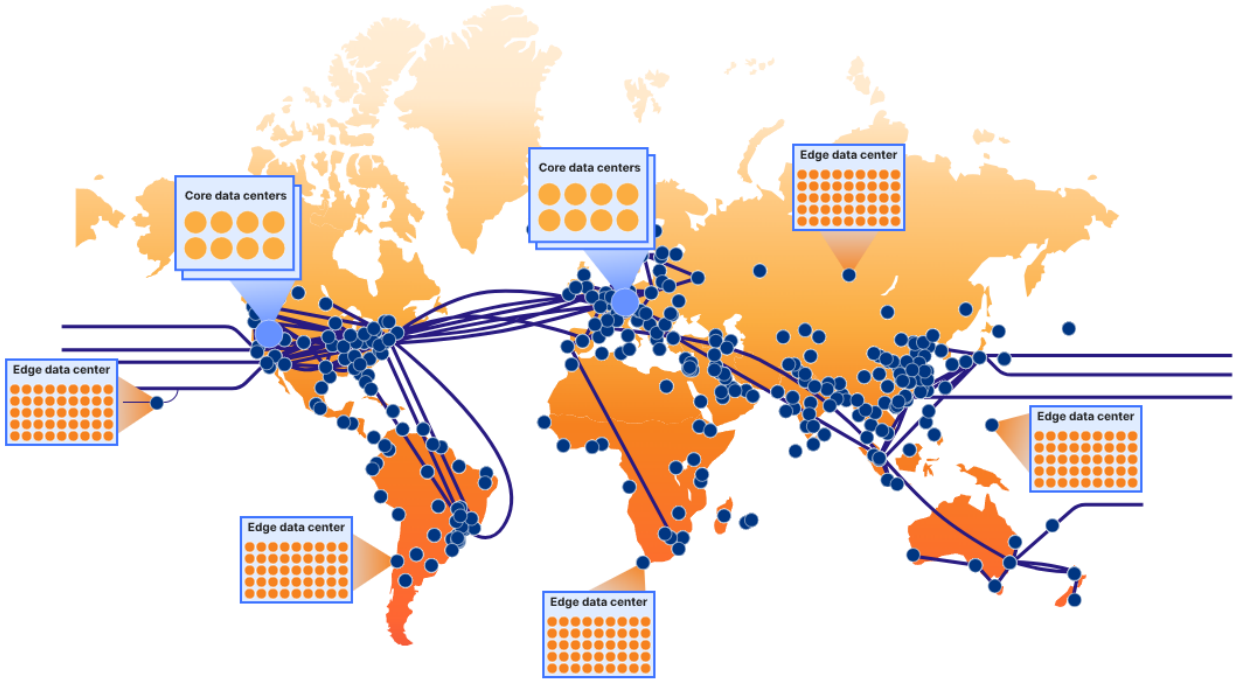


Рисунок 2.1 – Дата-центри Cloudflare у світі

Функціональні можливості. Використання Cloudflare для захисту CDN (Content Delivery Network). Cloudflare CDN є важливим інструментом для забезпечення ефективної доставки контенту кінцевим користувачам. Завдяки ньому, контент доставляється з найближчого до користувача сервера, що суттєво

зменшує затримки та покращує швидкість завантаження вебсторінок. Основні переваги:

– *зниження латентності.* Затримки між сервером та кінцевим користувачем. Це досягається шляхом кешування статичного контенту на периферійних серверах, розташованих у різних частинах планети. Кешування дозволяє зберігати копії популярного контенту, такого як зображення, відео та скрипти, на серверах CDN. Коли користувач запитує ці ресурси, запит обробляється сервером, що знаходиться найближче до нього, що суттєво скорочує час доставки та підвищує швидкість завантаження сторінки;

– *підвищення доступності.* Cloudflare також забезпечує підвищену доступність ресурсів, оскільки у разі відмови одного з дата-центрів, система автоматично перемикає запити на інші доступні сервери. Це дозволяє мінімізувати час простою та забезпечити стабільну роботу сайту навіть у випадку технічних збоїв або надмірних навантажень на окремі компоненти мережі;

– *оптимізація трафіку.* Використання Cloudflare CDN дозволяє значно зменшити навантаження на основні сервери, оскільки більшість запитів обробляються на периферійних серверах. Це знижує навантаження на основний сервер, а також економить пропускну здатність мережі. Таким чином, Cloudflare забезпечує більш ефективне використання мережевих ресурсів та знижує витрати на інфраструктуру.

WAF (Web Application Firewall). Cloudflare WAF – це вебекран прикладного рівня, який забезпечує захист вебдодатків від широкого спектра кібератак, орієнтованих на вразливості прикладного рівня (рис. 2.2). Він працює шляхом фільтрації, моніторингу та блокування шкідливого HTTP/S трафіку до вебдодатків, що дозволяє запобігати несанкціонованому доступу, крадіжці даних, зміненню контенту та іншим формам зловмисної активності [15].

Основним завданням Cloudflare WAF є захист вебдодатків від різноманітних атак, які націлені на вразливості вебресурсів, такі як SQL-ін'єкції, XSS (Cross-Site Scripting), CSRF (Cross-Site Request Forgery) та інші. Ці типи атак можуть призвести до серйозних порушень безпеки, тому використання WAF є критично важливим

для забезпечення безпеки вебдодатків.

Однією з основних переваг WAF, який пропонує Cloudflare, є здатність автоматично оновлювати правила захисту без необхідності ручного втручання з боку адміністратора. Це означає, що Cloudflare постійно оновлює базу даних із загрозами та новими уразливими точками для того, щоб захистити вебдодатки від новітніх атак. Адміністраторам не потрібно стежити за кожною новою загрозою або регулярно вносити зміни до правил – система забезпечує оновлення без участі людини, що значно спрощує процес захисту і робить його більш ефективним.

Іншою важливою особливістю Cloudflare WAF є інтеграція з OWASP Top 10 – списком найбільш поширених уразливостей в вебдодатках, який складається організацією OWASP (Open Web Application Security Project). Цей список включає такі загрози, як SQL-ін'єкції, XSS, CSRF, неправильна конфігурація безпеки та інші. Cloudflare WAF фільтрує трафік на основі цих уразливостей, що дозволяє забезпечити високий рівень захисту вебдодатків від загроз, до яких найбільш вразливі вебдодатки.

Ще однією важливою характеристикою WAF від Cloudflare є можливість налаштування і створення своїх правил для специфічних потреб кожного окремого вебдодатку. Це особливо корисно для організацій, які мають особливі вимоги до безпеки, або для тих, хто створює складні вебдодатки з нестандартними характеристиками. Власні правила дозволяють адаптувати систему безпеки до конкретних потреб і забезпечити максимальний рівень захисту від потенційних загроз, що можуть бути специфічними для певного додатку.

Це налаштування дає змогу більш точно реагувати на різні типи атак, блокуючи певні види трафіку або поведінки користувачів, які можуть бути шкідливими. До прикладу, можна створити правило, яке обмежує доступ до певних URL-адрес або блокує трафік з підозрілих географічних регіонів.

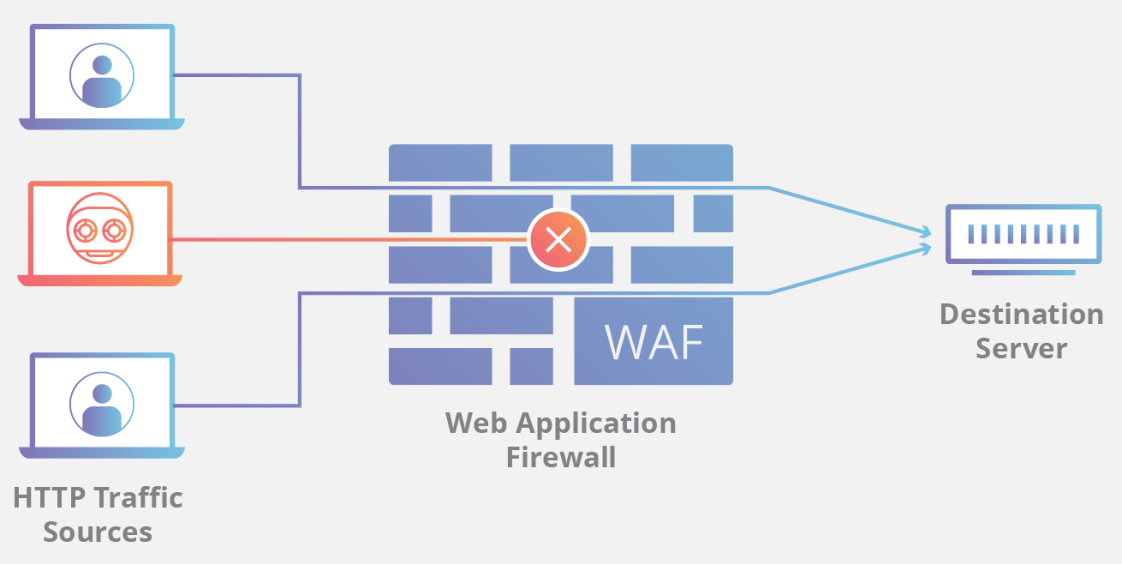


Рисунок 2.2 – Принцип роботи WAF

DDoS-захист. Cloudflare забезпечує надійний та масштабований захист від DDoS-атак, охоплюючи всі ключові рівні моделі OSI – мережевий (L3), транспортний (L4) та прикладний (L7) (рис. 2.3). Завдяки тому, що його архітектура розподілена по всьому світу та налічує велику кількість дата-центрів, об'єднаних у глобальну мережу з пропускною здатністю до 348 Тбіт/с, Cloudflare здатен не лише продіяти надзвичайно масштабним атакам, а й зводити до мінімуму затримки при обробці трафіку в будь-якій точці планети [16].

Захист від атак у Cloudflare побудований на автономних системах виявлення, які працюють у реальному часі. Алгоритми глибокого аналізу та машинного навчання дозволяють виявляти аномалії трафіку з високою точністю та застосовувати необхідні контрзаходи без втручання людини. Це дозволяє сильно зменшити час реакції на загрозу і підвищити ефективність протидії навіть найскладнішим атакам, у тому числі загрозам нульового дня.

Однією з ключових переваг підходу Cloudflare є адаптивність: система автоматично підлаштовується під типовий трафік конкретного клієнта, визначаючи відхилення від нормального профілю поведінки. У разі виявлення підозрілих патернів або сплесків запитів система динамічно генерує правила захисту, які можуть бути миттєво застосовані у всіх дата-центрах компанії. Така гнучкість особливо важлива для захисту від складних атак на прикладному рівні, які часто

імітують легітимні запити.

Cloudflare надає захист від DDoS-атак для всіх користувачів, незалежно від обраного тарифного плану – навіть безкоштовний рівень сервісу включає повноцінний базовий захист. Це є можливим завдяки масштабованості інфраструктури та ефективному використанню розподілених потужностей. Окрім автоматичних механізмів, користувачі також мають можливість самостійно налаштовувати політики безпеки, створювати правила фільтрації та користуватися інструментами моніторингу для оперативного реагування на інциденти.

Таким чином, комплексний підхід до захисту від DDoS-атак у Cloudflare охоплює технічну, алгоритмічну та організаційну складову. Компанія поєднує автономну аналітику, глобальну мережу, адаптивні системи та прозору модель управління безпекою, що дозволяє гарантувати високу надійність і безперервність роботи онлайн-ресурсів навіть в умовах інтенсивних і складних атак.

Zero Trust. Модель Zero Trust від Cloudflare Модель Zero Trust є однією з найбільш передових концепцій у сучасних системах безпеки, і вона реалізується через різноманітні технології, включаючи сервіси, пропоновані Cloudflare. Згідно з принципами Zero Trust, жоден користувач або пристрій не є довіреним за замовчуванням, незалежно від того, чи знаходяться вони всередині або поза межами корпоративної мережі. Це означає, що кожен запит на доступ до системи або ресурсу повинен проходити перевірку і верифікацію перед тим, як надасться доступ. Модель Zero Trust націлена на мінімізацію можливих ризиків, пов'язаних з компрометацією внутрішніх або зовнішніх елементів інфраструктури, шляхом впровадження жорсткої перевірки на кожному етапі взаємодії. Основні компоненти:

– *Cloudflare Access.* Сервіс для контролю доступу до корпоративних додатків на основі ідентифікації користувачів та пристроїв. Cloudflare Access використовує принципи багатофакторної автентифікації (MFA), що дозволяє створити високий рівень безпеки при доступі до критичних систем. Система перевіряє не лише особу користувача, а й конкретні параметри пристрою, з якого здійснюється підключення, такі як стан безпеки пристрою, наявність оновлень

безпеки та інші фактори. Цей підхід дозволяє впевнено контролювати доступ до ресурсів, навіть якщо пристрій користувача потрапив у мережу після компрометації. Важливо, що Cloudflare Access інтегрується з різними механізмами автентифікації та ідентифікації, включаючи існуючі служби Active Directory, Google Workspace або інші рішення для управління ідентифікацією, що дозволяє організаціям зберігати централізований контроль над користувачами та їх доступом [17];

– *Cloudflare Gateway.* Рішення для фільтрації вихідного трафіку та захисту від шкідливих ресурсів. Враховуючи, що в традиційних моделях безпеки мережі часто розглядаються як захищені зони з вбудованими механізмами безпеки, Cloudflare Gateway займається забезпеченням безпеки за межами корпоративної мережі. Система фільтрує трафік на основі поведінки користувачів та пристроїв, визначаючи шкідливі сайти, спам, фішинг-ресурси та інші загрози, що можуть вплинути на цілісність мережі. Завдяки інтеграції з аналітичними системами Cloudflare, Cloudflare Gateway здатний надавати розширену інформацію про шкідливий трафік та автоматично реагувати на загрози шляхом блокування доступу до небезпечних ресурсів. Цей підхід є важливим для організацій, які намагаються захистити свої користувацькі пристрої та корпоративні мережі від інтернет-загроз, зберігаючи при цьому високу продуктивність і мінімізуючи втрати від потенційних атак;

– *інтеграція з SASE (Secure Access Service Edge)*. Архітектурою, яка об'єднує мережеві та безпекові функції в єдину хмарну платформу. Це дозволяє забезпечити єдину точку доступу для всіх користувачів і пристроїв, незалежно від їхнього фізичного розташування. Завдяки такій інтеграції, безпека мережі стає більш централізованою та зручнішою для масштабування, оскільки всі безпекові функції (захист від загроз, контроль доступу, моніторинг) реалізуються через єдину хмарну платформу. Інтеграція Zero Trust з SASE дозволяє організаціям об'єднати функції фільтрації трафіку, управління доступом та забезпечення безпеки в єдину, масштабовану платформу, яка може ефективно працювати як для працівників, що перебувають у офісі, так і для віддалених користувачів або пристроїв[18].

Переваги Cloudflare:

- *швидке розгортання.* Інтеграція з існуючою інфраструктурою без необхідності зміни коду додатків;
- *масштабованість.* Можливість обробки великого обсягу трафіку без зниження продуктивності;
- *комплексний захист.* Поєднання різних механізмів безпеки в одній платформі;
- *безкоштовний план.* Надає базові функції без додаткових витрат.

Обмеження Cloudflare:

- *обмежена кастомізація.* Деякі функції можуть мати обмежені можливості налаштування;
- *залежність від третьої сторони.* Передача контролю над трафіком сторонньому провайдеру;
- *можливі конфлікти з існуючими системами.* Інтеграція може вимагати додаткових налаштувань для уникнення конфліктів з поточними рішеннями безпеки.

2.2 AWS Shield: Огляд типів, інтеграції та рівнів захисту

У зв’язку з тим, що однією з основних загроз є розподілені атаки типу відмови в обслуговуванні (DDoS), які можуть призвести до недоступності сервісів та значних фінансових втрат, компанія Amazon Web Services (AWS) пропонує сервіс AWS Shield, який забезпечує захист від DDoS-атак на різних рівнях [19].

AWS Shield Standard. AWS Shield Standard надається автоматично та безкоштовно всім клієнтам AWS. Цей рівень захисту забезпечує базовий захист від найпоширеніших мережевих та транспортних DDoS-атак (рівні 3 та 4 моделі OSI), таких як SYN/UDP-флуди, атаки відображення та інші. Shield Standard інтегрується

з такими сервісами, як Amazon CloudFront, Route 53 та AWS Global Accelerator, забезпечуючи їхню високу доступність навіть під час атак [18].

AWS Shield Advanced – це платний сервіс захисту від DDoS-атак, що пропонує користувачам хмарної інфраструктури Amazon Web Services більш високий рівень безпеки у порівнянні з базовою безкоштовною версією AWS Shield Standard (табл.2.1). Основна мета Shield Advanced – забезпечити автоматизований, гнучкий та ефективний захист як від типових атак, так і від більш витончених та масштабованих загроз, які можуть мати суттєвий вплив на працездатність хмарних сервісів і бізнесу в цілому. Основні можливості Shield Advanced включають:

- *розширене виявлення та пом'якшення наслідків атак.* Shield Advanced працює на рівнях 3, 4 та 7 моделі OSI, тобто забезпечує захист від атак як на мережевому (наприклад, volumetric floods), так і на транспортному рівнях (TCP SYN floods, UDP reflection), а також від складних атак прикладного рівня (HTTP floods, спроби перевантаження вебсерверів). Сервіс застосовує методи поведінкового аналізу трафіку, аномалій, сигнатурного виявлення та евристичних алгоритмів для точного визначення загроз;

- *Shield Advanced має повну інтеграцію з фаєрволом вебзастосунків (AWS Web Application Firewall)*, що дозволяє користувачам створювати гнучкі правила на основі IP-адрес, параметрів запитів, типів User-Agent тощо. Ця інтеграція дозволяє не лише автоматично блокувати шкідливий трафік, але й тонко налаштовувати відповіді на загрози в залежності від контексту додатку;

- *команда експертів*, яка допомагає виявляти, аналізувати та реагувати на атаки в режимі реального часу має цілодобовий доступ до команди реагування AWS Shield (SRT);

- *захист від фінансових втрат*. Однією з унікальних можливостей Shield Advanced є підтримка програми компенсації збитків від DDoS-атак. Якщо внаслідок атаки клієнт зазнає підвищених витрат на масштабування сервісів (наприклад, Amazon EC2, Elastic Load Balancing), він може подати запит на компенсацію у вигляді кредитів на обслуговування в AWS. Це важливий

інструмент для зменшення економічних ризиків, пов’язаних з кіберінцидентами;

– *групування ресурсів*. Для спрощення керування політиками захисту Shield Advanced дозволяє групувати кілька AWS-ресурсів у захищені групи. Наприклад, можна об’єднати разом кілька Load Balancer’ів, CloudFront-дистрибутивів або IP-адрес в єдину структуру. Це спрощує налаштування захисту на великомасштабних інфраструктурах та забезпечує централізоване управління [20].

Таблиця 2.1 – Порівняння Shield Standard та Shield Advanced

Характеристика	**AWS Shield Standard**	**AWS Shield Advanced**
Вартість	Безкоштовно для всіх AWS-користувачів	Платна підписка ($3 000/місяць на обліковий запис)
Рівні захисту (OSI)	L3 та L4	L3, L4 та L7
Захист від типових DDoS-атак	Так	Так
Аналіз складних атак	Базовий	Розширене виявлення, машинне навчання, поведінковий аналіз
Інтеграція з AWS WAF	Немає	Повна інтеграція, гнучке налаштування правил фільтрації
Підтримка від AWS SRT (24/7)	Немає	Так, доступ до спеціалізованої команди реагування у реальному часі
Компенсація фінансових втрат	Немає	Так, відшкодування у вигляді кредитів при збільшенні витрат через атаки
Групування ресурсів у політики захисту	Немає	Так, централізоване управління групами ресурсів
Аналітика та сповіщення	Базова панель моніторингу	Розширені звіти, аналітика, адаптивні сповіщення

Інтеграція з іншими сервісами AWS:

– *AWS WAF (Web Application Firewall)*. Вебаплікаційний фаєрвол, який дозволяє створювати правила для фільтрації HTTP/HTTPS-трафіку. Інтеграція AWS Shield Advanced з AWS WAF забезпечує багаторівневий захист, дозволяючи блокувати шкідливі запити ще до того, як вони досягнуть додатку. Крім того, підписка на Shield Advanced покриває витрати на використання стандартних

можливостей AWS WAF для захищених ресурсів [21];

– *Amazon Route 53*. Керований сервіс доменних імен (DNS) від Amazon Web Services, який забезпечує надзвичайно надійний, масштабований та високодоступний механізм для маршрутизації користувацьких запитів до інтернет-ресурсів, таких як вебсайти, API, додатки або інші сервіси, що розміщені в хмарі або локально. Цей сервіс є невід'ємною частиною захисної інфраструктури AWS, особливо у контексті DDoS-атак. Завдяки інтеграції з AWS Shield (як стандартної, так і розширеної версії), Route 53 здатен автоматично виявляти і блокувати аномальну активність, спрямовану на перевантаження DNS-рівня, який є критично важливим для функціонування будь-якого ресурсу в Інтернеті. Захист DNS дозволяє зменшити ризик відмови в обслуговуванні через спрямовані атаки на систему іменування;

Один із ключових технічних принципів Route 53 – використання Anycast-маршрутизації. Завдяки їй DNS-запити клієнтів спрямовуються до найближчої географічно точки присутності (PoP), що забезпечує не лише високу швидкість обробки запитів, але й зменшує вразливість до атак, які націлені на конкретні регіони або сервери. Якщо якась точка мережі стає недоступною або зазнає атаки, трафік автоматично перенаправляється до іншої доступної точки без шкоди для кінцевого користувача. Таким чином, Amazon Route 53 виконує не лише функції класичного DNS-сервісу, а й виступає в ролі першої лінії оборони, підвищуючи загальну стійкість інфраструктури AWS до DDoS-загроз [22];

– *Amazon CloudFront*. Мережа доставки контенту (CDN), яка забезпечує швидку доставку контенту користувачам по всьому світу. Інтеграція з AWS Shield дозволяє фільтрувати трафік на периферії мережі, зменшуючи навантаження на основні сервери та забезпечуючи захист від DDoS-атак на ранніх етапах.

Рівні захисту та можливості. AWS Shield забезпечує захист на трьох основних рівнях моделі OSI:

– *рівень 3 (мережевий).* Захист від об'ємних атак, таких як UDP-флуди;

– *рівень 4 (транспортний).* Захист від атак, спрямованих на виснаження

ресурсів, таких як TCP SYN-флуди;

- *рівень 7 (прикладний, лише у платній версії).* Захист від специфічних атак на вебдодатки, таких як HTTP GET/POST-флуди, за допомогою інтеграції з AWS WAF.

Також на відміну від стандартної версії, AWS Shield Advanced надає:

- можливість відстежувати атаки та отримувати сповіщення через Amazon CloudWatch в реальному часі;
- здатність автоматично реагувати на атаки без втручання користувача;
- захист ресурсів у всіх регіонах AWS та на глобальних точках присутності.

2.3 Порівняння Cloudflare та AWS Shield

У ході аналізу основних критеріїв оцінки хмарних сервісів Cloudflare та AWS Shield було виявлено низку суттєвих відмінностей, які визначають доцільність їх застосування у різних проєктних умовах.

Cloudflare є комплексною хмарною платформою, що поєднує функції доставки контенту (CDN), захисту від DDoS-атак, брандмауера вебзастосунків (WAF), системи управління DNS, оптимізації трафіку та кешування. Його архітектура дозволяє ефективно захищати вебресурси незалежно від середовища їх розміщення. Натомість AWS Shield – це спеціалізоване рішення, призначене винятково для забезпечення захисту інфраструктури в межах хмарної екосистеми Amazon Web Services. Відтак, його застосування обмежується лише проєктами, що побудовані на AWS [23].

Функціональні можливості Cloudflare є більш розвиненими. Зокрема, сервіс має вбудовану глобальну CDN-мережу з понад 300 дата-центрами, яка активується автоматично та не потребує додаткового налаштування. AWS для доставки контенту використовує окремий сервіс CloudFront, який повинен бути попередньо

інтегрований. Захист від DDoS-атак у Cloudflare реалізований на всіх рівнях моделі OSI – мережевому (L3), транспортному (L4) та прикладному (L7). При цьому навіть базовий безкоштовний тариф Cloudflare включає ці механізми. AWS Shield у безкоштовній версії Standard забезпечує лише обмежений рівень захисту (L3/L4), а повноцінний захист з підтримкою L7 можливий лише у платній версії Shield Advanced, яка вимагає окремої підписки.

Іншою відмінністю Cloudflare є наявність вбудованого WAF, доступного у платних тарифах, що дозволяє фільтрувати шкідливий трафік ще на рівні мережевої периферії. У AWS цей функціонал реалізується через окремий сервіс – AWS WAF, що ускладнює загальну архітектуру та вимагає додаткового налаштування. Підтримка шифрування в Cloudflare реалізується через автоматично активований Universal SSL, який надає безпечне з'єднання з використанням TLS без додаткових витрат. У випадку з AWS, сертифікати SSL/TLS необхідно видавати через AWS Certificate Manager (ACM), що додає етап у налаштуваннях [24].

Захист від ботів у Cloudflare інтегрований у платні тарифи та базується на поведінковому аналізі, тоді як у AWS подібна функція реалізується через окремий сервіс AWS Bot Control. Ще однією перевагою Cloudflare є його повна незалежність від платформи розміщення – він може бути легко інтегрований у будь-який проєкт через зміну DNS-записів. У випадку з AWS Shield, підключення можливе лише до внутрішніх сервісів AWS (таких як EC2, ELB, CloudFront), що значно звужує спектр застосування.

Cloudflare також забезпечує ефективне кешування контенту, зокрема через механізми оптимізації зображень, маршрутизацію трафіку за допомогою Argo Smart Routing, а також підтримку edge-обчислень через Cloudflare Workers. Подібна функціональність в AWS реалізується роздільно через сервіси Lambda@Edge або CloudFront, що потребує додаткових витрат і технічної інтеграції. В аналітичному аспекті Cloudflare надає базову статистику навіть у безкоштовному тарифі, а в розширених – повний набір журналів подій та звітів. AWS Shield Standard має обмежену аналітику, тоді як Shield Advanced підтримує повну інтеграцію з AWS CloudWatch, VPC Flow Logs та GuardDuty.

У контексті масштабованості Cloudflare демонструє перевагу за рахунок глобального розгортання без участі користувача. AWS Shield масштабується в межах заданої архітектури AWS, що вимагає правильного проєктування інфраструктури. З точки зору вартості Cloudflare пропонує гнучку модель з можливістю безкоштовного користування, тоді як Shield Advanced має фіксовану високу ціну – близько $3000 на місяць, що не є оптимальним для малих та середніх проєктів. У частині сервісного обслуговування Cloudflare забезпечує email-підтримку в безкоштовному тарифі та пріоритетне обслуговування у платних планах. AWS Shield надає повноцінну підтримку тільки у версії Advanced.

Окремої уваги заслуговує зручність налаштування. Cloudflare вирізняється простим вебінтерфейсом, автоматизованим керуванням та швидким впровадженням, тоді як AWS Shield вимагає від фахівців глибокого знання хмарної екосистеми Amazon.

Результати порівняльного аналізу основних характеристик сервісів були внесені в таблицю 2.2.

Таблиця 2.2 – Порівняльний аналіз Cloudflare та AWS Shield

Параметр	**Cloudflare**	**AWS Shield**
Тип сервісу	Хмарна платформа для забезпечення безпеки, кешування, маршрутизації та CDN.	Хмарний сервіс для захисту від DDoS-атак у межах інфраструктури AWS.
Призначення	Захист вебресурсів, незалежно від хостингу.	Захист AWS-ресурсів, таких як EC2, CloudFront, ELB.
CDN	Так, з понад 300 дата-центрами по всьому світу.	Через AWS CloudFront (окремо).
DDoS-захист	Включено у всі тарифи. Захищає на рівнях L3, L4 та L7.	Standard – базовий захист (L3, L4). Advanced – повний захист (включно з L7 через WAF).
WAF (Web Application Firewall)	Так, доступний у платних тарифах.	Окремий сервіс AWS WAF, інтегрується з Shield Advanced.
Підтримка SSL/TLS	Безкоштовні сертифікати, підтримка шифрування end-to-end.	Через AWS ACM – оплачується окремо.
Bot Management	Вбудований механізм для виявлення та блокування ботів (в платних тарифах).	AWS Bot Control – як окремий сервіс.

Інтеграція з хостингом	Працює з будь-якими платформами через зміну DNS.	Тільки для ресурсів, що працюють в AWS.
Кешування та оптимізація	Потужні можливості кешування, оптимізація зображень та маршрутів (Argo).	Відсутнє. Використовується через сторонні AWS-сервіси.

Продовження таблиці 2.2 – Порівняльний аналіз Cloudflare та AWS Shield

Параметр	**Cloudflare**	**AWS Shield**
Аналітика та логування	У базовому плані обмежено, в платному – повна аналітика, журналювання подій, інтеграція з SIEM.	Shield Advanced підтримує CloudWatch, VPC Flow Logs, GuardDuty.
Масштабованість	Автоматичне глобальне масштабування.	Автоматично масштабується в межах AWS.
Ціноутворення	Від безкоштовного до корпоративних тарифів із розширеним захистом.	Standard – безкоштовно, Advanced – від $3000/місяць.
SLA	100% для Enterprise тарифів.	99.99% у Shield Advanced.
Підтримка	Email-підтримка у безкоштовному, пріоритетна у платних тарифах.	Стандартна або розширена в Shield Advanced.
Зручність у використанні	Простий вебінтерфейс, швидке налаштування.	Потребує знань AWS-інфраструктури.
Переваги при виборі	Підтримка різних платформ, безкоштовна версія, простота у впровадженні, можливість кешування та edge-обчислень (Workers).	Глибока інтеграція з AWS, автоматична координація з іншими AWS-сервісами.

2.4 Вибір оптимального сервісу на основі аналізу

У сучасних умовах вебресурси можуть бути розгорнуті в різних середовищах – від класичних віртуальних серверів до масштабованих хмарних платформ. Залежно від конкретних потреб системи – зокрема типу розміщення, обсягу трафіку, бюджету та рівня ризику кіберзагроз – доцільність застосування того чи іншого хмарного сервісу безпеки може суттєво відрізнятись.

У випадках, коли вебресурс функціонує поза межами хмарної інфраструктури Amazon Web Services, найбільш практичним є використання сервісу, який не вимагає глибокої інтеграції з певною платформою. Cloudflare у цьому аспекті демонструє високу гнучкість: він може бути впроваджений на будь-якому хостингу шляхом делегування DNS-записів, не змінюючи архітектуру самої системи. Натомість AWS Shield не підтримує зовнішні інфраструктури і обмежений тільки ресурсами, що безпосередньо розгорнуті в AWS.

При роботі з проєктами, розміщеними всередині хмарного середовища Amazon, AWS Shield виглядає логічним варіантом. Його можливості максимально розкриваються у версії Shield Advanced, яка підтримує взаємодію з внутрішніми сервісами AWS (EC2, ELB, CloudFront) та інтегрується з системами моніторингу. Однак така функціональність передбачає високу вартість ($3000/місяць), а також додаткову потребу в налаштуванні суміжних сервісів для повноцінного захисту.

У випадках, коли система обслуговує значну кількість користувачів або відчуває пікові навантаження, особливої ваги набувають функції кешування, оптимізації трафіку та розподілу навантаження. Cloudflare забезпечує ці можливості вбудовано, включно з глобальною CDN-мережею, Argo Smart Routing та підтримкою edge-обчислень. У AWS відповідні функції реалізуються через окремі сервіси, що призводить до ускладнення архітектури й підвищення витрат.

Фінансова доступність сервісу також є критичним фактором, особливо на початкових етапах розвитку проєкту або у випадку створення MVP (minimum viable product). Cloudflare має чітко структуровану модель ціноутворення, де базовий

рівень доступний безкоштовно, при цьому вже забезпечує базовий захист, DNS, SSL, WAF та кешування. AWS Shield не пропонує безкоштовного рівня із розширеною функціональністю – для отримання повного спектру послуг необхідно використовувати версію Advanced.

У випадку підвищеного ризику кіберзагроз, вимог до аналітики та аудиту подій безпеки, Cloudflare також забезпечує надійне рішення. У платних тарифах доступні інструменти поведінкового аналізу, деталізовані журнали доступу, інтерфейси інтеграції з SIEM-системами. AWS Shield Advanced також підтримує глибоке журналювання та аналітику, проте реалізується це шляхом підключення до декількох окремих сервісів (CloudWatch, GuardDuty, VPC Flow Logs), що збільшує вимоги до адміністрування.

Загалом, аналіз показує, що Cloudflare значно перевершує AWS Shield у категоріях універсальності, простоти впровадження, гнучкості налаштувань, інтегрованості функцій і фінансової доступності. Саме ці переваги визначають доцільність його використання в межах цього проєкту. Обраний підхід дозволяє забезпечити комплексний захист вебресурсу без залежності від конкретної інфраструктури, при цьому мінімізуючи витрати на підтримку, розгортання та технічне адміністрування.

3 ПРАКТИЧНА РЕАЛІЗАЦІЯ ЗАХИСТУ СЕРВЕРА З ВИКОРИСТАННЯМ CLOUDFLARE

3.1 Розгортання тестового вебресурсу

У межах практичної частини було розгорнуто тестовий вебсайт, який надалі використовується для демонстрації підключення та налаштування хмарного захисту за допомогою сервісу Cloudflare. Для розміщення тестового вебресурсу було обрано безкоштовний хостинг Netlify, який надає базовий функціонал для розміщення статичних сайтів без фінансових витрат, що повністю відповідає вимогам практичної частини дослідження (рис. 3.1).

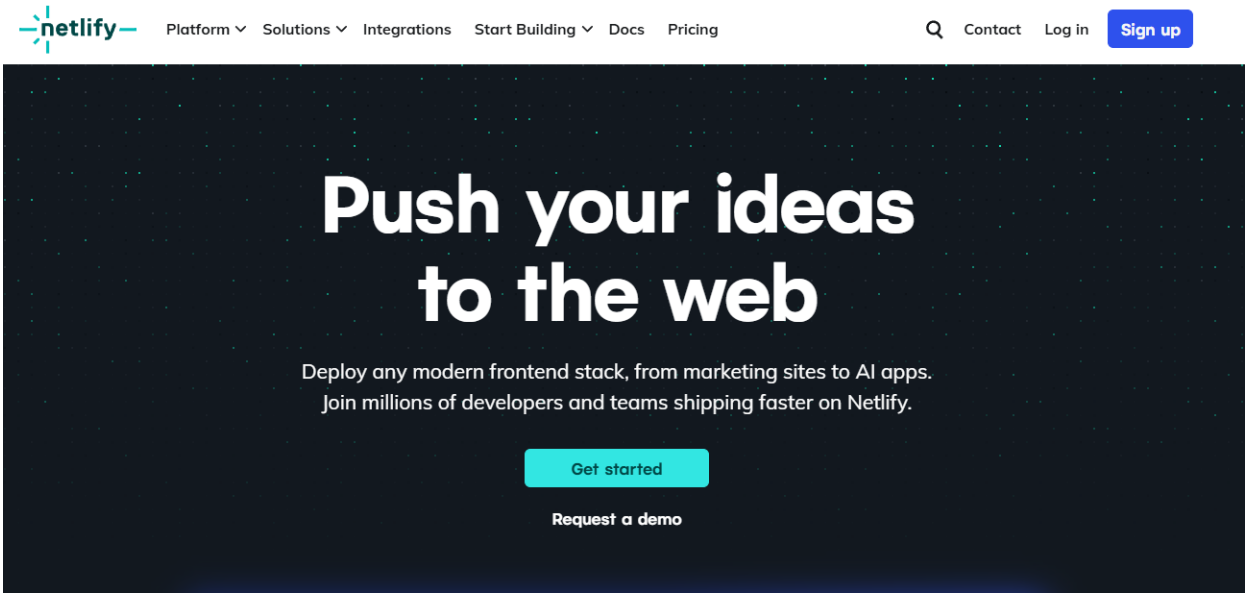


Рисунок 3.1 – Головна сторінка хостингу Netlify

Для публікації тестового вебресурсу було створено новий обліковий запис. Для синхронізації із Cloudflare до нього було підв’язано завчасно придбаний домен testcloudsecurity2025.website (рис. 3.2).

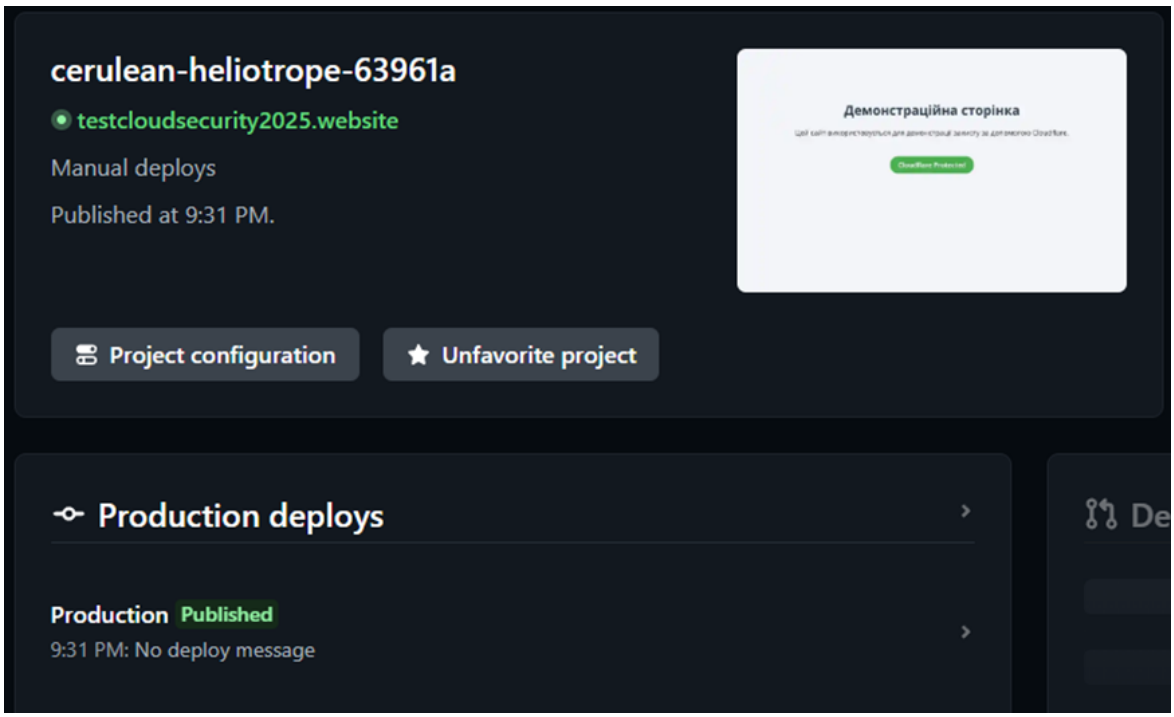


Рисунок 3.2 – Реєстрація доменного імені

Для демонстрації був написаний простий HTML-код вебсторінки (дод.А), що містить заголовок, опис призначення сайту і стилізований елемент із позначенням майбутнього захисту через Cloudflare (рис. 3.3).

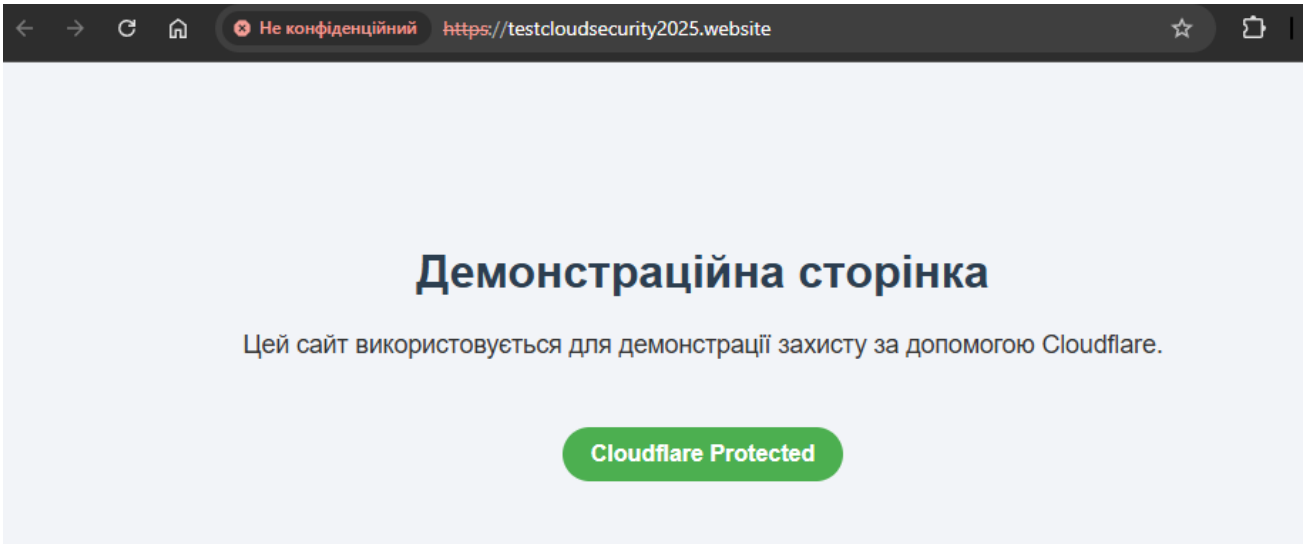


Рисунок 3.3 – Демонстраційна сторінка, яку було завантажено на хостинг

Цей етап забезпечує створення інфраструктурної основи для подальшого підключення захисних механізмів.

3.2 Підключення вебресурсу до Cloudflare та налаштування захисту

Для забезпечення додаткового рівня захисту вебресурсу було прийнято рішення використовувати сервіс Cloudflare. Цей сервіс дозволяє не лише пришвидшити роботу сайту завдяки CDN, але й ефективно захистити його від DDoS-атак, шкідливого трафіку, ботів тощо.

На сторінці https://dash.cloudflare.com було створено новий обліковий запис і підв'язано до нашого вебресурсу http://testcloudsecurity2025.website (рис. 3.4). Обрано безкоштовний тарифний план (Free Plan).

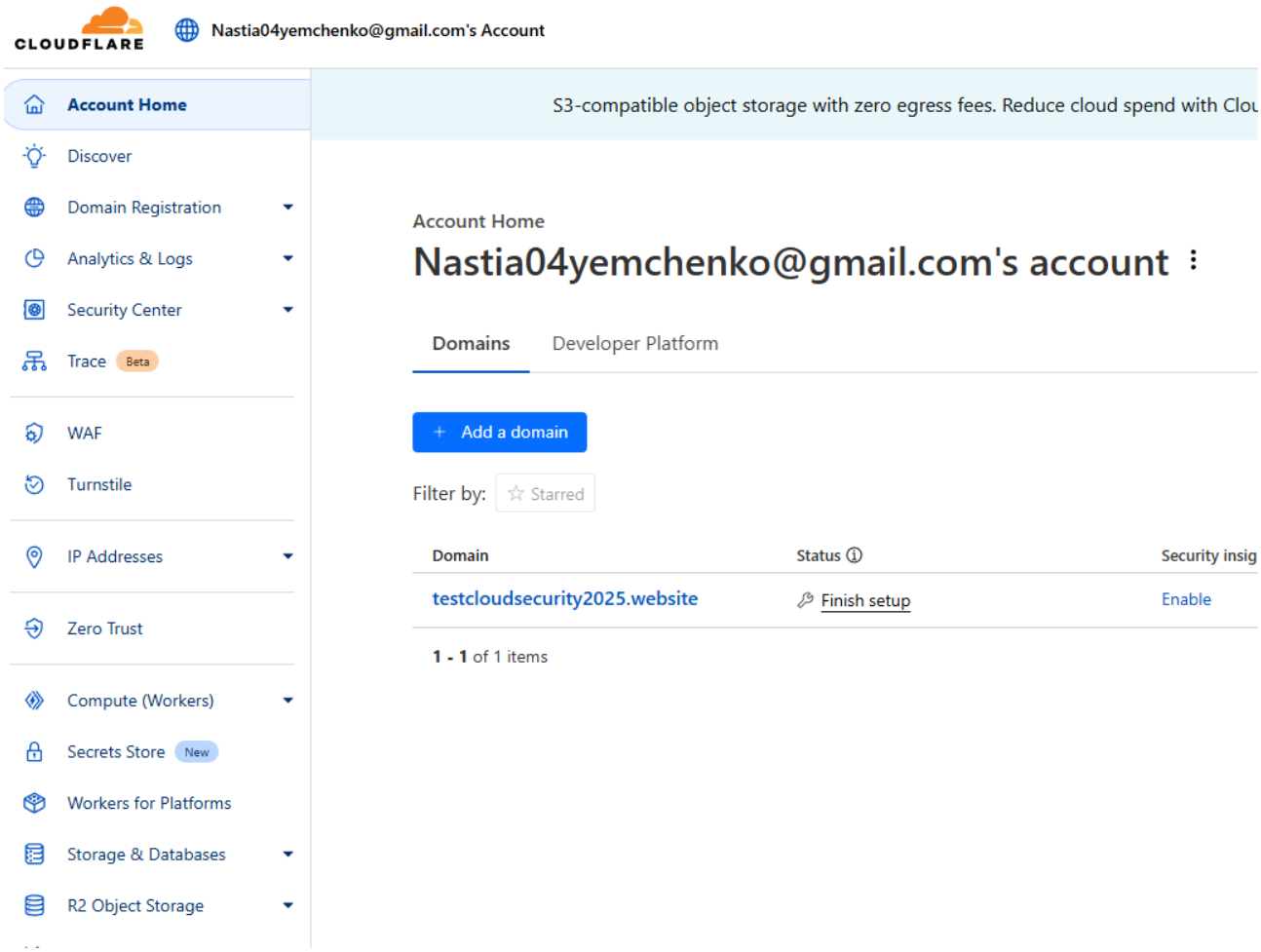


Рисунок 3.4 – Прив'язка тестового вебресурсу до облікового запису Cloudflare

За запитом Cloudflare в налаштуваннях DNS провайдера було також змінено name-сервери на ті, що належать Cloudflare замість тих, що надав хостинг Netlify (рис. 3.5).

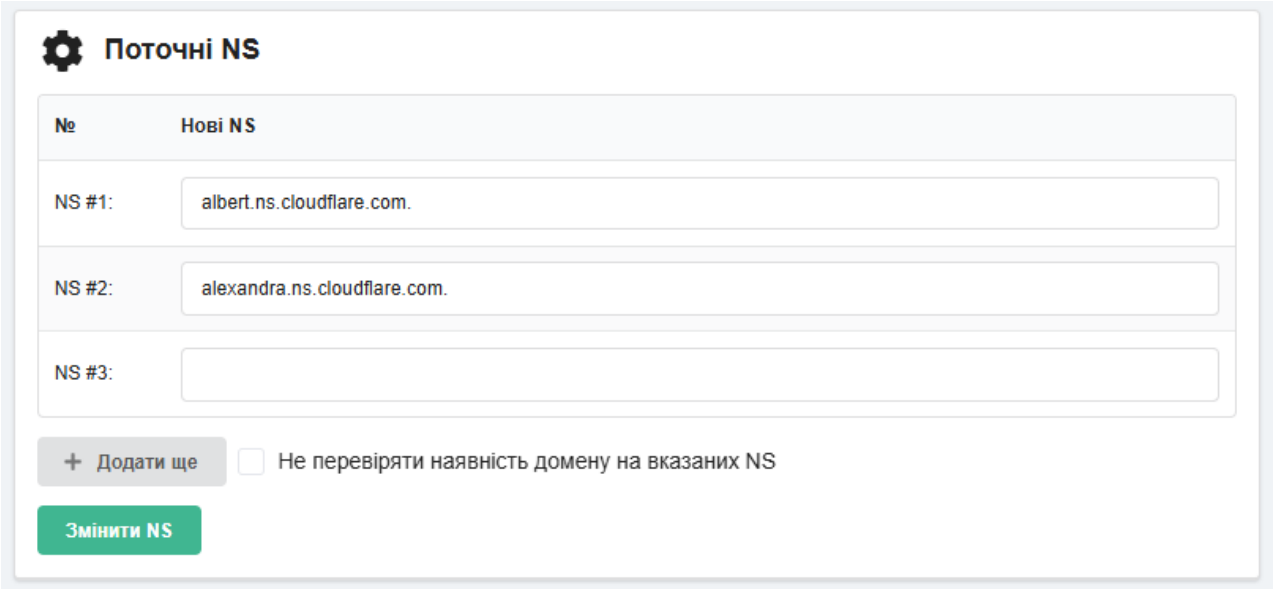


Рисунок 3.5 – Нейм-сервери Cloudflare

Оскільки сайт було завантажено на платформу Netlify, для коректного відображення ресурсу необхідно налаштувати CNAME-запис у Cloudflare (рис. 3.6). Таким чином, запити до основного домену переадресовуються на відповідний ресурс Netlify через мережу Cloudflare, що дозволяє застосовувати до сайту хмарний захист.

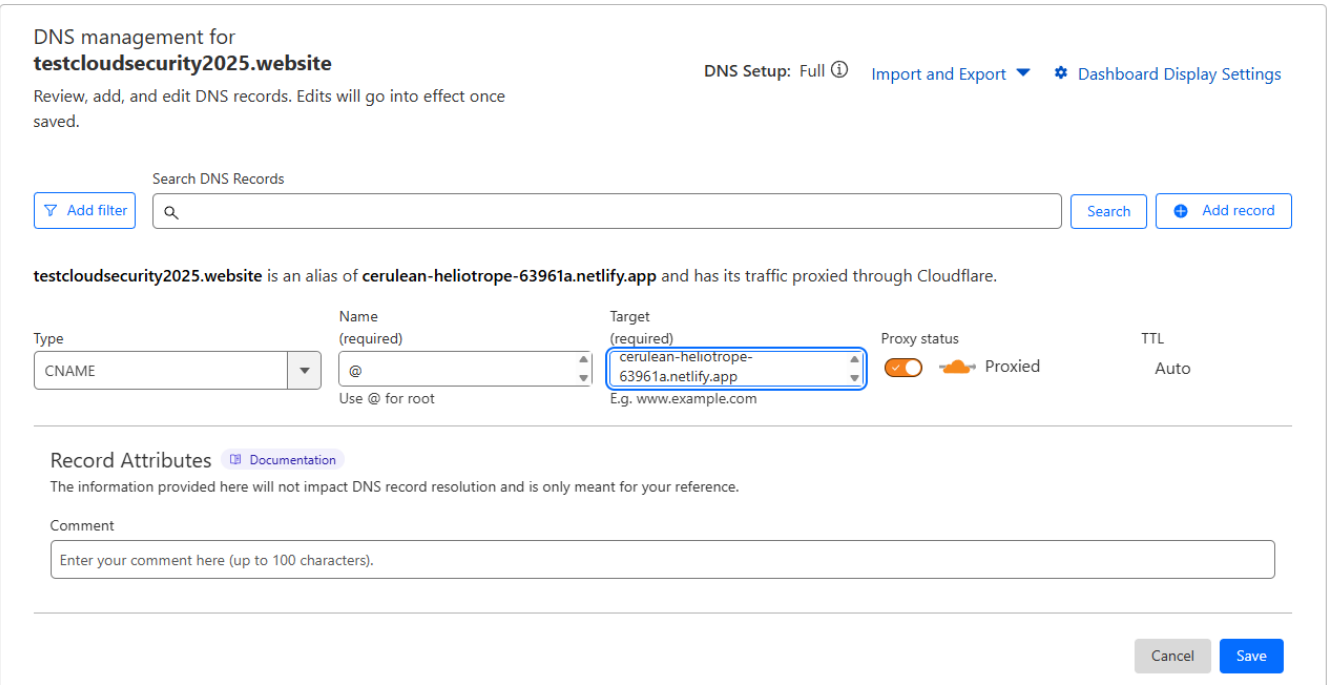


Рисунок 3.6 – Налаштування CNAME-запису у Cloudflare

У панелі Cloudflare було активовано наступні функції:

– *у SSL/TLS увімкнено режим Full* (рис. 3.7), що дозволяє шифрувати з'єднання між клієнтом і Cloudflare, якщо хостинг самостійно надає SSL-сертифікат;

– *Security Level.* Cloudflare автоматично застосовує оптимальний рівень безпеки, на даний момент ручне налаштування недоступне (рис. 3.8);

– *Bot Fight Mode.* Даний режим дозволяє Cloudflare самостійно виявляти та блокувати небажаних ботів (рис. 3.9);

– *у Caching активовано стандартне кешування*, що підвищує швидкодію сайту та знижує навантаження на Netlify (рис. 3.10).

Усі налаштування доступні у відповідних вкладках панелі керування Cloudflare та не вимагають додаткових технічних знань, що робить платформу зручною для користувачів-початківців.

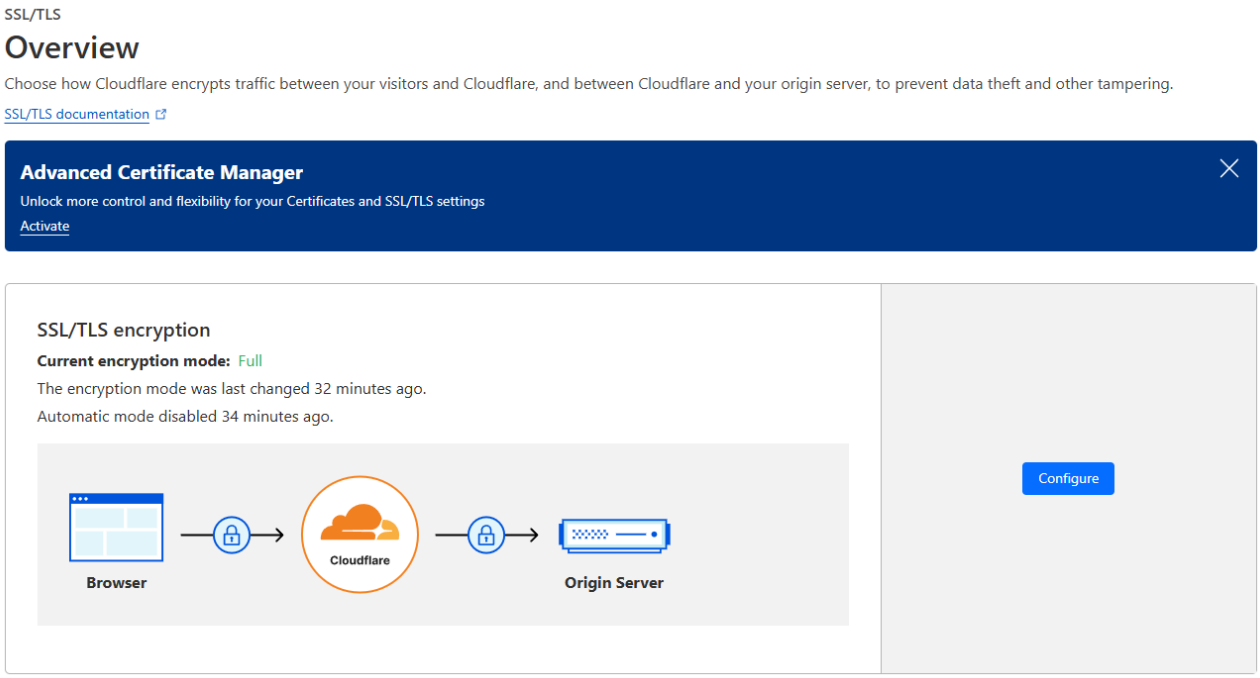


Рисунок 3.7 – Встановлений режим шифрування – “Full”

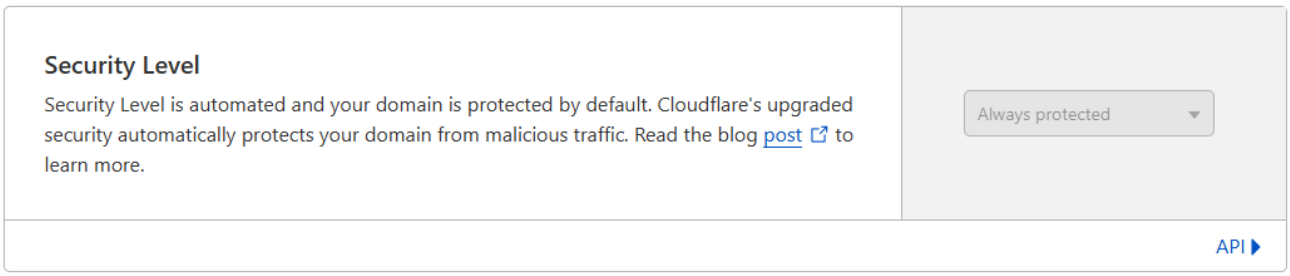


Рисунок 3.8 – Рівень захисту задано автоматично

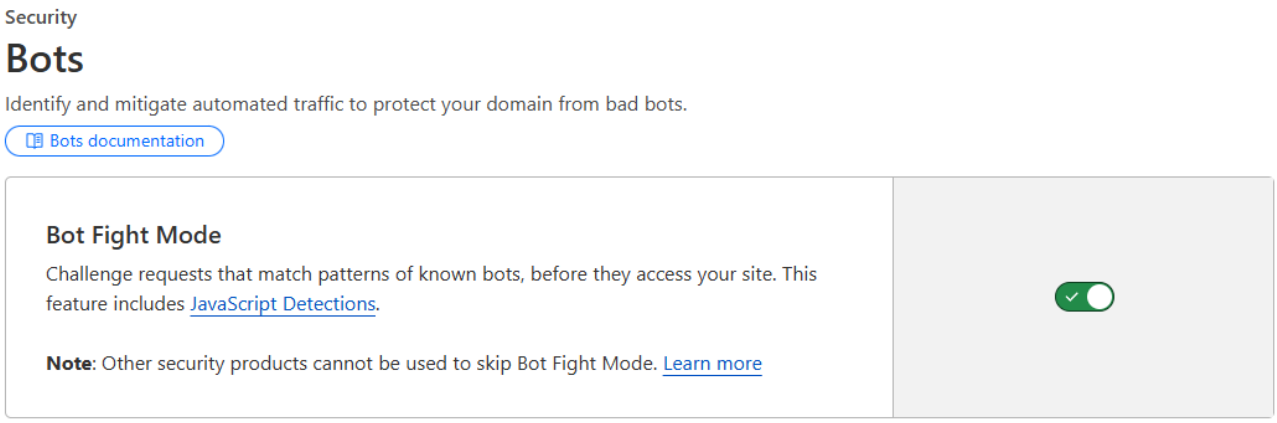


Рисунок 3.9 – Захист від ботів активовано

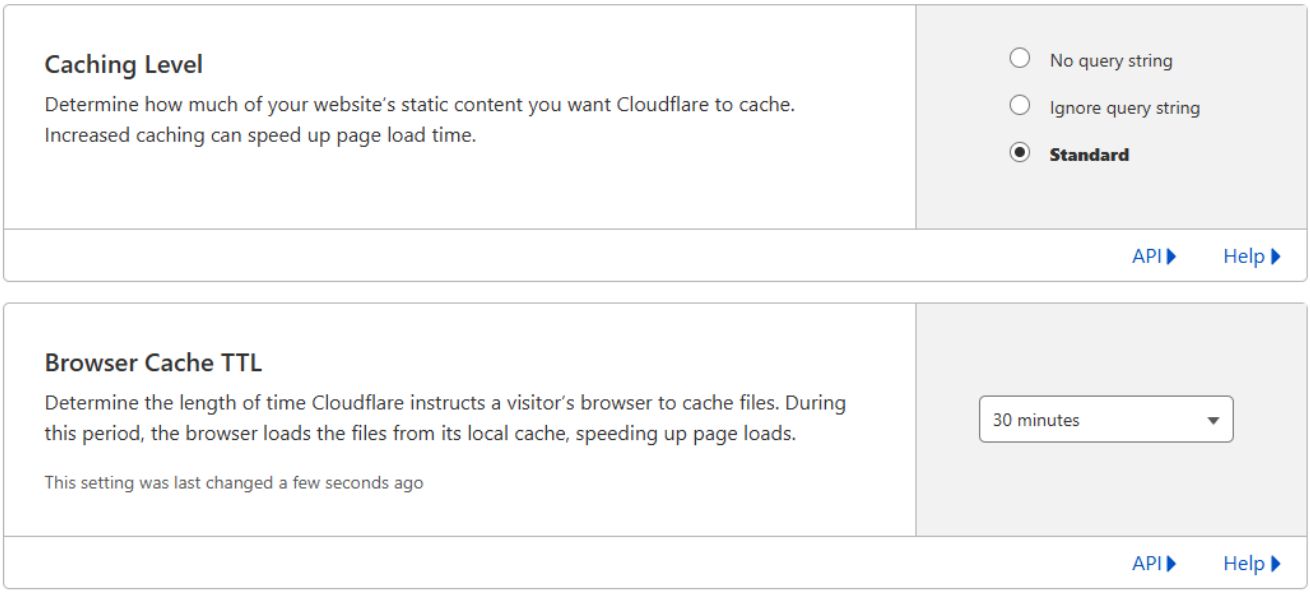


Рисунок 3.10 – Увімкнене кешування

Цей етап забезпечив базовий захист для нашого вебсайту, необхідний для тестування функціоналу Cloudflare на реальному кейсі.

3.3 Імітація загроз та аналіз захисту

З метою перевірки ефективності застосованих захисних механізмів Cloudflare було проведено експериментальну імітацію умовної атаки на вебресурс, розміщений за доменом testcloudsecurity2025.website. Основною метою даного етапу є перевірка здатності системи до виявлення та нейтралізації потенційно шкідливого трафіку, а також аналіз відповідних аналітичних даних у Cloudflare Dashboard.

Apache Benchmark – це командна утиліта, що входить до складу HTTP-сервера Apache і призначена для тестування продуктивності вебсервера шляхом створення множинних HTTP-запитів за короткий проміжок часу [25].

У контексті даного дослідження Apache Benchmark використовувався для симуляції великої кількості паралельних запитів до вебресурсу, що дозволило змоделювати умови низькоінтенсивної атаки. Таке навантаження дало змогу перевірити, як сервіс Cloudflare обробляє аномальний трафік, чи відбувається автоматичне блокування чи обмеження запитів, і наскільки ефективно працюють механізми фільтрації.

Проведення умовної атаки. Для генерації надмірної кількості HTTP-запитів було використано утиліту Apache Benchmark, яка має змогу згенерувати задану кількість запитів на сервер. Запити можуть оброблятися паралельно, можлива кількість і швидкість залежить від потужності центрального процесора. Для імітації масового звернення до вебресурсу у даному прикладі було введено команду, яка відправлятиме 10000 запитів, по 100 одночасно [26]. Приклад команди можна побачити на рисунку 3.11. Ця команда дозволила сформувати штучне навантаження на вебресурс, що має на меті викликати спрацювання захисних механізмів Cloudflare, зокрема фільтрації запитів або тимчасового блокування підозрілої активності [27].

```
PS C:\Users\Anastasiia\Apache24\bin> ./ab.exe -n 10000 -c 100 http://testcloudsecurity2025.website/
This is ApacheBench, Version 2.3 <$Revision: 1923142 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking testcloudsecurity2025.website (be patient)
Completed 1000 requests
Completed 2000 requests
Completed 3000 requests
Completed 4000 requests
```

Рисунок 3.11 – Команда для створення запиту, процес її виконання

Таким чином, Apache Benchmark виступив зручним інструментом для імітації високочастотного доступу до ресурсу, що є типовим прикладом навантаження, яке може використовуватися під час атаки, і дозволив адекватно перевірити реакцію захисної системи Cloudflare [28].

Спостереження за реакцією Cloudflare. У результаті моделювання умовної атаки, через декілька хвилин на Cloudflare Dashboard почала фіксуватись активність, що відповідає параметрам аномального трафіку. Автоматизовані системи Cloudflare ідентифікували велику кількість однотипних запитів з однієї IP-адреси та відповідно застосували захисні міри у вигляді блокування цих запитів (рис. 3.12).

Action taken	Country	IP address	Service
Block	Ukraine	92.249.[redacted]	HTTP DDoS
Block	Ukraine	92.249.[redacted]	HTTP DDoS
Block	Ukraine	92.249.[redacted]	HTTP DDoS
Block	Ukraine	92.249.[redacted]	HTTP DDoS
Block	Ukraine	92.249.[redacted]	HTTP DDoS
Block	Ukraine	92.249.[redacted]	HTTP DDoS
Block	Ukraine	92.249.[redacted]	HTTP DDoS
Block	Ukraine	92.249.[redacted]	HTTP DDoS
Block	Ukraine	92.249.[redacted]	HTTP DDoS
Block	Ukraine	92.249.[redacted]	HTTP DDoS
Block	Ukraine	92.249.[redacted]	HTTP DDoS
Block	Ukraine	92.249.[redacted]	HTTP DDoS
Block	Ukraine	92.249.[redacted]	HTTP DDoS
Block	Ukraine	92.249.[redacted]	HTTP DDoS

Рисунок 3.12 – Зафіксована і заблокована DDoS атака в дашборді Cloudflare

Також після запуску сервера спостерігалася підозріла активність ботів, які автоматично відслідкували реєстрацію нового домену, і робили до нього запити з різних регіонів. Така ситуація нерідко відбувається при реєстрації нового домену, тому в певний період навіть без спеціально згенерованої атаки на тестовому вебресурсі спостерігався високий трафік (рис. 3.13). Cloudflare, як і очікувалося, ідентифікував цей трафік, зафіксував його в дашборді і навіть заблокував один запит з Нідерландів, що є достатньо показовим прикладом якісної роботи даного сервісу (рис. 3.14).

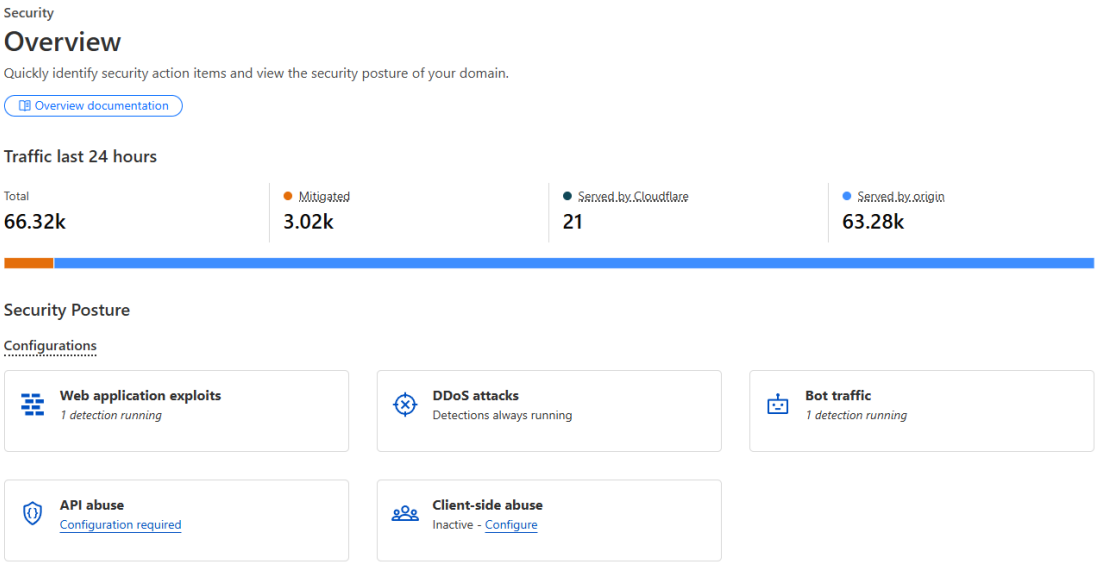


Рисунок 3.13 – Зафіксований трафік вебресурсу

Export | Edit columns

Action taken	Country	IP address	Service
Managed Challenge	United States	35.87.187.184	Bot fight mode
Managed Challenge	United States	44.243.136.32	Bot fight mode
Managed Challenge	United States	34.228.148.16	Bot fight mode
Managed Challenge	United States	54.242.114.111	Bot fight mode
Managed Challenge	United States	34.200.164.226	Bot fight mode
Block	Netherlands	109.202.99.36	Managed rules
Managed Challenge	United Kingdom	198.244.164.27	Bot fight mode
Managed Challenge	United States	35.94.145.195	Bot fight mode
Managed Challenge	United States	35.94.145.195	Bot fight mode

Рисунок 3.14 – Підозріла активність, зафіксована сервісом Cloudflare

Ці дані підтверджують, що Cloudflare ефективно виявив аномалії у трафіку та автоматично вжив заходів для їх нейтралізації без втручання адміністратора.

Оцінка поведінки сайту. Під час проведення атаки ресурс залишався доступним для легітимних користувачів. Спостерігались мінімальні затримки в завантаженні сторінки, однак це не вплинуло на загальну працездатність сайту. Після завершення імітації навантаження рівень запитів повернувся до нормального, що підтвердило стабільність функціонування захищеного ресурсу.

3.4 Оцінка результатів

У ході реалізації заходів із підключення вебресурсу до хмарного сервісу Cloudflare було проведено аналіз користувацького досвіду, ефективності базового рівня захисту та потенційних напрямів подальшого вдосконалення безпеки.

Платформа Cloudflare продемонструвала високий рівень зручності при початковому налаштуванні. Процес додавання домену, зміни NS-записів та верифікації ресурсу виявився інтуїтивно зрозумілим і добре задокументованим. Навіть у межах безкоштовного тарифного плану користувач має доступ до базових інструментів захисту, таких як автоматична фільтрація шкідливого трафіку, перевірка запитів, блокування потенційно небезпечних IP-адрес та перегляд статистики безпеки в режимі реального часу.

Імітація атаки з використанням утиліти Apache Benchmark, яка створювала велику кількість HTTP-запитів до ресурсу, дозволила оцінити реакцію системи захисту Cloudflare. В аналітичному розділі панелі керування було зафіксовано аномальне зростання кількості запитів, частина з яких була автоматично обмежена або заблокована. Це свідчить про ефективність реалізованих механізмів захисту навіть без застосування розширених (платних) функцій.

Окрім базових функцій, Cloudflare пропонує розширені можливості захисту в межах платних тарифних планів, зокрема:

- налаштування власних WAF-правил (Web Application Firewall);
- доступ до бот-менеджменту та машинного навчання для розпізнавання загроз;
- пріоритетну підтримку та глибшу аналітику в режимі реального часу [29].

Ці функціональні можливості можуть бути використані для подальших досліджень у напрямі поглибленого аналізу кіберзагроз, а також для забезпечення захисту вебресурсів, що піддаються більш складним або цільовим атакам [30].

У підсумку, використання Cloudflare як рішення для захисту вебресурсів у хмарному середовищі можна вважати доцільним і ефективним навіть на безкоштовному рівні. Платформа забезпечує надійну базову безпеку та пропонує широкий спектр засобів для масштабування захисту відповідно до потреб користувача.

ВИСНОВКИ

Проведене дослідження підтвердило ключову роль хмарних сервісів безпеки, зокрема Cloudflare та AWS Shield, у забезпеченні захисту вебресурсів в умовах зростання кіберзагроз. Аналіз продемонстрував зручність використання Cloudflare, ефективність його базового захисту та значний потенціал для масштабування безпеки.

Порівняння Cloudflare з AWS Shield виявило їхні відмінні характеристики, підкреслюючи важливість вибору сервісу, що найкраще відповідає індивідуальним потребам. Cloudflare, завдяки своєму широкому набору функцій, включаючи CDN та оптимізацію, а також гнучкій моделі ціноутворення, виглядає особливо перспективним для широкого кола користувачів.

Зважаючи на постійний розвиток кіберзагроз та зростаючу залежність бізнесу від онлайн-ресурсів, перспективи використання хмарних сервісів безпеки є надзвичайно високими. Очікується подальше вдосконалення їхніх можливостей, зокрема в галузі штучного інтелекту та машинного навчання для більш ефективного виявлення та нейтралізації складних атак. Cloudflare, з його активним розвитком технологій безпеки та глобальною інфраструктурою, має значний потенціал для подальшого зміцнення своїх позицій як провідного рішення у сфері захисту вебресурсів. Його здатність забезпечувати як базовий, так і просунутий захист робить його стратегічно важливим інструментом для забезпечення стабільності та безпеки онлайн-бізнесу в майбутньому.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1) Що таке сервер? Визначення, типи та особливості | ITEDU. IT Education Center Blog. URL: https://itedu.center/ua/blog/comparisons/types_of_servers/ (дата звернення: 01.03.2025).

2) Tomar A. JSP and Servlet in Java. Medium. URL: https://medium.com/nerd-for-tech/jsp-and-servlet-in-java-9981759972f0 (дата звернення: 01.03.2025).

3) Що таке атаки на стороні сервера - Терміни та Визначення у Кібербезпеці. VPN Unlimited - Fast & Secure VPN service. URL: https://www.vpnunlimited.com/ua/help/cybersecurity/server-side-attacks?srsltid=AfmBOoq2X0OgPCrtkiD7crpW3ghaXDw8Dsf6WhgCUuNtdzmYl6fB5pxk (дата звернення: 02.03.2025).

4) DSpace :: ELAKPI :: Репозитарій КПІ ім. Ігоря Сікорського. URL: https://ela.kpi.ua/server/api/core/bitstreams/cb76a75f-6e90-4066-abad-7aa320f39a52/content (дата звернення: 02.03.2025).

5) URL: https://www.imperva.com/learn/application-security/buffer-overflow/ (дата звернення: 03.03.2025).

6) What Is server security? {+ checklist}. Knowledge Base by phoenixNAP. URL: https://phoenixnap.com/kb/server-security (дата звернення: 03.03.2025).

7) Брандмауер (FireWall) – Вікі ЦДУ. Вікі ЦДУ. URL: https://wiki.cusu.edu.ua/index.php/Брандмауер_(FireWall) (дата звернення: 03.03.2025).

8) SSH | SSH Keys | What are SSH keys? | Secure Shell. AppViewX. URL: https://www.appviewx.com/education-center/what-are-ssh-keys/ (дата звернення: 03.03.2025).

9) Підвищення безпеки сервера: що зробити насамперед - TutHost.ua. TutHost.ua. URL: https://tuthost.ua/uk/blog/povyshenie-bezopasnosti-servera-chto-sdelat-v-pervuyu-ochered/#защита-ssh-авторизация-по-ключам (дата звернення: 06.03.2025).

10) Що таке логи та як з ними працювати. Онлайн-курси від компанії QATestLab | Головна сторінка. URL: https://training.qatestlab.com/blog/technical-articles/logging-in-concepts-requirements-levels/ (дата звернення: 07.03.2025).

11) Що таке IPS/IDS і де застосовується - Блог - HostZealot. HostZealot. URL: https://www.hostzealot.com.ua/blog/about-solutions/shho-take-ipsids-i-de-zastosovujetsya (дата звернення: 07.03.2025).

12) Що таке безпека в хмарі? | Захисний комплекс Microsoft. Your request has been blocked. This could be due to several reasons. URL: https://www.microsoft.com/uk-ua/security/business/security-101/what-is-cloud-security (дата звернення: 07.03.2025).

13) AWS WAF vs. Cloudflare. Minuto da Segurança da Informação. URL: https://minutodaseguranca.blog.br/aws-waf-vs-cloudflare/ (дата звернення: 09.03.2025).

14) AWS WAF vs. Cloudflare | Indusface Blog. Indusface. URL: https://www.indusface.com/blog/aws-waf-vs-cloudflare/ (дата звернення: 09.03.2025).

15) Our Plans | Pricing. Connect, protect, and build everywhere | Cloudflare. URL: https://www.cloudflare.com/plans/ (дата звернення: 10.03.2025).

16) Free DDoS Protection From Cloudflare. Connect, protect, and build everywhere | Cloudflare. URL: https://www.cloudflare.com/application-services/products/ddos-for-web/ (дата звернення: 10.03.2025).

17) Features. Amazon Web Services, Inc. URL: https://aws.amazon.com/shield/features/?nc1=h_ls#topic-1 (дата звернення: 11.03.2025).

18) Waf. Amazon Web Services, Inc. URL: https://aws.amazon.com/waf/ (дата звернення: 11.03.2025).

19) AWS Shield Reviews, Competitors and Pricing. PeerSpot. URL: https://www.peerspot.com/products/aws-shield-reviews?hl=uk (дата звернення: 12.03.2025).

20) AWS Console - Signup. URL: https://portal.aws.amazon.com/billing/signup?nc2=h_ct&src=header_signup&redirect_url=https://aws.amazon.com/registration-confirmation (дата звернення: 13.03.2025).

21) Astral Web Inc. Cloudflare Introduction (Tutorial for Beginners), 2019. YouTube. URL: https://www.youtube.com/watch?v=7riFvux9Bw4 (дата звернення: 13.03.2025).

22) AWS Shield vs CloudFlare | What are the differences?. StackShare. URL: https://stackshare.io/stackups/aws-shield-vs-cloudflare (дата звернення: 13.03.2025).

23) AWS WAF vs. Cloudflare | Indusface Blog. Indusface. URL: https://www.indusface.com/blog/aws-waf-vs-cloudflare/ (дата звернення: 14.03.2025).

24) 9 Best DDoS Protection Service Providers in 2025. eSecurity Planet. URL: https://www.esecurityplanet.com/products/distributed-denial-of-service-ddos-protection-vendors/ (дата звернення: 14.03.2025).

25) How to use ApacheBench for web server performance testing | Datadog. Datadog. URL: https://www.datadoghq.com/blog/apachebench/ (дата звернення: 15.03.2025).

26) Apache Benchmark Tutorial. Tutorials on Technical and Non Technical Subjects. URL: https://www.tutorialspoint.com/apache_bench/index.htm (дата звернення: 15.03.2025).

27) CodingLikeDavid. Install Apache Benchmark Tool (AB) - Windows 10 - No commentary, 2021. YouTube. URL: https://www.youtube.com/watch?v=hUZso9TpEes (дата звернення: 15.03.2025).

28) Tony Teaches Tech. How to Use Apache Bench to Load Test Your Website (ab linux command tutorial), 2020. YouTube. URL: https://www.youtube.com/watch?v=2kpd7HceFz4 (дата звернення: 15.03.2025).

29) ab - Apache HTTP server benchmarking tool - Apache HTTP Server Version 2.4. Welcome! - The Apache HTTP Server Project. URL: https://httpd.apache.org/docs/2.4/programs/ab.html (дата звернення: 15.03.2025).

30) Benchmarking & load testing tools. Gist. URL: https://gist.github.com/aliesbelik/840eff7c5bc78a141eab8e36f2f4edf2 (дата звернення: 15.03.2025).

HTML-КОД ДЛЯ ДЕМОНСТРАЦІЙНОЇ ВЕБСТОРІНКИ

```
<!DOCTYPE html>
<html lang="uk">
<head>
   <meta charset="UTF-8">
   <title>Захищений сайт</title>
   <style>
      body {
         font-family: Arial, sans-serif;
         background-color: #f2f4f8;
         text-align: center;
         padding-top: 100px;
         color: #333;
      }
      h1 {
         font-size: 2em;
         color: #2c3e50;
      }
      p {
         font-size: 1.1em;
         margin-top: 20px;
      }
      .badge {
         display: inline-block;
         background-color: #4CAF50;
         color: white;
         padding: 10px 20px;
         border-radius: 20px;
         margin-top: 30px;
         font-weight: bold;
      }
   </style>
</head>
<body>
   <h1>Демонстраційна сторінка</h1>
   <p>Цей сайт використовується для демонстрації захисту за допомогою Cloudflare.</p>
   <div class="badge">Cloudflare Protected</div>
</body>
</html>
```

Додаток Б

ДЕМОНСТРАЦІЙНІ МАТЕРІАЛИ

(Презентація)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут Інформаційних технологій
Кафедра Комп'ютерної інженерії

КВАЛІФІКАЦІЙНА РОБОТА
НА ЗДОБУТТЯ ОСВІТНЬОГО СТУПЕНЯ БАКАЛАВРА

на тему: «Хмарні сервіси для кращого захисту серверів»

Виконала студентка групи КІД-41
Ємченко Анастасія Володимирівна

Керівник кваліфікаційної роботи:
Бученко Ігор Анатолійович, старший викладач кафедри КІ

Київ – 2025

Мета роботи – Порівняльний аналіз хмарних сервісів Cloudflare та AWS Shield за низкою технічних, функціональних та організаційних критеріїв з метою обґрунтування вибору найбільш доцільного рішення для захисту вебресурсу.

Об`єкт дослідження – Хмарні технології захисту вебресурсів.

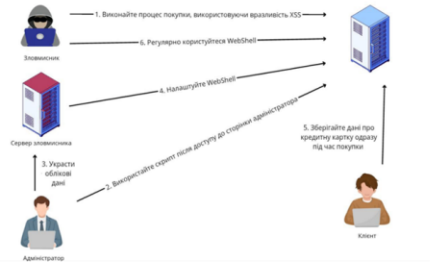
Предмет дослідження – Засоби забезпечення безпеки вебінфраструктури з використанням сучасних хмарних сервісів

Актуальність обраної теми

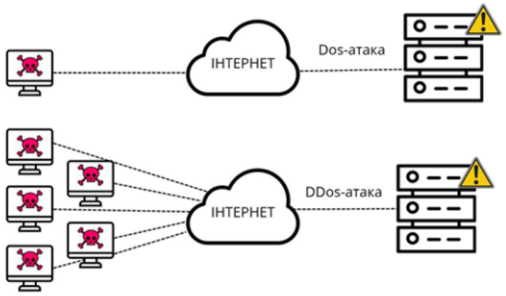
Актуальність теми обумовлена необхідністю здійснення комплексної порівняльної оцінки хмарних сервісів захисту від сучасних загроз для вибору оптимального рішення відповідно до умов конкретного застосування. Адже хибний вибір технологічного засобу безпеки може призвести до перевитрати ресурсів, неефективного або недостатнього захисту та надлишкової складності адміністрування.

Основні види атак на сервери

- **DoS** (Denial-of-Service) та **DDoS** (Distributed Denial of Service) атаки
- **SQL-ін'єкції** (неочікуваний SQL-синтаксис у запитах)
- **XSS-атаки** (Cross-Site Scripting)
- **CSRF-атаки** (Cross-Site Request Forgery)
- **Buffer Overflow** (переповнення буфера)



Принцип роботи XSS-атаки



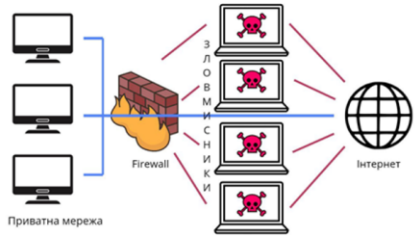
Різниця між DoS та DDoS-атакою



Схематичне зображення переповнення буфера

Методи захисту серверів

- **Фаєрвол** - фільтр трафіку між мережею і пристроєм. Види: програмні, мережеві, вебфаєрволи (WAF)
- **Захист SSH, авторизація за ключами** - захист від несанкціонованого доступу до сервера
- **Сегментація мережі** - розподіл мережі на ізольовані частини для підвищення безпеки.
- **Системи виявлення вторгнень (IDS/IPS)** - виявляють та зупиняють підозрілу активність у мережі.
- **Шифрування** - захист даних шляхом перетворення у нечитабельний вигляд.



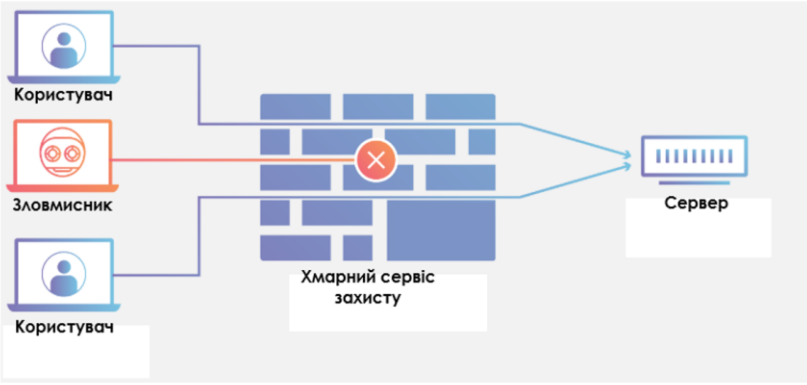
Мережа з використанням фаєрволу



Схема роботи SSH

Хмарні сервіси у забезпеченні безпеки серверів

- Розміщені в інтернеті сервіси, які обробляють і фільтрують трафік до серверів
- Забезпечують захист без складної інфраструктури
- Працюють на рівні DNS, мережі або додатків
- Автоматичне виявлення атак (DDoS, ботів тощо)
- Дозволяють швидке масштабування при навантаженні



Порівняння Cloudflare та AWS Shield

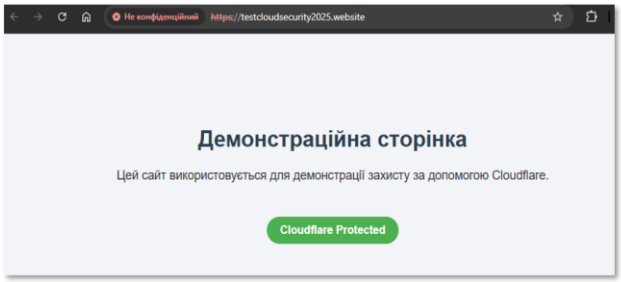
Параметр	Cloudflare	AWS Shield
Тип сервісу	Хмарна платформа для забезпечення безпеки, кешування, маршрутизації та CDN.	Хмарний сервіс для захисту від DDoS-атак у межах інфраструктури AWS.
Призначення	Захист вебресурсів, незалежно від хостингу.	Захист AWS-ресурсів, таких як EC2, CloudFront, ELB.
CDN	Так, з понад 300 дата-центрами по всьому світу.	Через AWS CloudFront (окремо).
DDoS-захист	Включено у всі тарифи. Захищає на рівнях L3, L4 та L7.	Standard — базовий захист (L3, L4). Advanced — повний захист (включно з L7 через WAF).
WAF (Web Application Firewall)	Так, доступний у платних тарифах.	Окремий сервіс AWS WAF, інтегрується з Shield Advanced.
Підтримка SSL/TLS	Безкоштовні сертифікати, підтримка шифрування end-to-end.	Через AWS ACM — оплачується окремо.
Bot Management	Вбудований механізм для виявлення та блокування ботів (в платних тарифах).	AWS Bot Control — як окремий сервіс.
Інтеграція з хостингом	Працює з будь-якими платформами через зміну DNS.	Тільки для ресурсів, що працюють в AWS.

Порівняння Cloudflare та AWS Shield

Параметр	Cloudflare	AWS Shield
Кешування та оптимізація	Потужні можливості кешування, оптимізація зображень та маршрутів (Argo).	Відсутнє. Використовується через сторонні AWS-сервіси.
Аналітика та логування	У базовому плані обмежено, в платному — повна аналітика, журналювання подій, інтеграція з SIEM.	Shield Advanced підтримує CloudWatch, VPC Flow Logs, GuardDuty.
Масштабованість	Автоматичне глобальне масштабування.	Автоматично масштабується в межах AWS.
Ціноутворення	Від безкоштовного до корпоративних тарифів із розширеним захистом.	Standard — безкоштовно, Advanced — від $3000/місяць.
SLA	100% для Enterprise тарифів.	99.99% у Shield Advanced.
Підтримка	Email-підтримка у безкоштовному, пріоритетна у платних тарифах.	Стандартна або розширена в Shield Advanced.
Зручність у використанні	Простий вебінтерфейс, швидке налаштування.	Потребує знань AWS-інфраструктури.
Переваги при виборі	Підтримка різних платформ, безкоштовна версія, простота у впровадженні, можливість кешування та edge-обчислень (Workers).	Глибока інтеграція з AWS, автоматична координація з іншими AWS-сервісами.

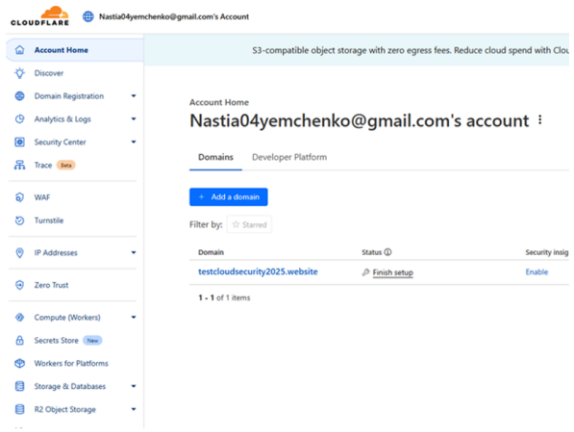
Розгортання тестового ресурсу

```html
<!DOCTYPE html>
<html lang="uk">
<head>
    <meta charset="UTF-8">
    <title>Захищений сайт</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f2f4f8;
            text-align: center;
            padding-top: 100px;
            color: #333;
        }
        h1 {
            font-size: 2em;
            color: #2c3e50;
        }
        p {
            font-size: 1.1em;
            margin-top: 20px;
        }
        .badge {
            display: inline-block;
            background-color: #4CAF50;
            color: white;
            padding: 10px 20px;
            border-radius: 20px;
            margin-top: 30px;
            font-weight: bold;
        }
    </style>
</head>
<body>
    <h1>Демонстраційна сторінка</h1>
    <p>Цей сайт використовується для демонстрації захисту за допомогою Cloudflare.</p>
    <div class="badge">Cloudflare Protected</div>
</body>
</html>
```

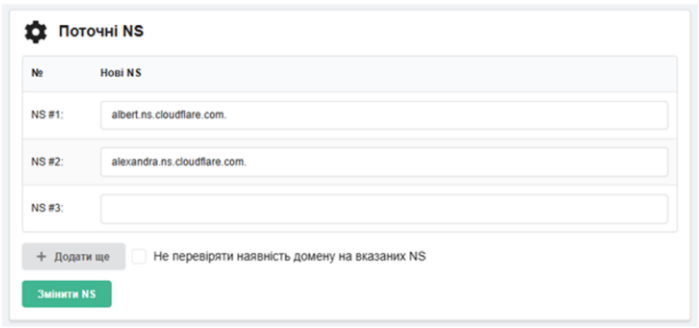


Тестова вебсторінка та код до неї

Підключення до Cloudflare



Прив'язка тестового вебресурсу до облікового запису Cloudflare



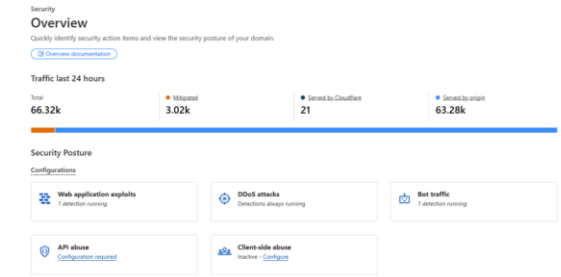
Нейм-сервери Cloudflare у налаштуванні DNS

Імітація атаки і результат

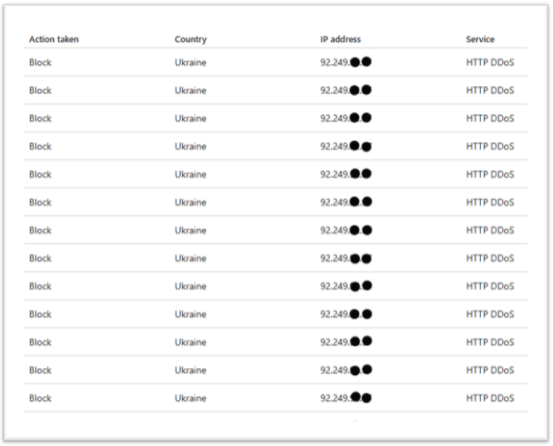
```
PS C:\Users\Anastasiia\Apache24\bin> ./ab.exe -n 10000 -c 100 http://testcloudsecurity2025.website/
This is ApacheBench, Version 2.3 <$Revision: 1923142 $>
Copyright 1996 Adam Twiss, Zeus Technology Ltd, http://www.zeustech.net/
Licensed to The Apache Software Foundation, http://www.apache.org/

Benchmarking testcloudsecurity2025.website (be patient)
Completed 1000 requests
Completed 2000 requests
Completed 3000 requests
```

Виконання команди для тестової атаки



Зафіксований трафік



Action taken	Country	IP address	Service
Block	Ukraine	92.249.●●	HTTP DDoS
Block	Ukraine	92.249.●●	HTTP DDoS
Block	Ukraine	92.249.●●	HTTP DDoS
Block	Ukraine	92.249.●●	HTTP DDoS
Block	Ukraine	92.249.●●	HTTP DDoS
Block	Ukraine	92.249.●●	HTTP DDoS
Block	Ukraine	92.249.●●	HTTP DDoS
Block	Ukraine	92.249.●●	HTTP DDoS
Block	Ukraine	92.249.●●	HTTP DDoS
Block	Ukraine	92.249.●●	HTTP DDoS
Block	Ukraine	92.249.●●	HTTP DDoS
Block	Ukraine	92.249.●●	HTTP DDoS
Block	Ukraine	92.249.●●	HTTP DDoS

Заблоковані підключення

Незапланована активність ботів

Export | Edit columns

Action taken	Country	IP address	Service
Managed Challenge	United States	35.87.187.184	Bot fight mode
Managed Challenge	United States	44.243.136.32	Bot fight mode
Managed Challenge	United States	34.228.148.16	Bot fight mode
Managed Challenge	United States	54.242.114.111	Bot fight mode
Managed Challenge	United States	34.200.164.226	Bot fight mode
Block	Netherlands	109.202.99.36	Managed rules
Managed Challenge	United Kingdom	198.244.164.27	Bot fight mode
Managed Challenge	United States	35.94.145.195	Bot fight mode
Managed Challenge	United States	35.94.145.195	Bot fight mode

Висновки

Проведене дослідження підтвердило ключову роль хмарних сервісів безпеки, зокрема Cloudflare та AWS Shield, у забезпеченні захисту вебресурсів в умовах зростання кіберзагроз. Аналіз продемонстрував зручність використання Cloudflare, ефективність його базового захисту та значний потенціал для масштабування безпеки.

Порівняння Cloudflare з AWS Shield виявило їхні відмінні характеристики, підкреслюючи важливість вибору сервісу, що найкраще відповідає індивідуальним потребам. Cloudflare, завдяки своєму широкому набору функцій, включаючи CDN та оптимізацію, а також гнучкій моделі ціноутворення, виглядає особливо перспективним для широкого кола користувачів.

Продовження висновків

Зважаючи на постійний розвиток кіберзагроз та зростаючу залежність бізнесу від онлайн-ресурсів, перспективи використання хмарних сервісів безпеки є надзвичайно високими. Очікується подальше вдосконалення їхніх можливостей, зокрема в галузі штучного інтелекту та машинного навчання для більш ефективного виявлення та нейтралізації складних атак.

Cloudflare, з його активним розвитком технологій безпеки та глобальною інфраструктурою, має значний потенціал для подальшого зміцнення своїх позицій як провідного рішення у сфері захисту вебресурсів. Його здатність забезпечувати як базовий, так і просунутий захист робить його стратегічно важливим інструментом для забезпечення стабільності та безпеки онлайн-бізнесу в майбутньому.