Міністерство освіти і науки України
Тернопільський національний технічний університет імені Івана Пулюя
(повне найменування вищого навчального закладу)
Факультет комп'ютерно-інформаційних систем і програмної інженерії
(назва факультету )
Кафедра кібербезпеки
(повна назва кафедри)

# КВАЛІФІКАЦІЙНА РОБОТА

на здобуття освітнього ступеня

магістр
(освітній рівень)

на тему: "Аналіз та способи удосконалення захищеності веб−сайту My Health"

Виконав: студент (ка)
Спеціальності:
125 «Кібербезпека та захист інформації»
(шифр і назва напряму підготовки, спеціальності)
Комендат Андрій Русланович
підпис (прізвище та ініціали)

Керівник Лечаченко Т. А.
підпис (прізвище та ініціали)

Нормоконтроль Тимощук Д. І.
підпис (прізвище та ініціали)

Завідувач кафедри Загородна Н.В.
підпис (прізвище та ініціали)

Рецензент Лещишин Ю.З
підпис (прізвище та ініціали)

м. Тернопіль – 2024

Міністерство освіти і науки України
**Тернопільський національний технічний університет імені Івана Пулюя**

Факультет комп'ютерно-інформаційних систем і програмної інженерії
(повна назва факультету)

Кафедра Кібербезпеки
(повна назва кафедри)

ЗАТВЕРДЖУЮ
Завідувач кафедри
_____ Загородна Н.В.
(підпис) (прізвище та ініціали)
«__» _____ 2024 р.

# ЗАВДАННЯ
## НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття освітнього ступеня Магістр
(назва освітнього ступеня)

за спеціальністю 125 Кібербезпека та захист інформації
(шифр і назва спеціальності)

Студенту Комендату Андрію Руслановичу
(прізвище, ім'я, по батькові)

1. Тема роботи Аналіз та способи удосконалення захищеності веб−сайту My Health

Керівник роботи Лечаченко Тарас Анатолійович, доктор філософії., старший викладач кафедри КБ
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

Затверджені наказом ректора від «20» 11 2024 року № 4/7-1112

2. Термін подання студентом завершеної роботи 16.12.2024

3. Вихідні дані до роботи Visual Studio Code, Postman, Zed Attack Proxy

4. Зміст роботи (перелік питань, які потрібно розробити)
Перелік умовних позначень, символів, одиниць, скорочень і термінів, Вступ. Розділ 1.Аналіз предметної область. 1.1 Інформаційні технології аналізу веб-сайтів. 1.2 Аналіз сучасних підходів до забезпечення безпеки веб-сайтів. 1.3 Основні загрози та проблеми безпеки веб-сайтів. Розділ 2 Основи теорії та методи досліджень. 2.1 Методи і технології кіберзахисту веб-ресурсів. 2.2 Обґрунтування використовуваних технологій. 2.3 Інтеграція методів захисту у веб-ресурсах. Розділ 3. Аналіз та оцінка результатів. 3.1 Результати аудиту безпеки веб-сайту. 3.2 Реалізація додаткового сервера. 3.3 Впровадження багатофакторної автентифікації як засобу посилення безпеки. 3.4 Впровадження системи WAF.3.5 Тестування та оцінка ефективності запропонованих рішень.3.6 Обґрунтування доцільності впроваджених засобів. Розділ 4. Охорона праці та безпека в надзвичайних ситуаціях. 4.1 Охорона праці. 4.2 Безпека в надзвичайних ситуаціях. Висновки. Список використаних джерел. Додатки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень, слайдів)
1. Титулка. 2. Мета кваліфікаційної роботи. 3. Актуальність теми. 4. Основні загрози. 5 Проблеми безпеки веб-сайтів. 6. Огляд веб-сайту My Health. 7. Проведення аудиту безпеки 8. Таблиця із результатами аудиту. 9 Впровадження додаткового сервера. 10 Впровадження багатофакторної автентифікації. 11 Впровадження системи WAF. 12. Тестування багатофакторної автентифікації 13. Тестування WAF. 14. Подальші покращення. 15. Висновок

6. Консультанти розділів роботи

| Розділ | Прізвище, ініціали та посада консультанта | Підпис, дата | |
|---|---|---|---|
| | | завдання видав | завдання прийняв |
| Охорона праці | Осухівська Г.М., к.т.н., доцент | | |
| Безпека в надзвичайних ситуаціях | Клепчик В.М., проректор з адміністративно-господарської роботи та будівництва | | |

7. Дата видачі завдання 20.09.2024 р.

КАЛЕНДАРНИЙ ПЛАН

| № з/п | Назва етапів роботи | Термін виконання етапів роботи | Примітка |
|---|---|---|---|
| 1. | Ознайомлення з завданням до кваліфікаційної роботи | 20.09 – 22.09 | Виконано |
| 2. | Підбір джерел про інформаційну безпеку | 23.09 – 01.10 | Виконано |
| 3. | Опрацювання джерел про інформаційну безпеку | 02.10 – 18.10 | Виконано |
| 4. | Пошук джерел про захист веб-сайтів | 19.10 – 03.11 | Виконано |
| 5. | Опрацювання джерел про захист веб-сайтів | 04.11 – 22.11 | Виконано |
| 6. | Аналіз веб-сайта «My Health» | 23.11 – 01.12 | Виконано |
| 7. | Оформлення першого розділу | 02.12 – 05.12 | Виконано |
| 8. | Оформлення другого розділу | 06.12 – 08.12 | Виконано |
| 9. | Оформлення третього розділу | 09.12 – 12.12 | Виконано |
| 10. | Оформлення розділу «Охорона праці та безпека в надзвичайних ситуаціях» | 13.12 – 14.12 | Виконано |
| 11. | Оформлення кваліфікаційної роботи | 15.12.2024 | Виконано |
| 12. | Нормоконтроль | 16.12.2024 | |
| 13. | Перевірка на плагіат | 19.12.2024 | |
| 14. | Попередній захист кваліфікаційної роботи | | |
| 15. | Захист кваліфікаційної роботи | 23.12.2024 | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Студент _____ Комендат А.Р.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Лечаченко Т. А.
(підпис) (прізвище та ініціали)

# АНОТАЦІЯ

Аналіз та способи удосконалення захищеності веб−сайту My Health // ОР «Магістр» // Комендат Андрій Русланович // Тернопільський національний технічний університет імені Івана Пулюя, факультет комп'ютерно-інформаційних систем і програмної інженерії, кафедра кібербезпеки, група СБм-61 // Тернопіль, 2024 // С. 83, рис. – 27, табл. – 1 , кресл. – , додат. – 5.

Ключові слова: кібербезпека, захист сайту, WAF, MFA, аналіз сайту, веб-вразливість.

Мета роботи – аналіз та удосконалення захищеності веб-сайту My Health, зосереджуючи увагу на виявленні вразливостей та впровадженні сучасних засобів захисту, таких як багатофакторна автентифікація (MFA), система веб-аплікаційного файрволу (WAF) та додатковий сервер для безпечного зберігання даних.

У першому розділі проведено дослідження сучасних інформаційних технологій аналізу веб-сайтів. Розглянуто класифікацію методів аналізу, включаючи статичний, динамічний та інтерактивний підходи, що дозволяють оцінити структуру, функціональність та безпеку веб-ресурсів.

Другий розділ присвячено аналізу методів і технологій кіберзахисту веб-ресурсів. Зокрема, висвітлено впровадження багатофакторної автентифікації (MFA), її переваги та методи інтеграції для захисту облікових записів користувачів. Детально розглянуто використання систем WAF для блокування атак, таких як SQL-ін'єкції та DDoS.

У третьому розділі описано результати аудиту безпеки веб-сайту, виконаного за допомогою інструментів динамічного аналізу. Розглянуто впровадження додаткового сервера для сегментації мережі та зниження ризиків компрометації даних. Результатом роботи є комплекс удосконалень, що забезпечують покращення захищеності веб-сайта My Health.

# ABSTRACT

Analysis and Improvement Methods for Securing the "My Health" Website // Thesis of educational level "Master"// Komendat Andrii // Ternopil Ivan Puluj National Technical University, Faculty of Computer Information Systems and Software Engineering, Department of Cybersecurity, group СБм-61 // Ternopil, 2024 // P. 84, , tables 1, drawings 27, added. – 5.

Keywords: cyber security, website protection, WAF, MFA, website analysis, web vulnerability.

In the first chapter, a study of modern information technologies for website analysis is conducted. The classification of analysis methods is considered, including static, dynamic, and interactive approaches that enable the evaluation of the structure, functionality, and security of web resources.

The second chapter focuses on the analysis of methods and technologies for cybersecurity of web resources. Specifically, it highlights the implementation of multi-factor authentication (MFA), its advantages, and integration methods for protecting user accounts. The use of WAF systems to block attacks such as SQL injections and DDoS is examined in detail.

The third chapter describes the results of a website security audit conducted using dynamic analysis tools. The implementation of an additional server for network segmentation and risk mitigation of data compromise is discussed. The outcome of the work is a comprehensive set of improvements that enhance the security of the My Health website.

ЗМІСТ

# ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

| | | |
|---|---|---|
| WAF | — | Web Application Firewall |
| MFA | — | Secure Sockets Layer |
| XSS | — | Cross Site Scripting |
| MITM | — | Man in the middle |
| CSRF | — | Cross-site request forgery |
| DoS | — | denial-of-service attack |
| ZAP | — | Zed Attack Proxy |

# ВСТУП

Для виконання дипломної роботи обрано тему: «Аналіз та способи удосконалення захищеності веб-сайту My Health».

Актуальність теми. Сучасний розвиток цифрових технологій визначає напрями інновацій у різних сферах людської діяльності. Інформаційні системи та веб-додатки стають невід'ємною частиною процесів обробки, аналізу та передачі даних, що є важливим для вирішення завдань сучасного суспільства. Однією з ключових проблем залишається забезпечення інформаційної безпеки таких систем, особливо в умовах зростання кіберзагроз та ускладнення атак на інфраструктуру.

Мета і задачі дослідження. Метою дослідження є аналіз методів і засобів захисту веб-додатків, спрямованих на виявлення, аналіз та усунення вразливостей

Дослідження спрямоване на визначення оптимальних рішень для інтеграції механізмів безпеки в архітектуру веб-додатків, що дозволить забезпечити їхню стійкість до сучасних атак.

Завданнями дослідження є проведення аналізу сучасних кіберзагроз та їх впливу на веб-додатки, що працюють із даними, а також дослідження існуючих підходів до забезпечення інформаційної безпеки. У процесі дослідження необхідно визначити типові вразливості веб-додатків, розробити методику їх виявлення та усунення, а також запропонувати ефективні засоби для захисту даних. Важливою складовою завдання є тестування розроблених рішень у реальних або симуляційних умовах для оцінки їхньої ефективності, а також формування рекомендацій для покращення захищеності інформаційних систем.

Об’єкт дослідження – Інформаційні системи та веб-додатки, які працюють із даними.

Предмет дослідження – методи забезпечення інформаційної безпеки веб-додатків, спрямовані на виявлення та мінімізацію кіберзагроз.

Наукова новизна одержаних результатів кваліфікаційної роботи. Наукова новизна роботи полягає у розробці нових підходів до аналізу та усунення вразливостей у веб-додатках, що базуються на поєднанні сучасних методів виявлення кіберзагроз та інтеграції засобів захисту в архітектуру додатків. У процесі дослідження сформовано методику оцінки рівня захищеності веб-додатків із врахуванням сучасних технологій атак. Запропоновані підходи дозволяють не лише виявляти потенційні загрози, але й мінімізувати ризики їхнього виникнення, що сприяє підвищенню загальної інформаційної безпеки.

Практичне значення одержаних результатів. Практичне значення одержаних результатів полягає у можливості їх використання для створення та вдосконалення захищених веб-додатків, які працюють із конфіденційними даними. Розроблені методи та засоби можуть бути інтегровані в існуючі інформаційні системи з метою підвищення їхньої стійкості до атак. Також результати дослідження можуть бути корисними для навчання спеціалістів з кібербезпеки, адаптації інформаційних систем до нових умов і розробки рекомендацій для зменшення ризиків витоку або пошкодження даних.

Апробація результатів магістерської роботи. Основні результати проведених досліджень обговорювались на: XII науково-технічної конференції «Інформаційні моделі, системи та технології» (м.Тернопіль, Україна).

Публікації. Основні результати кваліфікаційної роботи опубліковано у працях конференції (див. Додаток А).

# РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

## 1.1 Інформаційні технології аналізу веб-сайтів

Сучасні інформаційні технології аналізу веб-сайтів відкривають широкі можливості для вивчення різних аспектів їхньої роботи. До них належать продуктивність, безпека, зручність використання, відповідність стандартам та інші вимоги. За допомогою інструментів веб-аналітики можна отримати докладні дані про поведінку користувачів, джерела трафіку та ефективність рекламних кампаній, а також виявити проблемні зони, що негативно впливають на роботу ресурсу Сучасні сервіси, як-от Google Analytics, відстежують поведінку відвідувачів та сегменти для персоналізованих маркетингових стратегій і ухвалення обґрунтованих рішень щодо поліпшення взаємодії з аудиторією.

Однією з основних сфер аналізу є безпека веб-сайтів, що особливо актуально в умовах зростання кіберзагроз. Використання спеціалізованих сканерів вразливостей, таких як OWASP ZAP, Nessus, Burp Suite, дозволяє ідентифікувати проблеми, включаючи SQL-ін’єкції, XSS-атаки, неправильну конфігурацію серверів та недостатній захист конфіденційних даних. Інтеграція цих інструментів із процесами DevOps забезпечує своєчасну перевірку безпеки на всіх етапах розробки, що значно знижує ризики порушення цілісності даних або несанкціонованого доступу.

Продуктивність сайту - важливий фактор, який безпосередньо впливає на користувацький досвід і результати SEO, а такі сервіси, як GTmetrix і Google PageSpeed Insights, мають можливість аналізувати швидкість завантаження сторінок і ефективність використання ресурсів, а також надавати рекомендації щодо оптимізації. Наприклад:

–Зменшення розміру зображень шляхом стиснення;

–Налаштування кешування для зменшення часу повторного завантаження;

–Оптимізація HTTP-запитів шляхом об'єднання файлів CSS/JS.

Ці дії сприяють швидкому завантаженню контенту, що важливо як для пошукових систем, так і для користувачів.

Оцінка користувацького досвіду (UX) використовує такі технології, як теплові карти (Hotjar, Crazy Egg), щоб відстежувати поведінку користувачів і виявляти проблеми навігації та взаємодії. Це дозволяє створити більш інтуїтивно зрозумілий і зручний інтерфейс для залучення вашої аудиторії.

Не менш важливим аспектом є забезпечення постійної доступності веб-ресурсу. Інструменти, такі як Pingdom та Uptime Robot, автоматично моніторять час безвідмовної роботи сайту, повідомляють про збої в реальному часі і надають аналітику щодо частоти виникнення проблем. Це допомагає оперативно усувати технічні несправності та мінімізувати вплив на відвідувачів.

Застосування сучасних інформаційних технологій для аналізу веб-сайтів дозволяє не лише забезпечити стабільну роботу ресурсу, але й підвищити ефективність бізнесу. Завдяки поєднанню інструментів аналітики, безпеки, оптимізації продуктивності та моніторингу власники веб-ресурсів отримують змогу:

–Швидко адаптувати сайти до змін ринку;

–Поліпшувати взаємодію з аудиторією;

–Зменшувати операційні ризики.

Таким чином, інтеграція сучасних технологій є не лише вимогою часу, але й конкурентною перевагою у цифрову епоху.

### 1.1.1 Класифікація методів аналізу веб-сайтів

Методи аналізу веб-сайтів можна класифікувати за кількома критеріями залежно від цілей дослідження, застосованих технологій і аспектів функціонування веб-ресурсів. Основні категорії методів аналізу включають:

–За об’єктом аналізу. Методи аналізу, що класифікуються за об’єктом, спрямовані на конкретні складові веб-сайту. Наприклад, продуктивність оцінюється через аналіз швидкості завантаження сторінок, оптимізації кодів та

зображень. Веб-сайти часто досліджуються на рівні серверної частини, клієнтської взаємодії, баз даних і мережевої інфраструктури. Безпека, як ключовий аспект, включає перевірку на вразливості, такі як SQL-ін’єкції або міжсайтовий скриптинг, що дозволяє виявити слабкі місця. Оцінка зручності користування допомагає зрозуміти, наскільки інтуїтивно користувачі можуть взаємодіяти із сайтом, тоді як SEO-аналіз забезпечує високу видимість у пошукових системах;

–За методами збору даних. Методи збору даних у процесі аналізу варіюються від ручного до автоматизованого підходу. Ручний аналіз передбачає глибоку індивідуальну оцінку функціональних та естетичних аспектів сайту, але потребує більше часу й кваліфікації. Автоматизований підхід базується на використанні спеціалізованих програмних засобів, які можуть швидко сканувати веб-сайт, збираючи великі обсяги даних для їх подальшої обробки. Комбінований метод дозволяє об’єднати переваги обох підходів, забезпечуючи всебічний аналіз із гнучкими налаштуваннями відповідно до цілей дослідження;

–За періодичністю проведення. За частотою аналіз може бути одноразовим, регулярним чи постійним. Одноразовий аналіз зазвичай проводиться під час розробки чи впровадження сайту, щоб оцінити його готовність до роботи. Регулярний аналіз дозволяє відстежувати зміни в роботі ресурсу, наприклад, щомісячно чи щоквартально, щоб вчасно виявляти та усувати проблеми. Постійний моніторинг, який забезпечують інструменти для реального часу, дозволяє негайно реагувати на збої чи атаки, що є особливо важливим для сайтів з високою критичністю функцій;

–За технічними засобами. Технічні засоби аналізу поділяються на локальні та хмарні. Локальні інструменти встановлюються безпосередньо на пристроях розробників чи аналітиків, що дозволяє працювати незалежно від інтернет-з’єднання, але вимагає ресурсів системи. Хмарні інструменти, які надають аналіз через веб-сервіси, є більш доступними й масштабованими, оскільки вони не залежать від потужності локального обладнання й можуть обробляти великі

обсяги даних одночасно. Ці інструменти також пропонують інтеграцію з іншими сервісами для оптимізації процесу аналізу;

–За спрямованістю. Аналіз веб-сайтів може бути діагностичним або прогностичним. Діагностичний підхід спрямований на виявлення існуючих проблем, таких як низька продуктивність, помилки в коді або слабкі місця в архітектурі сайту. Прогностичний аналіз фокусується на передбаченні можливих проблем, наприклад, оцінка навантаження, що дозволяє вчасно впровадити заходи для уникнення збоїв. Ці підходи часто комбінуються для забезпечення максимальної ефективності управління веб-ресурсом.

Аналіз веб-сайтів може бути діагностичним або прогностичним. Діагностичний підхід спрямований на виявлення існуючих проблем, таких як низька продуктивність, помилки в коді або слабкі місця в архітектурі сайту. Прогностичний аналіз фокусується на передбаченні можливих проблем, наприклад, оцінка навантаження, що дозволяє вчасно впровадити заходи для уникнення збоїв. Ці підходи часто комбінуються для забезпечення максимальної ефективності управління веб-ресурсом.

Класифікація методів аналізу веб-сайтів охоплює широкий спектр підходів, які дозволяють всебічно оцінити технічний стан, функціональність та ефективність веб-ресурсів. Завдяки поділу за об’єктом аналізу можна зосередитися на конкретних аспектах, таких як продуктивність або безпека. Розгляд методів збору даних, від ручних до автоматизованих, дозволяє адаптувати процес залежно від складності та обсягу роботи. Врахування періодичності проведення аналізу забезпечує можливість як одноразової оцінки, так і постійного моніторингу, що є критично важливим для динамічних систем.

## 1.2 Аналіз сучасних підходів до забезпечення безпеки веб-сайтів

Аналіз веб-сайту є багатогранним процесом, що вимагає комплексного підходу та врахування широкого спектра факторів. Для цього потрібно оцінити різні аспекти, такі як серверна частина, клієнтська взаємодія, бази даних,

мережеві інфраструктури та фізичні сервери, якщо вони використовуються. Оскільки веб-сайт є інтерактивним елементом, що активно взаємодіє з користувачами, важливо оцінювати його в контексті реальних умов функціонування, що включає рівень навантаження в пікові періоди, географічне розташування користувачів, а також типи пристроїв, через які здійснюється доступ до сайту. Кожен із цих аспектів може суттєво впливати на ефективність ресурсу, тому необхідно провести всебічний аналіз, щоб забезпечити стабільну роботу веб-сайту в усіх умовах.

Перш за все, важливою частиною аналізу є безпека веб-сайту. Враховуючи постійне зростання кіберзагроз, захист ресурсу від можливих атак має бути пріоритетом. Уразливості, які можуть бути виявлені на різних етапах життєвого циклу веб-сайту, створюють серйозні ризики. Тому перший етап безпекового аудиту полягає у виявленні можливих точок уразливості. Це може включати сканування сайту на наявність загроз, таких як SQL-ін’єкції, XSS-атаки, міжсайтова фальсифікація запитів (CSRF) та інші види атак, що можуть бути використані для несанкціонованого доступу до системи. Важливо також звертати увагу на вразливості в компонентному програмному забезпеченні, яке може використовуватися на веб-сайті, таких як старі версії CMS або бібліотек JavaScript, які не мають достатнього захисту від відомих загроз.

Для аудиту безпеки зазвичай використовуються спеціалізовані інструменти, такі як Acunetix, Nessus, OpenVAS або Burp Suite. Ці інструменти дозволяють не лише виявити існуючі уразливості, але й надати рекомендації щодо їх усунення. Наприклад, сканери вразливостей можуть перевірити, чи використовуються на сайті належні методи шифрування для зберігання паролів або чи має сайт захист від атак типу "відмова в обслуговуванні" (DoS). У свою чергу, Burp Suite є потужним інструментом для тестування безпеки веб-застосунків, який дозволяє проводити мануальні та автоматизовані атаки для перевірки міцності сайту до різних видів вторгнень.

Аудит безпеки не обмежується лише перевіркою наявних уразливостей. Потрібно також провести аналіз політики безпеки, яка охоплює збереження

конфіденційної інформації, механізми автентифікації користувачів, рівні доступу та захист від несанкціонованого доступу. Важливо перевірити, чи використовуються сучасні методи шифрування для захисту даних під час передачі (наприклад, протокол HTTPS), а також чи здійснюється регулярне оновлення програмного забезпечення, щоб уникнути використання відомих уразливостей. Окремо варто зазначити роль забезпечення безпеки на рівні хостингу та сервера. Невірні налаштування на рівні сервера, неправильна конфігурація файрволів або незахищені порти можуть стати серйозними ризиками для ресурсу.

Іншим важливим аспектом аналізу веб-сайту є продуктивність, яка безпосередньо впливає на користувацький досвід. Швидкість завантаження сторінок має вирішальне значення, і навіть затримка в кілька секунд може негативно вплинути на кількість відвідувачів, а отже, і на успіх веб-сайту. Неоптимізовані зображення, великі файли JavaScript і надмірні HTTP-запити можуть значно сповільнювати роботу сайту, збільшуючи час очікування користувачів і знижуючи коефіцієнт конверсії. Тому важливо використовувати такі інструменти, як Google PageSpeed Insights і GTmetrix, для аналізу часу завантаження сторінок, виявлення потенційних проблем і оптимізації швидкості.

Однак продуктивність веб-сайту не обмежується лише швидкістю завантаження сторінок. Важливо також оцінити ефективність роботи сервера, обсяг завантажуваних даних та використання кешування. Зазначені інструменти також надають рекомендації щодо зменшення розміру зображень, використання стиснення для файлів CSS і JavaScript, а також налаштування кешування для повторних відвідувань користувачами. Ця оптимізація значно покращує швидкість завантаження сторінок та знижує навантаження на сервер.

Однією важливою складовою є забезпечення доступності веб-сайту. Це поняття включає не лише технічну доступність сайту, тобто забезпечення його безперервної роботи, але й врахування потреб людей з обмеженими можливостями. Стандарти WCAG (Web Content Accessibility Guidelines) визначають вимоги до доступності контенту для людей з різними видами

інвалідності. Це можуть бути правила щодо використання контрастних кольорів для покращення читабельності, можливість навігації за допомогою клавіатури для користувачів, які не можуть використовувати мишу, або застосування альтернативних текстів для зображень, що дозволяє людям з обмеженим зором використовувати допоміжні технології для доступу до інформації на сайті.

Важливим є також вивчення користувацького досвіду, або UX. Структура сайту має бути інтуїтивно зрозумілою, щоб користувачі могли легко знаходити необхідну інформацію. Тут важливо оцінити, наскільки ефективно організована навігація по сайту, як зрозуміло представлені функції та чи зручний інтерфейс. Користувачі повинні мати можливість без зусиль взаємодіяти з елементами сайту та знаходити потрібну інформацію. Використання таких інструментів, як теплові карти (наприклад, Hotjar), дозволяє зрозуміти, як користувачі взаємодіють з елементами сторінки, і де вони стикаються з труднощами. Ці дані допомагають розробникам і дизайнерам оптимізувати інтерфейс для покращення користувацького досвіду.

Завершення аналізу веб-сайту має бути документоване у вигляді детального звіту, що містить рекомендації щодо покращення безпеки, продуктивності, доступності та користувацького досвіду. Ці рекомендації можуть стосуватися технічних аспектів, таких як оновлення програмного забезпечення, оптимізація ресурсів, покращення конфігурації сервера, або стосуватися змін у дизайні сайту, щоб зробити його більш зручним і доступним для користувачів. У підсумку, правильне проведення аудиту веб-сайту допомагає забезпечити стабільну роботу ресурсу, підвищити його ефективність та конкурентоспроможність на ринку, а також покращити взаємодію з користувачами.

### 1.2.1 Класифікація методів аналізу веб-сайтів

Аналіз безпеки веб-сайтів — це ключовий процес, який допомагає виявляти вразливості, захищати дані та забезпечувати стабільну роботу ресурсу. Для цього використовуються різноманітні спеціалізовані інструменти, а саме:

OWASP ZAP (Zed Attack Proxy) – є одним із найпопулярніших інструментів для тестування безпеки [7]. Це безкоштовне рішення з відкритим вихідним кодом, яке дозволяє виявляти такі вразливості, як:

–SQL-ін'єкції;

–міжсайтовий скриптинг (XSS);

–проблеми автентифікації та авторизації.

ZAP автоматизує процеси сканування та забезпечує зручний інтерфейс для тестувальників, незалежно від їхнього досвіду (див. рис. 1.1).

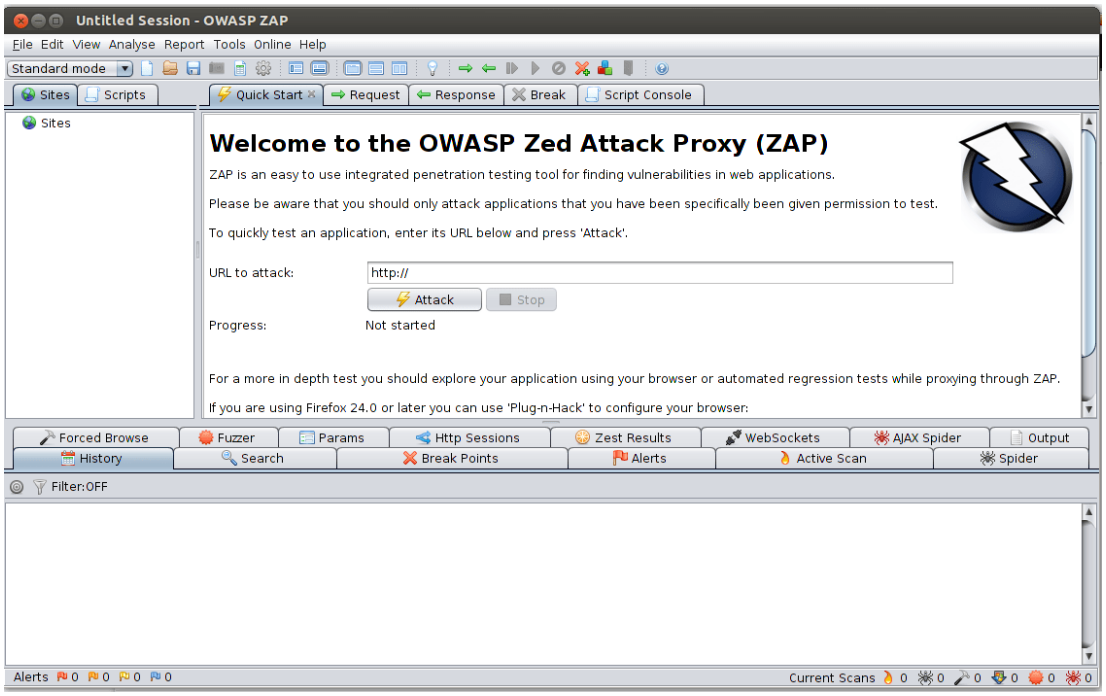



Рисунок 1.1 – OWASP ZAP

Burp Suite – професійний інструмент для перевірки безпеки веб-додатків, що використовується експертами в галузі кібербезпеки [8]. Його основні функції включають:

– перевірку сеансів користувачів;

– аналіз взаємодії між клієнтом і сервером;

– виявлення складних вразливостей.

Програма має платну та безкоштовну версії, що дозволяє обирати функціонал залежно від потреб(див. рис. 1.2).

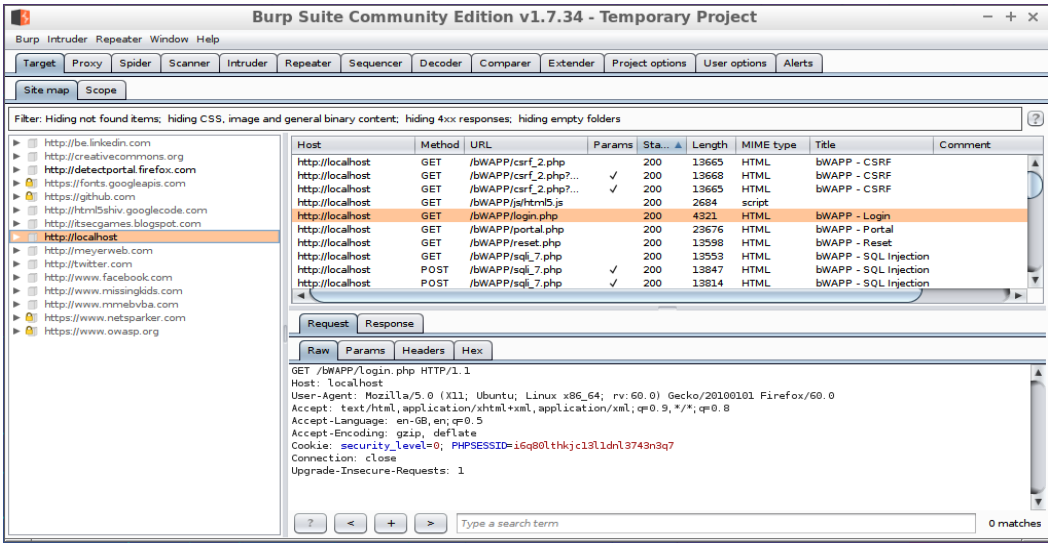



Рисунок 1.2 – Burp Suite

Acunetix — це потужний інструмент для автоматизованого сканування веб-додатків [9]. Він допомагає виявляти не тільки вразливості в програмному коді, але й слабкі місця в конфігурації серверів. Його перевагою є інтеграція з CI/CD-процесами, що дозволяє забезпечувати безпеку на всіх етапах розробки (див. рисунок 1.3).

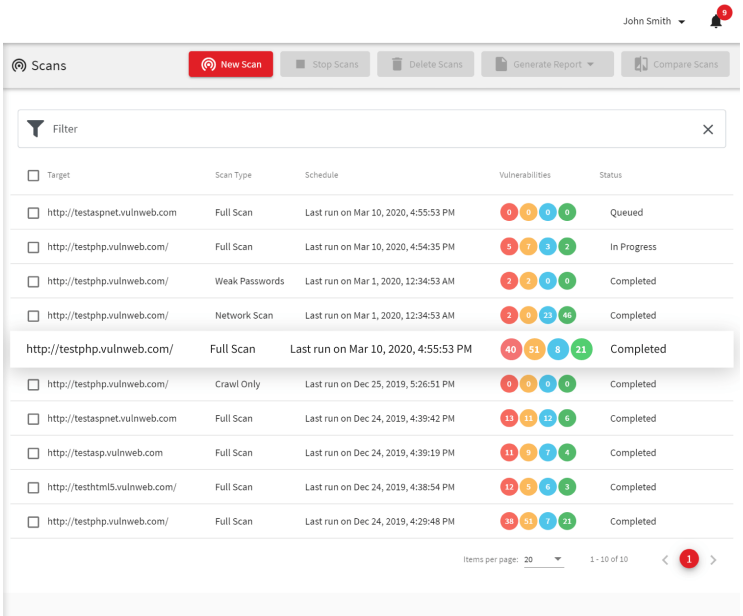



Рисунок 1.3 – Acunetix

Postman — це популярний інструмент для тестування API, який дозволяє автоматизувати запити, перевіряти відповіді сервера та аналізувати їхню поведінку. Він особливо корисний для тестування безпеки в контексті багатофакторної автентифікації (MFA) та інших механізмів авторизації [10]. Використовуючи Postman, тестувальники можуть надсилати запити на реєстрацію, входити в систему і тестувати функціональність токенів доступу, забезпечуючи комплексний підхід до аналізу безпеки веб-додатків (див. рис. 1.4).

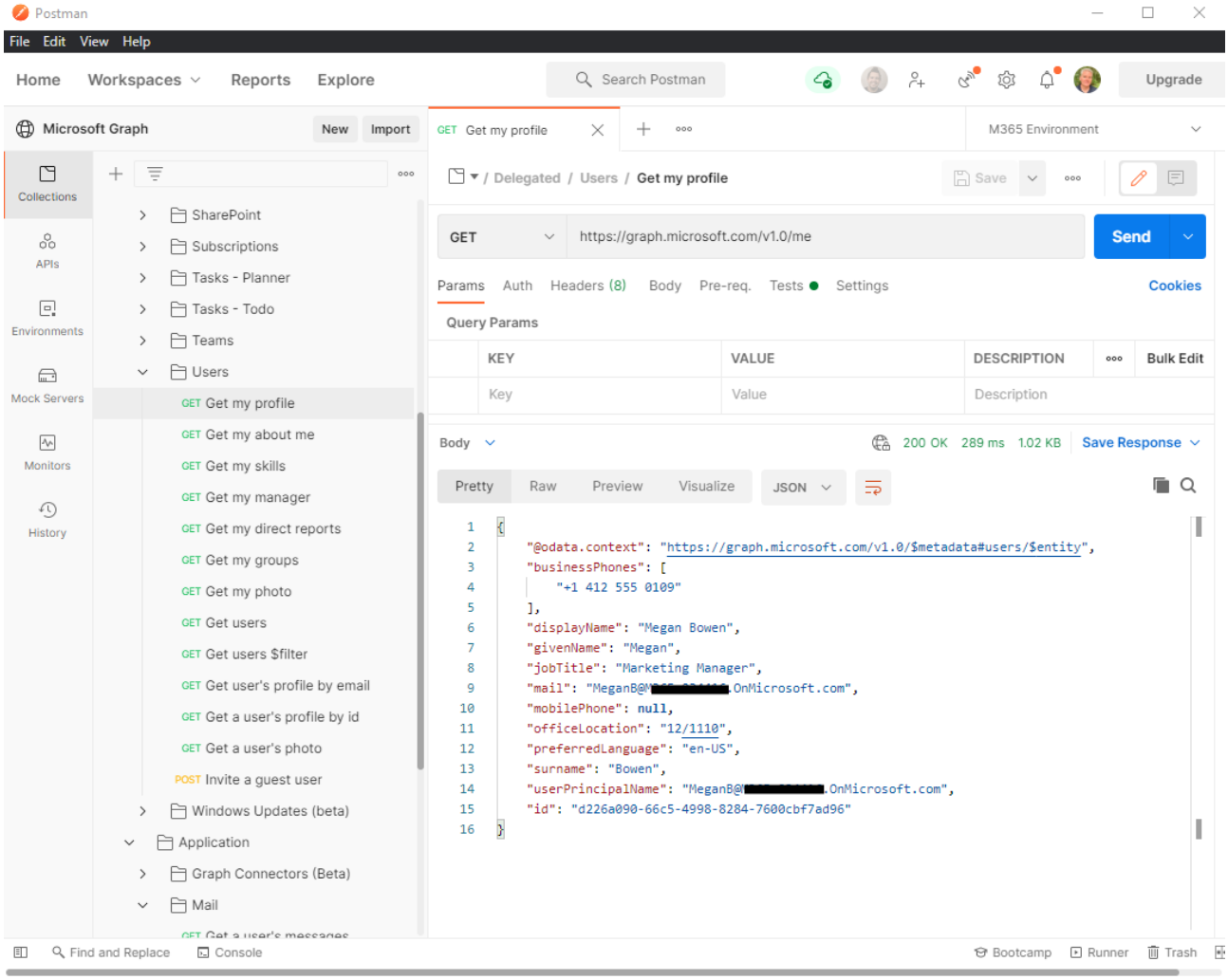



Рисунок 1.4 – Postman

Ефективне застосування цих інструментів дозволяє створити багаторівневий підхід до забезпечення безпеки веб-ресурсів. Комбінація автоматизованого сканування та ручного тестування є найкращою практикою для захисту сайтів від сучасних кіберзагроз.

Ці програмні рішення є незамінними інструментами для адміністраторів, розробників та експертів із кібербезпеки, які прагнуть забезпечити захист веб-ресурсів і мінімізувати ризики витоку даних.

## 1.3 Основні загрози та проблеми безпеки веб-сайтів

Забезпечення безпеки веб-ресурсів стає дедалі складнішим завданням через зростання кількості та складності кіберзагроз. Відсутність належного захисту може призводити до значних втрат — як фінансових, так і репутаційних. Розглянемо детальніше основні загрози, які мають найбільший вплив на веб-додатки.

SQL-ін'єкції залишаються однією з найпоширеніших форм атак, оскільки вони дозволяють маніпулювати базами даних через уразливості в коді веб-додатків. Використовуючи такі методи, зловмисники можуть отримати доступ до конфіденційних даних, таких як паролі користувачів, фінансова або особиста інформація [1]. Основна проблема полягає в тому, що введення даних користувачем не є належним чином автентифікованим. Наприклад, якщо поле для входу дозволяє вставляти шкідливі SQL-запити, це відкриває доступ до конфіденційної інформації. (див. рисунок 1.5).

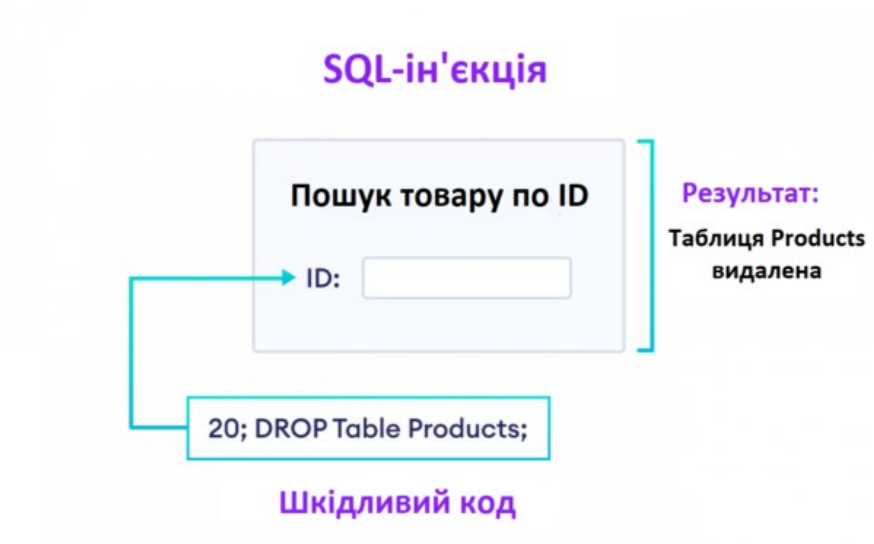



Рисунок 1.5 – приклад SQL-ін'єкції

XSS-атаки спрямовані на впровадження та виконання шкідливих скриптів на стороні користувача. Це може включати крадіжку кукі, фішингові атаки або маніпуляцію з відображенням сторінок [2]. Уразливості XSS зазвичай виникають через відсутність перевірки та очищення введених даних. Особливо небезпечними вони стають для ресурсів, які обробляють персональні або фінансові дані (див. рисунок 1.6).

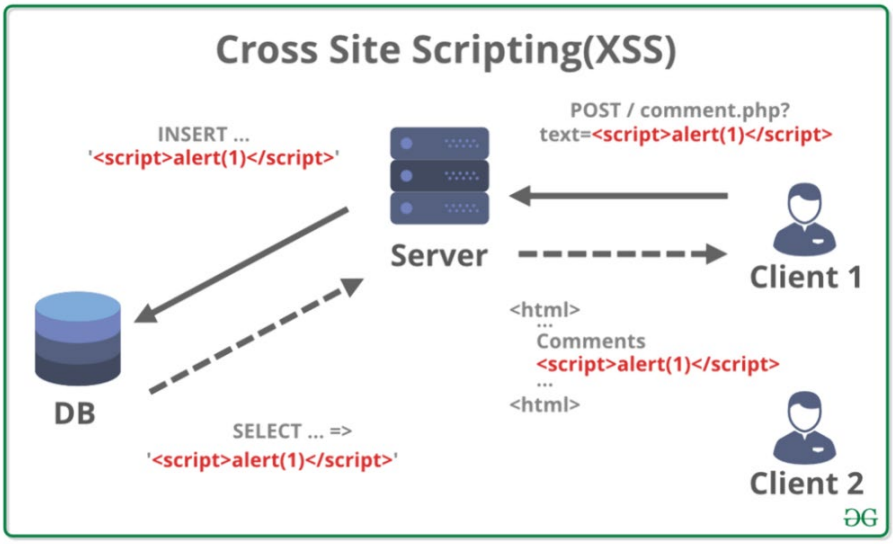



Рисунок 1.6 – приклад XSS-атаки

DDoS-атаки (розподілені атаки на відмову в обслуговуванні) спрямовані на виведення веб-сайту з ладу через навантаження його сервера запитами. Навіть тимчасова недоступність сайту може завдати значних збитків, особливо якщо мова йде про комерційні проекти [3]. Сучасні DDoS-атаки можуть бути дуже складними, комбінуючи різні техніки, щоб обійти традиційні засоби захисту (див. рисунок 1.7).

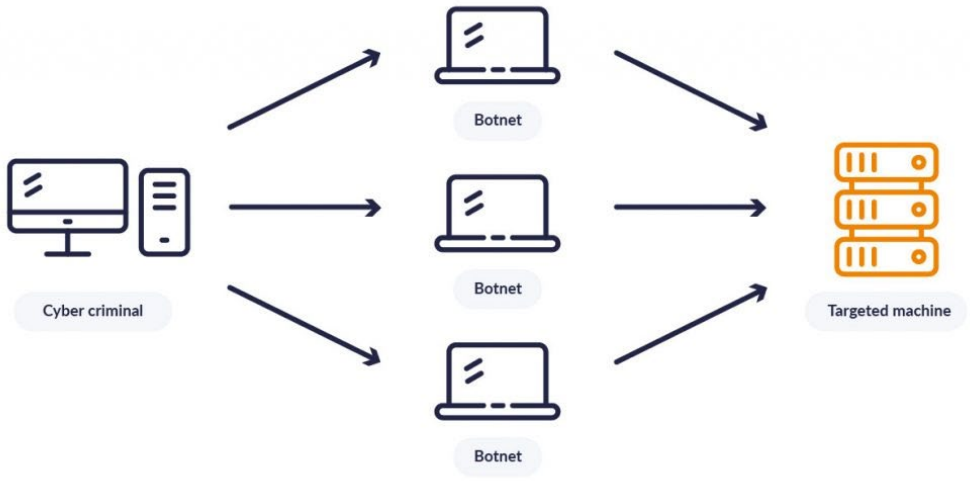



Рисунок 1.7 – приклад DDoS-атаки

MITM-атаки дозволяють перехоплювати комунікацію між сервером і клієнтом. Це можливо, якщо з'єднання не захищене протоколами шифрування, такими як SSL/TLS. Наприклад, зловмисник може отримати доступ до персональної інформації користувача, коли той заповнює форму на веб-сайті (див. рисунок 1.8).

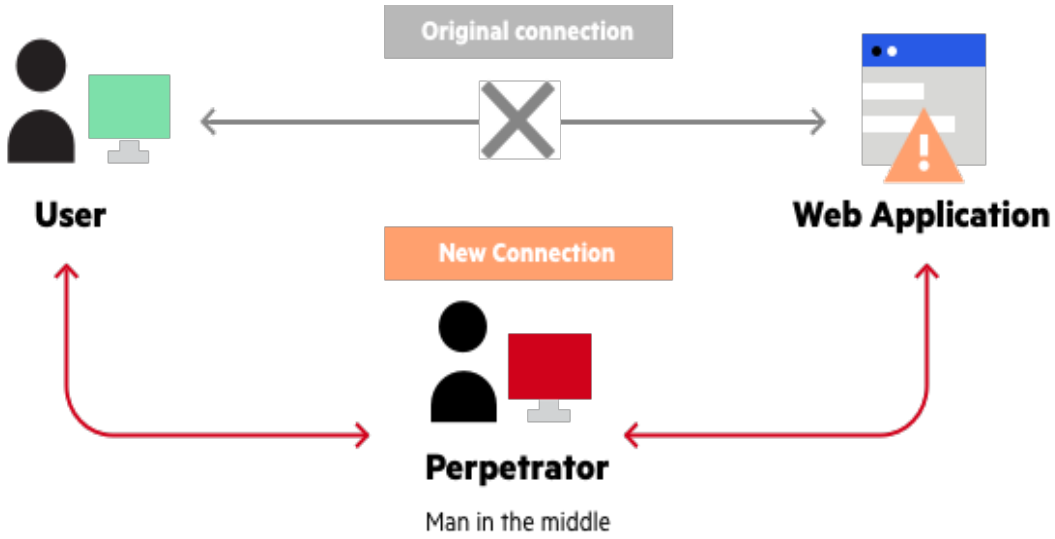



Рисунок 1.8 – приклад MITM-атаки

Безпека веб-сайтів залежить не лише від технічних заходів, але й від організаційних процесів, політики компанії та рівня підготовки персоналу.

Безпека веб-сайтів залежить не лише від технічних заходів, але й від організаційних процесів, політики компанії та рівня підготовки персоналу.

Однією з головних проблем є недооцінка важливості захисту веб-ресурсів, що може проявлятися в небажанні інвестувати в сучасні засоби кіберзахисту. Багато компаній нехтують регулярними аудитами безпеки, через що залишаються не виявленими уразливості, які можуть бути використані зловмисниками.

Крім того, значну роль відіграє людський фактор. Неналежна підготовка співробітників може призводити до нехтування простими заходами безпеки, такими як оновлення паролів, вчасне встановлення оновлень програмного забезпечення чи дотримання політики доступу.

Також до проблем відносять складність впровадження багатофакторної автентифікації. Хоча цей метод значно підвищує рівень захисту, багато користувачів вважають його незручним і схильні ігнорувати.

Організаційні проблеми можуть також стосуватися відсутності чітких політик щодо управління інцидентами безпеки. Без заздалегідь визначеного плану дій у випадку атаки компанії часто не в змозі оперативно реагувати на інциденти, що призводить до тривалих перебоїв у роботі та витоку інформації.

Усі ці аспекти вимагають системного підходу до безпеки, що включає регулярний аналіз загроз, впровадження сучасних технологій захисту, навчання персоналу та дотримання міжнародних стандартів, таких як PCI DSS та ISO 27001 [11].

# РОЗДІЛ 2 ОСНОВИ ТЕОРІЇ ТА МЕТОДИ ДОСЛІДЖЕНЬ

## 2.1 Методи і технології кіберзахисту веб-ресурсів

У сучасному світі кіберзахист веб-ресурсів став необхідністю для забезпечення безпеки та конфіденційності даних. Серед основних методів і технологій, що використовуються для захисту веб-сайтів, можна виділити кілька ключових аспектів.

Мережеві фаєрволи (Firewall) є базовими засобами для фільтрації вхідного та вихідного трафіку, блокуючи потенційно небезпечні запити. Проте для захисту саме веб-сайтів часто використовують Web Application Firewall (WAF) [12], які спеціалізуються на захисті від специфічних атак, таких як SQL-ін'єкції або міжсайтові скрипти (XSS). Інструменти як Cloudflare WAF або ModSecurity забезпечують додатковий рівень захисту, фільтруючи шкідливі запити на рівні додатка.

Забезпечення конфіденційності даних користувачів є одним з основних аспектів захисту веб-ресурсів. Використання SSL/TLS шифрування дозволяє захистити дані під час їх передачі між сервером і клієнтом, запобігаючи можливості їх перехоплення чи маніпуляцій. Сайт, що працює за протоколом HTTPS, гарантує більш високий рівень безпеки, що важливо для захисту особистої інформації користувачів, зокрема при проведенні фінансових транзакцій.

Використання багатофакторної аутентифікації (MFA) значно підвищує захищеність користувачів. Цей метод забезпечує додатковий рівень безпеки, коли користувач, окрім введення пароля, повинен пройти додаткову перевірку через одноразовий код, отриманий за допомогою додатків, таких як Google Authenticator чи Authy. Такий підхід значно зменшує ризик несанкціонованого доступу, навіть у разі компрометації пароля.

Для виявлення потенційних вразливостей веб-сайтів необхідно регулярно проводити аудит безпеки та тестування на проникнення. Інструменти на кшталт OWASP ZAP, Burp Suite і Nessus дозволяють автоматизувати процес виявлення

таких вразливостей, як SQL-ін'єкції, неправильна конфігурація доступу або уразливості, пов'язані з відсутністю належного шифрування. Постійний моніторинг системи, за допомогою таких засобів як Splunk або Nagios, дозволяє відстежувати діяльність веб-ресурсу і виявляти потенційні загрози в реальному часі, що дозволяє швидко реагувати на будь-які спроби атаки.

Загалом, комплексний підхід до кіберзахисту веб-ресурсів, який включає належне використання цих методів і технологій, є необхідним для забезпечення стабільної роботи сайту, захисту даних користувачів та запобігання можливим атакам.

### 2.1.1 Багатофакторна автентифікація

Багатофакторна автентифікація (MFA) — це спосіб забезпечення безпеки, що передбачає використання користувачем щонайменше двох різних факторів для підтвердження своєї особи. [13]. Основною метою MFA є ускладнити злом доступу до системи, навіть якщо один із факторів (наприклад, пароль) буде скомпрометований. Це додатковий рівень захисту, який знижує ризик несанкціонованого доступу, зокрема в ситуаціях фішинг-атак або використання вкрадених паролів.

Багатофакторна автентифікація базується на трьох основних категоріях факторів(див. рисунок 2.1):

–Що ви знаєте (Knowledge factor) — це інформація, яку знає користувач.

–Що у вас є (Possession factor) — це фізичний об'єкт, що знаходиться у користувача, наприклад, мобільний телефон для отримання одноразових кодів, апаратні токени або смарт-карти.

–Що ви є (Inherence factor) — фізіологічні характеристики, як-от відбитки пальців, аналіз обличчя або сканування сітківки ока.


Рисунок 2.1 – Багатофакторна автентифікація

MFA суттєво посилить захист веб-сайтів і онлайн-сервісів. Наприклад, навіть якщо зловмисник отримає доступ до вашого пароля, він не зможе увійти до облікового запису без додаткового фактора автентифікації, такого як одноразовий код чи біометричні характеристики.

Перевагами багатофакторної автентифікації є:

–Підвищена безпека — додаткові фактори підтвердження забезпечують більший захист від спроб несанкціонованого доступу;

–Захист від фішингу та крадіжки паролів — навіть якщо пароль скомпрометовано, зловмисники не зможуть отримати доступ до акаунта без додаткових факторів;

–Відповідність стандартам — багато регуляторних вимог, таких як PCI DSS для фінансових транзакцій, вимагають впровадження MFA для захисту платіжних систем та облікових записів користувачів.

Технологіями реалізації MFA є:

–Одноразові паролі (OTP) — це коди, що створюються за допомогою мобільних додатків (наприклад, Google Authenticator або Authy) чи надсилаються через SMS-повідомлення.

–Біометрія — сучасні смартфони та гаджети застосовують біометричні дані, зокрема відбитки пальців або розпізнавання обличчя, для ідентифікації користувача.

–Апаратні токени — спеціальні пристрої, які генерують одноразові паролі або використовуються для фізичної автентифікації (наприклад, USB-токени).

Таким чином, багатофакторна автентифікація є необхідною для захисту не тільки від стандартних атак, але й для забезпечення високого рівня безпеки на веб-ресурсах і в онлайн-сервісах, що зберігають особисті та фінансові дані. Її використання допомагає забезпечити додатковий рівень захисту і є важливою складовою частиною кібербезпеки в будь-якій організації чи особистому обліковому записі.

### 2.1.2 Використання систем WAF

Системи WAF (Web Application Firewall) (див. рис. 2.2) є важливим компонентом сучасної кібербезпеки, що забезпечують захист веб-додатків від численних загроз шляхом фільтрації та моніторингу HTTP/HTTPS-трафіку між користувачами та веб-серверами. Основною функцією систем є виявлення та блокування шкідливих запитів до веб-додатків.

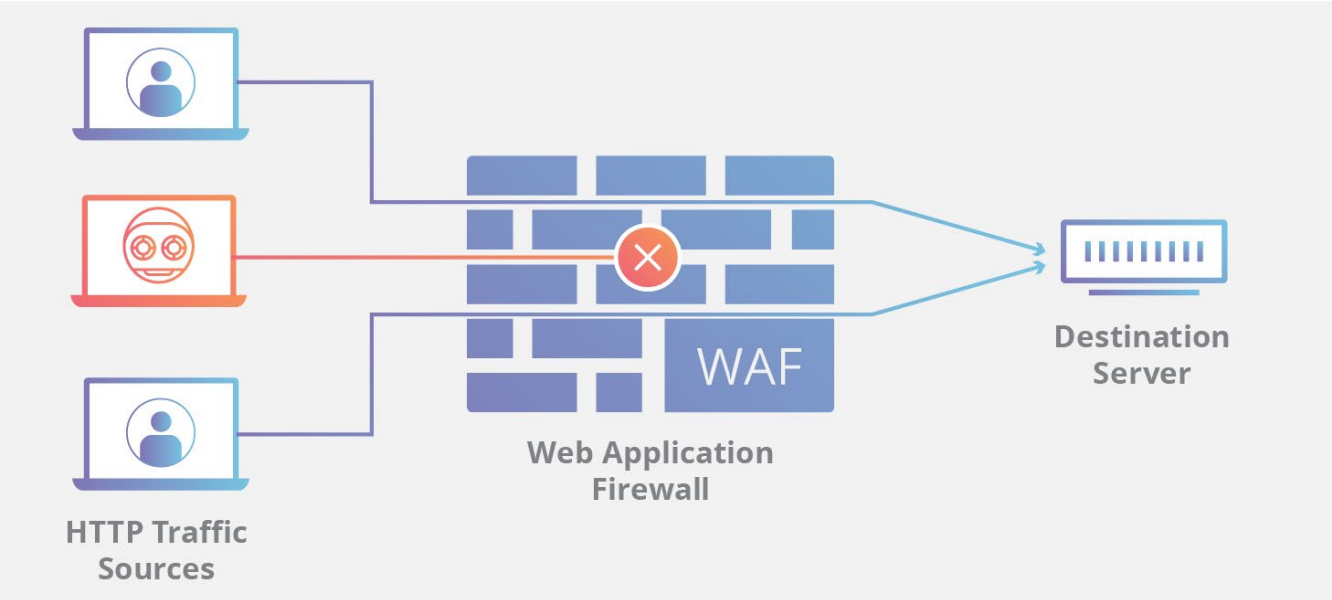



Рисунок 2.2 – система WAF

Основними функціями WAF є:

–Запобігання атакам на веб-додатки;

–Моніторинг трафіку в реальному часі;

–Фільтрація контенту та доступу;

–Адаптація до змін загроз.

Системи розрізняються за способом розгортання, функціоналом і сценаріями використання. Залежно від потреб організації, можна вибрати один із кількох типів WAF:

–Хмарні WAF це — хмарні системи забезпечують масштабованість і зручність у розгортанні, дозволяючи захищати веб-додатки без необхідності складної інфраструктури. Популярними рішеннями є AWS WAF, Cloudflare та Imperva;

–Апаратні WAF це — апаратні пристрої, встановлені у мережі організації. Вони пропонують високу продуктивність і контроль над конфігурацією, але мають високу вартість і складність в управлінні;

–Програмні WAF це — програми, які інтегруються з локальними системами і є гнучкими у налаштуванні, хоча вимагають більшого технічного досвіду.

Тип WAF вибирається залежно від потреб бізнесу, бюджету, технічної інфраструктури та рівня загроз. У великих організаціях із високим рівнем трафіку можуть використовуватися апаратні або гібридні рішення, тоді як малі підприємства та стартапи можуть зупинитися на хмарних або програмних WAF через їхню доступність і простоту у використанні.

## 2.2 Обґрунтування використовуваних технологій

Захист веб-ресурсів стає надзвичайно важливим на тлі зростання кількості та складності кібератак. Досягнення високого рівня безпеки потребує використання комплексних підходів, серед яких ключову роль відіграють системи Web Application Firewall (WAF) та багатофакторна автентифікація (MFA).

WAF використовується для захисту веб-додатків на рівні HTTP/HTTPS, аналізуючи запити між клієнтами та серверами. Ця технологія дозволяє блокувати атаки, такі як SQL-ін'єкції, міжсайтовий скриптинг (XSS), експлуатація вразливостей, пов’язаних із неправильною конфігурацією серверів, та інші загрози.

Перевага WAF у тому, що він працює в реальному часі, запобігаючи шкідливому трафіку ще до його обробки сервером. Гнучкі налаштування дозволяють адаптувати захист до специфіки конкретного веб-додатка, що є особливо важливим для бізнес-додатків із чутливими даними. Впровадження систем також сприяє дотриманню стандартів безпеки, таких як PCI DSS, які є обов’язковими для обробки платіжних даних.

MFA забезпечує додатковий рівень захисту, запитуючи декілька різних видів інформації для підтвердження особи користувача. До цих факторів належать: дані, які знає лише користувач, наприклад пароль або PIN-код; об'єкти, які користувач має при собі, як-от одноразовий код, смартфон чи токен; а також унікальні фізичні характеристики користувача, наприклад відбитки пальців чи зображення обличчя.

Це забезпечує багатошаровий підхід до безпеки, де компрометація одного з факторів не дозволяє зловмиснику отримати доступ до системи без інших необхідних даних. Таким чином, MFA ефективно захищає від несанкціонованого доступу навіть у разі втрати або крадіжки облікових даних.

Поєднання систем WAF (Web Application Firewall) та багатофакторної автентифікації (MFA) забезпечує надійний багатошаровий захист веб-ресурсів, спрямований на усунення широкого спектра кіберзагроз. WAF працює на рівні додатка, забезпечуючи про активний захист від таких атак, як SQL-ін'єкції, XSS або спроби доступу до конфіденційних даних через уразливості веб-сайтів. Він ефективно блокує підозрілі запити, аналізуючи їхній вміст і походження, запобігаючи виконанню потенційно небезпечного коду.

MFA, своєю чергою, додає додатковий рівень контролю доступу, запитуючи підтвердження користувача за кількома незалежними факторами. Це дозволяє

знизити ризик несанкціонованого доступу, навіть якщо паролі або інші облікові дані були скомпрометовані. Наприклад, навіть якщо зловмисник отримає пароль користувача, йому буде неможливо увійти до системи без одноразового коду чи підтвердження через біометричні дані.

Разом WAF і MFA створюють синергію: перший обмежує можливості здійснення зовнішніх атак, тоді як другий захищає систему від компрометації облікових записів. Ця комбінація не лише знижує ймовірність успішного злому, але й підвищує довіру користувачів до захисту їхніх даних. Застосування цих технологій є важливим кроком до створення комплексної системи кіберзахисту, особливо для організацій, які працюють із чутливою інформацією або фінансовими даними.

### 2.2.1 Аналіз ефективності багатофакторної автентифікації

Багатофакторна автентифікація (MFA) є однією з найбільш потужних технологій для забезпечення кібербезпеки, оскільки вона суттєво підвищує рівень захисту облікових записів і даних користувачів. Замість того, щоб покладатися лише на один фактор — пароль або PIN-код, MFA вимагає від користувача підтвердження особи за допомогою щонайменше двох різних факторів, що належать до трьох основних категорій: "щось, що знає користувач" (пароль або PIN), "щось, що має користувач" (наприклад, одноразовий код, апаратний токен або мобільний пристрій), і "щось, що є користувачем" (біометричні дані, такі як відбитки пальців або розпізнавання обличчя). Такий багаторівневий підхід до автентифікації значно знижує ймовірність того, що зловмисники зможуть отримати доступ до облікових записів, навіть якщо їм вдасться викрасти основний пароль користувача.

Однією з найбільших проблем сучасної кібербезпеки є слабкість паролів. Згідно з численними дослідженнями, понад 80% всіх випадків злому облікових записів відбуваються через вразливість паролів. Зловмисники використовують різні методи, зокрема соціальні інженерії, щоб викрасти паролі користувачів. В

такому контексті MFA стає критично важливим, оскільки навіть якщо зловмисник зможе отримати доступ до пароля, він не зможе увійти в систему без другого фактора автентифікації. Наприклад, у випадку фішингової атаки, коли пароль користувача був викрадений через підроблену веб-сторінку або фальшивий лист, система вимагатиме введення одноразового коду або використання біометрії для підтвердження особи. Це ускладнює можливість несанкціонованого доступу, оскільки зловмисник не матиме доступу до додаткових факторів автентифікації.

Фішинг є однією з найбільших загроз у кіберпросторі. Зловмисники часто використовують фальшиві сайти або електронні листи для того, щоб обдурити користувачів і отримати їхні облікові дані. Завдяки MFA, навіть якщо зловмисник отримає доступ до пароля користувача, йому не вдасться увійти до облікового запису без додаткових факторів, наприклад, одноразових кодів чи апаратних токенів. Це робить MFA одним з найбільш ефективних способів протидії фішинговим атакам, оскільки зловмисники не можуть здобути доступ до всіх необхідних даних за допомогою лише крадіжки пароля.

Ще однією важливою перевагою багатофакторної автентифікації є її здатність протидіяти атакам типу "brute force", коли зловмисники використовують програмне забезпечення для того, щоб спробувати вгадати пароль, перебираючи величезну кількість варіантів. В таких умовах традиційний пароль може бути легко зламаний за допомогою простих інструментів, однак із введенням другого фактору автентифікації атака стає значно складнішою і потребує значно більше часу та ресурсів. Таким чином, MFA захищає від успішних спроб зламати обліковий запис шляхом простого перебору паролів.

Крім того, MFA є ключовим інструментом для дотримання вимог міжнародних стандартів безпеки, таких як PCI DSS (захист даних платіжних карток), HIPAA (захист медичної інформації), GDPR (захист персональних даних у ЄС) та ISO 27001 (стандарти управління інформаційною безпекою). У багатьох галузях багатофакторна автентифікація є не лише рекомендованим, але й обов’язковим елементом для захисту критично важливої інформації та

персональних даних. Наприклад, у банківському секторі MFA застосовується для захисту онлайн-банкінгу. Для здійснення транзакцій недостатньо лише пароля; користувач має пройти додаткову перевірку, наприклад, ввести одноразовий код, отриманий на телефон, або скористатися іншими засобами автентифікації.

У випадку організацій, що працюють із чутливими даними, наприклад, у медичній або юридичній сферах, MFA є важливим елементом для забезпечення безпеки інформації, що зберігається в електронних архівах. Лікарі, медичний персонал та інші фахівці можуть получити доступ до медичних записів лише після успішного проходження багатофакторної автентифікації.

Це не лише підвищує рівень безпеки, а й сприяє збереженню конфіденційності даних пацієнтів, що є важливим аспектом для виконання вимог законодавства.

Впровадження MFA в організаціях також має багато переваг з точки зору зниження ризиків і управління безпекою. Використання багатофакторної автентифікації дозволяє скоротити кількість успішних кібератак, знижуючи ймовірність несанкціонованого доступу до критично важливої інформації систем і даних. Крім того, MFA може бути інтегрована в різні технологічні платформи та додатки, що робить її універсальним і гнучким рішенням для широкого кола користувачів та організацій. Враховуючи постійне зростання кількості кіберзагроз, MFA стає ключовим інструментом для захисту електронної інформації та забезпечення безпеки в онлайн-середовищі.

Загалом, багатофакторна автентифікація є невід'ємною частиною сучасної кібербезпеки, яка допомагає організаціям і користувачам захищати свої дані та облікові записи від численних кіберзагроз. Враховуючи постійне вдосконалення методів атак, MFA продовжує залишатися одним із найбільш ефективних засобів для зменшення ризиків і забезпечення високого рівня захисту в цифровому світі.

### 2.2.2 Переваги впровадження систем WAF для захисту веб-ресурсів

Розгортання систем WAF (Web Application Firewall) стало важливим компонентом сучасних стратегій кіберзахисту, оскільки вони забезпечують потужний бар'єр між користувачами і веб-серверами. Здатність WAF фільтрувати вхідний та вихідний трафік дозволяє захищати веб-ресурси від різноманітних загроз, таких як SQL-ін'єкції, міжсайтовий скриптинг (XSS), атаки типу «ін'єкція сеансу», а також інші вразливості на рівні додатків. Ці системи працюють шляхом виявлення та блокування шкідливих запитів до того, як вони досягнуть цілі, що значно знижує ймовірність успішних атак і мінімізує можливі втрати для організації.

Однією з основних переваг WAF є їх здатність не лише реагувати на вже відомі загрози, а й забезпечувати проактивний захист, аналізуючи вхідний трафік на ранніх етапах і блокуючи підозрілі запити до того, як вони можуть завдати шкоди. Це дозволяє суттєво знизити ймовірність успішного виконання атак, таких як SQL-ін'єкції, де зловмисники намагаються ввести шкідливі SQL-запити через форми на веб-сайтах. Проактивний захист WAF здатний ефективно фільтрувати ці запити, запобігаючи проникненню шкідливого коду в систему. Такий підхід дозволяє зменшити ймовірність компрометації веб-ресурсу, що є критично важливим для підтримки безпеки бізнесу.

Крім того, WAF є ефективним інструментом для забезпечення відповідності численним нормативним вимогам та стандартам безпеки. Одним з таких стандартів є PCI DSS, який регулює зберігання, обробку та передачу платіжної інформації. Вимоги цього стандарту зобов'язують компанії, які працюють із платіжними картками, впроваджувати додаткові заходи для захисту даних від несанкціонованого доступу. Використання WAF дозволяє організаціям забезпечити належний рівень безпеки та відповідність цьому стандарту, оскільки WAF можуть бути налаштовані за допомогою індивідуальних правил безпеки, які відповідають вимогам та потребам конкретної організації.

Окрім цього, системи WAF мають важливе значення для запобігання відключенню веб-ресурсів через атаки на сервери. Однією з найбільш популярних атак є DDoS-атака (distributed denial-of-service), коли зловмисники Розгортання систем WAF (Web Application Firewall) стало важливим компонентом сучасних стратегій кіберзахисту, оскільки вони забезпечують потужний бар'єр між користувачами і веб-серверами. Здатність WAF фільтрувати вхідний та вихідний трафік дозволяє захищати веб-ресурси від різноманітних загроз, таких як SQL-ін'єкції, міжсайтовий скриптинг, атаки типу «ін'єкція сеансу», а також інші вразливості на рівні додатків. Ці системи працюють шляхом виявлення та блокування шкідливих запитів до того, як вони досягнуть цілі, що значно знижує ймовірність успішних атак і мінімізує можливі втрати для організації.

Однією з основних переваг WAF є їх здатність не лише реагувати на вже відомі загрози, а й забезпечувати проактивний захист, аналізуючи вхідний трафік на ранніх етапах і блокуючи підозрілі запити до того, як вони можуть завдати шкоди. Це дозволяє суттєво знизити ймовірність успішного виконання атак, таких як SQL-ін'єкції, де зловмисники намагаються ввести шкідливі SQL-запити через форми на веб-сайтах. Проактивний захист WAF здатний ефективно фільтрувати ці запити, запобігаючи проникненню шкідливого коду в систему. Такий підхід дозволяє зменшити ймовірність компрометації веб-ресурсу, що є критично важливим для підтримки безпеки бізнесу.

Крім того, WAF є ефективним інструментом для забезпечення відповідності численним нормативним вимогам та стандартам безпеки. Одним з таких стандартів є PCI DSS, який регулює зберігання, обробку та передачу платіжної інформації. Вимоги цього стандарту зобов'язують компанії, які працюють із платіжними картками, впроваджувати додаткові заходи з охорони даних від несанкціонованого доступу. Використання WAF дозволяє організаціям забезпечити належний рівень безпеки та відповідність цьому стандарту, оскільки WAF можуть бути налаштовані за допомогою індивідуальних правил безпеки, які відповідають вимогам та потребам конкретної організації.

Окрім цього, системи WAF мають важливе значення для запобігання відключенню веб-ресурсів через атаки на сервери. Однією з найбільш популярних атак є DDoS-атака (distributed denial-of-service), коли зловмисники використовують величезну кількість скомпрометованих пристроїв для надання серверу великої кількості запитів, що може призвести до його перевантаження та відключення. Завдяки інтеграції з WAF, ці атаки можна нейтралізувати, знижуючи навантаження на сервер та запобігаючи його перебоям у роботі. Це дозволяє зберігати працездатність веб-ресурсу навіть під час великих навантажень, забезпечуючи стабільність і безперервність надання послуг.

Запобігання відключенню веб-ресурсів не тільки підтримує стабільність і працездатність сервісу, а й має важливе значення для репутації компанії. Довірливі користувачі і клієнти очікують безперервної роботи онлайн-сервісів, і будь-які перебої в роботі можуть призвести до втрати довіри та фінансових втрат. Системи WAF, надаючи захист від атак, допомагають зберегти репутацію організації, що особливо важливо для бізнесів, які працюють у конкурентних галузях, таких як електронна комерція або фінансові послуги.

Ще однією суттєвою перевагою WAF є їх здатність до гнучкої інтеграції як в хмарні, так і в локальні інфраструктури. Завдяки такій гнучкості, компанії можуть налаштовувати свої системи безпеки відповідно до специфічних потреб і вимог. У разі хмарної інфраструктури, WAF можуть бути використані для захисту від атак на рівні додатків, забезпечуючи додатковий рівень захисту для хмарних сервісів і даних, що зберігаються в цих середовищах. У локальних інфраструктурах ці системи можуть бути налаштовані для захисту серверів, що забезпечує безпеку критично важливих даних і додатків.

Аналіз трафіку в реальному часі також є важливою складовою частиною функціонування WAF. Системи мають можливість детально вивчати запити, що надходять, і швидко реагувати на нові загрози. Це дозволяє операторам безпеки адаптувати свої стратегії захисту до нових викликів, що з'являються у кіберпросторі. Наприклад, зловмисники постійно вдосконалюють свої методи

атак, і WAF забезпечує ефективний моніторинг і адаптацію захисту до таких змін, дозволяючи знизити ризики успішних атак.

З огляду на ці численні переваги, WAF стали важливим елементом комплексних систем веб-безпеки, сприяючи підвищенню рівня захисту організацій від різноманітних кіберзагроз. Вони допомагають забезпечити стабільність і безпеку веб-ресурсів, захищають конфіденційні дані від несанкціонованого доступу і знижують ймовірність успішних атак. У поєднанні з іншими заходами безпеки, такими як багатофакторна автентифікація, системи WAF стають частиною більш широкої стратегії захисту, яка гарантує збереження цілісності та безпеки організаційних систем у цифровому середовищі.

## 2.3 Інтеграція методів захисту у веб-ресурсах

Використання додаткових серверів для підвищення безпеки веб-ресурсів є важливим аспектом комплексного захисту, особливо коли йдеться про захист від численних загроз в Інтернеті. Інтеграція таких методів захисту, як WAF (Web Application Firewall) та MFA (Multi-Factor Authentication), дає змогу організувати багаторівневу систему безпеки, яка ефективно запобігає атакам і підвищує рівень захисту даних. Впровадження таких технологій у реальних умовах продемонструвало їхню ефективність у різних галузях, від роздрібної електронної комерції до фінансового та медичного секторів.

Одним із найяскравіших прикладів застосування WAF є роздрібна електронна комерція, де високий рівень навантаження на сайти та численні спроби атак роблять їх уразливими до багатьох видів загроз. Наприклад, у випадку великого онлайн-маркетплейсу, такого як Amazon, використання WAF дозволяє ефективно захищати веб-ресурси від атак, таких як SQL-ін'єкції, DDoS-атаки та інших методів вторгнення. Ці системи здатні фільтрувати трафік і блокувати шкідливі запити в режимі реального часу, що дозволяє компанії забезпечити безперебійну роботу сайту навіть під час пікових навантажень, таких як «Чорна п’ятниця», коли кількість запитів може збільшитися в десятки разів.

Водночас, застосування багатофакторної автентифікації (MFA) у таких умовах додатково підвищує рівень безпеки. Використання двофакторної автентифікації через SMS-коди або мобільні додатки дозволяє захистити транзакції та облікові записи користувачів. Завдяки інтеграції WAF і MFA, компанії можуть бути впевненими, що їхня платформа захищена від широкого спектра загроз, а клієнти можуть здійснювати покупки та виконувати інші операції в безпечному середовищі.

Подібний підхід застосовується й у банківському секторі. Наприклад, банк JPMorgan Chase активно інтегрує MFA для авторизації клієнтів при доступі до онлайн-банкінгу. У цьому випадку користувачі отримують одноразові коди через мобільні додатки або SMS, що додає ще один рівень захисту. Паралельно з MFA банк застосовує WAF для моніторингу веб-запитів і запобігання спробам крадіжки даних, таких як міжсайтові скриптові атаки (XSS), що дозволяє забезпечити безпеку транзакцій і захистити конфіденційну інформацію клієнтів. Крім того, ці системи допомагають знизити ймовірність виникнення шахрайських схем і вчасно виявляти підозрілі дії.

У сфері охорони здоров’я технології захисту також відіграють важливу роль. Наприклад, платформи для медичних записів, такі як Epic Systems, використовують WAF для захисту чутливих даних пацієнтів від несанкціонованого доступу. У цих системах також активно впроваджуються методи багатофакторної автентифікації, щоб забезпечити доступ до медичних записів тільки авторизованим особам. Це дозволяє лікарям і медичним працівникам працювати з даними пацієнтів лише після додаткової перевірки, що значно підвищує рівень конфіденційності та безпеки медичних даних.

Хмарні сервіси, такі як Google Cloud, також активно використовують WAF для фільтрації небажаних запитів і захисту від атак на рівні додатків. Окрім того, MFA є важливою складовою частиною безпеки таких сервісів, як Gmail та Google Drive. Це дозволяє значно підвищити рівень захисту акаунтів користувачів, навіть для приватних осіб. Завдяки застосуванню таких технологій, користувачі

можуть бути впевнені в тому, що їхні дані захищені від несанкціонованого доступу та можливих загроз.

Одним із ключових аспектів інтеграції додаткових серверів у процес захисту є надання можливості додаткової обробки і фільтрації трафіку. Встановлення окремих серверів для аналізу запитів дозволяє створити спеціалізовану інфраструктуру, яка здатна виконувати глибоке сканування та фільтрацію трафіку до того, як він досягне основних серверів веб-сайту. Це дозволяє зменшити навантаження на основний сервер і забезпечити більш ефективне реагування на потенційні загрози.

Така інфраструктура може включати додаткові сервери для перевірки запитів на наявність шкідливих елементів або невірно сформованих пакетів даних [16]. Наприклад, ці сервери можуть використовувати методи обробки та фільтрації трафіку, що базуються на інтелектуальному аналізі (такі як машинне навчання для виявлення аномалій), що дозволяє зменшити кількість фальшивих сповіщень і підвищити ефективність виявлення загроз.

Таким чином, комбіноване використання WAF та MFA, а також додаткових серверів для обробки і фільтрації трафіку, дозволяє створити багаторівневу систему захисту, яка ефективно протистоїть численним загрозам і зберігає безпеку веб-ресурсів. Це не лише підвищує рівень захисту даних, але й дозволяє компаніям забезпечити стабільну роботу своїх веб-сайтів, навіть під час пікових навантажень або в умовах постійних кіберзагроз. Інтеграція цих технологій є необхідною для забезпечення безпеки та збереження довіри користувачів у сучасному цифровому середовищі.

### 2.3.1 Способи інтеграції WAF та MFA в інфраструктуру веб-додатків

Інтеграції систем WAF (Web Application Firewall) та MFA (Multi-Factor Authentication) в інфраструктуру веб-додатків є важливим кроком у створенні багатошарового захисту, який ефективно знижує ризики від різноманітних кіберзагроз. Ці дві технології є важливими компонентами сучасної кібербезпеки,

адже вони забезпечують не лише захист від атак на рівні додатка, але й підвищують рівень автентифікації користувачів. Зростаюча складність і кількість атак вимагають від організацій впровадження комплексних рішень, які зможуть забезпечити максимальний рівень безпеки.

Інтеграція WAF зазвичай передбачає його розміщення між веб-сервером і інтернетом. Це дає можливість фільтрувати вхідний трафік і блокувати небезпечні запити ще до того, як вони досягнуть серверу. Система WAF аналізує HTTP/HTTPS запити на наявність загроз. Налаштування правил безпеки дозволяє адаптувати WAF до специфіки конкретного веб-додатка, блокуючи лише небажані запити або створюючи політики для різних груп користувачів. Наприклад, для організацій, які обробляють чутливу інформацію, WAF може бути налаштований на більш суворі правила, що знижують ризики витоку даних.

Інтеграція MFA додає додатковий рівень безпеки завдяки використання двох або більше факторів для підтвердження особи користувача. Це значно знижує ймовірність несанкціонованого доступу, навіть якщо пароль був скомпрометований. Інтеграція MFA може бути реалізована через різні механізми, такі як одноразові паролі (OTP), біометрія чи апаратні токени, що забезпечують високий рівень захисту, надаючи другий фактор підтвердження для отримання доступу до веб-додатка. Таким чином, навіть якщо зловмисник отримує доступ до пароля, йому буде важче обійти багатофакторну автентифікацію.

Існує кілька способів інтеграції цих двох систем, які дозволяють забезпечити максимальний рівень безпеки. Один із них — це інтеграція на рівні веб-сервера, коли WAF розташований перед веб-сервером для фільтрації трафіку, а MFA активується після спроби входу користувача. У такому випадку запити, які пройшли через WAF, перенаправляються на сервер, що вимагає додаткової автентифікації через MFA. Це дозволяє створити багаторівневий захист, який забезпечує безпеку як на рівні додатка, так і на рівні доступу користувача.

Ще одним методом є API-орієнтована інтеграція, коли WAF взаємодіє з системами MFA через API, що дозволяє більш гнучко налаштувати правила безпеки і реагувати на змінювані загрози. У цій моделі WAF може миттєво адаптуватися до нових типів атак, забезпечуючи своєчасний захист. Також можна використовувати проксі-сервери, які перехоплюють трафік та перенаправляють його для перевірки MFA. Інтеграція через проксі забезпечує прозорість для користувачів, зберігаючи високий рівень безпеки. Цей підхід дозволяє уникнути значних затримок у взаємодії користувача з веб-додатком, що підвищує загальний рівень користувацького досвіду.

Використання хмарних рішень, таких як AWS [17] WAF або Azure Web Application Firewall, також є потужним варіантом інтеграції WAF і MFA. Хмарні сервіси забезпечують масштабованість, високу доступність та зручність налаштування, що дає змогу організаціям швидко реагувати на змінні умови та нові загрози. Інтеграція MFA через спеціалізовані хмарні сервіси забезпечує надійний захист, одночасно знижуючи витрати на інфраструктуру.

Комбінування WAF і MFA дозволяє створити високоефективний бар'єр, який мінімізує ймовірність успішних атак. Забезпечуючи захист на всіх етапах взаємодії користувача з веб-додатком, ці технології разом утворюють потужний механізм безпеки. В результаті, навіть якщо одна з ліній захисту буде зламана, друга (наприклад, автентифікація через MFA) забезпечить надійний бар'єр для зловмисників. Впровадження WAF і MFA не тільки підвищує рівень безпеки веб-додатків, але й підвищує довіру користувачів до сервісів, що є ключовим фактором для бізнесу в умовах сучасного ринку.

Отже, інтеграція WAF і MFA в інфраструктуру веб-додатків є стратегічно важливим кроком у забезпеченні комплексної безпеки. Це дозволяє організаціям не тільки захищати свої ресурси, але й запобігати можливим атакам, створюючи таким чином стійку та надійну систему безпеки. Впровадження таких заходів безпеки сприяє не лише захисту даних, але й збереженню репутації організації в очах користувачів та партнерів, що є надзвичайно важливим у сучасному цифровому середовищі.

# РОЗДІЛ 3 АНАЛІЗ ТА ОЦІНКА РЕЗУЛЬТАТІВ

## 3.1 Результати аудиту безпеки веб-сайту

Аудит веб сайта My Health (див. рисунок 3.1) проводився згідно з методологією OWASP (Open Web Application Security Project) [18], яка є стандартом для аналізу безпеки веб-додатків. Основну увагу було приділено виявленню поширених вразливостей, зокрема:

–Вразливість до XSS (Cross-Site Scripting);

–SQL-ін’єкції;

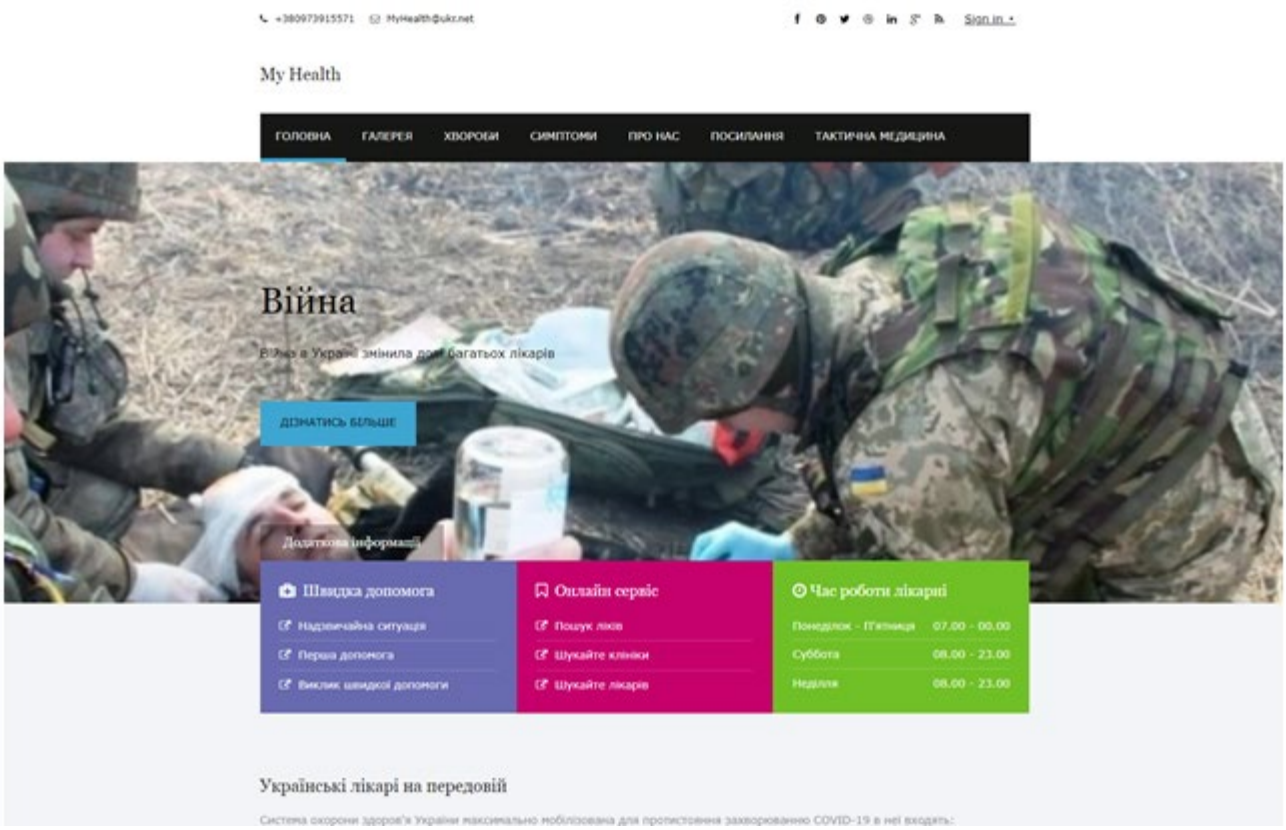



Рисунок 3.1 – Головна сторінка сайту My Health

Для аудиту безпеки веб-додатку використовувались такі інструменти:

–OWASP ZAP – для автоматизованого сканування та виявлення поширених уразливостей;

–Postman – для ручного тестування запитів API та перевірки безпеки передачі даних.

З використанням OWASP ZAP проведено динамічне тестування веб додатка (див. рисунок 3.2). Основний акцент зроблено на пошук наступних типів вразливостей:

–XSS (Cross-Site Scripting): виконано автоматичний аналіз введення користувачів, перевірено, чи не допускається виконання шкідливого JavaScript-коду;

–CSRF (Cross-Site Request Forgery): оцінено ризики передачі шкідливих запитів через довірені сесії.

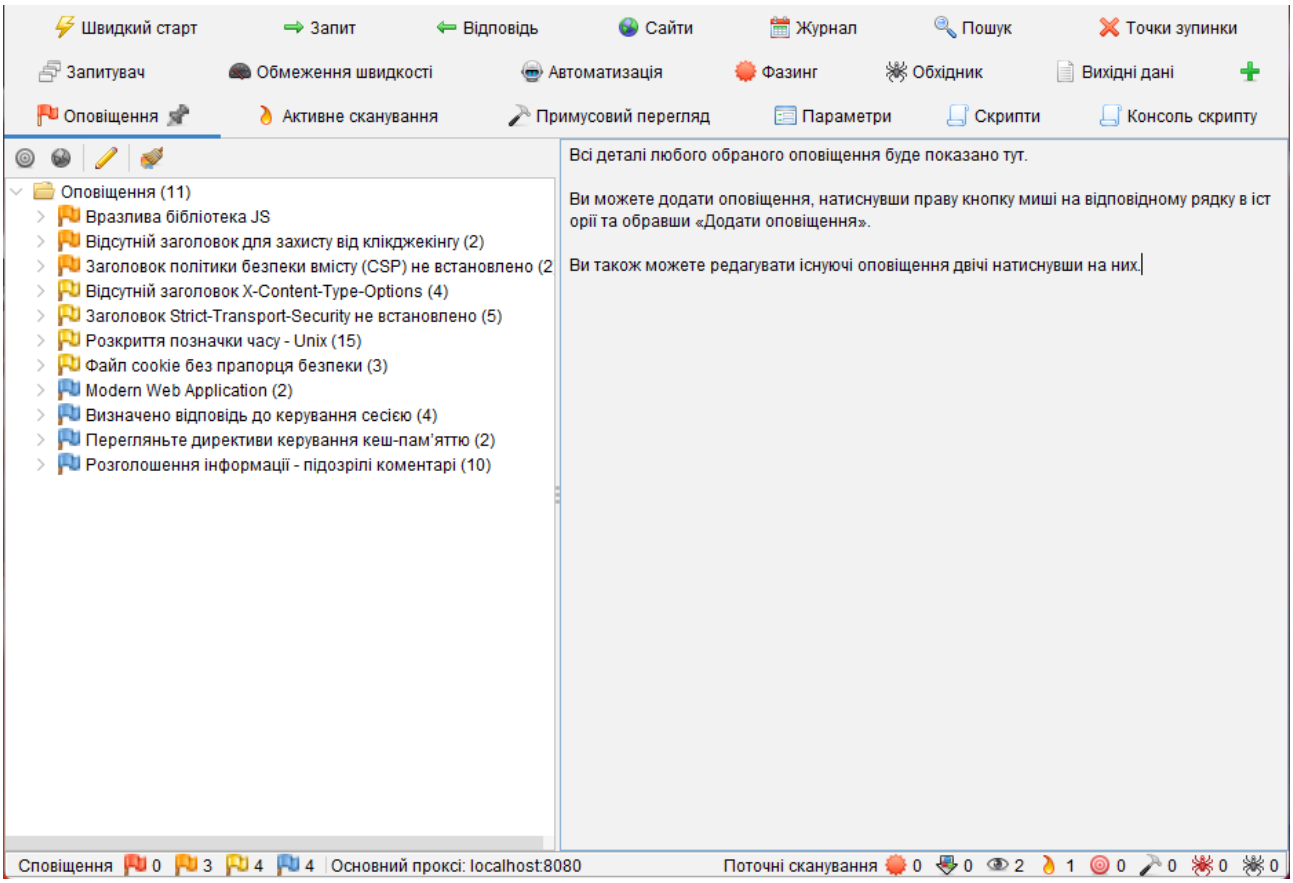



Рисунок 3.2 – Тестування у OWASP ZAP

Використовуючи інструмент Postman, було виконано ручне тестування API-запитів (див. рисунок 3.3). Перевірено:

–Коректність обробки HTTP-запитів (GET, POST, PUT, DELETE);

–Відповідність вхідних і вихідних даних заявленим специфікаціям API;

–Обробку помилок, зокрема реакцію на неправильно сформовані запити;

–Наявність аутентифікації та захисту від небажаного доступу до приватних даних.

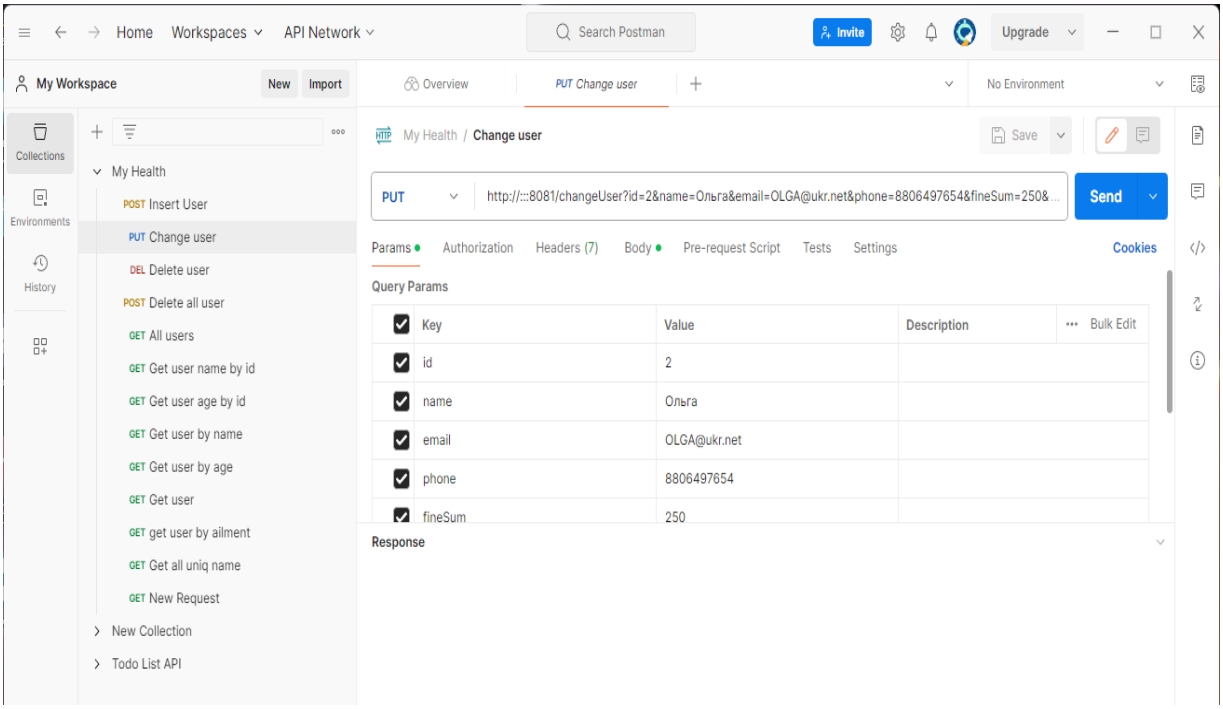



Рисунок 3.3 – Тестування у Postman

Мета аудиту полягала у визначенні потенційних вразливостей, аналізі архітектури сайту та бази даних, а також у розробці рекомендацій для покращення безпеки. Аудит внесений у таблицю 1.1

Таблиця 1.1 – Аудит безпеки веб сайта My Health

| Секція | Деталі |
|---|---|
| Мета аудиту | Виявлення вразливостей у сайті "My Health", аналіз архітектури та бази даних, надання рекомендацій для покращення безпеки. |
| Інфраструктура | – Архітектура: компонентний підхід на JavaScript (компоненти: App.js, Header.js, Router, Switch).<br>– База даних: три таблиці (Users, Ailments, Symptoms). |

Продовження таблиці 1.1

| | |
|---|---|
| Виявлені вразливості | –Недостатній захист від XSS-атак.<br>– Вразливість до SQL-ін’єкцій.<br>– Відсутність захисту від CSRF-атак.<br>Низький рівень шифрування даних. |
| Рекомендації | –Впровадження фільтрації вхідних даних для XSS-захисту.<br>– Використання підготовлених SQL-запитів.<br>– Реалізація CSRF-токенів.<br>–Хешування паролів (bcrypt). |
| Висновок | Рекомендовано впровадити зазначені заходи для зменшення ризиків, захисту конфіденційності користувачів і покращення безпеки сайту. |

На основі аудити можна побачити що:

Сайт побудовано з використанням компонентного підходу на основі JavaScript. Основні компоненти:

–App.js — головний компонент, який об’єднує інші модулі;

–Header.js — модуль, що керує заголовками та навігаційними елементами;

–Роутинг здійснюється за допомогою компонентів Router та Switch, які спрямовують користувачів до таких сторінок, як: Home.js, About.js, Symptoms.js, Ailments.js;

–Автоматизація розгортання здійснюється через Cloud Manager;

–Присутній ручний та автоматизований процес тестування для запобігання помилкам у виробничому середовищі.

База даних включає три основні таблиці:

–Users (зберігає інформацію про користувачів: ім'я, вік, наявні захворювання);

–Ailments (дані про захворювання, їх симптоми, зв’язок з користувачами);

–Symptoms (описує симптоми, пов'язані з конкретними захворюваннями).

Було виявлено такі вразливості:

–Відсутні механізми фільтрації вхідних даних для компонентів, таких як Symptom і Ailment;

–Ймовірність виконання шкідливого коду через форми введення;

–Немає використання підготовлених запитів для взаємодії з базою даних;

–Не було виявлено захисту у формі токенів для автентифікації запитів.

–Поля password у таблиці Users не зашифровані.

На основі аудиту можна дати такі рекомендації:

–Впровадити валідацію та фільтрацію вхідних даних;

–Використовувати бібліотеки, такі як DOMPurify;

–Використовувати ORM-бібліотеки (наприклад, Sequelize або TypeORM) для безпечної взаємодії з базою даних;

–Впровадити підготовлені запити;

–Впровадити CSRF-токени у всіх формах;

–Використовувати JWT-токени для авторизації;

–Для паролів використовувати алгоритм хешування, наприклад, bcrypt;

–Забезпечити SSL/TLS для всіх запитів до сервера.

–Регулярно виконувати Penetration Testing та автоматизоване сканування.

Проведений аудит виявив низку критичних вразливостей, які можуть поставити під загрозу конфіденційність та безпеку користувачів сайту "My Health". Рекомендовано вжити зазначених заходів для покращення загального рівня захищеності системи.

## 3.2 Реалізація додаткового сервера для покращення безпеки

Після аналізу аудиту, було розроблено додатковий сервер для забезпечення безпеки системи. Додатковий сервер відповідає за захист основного серверу, зокрема за фільтрацію вхідних запитів, логування та моніторинг активності. Сервер реалізовано на основі фреймворку NestJS.

Результатом реалізації додаткового сервера є його інтеграція з основним сервером Meteor.js [19], що дозволяє підвищити загальну стійкість системи до потенційних загроз.

До процесу створення додаткового сервера входять такі кроки:

–Завантаження інсталяційного пакету NestJS з офіційного сайту.

–Налаштування нового проекту за допомогою CLI NestJS.

–Реалізація служб та контролерів для захисту.

–Інтеграція додаткового сервера з основним сервером Meteor.js.

–Тестування взаємодії серверів.

Першим кроком процесу розгортання сервера є завантаження фреймворку NestJs [20]. Встановлення відбувається після запуску команди nest new.

Після завантаження та встановлення фреймворку NestJS необхідно налаштувати його, додавши необхідні пакети для забезпечення функціональності та захисту додаткового сервера. До списку обов'язкових залежностей належать такі пакети, як helmet, xss-clean, express, та інші інструменти, які покращують безпеку та продуктивність сервера.

Далі виконується налаштування середовища та інтеграція серверу з основним сервером Meteor.js. Для цього додаються спеціальні бібліотеки та модулі для обробки запитів і забезпечення їх пересилання між серверами. Після внесення змін проводиться тестовий запуск сервера для перевірки його коректної роботи за допомогою команди: nest start(див. рисунок 3.4).

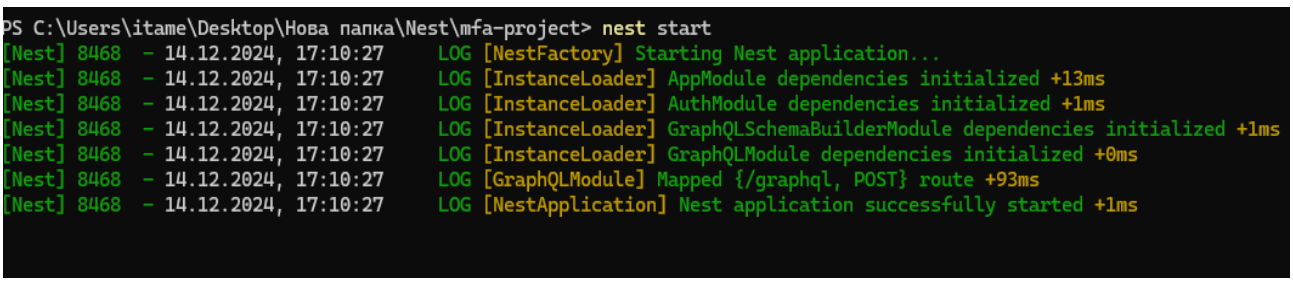



Рисунок 3.4 – Запуск додаткового сервера

Додатковий сервер працює як проксі, пересилаючи перевірені запити до основного серверу Meteor.js. Для цього створено сервіс, який здійснює пересилання який зображений на рисунку 3.5.

```
import { Injectable, HttpService } from '@nestjs/common';
@Injectable()
export class AppService {
  private readonly meteorBaseUrl = 'http://localhost:3000';
  constructor(private readonly httpService: HttpService) {}
  async forwardRequest(data: any): Promise<any> {
    return await this.httpService.post(`${this.meteorBaseUrl}/api`, data).toPromise();
  }
```

Рисунок 3.5 – Інтеграція з основним сервером

Після цього додатковий сервер повністю готовий до експлуатації.

## 3.3 Впровадження багатофакторної автентифікації

Для забезпечення належного рівня захисту веб-додатків необхідно враховувати два основних аспекти безпеки:

–Захист додатка — гарантія того, що сторонні користувачі або зловмисники не зможуть модифікувати або порушити функціонування веб-додатка;

–Інформаційна безпека — захист персональних даних та іншої чутливої інформації, що зберігається в додатку.

Перед розробкою багатофакторної автентифікації було проаналізовано вимоги до веб-додатка, включаючи сценарії автентифікації, обробку даних користувачів, а також захист від атак, таких як brute-force, фішинг та викрадення сесій.

Для реалізації автентифікації використовувався React [21] (для клієнтської частини) і Nest (для серверної частини) через наступні причини:

–React дозволяє створювати динамічні інтерфейси користувача, зокрема, для форм введення паролів, підтверджень через код або біометричних даних;

–Nest забезпечує просту інтеграцію серверної логіки, автентифікації та безпечну взаємодію з базою даних;

Процес реалізації автентифікації було розподілено на кілька етапів:

На першому етапі користувач вводить свої облікові дані, які зберігаються на сервері у вигляді хешів. Для хешування використовується алгоритм bcrypt, що забезпечує надійний рівень захисту, навіть якщо база даних буде зламана (див. рисунок 3.6).

```
@Injectable()
export class AuthService {
  private readonly otpStorage: { [email: string]: { otp: string; expiry: number } } =

  async hashPassword(password: string): Promise<string> {
    return await bcrypt.hash(password, 10);
  }

  async verifyPassword(password: string, hashedPassword: string): Promise<boolean> {
    return await bcrypt.compare(password, hashedPassword);
  }
```

Рисунок 3.6 – захист паролів із використанням bcrypt

На другому етапі після успішного входу користувача система генерує одноразовий пароль і надсилає його через SMS, електронну пошту або мобільний додаток. Цей код має обмежений термін дії, щоб мінімізувати ризик використання вкрадених кодів (див.рисунок.3.7).

```
  generateOtp(email: string): { otp: string; expiry: number } {
    const otp = uuidv4().slice(0, 6); // Generate 6-digit OTP
    const expiry = Date.now() + 300000; // OTP valid for 5 minutes
    this.otpStorage[email] = { otp, expiry };
    console.log(`Generated OTP for ${email}: ${otp}`);
    return { otp, expiry };
  }

  verifyOtp(email: string, otp: string): boolean {
    const record = this.otpStorage[email];
    if (!record || record.otp !== otp || record.expiry < Date.now()) {
      return false;
    }
    delete this.otpStorage[email];
    return true;
  }
}
```

Рисунок 3.7 – Генерація одноразового пароля

У третьому етапі користувач вводить отриманий код через інтерфейс, створений у React. Сервер перевіряє правильність коду та його термін дії (див. рисунок 3.8).

```
import { Resolver, Mutation, Args } from '@nestjs/graphql';
import { AuthService } from './auth.service';

@Resolver()
export class AuthResolver {
  constructor(private readonly authService: AuthService) {}

  @Mutation(() => String)
  async login(@Args('email') email: string, @Args('password') password: string): Promise<string> {
    // Example logic
    const otp = this.authService.generateOtp(email);
    return `OTP sent: ${otp.otp}`;
  }

  @Mutation(() => String)
  async verifyOtp(@Args('email') email: string, @Args('otp') otp: string): Promise<string> {
    const isValid = this.authService.verifyOtp(email, otp);
    if (!isValid) {
      throw new Error('Invalid OTP or expired');
    }
    return 'Authentication successful';
  }
}
```

Рисунок 3.8 – Перевірка одноразового пароля

Додатково для забезпечення зберігання сесій Nest використовує токени автентифікації з обмеженим терміном дії. При завершенні терміну дії токена

автоматично генерується новий, що дозволяє уникнути ризику втрати доступу через прострочені токени.

Реалізація багатофакторної автентифікації в веб-додатку значно знижує ризики, пов'язані з атакою на облікові записи користувачів. Комбінація React та Nest дозволяє створювати безпечні та зручні інтерфейси для користувачів, забезпечуючи водночас гнучкість і масштабованість серверної частини.

## 3.4 Впровадження системи WAF

Сайт My Health має розширену структуру, що забезпечує доступ до інформації про здоров'я, медичні дані, а також дозволяє створювати та обновлювати дані. Для захисту веб-додатку від кіберзагроз, таких як SQL-ін'єкції, XSS-атаки та DDoS-атаки, впроваджено систему Web Application Firewall (WAF).

Система має бути інтегрована у ключові структури сайту, а саме:

–Дані про користувачів;

–Дані про симптоми;

–Дані про хвороби.

Розділ користувачів забезпечує обробку персональних даних, таких як ім'я, та пов'язані захворювання і симптоми. Щоб захистити ці дані усі запити До бази даних застосовується валідація, яка запобігає SQL-ін'єкціям, а також під час внесення змін до профілю користувача дані перевіряються на стороні клієнта та сервера (див. рисунок 3.9).

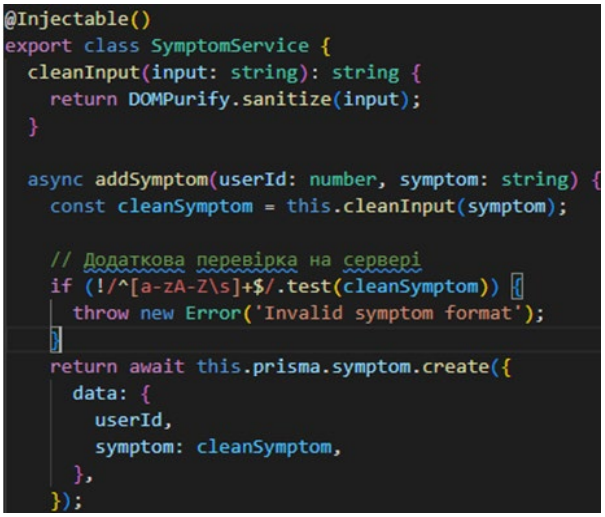
```
import { IsString, Matches } from 'class-validator';

export class CreateDiseaseDto {
  @IsString()
  @Matches(/^[a-zA-Z\s]+$/, { message: 'Invalid disease name format' })
  name: string;

  @IsString()
  description: string;
}
```

Рисунок 3.9 – Перевірка вхідних даних

Модуль для обробки симптомів включає механізми очищення та перевірки вхідних даних, щоб уникнути XSS-атак. Введені користувачем симптоми перевіряються бібліотекою DOMPurify. Дані перед записом у базу додатково перевіряються регулярними виразами на сервері (див .рисунок 3.10).



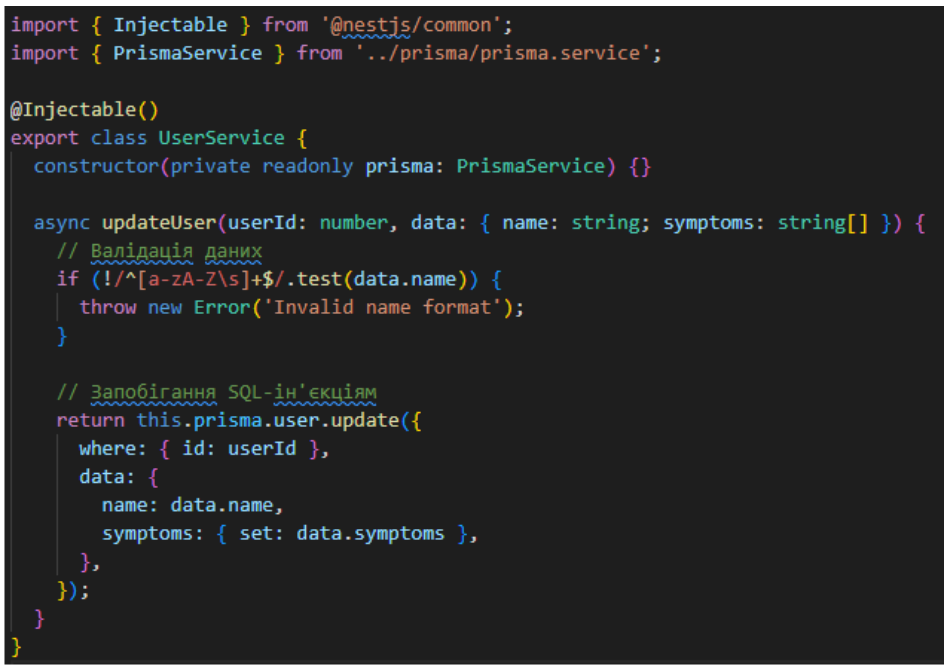
```
@Injectable()
export class SymptomService {
  cleanInput(input: string): string {
    return DOMPurify.sanitize(input);
  }

  async addSymptom(userId: number, symptom: string) {
    const cleanSymptom = this.cleanInput(symptom);

    // Додаткова перевірка на сервері
    if (!/^[a-zA-Z\s]+$/.test(cleanSymptom)) {
      throw new Error('Invalid symptom format');
    }
    return await this.prisma.symptom.create({
      data: {
        userId,
        symptom: cleanSymptom,
      },
    });
```

Рисунок 3.10 – Фрагмент коду валідації

Модуль хвороб захищає функціонал створення та оновлення інформації про хвороби. Запити до бази даних перевіряються на відповідність очікуваним форматам. Пошук і створення записів захищені від SQL-ін'єкцій за допомогою регулярних виразів(див. рисунок 3.11).



```
import { Injectable } from '@nestjs/common';
import { PrismaService } from '../prisma/prisma.service';

@Injectable()
export class UserService {
  constructor(private readonly prisma: PrismaService) {}

  async updateUser(userId: number, data: { name: string; symptoms: string[] }) {
    // Валідація даних
    if (!/^[a-zA-Z\s]+$/.test(data.name)) {
      throw new Error('Invalid name format');
    }

    // Запобігання SQL-ін'єкціям
    return this.prisma.user.update({
      where: { id: userId },
      data: {
        name: data.name,
        symptoms: { set: data.symptoms },
      },
    });
  }
}
```

Рисунок 3.11 – Захист від SQL-ін'єкцій

Завдяки впровадженню WAF система забезпечує захист даних користувачів, симптомів і хвороб від основних кіберзагроз. Комбінація Nest.js, DOMPurify, валідації вхідних даних і ORM дозволяє створити гнучку та безпечну архітектуру.

## 3.5 Тестування та оцінка ефективності запропонованих рішень

Тестування та оцінка ефективності запропонованих рішень, таких як система веб-аплікаційного файрволу (WAF) та багатофакторна автентифікація (MFA), є критично важливими етапами у забезпеченні кібербезпеки веб-додатка "My Health". Ці процеси дозволяють не лише перевірити функціональність впроваджених технологій, але й оцінити їхню здатність ефективно захищати дані від реальних загроз.

Система WAF тестується на здатність блокувати різноманітні атаки, такі як SQL-ін'єкції, XSS та DDoS. Для цього використовуються методи пенетраційного тестування, які імітують дії зловмисників. Результати таких тестів дозволяють оцінити ефективність виявлення та блокування потенційно шкідливого трафіку. Важливим аспектом є перевірка налаштувань WAF, які мають бути достатньо жорсткими для захисту даних, але не створювати перешкод для законного користувацького трафіку.

Багатофакторна автентифікація перевіряється на надійність генерації одноразових кодів, їх захищеність від перехоплення та стабільність каналів передачі (SMS, електронна пошта або мобільні додатки). Тестування також охоплює стійкість системи до атак типу brute-force і перевірку механізмів блокування після багаторазових невдалих спроб входу. Важливо також оцінити зручність MFA для користувачів, адже складність у використанні може спричинити зниження довіри або відмови від технології.

Оцінка ефективності базується на аналізі результатів тестування: вимірюванні часу відгуку системи, кількості успішно заблокованих атак, стійкості до навантажень та забезпеченні безперервної роботи під час несправностей. Отримані результати дозволяють виявити можливі слабкі місця,

внести необхідні корективи у конфігурацію та підтвердити відповідність системи вимогам нормативних актів, таких як GDPR [22].

Таким чином, тестування і оцінка ефективності WAF і MFA є необхідними для забезпечення стабільної, надійної та захищеної роботи веб-додатка. Це дозволяє не лише знизити ризики компрометації даних, але й підвищити довіру користувачів до ресурсу, який відповідає найвищим стандартам кібербезпеки.

### 3.5.1 Тестування багатофакторної автентифікації

Для тестування багатофакторної автентифікації використовується Postman. Цей інструмент дозволяє провести тестування API, який дозволяє автоматизувати запити, перевіряти відповіді сервера та аналізувати їхню поведінку в реальних умовах.

При тестуванні MFA через Postman основний акцент робиться на перевірці процесу авторизації, включаючи отримання одноразових кодів, перевірку їхньої дійсності, а також стабільність та безпеку передачі даних.

Першим етапом необхідно надіслати запит на реєстрацію нового користувача до API. В запиті вказується основна інформація, така як ім’я користувача, електронна пошта та пароль. Цей запит перевіряє, чи система коректно приймає дані та створює акаунт (див. рисунок 3.12).

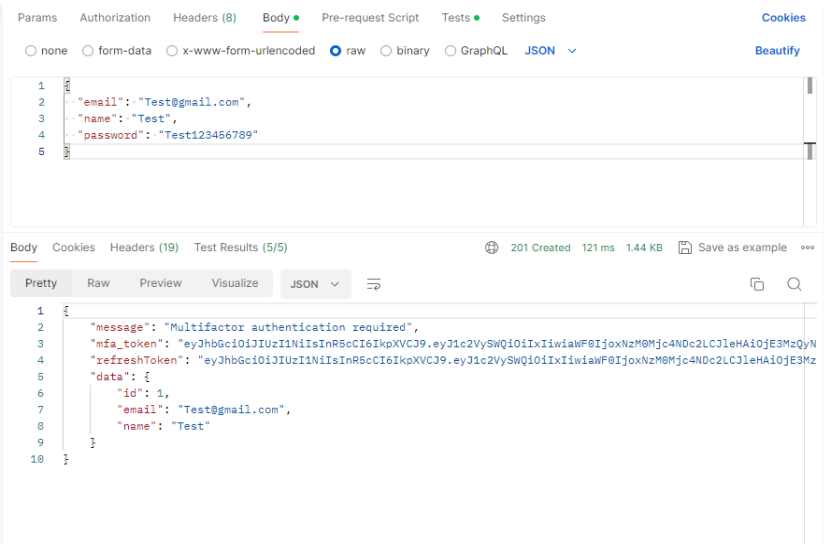



Рисунок 3.12 – Створення користувач

Після успішного створення акаунту наступним важливим етапом є перевірка можливості входу в акаунт. Цей етап тестування дозволяє оцінити, чи коректно працює процес авторизації. Користувач надсилає запит на вхід до системи, вказуючи свою електронну пошту та пароль (див. рисунок 3.13).

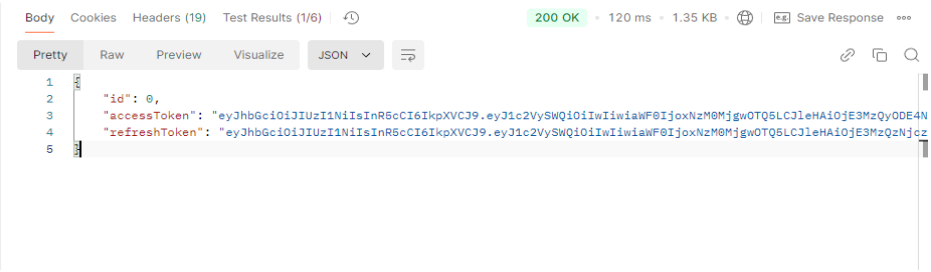



Рисунок 3.13 – Вхід у акаунт

Після успішного входу в акаунт користувач отримує токен доступу (наприклад, JWT), який використовується для аутентифікації подальших запитів до системи. Наступним важливим кроком є включення двоетапної автентифікації (MFA), що значно підвищує рівень безпеки облікового запису. Цей процес також можна протестувати за допомогою Postman, щоб переконатися у коректності реалізації API та надійності системи.

Користувач надсилає запит на сервер із використанням токену доступу для ініціації процесу активації двоетапної автентифікації. Запит має містити заголовок авторизації з токеном (див. рисунок 3.14).

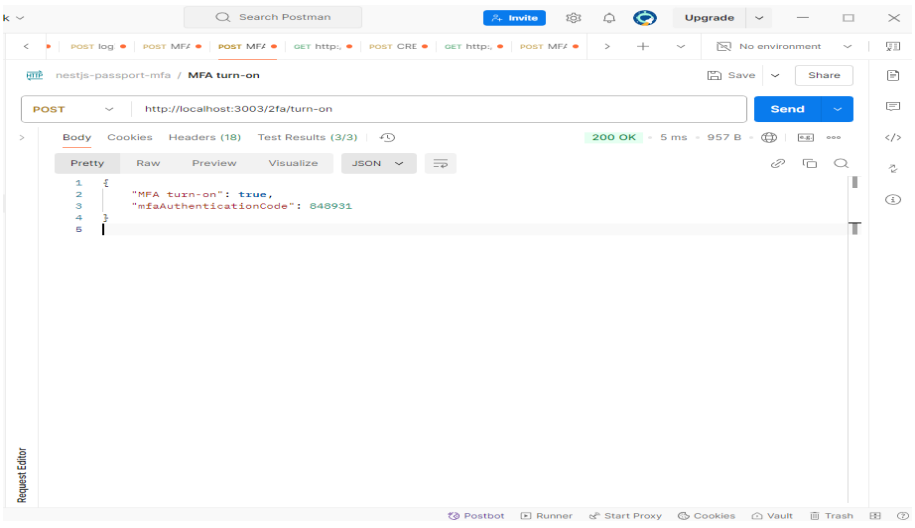



Рисунок 3.14 – Включення двоетапної автентифікації

Після успішного включення двоетапної автентифікації (MFA) сервер надає користувачеві MFA-токен, який дозволяє додатково захищати запити. Наступним кроком у тестуванні є перевірка функціональності цього токена, що включає перевірку, чи він дійсно потрібен для виконання операцій після виходу з облікового запису. Щоб перевірити токен потрібно вийти із системи (див. рисунок 3.15) та увійти використовуючи отриманий токен (див. рисунок 3.16).

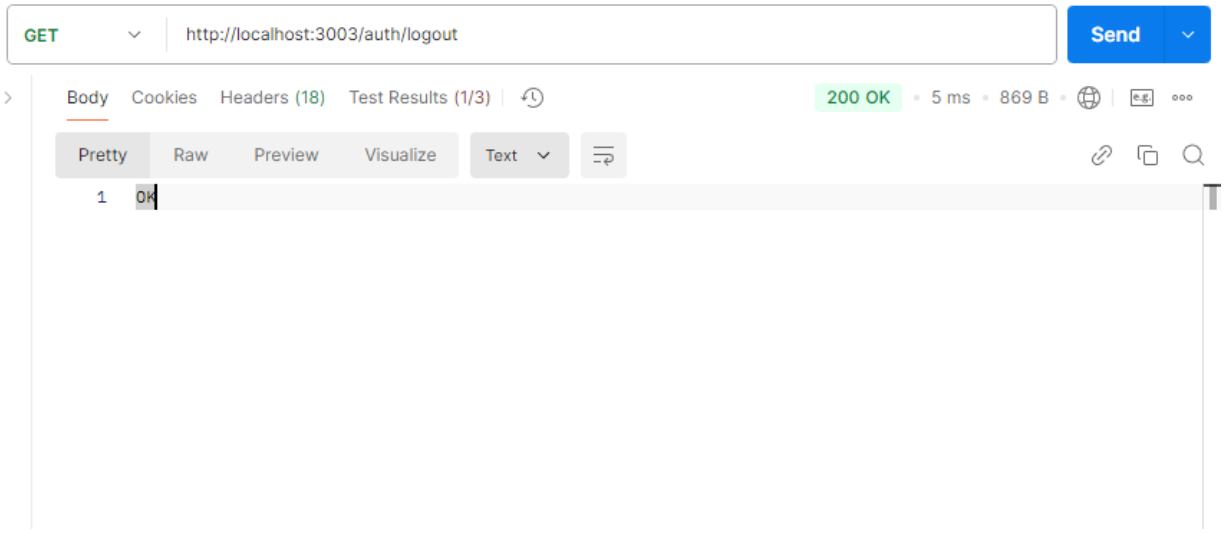



Рисунок 3.15 – Вихід із системи

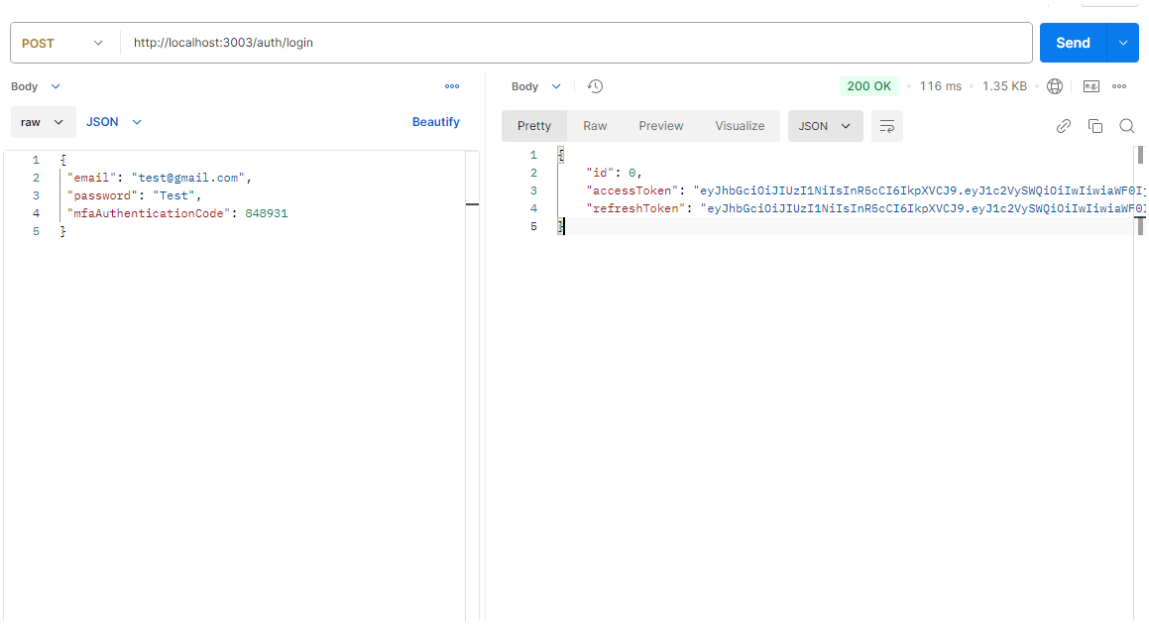



Рисунок 3.16 – Вхід у систему із використанням токена

Перевірка MFA-токена через Postman є критичною для підтвердження того, що двоетапна автентифікація працює належним чином. Це забезпечує додатковий рівень захисту для облікових записів користувачів веб-додатка "My Health" і значно ускладнює доступ зловмисникам, навіть у разі компрометації пароля.

### 3.5.2 Тестування системи WAF

Під час тестування системи за допомогою OWASP ZAP було виявлено кілька ключових вразливостей, які потребують уваги (див. рисунок 3.17).

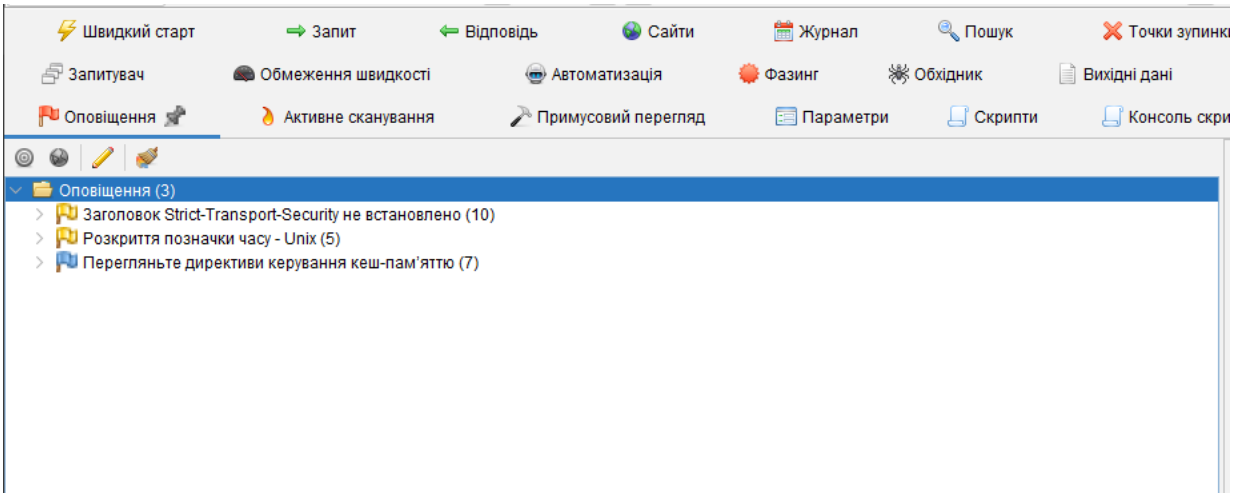



Рисунок 3.17 – Тестування WAF за допомогою ZAP

Заголовок Strict-Transport-Security (HSTS) не встановлено, що створює ризик атак типу Man-in-the-Middle (MITM) через відсутність примусового використання HTTPS-з'єднання. Розкриття позначки часу Unix свідчить про можливість аналізу роботи системи зловмисниками, а недостатня конфігурація директив керування кеш-пам’яттю відкриває ризики зберігання чутливої інформації в небезпечних місцях, таких як кеш браузера або проксі-серверів.

Незважаючи на ці проблеми, впровадження WAF позитивно вплинуло на загальний рівень безпеки системи. Система почала ефективніше блокувати підозрілі запити, зокрема ті, що спрямовані на експлуатацію відсутності HTTPS

або на аналіз часових позначок. WAF також забезпечив автоматичний аналіз і фільтрацію запитів, які могли б призвести до витоку даних через неправильне кешування. Завдяки активному моніторингу та виявленню небезпечних шаблонів поведінки вдалося значно зменшити ймовірність успішної реалізації атак.

Попри позитивні зміни, тестування виявило, що існують аспекти, які потребують вдосконалення. Для посилення захисту слід додати заголовки, такі як HSTS та Cache-Control, а також змінити серверну конфігурацію для приховування часових позначок. WAF демонструє ефективність як перший рівень захисту, проте повна безпека залежить від комплексного підходу, який включатиме налаштування серверів, впровадження додаткових політик та регулярний моніторинг.

Впровадження WAF дало значний прогрес у забезпеченні безпеки. Автоматичний аналіз запитів та блокування загроз суттєво знизили ризики, пов’язані з недостатньою конфігурацією серверів. Система тепер більш ефективно захищена від MITM-атак, витоку даних та можливих зловживань кешем. Однак, залишаються деякі вразливості, які слід вирішити для досягнення оптимального рівня безпеки. Для цього необхідне подальше вдосконалення серверної конфігурації та посилення політик безпеки, які забезпечать додатковий захист навіть за відсутності WAF.

## 3.6 Обґрунтування доцільності впроваджених засобів

Впровадження сучасних засобів кібербезпеки є критично важливим для забезпечення безпеки веб-ресурсів, зокрема в тих, що оперують чутливою інформацією, як, наприклад, медичні дані на сайті "My Health". Веб-додатки, що зберігають особисті дані користувачів, зокрема медичні записи, піддаються численним загрозам, і тому потребують особливо ретельного підходу до їх захисту. Одними з найбільш ефективних засобів для підвищення рівня безпеки є багатофакторна автентифікація (MFA) та системи веб-аплікаційного файрволу (WAF). Вебсайт "My Health" застосував ці технології для запобігання

зловмисним діям і забезпечення надійного захисту персональних даних своїх користувачів.

Багатофакторна автентифікація є одним з найефективніших способів захисту облікових записів користувачів. Вона вимагає не лише введення пароля, але й додаткових факторів, таких як одноразовий пароль (OTP), що надсилається через SMS, електронну пошту чи інші канали зв'язку, або використання біометричних даних. Такий підхід ускладнює доступ до акаунтів користувачів для зловмисників, оскільки для отримання доступу потрібно зламати не лише пароль, а й отримати доступ до іншого каналу підтвердження.

Тестування багатофакторної автентифікації для веб-сайту "My Health" показало високий рівень захисту. Всі вхідні запити перевірялися через систему, яка аналізувала правильність введених даних і їх відповідність додатковим критеріям безпеки. Після введення пароля система генерувала одноразовий код, який надсилався користувачу через безпечний канал, наприклад, SMS або мобільний додаток. Цей код має обмежений термін дії, що мінімізує можливість його використання після того, як його отримають зловмисники.

Навіть якщо пароль користувача буде зламано або вкрадено, зловмисник не зможе отримати доступ до його акаунту без додаткових факторів автентифікації. Крім того, система автоматично блокує доступ після декількох невдалих спроб входу, що захищає від атак типу brute-force.

Одним із основних переваг багатофакторної автентифікації є те, що вона не лише забезпечує додатковий рівень безпеки, але й адаптується до різних вимог користувачів, дозволяючи налаштовувати методи автентифікації залежно від конкретних потреб або рівня ризику. Для користувачів, які використовують особливу чутливу інформацію, можна застосовувати більш строгі методи автентифікації, як-от біометричні дані, що ще більше покращує рівень безпеки.

Другим важливим елементом кіберзахисту веб-сайту "My Health" є система веб-аплікаційного файрволу (WAF). WAF забезпечує захист веб-ресурсів від різноманітних атак на рівні додатків, таких як-от SQL-ін'єкції, міжсайтовий скриптинг (XSS), атаки типу "крадіжка сеансу" та багато інших. Ця система

фільтрує і моніторить вхідний і вихідний трафік між користувачами і веб-серверами, аналізуючи запити і блокуючи ті, що можуть бути шкідливими. Вона здатна виявляти та нейтралізувати загрози ще до того, як вони досягнуть серверу, що значно знижує ймовірність успішного проникнення.

Для реалізації системи WAF на веб-сайті "My Health" були проведені ретельні тести, які показали високу ефективність цього інструменту в боротьбі з поширеними видами атак. Наприклад, при тестуванні на SQL-ін'єкції було виявлено, що система виявляє та блокує небезпечні запити, що намагаються втрутитися в бази даних або змінити їх структуру. Зважаючи на важливість захисту медичних даних, ця функція є критичною, оскільки будь-яке втручання в базу даних може призвести до серйозних наслідків.

Подібний рівень захисту забезпечується і для атак типу XSS, які можуть використовуватися для виконання шкідливих сценаріїв на стороні користувача. WAF на сайті "My Health" успішно фільтрує такі запити, тим самим блокуючи можливість виконання шкідливого коду на веб-сторінках.

Також система WAF дозволяє блокувати DDoS-атаки, що є дуже важливим для збереження безперебійної роботи сайту. Вона може миттєво виявляти аномальні запити та нейтралізувати їх до того, як вони досягнуть серверу. Це дає змогу забезпечити стабільну роботу сайту навіть при високих навантаженнях, зокрема у випадку масових атак з різних джерел.

Впровадження додаткового сервера, є важливим кроком для посилення захисту та підвищення продуктивності веб-додатка "My Health". Цей сервер виконує кілька ключових функцій. По-перше, він забезпечує розподіл навантаження між основними серверами, що запобігає перевантаженню інфраструктури під час високих пікових навантажень або масованих DDoS-атак. По-друге, сервер слугує резервним вузлом для зберігання та обробки критично важливих даних, що гарантує безперервність роботи навіть у разі відмови основного серверу.

Крім того, додатковий сервер виконує функцію фільтрації трафіку. Він аналізує вхідні та блокує потенційно небезпечні запити. Це посилює

ефективність системи WAF, яка інтегрована у загальну архітектуру додатку. Важливим аспектом впровадження є також підтримка масштабованості: додатковий сервер дозволяє оперативно розширювати інфраструктуру, забезпечуючи її адаптацію до зростаючих потреб, таких як збільшення кількості користувачів або обсягу даних.

Результатом впровадження є значне підвищення стійкості системи до атак, таких як DDoS, SQL-ін'єкції чи XSS. Тестування показало, що навіть за умов високого навантаження або цілеспрямованих атак сайт "My Health" залишається стабільним і захищеним. Додатковий сервер сприяє оптимізації продуктивності, забезпечуючи швидкість обробки запитів і низькі затримки у взаємодії з користувачами.

Таким чином, завдяки застосуванню таких засобів кіберзахисту, як додатковий сервер, багатофакторна автентифікація та веб-аплікаційний файрвол, сайт "My Health" забезпечує надійний захист своїх користувачів, що дозволяє зберігати конфіденційність та цілісність медичних даних і запобігати можливим кіберзагрозам.

# 4 ОХОРОНА ПРАЦІ ТА БЕЗПЕКА В НАДЗВИЧАЙНИХ СИТУАЦІЯХ

## 4.1 Охорона праці

Сучасна діяльність у сфері розробки та підтримки веб-сайтів, зокрема "My Health", вимагає створення безпечних і комфортних умов праці для персоналу. Охорона праці є невід'ємною складовою забезпечення ефективності та безперервності процесів, пов'язаних із інформаційними технологіями. Це включає заходи з підтримання оптимального мікроклімату, протипожежної безпеки, ергономіки, освітлення, а також дотримання вимог діючих нормативних документів. У цьому розділі розглядаються основні аспекти, які впливають на якість умов праці, а також їх відповідність нормативним стандартам України.

Протипожежна безпека є одним із ключових аспектів. Для її забезпечення необхідно дотримуватись вимог нормативних документів, зокрема ДБН В.1.1-7:2016 "Пожежна безпека об'єктів будівництва". Основними заходами є використання сертифікованих електричних пристроїв з вогнестійкою ізоляцією, регулярна перевірка стану електромережі та забезпечення доступу до первинних засобів пожежогасіння. Згідно з цим документом, серверні приміщення повинні бути обладнані системами автоматичного пожежогасіння, які не пошкоджують електроніку, наприклад, газового типу. Крім того, евакуаційні шляхи мають відповідати нормам ДБН В.2.5-56:2014 та бути оснащеними сигнальними покажчиками й належним освітленням.

Ергономіка та захист здоров'я працівників також є важливими. Тривала робота з комп'ютерним обладнанням може спричинити фізичне та психоемоційне навантаження. Для зменшення цих ризиків необхідно облаштовувати ергономічні робочі місця з правильним розташуванням моніторів, клавіатур і меблів. Наприклад, монітор повинен розташовуватися на рівні очей, а клавіатура – у зручному доступі для рук. Регулярні перерви для вправ, що знижують напруження, є обов’язковими. Освітленість приміщень має відповідати стандартам ДБН В.2.5-28-2018, забезпечуючи рівномірне освітлення

без відблисків. Світлодіодні лампи з регульованою яскравістю сприяють підтримці комфортних умов роботи, знижуючи ризик перевтоми очей.

Захист інформаційних систем у контексті роботи веб-сайту "My Health" передбачає забезпечення фізичної безпеки серверного обладнання, реалізацію політик резервного копіювання даних і встановлення систем моніторингу температури та вологості у серверних приміщеннях. Такі системи допомагають уникнути перегріву обладнання, що може призвести до збоїв у роботі веб-сайту. Відповідно до ДБН В.2.5-67:2013 [23] "Опалення, вентиляція та кондиціонування", необхідно забезпечити регулярну перевірку та налаштування кліматичних систем.

Щодо вентиляції та мікроклімату, важливо підтримувати оптимальні умови. Параметри мікроклімату в приміщеннях з комп'ютерним обладнанням регулюються ДСанПіН 3.3.2.007-98 [24]. Зокрема, слід забезпечити оптимальну температуру (18-25°C) і відносну вологість повітря (40-60%), регулярну вентиляцію для видалення тепла та підтримання рівня кисню. Комбіновані системи вентиляції, що включають природну та механічну вентиляцію, є ефективним рішенням. Наприклад, механічна вентиляція дозволяє підтримувати стабільний повітрообмін, навіть у складних умовах, тоді як природна вентиляція може бути використана для економії енергії. Додатково варто використовувати системи моніторингу та сигналізації для контролю якості повітря. Монітори рівня вуглекислого газу можуть сигналізувати про необхідність збільшення об’єму свіжого повітря у приміщенні.

Проти правова безпека та дотримання законодавства є невід'ємною частиною роботи над забезпеченням охорони праці. Всі заходи повинні відповідати законодавству України, зокрема Законам України "Про охорону праці" та "Про основні засади забезпечення кібербезпеки України". Усі дії мають бути задокументовані відповідними актами, інструкціями та сертифікатами. Це забезпечує прозорість процесів і мінімізує ризики юридичних претензій.

Освітлення також відіграє важливу роль у створенні безпечного та комфортного середовища. У приміщеннях, де працює комп’ютерне обладнання,

освітлення має бути рівномірним, без різких перепадів яскравості, щоб зменшити навантаження на зір. Рекомендується використовувати світлодіодні джерела світла, які забезпечують тривалий термін служби та стабільну яскравість. Відповідно до ДБН В.2.5-28-2018, освітленість робочих зон повинна відповідати встановленим нормам і бути налаштованою так, щоб мінімізувати утворення тіней. Колірна температура ламп має бути в діапазоні 4000–5000 К, що найбільш комфортна для очей. Регулярне технічне обслуговування систем освітлення допомагає підтримувати їх ефективність та відповідність стандартам.

Для вдосконалення практик охорони праці та підвищення рівня безпеки на робочих місцях, доцільно орієнтуватися не лише на національні нормативні акти, але й на міжнародні стандарти, такі як ISO 45001 "Системи управління охороною здоров'я та безпеки праці". Цей стандарт є міжнародно визнаним орієнтиром, що допомагає організаціям забезпечити ефективне управління безпекою праці, зменшити ризики травм та професійних захворювань, а також покращити загальний клімат у колективі.

Згідно з ISO 45001, організації повинні розробляти та впроваджувати системи управління безпекою праці, що включають оцінку ризиків, визначення заходів щодо їх зменшення та постійний моніторинг ефективності цих заходів. Врахування цього стандарту дозволяє забезпечити інтеграцію охорони праці в загальну систему управління організацією, що сприяє підвищенню ефективності та зниженню кількості нещасних випадків.

Важливими аспектами є:

–Оцінка та контроль ризиків для здоров'я та безпеки працівників.

–Впровадження заходів щодо зменшення небезпек та покращення умов праці.

–Підвищення кваліфікації персоналу з питань охорони праці та безпеки.

–Постійний моніторинг та аудит системи для досягнення безперервного вдосконалення.

Застосування стандартів ISO допомагає забезпечити високий рівень охорони праці, відповідність міжнародним вимогам та підвищення репутації компанії на ринку.

Ці аспекти охоплюють як безпосередню роботу з комп'ютерним обладнанням, так і створення безпечного середовища для функціонування веб-сайту "My Health".

## 4.2 Безпека в надзвичайних ситуаціях

На сьогодні питання стійкості роботи промислових підприємств до впливу вторинних вражаючих факторів є одним із ключових аспектів забезпечення їх безперебійного функціонування. Вторинні вражаючі фактори, що виникають внаслідок аварій, техногенних катастроф або стихійних лих, можуть спричиняти значні загрози для виробничих процесів, персоналу та довкілля. У контексті глобалізації економіки та зростання обсягів промислового виробництва важливо забезпечити не лише ефективність діяльності підприємства, але й його здатність протидіяти кризовим ситуаціям.

Вторинні вражаючі фактори – це небезпеки, що виникають як наслідок основних аварійних подій. Їхній вплив може бути як локальним, так і значно ширшим, залежно від масштабу та специфіки виробництва. Основні типи таких факторів включають:

–Пожежі та вибухи. Вони є одними з найнебезпечніших наслідків техногенних аварій, оскільки несуть значний ризик для здоров’я і життя персоналу, а також призводять до серйозних руйнувань виробничої інфраструктури.

–Токсичні викиди. Під час аварій на підприємствах хімічної або нафтохімічної промисловості можливе виділення шкідливих речовин у повітря, ґрунт або воду, що становить загрозу для екосистем і населення прилеглих територій.

–Механічні руйнування. Обвали конструкцій, утворення уламків і падіння важкого обладнання створюють небезпеку для персоналу та можуть пошкодити суміжні ділянки виробництва.

–Порушення енергозабезпечення. Зупинка електропостачання або інших енергоресурсів (газ, вода, тепло) може паралізувати виробничі процеси, спричиняючи матеріальні втрати та загрози для збереження технологічного циклу.

Крім вищезгаданих загроз, до вторинних факторів також належать збої в системах управління, порушення комунікацій і аварії транспортної інфраструктури.

Забезпечення стійкості роботи промислових підприємств регулюється законодавством України. Одним із ключових нормативно-правових актів, що стосується цього питання, є Закон України «Про охорону праці» [25]. Цей закон встановлює основні вимоги до безпечного ведення робіт, мінімізації ризиків на підприємствах і запобігання аваріям, а також зобов’язує роботодавців розробляти та впроваджувати заходи для захисту працівників і майна у разі виникнення надзвичайних ситуацій.

Окрему увагу приділено розробці та реалізації планів локалізації та ліквідації аварій (ПЛАС), які є обов’язковими для підприємств із підвищеним рівнем небезпеки.

Основні заходи щодо підвищення стійкості роботи підприємств

Для зменшення впливу вторинних вражаючих факторів необхідно впроваджувати комплексний підхід до їх попередження та ліквідації наслідків.

–Планування реагування на надзвичайні ситуації. Кожне підприємство повинно мати детальний план дій у разі виникнення аварії або катастрофи. План має враховувати специфіку виробництва, можливі загрози та порядок евакуації персоналу.

–Інженерно-технічний захист. До таких заходів належать встановлення пожежних сигналізацій, систем автоматичного пожежогасіння, датчиків витоку

шкідливих речовин, а також модернізація конструкцій будівель для їхньої стійкості до механічних пошкоджень.

–Забезпечення резервного енергопостачання. Резервні джерела енергії, такі як дизель-генератори або акумулятори, дозволяють забезпечувати роботу критично важливого обладнання навіть у разі відключення основного енергопостачання.

–Навчання персоналу. Підготовка працівників до дій у надзвичайних ситуаціях є важливою складовою стійкості підприємства. Регулярне проведення тренувань і навчань дозволяє швидко реагувати на небезпеки та знижувати ризики.

–Технічне обслуговування обладнання. Регулярний огляд і модернізація обладнання знижують ймовірність його виходу з ладу в аварійних умовах.

Особливу увагу слід приділяти правильній організації простору на підприємствах. Наприклад, евакуаційні шляхи повинні бути вільними, а обладнання – розташоване таким чином, щоб у разі аварії мінімізувати ризик ураження працівників.

Приміщення підприємств мають бути оснащені:

–Засобами пожежогасіння та вентиляції для видалення шкідливих речовин.

–Захисними укриттями для персоналу в разі виникнення надзвичайних ситуацій.

–Аварійними вимикачами, які дозволяють швидко відключити енергозабезпечення в окремих зонах.

Оцінка стійкості роботи промислового підприємства до впливу вторинних вражаючих факторів є важливою складовою забезпечення його стабільності. Реалізація комплексу захисних заходів дозволяє мінімізувати ризики, захистити персонал, майно та навколишнє середовище, а також зберегти конкурентоспроможність підприємства навіть у кризових умовах.

У сучасному світі значна увага приділяється впровадженню інновацій для підвищення рівня безпеки на підприємствах. Використання цифрових технологій, таких як системи моніторингу на основі штучного інтелекту,

дозволяє оперативно виявляти потенційні загрози та реагувати на них у режимі реального часу. Наприклад, автоматизовані системи можуть прогнозувати можливість виникнення аварій, аналізуючи стан обладнання та умови експлуатації.

Додатково, багато підприємств впроваджують цифрові платформи для управління кризовими ситуаціями. Такі платформи дозволяють координувати дії персоналу, здійснювати аналіз поточної ситуації та забезпечувати доступ до важливої інформації в умовах надзвичайної події.

Не менш важливим є впровадження екологічно чистих технологій. Використання сучасних фільтрів, систем очищення викидів і безпечних хімічних реагентів знижує ризики токсичних викидів і їхній вплив на довкілля. Також, розробка й застосування матеріалів із підвищеною стійкістю до механічних навантажень сприяє зменшенню ймовірності руйнувань під час аварій.

Проведення оцінки стійкості роботи підприємства з використанням сучасних методів аналізу вкотре підкреслює значення системного підходу до забезпечення безпеки на виробництві. Згідно із Законом України «Про охорону праці», дотримання вимог безпеки є обов’язковим елементом діяльності кожного підприємства, і лише їхнє суворе виконання гарантує захист здоров’я працівників, ресурсів підприємства та довкілля.

Питання забезпечення стійкості роботи промислових підприємств до впливу вторинних вражаючих факторів є надзвичайно актуальним у сучасних умовах зростання техногенних ризиків. Реалізація комплексних заходів безпеки, впровадження інноваційних технологій та системний підхід до управління ризиками дозволяють мінімізувати негативний вплив аварій і катастроф. Тільки злагоджена робота всіх структур підприємства, підтримка законодавчих вимог і підвищення культури безпеки можуть забезпечити стабільне функціонування промислових об’єктів та їхній внесок у економічний розвиток країни.

# ВИСНОВКИ

У ході проведеного дослідження виконано всебічний аналіз веб-додатку "My Health", спрямований на виявлення його слабких місць та розробку рекомендацій для підвищення рівня безпеки, продуктивності та зручності використання. Результати роботи продемонстрували важливість інтеграції сучасних технологій кіберзахисту та оптимізації архітектури додатка.

Зокрема, впровадження системи веб-аплікаційного файрволу (WAF) забезпечує ефективний захист від поширених веб-атак, таких як SQL-ін'єкції, XSS, CSRF і DDoS. Завдяки WAF додаток отримав можливість не лише ідентифікувати шкідливий трафік, але й блокувати його в реальному часі, що значно підвищує його стійкість до зовнішніх загроз.

Багатофакторна автентифікація (MFA) стала ще одним ключовим компонентом у забезпеченні захисту облікових записів користувачів. Використання одноразових кодів, мобільних додатків чи біометричних даних мінімізує ризик компрометації паролів та значно ускладнює процес несанкціонованого доступу для зловмисників. Крім того, механізми блокування після багаторазових невдалих спроб входу та шифрування даних дозволяють забезпечити високий рівень надійності системи.

Ще одним кроком до підвищення ефективності веб-додатку стало розгортання додаткового сервера. Це рішення сприяє рівномірному розподілу навантаження між серверами, забезпечує масштабованість додатка та зменшує час відгуку під час високих навантажень. Завдяки цьому користувачі отримують швидкий доступ до ресурсу навіть у пікові періоди його використання.

Проведений аудит, зокрема, включав:

–ревізію архітектури додатка та бази даних, що дозволило виявити слабкі місця в організації компонентів та логіці взаємодії між ними;

–тестування за допомогою сучасних інструментів, таких як OWASP ZAP, SQLmap та Postman, для ідентифікації вразливостей;

–оцінку відповідності системи міжнародним стандартам, включаючи GDPR та HIPAA.

У результаті проведеного аналізу сформульовано низку рекомендацій, серед яких:

–фільтрація вхідних даних для запобігання XSS-атакам;

–використання підготовлених SQL-запитів для захисту бази даних;

–реалізація механізму CSRF-токенів для запобігання міжсайтовим запитам;

–впровадження алгоритмів хешування паролів, таких як bcrypt.

Таким чином, впровадження запропонованих удосконалень дозволяє створити стабільну, масштабовану та захищену систему, яка відповідає сучасним вимогам кібербезпеки та покращує досвід користувачів.

# СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ


1. T. Lechachenko, R. Kozak, Y. Skorenkyy, O. Kramar, O. Karelina. Cybersecurity Aspects of Smart Manufacturing Transition to Industry 5.0 Model. CEUR Workshop Proceedings, 2023, 3628, pp. 325–329
2. ZAGORODNA, N., STADNYK, M., LYPA, B., GAVRYLOV, M., & KOZAK, R. (2022). Network Attack Detection Using Machine Learning Methods. Challenges to national defence in contemporary geopolitical situation, 2022(1), 55-61.
3. The etalon models of linguistic variables for sniffing-attack detection / M. Karpinski et al. 2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS), Bucharest, 21–23 September 2017.. URL:https://doi.org/10.1109/idaacs.2017.8095087 (date of access: 15.12.2024).
4. Use of augmented reality-enabled prototyping of cyber-physical systems for improving cyber-security education / Y. Skorenkyy et al. Journal of Physics: Conference Series. 2021. Vol. 1840, no. 1. P. 012026. URL: https://doi.org/10.1088/1742-6596/1840/1/012026 (date of access: 15.12.2024).
5. The improvement of web-application SDL process to prevent Insecure Design vulnerabilities / O. A. Revniuk et al. Applied Aspects of Information Technology. 2024. Vol. 7, no. 2. P. 162–174. URL: https://doi.org/10.15276/aait.07.2024.12 (date of access: 15.12.2024).
6. Lypa, B., Horyn, I., Zagorodna, N., Tymoshchuk, D., Lechachenko T., (2024). Comparison of feature extraction tools for network traffic data. CEUR Workshop Proceedings, 3896, pp. 1-11.
7. The ZAP Homepage. ZAP. URL: https://www.zaproxy.org/ (date of access: 15.12.2024).
8. Tymoshchuk, D., Yasniy, O., Mytnyk, M., Zagorodna, N., Tymoshchuk, V., (2024). Detection and classification of DDoS flooding attacks by machine learning methods. CEUR Workshop Proceedings, 3842, pp. 184 - 195.

9. Acunetix | Web Application and API Security Scanner. Acunetix. URL: https://www.acunetix.com/ (date of access: 15.12.2024).
10. ТИМОЩУК, Д., ЯЦКІВ, В., ТИМОЩУК, В., & ЯЦКІВ, Н. (2024). INTERACTIVE CYBERSECURITY TRAINING SYSTEM BASED ON SIMULATION ENVIRONMENTS. MEASURING AND COMPUTING DEVICES IN TECHNOLOGICAL PROCESSES, (4), 215-220.
11. Maziriri T. ISO/IEC 27001 Advanced Lead Implementer's Guide. Independently Published, 2020.
12. Manaseer S., Al Hwaitat A. K. Centralized Web Application Firewall Security System. Modern Applied Science. 2018. Vol. 12, no. 10. P. 164. URL: https://doi.org/10.5539/mas.v12n10p164 (date of access: 15.12.2024).
13. ТИМОЩУК, Д., & ЯЦКІВ, В. (2024). USING HYPERVISORS TO CREATE A CYBER POLYGON. MEASURING AND COMPUTING DEVICES IN TECHNOLOGICAL PROCESSES, (3), 52-56.
14. Boonkrong S. Multi-factor Authentication. Authentication and Access Control. Berkeley, CA, 2020. P. 133–162. URL: https://doi.org/10.1007/978-1-4842-6570-3_6 (date of access: 15.12.2024).
15. Prema Sindhuri B., Kameswara Rao M. IoT security through web application firewall. International Journal of Engineering & Technology. 2018. Vol. 7, no. 2.7. P. 58. URL: https://doi.org/10.14419/ijet.v7i2.7.10259 (date of access: 15.12.2024).
16. Tymoshchuk, D., & Yatskiv, V. (2024). Slowloris ddos detection and prevention in real-time. Collection of scientific papers «ΛΌГОΣ», (August 16, 2024; Oxford, UK), 171-176.
17. Cloud Computing Services - Amazon Web Services (AWS). Amazon Web Services, Inc. URL: https://aws.amazon.com/ (date of access: 15.12.2024).
18. Junker H. OWASP Enterprise Security API. Datenschutz und Datensicherheit - DuD. 2012. Vol. 36, no. 11. P. 801–804. URL: https://doi.org/10.1007/s11623-012-0275-3 (date of access: 15.12.2024).

19. The full-stack JavaScript framework for real-time apps - Meteor.js. Meteor Software - Build with Meteor.js, deploy on Galaxy. URL: https://www.meteor.com/ (date of access: 15.12.2024).
20. Tymoshchuk, V., Dolinskyi, A., & Tymoshchuk, D. (2024). MESSENGER BOTS IN SMART HOMES: COGNITIVE AGENTS AT THE FOREFRONT OF THE INTEGRATION OF CYBER-PHYSICAL SYSTEMS AND THE INTERNET OF THINGS. Матеріали конференцій МЦНД, (07.06. 2024; Луцьк, Україна), 266-267.
21. React Native · Learn once, write anywhere. React Native · Learn once, write anywhere. URL: https://reactnative.dev/ (date of access: 15.12.2024).
22. Laybats C., Davies J. GDPR. Business Information Review. 2018. Vol. 35, no. 2. P. 81–83. URL: https://doi.org/10.1177/0266382118777808 (date of access: 15.12.2024).
23. ДБН В.2.5-67:2013 "Опалення, вентиляція та кондиціонування" : від 28.08.2013.URL:https://econstruction.gov.ua/laws_detail/3074971619479783152?doc_type=2 (дата звернення: 15.12.2024).
24. Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин: ДСанПіН 3.3.2.007-98. – [Чинний від 10.12.1998]. – Режим доступу: URL: http://mozdocs.kiev.ua/view.php?id=2445. (дата звернення: 15.12.2024).
25. Законодавство України про охорону праці : У 3-х т. Збірник норм. документів. Київ, 1997. Т. 1. 558 с.

Додаток А Публікація

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**
**ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ**
**імені ІВАНА ПУЛЮЯ**
**НАУКОВЕ ТОВАРИСТВО ім. ШЕВЧЕНКА**

**М А Т Е Р І А Л И**

**XII НАУКОВО-ТЕХНІЧНОЇ КОНФЕРЕНЦІЇ**

# «ІНФОРМАЦІЙНІ МОДЕЛІ, СИСТЕМИ ТА ТЕХНОЛОГІЇ»


**18-19 грудня 2024 року**

**ТЕРНОПІЛЬ**
**2024**

УДК 001
М34

## ПРОГРАМНИЙ КОМІТЕТ

**Голова:** Приймак Микола – професор кафедри комп'ютерних систем та мереж, д.т.н., професор.
**Співголови:** Марущак Павло – проректор з наукової роботи, докт. техн. наук, професор.
Баран Ігор – канд. техн. наук, доцент, декан факультету ФІС.
**Науковий секретар:** Надія Крива– старший викладач.

**Члени:** Василь Кривень - завідувач кафедри математичних методів в інженерії д.ф.-м.н., професор; Галина Осухівська – завідувач кафедри комп'ютерних систем та мереж, к.т.н., доцент; Микола Карпінський - професор кафедри кібербезпеки, д.т.н., професор; Жанна Баб'як - завідувач кафедри української та іноземних мов, к.пед. н., доцент; Ярослав Литвиненко – професор кафедри комп'ютерних наук, д.т.н., професор; Михайло Петрик - завідувач кафедри програмної іженерії, д.ф.-м.н., професор; Наталія Загородна – завідувач кафедри кібербезпеки, к.т.н., доцент.

## ОРГАНІЗАЦІЙНИЙ КОМІТЕТ

**Голова:** Скоренький Юрій Любомирович – канд. техн. наук, доцент кафедри фізики.
**Члени:** доцент кафедри комп'ютерних наук, к.т.н. В. Никитюк; доцент кафедри програмної інженерії, к.т.н. Д. Михалик; доцент кафедри кібербезпеки, к.т.н. М. Стадник; доцент кафедри комп'ютерних систем та мереж, к.т.н. Є. Тиш; ст. викладач Л. Джиджора.

Матеріали XII науково-технічної конфції «Інформаційні моделі, системи та технології» Тернопільського національного технічного університету імені Івана Пулюя, (Тернопіль, 18–19 грудня 2024 р.). – Тернопіль : Тернопільський національний технічний університет імені Івана Пулюя, 2024.

М34

**Адреса оргкомітету:** ТНТУ ім. І. Пулюя, м. Тернопіль, вул. Руська, 56, 46001,
тел. (0352) 52-41-33, факс (0352) 25-49-83.
E-mail: conffis2024@gmail.com
Редагування, оформлення та верстка: Надія Крива

## СЕКЦІЇ КОНФЕРЕНЦІЇ, ЯКІ ПРЕДСТВЛЕНІ В ЗБІРНИКУ

- Математичне моделювання
- Інформаційні системи та технології, кібербезпека
- Комп'ютерні системи та мережі
- Програмна інженерія та моделювання складних розподілених систем
- Новітні фізико-технічні та освітні технології

В збірнику надруковано тези доповідей XII науково-технічної конференції «Інформаційні моделі, системи та технології» (Тернопіль, 18–19 грудня 2024 р.) за такими науковими напрямками: математичне моделювання; інформаційні системи та технології, кібербезпека; комп'ютерні системи та мережі; програмна інженерія та моделювання складних розподілених систем; новітні фізико-технічні та освітні технології.

Розрахований на науковців, викладачів та студентів вузів.

**За зміст тез та дотримання норм академічної доброчесності відповідальність несе автор**.

**УДК 004.4**

**А.Р. Комендат, Т.А. Лечаченко**
(Тернопільський національний технічний університет імені Івана Пулюя, Україна)

## АНАЛІЗ ТЕХНОЛОГІЙ ЗАХИСТУ ВЕБ-САЙТІВ

**A.R. Komendat, T.A. Lechachenko**

## ANALYSIS OF WEB SITE PROTECTION TECHNOLOGIES

Веб-сайти є основою сучасного цифрового середовища, об'єднуючи бізнес, науку, освіту, розваги та інші сфери діяльності. Зі зростанням значення цифрової присутності для компаній і організацій, удосконалення веб-сайтів стало ключовим завданням для забезпечення їх ефективності, доступності та безпеки. Інформаційні технології (ІТ) виступають рушійною силою цих змін, забезпечуючи інструменти для всебічного аналізу, оптимізації та захисту веб-ресурсів.

Зростання кількості атак на веб-сайти, спрямованих на викрадення даних, злам систем або виведення ресурсів із ладу, вимагає впровадження сучасних рішень для забезпечення безпеки. Системи WAF (Web Application Firewall) дозволяють блокувати підозрілі запити в реальному часі, запобігаючи SQL-ін'єкціям, XSS-атакам та іншим загрозам. Використання SSL/TLS забезпечує шифрування даних, що передаються між сервером і клієнтом, запобігаючи їх перехопленню[1].

Одним із найбільш ефективних способів захисту користувачів веб-сайту – є впровадження багатофакторної автентифікації. Наприклад, користувач під час входу на сайт вводить свій логін і пароль, після чого система запитує одноразовий код, згенерований додатком Google Authenticator. У разі компрометації пароля зловмисник не зможе отримати доступ без цього додаткового коду[2].

Впровадження штучного інтелекту в системи кібербезпеки відкриває нові можливості для захисту веб-ресурсів. Алгоритми машинного навчання дозволяють виявляти нові типи загроз, аналізуючи поведінку трафіку та патерни атак. Наприклад, технології Deep Packet Inspection, що застосовуються у продуктах компаній, таких як Palo Alto Networks, здатні розпізнавати складні кіберзагрози[3].

Удосконалення захисту веб-сайтів за допомогою сучасних інформаційних технологій є основою їхньої стабільної роботи та довіри користувачів. Використання автоматизованих систем, машинного навчання, регулярний аудит і навчання персоналу забезпечують багаторівневий підхід до безпеки, що дозволяє запобігти більшості сучасних загроз. Тільки комплексні заходи зможуть гарантувати ефективність, надійність і захист веб-ресурсів у цифрову епоху.

### Література

1. → Гришина І. П., Коваленко А. С. "Сучасні методи забезпечення безпеки веб-додатків: теоретичний аналіз і практичні рекомендації." — [Електронний ресурс]: journals.dut.edu.ua/index.php/dataprotect. Доступ: 05.12.2024. У статті розглядаються основні технології, включаючи SSL/TLS та WAF, як ефективні інструменти для запобігання сучасним кіберзагрозам

2. → Добринін І.С., Пшеничних С.В. "Аналіз методів автентифікації для вебзастосунків та реалізація системи автентифікації." — [Електронний ресурс]: ITSSI Journal. Доступ: 05.12.2024. У статті досліджуються сучасні технології багатофакторної автентифікації, включаючи Google Authenticator.

3. → Савченко В. А., Шаповаленко О. Д. "Основні напрями застосування технологій штучного інтелекту у кібербезпеці." — [Електронний ресурс]: journals.dut.edu.ua/index.php/sciencenotes. Доступ: 05.12.2024.

Додаток Б Лістинг файлу authentication.controller.ts

```
import {
  Body,
  Req,
  Controller,
  HttpCode,
  Post,
  UseGuards,
  Res,
  Get,
  Logger,
  UsePipes,
  ValidationPipe,
} from '@nestjs/common'
import { ApiBody } from '@nestjs/swagger'
import { Response } from 'express'
import { UsersService } from '../users/users.service'
import { AuthenticationService } from './authentication.service'
import LogInDto from './dto/logIn.dto'
import RegisterDto from './dto/register.dto'
import JwtAuthenticationGuard from './jwt-authentication.guard'
import JwtRefreshGuard from './jwt-refresh.guard'
import { LocalAuthenticationGuard } from
'./localAuthentication.guard'
import RequestWithUser from './requestWithUser.interface'
@Controller('auth')
export class AuthenticationController {
        private readonly logger = new
Logger(AuthenticationService.name)
  constructor(
            private readonly authenticationService:
AuthenticationService,
    private readonly usersService: UsersService,
  ) {}
  @Post('register')
```

```
      async register(@Body() registrationData: RegisterDto, @Res()
response: Response) {

        return response.send({
          message: 'Multifactor authentication required',
          mfa_token: accessToken,
          refreshToken,
          data: { ...userInfo },
        })
      }

      @HttpCode(200)
      @UseGuards(LocalAuthenticationGuard)
      @Post('login')
      @ApiBody({ type: LogInDto })
      @UsePipes(ValidationPipe)
      async logIn(@Req() request: RequestWithUser, @Res() response:
Response) {
        const { user } = request
                              const           accessToken           =
this.authenticationService.getJwtToken(String(user.id))
                    response.setHeader('Authorization',       `Bearer
${accessToken}`)
        this.logger.log('[login]', user)

        if (user.isMfaAuthEnabled) {
          console.log(3)
          return response.send({
            title: 'mfa_required',
            description: 'Multifactor authentication required',
            mfa_token: accessToken,
            refreshToken,
          })
        }
            return   response.send({   id:   user.id,   accessToken,
refreshToken })
```

```
    }

    @UseGuards(JwtAuthenticationGuard)
    @Get('logout')
    async logOut(@Req() request: RequestWithUser, @Res() response:
Response) {
      await this.usersService.removeRefreshToken(request.user.id)
      return response.sendStatus(200)
    }

    @UseGuards(JwtRefreshGuard)
    @Get('refresh')
    refresh(@Req() request: RequestWithUser) {
                          const          refreshToken          =
this.authenticationService.getJwtRefreshToken(String(request.user.
id))
                           const          accessToken          =
this.authenticationService.getJwtToken(String(request.user.id))
      const { id } = request.user
      return {
        id,
        accessToken,
        refreshToken,
      }
    }
  }
  #include <ESP8266WiFi.h>
  #include <PubSubClient.h>
  #include <DHT.h>
  #define DHTPIN D1
  #define DHTTYPE DHT11
  #define WATER_SENSOR_PIN A0
  #define LIGHT_SENSOR_PIN D3
  const char* ssid = "";
  const char* password = "";
  const char* mqtt_server = "192.168.0.104";
```

# Додаток В Лістинг файлу authentication.service.ts

```
    import { HttpException, HttpStatus, Injectable, Logger, Body }
from '@nestjs/common'
    import { ConfigService } from '@nestjs/config'
    import { JwtService } from '@nestjs/jwt'
    import * as bcrypt from 'bcrypt'

    import { UsersService } from '../users/users.service'
    import RegisterDto from './dto/register.dto'
    import   {   MfaAuthenticationService   }   from   './mfa/mfa-
auth.service'
    import TokenPayload from './tokenPayload.interface'

    @Injectable()
    export class AuthenticationService {
            private       readonly       logger       =       new
Logger(AuthenticationService.name)
      constructor(
                private     readonly     mfaAuthenticationService:
MfaAuthenticationService,
        private readonly usersService: UsersService,
        private readonly jwtService: JwtService,
        private readonly configService: ConfigService,
      ) {}

      public async register(@Body() registrationData: RegisterDto)
{

          this.logger.log('[createdUser]', createdUser)
          return createdUser
        } catch (error) {
             throw   new   HttpException('Something   went   wrong',
HttpStatus.INTERNAL_SERVER_ERROR)
        }
      }
```

```
      public getJwtToken(userId: string) {
        const payload: TokenPayload = { userId }

        const accessToken = this.jwtService.sign(payload, {
                                                          expiresIn:
`${this.configService.get('JWT_EXPIRATION_TIME')}s`,
        })
        return accessToken
      }

          public      getJwtAccessTokenWithMFA(userId:      string,
isSecondFactorAuthenticated = false) {
             const    payload:    TokenPayload    =    {    userId,
isSecondFactorAuthenticated }
        const accessToken = this.jwtService.sign(payload, {
          secret: this.configService.get('JWT_SECRET'),
                                                          expiresIn:
`${this.configService.get('JWT_EXPIRATION_TIME')}s`,
        })
        return accessToken
      }

      public getJwtRefreshToken(userId: string) {
        const payload: TokenPayload = { userId }
        const refreshToken = this.jwtService.sign(payload, {
                                                //            secret:
this.configService.get('JWT_REFRESH_TOKEN_SECRET'),
                                                          expiresIn:
`${this.configService.get('JWT_REFRESH_TOKEN_EXPIRATION_TIME')}s`,
        })

        return refreshToken
      }
```

```
        public async getAuthenticatedUser(email: string,
plainTextPassword: string) {
        try {
          console.log('email')
                      const authenticatedUser = await
this.usersService.getByEmail(email)
          console.log('pass')
                    await this.verifyPassword(plainTextPassword,
authenticatedUser.password)
          return authenticatedUser
        } catch (error) {
            throw new HttpException('Wrong credentials provided',
HttpStatus.BAD_REQUEST)
        }
      }

       private async verifyPassword(plainTextPassword: string,
hashedPassword: string) {
        console.log(plainTextPassword, hashedPassword)
                  const isPasswordMatching = await
bcrypt.compare(plainTextPassword, hashedPassword)
        if (!isPasswordMatching) {
            throw new HttpException('Wrong credentials provided',
HttpStatus.BAD_REQUEST)
        }
      }
    }
```

## Додаток Г Лістинг файлу authentication.module.ts

```
import { Module } from '@nestjs/common'
import { ConfigModule, ConfigService } from '@nestjs/config'
import { JwtModule } from '@nestjs/jwt'
import { PassportModule } from '@nestjs/passport'

import { UsersModule } from '../users/users.module'
import { UsersService } from '../users/users.service'
import { AuthenticationController } from './authentication.controller'
import { AuthenticationService } from './authentication.service'
import { JwtMfaStrategy } from './jwt-mfa.strategy'
import { JwtRefreshTokenStrategy } from './jwt-refresh-token.strategy'
import { JwtStrategy } from './jwt.strategy'
import { LocalStrategy } from './local.strategy'
import { MfaAuthenticationController } from './mfa/mfa-auth.controller'
import { MfaAuthenticationService } from './mfa/mfa-auth.service'

@Module({
  imports: [
    UsersModule,
    PassportModule,
    ConfigModule,
    JwtModule.registerAsync({
      imports: [ConfigModule],
      inject: [ConfigService],
      useFactory: async (configService: ConfigService) => ({
        secret: configService.get('JWT_SECRET'),
        signOptions: {
          expiresIn:
`${configService.get('JWT_EXPIRATION_TIME')}s`,
export class AuthenticationModule {}
```

Додаток Д Лістинг файлу main.ts

```
import { ConfigService } from '@nestjs/config'
import { NestFactory } from '@nestjs/core'
import helmet from 'helmet'

import { AppModule } from './app.module'
import SetupSwagger from './utils/swagger'

async function bootstrap() {
  const app = await NestFactory.create(AppModule)
  const configService = app.get(ConfigService)
  app.use(helmet())

  SetupSwagger(app)

  const PORT = configService.get('PORT') ?? 3000
  console.log(` The magic happens at: http://localhost:${PORT}`)
  await app.listen(PORT)
}
bootstrap()
```