ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
КАФЕДРА ІНЖЕНЕРІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ АВТОМАТИЗОВАНИХ СИСТЕМ

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «ДОСЛІДЖЕННЯ АЛГОРИТМІВ АУТЕНТИФІКАЦІЇ КОРИСТУВАЧА У WEB-ДОДАТКАХ»

на здобуття освітнього ступеня магістра
зі спеціальності 126 Інформаційні системи та технології
(код, найменування спеціальності)
освітньо-професійної програми Інформаційні системи та технології
(назва)

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

_____ Денис ДОВГОРУК
(підпис) *Ім'я, ПРІЗВИЩЕ здобувача*

Виконав:
здобувач вищої освіти Денис ДОВГОРУК
група ІСДМ-62
Керівник: Вадим ВЛАСЕНКО
науковий ступінь, вчене звання доцент ПУАД
Рецензент:
науковий ступінь, вчене звання _____
Ім'я, ПРІЗВИЩЕ

Київ 2023

ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ

Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення автоматизованих систем

Ступінь вищої освіти Магістр

Спеціальність Інформаційні системи та технології

Освітньо-професійна програма Інформаційні системи та технології

ЗАТВЕРДЖУЮ
Завідувач кафедрою ІПЗАС
_____ Каміла СТОРЧАК
«_____» _____ 2023 р.

ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ

Довгоруку Денису Олеговичу

(прізвище, ім'я, по батькові здобувача)

1. Тема кваліфікаційної роботи: Дослідження алгоритмів аутентифікації користувача у WEB-додатках
керівник кваліфікаційної роботи Вадим ВЛАСЕНКО доцент каф. ПУАД,
(Ім'я, ПРІЗВИЩЕ науковий ступінь, вчене звання)
затверджені наказом Державного університету інформаційно-комунікаційних технологій від «19» 10.2023р. №145
2. Строк подання кваліфікаційної роботи «29» грудня 2023р.

3. Вихідні дані до кваліфікаційної роботи: науково-технічна література, рішення безпеки у веб-додатках, методи побудови комплексної системи безпеки для веб-додатку, методи автентифікації.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

Огляд поточного стану захисту WEB-додатків

Аналіз основних загроз та вразливостей у сучасних веб-додатках

Сучасні методи автентифікації

5. Перелік графічного матеріалу: *презентація*

6. Дата видачі завдання «19» жовтня 2023 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Отримання завдання	19.10-05.11.23	Виконано
2	Аналіз наявної науково-технічної літератури	05.11-12.11.23	Виконано
3	Аналіз предметної області	13.11-19.11.23	Виконано
4	Проведення теоретичних досліджень	20.11-25.11.23	Виконано
5	Розробка теоретичного розділу роботи	27.11-03.12.23	Виконано
6	Розробка практичного розділу роботи	04.12-10.12.23	Виконано
7	Оформлення роботи: вступ, висновки, реферат	11.12-20.12.23	Виконано
8	Розробка демонстраційних матеріалів	21.12-29.12.23	Виконано

Здобувач вищої освіти _____ Денис ДОВГОРУК

(підпис) *(Ім'я, ПРІЗВИЩЕ)*

Керівник

кваліфікаційної роботи _____ Вадим ВЛАСЕНКО

(підпис) *(Ім'я, ПРІЗВИЩЕ)*

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра: 73 стор., 27 рис., 33 джерела.

Мета роботи – надання рекомендацій, які можна застосувати для посилення автентифікації користувачів при створенні веб-додатків, забезпечуючи баланс між підвищеною безпекою та позитивним користувацьким досвідом.

Об'єкт дослідження – виявлення недоліків і проблем традиційних методів автентифікації.

Предмет дослідження – безпека WEB-додатків.

Короткий зміст роботи: Проведено порівняльний аналіз різних алгоритмів автентифікації користувачів з точки зору безпеки, ефективності та зручності використання. Визначено сильні та слабкі сторони різних алгоритмів у різних сценаріях. Проведено оцінку стійкості розширених механізмів автентифікації проти типових атак та кіберзагроз.

КЛЮЧОВІ СЛОВА: АЛГОРИТМИ АВТЕНТИФІКАЦІЇ, ВЕБ-СЕРЕДОВИЩЕ, КОНФІДЕНЦІЙНІСТЬ ДАНИХ, БЕЗПЕКА ДАНИХ

ABSTRACT

Text part of the master's qualification work: 73 pages, 27 pictures, 33 sources.

The purpose of the work providing recommendations that can be applied to strengthen user authentication when building web applications, balancing increased security with a positive user experience.

Object of research – identifying shortcomings and problems of traditional authentication methods.

Subject of research – security of WEB applications.

Summary of the work: In the work a comparative analysis of various user authentication algorithms was conducted from the point of view of security, efficiency and usability. Strengths and weaknesses of different algorithms in different scenarios are determined. The stability of advanced authentication mechanisms against typical attacks and cyberthreats was assessed.

KEYWORDS: AUTHENTICATION ALGORITHMS, WEB ENVIRONMENT, DATA PRIVACY, DATA SECURITY

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

2FA - 2-factor authentication

MFA - multi-factor authentication

XSS - Cross-Site Scripting

OTP - One Time Password

PKI - Public key infrastructure

IoT - Internet of Things

ВСТУП

Веб-програми, від платформ електронної комерції до сайтів соціальних мереж, змінили спосіб взаємодії людей, ведення бізнесу та доступу до інформації. Поширення цих програм підкреслює критичну роль, яку вони відіграють у сучасному суспільстві.

Автентифікація користувача служить першою лінією захисту від несанкціонованого доступу, захищаючи дані користувача, фінансову інформацію та конфіденційні ресурси, гарантуючи, що лише законні користувачі отримають доступ до важливої інформації. Оскільки веб-додатки продовжують поширюватися, уразливості, пов’язані з традиційними методами автентифікації, такими як комбінації імені користувача та пароля, стають більш очевидними. Зростання кількості кіберзагроз, включаючи фішингові атаки, перекид облікових даних і атаки грубою силою, підкреслює необхідність більш безпечних механізмів автентифікації.

У сфері автентифікації користувачів для веб-додатків залишається кілька проблем. Ненадійні паролі, повторне використання паролів і недбалість користувачів залишаються поширеними проблемами. Крім того, такі традиційні методи, як двофакторна автентифікація (2FA), стикаються з проблемами, такими як незручності, яких відчувають користувачі, і потенційні атаки під час заміни SIM-карти. Встановлення балансу між безпекою та досвідом користувача — це постійна боротьба при розробці систем автентифікації.

У відповідь на недоліки традиційних методів автентифікації дослідники та розробники досліджують передові алгоритми для підвищення безпеки. Біометрична автентифікація, яка включає сканування відбитків пальців, розпізнавання обличчя та розпізнавання голосу, пропонує більш безпечну та зручну альтернативу. Алгоритми машинного навчання відіграють вирішальну роль у визначенні моделей поведінки користувачів, дозволяючи системам виявляти аномалії та потенційні загрози безпеці.

Основні цілі дипломної роботи полягають у аналізі проблем, ретельному аналізі сильних і слабких сторін нових технологій і пропозиції до стратегій розвитку для ефективного впровадження безпечних механізмів автентифікації. Завдяки ретельному дослідженню, аналізу та практичному впровадженню це дослідження має на меті внести цінний внесок у поточний дискурс щодо безпеки веб-додатків.

1 ОГЛЯД ПОТОЧНОГО СТАНУ WEB-ДОДАТКІВ

1.1 Огляд понять

Веб-додаток – це програмне забезпечення або програма, яку можна відкрити за допомогою будь-якого браузера. Зовнішній інтерфейс веб програми розробляється за допомогою таких мов програмування: HTML, CSS, Javascript, які підтримуються на будь-якому браузері. У той час як для написання серверної частини (Back-end) може використовуватися будь-яка інша мова програмування або фреймворк, Python, PhP, Ruby, Java.

Веб-програми мають архітектуру клієнт-сервер. Їх код поділено на два компоненти: скрипти на стороні клієнта та скрипти на стороні сервера.

Архітектура на стороні клієнта

Скрипт на стороні клієнта відповідає за функціональність інтерфейсу користувача, наприклад, кнопки і поля, що випадають. Коли кінцевий користувач натискає на веб-програму, веб-браузер завантажує скрипт на стороні клієнта і відображає графічні елементи та текст для взаємодії з користувачем.

Наприклад, користувач може читати контент, переглядати відео або заповнювати дані у формі контакту. Такі дії, як натискання кнопки надсилання, передаються на сервер у вигляді запиту клієнта.

Архітектура на стороні сервера

Скрипт на стороні сервера обробляє дані. Сервер веб-додатків обробляє запити клієнтів та надсилає відповідь. Зазвичай запити стосуються отримання додаткових даних або зміни чи збереження нових даних.

Наприклад, якщо користувач натискає кнопку Детальніше, сервер веб-додатків відправляє контент назад користувачу. Якщо натиснути кнопку Надіслати, сервер додатків збереже дані користувача в базі даних. У деяких випадках сервер завершує запит даних і відправляє повну HTML сторінку назад клієнту. Це називається рендерингом сервера.

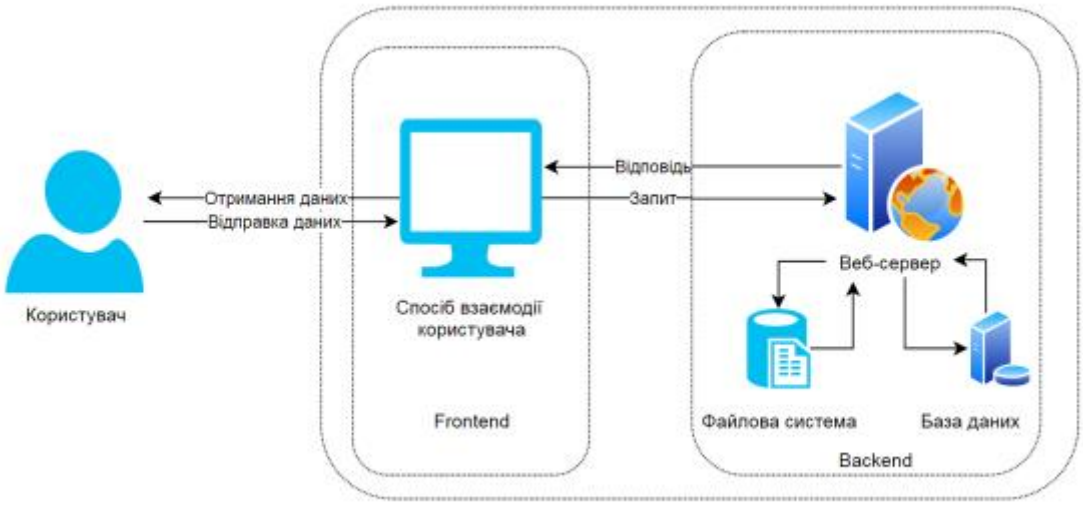

Рисунок 1.1 - Архітектура web-додатку

Коріння веб-додатків сягають ранніх днів Інтернету, коли домінували статичні сторінки HTML. Поява динамічних веб-сторінок, які працюють на мовах сценаріїв на стороні сервера, ознаменувала значний крок вперед. Еволюція продовжилася з розвитком AJAX (асинхронного JavaScript і XML), який дозволив безперебійний обмін даними між браузером користувача та сервером, забезпечивши більш інтерактивний і оперативний досвід користувача. Сьогодні сучасні веб-додатки використовують фреймворки та бібліотеки, такі як React, Angular і Vue.js, для створення високоскладних і ефективних інтерфейсів користувача.

Веб-додатки революціонізували взаємодію з користувачем, забезпечивши неперевершену доступність та інтерактивність. Від платформ соціальних медіа, які об’єднують людей у всьому світі, до сайтів електронної комерції, які переосмислюють досвід покупок, веб-додатки стали провідниками, через які ми орієнтуємося в цифровий світ і взаємодіємо з ним. Принципи адаптивного дизайну забезпечують безперебійну роботу користувачів на різних пристроях, дозволяючи користувачам однаково легко отримувати доступ до інформації та послуг на смартфонах, планшетах і настільних комп’ютерах.

Вони стали стрижнем цифрової ери, глибоко впливаючи на те, як ми взаємодіємо, здійснюємо транзакції та співпрацюємо. Їхня еволюція від

статичних веб-сторінок до динамічного інтерактивного досвіду відображає невпинне прагнення до більш бездоганного та доступного цифрового середовища.

Аутентифікація в свою чергу — це основа безпеки будь-якої системи, яка полягає в перевірці достовірності даних про користувача сервером.

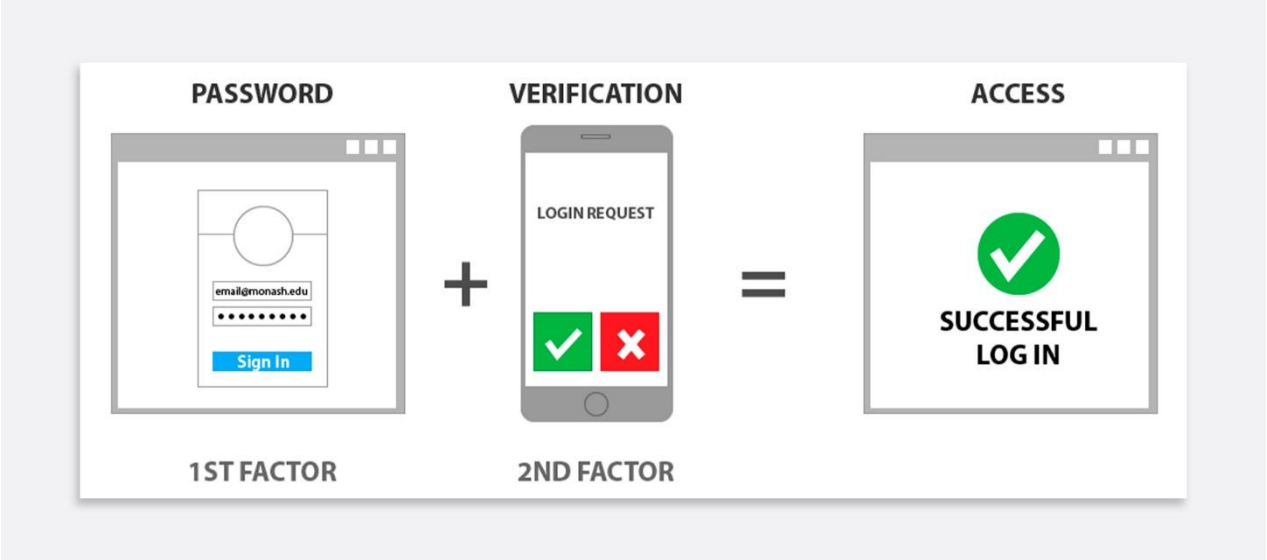

Рисунок 1.2 - Короткий принцип роботи аутентифікації

Коли клієнти вперше підключаються до типової веб-програми, вони надають ім’я користувача та пароль для програми. Потім веб-додаток виконує перевірку автентифікації, гарантуючи, що ім’я користувача та пароль є дійсними. Після перевірки облікових даних користувача веб-програма створює сеанс входу для користувача.

Це дозволяє користувачеві отримати доступ до веб-програми без необхідності входити в систему кожного разу, коли відкривається нова сторінка. Сеанси входу створюються шляхом аутентифікації інформації безпосередньо в файлі cookie, який повертається до користувача або шляхом збереження інформації про автентифікацію у файлі сеансу, що зберігається на сервері, і поверненням файлу cookie користувача, що містить випадковий унікальний ідентифікатор сеансу.

Таким чином, запит користувача вважається автентифікованим, якщо запит містить файл cookie з дійсною інформацією автентифікації чи ідентифікатором сеансу, або якщо він безпосередньо містить дійсне ім'я користувача та пароль.

Після того, як програма встановить сеанс входу для користувача, вона дозволяє користувачеві надсилати запити, такі як публікація коментарів у блозі, які можуть вставити рядок у таблицю бази даних або завантаження зображення, для чого може знадобитися запис файлу на сервер. Проте є семантичний розрив між механізмом автентифікації користувача, реалізованим веб-додатком, і доступом механізм контролю або авторизації, реалізовані нижні рівні, такі як база даних SQL або файлова система.

Нижні рівні в системі зазвичай не мають поняття користувачів прикладного рівня; натомість операції з базою даних і файлами зазвичай виконуються з привілеями та облікові дані самої веб-програми.

Користувач проходить автентифікацію, коли надані облікові дані дійсні та адекватні. Насправді автентифікація не визначає, якій сутності слід надати доступ, а лише перевіряє, чи користувачі те, що вони претендують на себе. Отже, лише після автентифікації користувачі отримають право доступу до ресурсів на основі своїх визначені привілеї.

Це загальна вимога для сучасних веб-додатків, оскільки багато, якщо не всі служби потребують персоналізації. Включаючи веб-програми для соціальних мереж. Їх особливість полегшує обмін даними для користувачів і може синхронізуватися зі смартфоном або комп’ютером користувача. Завдяки збільшення кількості цих служб в Інтернеті, автентифікація входу стає мішенню для зловмисників. Тому є потреба в безпечній та ефективній схемі автентифікації входу в веб-додатки для забезпечення контролю доступу користувачів, безпеки та конфіденційності.

Стандартний спосіб автентифікації користувача, як ім’я користувача та пароль, уже недостатньо потужний, щоб гарантувати контроль доступу, безпека та конфіденційність.

Забезпечення надійних методів автентифікації, включаючи багатофакторну автентифікацію (MFA), є критично важливим. Необхідно вживати заходів контролю доступу, щоб обмежити привілеї користувачів і зменшити ризик несанкціонованого доступу.

1.2 Наслідки нехтування безпекою

Згідно звіту Allianz Risk Barometer за 2023 рік кіберінциденти - це найбільший привід для хвилювань компаній другий рік поспіль.

Рисунок 1.3 – Рейтинг найбільших ризиків для бізнесу

Кіберінциденти, такі як збої в IT, атаки програм-вимагачів або витоки даних, вже другий рік поспіль вважаються найважливішими ризиками в усьому світі. Ризик кіберінцидентів також вважається найбільшою небезпекою в 19 країнах, зокрема, в Канаді, Франції, Японії, Індії та Великобританії. Це ризик, який є найбільшим болем маленьких компаній (<$250 млн. річного доходу).

Оскільки цифрові підприємства освоюють переваги електронного бізнесу, використання веб-технологій продовжуватиме зростати. Компанії сьогодні використовують Інтернет як канал CRM (Customer Relationship Management), як засіб покращення ланцюжків поставок, засіб виходу на нові ринки, а також для надання продуктів і послуг клієнтам і співробітникам. Однак успішного впровадження з використанням веб-технологій неможливо досягти без послідовного підходу до безпеки веб-додатків. Розглядаючи безпеку веб-додатків, корпорації часто не враховують наступне:

• Сьогодні хакери на крок попереду підприємств (хакери використовують новітні програми, а компанії мають проблеми з налаштуванням базової безпеки заходи).

• Питання не в тому, ЧИ ваш сайт буде атаковано, а КОЛИ.

• Паролей недостатньо.

• SSL і шифрування даних недостатньо (SSL гарантує секретність трафіку, але не захищає від неправильного використання веб-додатків).

• Брандмауерів недостатньо.

• Стандартних програм сканування недостатньо (ці програми сканують стандартні помилки та не можуть оцінити безпеку, перевіряючи вміст веб-програми).

• Найчастіше нехтування безпекою полягає в коді, оскільки програми реалізуються непрофесійними розробниками або розробниками, які не приділяють особливої уваги безпеці, оскільки основними проблемами часто є швидкість і впровадження необхідні функції.

• Маніпулювання веб-додатком просте (достатньо простого веб-браузера та певної рішучості). Програміст повинен знати, як правильно кодувати, або в контексті безпеки, як кодувати оборонно.

Основна ідея захисного програмування полягає в обробці всіх можливих помилок, включаючи помилки в інших кооперативних процедурах або програмах. Загалом, це визнання того, що програми матимуть проблеми та модифікації, і що розумний програміст розроблятиме код відповідно.

Хоча веб-програми пропонують безліч переваг, вони також стикаються з проблемами, зокрема у сфері безпеки. Такі кіберзагрози, як впровадження SQL, міжсайтовий сценарій (XSS) і витік даних, створюють значні ризики. Розробники та організації повинні впроваджувати надійні заходи безпеки, включаючи методи безпечного кодування, регулярні перевірки безпеки та протоколи шифрування, щоб захистити дані користувачів і підтримувати цілісність веб-додатків.

Ключова проблема, що лежить в основі багатьох вразливостей безпеки полягає в тому, що код веб-додатку виконується з повними привілеями під час обробки запитів від імені користувачів, які лише мають обмежені привілеї, що порушують принцип найменших привілеїв.

Веб-додаток є виконанням операцій з файлами та базами даних від імені користувачів використовуючи власні облікові дані, і якщо зловмисники можуть обдурити систему для виконання неправильної операції, вони може підірвати безпеку програми.

Атаки на веб-додатки можуть мати різноманітні наслідки для організацій, користувачів та інших зацікавлених сторін. Деякі з потенційних наслідків атак веб-додатків включають:

Порушення даних: зловмисники можуть отримати несанкціонований доступ до конфіденційних даних, наприклад особистої інформації, фінансових даних або інтелектуальної власності, що призведе до порушення даних. Це може призвести до серйозних фінансових, репутаційних і юридичних наслідків для постраждалої організації.

Крадіжка особистих даних: зловмисники можуть викрасти особисту інформацію під час атак веб-додатків, що призводить до викрадення особистих даних. Жертви викрадення особистих даних можуть зіткнутися з фінансовими втратами, кредитними проблемами та тривалими процесами відновлення.

Фінансові збитки: атаки на веб-додатки можуть призвести до прямих фінансових збитків для компаній через крадіжку коштів, шахрайство або витрати, пов’язані з виправленням і відновленням.

Шкода репутації: успішна атака веб-програми може завдати шкоди репутації організації, призвести до втрати довіри клієнтів, негативного розголосу та зменшення можливостей для бізнесу.

Юридичні наслідки: організації, які не зможуть захистити свої веб-програми, можуть зіткнутися з юридичними наслідками, такими як штрафи, судові позови чи регулятивні санкції, особливо якщо атака призведе до витоку даних, особисту інформацію.

Порушення роботи бізнесу: атаки веб-додатків можуть порушити роботу бізнесу, спричинивши простой системи, впливаючи на доступність онлайн-сервісів або порушуючи критичну інфраструктуру.

Уразливість порушення автентифікації стосується слабких місць у механізмах автентифікації системи, якими можуть скористатися зловмисники для отримання несанкціонованого доступу.

IoT також стикається з проблемою слабкої або відсутньої автентифікації. Багато пристроїв IoT розроблено з мінімальною безпекою, що робить їх уразливими до атак. Обмежена фізична безпека є серйозною проблемою, з якою стикаються пристрої IoT, оскільки вони часто невеликі та їх легко приховати, що робить їх уразливими для фізичних атак. Фізична атака на пристрій IoT може включати втручання, крадіжку або знищення пристрою. Це може призвести до несанкціонованого доступу до конфіденційної інформації, простою системи та втрати даних.

Пристрої IoT часто підключаються до Інтернету через незахищені мережі, що робить їх уразливими до атак. Наприклад, зловмисник може перехопити зв’язок між пристроєм IoT та Інтернетом, потенційно отримавши доступ до конфіденційних даних. Крім того, незахищені мережі також можна використовувати для атак на інші пристрої в мережі. Неналежний захист даних є серйозною проблемою безпеки, з якою стикаються пристрої IoT, оскільки вони генерують і збирають велику кількість даних, що робить їх вразливими до атак. Ці дані можуть включати особисту інформацію, фінансову інформацію та іншу конфіденційну інформацію. Якщо ці дані не захищені належним чином,

вони можуть потрапити в чужі руки та бути використані для зловмисних цілей. Багато пристроїв IoT важко або неможливо оновити чи виправити, що робить їх уразливими до атак. Це означає, що після виявлення вразливості її неможливо виправити, що робить пристрій вразливим до атак. Крім того, деякі пристрої більше не підтримуються їх виробниками, що унеможливлює отримання будь-яких оновлень або виправлень безпеки. Обмежений регуляторний нагляд за пристроями IoT може бути серйозною проблемою безпеки, оскільки це ускладнює гарантію безпеки цих пристроїв.

Пристрої IoT розроблені для роботи у фоновому режимі, часто без відома або взаємодії з користувачем. Через це може бути важко зрозуміти їхню поведінку та контролювати їхні дії. Наприклад, IoT-пристрій, наприклад інтелектуальна камера, може надсилати дані в хмарну службу без відома користувача. Відсутність видимості поведінки пристрою може ускладнити виявлення та запобігання зловмисній діяльності.

Веб-програми використовуються повсюди для обробки чутливих даних, таких як особиста інформація користувачів, банківські дані та корпоративні секрети. Недотримання стандартів безпеки може призвести до витоків даних, порушення конфіденційності і навіть втрати репутації компанії. Тому забезпечення безпеки веб-застосунків є критично важливим аспектом.

В наш час кібербезпека це надважливо, бо після 24 лютого 2022-го у нашій країні розпочалася повномасштабна війна не тільки на фронті, а й у кіберпросторі.

Кіберворог стабільно атакує різноманітні об’єкти в Україні. Ці атаки часто координуються з наземними та повітряними ударами по насамперед цивільній інфраструктурі. Також хакери використовують інформаційно-психологічні операції.

Зараз кіберворог використовує відомі інструменти: пошук вразливостей, поєднання соціальної інженерії та фішингу, порушення правил цифрової гігієни. Російські хакери активно полюють за персональними та конфіденційними даними українців, бізнесу, держструктур.

Сьогодні атакують весь спектр компаній із вразливими місцями у налаштуваннях віддаленого доступу та системах аутентифікації, з низькою обізнаністю персоналу, відсутністю інструментів виявлення та реагування на інциденти. Атаки на ланцюжки постачання, зокрема програмного забезпечення, залишаються найпривабливішим видом кібернападів.

Україна активно долучена до використання міжнародних стандартів та рекомендацій у сфері кібербезпеки. Організації та установи використовують різні стандарти для забезпечення безпеки систем аутентифікації.

З метою підвищення рівня кібербезпеки в Україні, проводяться навчання та тренінги для фахівців у галузі інформаційної безпеки, включаючи аспекти аутентифікації.

Бізнес має знати слабкі місця свого кіберзахисту та укріплювати їх. Хакери постійно шукають вразливості та атакують через них.

Кібербезпека аутентифікації в Україні є важливою складовою в цілому комплексі заходів з забезпечення кібербезпеки в країні.

Рисунок 1.4 – Галузі, які найбільше страждають через кіберзагрози

2 АНАЛІЗ ОСНОВНИХ ЗАГРОЗ ДЛЯ БЕЗПЕКИ WEB-ДОДАТКІВ

2.1 Першопричини внутрішнього втручання

Слабкі паролі: користувачі часто використовують слабкі паролі, які легко вгадати або зламати за допомогою атак грубої сили.

Наповнення облікових даних: зловмисники використовують облікові дані, отримані в результаті попередніх витоків даних, щоб отримати несанкціонований доступ до облікових записів, де користувачі повторно використовують паролі.

Фішингові атаки: фішингові електронні листи обманом змушують користувачів розкрити свої облікові дані, видаючи себе за законних осіб.

Атаки "людина посередині" (MitM): зловмисники перехоплюють і маніпулюють зв’язком між користувачем і системою автентифікації.

Викрадення сесії: зловмисники викрадають або викрадають автентифіковані маркери сеансу, щоб видати себе за законного користувача.

Атаки грубою силою: зловмисники намагаються отримати доступ, систематично пробуючи всі можливі комбінації паролів.

Недостатнє блокування облікового запису: відсутність належної політики блокування облікового запису може дозволити зловмисникам здійснювати атаки грубої сили без обмежень.

Незахищене зберігання облікових даних: зберігання паролів або маркерів автентифікації в небезпечний спосіб, як-от звичайний текст або слабко хешовані формати.

Біометричний спуфінг: методи біометричної автентифікації можуть бути сприйнятливими до спуфінгу, коли зловмисники використовують підроблені відбитки пальців, обличчя чи інші біометричні дані.

Уразливості програмного забезпечення: зловмисники можуть використати вразливості системи безпеки в програмному забезпеченні автентифікації або бібліотеках.

2.2 Види загроз та методи їх усунення

Число компаній, які застосовують веб-технології для підвищення продуктивності роботи і залучення нових клієнтів, зростає з кожним роком. Безсумнівно, інтернет-сервіси несуть з собою безліч переваг, але є й зворотна сторона медалі – з ростом числа додатків збільшується і кількість кіберзагроз.

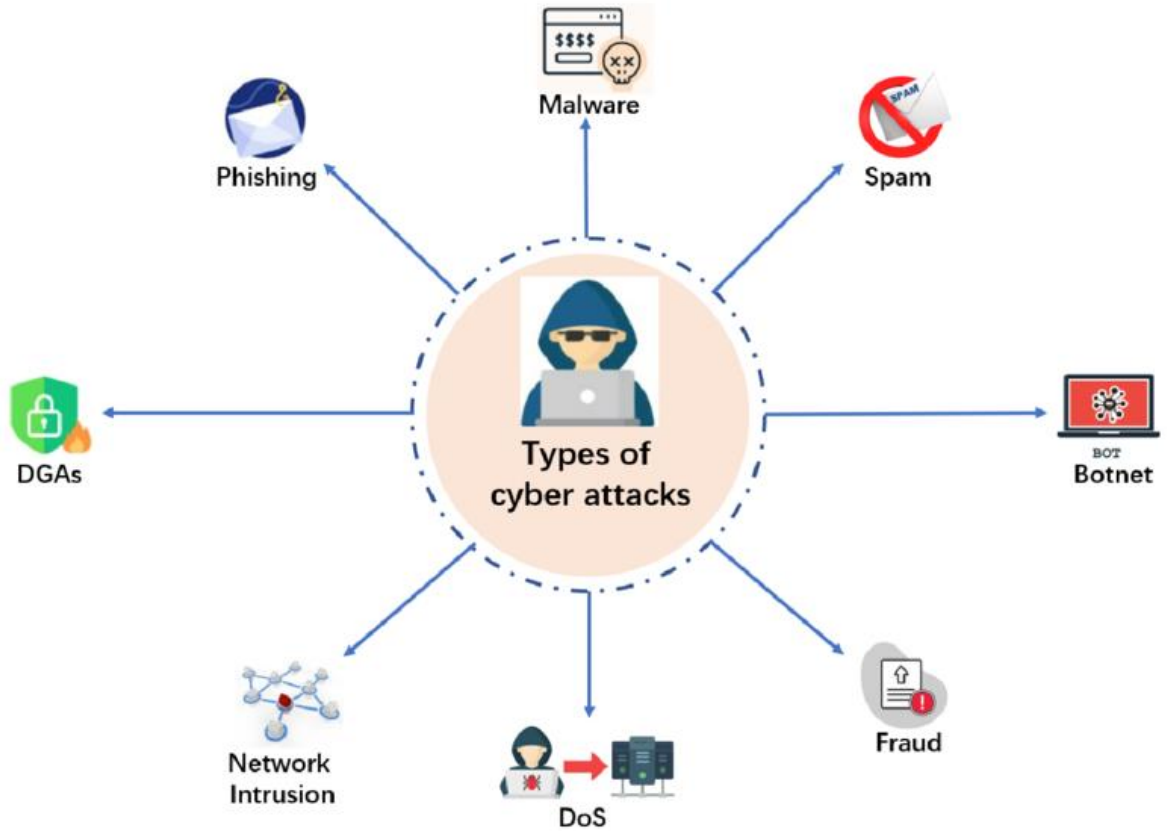

Рисунок 2.1 - Основні види кіберзагроз

Організації повинні мати надійні плани реагування на інциденти, щоб оперативно вирішувати питання безпеки. Це включає моніторинг незвичайних дій, виявлення потенційних порушень і наявність чітко визначеної стратегії реагування.

Інтеграція нових технологій, таких як штучний інтелект (AI) і машинне навчання (ML), стає все більш поширеною для покращення аналітики безпеки та можливостей виявлення загроз.

Поширені типи атак веб-додатків:

1. Міжсайтовий сценарій (XSS)

Міжсайтовий сценарій (XSS) — це тип атаки веб-додатків, який включає введення шкідливих сценаріїв на веб-сторінки, які переглядають інші користувачі. Зазвичай це досягається шляхом введення сценарію в поле введення форми або параметр URL-адреси, який потім зберігається в базі даних веб-додатку.

Коли інший користувач переглядає сторінку, яка містить шкідливий сценарій, сценарій виконується в його браузері, дозволяючи зловмиснику викрасти дані або виконати інші зловмисні дії від імені користувача. Атаки XSS можна запобігти належним чином очищаючи введені користувачем дані, використовуючи заголовки політики безпеки вмісту (CSP) і видаляючи ненадійні дані.

2. Міжсайтова підробка запитів (CSRF)

Підробка міжсайтового запиту (CSRF) — це тип атаки на веб-програму, яка обманом спонукає користувача виконати небажану дію з веб-програмою, у якій він уже автентифікований. Зазвичай це досягається шляхом надсилання спеціально створеного посилання або сценарію користувачеві, який потім виконує небажану дію після натискання.

Наприклад, атака CSRF може бути використана для здійснення несанкціонованих покупок або зміни налаштувань облікового запису. Атаки CSRF можна запобігти за допомогою анти-CSRF маркерів, які є унікальними маркерами, які генеруються веб-програмою для кожного сеансу користувача та мають бути включені в кожен запит до програми.

3. Зовнішня сутність XML (XXE)

Зовнішня сутність XML (XXE) — це тип атаки на веб-програму, яка передбачає використання вразливостей у аналізаторах XML, що

використовуються веб-програмою. Це може дозволити зловмиснику прочитати конфіденційні дані або виконати несанкціоновані дії на сервері веб-додатку.

Атаки XXE зазвичай включають впровадження спеціально створених корисних даних XML, які використовують здатність синтаксичного аналізатора XML читати зовнішні сутності. Атаки XXE можна запобігти, вимкнувши синтаксичний аналіз зовнішніх об’єктів або використовуючи захищені аналізатори XML, які належним чином очищають вхідні дані.

4. Ін'єкційні атаки

Ін’єкційні атаки передбачають вставлення шкідливого коду у веб-програму, як правило, у формі вхідних даних, таких як SQL-запити, команди або сценарії. Ін’єкційні атаки є успішними, коли програма не може належним чином перевірити та очистити вхідні дані. Ці атаки можна запобігти шляхом належної перевірки та дезінфекції вхідних даних і використання параметризованих запитів для доступу до баз даних.

5. Фазз тестування (Fuzzing)

Fuzz-тестування, також відоме як fuzzing, — це техніка, яка використовується для виявлення вразливостей у веб-програмі шляхом надсилання їй випадкових або недійсних вхідних даних . Метою фазз-тестування є визначення того, як веб-програма реагує на різні вхідні дані, а також виявлення помилок і збоїв.

Фаз-тестування можна виконувати вручну або за допомогою автоматизованих інструментів. Fuzz-тестування може виявити вразливості, які не можуть бути виявлені іншими методами перевірки безпеки, такими як тест на проникнення. Щоб виконати ефективне тестування нечіткості, тестувальник повинен розуміти механізми введення та виведення веб-програми та типи даних, які програма обробляє.

6. DDoS (розподілена відмова в обслуговуванні)

Розподілена атака типу «відмова в обслуговуванні» (DDoS) — це тип атаки на веб-програму, яка передбачає перевантаження веб-програми великим обсягом трафіку з кількох джерел, наприклад ботнетів або зламаних пристроїв.

Це може призвести до того, що веб-програма стане недоступною для законних користувачів.

DDoS-атаки можна запобігти за допомогою пристроїв мережевої безпеки, таких як брандмауери та системи запобігання вторгненням, які можуть виявляти та блокувати зловмисний трафік. Крім того, розробники веб-додатків можуть використовувати мережі доставки контенту (CDN) і балансувальники навантаження для розподілу трафіку між декількома серверами, щоб пом’якшити наслідки DDoS-атак.

7. Атака грубою силою

Атака грубою силою — це автоматичний метод вгадування комбінації імені користувача та пароля для отримання несанкціонованого доступу до веб-програми. Зловмисники використовують програмні інструменти, щоб спробувати різні комбінації імен користувачів і паролів, поки не вгадають правильну.

Щоб запобігти атакам грубої сили, веб-додатки можуть запровадити політику обмеження швидкості та блокування облікових записів. Обмеження швидкості обмежує кількість спроб входу з однієї IP-адреси, тоді як блокування облікового запису тимчасово блокує доступ до облікового запису після певної кількості невдалих спроб входу.

8. Обхід шляху

Обхід шляху – це тип атаки веб-програми, яка передбачає маніпулювання шляхами файлів у веб-програмі з метою доступу до неавторизованих файлів або каталогів на сервері. Атаки з обходом шляху зазвичай відбуваються, коли веб-додаток не перевіряє належним чином введені користувачем дані, що дозволяє зловмиснику переходити вверх і вниз по структурам каталогів для доступу до конфіденційних файлів.

Атаки з проходженням шляху можна запобігти належним чином перевіряючи введені користувачем дані та очищаючи шляхи до файлів, а також використовуючи безпечні методи доступу до файлів, які обмежують доступ до конфіденційних файлів і каталогів.

Атаки на основі ідентифікації стають все більш поширеною загрозою в галузі кібербезпеки. Вони використовують особисті дані людей для доступу до даних і мереж, щоб викрасти, знищити або отримати контроль над ними.

За рахунок зростаючої залежності від хмарних ресурсів і віддаленої роботи, зловмисники заволодівають безпрецедентними ідентифікаційними даними. Крім того, зловмисники все частіше використовують викрадені облікові дані та інші вразливості, щоб порушити заходи безпеки ідентифікації.

Щоб захиститися від цих загроз, організації повинні забезпечити, щоб їхні стратегії безпеки включали ідентифікацію як базовий рівень безпеки. Таким чином, групи безпеки повинні прийняти підхід, який обробляє ідентифікаційні дані так само, як вони обробляють кінцеві точки, мережі та хмарні платформи.

Щоб досягти цієї мети, служби безпеки повинні постійно контролювати ідентифікаційні дані своєї організації та використання привілеїв доступу. Ці контекстні дані можна надсилати як сповіщення до систем SIEM, SOAR і XDR як частину інтегрованого процесу реагування на операції безпеки. Наявність такого типу інформації значно спрощує для команд розслідування та вирішення будь-яких інцидентів, які можуть виникнути.

Окрім зобов’язання користувачів регулярно змінювати свої паролі, команди безпеки повинні розглянути можливість впровадження багатофакторної автентифікації (MFA). Вимагання одноразового коду або біометричного маркера для MFA може суттєво перешкодити спробам введення облікових даних і розпилення пароля.

3 СУЧАСНІ МЕТОДИ АВТЕНТИФІКАЦІЇ

Сучасні методи автентифікації виходять за рамки традиційних імен користувачів і паролів, щоб підвищити безпеку та взаємодію з користувачем. Ці методи використовують передові технології та методи перевірки особи користувача.

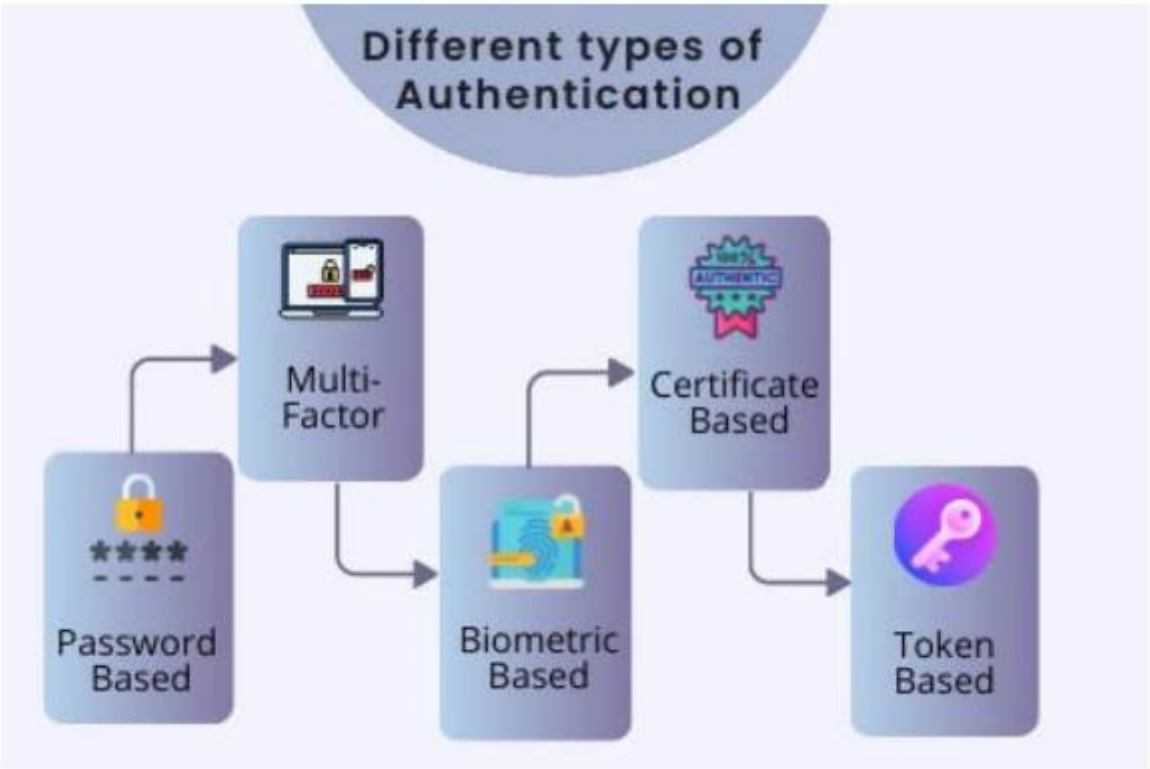

Рисунок 3.1 – Найпопулярніші методи аутентифікації

Ось кілька сучасних методів автентифікації:

3.1 Багатофакторна автентифікація (MFA)

Багатофакторна автентифікація стає все більш критичним заходом безпеки, коли йдеться про захист від захоплення облікового запису.

MFA додає додатковий рівень безпеки, вимагаючи від користувачів надати кілька форм ідентифікації перед наданням доступу. Трьома загальними

факторами часто є те, що користувач знає (пароль), те, що має (токен безпеки чи смартфон), і те, чим є користувач (біометричні дані).

Його створено для покращення безпеки облікового запису та запобігання шахрайському доступу до облікового запису, покращуючи базовий рівень безпеки, досягнутий лише одним фактором автентифікації, зазвичай паролем. MFA все частіше використовується серед компаній.

Корпоративні облікові записи можуть містити дуже конфіденційні дані компанії або особисту інформацію, яку потрібно захистити від зловмисників. З цієї причини багато компаній звертаються до платформ керування ідентифікацією, щоб застосувати багатофакторну автентифікацію в корпоративних програмах.

Традиційно облікові записи захищені лише одним фактором автентифікації, який знає користувач: зазвичай це пароль облікового запису. Щоб підвищити безпеку, MFA означає, що користувач також має надати щось, що він має, як-от одноразовий пароль, надісланий на смартфон, або щось, чим вони є, як-от біометричне сканування.

Коли кінцевий користувач входить до облікового запису, він вводить своє ім’я користувача та пароль, як зазвичай. Потім їх попросять підтвердити свою особу, як правило, з кількома доступними варіантами, як це зробити. Це може включати надсилання одноразового пароля (OTP) через SMS або програму автентифікації або використання програми автентифікації для введення біометричної інформації, наприклад відбитка пальця чи сканування обличчя. Деякі корпоративні організації можуть забажати, щоб користувачі автентифікувалися за допомогою фізичного маркера, наприклад ключа або картки.

Корпоративні рішення для керування ідентифікацією та доступом можуть надавати різні політики адміністратора щодо реалізації багатофакторної автентифікації. Це може включати впровадження двофакторної автентифікації, поширеної форми MFA, у якій фактори автентифікації обмежені лише двома. Багато корпоративних рішень MFA також підтримують адаптивну

автентифікацію, тип автентифікації користувачів, який має на меті полегшити користувачам отримання доступу до критично важливих систем без шкоди для безпеки облікового запису.

За останні кілька років відбулася революція в тому, як працює бізнес. Зараз компанії покладаються на хмарні програми, щоб використовувати їхні потужні функції, бути більш продуктивними та співпрацювати з віртуальними командами. Це стало ще більш важливим під час пандемії Covid-19, оскільки для багатьох команд віддалена співпраця стала абсолютно необхідною для подальшого успіху бізнесу.

Багатофакторна автентифікація захищає від компрометації облікового запису, забезпечуючи додатковий рівень безпеки для кожної окремої спроби входу. Якщо зловмиснику вдасться скомпрометувати пароль облікового запису, але немає додаткового MFA, він зможе змінити пароль облікового запису та фактично заморозити законного користувача з облікового запису. Іноді може знадобитися кілька місяців, перш ніж зламані облікові записи виявляться.

Завдяки MFA користувачі отримують сповіщення про всі підозрілі спроби входу, а зловмисникам фактично блокується доступ, навіть якщо вони мають пароль облікового запису. Дуже малоймовірно, що кіберзлочинець матиме ваш смартфон або відбиток пальця, а також пароль вашого облікового запису, тому MFA значно покращує безпеку облікового запису. Це не означає, що облікові записи на 100% захищені з MFA. Існують обхідні шляхи, які можуть використовувати зловмисники, тому важливо мати багаторівневий підхід до безпеки, як і для всіх рішень безпеки. Однак MFA настійно рекомендується як абсолютний базовий стандарт безпеки облікового запису. Якби кожна організація використовувала багатофакторну автентифікацію, атаки на захоплення облікових записів були б набагато менш поширеними, і ми б також побачили, що успішні атаки на фішинг і компрометацію корпоративної електронної пошти також занепадають.

Адміністратори можуть застосувати багатофакторну або адаптивну автентифікацію для всіх корпоративних облікових записів за допомогою

рішення автентифікації користувачів. Ці рішення дозволяють адміністраторам керувати доступом до облікових записів і гарантувати, що користувачі підтвердять свою особу. Це особливо добре працює з єдиним входом (SSO), який дозволяє користувачам отримувати доступ до всіх своїх облікових записів лише за допомогою одного набору облікових даних, керованих централізовано за допомогою рішення для керування ідентифікацією та доступом. Це усуває потребу в паролях разом.

Переваги: підвищує безпеку, вимагаючи кількох форм перевірки, зменшуючи ризик неавторизованого доступу, навіть якщо один фактор скомпрометований.

Також значною перевагою є гнучка автентифікація. Найкращі рішення автентифікації мають підтримувати адаптивну автентифікацію та єдиний вхід, щоб справжнім користувачам було якомога легше отримати доступ до облікових записів без шкоди для безпеки облікового запису. Вони також повинні підтримувати ряд і різноманітність методів автентифікації, включаючи SMS-коди доступу, одноразові паролі, біометричні засоби контролю та, у деяких випадках, фізичні маркери, якщо це необхідно. Це забезпечить кожному користувачеві доступ до облікових записів, навіть якщо вони, наприклад, не користуються смартфонами.

Найбільшим недоліком MFA є збільшення складності управління як для адміністраторів, так і для кінцевих користувачів. Багатьом менш технічним користувачам може бути важко налаштувати та використовувати MFA. Крім того, існує низка інших поширених проблем:

1. Типи MFA, які вимагають від користувачів певного апаратного забезпечення, можуть призвести до значних витрат і адміністративних витрат.
2. Користувачі можуть втратити доступ до своїх облікових записів, якщо вони втратять або не зможуть скористатися іншими факторами.
3. MFA вносить додаткову складність у додаток.

4. Багато рішень MFA додають зовнішні залежності до систем, які можуть створювати вразливі місця в безпеці або окремі точки збою.
5. Процеси, реалізовані для того, щоб дозволити користувачам обходити або скидати MFA, можуть використовуватися зловмисниками.
6. Вимагання MFA може перешкодити деяким користувачам отримати доступ до програми.

3.2 Біометрична автентифікація

Біометрична автентифікація використовує унікальні біологічні або поведінкові характеристики для перевірки особи. Поширені біометричні методи включають розпізнавання відбитків пальців, розпізнавання обличчя, сканування райдужної оболонки ока, розпізнавання голосу та навіть поведінкові біометричні дані, як-от шаблони друку або аналіз ходи.

Пристрої біометричної автентифікації використовують такі фізичні та поведінкові характеристики як відбитки пальців, образи обличчя, райдужна оболонка, клавіатура або почерк сітківки ока для перевірки на ідентичність.

Біометрична автентифікація стає все більш популярною для багатьох цілей, включаючи вхід у мережу. Оскільки важко завжди носити з собою спеціальні пристрої для підтвердження достовірності особи.

Біометричний шаблон або ідентифікатор (зразок, який, як відомо, належить авторизованому користувачеві), має зберігатися в базі даних, щоб пристрій міг порівняти його з новим зразком, отриманим під час процесу входу. Найпопулярніші види біометричних пристроїв це:

- Сканери відбитків пальців
- Пристрої для розпізнавання образів обличчя
- Пристрої розпізнавання геометрії руки
- Ідентифікаційні пристрої сканування райдужки
- Ідентифікаційні пристрої сканування сітківки
- Динаміка натискання клавіш

Біометрія поведінкових клавіш використовує спосіб і ритм, у якому окремі символи друкуються на клавіатурі. Вимірюється ритм натискання клавіш користувачем для розробки унікального біометричного шаблону текстового набору користувача для подальшої аутентифікації. Інформацію про вібрацію можна використовувати для створення шаблону для майбутнього використання в обох завданнях ідентифікації.

Для веб-додатків найкращим підходом двоетапної автентифікації є динаміка натискання клавіш. Головним чином тому, що для цього методу потрібна лише клавіатура. У кожного є свій унікальний почерк, який дуже важко підробити. Те саме можна сказати про почерк на клавіатурі.

Біометрична автентифікація також може використовувати особливості поведінки, а також фізичні риси, щоб підтвердити, ким є особа, і надати їй належний рівень безпеки доступу.

Біометрична автентифікація працює шляхом порівняння живих біологічних даних із даними біометричного індикатора, які зберігаються у файлі. Якщо вони збігаються, система автентифікує користувача. Якщо вони цього не роблять, автентифікація не вдається. Автентифікація різних біометричних показників працює різними способами, від використання вимірювань до перевірки мереж кровоносних судин.

Важливо зауважити, що біометрична автентифікація – це не те саме, що біометрична перевірка. Процес перевірки біометричної інформації може бути подібним в обох випадках, але їх мета різна. Біометрична автентифікація порівнює біологічні риси або поведінку людини з тими, що вже зберігаються, щоб визначити, чи є ця людина раніше відомою особою, за яку себе видає. З іншого боку, біометрична перевірка – це коли біологічні риси особи використовуються для підтвердження її документів, що посвідчують особу, і, отже, її фактичної особи – зазвичай для цього потрібні як документи, що посвідчують особу, так і її біологічні особливості. Насправді вам може знадобитися спочатку пройти біометричну перевірку у ваших стосунках з користувачем, клієнтом, співробітником тощо, щоб пізніше мати змогу

автентифікувати їх за допомогою біометричних маркерів. Процес біометричної перевірки відбудеться під час їх першого візиту або реєстрації. Після цього, під час біометричної автентифікації, представлені функції будуть порівнюватися з тими, що є в файлі, щоб перевірити, чи вони збігаються.

Ви використовуєте відбиток пальця, щоб розблокувати вхідні двері, телефон, ноутбук, планшет чи інший пристрій? Цей вид біометричної автентифікації використовує унікальні відбитки пальців людини для ідентифікації та автентифікації за допомогою об'єктива та датчика. Перегляд відбитка пальця у високій роздільній здатності дозволяє пристрою визначити, чи збігається він з одним із відбитків пальців у своїй базі даних. Якщо це так, система автентифікує користувача.

Розпізнавання обличчя порівнює характерні деталі обличчя людини з тими, що зберігаються в базі даних системи. Такий сканер вимірює такі деталі, як форма підборіддя людини, відстань між очима, ширина носа тощо.

Сканування сітківки ока полягає в дослідженні тонкої капілярної мережі сітківки – частини ока, яка доставляє необхідний кисень і поживні речовини до сітківки. Щоб це спрацювало, око має бути близько до сканера. Сканер направляє в око інфрачервоне світло низької енергії, за допомогою якого він може побачити капілярну мережу та порівняти її з тією, що зберігається в базі даних системи. Сітківка, як і відбитки пальців, унікальна – навіть сітківка однояйцевих близнюків відрізняється одна від одної.

Розпізнавання голосу працює шляхом порівняння зразка голосу користувача (відомого як голосовий відбиток) із зразком, який зберігається у файлі. Система розбиває голос на кілька частот, щоб порівняти їх. Як і відбитки пальців, відбитки голосу унікальні та залишаються незмінними протягом усього життя людини. Ринок біометричного розпізнавання голосу також швидко зростає. У 2020 році його вартість становила 1,1 мільярда доларів. Очікується, що до 2026 року ця вартість зросте більш ніж утричі до 3,9 мільярда доларів.

Іншою формою біометричної автентифікації є розпізнавання шаблонів набору тексту. Це працює шляхом аналізу динаміки натискання клавіш

користувачем. Він вимірює ряд факторів, таких як час преси, час пошуку, час польоту та інші фактори, щоб ідентифікувати особу.

Недоліки:

1. Реалізація може бути дорогою
2. Часові наслідки для компаній, які потребують підвищення кваліфікації своєї робочої сили, щоб впровадити дане рішення
3. Технологічна складність
4. Занепокоєння щодо конфіденційності для осіб, які не хочуть, щоб їхні біометричні ідентифікатори були в базі даних

Враховуючи унікальні особливості, які кожен має на своїй особистості – відбитки пальців, райдужну оболонку ока тощо – можна вибачити припущення, що біометричну автентифікацію неможливо підробити чи зламати шахраї. Але реальність інша. Насправді той факт, що біометричну автентифікацію багато хто помилково вважає неможливим для зламу, значно погіршує проблему. Оскільки окремі особи та навіть компанії помилково вважають, що біометрична автентифікація є непроникним захистом, їх обережність, швидше за все, буде знижена. Потім технічно підковані шахраї можуть використовувати підробки, витоки інформації та методи соціальної інженерії, щоб обдурити біометричну автентифікацію та отримати доступ до фізичної та цифрової інфраструктури.

Переваги: Забезпечує високий рівень безпеки, оскільки біометричні дані є унікальними для кожної людини.

3.3 Поведінкова біометрія

Поведінкова біометрія аналізує моделі поведінки користувача, наприклад динаміку натискання клавіш, рух миші або жести сенсорного екрана. Система встановлює базову лінію нормальної поведінки та позначає аномалії для подальшої перевірки.

Подібно до того, як під час першого налаштування облікового запису, захищеного паролем, вам потрібно зареєструвати пароль, під час налаштування

біометричних систем вам потрібно зареєструватися в системі, надавши свої біометричні дані. Це робиться для того, щоб система могла створити модель, яку вона зможе використовувати для відповідності вашим біометричним даним у майбутньому. Під час використання фізіологічної біометрії вам потрібно зареєструвати свої біометричні дані в системі, перш ніж почати використовувати їх для входу. Перше сканування, яке ви реєструєте, створює свого роду особистий біометричний шаблон, який потім стає посиланням, з яким порівнюватиметься кожна наступна спроба входу. Але поведінкова біометрична автентифікація працює дещо інакше. Натомість поведінкова біометрична автентифікація безперервно аналізує ваші дані у фоновому режимі протягом тривалого часу, а не використовує лише одну контрольну точку. Це означає, що система не тільки має багато зразків даних, з якими можна порівняти ваші спроби входу, але й може адаптуватися з часом, коли ваша поведінка змінюється. І оскільки технологія працює безшумно, вона може аналізувати мікро шаблони, які ви демонструєте у своїй поведінці, і створювати профіль ваших унікальних рухів, не заважаючи вам працювати.

Технологія може створити ваш профіль на основі таких речей, як швидкість набору тексту, тривалість часу, протягом якого ви натискаєте клавішу, і моделі натискання клавіш під час введення певних слів або послідовностей. Потім ваші дані аналізуються за допомогою технологій і алгоритмів AI і ML, які, коли ви намагаєтесь увійти, пасивно призначають вам оцінку ризику залежно від того, наскільки ваша поведінка відповідає минулій поведінці. Адміністратори можуть визначати та запроваджувати порогові значення для того, наскільки вище або нижче певного числа може бути ваш показник ризику. І якщо буде виявлено аномалії та ваш показник ризику перевищить встановлений поріг, вам буде відмовлено в доступі та/або запропоновано пройти автентифікацію за допомогою додаткового фактора. Крім того, багато рішень поєднують цю техніку підрахунку ризику з контекстними факторами для посилення безпеки вашого облікового запису. Такі фактори можуть включати місцезнаходження,

мережу WiFi, відомі/невідомі пристрої та час доби, коли ви зазвичай отримуєте доступ до певних облікових записів і виконуєте певні дії.

Ключова річ, яку слід враховувати при дослідженні точності будь-якої поведінкової біометричної системи, — це те, про що ми коротко торкалися раніше, — порогові значення. Паролі не мають порогу — ви або отримуєте 100% збіг, або ні.

Поведінкова біометрія, з іншого боку, працює як свого роду механізм ризику, із заданими пороговими значеннями, які визначають, наскільки далеко ви можете перевищити певний показник ризику, перш ніж вам буде відмовлено в доступі. І ці порогові значення, які визначають ваші адміністратори, мають величезний вплив на точність вашої системи. Чим суворіший поріг, тим точнішою буде ваша система. Але це, ймовірно, призведе до вищого рівня помилкових відмов (FRR — коли користувачеві відмовляють у доступі до свого облікового запису, навіть якщо він є його законним власником), що також ускладнить роботу користувача. І навпаки, що нижчий ваш поріг, то менш точною буде ваша система, і ви ризикуєте отримати вищий рівень помилкового прийняття (FAR — коли користувач помилково отримує доступ до облікового запису, який йому не належить). Отже, якщо ви працюєте в організації з високим рівнем безпеки, ви можете запровадити суворіші порогові значення. Але якщо користувальницький досвід для вас є пріоритетом, вам слід шукати баланс між високою безпекою та точністю та користувальницьким досвідом. З часом точність зростає Оскільки поведінкова біометрія постійно відстежує вашу поведінку, це означає, що пул даних, який створює базову лінію поведінки, постійно розширюється. Отже, ключова річ, яку слід зауважити щодо цього типу автентифікації, полягає в тому, що з часом вона стає дедалі точнішою, оскільки стає більш знайомою з вашими моделями поведінки. Цей постійний моніторинг у поєднанні з технологією машинного навчання також означає, що будь-які зміни у вашій поведінці будуть враховані, щоб система могла адаптуватися та зберігати високий рівень точності.

Переваги: підвищує безпеку шляхом постійного моніторингу поведінки користувачів. Прозорий для користувачів і мінімізує залежність від традиційних облікових даних.

3.4 Апаратні токени та смарт-карти

Апаратні маркери та смарт-карти генерують одноразові паролі або зберігають криптографічні ключі. Користувачі повинні мати фізичний токен або картку на додаток до знання пароля для автентифікації.

Інфраструктура відкритих ключів (PKI), поєднання політик, апаратного та програмного забезпечення, дуже зручна, коли справа доходить до автентифікації. Він забезпечує найвищий рівень безпеки та може бути легко вбудований у фізичні чіпи в жетонах або смарт-картках.

Апаратні елементи в інфраструктурі відкритих ключів (PKI) є важливими. Існує широкий спектр фізичних токенів, карток і пристроїв, які підтримують PKI.

Є багато обладнання, яке може йти рука об руку з PKI, наприклад токени, картки та навіть смартфон у вашій кишені.

Телефон стане дуже важливим предметом у найближчі роки для кращої кібербезпеки.

Проблема навколо паролів реальна. До них легко отримати доступ, їх легко виманювати та фішингувати. Вони мають низький рівень безпеки, але це все одно для багатьох людей зручно. Це означає, що поки без пароля не стане справді зручніше, пароль залишатиметься серед нас.

Переваги: Забезпечує додатковий рівень безпеки, вимагаючи фізичного пристрою. Підходить для середовищ з підвищеними вимогами до безпеки.

3.5 Автентифікація Push-сповіщень

Користувачі отримують push-сповіщення на свій зареєстрований мобільний пристрій із пропозицією схвалити або відхилити запит на

автентифікацію. Зазвичай використовується в поєднанні з мобільними програмами.

Push-сповіщення є частиною веб-технології push-сповіщень. Технологія push-сповіщень — це онлайн-спілкування тип протоколу. Push-сповіщення та технології використовуються в онлайн-маркетингу та онлайн-комунікації, де запити push-повідомлень не надсилаються сервером. Типовим прикладом веб-технології push-повідомлень, яка використовується на смартфонах, є обмін миттєвими повідомленнями. Технологія Push відрізняється від більш поширеного еквівалента, технології Pull або Client Pull , який переважно використовується на традиційних веб-сторінках. Наприклад, веб-сайт надсилає інформацію клієнту, лише коли клієнт запитує її від сервера, на якому розміщено веб-сайт. Замість того, щоб клієнт отримував дані чи повідомлення із сервера, технологія push дозволяє серверам надсилати дані безпосередньо клієнтам без початкового запиту. Push-сповіщення застосовують технологію push для надсилання повідомлень безпосередньо з центрального сервера на пристрій користувача. Зазвичай push-сповіщення надсилаються на мобільні пристрої, де вони з’являються на верхньому банері пристрою, у центрі сповіщень або на екрані блокування. Однак деякі типи push-сповіщень можна надсилати через веб-браузери, що робить їх сумісними з мобільними пристроями та настільними пристроями.

Хоча автентифікація push-повідомлень пропонує багато переваг, вона також пов’язана з деякими проблемами та обмеженнями, які необхідно враховувати. Однією з головних проблем є залежність від смартфона користувача та підключення до мережі. Якщо користувач втратить, забуде або пошкодить свій смартфон, або якщо у нього немає Інтернету чи стільникового зв’язку, він не зможе отримувати або відповідати на push-сповіщення, і, таким чином, буде заблоковано доступ до системи чи служби. Іншою проблемою є можливість фішингу або спуфінгу, коли хакери можуть спробувати обманом змусити користувачів натиснути на підроблені або зловмисні сповіщення, які скомпрометують їхні облікові дані або дані. Користувачі повинні бути обізнані

та бути пильними щодо таких загроз і перевіряти джерело та вміст сповіщень, перш ніж відповідати. Третім завданням є сумісність і взаємодія автентифікації push-повідомлень з різними системами та службами. Не всі системи та служби можуть підтримувати або приймати автентифікацію push-сповіщень як дійсний метод, а деякі можуть вимагати додаткового налаштування або інтеграції для роботи з ним. Користувачі повинні перевірити та дотримуватися конкретних вимог та інструкцій кожної системи чи служби, до яких вони хочуть отримати доступ за допомогою автентифікації push-повідомлень.

Переваги: Пропонує бездоганний і зручний досвід. Користувачі можуть швидко схвалити або відхилити запити автентифікації зі своїх мобільних пристроїв.

3.6 Автентифікація на основі ризиків

Автентифікація на основі ризиків оцінює ризик, пов’язаний із конкретною спробою входу, на основі різних факторів, зокрема поведінки користувача, місцезнаходження та пристрою. Дії з високим ризиком можуть ініціювати додаткові кроки автентифікації.

Автентифікація на основі ризиків враховує багато унікальних факторів, зокрема:

- Час дня
- Місцезнаходження
- Інформація про пристрій і браузер
- IP-адреса
- Інформація про користувача
- Контекст запиту

Ці фактори впливають на рівень ризику операції. Залежно від передбачуваного ризику користувачам може бути запропоновано другий фактор автентифікації. Наприклад, якщо ви спробуєте ввійти на свій банківський рахунок з іншої країни, вам, швидше за все, доведеться підтвердити свою особу.

З іншого боку, користувачі можуть отримати безперебійний досвід, якщо розрахований ризик низький. Наприклад, користувачеві може не знадобитися повторно вводити свої облікові дані після закінчення сеансу, якщо він користується тим самим корпоративним пристроєм і мережею протягом звичайного робочого часу.

Деякі рішення автентифікації на основі ризиків також можуть створювати та оновлювати динамічні профілі кожного користувача. З часом програмне забезпечення вивчає моделі поведінки користувача, що дозволяє набагато точніше обчислювати ризики.

Часто користувачі навіть не знають, що відбувається автентифікація на основі ризику. Оскільки більшість транзакцій не підпадають під високий ризик, поетапна автентифікація потрібна лише зрідка. Це найбільша перевага автентифікації на основі ризиків: вона забезпечує безперебійну взаємодію з користувачем, додаючи значний рівень безпеки інфраструктурі компанії.
Автентифікація на основі ризиків контролюється наборами правил, також відомих як політики, які класифікують ступінь ризику конкретної транзакції. Щоразу, коли надходить запит, ці політики аналізують різні сигнали ризику та порівнюють їх один з одним, щоб обчислити оцінку ризику. У свою чергу, показник ризику визначає досвід автентифікації для кінцевого користувача. Вони можуть мати безперебійний досвід або їм може знадобитися виконати додаткові кроки перевірки.

Політику ризиків можна налаштувати. Вона залежить від організації та типу даних, до яких здійснюється доступ. Наприклад, якщо компанія має лише працівників у Канаді, архітектор безпеки може створити правило, яке класифікує будь-яку транзакцію з будь-якої іншої країни як високоризикову. Подібним чином спроба отримати доступ до загальної інформації про компанію вважатиметься менш ризикованою, ніж доступ до конфіденційних даних, таких як номер соціального страхування працівника.

Якщо транзакція класифікується як високоризикова, посилену автентифікацію також можна налаштувати відповідно до політики компанії.

Користувачеві може знадобитися виконати різні методи автентифікації за другим фактором залежно від оцінки ризику або чутливості програми. Їм навіть може бути заборонено доступ, якщо занадто багато факторів ризику є підозрілими.

Найскладніші системи автентифікації на основі ризиків використовують машинне навчання для встановлення базових показників типової поведінки груп користувачів, а потім виявляють аномалії поведінки, коли вони виникають у режимі реального часу, класифікуючи їх за різними рівнями ризику. Адміністратор або група безпеки може призначити певні дії для кожної категорії в політиках ризиків.

Переваги: адаптує заходи безпеки на основі контекстної інформації, забезпечуючи баланс між безпекою та досвідом користувача.

3.7 Управління ідентифікацією на основі блокчейну

Технологія блокчейн використовується для захисту ідентифікаційних даних користувачів і керування ними. Платформи децентралізованої ідентифікації дають користувачам можливість контролювати свою особисту інформацію, підвищуючи конфіденційність і безпеку.

Управління ідентифікацією на основі блокчейну пропонує безпечну та ефективну альтернативу, надаючи людям більше можливостей контролювати свою особисту інформацію. Це дозволяє їм створювати самостійні суверенні ідентифікатори, які можна переносити на різні платформи, усуваючи потребу в повторюваних процесах перевірки.

Відповідно до дослідницького звіту, опублікованого Market Research Future (MRFR), компанією з аналізу ринку, зазначено, що очікується, що ринок управління ідентифікацією на основі блокчейну досягне 17,81 мільярда доларів США до 2030 року при середньорічному темпі зростання (CAGR) 56,60%. Управління ідентифікацією на основі блокчейну має великий потенціал для таких галузей, як фінанси, охорона здоров’я, ланцюг постачання та державні

послуги. Інтегруючи технологію блокчейну в ці сектори, підприємства можуть підвищити цілісність даних, оптимізувати процеси, зменшити шахрайство та підвищити довіру між зацікавленими сторонами.

Однак важливо вирішити проблеми та обмеження, пов'язані з керуванням ідентифікацією на основі блокчейну. Для широкого впровадження необхідно подолати проблеми, пов'язані з масштабованістю, сумісністю та нормативними рамками.

Є кілька способів, за допомогою яких керування ідентифікація на основі блокчейну може покращити взаємодію з користувачем і захистити його:

• Самосуверенна ідентичність: Блокчейн дозволяє людям повністю контролювати свою цифрову ідентичність. Користувачі можуть створювати та керувати власними профілями ідентифікації, включаючи особисту інформацію, облікові дані та дозволи, не покладаючись на посередників або сторонні органи. Ця модель самосуверенної ідентифікації розширює можливості людей і надає їм більше конфіденційності та контролю над своїми даними.

• Покращена безпека: Мережі блокчейн використовують криптографічні алгоритми для захисту даних і транзакцій. Ідентифікаційні дані користувача та облікові дані можуть зберігатися в блокчейні в зашифрованому форматі, що ускладнює неавторизованим особам підробку або доступ до конфіденційної інформації. Крім того, розподілена природа блокчейна усуває єдину точку відмови, що робить його більш стійким до кібератак.

• Незмінний контрольний слід: Blockchain забезпечує незмінний і прозорий запис транзакцій, пов'язаних із ідентифікацією. Будь-які зміни або оновлення профілю ідентифікації реєструються в блокчейні, створюючи аудиторський слід, який може бути перевірений кількома сторонами. Це забезпечує цілісність даних і допомагає виявляти та запобігати шахрайським діям, таким як викрадення особистих даних або видавання себе за іншу особу.

• Інтероперабельність і портативність: Системи ідентифікації на основі блокчейну можуть забезпечити бездоганну взаємодію між різними платформами та програмами. Користувачі можуть підтримувати єдину цифрову

ідентифікацію в різних службах, усуваючи потребу в кількох облікових даних для входу та зменшуючи тертя у взаємодії з користувачем. Ця портативність дозволяє користувачам безпечно передавати свої ідентифікаційні дані між різними організаціями чи службами, підвищуючи зручність і гнучкість.

• Обмін даними на основі згоди: Завдяки управлінню ідентифікацією на основі блокчейну користувачі мають більше контролю над своїми особистими даними та можуть явно надавати або скасовувати дозволи на обмін даними. Цей підхід на основі згоди гарантує, що люди мають чітке розуміння того, як використовуються їхні дані, і дає їм змогу вибирати, які організації чи установи мають доступ до їх інформації. Це посилює конфіденційність і зменшує ризик неправомірного використання даних.

• Довіра та надійність: Прозорий і незмінний характер блокчейна допомагає встановити довіру між різними сторонами, залученими до процесів перевірки особи. Організації можуть перевірити автентичність ідентифікаційних даних особи безпосередньо в блокчейні, усуваючи потребу в посередниках і зменшуючи можливість шахрайства з ідентифікацією. Ця підвищена довіра може призвести до покращення взаємодії з користувачем і оптимізації процесів перевірки особи.

Керування ідентифікацією на основі блокчейну має потенціал для значного покращення взаємодії з користувачем і безпеки в різних сферах. Традиційні системи керування ідентифікацією часто покладаються на централізовані бази даних, які можуть бути вразливими до злому, витоку даних і несанкціонованого доступу. З іншого боку, технологія блокчейн пропонує децентралізовану та безпечну платформу для керування цифровими ідентифікаторами. У той час як керування ідентифікацією на основі блокчейна пропонує значні переваги, важливо вирішити такі проблеми, як масштабованість, відповідність нормативним вимогам і баланс між конфіденційністю та прозорістю. Однак завдяки постійному прогресу та дослідженням у цій галузі технологія блокчейн має потенціал для революції в

управлінні ідентифікацією та надає користувачам покращену безпеку та контроль над їхніми цифровими ідентифікаторами

Переваги: Зменшує ризик крадіжки особистих даних і несанкціонованого доступу, надаючи користувачам більший контроль над своєю ідентифікаційною інформацією.

3.8 Автентифікація за сертифікатами

Сертифікат являє собою набір атрибутів, що ідентифікують власника. Також сертифікат криптографічно пов'язаний з закритим ключем, який зберігається у власника сертифіката і дозволяє однозначно підтвердити факт володіння сертифікатом. На стороні клієнта сертифікат разом з закритим ключем можуть зберігатися в операційній системі, в браузері, в файлі, на окремому фізичному пристрої (smart card, USB token). Зазвичай закритий ключ додатково захищений паролем або PIN-кодом. У веб-додатках традиційно використовують сертифікати стандарту X.509. Автентифікація за допомогою X.509-сертифіката відбувається в момент з'єднання з сервером і є частиною протоколу SSL / TLS.

Цей механізм також добре підтримується браузерами, які дозволяють користувачеві вибрати і застосувати сертифікат, якщо веб-сайт допускає такий спосіб автентифікації.

Під час автентифікації сервер виконує перевірку сертифіката на підставі наступних правил:

1. Сертифікат повинен бути підписаний.
2. Сертифікат повинен бути дійсним на поточну дату (перевірка терміну дії).
3. Сертифікат не повинен бути відкликаний відповідним CA (перевірка списків виключення).

Використання сертифікатів для автентифікації - куди більш надійний спосіб, ніж автентифікація за допомогою паролів. Це досягається створенням в процесі автентифікації цифрового підпису, наявність якої доводить факт застосування закритого ключа в конкретній ситуації.

Автентифікація на основі сертифіката (CBA) використовує цифровий сертифікат, отриманий за допомогою криптографії, для ідентифікації користувача, пристрою чи машини перед наданням доступу до програми, мережі чи іншого ресурсу. На відміну від деяких рішень автентифікації, націлених на людей, таких як одноразові паролі (OTP) і біометрія, автентифікація на основі сертифіката може бути прийнята для всіх кінцевих точок, включаючи сервери, персональні комп'ютери, електронні паспорти та буквально все, що може бути класифіковано як Інтернет речей (IoT).

CBA є набагато безпечнішою альтернативою, ніж традиційна комбінація імені користувача та пароля, хоча її також можна використовувати разом із традиційними методами для надійної автентифікації користувача, щоб створити форму MFA, стійку до фішингу. Оскільки цифровий сертифікат зберігається на пристрої або комп'ютері окремої особи разом із закритим ключем, він дозволяє браузеру або клієнту користувача автоматично входити в різні системи без особливих додаткових зусиль з боку користувача, оскільки його можна просто надати за запитом.

Загалом автентифікація на основі сертифіката клієнта та інші методи, де секрет ніколи не відкривається навіть користувачеві, є кращими, ніж автентифікація на основі пароля. Автентифікація за іменем користувача та паролем базується лише на тому, що користувач знає (пароль), але автентифікація клієнта на основі сертифіката також використовує те, що є у користувача (приватний ключ), який неможливо підробити, вгадати чи спроектувати соціально.

Але також важливо виділити деякі умови, які допомагають підтримувати цей рівень контролю:

- жодні неавторизовані користувачі не отримали доступу до закритого ключа, що лежить в основі цифрового сертифіката
- належним чином керується життєвий цикл розповсюджених сертифікатів, включаючи реєстрацію, оновлення та відкликання

- створено відповідну інфраструктуру для підтримки надсилання та перевірки сертифікатів

Користувачі повинні бути особливо пильними, захищаючи приватні ключі на додаток до фізичної безпеки своїх пристроїв загалом, але сторони, відповідальні за інфраструктуру підтримки СВА, також відіграють свою роль у цьому процесі.

3.9 Модель безпеки з нульовою довірою

Модель нульової довіри передбачає, що жодному користувачу чи системі не можна довіряти за умовчанням, навіть якщо вони знаходяться в корпоративній мережі. Підтвердження вимагається від усіх, незалежно від їх місцезнаходження чи пристрою.

Модель Zero Trust покладається на надійну автентифікацію та авторизацію для кожного пристрою та особи перед будь-яким доступом або передачею даних у приватній мережі, незалежно від того, чи знаходяться вони всередині чи за межами периметра мережі. Процес також поєднує аналітику, фільтрацію для перевірки поведінки та постійного спостереження за сигналами компрометації. Якщо користувач або пристрій демонструють ознаки того, що діють інакше, ніж раніше, це береться до відома та контролюється як можлива загроза.

Ця основна зміна підходу усуває багато поширених загроз безпеці. Зловмисники більше не можуть витрачати час, користуючись слабкими місцями периметра, а потім використовуючи конфіденційні дані та програми, оскільки вони проникли всередину рову. Тепер рову немає. Існують лише програми та користувачі, кожен із яких має взаємно автентифікуватися та перевірити авторизацію, перш ніж отримати доступ. Взаємна автентифікація відбувається, коли дві сторони автентифікують одна одну одночасно, наприклад користувач із логіном і паролем і програма, до якої вони підключаються через цифровий сертифікат.

Оскільки нульова довіра є методологією безпеки, вона вимагає від організацій оцінювати свої стратегії безпеки та параметри для своєї системи та докладати послідовних зусиль для оцінки та вдосконалення існуючих стратегій. Деякі з основних проблем:

1. Підтримка системи Zero trust з боку керівництва та користувачів
2. Швидке збільшення кількості пристроїв збільшує ймовірність незахищеної кінцевої точки.
3. Експоненціальне зростання додатків збільшує потребу у відстеженні та моніторингу.

Переваги: підвищує безпеку, вимагаючи безперервної перевірки, зменшуючи ризик бокового руху та внутрішніх загроз.

Реалізація комбінації цих сучасних методів автентифікації може значно посилити безпеку, запропонувавши більш зручний і ефективний досвід для користувачів, які отримують доступ до систем і програм. Вибір методів автентифікації має базуватися на конкретних вимогах безпеки та характеристиках користувача середовища, в якому вони розгортаються.

Але не зважаючи на велику кількість методів найбільшого поширення набули 4 методи, що наразі залишаються актуальними при введенні нових продуктів, переважно через простоту використання. Їх можна виділити у загальній таблиці та провести порівняльний аналіз.

Метод Аутентифікації	Плюси	Мінуси
Пароль	+ Простий у використанні	- Його можна вгадати або вкрасти
	+ Зручний для користувача	- Піддається атакам типу "брутфорс"
	+ Легко змінюється	- Ризик "повторного використання паролю"
Двофакторна аутентифікація	+ Забезпечує додатковий рівень безпеки	- Можливість втрати/ крадіжки другого фактора
	+ Менша ймовірність несанкціонованого доступу	- Додаткові зусилля користувача при вході
	+ Зменшує ризик атак через витіснення	
Біометрична аутентифікація	+ Унікальність індивідуальних характеристик	- Зберігання та обробка біометричних даних
	+ Зручність використання для користувача	- Можливість підробки або використання "в обхід"
	+ Можливість використання різних біометрій	
OTP (Одноразовий пароль)	+ Тимчасовий і діє тільки певний час	- Можливість атаки "людина посередник"
	+ Зручний для використання через мобільний додаток або токен	- Ризик втрати/крадіжки генератора OTP
	+ Висока ступінь безпеки при відсутності повторного використання	- Залежність від електронних пристроїв

Рисунок 3.2 – Порівняльна характеристика методів автентифікації

4 РОЗРОБКА ВЕБ СТОРІНКИ З ДВУХФАКТОРНОЮ АВТЕНТИФІКАЦІЄЮ

Двофакторна аутентифікація (2FA) є ефективним інструментом для захисту облікових записів та інформації в онлайн-середовищі, оскільки вона вимагає від користувача подання двох незалежних факторів для підтвердження своєї ідентичності.

Нижче представлено порівняльний огляд плюсів двофакторної аутентифікації порівняно з ім'ям користувача та паролем:

- зараз паролі стають все більше вразливими до атак. Крім того, люди часто використовують слабкі паролі або використовують один і той же пароль для декількох служб;
- двофакторна аутентифікація (2FA) вимагає не тільки знання пароля, але і щось фізичне чи щось, чим користувач володіє (мобільний телефон або токен). Це робить процес аутентифікації значно безпечнішим;
- паролі можуть бути перехоплені або витягнуті із баз даних, особливо якщо вони не шифруються або використовуються повторно;
- додатковий фактор, такий як код або підтвердження на мобільному пристрої, важко отримати незаконно;
- щоб забезпечити високий рівень безпеки, користувачам часто доводиться використовувати складні паролі, які вони можуть важко запам'ятати;
- деякі методи 2FA дозволяють користувачам швидко відновити доступ до свого облікового запису, навіть якщо вони забули пароль.
- На відміну 2FA у звичайної форми входу ризик несанкціонованого доступу високий, особливо якщо пароль потрапив у руки атакуючого. у інших методах при несанкціонованому доступі до пароля все ще важко отримати доступ, оскільки не вистачить додаткового фактора;
- звичайна форма «пароль та логін» зазвичай проста у використанні, але може вимагати регулярної зміни та складних умов для підвищення безпеки;

- хоча двофакторна аутентифікація (2FA) і вимагає додаткових кроків, але сучасні технології, такі як мобільні додатки для 2FA, роблять процес відносно зручним.

Для данної дипломної роботи було розроблено шалон сторінки входу до веб додатку з двухфакторною автентифікацією через Google Authentication на локальному веб сервері.

Для роботи цього міні-проекту необхідне створення та закріплення бази даних за нашою формою аутентифікації користувача.

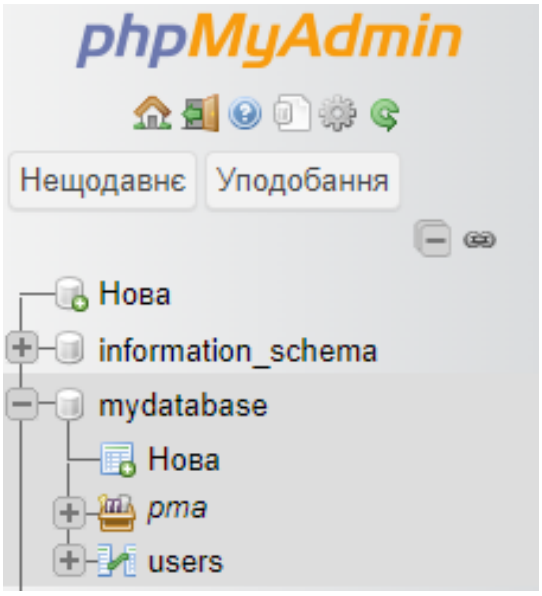

Рисунок 4.1 Скрін створеної бази данних mydatabase

Для даного коду було підвантажено необідні программи (XAMMP, Ngrok, Composer) та бібліотеки (google/apiclient, pragmarx/google2fa, spomky-labs/otphp)

В першу чергу було створено прийнятний зовнішній вигляд для данного меню, вибудувано архітектуру веб сторінки на мовах CSS, HTML та PHP.

```css
body {
    font-family: Arial, sans-serif;
    background-image: url('diplomback.jpg');
    background-size: cover;
    display: flex;
    justify-content: center;
    align-items: center;
    height: 100vh;
}

.container {
    background-color: #fff;
    box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
    padding: 20px;
    border-radius: 8px;
    width: 300px;
    text-align: center;
    margin: 50px auto;
    border: 1.1px solid #333;
}

.container h2 {
    color: #333;
}

.form-group {
    margin-bottom: 15px;
}

.label-container {
    margin-bottom: 10px;
}

label {
    font-weight: bold;
    margin-bottom: 5px;
    color: #555;
}

input {
    padding: 8px;
    border: 1px solid #ccc;
    border-radius: 4px;
    width: 70%;
}

button {
    background-color: #3498db;
    color: #fff;
    padding: 10px;
    border: none;
    border-radius: 4px;
    cursor: pointer;
    font-size: 16px;
    width: 80%;
}

button:hover {
    background-color: #2a80b9;
}

p {
    margin-top: 15px;
    margin-bottom: 10px;
    color: #555;
}

a {
    color: #3498db;
    text-decoration: none;
}

a:hover {
    text-decoration: underline;
}

a[href="google_login.php"] {
    display: flex;
    align-items: center;
    justify-content: center;
    margin-top: 15px;
    color: #3498db;
    text-decoration: none;
}

a[href="google_login.php"] .google-logo {
    width: 35px;
    height: 35px;
    margin-right: 5px;
}

a[href="google_login.php"]:hover {
    text-decoration: underline;
}

.back-button {
    display: inline-block;
    padding: 10px;
    background-color: #3498db;
    color: #fff;
    text-decoration: none;
    border-radius: 5px;
}
```

Рисунок 4.2 – CSS код, архітектура сторінки

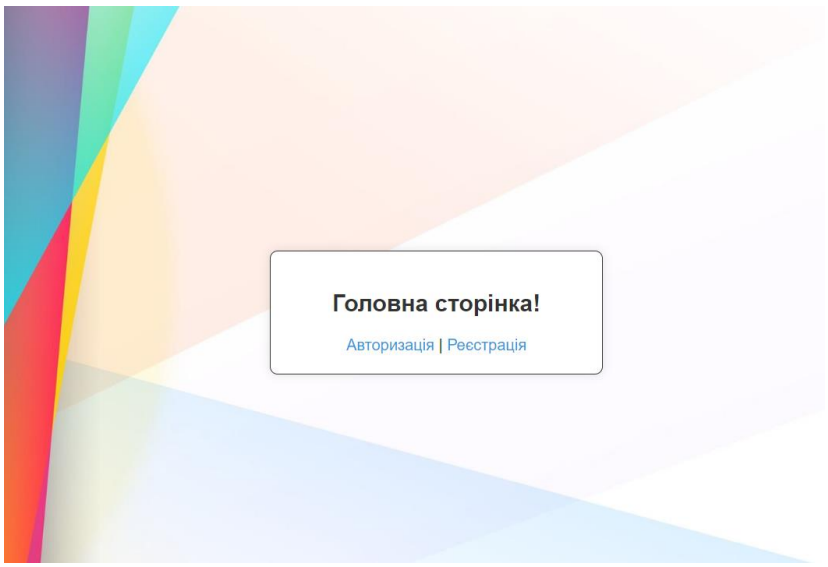

Рисунок 4.3 – Зовнішній вигляд початкової сторінки входу

Надалі було розроблено функціонал для базової реєстрації за допомогою паролю та імені користувача.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>Register</title>
</head>
<body>
    <div class="container">
        <h2>Реєстрація</h2>
        <form action="register_process.php" method="post">
            <div class="form-group">
            <label for="username">Ваше ім'я':</label>
            <input type="text" name="username" required>
            </div>

            <div class="form-group">
            <label for="password">Ваш пароль:</label>
            <input type="password" name="password" required>
            </div>

            <div class="form-group">
            <button type="submit">Реєстрація</button>
            </div>
        </form>
        <p>Вже є аккаунт? <a href="login.php">Авторизуйтесь</a>.</p>
    </div>
</body>
</html>
```

Рисунок 4.4 – HTML код сторінки реєстрації

```php
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>Login</title>
</head>
<body>

<?php
session_start();

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $username = $_POST["username"];
    $password = password_hash($_POST["password"], PASSWORD_DEFAULT);

    $db_host = "localhost";
    $db_user = "root";
    $db_password = "";
    $db_name = "mydatabase";

    $conn = new mysqli($db_host, $db_user, $db_password, $db_name);

    if ($conn->connect_error) {
        die("Connection failed: " . $conn->connect_error);
    }

    // Перевірка, чи не існує вже користувач з таким ім'ям
    $stmt = $conn->prepare("SELECT id FROM users WHERE username = ?");
    $stmt->bind_param("s", $username);
    $stmt->execute();
    $stmt->store_result();

    // Якщо користувач з таким ім'ям вже існує, вивести повідомлення та завершити виконання
    if ($stmt->num_rows > 0) {
        echo '<div class="form-group">';
        echo '<p>Користувач з даним ім'ям існує.Спробуйте інше.</p>';
        echo '<a href="index.php" class="back-button">Назад</a>';
        echo '<div>';
        $stmt->close();
        $conn->close();
        exit();
    }

    $stmt->close();

    // Вставка нового користувача тільки якщо його не існує
    $stmt = $conn->prepare("INSERT INTO users (username, password) VALUES (?, ?)");
    $stmt->bind_param("ss", $username, $password);
    $stmt->execute();
    $stmt->close();

    $conn->close();

    header("Location: index.php");
    exit();
}
?>

</body>
</html>
```

Рисунок 4.5 – PHP основа для форми реєстрації

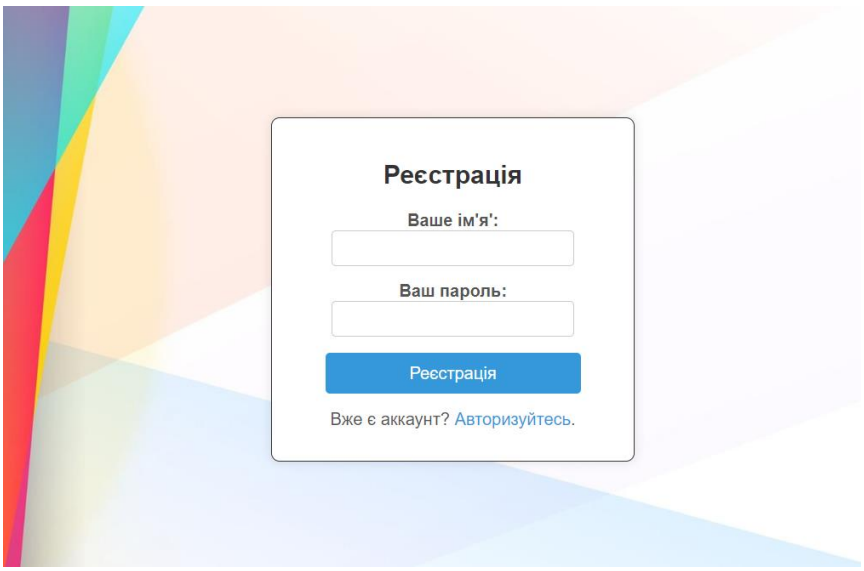

Рисунок 4.6 – Зовнішній вигляд сторінки реєстрації

А також вистроєно подібну схему для меню авторизації зареєстрованого користувача.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>Login</title>
</head>
<body>
    <div class="container">
        <h2>Авторизація</h2>
        <form action="login_process.php" method="post">
            <div class="form-group">

                <div class="label-container">
                <label for="username">Ваше ім'я:</label>
                <input type="text" name="username" required>
            </div>

            <div class="form-group">
                <label for="password">Ваш пароль:</label>
                <input type="password" name="password" required>
            </div>

            <div class="form-group">
                <button type="submit">Увійти</button>
            </div>

        </form>
        <p>Немає облікового запису? <a href="register.php">Зареєструйтесь</a>.</p>
        <a href="google_login.php">
            <img src="google_logo.png" class="google-logo">
            Увійти з Google
        </a>
    </div>
</body>
</html>
```

Рисунок 4.7 - HTML код сторінки авторизації

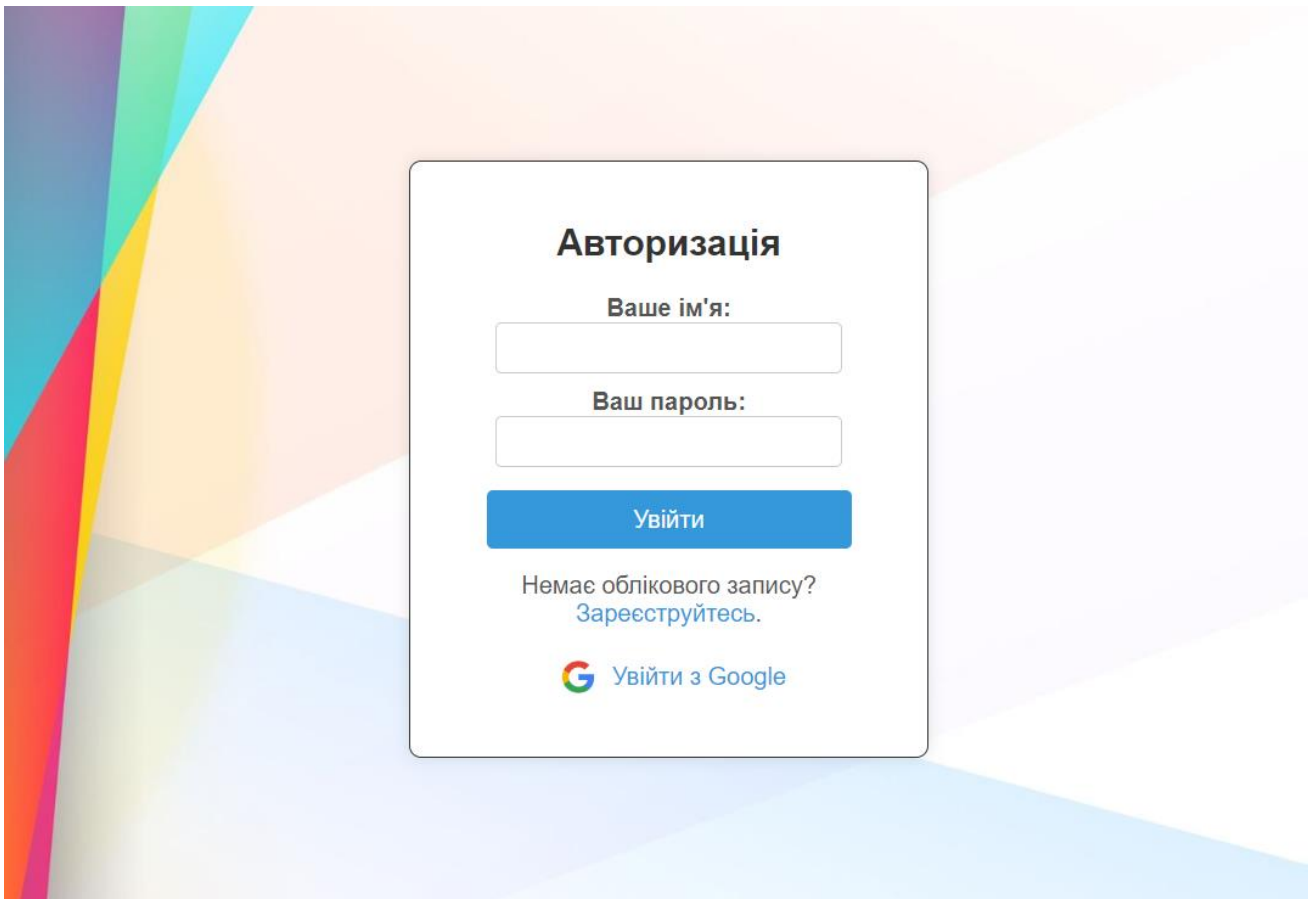

Рисунок 4.8 - Зовнішній вигляд сторінки авторизації

Більшій дій пов'язаних з автентифікацією описані у коді нижче:

```php
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>Login</title>
</head>
<body>

<?php
require 'vendor/autoload.php';
use OTPHP\TOTP;

if (session_status() == PHP_SESSION_NONE) {
    session_start();
}

$google2fa = new \PragmaRX\Google2FA\Google2FA();

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    if (isset($_POST["google2fa_code"])) {
        $username = $_SESSION["username"];
        $user_provided_code = $_POST["google2fa_code"];

        $db_host = "localhost";
        $db_user = "root";
        $db_password = "";
        $db_name = "mydatabase";

        $conn = new mysqli($db_host, $db_user, $db_password, $db_name);

        if ($conn->connect_error) {
            die("Connection failed: " . $conn->connect_error);
        }

        $stmt = $conn->prepare("SELECT id, username, password, secret FROM users WHERE username = ?");
        $stmt->bind_param("s", $username);
        $stmt->execute();
        $stmt->bind_result($user_id, $db_username, $db_password_hash, $db_secret);
        $stmt->fetch();
        $stmt->close();

        if ($user_id) {
            $otp = TOTP::create($db_secret);
            $current_otp = $otp->now();

            echo '<div class="form-group">';
            //echo "Generated OTP: " . $current_otp;
            echo '<div>';

            if ($otp->verify($user_provided_code)) {
                $_SESSION['user_id'] = $user_id;
                $_SESSION['authenticated'] = true;
                header("Location: index.php");
                exit();
            } else {
                echo '<p>Авторизація невдала. Введено неправильний Google Authenticator код.</p>';
                echo '<a href="index.php" class="back-button">Назад</a>';
            }
        } else {
            echo "Авторизація невдала. Користувача не знайдено";
        }

        $conn->close();
    } else {
        $username = isset($_POST["username"]) ? $_POST["username"] : null;
        $password = isset($_POST["password"]) ? $_POST["password"] : null;
```

Рисунок 4.9 – Основний код з 2 видами автентифікації (Частина перша)

```
        if ($username === null || $password === null) {
            echo "Недійсне надсилання форми.";
            exit();
        }

        $db_host = "localhost";
        $db_user = "root";
        $db_password = "";
        $db_name = "mydatabase";

        $conn = new mysqli($db_host, $db_user, $db_password, $db_name);

        if ($conn->connect_error) {
            die("Connection failed: " . $conn->connect_error);
        }

        $stmt = $conn->prepare("SELECT id, username, password, secret FROM users WHERE username = ?");
        $stmt->bind_param("s", $username);
        $stmt->execute();
        $stmt->bind_result($user_id, $db_username, $db_password_hash, $db_secret);
        $stmt->fetch();
        $stmt->close();

        if ($user_id && password_verify($password, $db_password_hash)) {
            $_SESSION["username"] = $username;
            $_SESSION["password"] = $password;

            if (!$db_secret) {
                $db_secret = $google2fa->generateSecretKey();

                $stmt = $conn->prepare("UPDATE users SET secret = ? WHERE id = ?");
                $stmt->bind_param("si", $db_secret, $user_id);
                $stmt->execute();
                $stmt->close();
            }

            // Generate QR code
            $text = $google2fa->getQRCodeUrl($username, 'http://localhost', $db_secret);
            $image_url = 'https://chart.googleapis.com/chart?cht=qr&chs=300x300&chl=' . $text;

            echo '<div class="container">';
            echo '<h2>Відскануйте QR-код Google Authenticator</h2>';
            echo '<img src="' . $image_url . '" /><br>';
            echo '<form method="post">';
            echo '<div class="form-group">';
            echo '<label for="google2fa_code">Введіть код:</label>';
            echo '<input type="text" name="google2fa_code" required>';
            echo '</div>';
            echo '<div class="form-group">';
            echo '<button type="submit">Увійти</button>';
            echo '</div>';
            echo '</form>';
            echo '</div>';
        } else {
            echo '<div class="form-group">';
            echo '<p>Авторизація невдала. Перевірте правильність свого паролю або імені.</p>';
            echo '<a href="index.php" class="back-button">Назад</a>';
            echo '<div>';
        }

        $conn->close();
    }
}
?>

</body>
</html>
```

Рисунок 4.10 – Основний код з 2 видами автентифікації (Частина друга)

В сторінках вище було встановлено кнопки переходу до відповідних розділів, обмеження на введення пустого тексту у поля, обмеження на введення існуючого імені користувача при реєстрації та інтуїтивно зрозумілий користувальницький інтерфейс.

Окрім звичайної автентифікації паролем та іменем користувача, в цілях захисту було додано двофакторну автентифікацію через мобільний додаток Google Authenticator за допомогою QR-коду, що генерується на екрані web-сайту.

Google Authenticator - це мобільний додаток для двофакторної аутентифікації, розроблений Google. Додаток генерує одноразові паролі (OTP) або часово-залежні коди, які використовуються для підтвердження ідентичності користувача під час входу в обліковий запис на різних онлайн-платформах.

Google Authenticator генерує шестизначні одноразові паролі, які зазвичай діють протягом короткого періоду часу (наприклад, 30 секунд). Користувач вводить цей код разом із своїм основним паролем для здійснення двофакторної аутентифікації.

Основою роботи Google Authenticator є алгоритм генерації кодів, який використовує поточний час і секретний ключ користувача. Це дозволяє системі, яка перевіряє код, знаходити його дійсність в певний час. Можливість використання без Інтернет-з'єднання – це одна з переваг Google Authenticator. Вона полягає в тому, що для генерації кодів не потрібне Інтернет-з'єднання. Коди генеруються безпосередньо на мобільному пристрої користувача.

Google Authenticator дозволяє додавати та керувати багатьма обліковими записами для різних сервісів та платформ. Кожен обліковий запис має свій унікальний секретний ключ для генерації власних кодів.

Багато онлайн-сервісів та платформ, таких як Google, Facebook, Dropbox, а також багато інших, підтримують Google Authenticator як засіб для встановлення двофакторної аутентифікації.

При додаванні облікового запису Google Authenticator використовує QR-код або ручний введення секретного ключа для встановлення зв'язку між додатком і сервісом.

Окрім генерації кодів, Google Authenticator також може підтримувати інші методи аутентифікації, такі як коди, які надсилаються через SMS або телефонний дзвінок.

Загалом, дана программа є популярним та зручним інструментом для забезпечення додаткового рівня безпеки при вході в онлайн-облікові записи.

Під час реєстрації генерується унікальний секрет для 2FA для нового користувача. Для цього використовується бібліотека OTP.

Секрет зберігається в базі даних разом із іншими обліковими даними користувача, такими як логін та захешований пароль.

При скануванні QR-коду користувачу надається ідентифікаційний цифровий 6-значний OTP код (окремо згенерований для кожного зареєстрованого власника кабінету), що дозволяє увійти у систему. OTP код автоматично оновлюється з інтервалом кожні 30 секунд та оновлюється у мобільному додатку. PHP-код перевіряє введений користувачем OTP. Якщо введений пароль у окремому полі веб-додатку збігається з тим, що створено для користувача, то система переводить на сторінку успішного входу. В іншому випадку видає помилку.

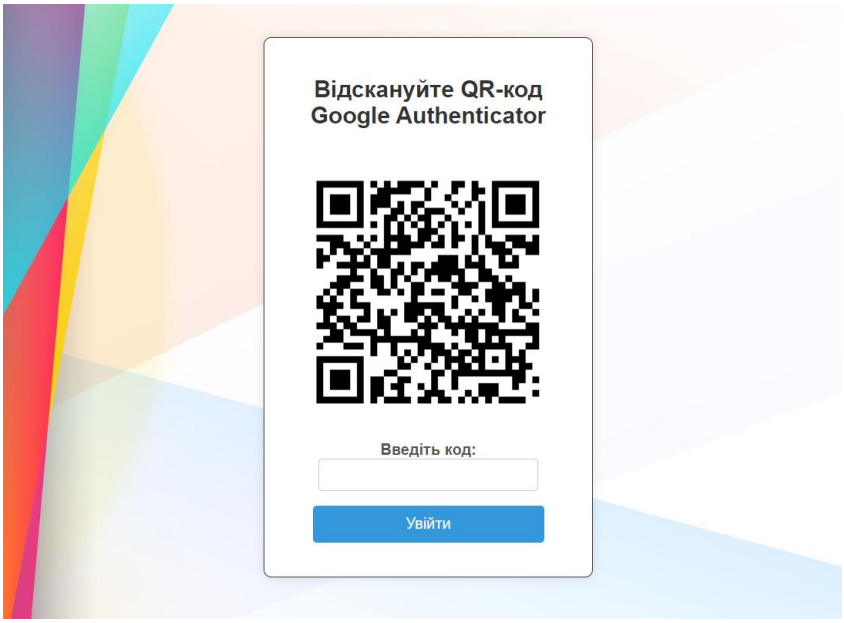

Рисунок 4.11 – Відображення згенерованого QR коду

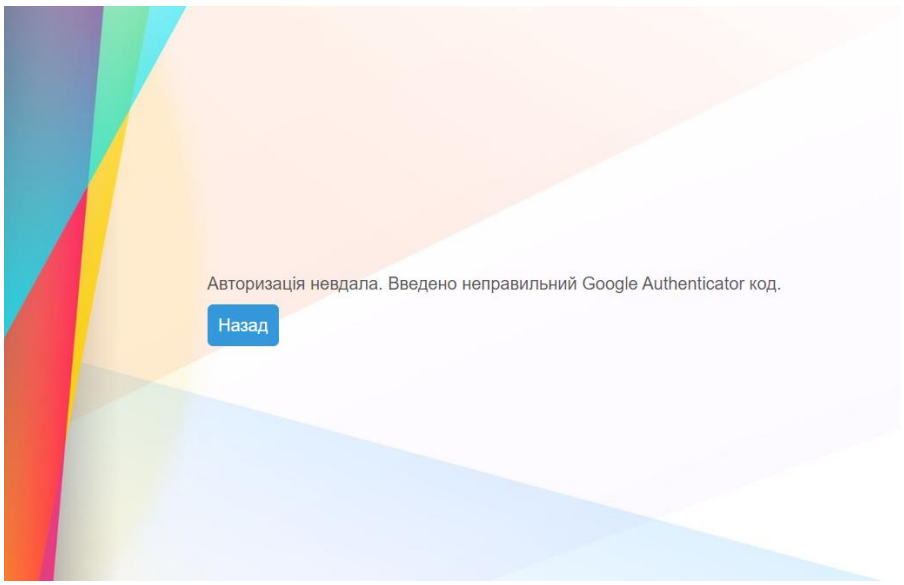

Рисунок 4.12 – Вікно помилки при неправильному введенні Google коду

Приблизно аналогічна помилка відображається при неправильному введенні пароля або логіна при першому етапі автентифікації чи спробі зареєструвати аккаунт, що уже є у базі даних.

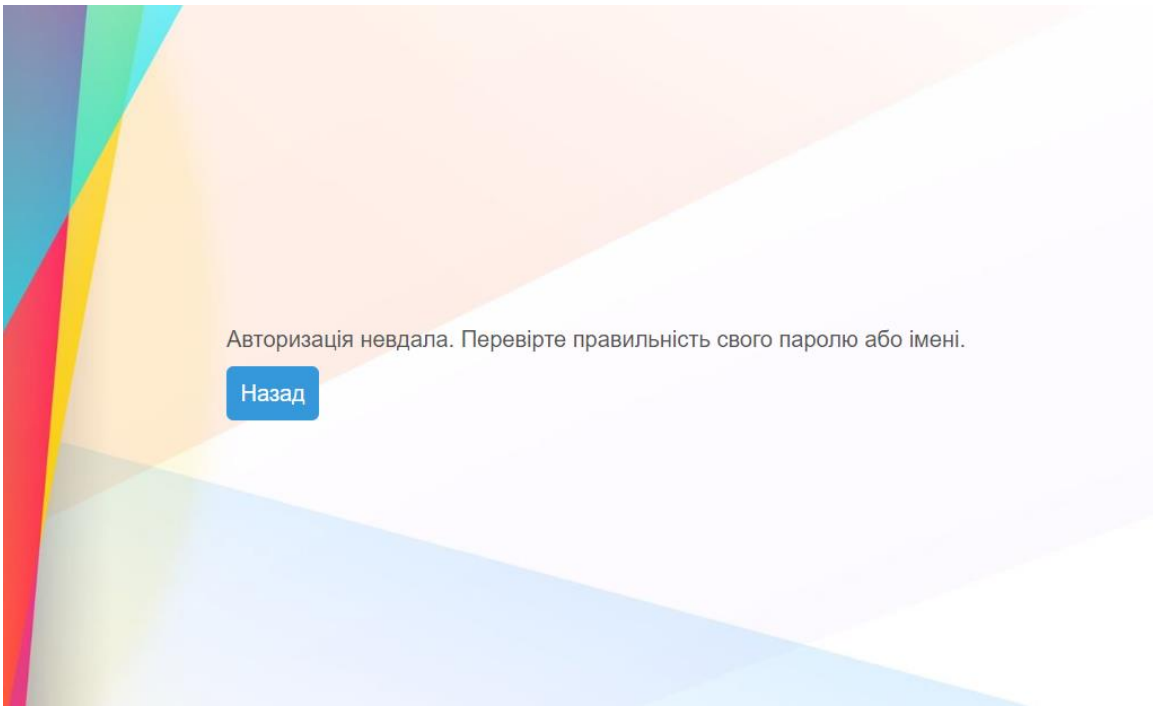

Рисунок 4.13 – Вікно помилки при неправильному введенні паролю та імені

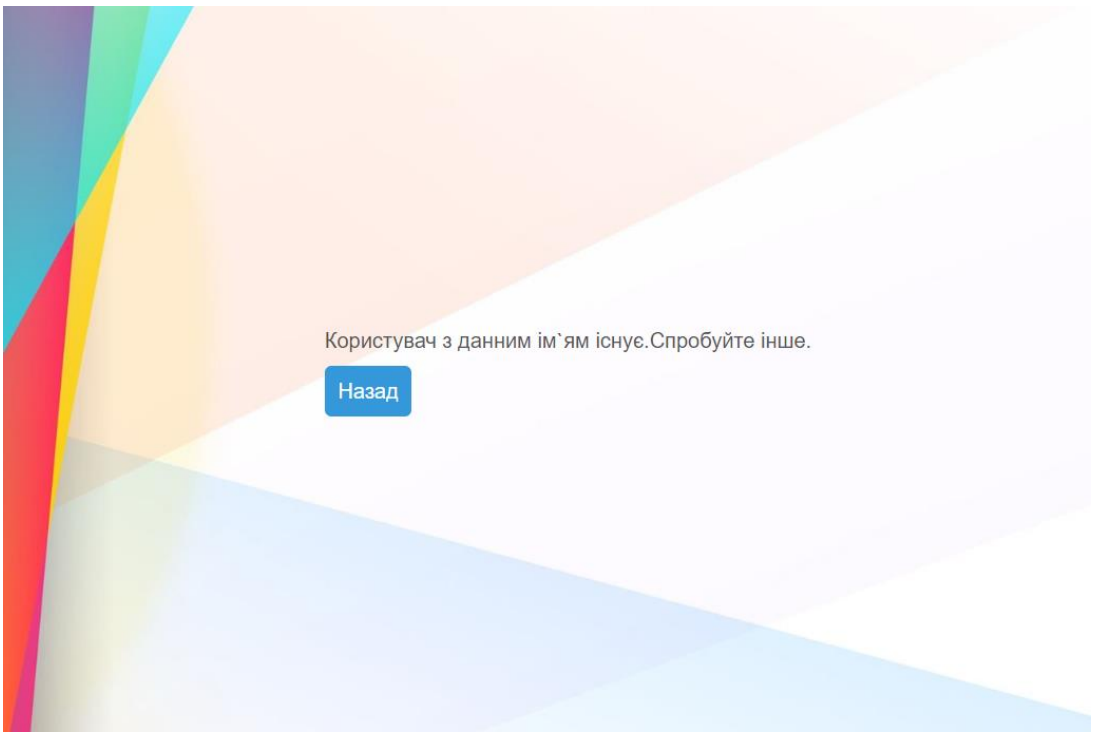

Рисунок 4.14 – Вікно помилки при однаковій повторній реєстрації

Проходження двох етапів входу виводить повідомлення у стилі привітання.

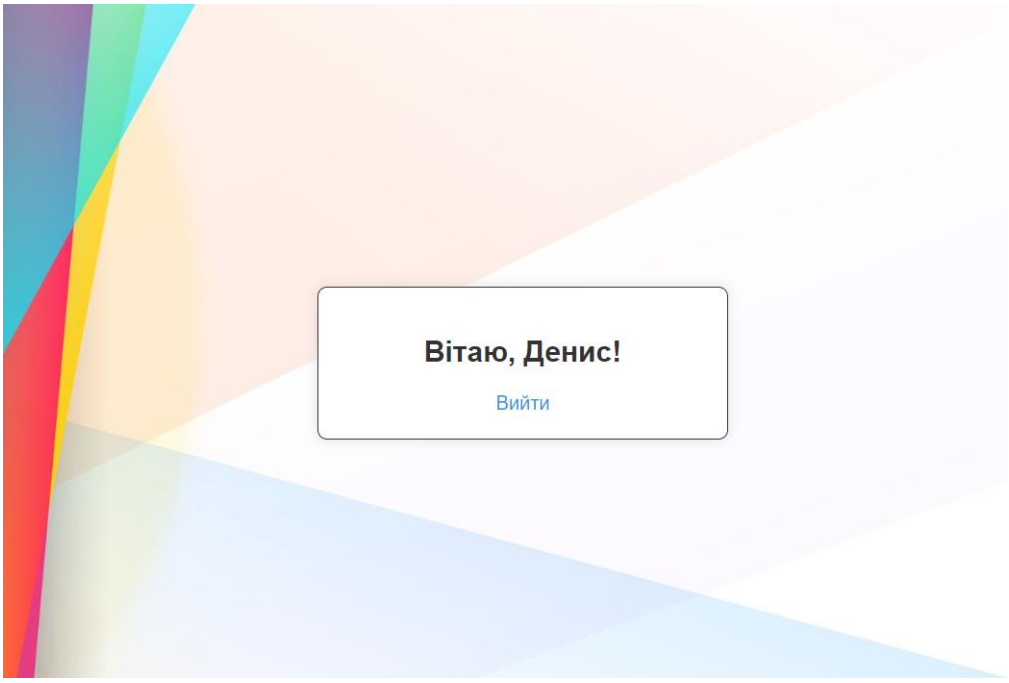

Рисунок 4.15 – Успішна автентифікація у WEB-додаток

```php
<!DOCTYPE html>
<html lang="uk"
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="style.css">
    <title>Home</title>
</head>
<body>
    <div class="container">
        <?php
            session_start();

            if (isset($_SESSION['user_id'])) {
                echo '<h2>Вітаю, ' . $_SESSION['username'] . '!</h2>';
                echo '<a href="logout.php">Вийти</a>';
            } else {
                echo '<h2>Головна сторінка!</h2>';
                echo '<a href="login.php">Авторизація</a> | <a href="register.php">Реєстрація</a>';
            }
        ?>
    </div>
</body>
</html>
```

Рисунок 4.16 HTML код для відображення кінцевого результату

```php
<?php
    session_start();
    session_destroy();
    header("Location: index.php");
    exit();
?>
```

Рисунок 4.17 HTML код для завершення поточної сесії користувача

Для даного коду було підвантажено необідні программи (XAMMP, Ngrok, Composer) та бібліотеки (google/apiclient, pragmarx/google2fa, spomky-labs/otphp)

Для більшої зручності додано авторизацію через Google аккаунт.

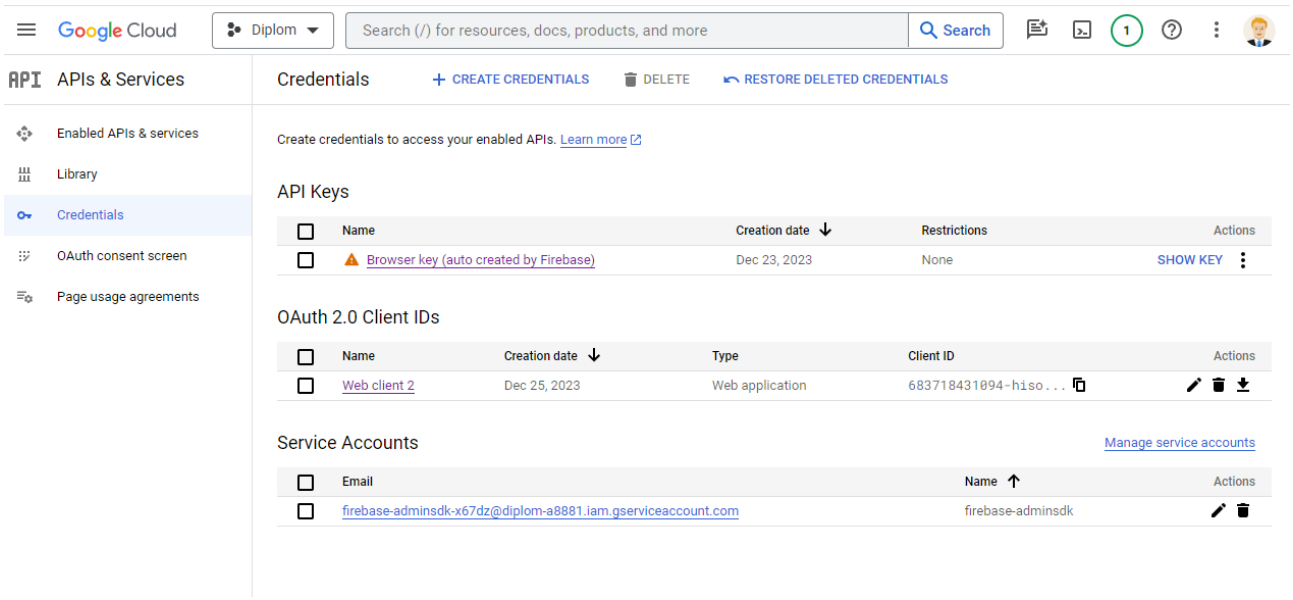

Рисунок 4.18 – Доданий проект у Google Developers Console

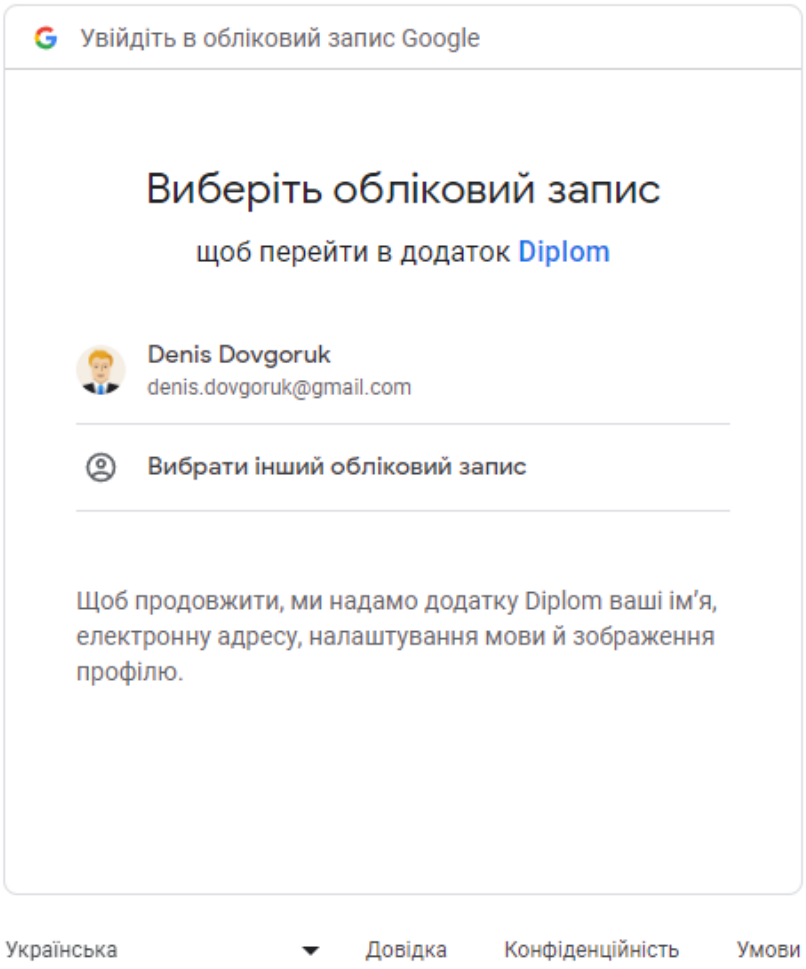

Рисунок 4.19 – Авторизація через Google аккаунт

Веб сторінка створюється на основі локального веб серверу. Для правильної роботи авторизації Google необхідно забезпечити зовнішній доступ до цього сервера. Це важливо, оскільки Google Developers Console не сприймає URL, що недоступний в Інтернеті і не дозволяє розмістити його у полі для зворотнього посилання. З цією метою використовуються інструменти, такі як Ngrok, які надають зовнішній доступ до локального сервера для забезпечення корректної роботи Google.

Використання авторизації через обліковий запис Google може бути більш безпечним порівняно із звичайним введенням паролю з декількох причин:

1. Менший ризик витоку паролю: Коли ви використовуєте авторизацію через обліковий запис Google, вам не потрібно створювати та зберігати пароль для свого власного аутентифікаційного механізму. Це зменшує ризик витоку паролю через атаки на сервер, базу даних або інші канали.
2. Двофакторна аутентифікація (2FA): Більшість облікових записів Google підтримують двофакторну аутентифікацію, що надає додатковий рівень безпеки. Після введення паролю може знадобитися ще один крок підтвердження (наприклад, код, відправлений на мобільний телефон), що робить важчим несанкціонований доступ.
3. Управління безпекою: Google витрачає значні зусилля на забезпечення безпеки своїх облікових записів. Вони використовують різні заходи безпеки, такі як моніторинг ненормальної активності, шифрування та інші технології безпеки.
4. Більша своєчасність безпеки: Google постійно оновлює свої механізми безпеки та реагує на нові загрози. Використовуючи їхній механізм авторизації, ви автоматично користуєтеся їхніми оновленнями та вдосконаленнями безпеки.

Загальный вигляд файлів роботи виглядає наступним чином:

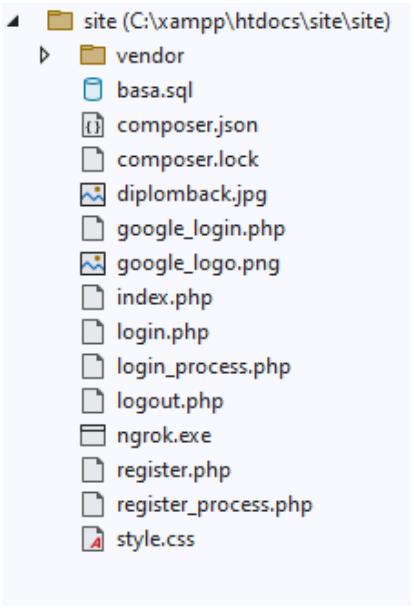

Рисунок 4.20 – Усі файли для виконання практичного розділу

ВИСНОВКИ

Це дослідження прокладає шлях для майбутніх досліджень у сфері методів автентифікації для веб-додатків. Подальші дослідження інтеграції штучного інтелекту для адаптивної автентифікації, дослідження квантово-стійких криптографічних методів і поглиблені дослідження аспектів зручності використання нових технологій представляють багатообіцяючі напрямки для продовження досліджень.

Дослідження нових технологій автентифікації, таких як біометрія та децентралізоване керування ідентифікацією, підкреслює їхній потенціал революції в системі безпеки. Біометричні методи, з притаманною їм унікальністю та можливістю адаптації, пропонують багатообіцяючі шляхи надійної та зручної аутентифікації. Децентралізований характер управління ідентифікацією за допомогою блокчейну змінює парадигму, підвищуючи конфіденційність і стійкість до централізованих атак. У міру розвитку цих технологій їх інтеграція в основні веб-додатки стає все більш життєздатною.

Дослідження висвітлило проблеми та компроміси, притаманні пошукам вдосконалених методів автентифікації. Хоча передові технології обіцяють підвищену безпеку, вони також можуть створити нові складності та занепокоєння щодо конфіденційності. Встановлення тонкого балансу між безпекою та досвідом користувача залишається проблемою, особливо тому, що сприйняття користувачами та простота використання є основними факторами широкого впровадження механізмів автентифікації.

Орієнтований на користувача підхід до дизайну автентифікації стає ключовим моментом. Навчання користувачів важливості безпечних методів у поєднанні з інтуїтивно зрозумілими та доступними інтерфейсами автентифікації може значно підвищити загальну безпеку. Визнаючи, що кінцевий користувач є невід’ємною частиною рівняння безпеки, зусилля повинні бути спрямовані на

мінімізацію тертя та максимізацію розуміння користувачами процесів автентифікації.

Оскільки кіберзагрози продовжують розвиватися, дослідження підкреслює необхідність постійної адаптації та пильності. Безпека – це не статичний стан, а динамічний процес, який вимагає постійних досліджень, розробки та впровадження інноваційних рішень. Організації та розробники повинні залишатися активними у відстеженні нових загроз і відповідній адаптації методів автентифікації.

Підсумовуючи результат роботи, можна сказати, що було проаналізовано сферу алгоритмів автентифікації користувачів у WEB-додатках, спрямованих на підвищення безпеки, поглиблено знання у даній сфері інформаційних технологій, було розглянуто проблеми, пов’язані з традиційними методами шифрування, досліджено досягнення в алгоритмах та запропоновано стратегії для ефективного впровадження безпечної автентифікації користувачів у WEB-додатках.

ПЕРЕЛІК ПОСИЛАНЬ

1. Автентифікація (веб)
   https://uk.wikipedia.org/wiki/%D0%90%D0%B2%D1%82%D0%B5%D0%BD%D1%82%D0%B8%D1%84%D1%96%D0%BA%D0%B0%D1%86%D1%96%D1%8F_(%D0%B2%D0%B5%D0%B1)
   [Електронний ресурс]
2. Popular Authentication Methods for Web Apps
   https://www.baeldung.com/cs/authentication-web-apps
   [Електронний ресурс]
3. https://habr.com/
   [Електронний ресурс]
4. Web Authentication Methods Compared
   https://testdriven.io/blog/web-authentication-methods/
   [Електронний ресурс]
5. Кіберварта. Як український бізнес захищає дані від російських хакерів
   https://delo.ua/telecom/kibervarta-yak-ukrayinskii-biznes-zaxishhaje-dani-vid-rosiiskix-xakeriv-421930/
   [Електронний ресурс]
6. Google Authenticator
   https://support.google.com/accounts/answer/1066447?hl=uk&co=GENIE.Platform%3DAndroid
   [Електронний ресурс]
7. Popular Authentication Methods for Web Apps
   https://www.baeldung.com/cs/authentication-web-apps
   [Електронний ресурс]
8. Comparison of web authentication methodology
   https://testdriven.io/blog/web-authentication-methods/
   [Електронний ресурс]

9. Composer
https://getcomposer.org/
[Електронний ресурс]
10. XAMPP Apache + MariaDB + PHP + Perl
https://www.apachefriends.org/ru/index.html
[Електронний ресурс]
11. How to Configure a Local NTP Server
https://techlibrary.hpe.com/docs/otlink-wo/How-to-Configure-a-Local-NTP-Server.html
[Електронний ресурс]
12. PHP With MySQL: Ultimate Step-By-Step Guide
https://www.simplilearn.com/tutorials/php-tutorial/php-with-sql
[Електронний ресурс]
13. PHP Manual
https://www.php.net/manual/en/index.php
[Електронний ресурс]
14. Authentication attacks
https://www.ibm.com/docs/en/snips/4.6.0?topic=categories-authentication-attacks
[Електронний ресурс]
15. Authentication Methods For Web Applications
https://logmeonce.com/resources/authentication-methods-for-web-applications/
[Електронний ресурс]
16. What is Authentication? Different Types of Authentication
https://www.miniorange.com/blog/different-types-of-authentication-methods-for-security/
[Електронний ресурс]
17. Кіберінциденти очолили список головних бізнес-ризиків — Allianz Risk Barometer 2023
https://10guards.com/ua/articles/cyber-incidents-among-top-business-risks-

allianz-risk-barometer-2023/
[Електронний ресурс]
18. Two Factor Authentication (2FA) with Google
https://medium.com/@richb_/easy-two-factor-authentication-2fa-with-google-authenticator-php-108388a1ea23
[Електронний ресурс]
19. QR Login in PHP
https://dev.to/sahilkashyap64/qr-login-in-php-2pgf
[Електронний ресурс]
20. Основы CSS
https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/CSS_basics
[Електронний ресурс]
21. Основы HTML
https://developer.mozilla.org/ru/docs/Learn/Getting_started_with_the_web/HTML_basics
[Електронний ресурс]
22. The Definitive Guide to Authentication
https://www.strongdm.com/authentication
[Електронний ресурс]
23. Biometric Authentication
https://www.logintc.com/types-of-authentication/biometric-authentication/
[Електронний ресурс]
24. What Is Token-Based Authentication?
https://www.okta.com/identity-101/what-is-token-based-authentication/
[Електронний ресурс]
25. Two-Factor Authentication (2FA)
https://www.onespan.com/topics/two-factor-authentication
[Електронний ресурс]

26. What is Behavioral Biometrics?
https://expertinsights.com/insights/a-guide-to-behavioral-biometrics/
[Електронний ресурс]
27. Hardware and Software Authentication: Choosing the Right Approach
https://www3.thalesgroup.com/adwords/authentication/whitepaper/assets/StrongAuthentication_DG_(EN)_web.pdf
[Електронний ресурс]
28. Understanding push authentication
https://www.twilio.com/blog/understanding-push-authentication
[Електронний ресурс]
29. Risk-based authentication (RBA)
https://www.techtarget.com/searchsecurity/definition/risk-based-authentication-RBA
[Електронний ресурс]
30. Blockchain in Digital Identity
https://consensys.io/blockchain-use-cases/digital-identity
[Електронний ресурс]
31. What is Certificate-Based Authentication?
https://www.yubico.com/resources/glossary/what-is-certificate-based-authentication/
[Електронний ресурс]
32. What is zero trust?
https://www.ibm.com/topics/zero-trust
[Електронний ресурс]
33. Zero Trust Security
https://www.fortinet.com/br/resources/cyberglossary/what-is-the-zero-trust-network-security-model
[Електронний ресурс]