Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ДОПУСТИТИ ДО ЗАХИСТУ:
В.о. завідувача кафедри
кібербезпеки та захисту інформації
_____ Іван ПАРХОМЕНКО
«____» червня 2023 р.

ПОЯСНЮВАЛЬНА ЗАПИСКА

кваліфікаційної роботи

галузь знань 12 Інформаційні технології
(шифр і назва галузі знань)
спеціальність 125 Кібербезпека
(код і назва спеціальності)
освітній ступень бакалавр
освітня програма Кібербезпека
(назва освітньо-професійної програми)
на тему: Засоби вдосконалення технологій доступу до захищених Інформаційних ресурсів

Виконавець: студентка IV курсу, групи КБ-43мс

_____ **Анастасія ЛИХВАР**
(підпис) (ім'я, прізвище)

	Ім'я, прізвище	**Підпис**
Керівник	Яніна ШЕСТАК	

Нормоконтроль	Сергій ДАКОВ	

Київ 2023

Міністерство освіти і науки України
Київський національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
В.о. завідувача кафедри кібербезпеки
та захисту інформації
_____ Сергій ТОЛЮПА
«24» жовтня 2022 р.

ЗАВДАННЯ
на виконання кваліфікаційної роботи

спеціальності 125 Кібербезпека
(код і назва спеціальності)

освітньої програми Кібербезпека
(назва освітньо-професійної програми)

Студентці **КБ-43мс** **Анастасії Валентинівни Лихвар**
(група) (прізвище ім'я по батькові)

Тема кваліфікаційної роботи Засоби вдосконалення технологій доступу до захищених інформаційних ресурсів

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Тема кваліфікаційної роботи затверджена на засіданні кафедри кібербезпеки та захисту інформації протокол №3 від 20.10.2022 р.

2. ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Технології двохфакторної аутентифікації, шифрування

3. ЗМІСТ РОЗРАХУНКОВО-ПОЯСНЮВАЛЬНОЇ ЗАПИСКИ

Необхідно ознайомитися з технологією двохфакторної аутентифікаціїї, видами, вразливостями з боку безпеки даних, обрати метод шифрування, проаналізувати їх, розробити по підвищенню безпеки та реалізувати їх

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Практична цінність Розроблені рекомендації з підвищення безпеки та Практичного використання двохфакторної аутентифікації

5. ДАТА ВИДАЧІ ЗАВДАННЯ

Дата видачі завдання: 24 жовтня 2022 року

Завдання видав _____ (підпис) Яніна ШЕСТАК (ім'я, прізвище)

Завдання прийняла до виконання _____ (підпис) Анастасія ЛИХВАР (ім'я, прізвище)

КАЛЕНДАРНИЙ ПЛАН

№ п/п	Найменування етапів робіт	Строки виконання робіт (початок-кінець)	Відмітка про виконання
1	Уточнення постановки задачі	24.10.2022 – 4.11.2022	*виконано*
2	Аналіз літератури	28.01.2023 –20.02.2023	*виконано*
3	Обґрунтування вибору рішення	21.02.2023 – 23.02.2023	*виконано*
4	Огляд нормативно-правової бази забезпечення захисту інформаційних ресурсів	24.02.2023 – 10.03.2023	*виконано*
5	Аналіз існуючих засобів вдосконалення технологій доступу до захищених інформаційних ресурсів	12.03.2023 – 24.03.2023	*виконано*
6	Аналіз захисних механізмів для забезпечення цілісності, конфіденційності та доступності інформаційних ресурсів	25.03.2023 – 07.04.2023	*виконано*
7	Вибір технологій доступу до захищених інформаційних ресурсів	07.04.2023 – 12.04.2023	*виконано*
8	Розробка програмного рішення	13.04.2023 – 12.05.2023	
9	Оформлення пояснювальної записки	13.06.2023 – 05.06.2023	*виконано*
10	Підготовка до захисту кваліфікаційної роботи	05.06.2023 – 19.06.2023	*виконано*

Завдання видав _____ (підпис) Яніна ШЕСТАК (ім'я, прізвище)

Завдання прийняв до виконання _____ (підпис) Анастасія ЛИХВАР (ім'я, прізвище)

Термін подання кваліфікаційної роботи до ЕК 12 червня 2023 року

РЕФЕРАТ

Пояснювальна записка дипломної роботи складається зі вступу, трьох розділів, загальних висновків, списку використаних джерел та додатків. Основний текст займає 61 сторінок, включає в себе зміст, вступ, три розділи дипломної роботи, висновки та список джерел. Крім того, робота містить 5 додатків із загальною кількістю сторінок 15. У пояснювальній записці дипломної роботи міститься 32 рисунка, 8 таблиць та 30 літературних джерел.

Метою роботи є вдосконалення технології доступу до захищених інформаційних ресурсів.

Для досягнення мети необхідно вирішити **наступні завдання**:

- огляд аспектів забезпечення захисту інформаційних ресурсів;
- дослідження методів і засобів захисту інформаційних ресурсів;
- вдосконалення технології доступу до захищених інформаційних ресурсів.

Об'єктом дослідження є процес доступу до захищених інформаційних ресурсів.

Предметом дослідження є засіб вдосконалення технологій доступу до захищених інформаційних ресурсів.

Методи дослідження:

- аналіз відкритих джерел;
- порівняння та протиставлення різних технологій;
- вдосконалення технології доступу до захищених інформаційних ресурсів»;

Практична цінність дослідження може допомогти організаціям випереджати постійно мінливий ландшафт загроз і забезпечити безпеку та захист своїх інформаційних ресурсів.

Практична новизна*:* запропонована система використовує мобільні пристрої як другий фактор у процесі автентифікації. Завдяки додатковому рівню перевірки, система значно зменшила ризик несанкціонованого доступу та підвищила безпеку інформаційних ресурсів.

Ключові слова*:* двохфакторна аутентифікація, пароль, шифрування, автентифікація, вразливості, захист персональних даних, виявлення та запобігання вторгнень, багатофакторної автентифікації.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕННЬ

DevSecOps	–	Development, Security, and Operation;
IDS	–	Intrusion Detection System
IDPS	–	Intrusion Detection and Prevention System
IPS	–	Intrusion Prevention System
MFA	–	Multifactor authentication
OOB	–	Out of Band
OSINT	–	Open Source Intelligence
IT	–	Information Technology
TOTP	–	Time-Based One-Time Password
2FA	–	Двухфакторна автентифікація
БФА	–	Багатофакторна аутентифікація
ЗІ	–	Захист інформації
ЗУ	–	Закон України
ІБ	–	Інформаційна безпека
ІС	–	Інформаційна система
ПЗ	–	Програмне забезпечення
НД	–	Нормативний документ

ЗМІСТ

ВСТУП

Сьогодні захист інформації є однією з найбільш актуальних проблем суспільства, особливо в контексті швидкого розвитку інформаційних технологій. Це пов'язано з тим, що інформація стала найціннішим ресурсом сучасного світу, який потребує надійного захисту від несанкціонованого доступу.

Технології захисту інформації стають все складнішими і вимагають постійного вдосконалення. Одним із важливих аспектів захисту інформації є технології доступу до захищених інформаційних ресурсів. У зв'язку з цим актуальним є питання вдосконалення засобів і механізмів доступу до таких ресурсів, які забезпечать їх надійний захист від несанкціонованого доступу.

Механізми управління доступом стали критично важливим елементом захисту інформаційних ресурсів. Зростаюча витонченість кіберзагроз і все більша залежність від технологій в сучасних організаціях роблять необхідним постійний перегляд і вдосконалення технологій контролю доступу. Ця теза особливо актуальна в наш час, коли інформація є джерелом життєдіяльності організацій, а її захист має вирішальне значення для їх виживання.

Метою роботи є вдосконалення технології доступу до захищених інформаційних ресурсів.

Для досягнення мети необхідно вирішити **наступні завдання**:

- огляд аспектів забезпечення захисту інформаційних ресурсів;
- дослідження методів і засобів захисту інформаційних ресурсів;
- вдосконалення технології доступу до захищених інформаційних ресурсів.

Об’єктом дослідження є процес доступу до захищених інформаційних ресурсів.

Предметом дослідження є засіб вдосконалення технологій доступу до захищених інформаційних ресурсів.

Методи дослідження:

- аналіз відкритих джерел;

- порівняння та протиставлення різних технологій;
- вдосконалення технології доступу до захищених інформаційних ресурсів»;

Практична цінність дослідження може допомогти організаціям випереджати постійно мінливий ландшафт загроз і забезпечити безпеку та захист своїх інформаційних ресурсів.

РОЗДІЛ 1
ТЕОРЕТИЧНІ АСПЕКТИ ЗАБЕЗПЕЧЕННЯ ЗАХИСТУ ІНФОРМАЦІЙНИХ РЕСУРСІВ

1.1 Нормативно-правова база забезпечення захисту інформаційних ресурсів

Нормативно-правова база забезпечення захисту інформаційних ресурсів в Україні є важливим аспектом дослідження засобів удосконалення технологій доступу до захищених інформаційних ресурсів. Під нормативно-правовою базою розуміється законодавча база та нормативні положення, які регулюють питання захисту інформаційних ресурсів в Україні.

Тож, ось деякі з нормативно-правових актів України для забезпечення захисту інформаційних ресурсів:

Закон України "Про інформацію"

Цей закон визначає інформацію як будь-які відомості, незалежно від їх форми, які створені, збережені, оброблені або передані за допомогою інформаційно-комунікаційних технологій. Закон також встановлює принципи захисту інформації, включаючи право на інформацію, право на недоторканність приватного життя та право на безпеку інформації.

Закон України "Про захист інформації в автоматизованих системах та інформаційних мережах"

Цей закон регулює питання захисту інформації в комп'ютерних системах та інформаційних мережах. Закон встановлює вимоги до безпеки інформаційних систем та мереж, включаючи використання заходів безпеки, навчання персоналу та інформування про інциденти безпеки.

Закон України "Про охорону державної таємниці"

Цей закон регулює питання захисту державної таємниці. Закон визначає державну таємницю як відомості, що мають конфіденційний характер і розголошення яких може завдати шкоди національній безпеці України. Закон встановлює порядок віднесення, охорони та розсекречення державної таємниці.

Закон України "Про захист персональних даних"

Цей закон регулює питання захисту персональних даних. Закон визначає персональні дані як будь-яку інформацію про фізичну особу, яка може бути ідентифікована. Закон встановлює вимоги до збору, обробки та використання персональних даних, включаючи згоду особи, безпеку даних та право особи на доступ до своїх персональних даних та їх виправлення.

Окрім цих законів, існує також низка інших нормативно-правових актів, які застосовуються до захисту інформаційних ресурсів в Україні. До них відносяться:

- Положення про порядок забезпечення охорони державної таємниці в інформаційних системах;
- Положення про порядок захисту персональних даних в інформаційних системах;
- Положення про державну систему захисту інформації.

Ці нормативно-правові акти надають більш детальні вказівки щодо того, як слід застосовувати закони про інформаційну безпеку.

Уряд України також працює над розробкою національної стратегії інформаційної безпеки. Ця стратегія стане дорожньою картою для зусиль уряду щодо захисту інформаційних ресурсів на найближчі роки.

Ось деякі з ключових положень цих законів і нормативних актів:

- Фізичні та юридичні особи зобов'язані захищати власні інформаційні ресурси.
- Держава створює систему державної інформаційної безпеки.
- Система державної інформаційної безпеки включає розробку та впровадження політики і стандартів інформаційної безпеки, створення системи контролю

інформаційної безпеки, підготовку кадрів з питань інформаційної безпеки, а також співробітництво з міжнародними організаціями з питань інформаційної безпеки.

- Уряд України прагне забезпечити безпеку інформаційних ресурсів і застосовує проактивний підхід до вирішення проблем кіберзлочинності та інформаційних війн.

Ці закони та нормативно-правові акти створюють всеосяжну основу для захисту інформаційних ресурсів в Україні. Однак важливо зазначити, що ці закони та нормативні акти постійно оновлюються, щоб відображати мінливий ландшафт загроз. Тому приватним особам та організаціям важливо бути в курсі останніх подій у сфері інформаційної безпеки, щоб ефективно захищати свої інформаційні ресурси.

На додаток до згаданих вище законів і нормативних актів, існує ряд інших заходів, які окремі особи та організації можуть вживати для захисту своїх інформаційних ресурсів. До них відносяться

- Використання надійних паролів і заходів безпеки для захисту своїх комп'ютерних систем і мереж.
- Навчання персоналу найкращим практикам інформаційної безпеки.
- Повідомляти про інциденти безпеки відповідним органам.
- Співпраця з правоохоронними органами в розслідуванні інцидентів безпеки.

Вживаючи цих заходів, приватні особи та організації можуть допомогти захистити свої інформаційні ресурси від несанкціонованого доступу, використання, розкриття, знищення або модифікації.

Загалом, нормативно-правова база для забезпечення захисту інформаційних ресурсів в Україні є розгалуженою і охоплює різні закони, підзаконні акти та політики. Дотримання цих законів і нормативно-правових актів має важливе значення для забезпечення захисту інформаційних ресурсів та запобігання несанкціонованому доступу до конфіденційної інформації. Постійний моніторинг і регулярне оновлення мають вирішальне значення для підтримки безпеки інформаційних ресурсів.

1.2 Сучасні методи захисту інформації

Сучасні методи захисту інформації є важливим аспектом дослідження засобів удосконалення технологій доступу до захищених інформаційних ресурсів. У зв'язку зі зростанням залежності від технологій зберігання та обробки конфіденційної інформації виникає потреба у вдосконалених методах захисту такої інформації від несанкціонованого доступу та порушень.

Одним із сучасних методів захисту інформації є шифрування. Шифрування даних - це процес перетворення даних у нечитабельний формат за допомогою ключа. Це робить дані нечитабельними для будь-кого, хто не має ключа. Шифрування даних можна використовувати для захисту конфіденційних даних, таких як фінансова інформація, медичні записи та інтелектуальна власність.

Ще одним сучасним методом захисту інформації є безпека з нульовою довірою - це модель безпеки, яка передбачає, що жодному користувачеві чи пристрою не можна довіряти за замовчуванням. Ця модель вимагає, щоб усі користувачі та пристрої були автентифіковані та авторизовані, перш ніж їм буде дозволено отримати доступ до будь-яких ресурсів. Безпека з нульовою довірою може допомогти захистити організації від різноманітних загроз, включаючи інсайдерські загрози та витоки даних.

Системи виявлення та запобігання вторгнень (IDPS) - ще один сучасний метод захисту інформації. IDPS призначені для моніторингу мережевого трафіку, виявлення та запобігання несанкціонованому доступу та атакам на інформаційні ресурси. IDPS можуть виявляти та реагувати на різні типи атак, включаючи шкідливе програмне забезпечення, атаки на відмову в обслуговуванні та спроби несанкціонованого доступу. Використовуючи складні механізми виявлення та проактивного реагування, IDPS відіграють вирішальну роль у захисті цілісності та доступності критично важливих даних і систем(рис 1.1). Їх безперервний моніторинг та аналіз загроз у режимі реального часу допомагають організаціям бути на крок попереду потенційних вторгнень, забезпечуючи надійний захист.

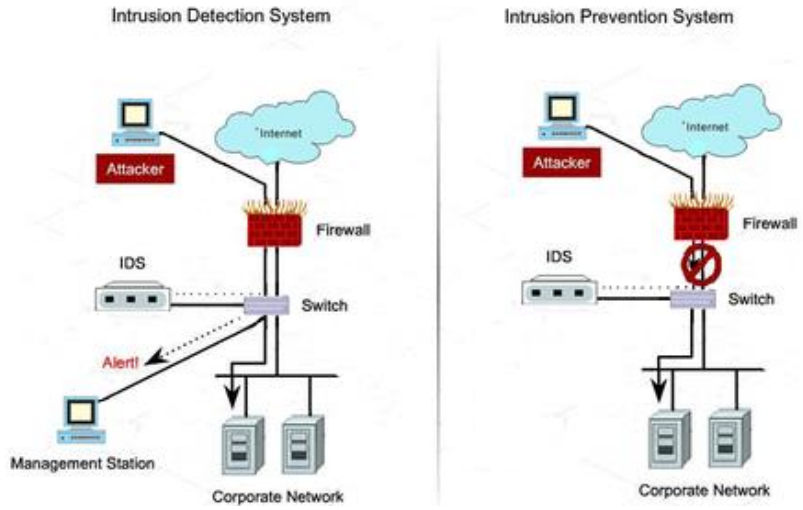

Рисунок 1.1 - Поведінка системи виявлення вторгнення (ліворуч), поведінка Системи запобігання вторгнення (праворуч).

DevSecOps - це підхід до безпеки, який інтегрує безпеку в життєвий цикл розробки та експлуатації. Цей підхід допомагає гарантувати, що безпека враховується з самого початку процесу розробки і протягом усього життя програми(рис.1.2). DevSecOps може допомогти зменшити ризик впровадження вразливостей безпеки в додатки.

Рисунок 1.2 - Стандартна схема DevSecOp

Розвідка загроз - це інформація про загрози, яка може бути використана для захисту організацій від цих загроз. Розвіддані про загрози можна збирати з різних джерел, таких як розвіддані з відкритих джерел (OSINT), соціальні мережі та канали

загроз. Розвіддані про загрози можна використовувати для виявлення та оцінки загроз, розробки стратегій пом'якшення наслідків та реагування на інциденти.

Підвищення обізнаності з питань безпеки - це процес навчання співробітників про ризики безпеки та способи захисту від них. Тренінги з безпеки можуть допомогти працівникам виявляти та уникати фішингових атак, створювати надійні паролі та ефективно використовувати програмне забезпечення для забезпечення безпеки.

Ще одним сучасним методом захисту інформації є фізичний захист. Фізичний захист передбачає використання фізичних бар'єрів, таких як замки, паркани та камери спостереження, для захисту інформаційних ресурсів. Фізична безпека має важливе значення для забезпечення того, щоб несанкціоновані особи не мали фізичного доступу до конфіденційної інформації.

На додаток до цих методів, існує ряд інших речей, які фахівці з ІБ можуть зробити для захисту інформаційних ресурсів в Україні. До них відносяться:

- Проведення регулярних оцінок безпеки для виявлення та усунення вразливостей.
- Впровадження політик і процедур безпеки для інформування співробітників про те, як захищати інформаційні ресурси.
- Моніторинг журналів безпеки для виявлення підозрілої активності.
- Швидке та ефективне реагування на інциденти безпеки.

Загалом, сучасні методи захисту інформації мають вирішальне значення для забезпечення захисту інформаційних ресурсів. Використання сучасних стандартів шифрування, механізмів контролю доступу, систем виявлення та запобігання вторгненням, а також заходів фізичної безпеки дозволяє значно знизити ризик несанкціонованого доступу та порушень. Дослідження засобів удосконалення технологій доступу до захищених інформаційних ресурсів повинні враховувати використання сучасних методів захисту інформації для підвищення безпеки інформаційних ресурсів.

1.3 Аналіз існуючих засобів вдосконалення технологій доступу до захищених інформаційних ресурсів

В останні роки зростає потреба в ефективних і надійних механізмах контролю доступу для забезпечення безпеки інформаційних ресурсів, що захищаються. Розвиток нових технологій та зростання залежності від цифрових інформаційних систем зумовили необхідність розробки та впровадження більш досконалих методів контролю доступу. Інформаційна безпека (ІБ) є критично важливим аспектом діяльності будь-якої організації. У сучасну цифрову епоху організації все більше покладаються на системи інформаційних технологій (ІТ) для зберігання, обробки та передачі конфіденційних даних. Як наслідок, організації постійно перебувають під загрозою різноманітних кібератак.

Одним з найважливіших аспектів ІБ є захист доступу до захищених інформаційних ресурсів. Захищені інформаційні ресурси - це ті, які вважаються конфіденційними, чутливими або критично важливими для діяльності організації. Прикладами захищених інформаційних ресурсів є фінансові дані, дані клієнтів та інтелектуальна власність.

Тому в цьому підрозділі проаналізовано існуючі засоби вдосконалення технологій доступу до захищених інформаційних ресурсів.

Одним з найпоширеніших методів контролю доступу є використання паролів. Цей метод широко використовується в різних системах і є відносно простим у реалізації. Однак безпека цього методу обмежена, оскільки паролі можуть бути легко скомпрометовані або викрадені, а користувачі можуть їх забути або втратити. Для подолання цих обмежень були розроблені методи двофакторної автентифікації, такі як використання токенів або біометрична автентифікація. Ці методи забезпечують вищий рівень безпеки, але є дорожчими у впровадженні та можуть мати технічні обмеження.

Таблиця 1.1

Переваги та недоліки впровадження паролів

Переваги	**Недоліки**
• Паролі широко відомі і зрозумілі користувачам, що робить їх простими у впровадженні та використанні. Більшість людей знайомі з концепцією паролів і вважають їх зручними для запам'ятовування та введення.	• Багато користувачів схильні створювати слабкі паролі, які легко вгадуються або вразливі до атак грубої сили (використання простих і легко вгадуваних паролів, повторне використання паролів у кількох облікових записах або використання особистої інформації, яку можна легко отримати)
• Впровадження систем контролю доступу на основі паролів є економічно вигідним. Паролями можна керувати і застосовувати їх за допомогою програмного забезпечення, і немає потреби в додатковому обладнанні або складній інфраструктурі.	• Паролі можуть бути скомпрометовані різними способами, наприклад, фішинговими атаками, кейлоггерами або витоком даних. Якщо зловмисник отримує доступ до бази даних паролів, він потенційно може розшифрувати або зламати паролі, поставивши під загрозу безпеку системи.
• Паролі пропонують гнучкість з точки зору кастомізації. Користувачі можуть обирати власні паролі на основі своїх уподобань і можуть легко змінювати їх у разі потреби.	• Керування паролями може бути проблемою як для користувачів, так і для системних адміністраторів. Користувачам може бути важко запам'ятовувати багато складних

продовження табл. 1.1

Ця гнучкість дозволяє людям мати певний рівень контролю над власною безпекою.	паролів, що призводить до небезпечних практик, таких як їх запис або ненадійне зберігання. Адміністратори можуть зіткнутися з труднощами у впровадженні надійних політик паролів та забезпеченні регулярного оновлення паролів.

Іншим поширеним методом є шифрування даних - це процес перетворення даних у нечитабельний формат за допомогою ключа. Це робить дані нечитабельними для будь-кого, хто не має ключа. Шифрування даних можна використовувати для захисту конфіденційних даних, таких як фінансова інформація, медичні записи та інтелектуальна власність.

Таблиця 1.2

Переваги та недоліки впровадження шифрування

Переваги	**Недоліки**
• Шифрування даних може дуже ускладнити доступ неавторизованих осіб до конфіденційних даних, навіть якщо вони вкрали ключ шифрування.	• Шифрування даних може бути складним і трудомістким процесом.
• Шифрування даних також можна використовувати для захисту даних під час передачі, наприклад, мережею.	• Шифрування даних також може сповільнювати продуктивність програм і систем.
• Шифрування даних можна використовувати для захисту даних у стані спокою, наприклад, коли вони зберігаються на жорсткому диску або	• Шифруванням даних може бути складно керувати, особливо якщо є велика кількість користувачів або пристроїв, яким потрібен доступ до

продовження табл. 1.2

в хмарному сховищі.	зашифрованих даних.

Безпека з нульовою довірою - це модель безпеки, яка передбачає, що жодному користувачеві чи пристрою не можна довіряти за замовчуванням. Ця модель вимагає, щоб усі користувачі та пристрої були автентифіковані та авторизовані, перш ніж їм буде дозволено отримати доступ до будь-яких ресурсів. Безпека з нульовою довірою може допомогти захистити організації від різноманітних загроз, включаючи інсайдерські загрози та витоки даних.

Таблиця 1.3

Переваги та недоліки впровадження безпеки з нульовою довірою

Переваги	**Недоліки**
• Безпека з нульовою довірою може допомогти захистити організації від різноманітних загроз, включаючи внутрішні загрози та витоки даних.	• Впровадження системи безпеки з нульовою довірою може бути складним і дорогим.
• Безпека з нульовою довірою також може допомогти поліпшити загальний стан безпеки організації.	• Безпекою з нульовою довірою також може бути складно керувати, особливо якщо є велика кількість користувачів або пристроїв, які потребують автентифікації та авторизації.
• Безпека з нульовою довірою може бути реалізована різними способами, що дозволяє адаптувати її до конкретних потреб організації.	• Прийняття та впровадження системи безпеки з нульовою довірою може призвести до додаткових рівнів складності та операційних міркувань, що потенційно може вплинути на продуктивність додатків та систем.

DevSecOps - це підхід до безпеки, який інтегрує безпеку в життєвий цикл розробки та експлуатації. Цей підхід допомагає гарантувати, що безпека враховується з самого початку процесу розробки і протягом усього життя програми. DevSecOps може допомогти зменшити ризик впровадження вразливостей безпеки в додатки.

Таблиця 1.4

Переваги та недоліки впровадження DevSecOps

Переваги	**Недоліки**
• DevSecOps може допомогти зменшити ризик впровадження вразливостей безпеки в додатки.	• Впровадження DevSecOps може бути складним і дорогим.
• DevSecOps також може допомогти поліпшити загальний стан безпеки організації.	• DevSecOps також може бути складним в управлінні, особливо якщо в ньому задіяна велика кількість розробників і операційних команд.
• DevSecOps можна впроваджувати різними способами, що дозволяє адаптувати його до конкретних потреб організації.	• DevSecOps також може впливати на життєвий цикл розробки та експлуатації, що може призвести до затримок у постачанні нових функцій і додатків.

Розвідка загроз - це інформація про загрози, яка може бути використана для захисту організацій від цих загроз. Розвіддані про загрози можна збирати з різних джерел, таких як розвіддані з відкритих джерел (OSINT), соціальні мережі та канали загроз. Розвіддані про загрози можна використовувати для виявлення та оцінки загроз, розробки стратегій пом'якшення наслідків та реагування на інциденти. Використовуючи дані про загрози, організації можуть проактивно виявляти

потенційні загрози, оцінювати їхню серйозність та впроваджувати ефективні заходи для захисту своїх систем і конфіденційних даних.

Таблиця 1.5

Переваги та недоліки OSINT

Переваги	**Недоліки**
• Розвідка загроз може допомогти організаціям виявляти та оцінювати загрози.	• Придбання та підтримка системи розвідки загроз може бути дорогим задоволенням.
• Розвідка загроз також може допомогти організаціям розробити стратегії пом'якшення наслідків цих загроз.	• Розвіддані про загрози також може бути важко інтерпретувати та ефективно використовувати.
• Виявлення загроз також може допомогти організаціям ефективніше реагувати на інциденти.	• Дані про загрози можуть бути застарілими, що може призвести до того, що організації прийматимуть рішення на основі неточної інформації.

Підвищення обізнаності з питань безпеки - це процес навчання співробітників про ризики безпеки і про те, як захистити себе від цих ризиків. Тренінги з безпеки можуть допомогти працівникам виявляти та уникати фішингових атак, створювати надійні паролі та ефективно використовувати захисне програмне забезпечення.

Таблиця 1.6

Переваги та недоліки підвищення обізнаності з питань безпеки

Переваги	**Недоліки**
• Тренінги з безпеки допоможуть працівникам виявляти фішингові	• Розробка та проведення тренінгів з підвищення обізнаності про безпеку

продовження табл. 1.6

атаки та уникати їх.	може бути дорогим задоволенням.
• Тренінги з безпеки можуть допомогти працівникам створювати надійні паролі та ефективно використовувати захисне програмне забезпечення.	• Також може бути важко виміряти ефективність тренінгу з безпеки.
• Тренінги з безпеки можуть допомогти створити культуру безпеки в організації.	• Тренінг з підвищення обізнаності з питань безпеки також може бути нудним і неефективним, якщо він погано розроблений і проведений.

На додаток до технологій і практик, згаданих вище, існує ряд інших факторів, які можуть сприяти поліпшенню контролю доступу до захищених інформаційних ресурсів. До них відносяться:

Сильне лідерство має важливе значення для будь-якої організації, яка хоче покращити свій стан ІС. Керівники повинні бути віддані безпеці і надавати ресурси та підтримку, необхідні для впровадження ефективних заходів безпеки.

Культура безпеки - це культура, в якій співробітники усвідомлюють важливість безпеки і прагнуть захищати інформаційні ресурси. Організації можуть створити культуру безпеки, проводячи тренінги з підвищення обізнаності щодо безпеки, винагороджуючи працівників за належні практики безпеки та вживаючи дисциплінарних заходів проти працівників, які порушують політику безпеки.

Постійне вдосконалення. Безпека - це сфера, що постійно розвивається. Організації повинні постійно контролювати свій стан безпеки і вносити зміни, якщо це необхідно, щоб залишатися на крок попереду ландшафту загроз. Це можна зробити, проводячи регулярні оцінки безпеки, впроваджуючи нові технології безпеки та навчаючи співробітників новим загрозам і найкращим практикам.

Існують різні засоби вдосконалення технологій доступу до захищених інформаційних ресурсів, кожен з яких має свої переваги та обмеження. Вибір найбільш підходящого методу контролю доступу буде залежати від конкретних вимог системи і необхідного рівня безпеки. Важливо ретельно оцінити наявні варіанти і вибрати найбільш підходящий метод для забезпечення ефективного захисту інформаційних ресурсів.

Висновки за розділом 1

Виходячи з представлених підрозділів, можна зробити висновок, що основна увага в аналізі зосереджена на забезпеченні захисту інформаційних ресурсів.

У першому підрозділі (1.1) обговорюється нормативно-правова база, яка лежить в основі захисту інформаційних ресурсів. Це стосується законів, політик та інструкцій, які визначають, як організації поводяться з конфіденційною інформацією та захищають її. Розуміння нормативно-правової бази має вирішальне значення для забезпечення відповідності та уникнення потенційних юридичних і фінансових ризиків.

Другий підрозділ (1.2) присвячений сучасним методам захисту інформації. Включає такі технології, як брандмауери, антивірусне програмне забезпечення, шифрування та системи виявлення вторгнень. Ознайомлення з новітніми інструментами та методами безпеки є важливим для того, щоб йти в ногу з еволюцією загроз і гарантувати, що інформаційні ресурси залишатимуться захищеними.

У третьому підрозділі (1.3) проводиться аналіз існуючих інструментів і технологій для покращення доступу до захищених інформаційних ресурсів. Це включає оцінку ефективності існуючих механізмів контролю доступу, таких як паролі, біометричні дані та двофакторна автентифікація, а також визначення сфер, в яких можна досягти поліпшень.

Загалом, аналіз спрямований на забезпечення належного захисту інформаційних ресурсів шляхом поєднання дотримання законодавчих і нормативних

вимог, використання найсучасніших засобів і методів безпеки, а також постійного оцінювання та вдосконалення механізмів контролю доступу.

РОЗДІЛ 2
ДОСЛІДЖЕННЯ МЕТОДІВ І ЗАСОБІВ ЗАХИСТУ ІНФОРМАЦІЙНИХ РЕСУРСІВ

2.1Дослідження захисних механізмів для забезпечення цілісності, конфіденційності та доступності інформаційних ресурсів

Захист цілісності, конфіденційності та доступності інформаційних ресурсів має вирішальне значення для будь-якої організації, яка покладається на ці ресурси у своїй роботі. Цілісність інформаційних ресурсів означає точність і повноту даних, конфіденційність - захист конфіденційної інформації від несанкціонованого доступу, а доступність гарантує, що дані будуть доступні уповноваженому персоналу, коли це необхідно.

Для досягнення цих цілей використовуються різні захисні механізми, включаючи брандмауери, контроль доступу, шифрування, системи виявлення та запобігання вторгненням, а також резервне копіювання даних. Брандмауери - це програмні або апаратні системи, які відстежують і контролюють потік мережевого трафіку, запобігаючи несанкціонованому доступу і обмежуючи вплив зовнішніх загроз.

Засоби контролю доступу, з іншого боку, регулюють доступ користувачів до інформаційних ресурсів, гарантуючи, що тільки уповноважений персонал може отримати доступ до конфіденційної інформації. Ці засоби контролю можуть приймати різні форми, включаючи паролі, біометричну автентифікацію та багатофакторну автентифікацію.

Шифрування даних - це процес перетворення даних у нечитабельний формат за допомогою ключа. Це робить дані нечитабельними для будь-кого, хто не має ключа. Шифрування даних можна використовувати для захисту конфіденційних даних, таких як фінансова інформація, медичні записи та інтелектуальна власність.

Наприклад, компанія може зашифрувати дані своїх клієнтів, щоб запобігти їх викраденню хакерами. Для цього компанії спочатку потрібно згенерувати ключ. Це може бути довгий рядок випадкових символів або фраза. Після того, як ключ буде згенеровано, його потрібно буде надійно зберігати. Потім компанія може використовувати ключ для шифрування даних своїх клієнтів. Коли клієнт здійснює покупку, компанія зашифровує інформацію про кредитну картку клієнта за допомогою ключа. Зашифровані дані зберігаються в базі даних компанії. Якщо хакер отримає доступ до бази даних компанії, він не зможе прочитати зашифровані дані без ключа.

Контроль доступу - це процес обмеження доступу до інформаційних ресурсів. Це можна зробити за допомогою паролів, сертифікатів безпеки або інших методів автентифікації. Контроль доступу допомагає гарантувати, що тільки авторизовані користувачі можуть отримати доступ до конфіденційних даних.

Наприклад, компанія може використовувати паролі для обмеження доступу до системи нарахування заробітної плати своїм працівникам. Пароль отримають лише ті працівники, які мають право доступу до системи нарахування заробітної плати. Якщо неавторизований користувач спробує отримати доступ до системи нарахування заробітної плати, йому буде відмовлено в доступі.

Фізичний захист не менш важливий. Заходи фізичної безпеки захищають інформаційні ресурси від фізичних загроз, таких як крадіжка або пошкодження. Це можуть бути такі заходи, як контроль доступу, камери спостереження та охорона.

Наприклад, компанія може використовувати камери спостереження для моніторингу свого центру обробки даних. Це допоможе відлякати злодіїв, а також ідентифікувати будь-кого, хто намагався отримати доступ до центру обробки даних без дозволу.

Заходи ж мережевої безпеки захищають інформаційні ресурси від загроз, що надходять через мережу, таких як шкідливе програмне забезпечення та хакерські атаки. Це можуть бути брандмауери, системи виявлення вторгнень та антивірусне програмне забезпечення.

Наприклад, компанія може використовувати брандмауер для захисту своєї мережі від несанкціонованого доступу. Брандмауер блокує будь-які спроби доступу до мережі компанії із зовнішніх джерел.

Резервне копіювання даних - це процес створення копій важливих даних. Це допомагає захистити дані від втрати або знищення. Резервне копіювання даних слід робити регулярно і зберігати в безпечному місці.

Наприклад, компанія може створювати резервні копії своїх даних у хмарі. Це забезпечить постійний доступ до даних компанії, навіть якщо її центр обробки даних буде зруйновано.

Тренінги з безпеки допомагають працівникам зрозуміти важливість безпеки та способи захисту інформаційних ресурсів. Таке навчання має охоплювати такі теми, як фішинг, шкідливе програмне забезпечення та безпека паролів.

Наприклад, компанія може регулярно проводити для своїх працівників тренінги з підвищення обізнаності про безпеку. Такі тренінги допоможуть працівникам виявляти та уникати фішингових атак, атак шкідливих програм та інших загроз безпеці.

На додаток до захисних механізмів, згаданих вище, існує низка інших факторів, які можуть вплинути на безпеку інформаційних ресурсів. До них відносяться

Сильне лідерство має важливе значення для будь-якої організації, яка хоче покращити свій стан безпеки. Керівники повинні бути віддані безпеці і надавати ресурси та підтримку, необхідні для впровадження ефективних заходів безпеки.

Культура безпеки - це культура, в якій співробітники усвідомлюють важливість безпеки і прагнуть захищати інформаційні ресурси. Організації можуть створити культуру безпеки, проводячи тренінги з підвищення обізнаності щодо безпеки, винагороджуючи працівників за належні практики безпеки та вживаючи дисциплінарних заходів проти працівників, які порушують політику безпеки.

Постійне вдосконалення дуже важливе, адже, безпека - це сфера, що постійно розвивається. Організації повинні постійно контролювати свій стан безпеки і вносити зміни, якщо це необхідно, щоб залишатися на крок попереду ландшафту загроз. Це

можна зробити, проводячи регулярні оцінки безпеки, впроваджуючи нові технології безпеки та навчаючи співробітників новим загрозам і найкращим практикам.

Тож, захисні механізми, такі як брандмауери, контроль доступу, шифрування, IDPS та резервне копіювання даних, є критично важливими для забезпечення цілісності, конфіденційності та доступності інформаційних ресурсів. Ці механізми працюють разом, щоб захистити конфіденційну інформацію та забезпечити доступ до неї лише уповноваженому персоналу. Організації повинні постійно оцінювати та вдосконалювати свої захисні заходи, щоб не відставати від нових кіберзагроз та ефективно захищати свої інформаційні ресурси.

2.2 Системне дослідження засобів і методів захисту інформації

Для захисту інформаційних ресурсів існує широкий спектр засобів і методів інформаційної безпеки. Систематичний аналіз цих засобів і методів необхідний для визначення найбільш прийнятних засобів для вдосконалення технологій доступу до захищених інформаційних ресурсів.

Одним з найпоширеніших засобів захисту інформації є брандмауери, які виступають бар'єром між довіреною внутрішньою мережею і недовіреними зовнішніми мережами. Брандмауери контролюють мережевий трафік і блокують спроби несанкціонованого доступу. Системи виявлення вторгнень (IDS) - ще один популярний інструмент інформаційної безпеки. IDS відстежують мережевий трафік і шукають ознаки зловмисної активності, такі як спроби використання вразливостей у програмному забезпеченні або спроби несанкціонованого доступу.

Шифрування - ще один важливий метод інформаційної безпеки, який можна використовувати для захисту даних у стані спокою та під час передачі. Шифрування перетворює дані в закодовану форму, яку можуть прочитати лише уповноважені особи з правильним ключем розшифрування. Двофакторна автентифікація - ще один широко використовуваний метод, який вимагає від користувачів надання двох різних типів автентифікації, таких як пароль і відбитки пальців, для доступу до конфіденційних інформаційних ресурсів.

Фізичні заходи безпеки, такі як біометричний контроль доступу, камери спостереження та охорона, також є важливими для захисту інформаційних ресурсів. Контроль доступу може обмежити фізичний доступ до критично важливих інформаційних ресурсів, тоді як камери спостереження можуть контролювати фізичний доступ до чутливих зон. Охоронці можуть забезпечити додатковий рівень фізичного захисту важливих інформаційних ресурсів.

Ще одним важливим інструментом інформаційної безпеки є програмне забезпечення для сканування вразливостей, яке можна використовувати для виявлення вразливостей у програмному забезпеченні та мережевих конфігураціях. Регулярне сканування вразливостей може допомогти виявити слабкі місця в системі безпеки, якими можуть скористатися зловмисники. Тестування на проникнення - ще один важливий метод інформаційної безпеки, який передбачає імітацію атаки на інформаційний ресурс для виявлення вразливостей і слабких місць.

Існуючі технології доступу до захищених інформаційних ресурсів мають ряд обмежень, які необхідно усунути. Наприклад, ось деякі з найпоширеніших обмежень:

- Системи автентифікації на основі паролів вразливі до хакерських та фішингових атак, які можуть поставити під загрозу безпеку інформаційних ресурсів.
- Системи біометричної автентифікації можуть бути ненадійними в деяких випадках, наприклад, коли користувач має фізичні вади або коли біометричні дані пошкоджені.
- Системи двофакторної автентифікації вимагають додаткового обладнання або програмного забезпечення, що може бути дорогим і складним у впровадженні.
- Системи автентифікації на основі смарт-карток вразливі до фізичних атак, таких як крадіжка або втрата смарт-картки.
- Системи автентифікації на основі токенів можуть вимагати частого перевипуску токенів, що може бути незручним для користувачів.

Таким чином, систематичний аналіз засобів і методів інформаційної безпеки необхідний для визначення найбільш прийнятних засобів для вдосконалення технологій доступу до захищених інформаційних ресурсів. Вищезгадані інструменти

та методи інформаційної безпеки є лише кількома прикладами широкого спектру доступних інструментів та методів. Організації повинні ретельно оцінити свої потреби в інформаційній безпеці і вибрати найбільш підходящі інструменти і методи для захисту своїх цінних інформаційних ресурсів.

2.3 Вибір технологій доступу до захищених інформаційних ресурсів

Вибір технологій доступу до захищених інформаційних ресурсів є критично важливим кроком у забезпеченні цілісності, конфіденційності та доступності цих ресурсів. Він вимагає всебічного аналізу наявних технологій та ретельної оцінки їх придатності для конкретного інформаційного ресурсу та вимог організації.

Процес вибору повинен починатися з чіткого визначення вимог до безпеки інформаційного ресурсу та оцінки потенційних загроз і вразливостей. На основі цього слід визначити відповідні заходи безпеки, включаючи контроль доступу, автентифікацію, шифрування та моніторинг.

Потім слід оцінити наявні технології на предмет їхньої здатності відповідати цим вимогам безпеки. Ця оцінка повинна враховувати такі фактори, як ефективність технології, простота використання, масштабованість та економічна ефективність. Вибрані технології також повинні бути сумісними з існуючою інфраструктурою та системами організації.

Процес вибору повинен включати ретельний аналіз документації постачальника, включаючи технічні специфікації, посібники користувача та сертифікати безпеки. Також рекомендується провести пілотне тестування обраних технологій перед повним розгортанням, щоб переконатися, що вони відповідають бажаним вимогам безпеки та безперешкодно інтегруються з існуючими системами організації.

Вибір технологій - це не одноразова подія, а безперервний процес, який вимагає регулярного перегляду та оновлення. Оскільки з'являються нові загрози і розвиваються технології, організації повинні постійно оцінювати свої інструменти і

методи інформаційної безпеки і вносити необхідні корективи для забезпечення захисту своїх інформаційних ресурсів.

Однією з найпростіших технологій, яка може бути використана для покращення доступу до захищених інформаційних ресурсів, є двофакторна аутентифікація.

Двофакторна автентифікація (2FA) - це процес безпеки, в якому користувачеві надається доступ до веб-сайту або додатку тільки після успішного надання двох різних доказів механізму автентифікації - як правило, того, що він знає (наприклад, пароль), і того, що він має (наприклад, токен безпеки).

2FA забезпечує додатковий рівень безпеки, вимагаючи від користувачів надання двох різних факторів для підтвердження своєї особи. Це значно знижує ризик несанкціонованого доступу, оскільки зловмиснику потрібно мати обидва фактори (наприклад, пароль і фізичний пристрій), щоб отримати доступ.

Також, 2FA знижує ризик атак, пов'язаних з паролями, таких як атака грубої сили або підбір пароля. Навіть якщо зловмиснику вдасться отримати або вгадати пароль користувача, йому все одно знадобиться другий фактор для отримання доступу.

Впровадження 2FA розширює сферу атаки, що ускладнює зловмисникам завдання скомпрометувати облікові записи користувачів. Це додає додатковий рівень складності для зловмисників, часто змушуючи їх відмовлятися від своїх спроб і шукати більш легкі цілі.

У той час як традиційні методи автентифікації покладаються виключно на паролі, 2FA забезпечує більш зручний користувацький досвід. Він пропонує такі варіанти, як використання біометрії (відбитків пальців або розпізнавання обличчя) або push-повідомлень, які часто є швидшими та зручнішими для користувачів.

Існує два основних типи 2FA:

- Одноразові паролі на основі часу (TOTP) - найпоширеніший тип 2FA. На телефон встановлюється додаток-генератор TOTP, який генерує новий код кожні 30 секунд. Коли здійснюється вхід на веб-сайт або в додаток, вводиться код з програми на додаток до пароля.

- Поза діапазоном (OOB) - надсилає код на телефон за допомогою SMS, електронної пошти або голосового дзвінка. Коли користувач входить на веб-сайт або додаток, він вводить код з телефону на додаток до свого пароля.

2FA - це дуже ефективний спосіб захисту облікових записів в Інтернеті. Його легко налаштувати і використовувати, і він може суттєво вплинути на безпеку акаунтів.

Таблиця 1.7

Переваги та недоліки TOTP

Переваги	**Недоліки**
• Надійніша автентифікація з додатковим рівнем безпеки.	• Залежність від точної синхронізації часу.
• Функціональність в автономному режимі.	• Вразливість до фішингових атак
• Простота впровадження.	• Обмежена масштабованість.
• Підтримка декількох пристроїв.	

Таблиця 1.8

Переваги та недоліки OOB

Переваги	**Недоліки**
• Посилення безпеки завдяки окремому каналу зв'язку.	• Залежність від додаткових каналів
• Гнучкість в каналах зв'язку	• Міркування щодо користувацького досвіду
• Зменшення залежності від паролів.	• Безпека каналу зв'язку.

Висновки за розділом 2

Виходячи з наведених підпунктів, можна зробити наступні висновки:

У підрозділі 2.1йдеться про аналіз захисних механізмів, які існують для забезпечення безпеки інформаційних ресурсів. Такі механізми можуть включати брандмауери, системи виявлення вторгнень та технології шифрування. Висновок, який можна зробити, полягає в тому, що важливо мати захисні механізми для захисту інформаційних ресурсів від несанкціонованого доступу, модифікації та знищення.

Підрозділ 2.2 фокусується на оцінці інструментів та методів інформаційної безпеки, що використовуються для захисту інформаційних ресурсів. Аналіз може включати оцінку ефективності інструментів та методів, виявлення будь-яких слабких місць та вразливостей, а також пропозиції щодо їх вдосконалення. Висновок, який можна зробити, полягає в тому, що дуже важливо періодично оцінювати та вдосконалювати інструменти та методи інформаційної безпеки, щоб забезпечити їхню ефективність проти нових загроз.

Підрозділ 2.3 стосується вибору відповідних технологій для доступу до захищених інформаційних ресурсів. Вибір може залежати від необхідного рівня безпеки, типу інформації, до якої надається доступ, та передбачуваних користувачів. Висновок, який можна зробити, полягає в тому, що вибір технології доступу до захищених інформаційних ресурсів повинен ґрунтуватися на ретельному аналізі вимог безпеки та потреб користувачів.

На основі аналізу трьох підрозділів (2.1, 2.2, 2.3) можна зробити висновок, що захист інформаційних ресурсів є складним і багатогранним процесом, який вимагає комплексного підходу. Він передбачає аналіз захисних механізмів для забезпечення цілісності, конфіденційності та доступності інформаційних ресурсів, а також вибір відповідних технологій доступу до захищеної інформації.

Для ефективного захисту інформаційних ресурсів необхідно мати потужну нормативно-правову базу, а також глибоке розуміння сучасних методів захисту та доступних інструментів. Використання передових технологій, таких як методи шифрування та аутентифікації, може значно підвищити безпеку інформаційних

ресурсів, але також важливо регулярно оцінювати і вдосконалювати ці заходи, щоб йти в ногу з новими загрозами.

Загалом, добре розроблена та впроваджена стратегія інформаційної безпеки має важливе значення для захисту конфіденційної інформації та збереження довіри клієнтів і зацікавлених сторін.

РОЗДІЛ 3
ВДОСКОНАЛЕННЯ ТЕХНОЛОГІЇ ДОСТУПУ ДО ЗАХИЩЕНИХ ІНФОРМАЦІЙНИХ РЕСУРСІВ

3.1 Опис запропонованого рішення

Для вдосконалення я обрала технологію двофакторної аутентифікації. Двофакторна аутентифікація - це процес безпеки, в якому користувачеві надається доступ до системи тільки після успішного надання двох різних доказів механізму автентифікації - як правило, того, що він знає (наприклад, пароль), і того, що він має (наприклад, токен безпеки або додаток-автентифікатор).

Двофакторна аутентифікація є важливим заходом безпеки, оскільки вона значно ускладнює зловмисникам доступ до системи, навіть якщо вони вкрали пароль користувача. Це пов'язано з тим, що навіть якщо зловмисник має пароль користувача, йому також потрібно мати фізичний доступ до токену безпеки користувача або програми-автентифікатора, щоб отримати доступ до системи.

Для початку необхідно визначити потреби системи. Сильні та слабкі сторони системи, цілі системи та потреби користувачів системи.

До сильних сторін системи можна віднести той факт, що двофакторна аутентифікація є дуже ефективним заходом безпеки, який може допомогти захистити користувачів від кібератак. До слабких сторін системи можна віднести те, що її налаштування та використання може бути дещо складним, а деяким користувачам може бути некомфортно користуватися нею, оскільки вони не розуміють, як вона працює.

Впроваджуючи двофакторну автентифікацію, ми задовольняємо потребу в посиленні заходів безпеки, а також враховуємо аспекти зручності використання та взаємодії з користувачем. Вона пропонує додатковий рівень захисту від спроб несанкціонованого доступу, зменшуючи ризики, пов'язані зі зламаними паролями, і підвищуючи загальну безпеку системи.

3.2 Реалізація запропонованого рішення

Для початку важливо проаналізувати та реалізувати базові програми, які демонструють функціональність двофакторної автентифікації (2FA) з використанням методів Time-Based One-Time Password (TOTP) та Out-of-Band (OOB).

Почнемо з розгляду коду програми на мові Python, яка реалізує TOTP (Додаток А). Вивчивши цей код, ми зможемо краще зрозуміти, як працює TOTP і як інтегрувати його в наші системи.

Реалізація TOTP передбачає генерацію одноразового пароля на основі спільного секретного ключа та поточного часу. Алгоритм TOTP використовує хеш-функцію, зазвичай HMAC-SHA1, для генерації 6-значного коду, який змінюється кожні 30 секунд. Потім цей код порівнюється з даними, введеними користувачем, щоб підтвердити його особу.

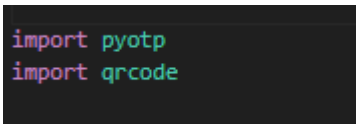

Рисунок 3.1 - Імпортування необхідних бібліотек

По-перше, імпортуємо дві необхідні для роботи програми бібліотеки(рис. 3.1). Бібліотека “pyotp” використовується для генерації та перевірки TOTP-кодів, а бібліотека “qrcode” - для створення qr коду.

Згенеруємо секретний код(рис. 3.2).

Рисунок 3.2 - Генерація секретного коду

В змінну secret_code зберігаємо випадково згенерований секретний код. За допомогою функції random_base32(), що надається бібліотекою PyOTP, генеруємо випадковий секретний код у форматі base32, тобто, у представленні двійкових даних за допомогою набору з 32 символів(A-Z, 2-7). Далі генеруємо QR-код (рис.3.3).

```
qr = qrcode.QRCode(version=1, box_size=10, border=5)
qr.add_data(pyotp.totp.TOTP(secret_code).provisioning_uri("MyApp:alikhvar2001@gmail.com",
issuer_name="alikhvar2001@gmail.com"))
qr.make(fit=True)
```

Рисунок 3.3 - Створення QR коду, який користувач зможе відсканувати через додаток

На рис. 3.3 ми бачимо створення QR коду за допомогою бібліотеки “qrcode”, подальше додання інформації до коду за допомогою URI, створеного бібліотекою “pyotp” та генерацію QR коду у вигляді зображення, що виводиться на єкран користувачу.

Рядок *qr = qrcode.QRCode(version=1, box_size=10, border=5)* ініціалізує новий обьект QRCode за допомогою бібліотеки QRcode, *version=1* - розмір QR коду, більші значення означають коди більшого розміру. *box_size=10* - визначає розмір кожного пікселю(або рамки) в QR-коді, *border=5* - вказує на кількість рамок, які будуть навколо QR-коду.

qr.add_data(pyotp.totp.TOTP(secret_code).provisioning_uri("MyApp:alikhvar2001 @gmail.com", issuer_name="alikhvar2001@gmail.com") - додає дані до QR-коду. За допомогою бібліотеки PyOTP генеруємо provisioning_uri. Ця функція викликається з двома аргументами - ідентифікатором облікового запису користувача, в даному випадку *"MyApp:alikhvar2001@gmail.com"* та другим аргументом є ім’я видавця, що є необов’язковим параметром. Цей параметр вказує назву сервісу або програми, що пов’язана з цим QR кодом. Згенерований *provisioning_uri* містить інформацію, що необхідна для надання користувачеві токену TOTP, включаючи ідентифікатор, секретний код та інш.

qr.make(fit=True) генерує зображення QR коду на основі отриманих даних. Параметр fit встановлено в True, це означає, що згенерований Qr код буде автоматично підлаштовувати свій розмір під дані. Отримане зображення зберігається в об’єкт qr і знадобиться нам далі.

```
img = qr.make_image(fill_color="black", back_color="white")
img.show()
```

Рисунок 3.4 - Генерація та вивід зображення QR коду

Виконання команди *img = qr.make_image(fill_color="black", back_color="white")* генерує зображення QR коду, в параметрі *fill_color* задаємо колір модулів, в нашому випадку - чорний. Отримане зображення зберігається у форматі JPEG, та виводиться на екран командою show() за допомогою програми перегляду зображень за замовчуванням(рис.3.4.)

Якщо ми запустимо програму, то одразу ж побачимо автоматично сгенерований QR код (рис.3.5).

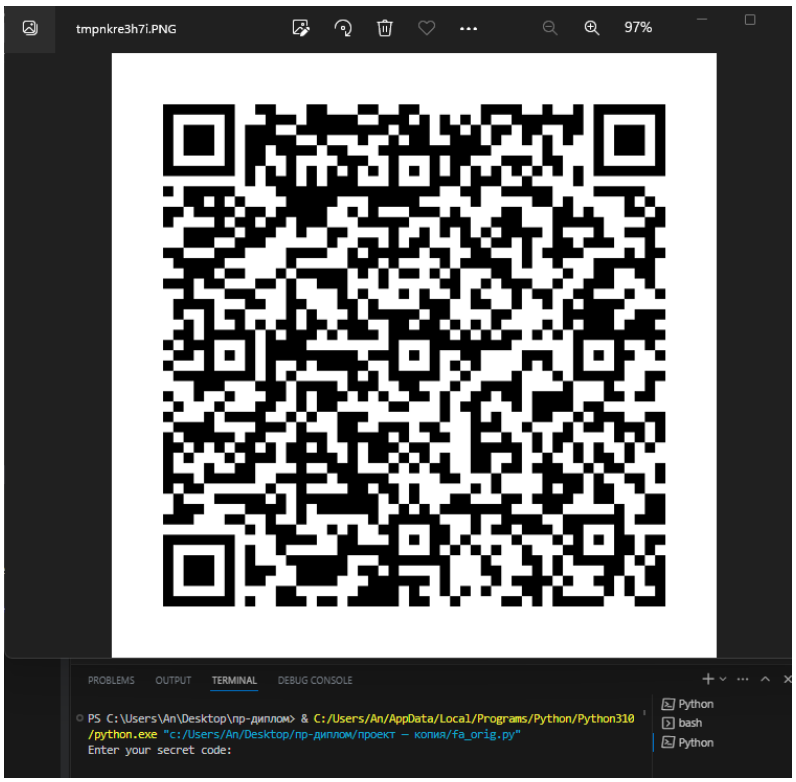

Рисунок 3.5 - Результат генерації QR коду для alikhvar2001@gmail.com

Отримавши QR-код, ми можемо відсканувати його за допомогою камери (рис. 3.6) або спеціального додатку, наприклад, Google Authenticator (рис. 3.7), що дозволить нам швидко і зручно отримати необхідну інформацію для автентифікації.

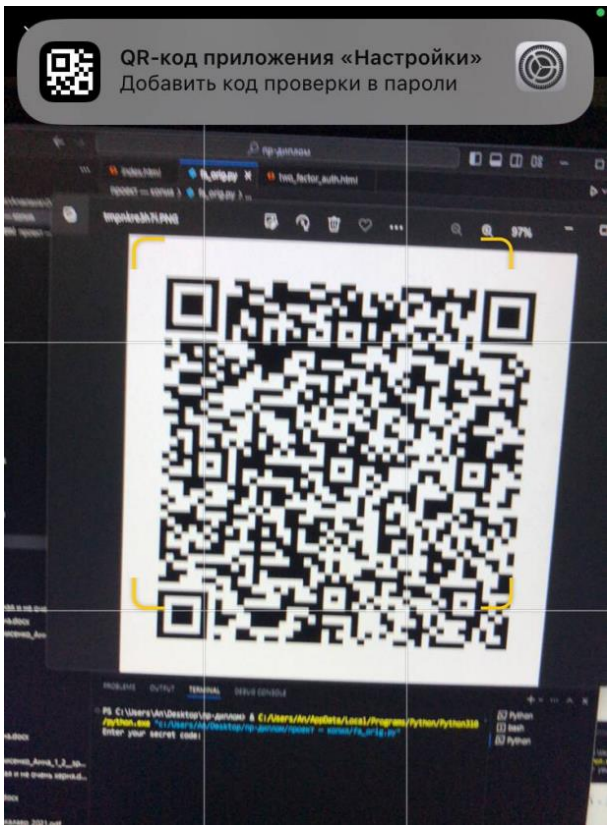

Рисунок 3.6 - Сканування QR коду за допомогою камери

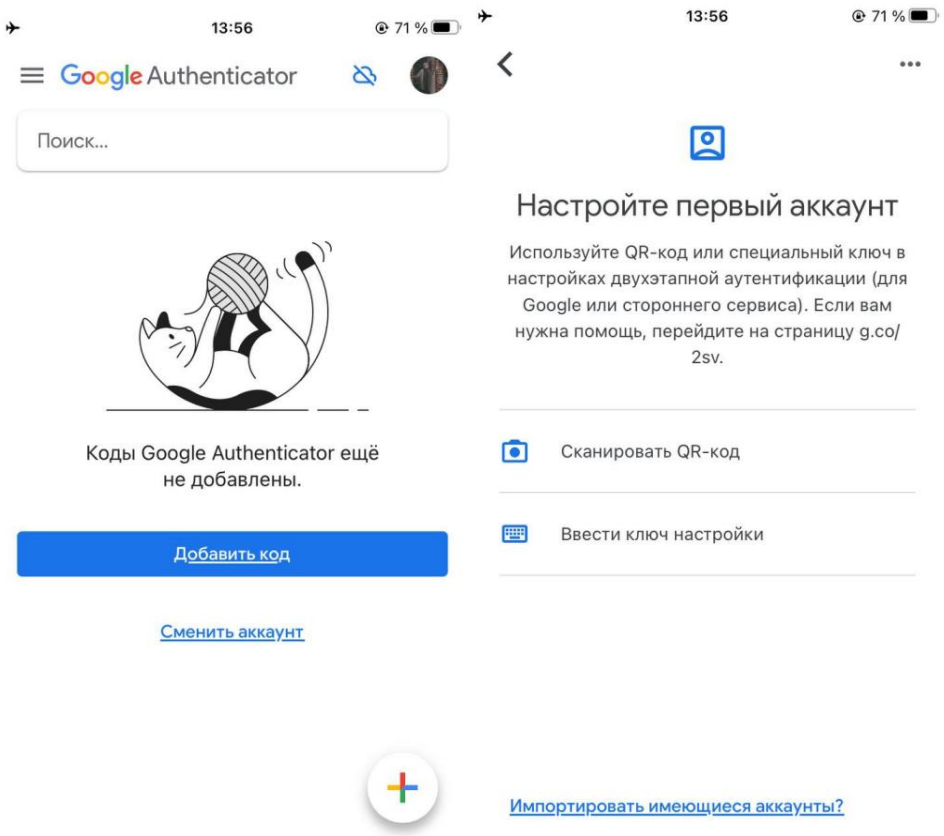

Рисунок 3.7 - Пустий аккаунт Google Authenticator для демонстрації

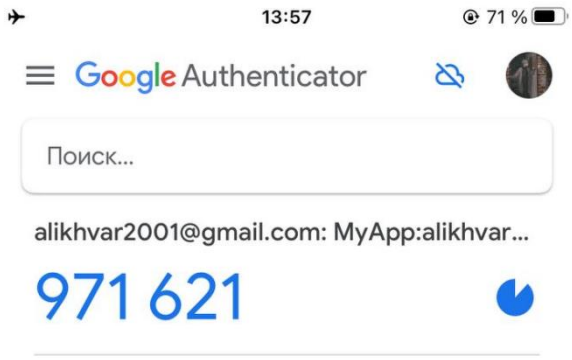

Рисунок 3.8 - Отриманий в результаті сканування QR коду секретний код

Відсканувавши QR код бачимо секретний код (рис.3.8) який ми можемо ввести у програму для верифікації(рис.3.9).

```
user_code = input("Enter your secret code: ")

totp = pyotp.TOTP(secret_code)
if totp.verify(user_code):
    print("Success!")
else:
    print("Incorrect code.")
```

Рисунок 3.9 - Верифікація введеного секретного коду

У змінну user_code зберігаємо введений користувачем для перевірки секретний код. Далі створюємо обьект TOTP, який використаємо для генерації одноразового паролю на основі часу. Та перевіряємо, чи збігається введений користувачем секретний код (user_code) з згенерованим об’єктом TOTP. Функція verify перевіряє правильність. Якщо коді збігаються, то программа виведе “Success” (рис.3.10), в іншому випадку програма виведе “Incorrect”(рис.3.11).

```
PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

PS C:\Users\An\Desktop\пр-диплом> & C:/Users/An/AppData/Local/Programs/Python/Python310
/python.exe "c:/Users/An/Desktop/пр-диплом/проект – копия/fa_orig.py"
Enter your secret code: 971621
Success!
PS C:\Users\An\Desktop\пр-диплом>
```

Рисунок 3.10 - Успішна верифікація

Але якщо ще раз використати той же самий код, то верифікація буде невдалою(рис.3.11).

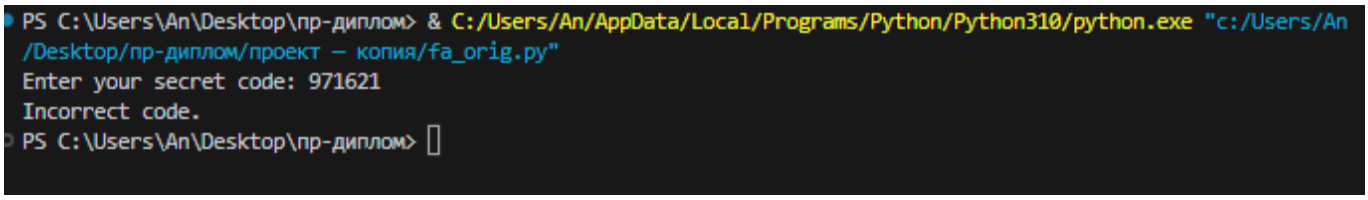

Рисунок 3. 11- Невдала верифікація

Для того, щоб підвищити зручність та безпеку цієї програми можна:

- Замість того, щоб генерувати секретний код щоразу під час запуску програми, подумати про його безпечне зберігання, наприклад, у базі даних або захищеному сховищі ключів. Це позбавить нас від необхідності щоразу генерувати новий секретний код і полегшить управління та пошук.

- Щоб захиститися від атак грубої сили на процес перевірки коду TOTP, можна впровадити обмеження швидкості. Це допоможе запобігти численним спробам швидкого введення TOTP-коду та сповільнить роботу потенційних зловмисників.

- Впровадите тайм-аути сеансів, щоб анулювати сеанс автентифікації користувача після певного періоду неактивності. Це додає додатковий рівень безпеки, автоматично виводячи користувачів з системи, коли їхня сесія залишається бездіяльною протягом певного часу.

- Хоча поточний код використовує бібліотеку qrcode для генерації QR-коду, потрібно переконатися, що використовується відома і надійна бібліотека. Необхідно перевірити її безпеку та надійність, а також розглянути можливість використання бібліотеки, яка активно підтримується та отримує регулярні оновлення безпеки.

- Додати чіткі інструкції для користувачів про те, як налаштувати додаток-автентифікатор та відсканувати QR-код. Це допоможе забезпечити безперебійну роботу користувача та зменшити ймовірність помилок під час процесу налаштування.

- Розглянути можливість впровадження додаткових рівнів автентифікації, наприклад, додавання другого фактору, такого як SMS-верифікація або біометрична автентифікація. Це додасть додатковий рівень безпеки до процесу автентифікації.

- Реалізувати безпечне введення TOTP-коду - замість використання функції введення, використовувати більш безпечний метод введення, якщо він доступний, щоб замаскувати введення користувача при введенні TOTP-коду. Це запобігає атакам "серфінгу через плече", коли хтось може побачити, як вводиться код.

Далі ми розглянемо реалізацію позасмугової автентифікації/OOB (Додаток Б).

Цей метод додає додатковий рівень безпеки, оскільки вимагає, щоб користувач володів як своїм основним пристроєм (наприклад, смартфоном), так і доступом до вторинного каналу.

```
import random
import string
```

Рисунок 3.12 - Імпорт необхідних модулів

Для початку імпортуємо необхідні модулі(рис.3.12). Ці модулі будуть використовуватися для генерації випадкових кодів та роботи з рядками.

```
def generate_code():
    return ''.join(random.choices(string.digits, k=6))
```

Рисунок 3.13 - Функція генерації випадкового коду з 6 символів

Потім створюємо першу необхідну функцію(рис.3.13) яка буде генерувати випадковий код з 6 цифр (k=6), функцію random.choices() використовуємо для випадкового вибору цифр з рядка string.digits.

```
def send_code(user_email, code):

    print(f"OOB secret code - '{code}' sent to {user_email}")
```

Рисунок 3.14 - Імітація відправки секретного коду на пошту

Далі функція send_code(), яка приймає 2 параметри: адреса електронної пошти користувача і згенерований код. Функція імітує відправку секретного коду на пошту користувача(рис.3.14).

```
def verify(user_code, code):
    return user_code == code
```

Рисунок 3.15 - Верифікація

Функція verify() також отримує два параметри: код введений користувачем і згенерований код, після чого порівнює їх (рис.3.15) та повертає True, якщо вони співпадають, що означає успішну перевірку, або False у протилежному випадку.

```
def main():

    code = generate_code()


    email = input('Enter ur email: ')
    send_code(email, code)

    user_code = input("Enter your OOB code: ")

    if verify(user_code, code):
        print("Success!")
    else:
        print("Incorrect code. Authentication failed.")
main()
```

Рисунок 3.16 - Головна функція

В головній функції(рис.3.16) ми за допомогою раніше створених функцій спочатку генеруємо код(рис.3.13), потім вводимо свою електронну адресу, після чого викликається функція send_code() для відправки згенерованого коду на вказану електронну пошту(рис.3.14). Після цього користувач вводить свій OOB код, що зберігається у змінній user_code і викликається функція verify для порівняння

наданого користувачем і згенерованого кодів(рис.3.15) і ми отримуємо результат True або False, в залежності від того, співвпадають вони чи ні(рис.3.16).

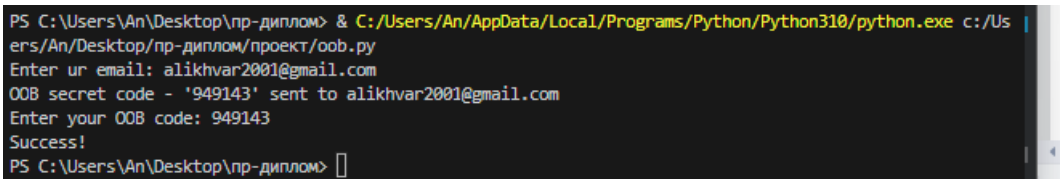

Рисунок 3.16 - Успішна верифікація

Для того, щоб підвищити зручність та безпеку цієї програми можна:

- Використовувати більшу довжину коду і більший набір символів для генерації коду. Це збільшить ентропію і ускладнить вгадування або грубий підбір коду. Можна використовувати комбінацію алфавітно-цифрових символів та спеціальних символів.

- Щоб захиститися від автоматизованих атак або спроб грубого перебору, ввести обмеження швидкості генерації коду та перевірки коду. Це може запобігти багаторазовому підбору кодів зловмисником.

- Встановити термін придатності для згенерованого коду. Це гарантує, що код стає недійсним після певного періоду, зменшуючи вікно можливостей для зловмисника використати витік або перехоплений код.

- Замість того, щоб друкувати код на консолі, можливо використати захищений каналу зв'язку для доставки коду користувачеві, наприклад, електронної пошти, SMS або спеціального сервісу обміну повідомленнями. Це зменшує ризик перехоплення коду зловмисником під час передачі

- Якщо потрібно зберігати згенеровані коди, переконатися, що вони надійно захищені за допомогою відповідного шифрування та контролю доступу. Використовувати безпечну систему управління ключами для захисту цілісності та конфіденційності кодів.

- Впровадити багатофакторну автентифікацію (БФА): OOB сам по собі може не забезпечити надійний захист. Можно поєднати з іншими факторами, такими як

пароль, біометрична автентифікація або апаратні токени, щоб забезпечити більш надійний механізм автентифікації.

- Замість загального повідомлення "Неправильний код", надати конкретні повідомлення про помилки, щоб допомогти користувачеві у випадку помилок при введенні або збоїв автентифікації. Це може допомогти користувачам усунути несправності та зрозуміти причину збоїв автентифікації.

- Регулярно переглядати та оновлювати код, щоб впроваджувати найновіші практики безпеки та усувати будь-які виявлені вразливості. Відслідковувати нові загроз і методи захисту, щоб забезпечити постійну ефективність впровадження.

Тепер перейдемо до реалізації та аналізу простої веб-сторінки, яка включає двофакторну автентифікацію (2FA) з використанням генерації QR-коду та JavaScript-верифікації (Додаток В та Додаток Г)

Веб-сторінка складається з контейнера з ідентифікатором "alles", який містить заголовну частину з логотипом і заголовком, за яким слідує форма верифікації (Рисунок 3.17).

Форма містить поле для введення електронної пошти(рис.3.18-3.20) та коду, а також кнопки для генерації QR-коду та перевірки коду.

JavaScript відіграє вирішальну роль в управлінні процесами генерації та перевірки QR-кодів. Він використовує бібліотеки, такі як qrcode.js, для генерації QR-коду на основі наданої адреси електронної пошти та інших параметрів.

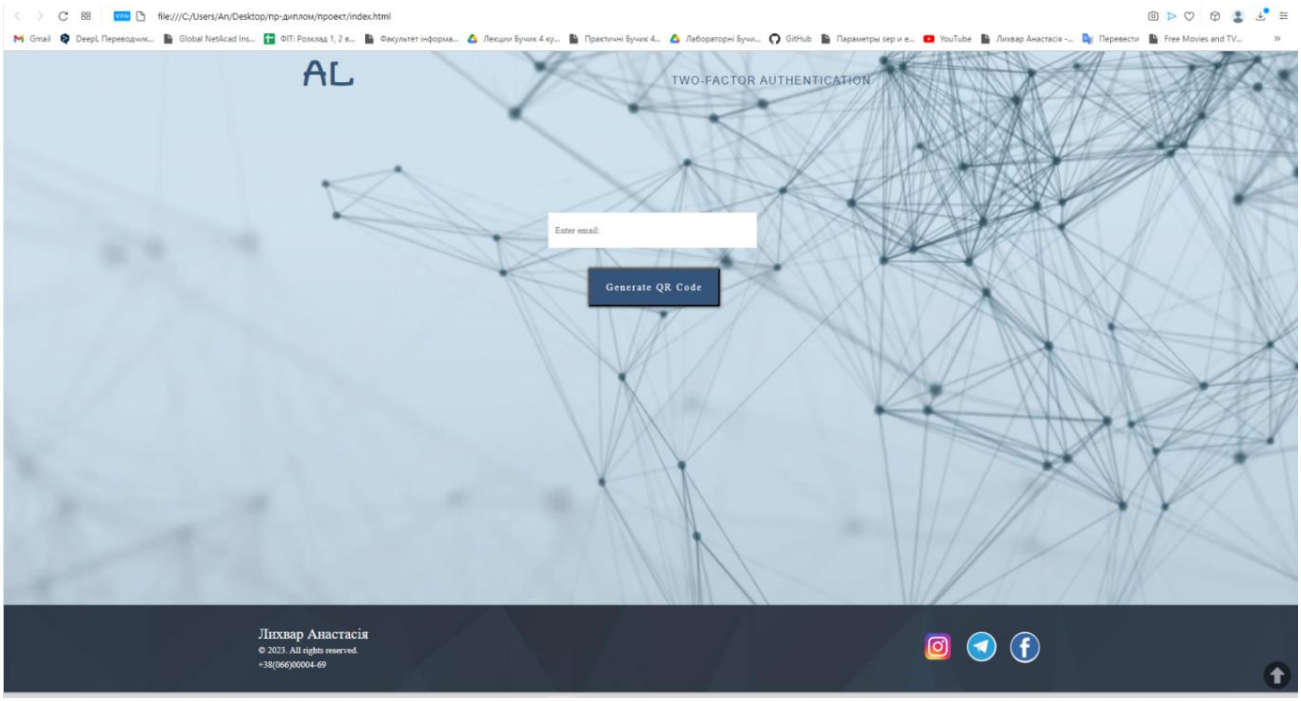

Рисунок 3.17 - Інтерфейс веб сторінки

Згенерований QR-код потім відображається на веб-сторінці для сканування користувачем.

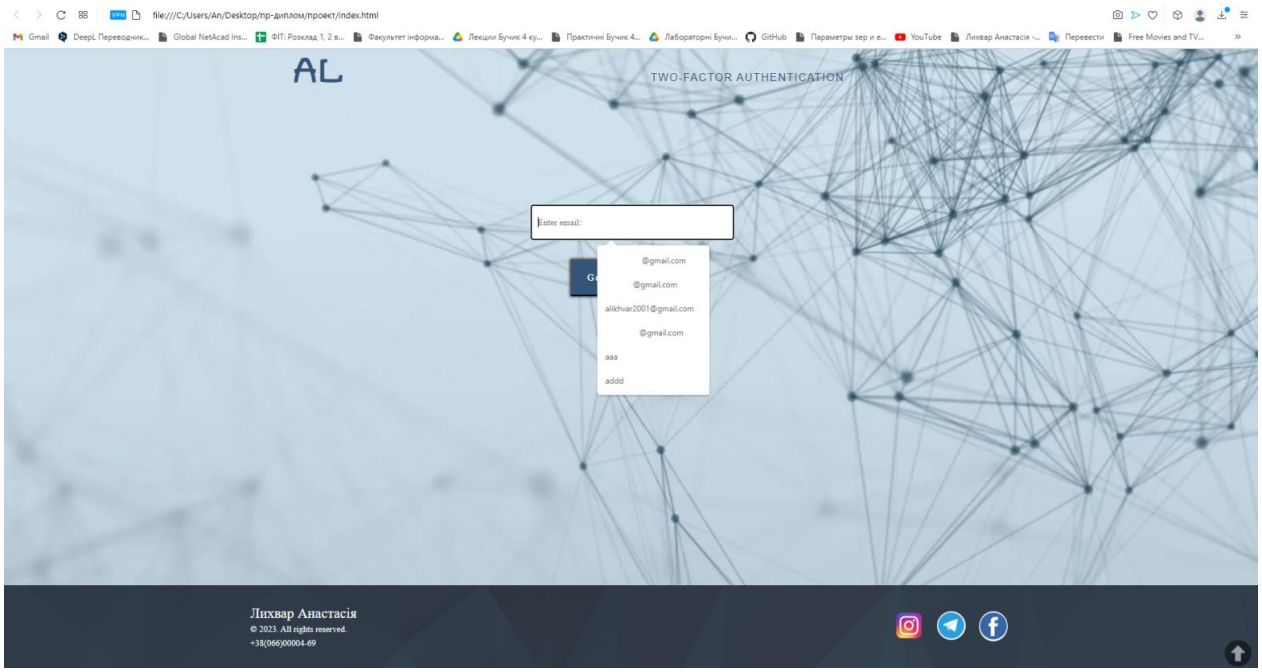

Рисунок 3.18 - Введення користувачем своєї адреси

Для реалізації цього функціоналу ми використовуємо HTML, CSS та JavaScript. HTML-структура веб-сторінки визначає необхідні елементи, такі як контейнер, заголовок і форма. CSS-стилі застосовуються для покращення візуального вигляду та розташування елементів.

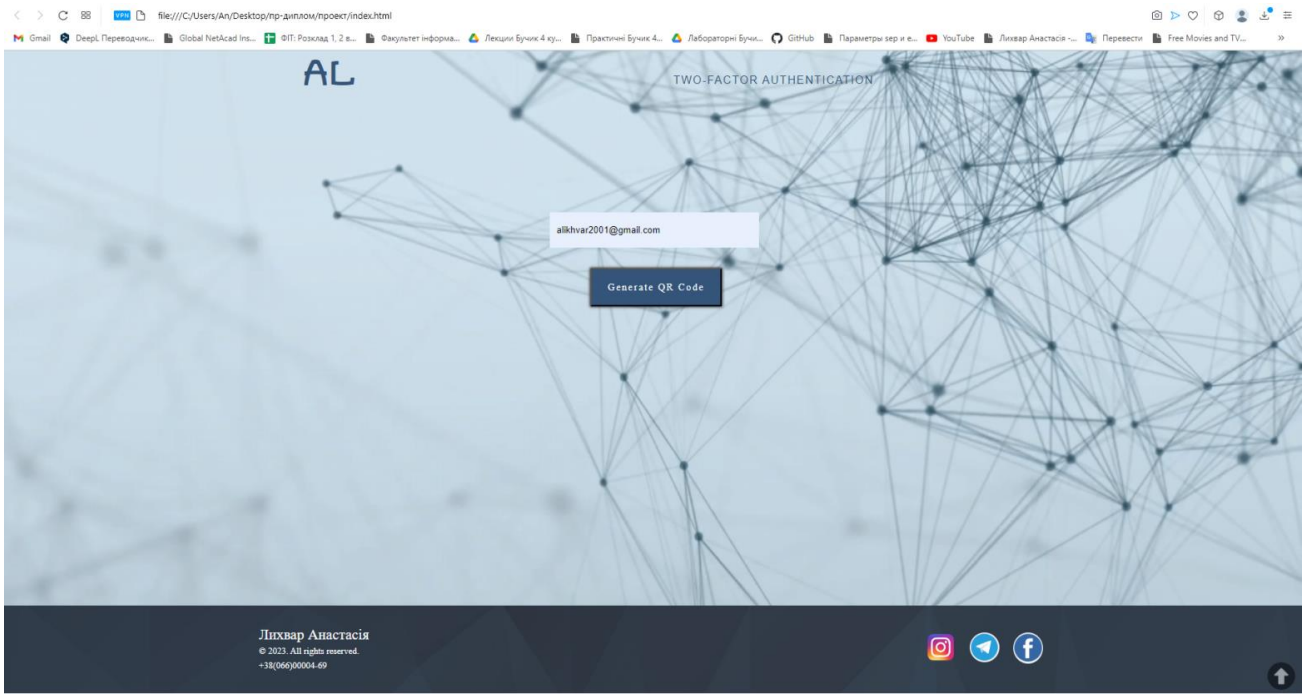

Рисунок 3.19 - Адреса користувача

QR-код буде відображатися в елементі div з ідентифікатором "qr-code"(рис.3.20).

Код на JavaScript реалізує той же самий алгоритм, що і на Python(Дод.А). Секретний код генерується шляхом виклику функції generateSecretKey, яка випадковим чином генерує 16-символьний ключ, використовуючи вказані символи.

На кнопці “Generate QR-code” висить прослуховувач подій. При натисканні він отримує значення електронної адреси, введеної користувачем, створює URI, використовуючи електронну адресу і секретний ключ, генерує QR-код та виводить його на екран, в той же час приховуючи поле вводу пошти. Цей QR-код можна відсканувати або за допомогою камери, або в спеціальному додатку(рис. 3.21). Після чого ми отримуємо необхідний нам код (рис.3.22).

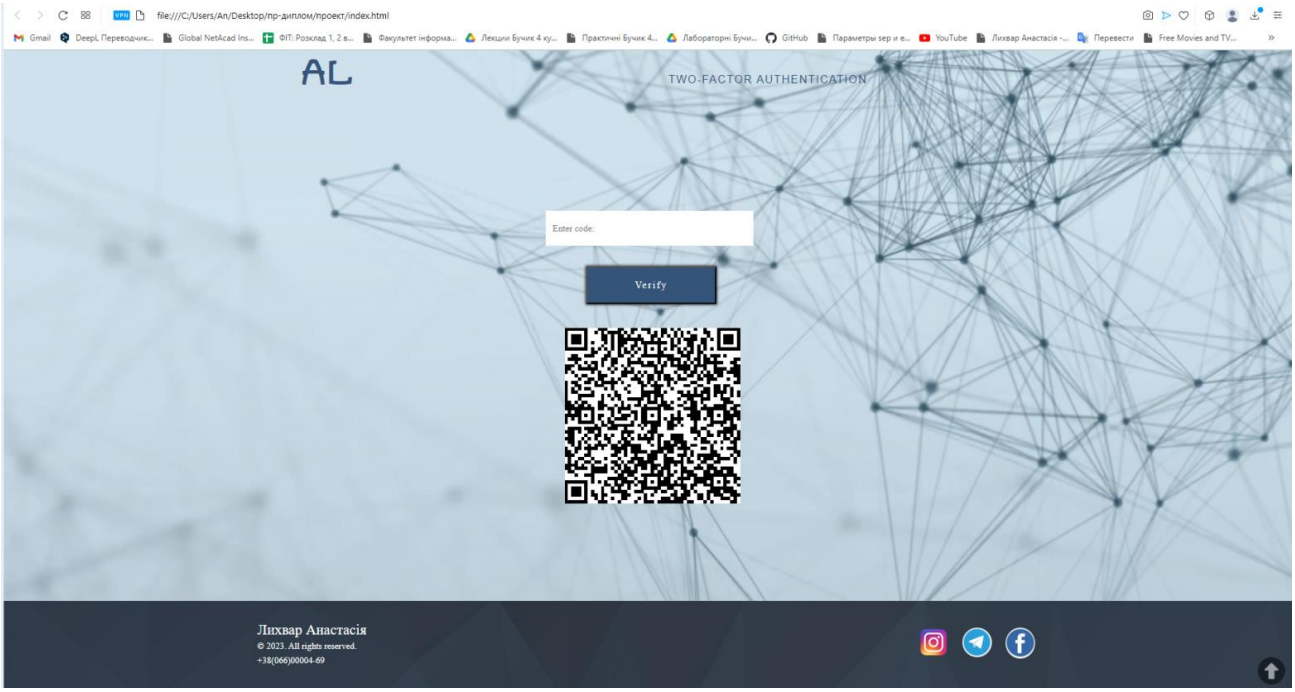

Рисунок 3.20 - Генерація QR коду для сканування в додатку GA

Після сканування QR-коду за допомогою мобільного додатку-автентифікатора користувач отримує код підтвердження з прив'язкою до часу. Цей код вводиться у форму, і логіка JavaScript перевіряє його правильність, порівнюючи його з очікуваним кодом, згенерованим на стороні сервера.

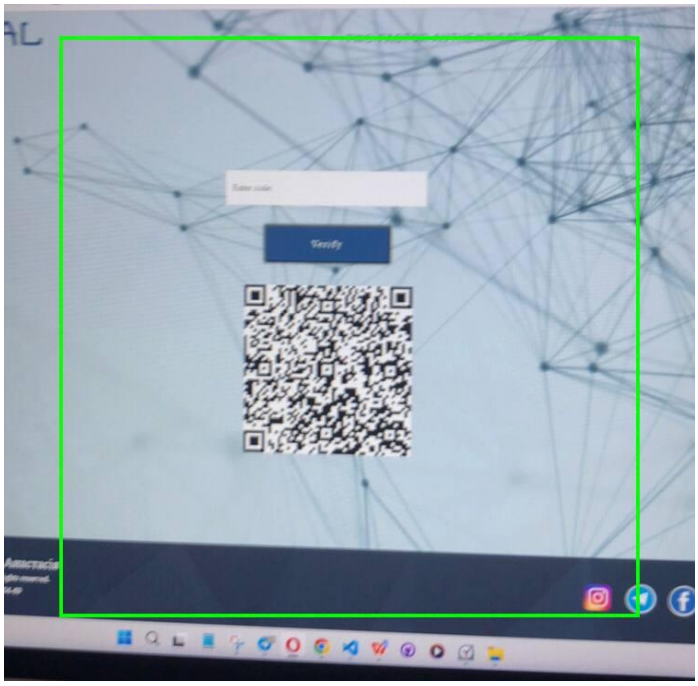

Рисунок 3.21 - Сканування користувачем у додатку

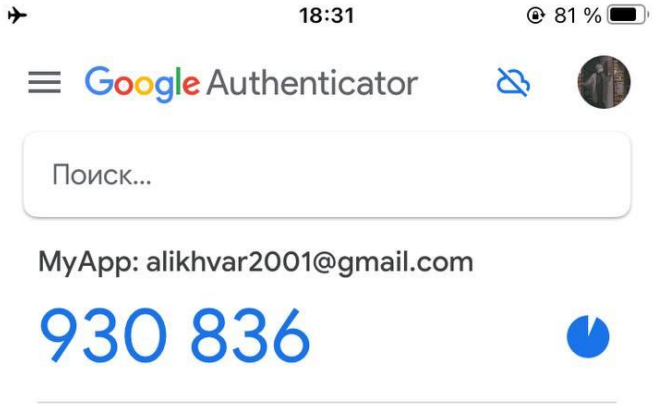

Рисунок 3.22 - Отриманий код

Для того, щоб підвищити зручність та безпеку цього сайту можна:

- Наданий код, працює повністю на стороні клієнта (у браузері). Для забезпечення безпеки та цілісності процесу автентифікації необхідно виконувати валідацію та верифікацію на стороні сервера.

- Секретний ключ генерується і зберігається в коді на стороні клієнта, що становить ризик для безпеки. Краще генерувати та безпечно зберігати секретний ключ на стороні сервера, щоб запобігти його розголошенню та несанкціонованому доступу.

- Переконатися, що веб-сторінка обслуговується за протоколом HTTPS для шифрування зв'язку між сервером і клієнтом, запобігаючи підслуховуванню і підробці даних.

- Використовувати відповідні формати вводу, такі як "електронна пошта" та "пароль", щоб забезпечити правильне введення даних та покращити користування.

- Очищати поля введення після того, як користувач відправив форму, щоб підвищити зручність використання та запобігти випадковому повторному відправленню.

- Надавати чіткі та інформативні повідомлення користувачу, що вказують на успіх або невдачу процесу верифікації.

- Впровадити належну обробку помилок, щоб впоратися з несподіваними сценаріями, такими як мережеві помилки або невірні вхідні дані, і надавати користувачеві зрозумілі повідомлення про помилки.

Покращимо програмну реалізацію TOTP з використанням QR кодів у консольному додатку на Python.

За рахунок чого ми покращили попередній код:

По - перше, покращена структура та організація коду:

Код розділено на кілька функцій, кожна з яких відповідає за певне завдання (рис.3.23). Це значно покращує читабельність коду, організацію та можливість повторного його використання.

```python
def generate_secret_key():
    return pyotp.random_base32()

def generate_qr_code(email, secret_key):
    provisioning_uri = pyotp.totp.TOTP(secret_key).provisioning_uri(
        name=email,
        issuer_name='MyApp'
    )
    qr = qrcode.QRCode(version=1, box_size=10, border=5)
    qr.add_data(provisioning_uri)
    qr.make(fit=True)
    img = qr.make_image(fill_color="black", back_color="white")
    img.show()

def verify_totp_code(secret_key, user_code):
    totp = pyotp.TOTP(secret_key)
    return totp.verify(user_code)
```

Рисунок 3.23 - Розділення коду на функції

По - друге, підвищенно безпеку за рахунок зберігання секретних ключів пов’язаних з електронною поштою у словнику user_secret_keys (рис.3.24). Також код тепер перевіряє чи існує адреса у `user_secret_keys` і отримує відповідний секретний ключ(рис.3.25). Якщо адресу не знайдено, генерується новий секретний ключ, який зберігається у `user_secret_keys`. Це забезпечує більш плавну роботу користувача з програмою завдяки повторному використанню секретних ключів і генерації QR-кодів для відомих адрес електронної пошти.

```
user_secret_keys = {}
```

Рисунок 3.24 - Створення словника user_secret_keys.

```
def main():
    email = input("Enter your email: ")


    if email in user_secret_keys:
        secret_key = user_secret_keys[email]
    else:
        secret_key = generate_secret_key()
        user_secret_keys[email] = secret_key


    generate_qr_code(email, secret_key)
```

Рисунок 3.25 - Перевірка наявності у словнику введеної адреси

І по - третє, покращено безпеку отримання TOTP-коду від користувача без виведення його на консоль при вводі, що покращує захист від потенційних спостерігачів. Для цього в коді підключено (рис.3.26) та використано функцію getpass() замість звичайного input() (рис.3.27).

```
import pyotp
import qrcode
import getpass
```

Рисунок 3.26 - Підключення модулю getpass

```
user_code = getpass.getpass("Enter your TOTP code: ")
```

Рисунок 3.27 - Введення користувачем TOTP коду

На рис 3.28 - 3.30 можна побачити результат виконання програмного коду.

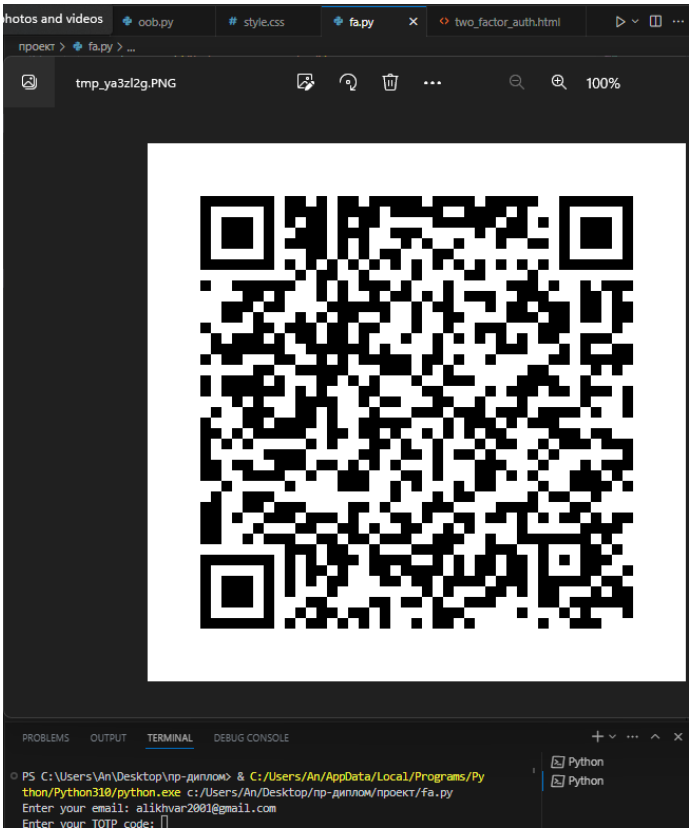

Рисунок 3.28 - Введення початкових даних та виведення QR-коду для сканування

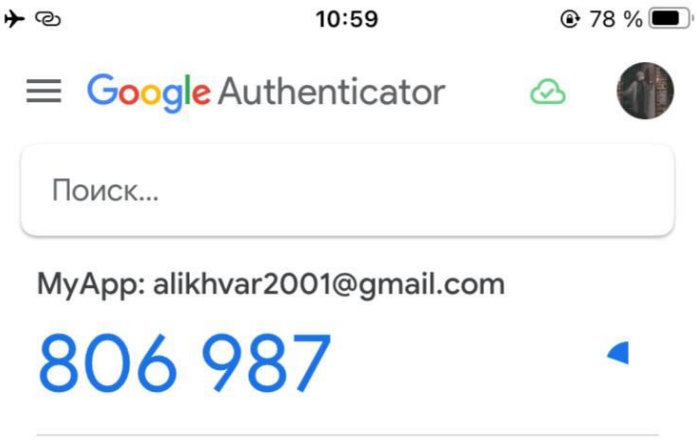

Рисунок 3.29 - Отриманий код в результаті сканування QR-коду

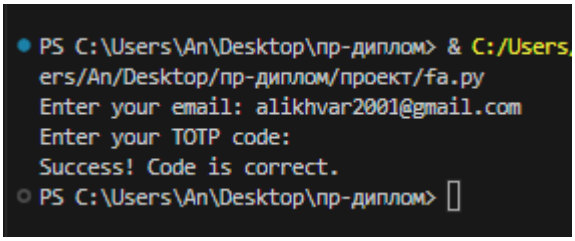

Рисунок 3.30 - Введення коду до додатку і вдала аутентифікація

3.3 Висновки про доцільність використання розроблених засобів для підвищення ефективності технологій доступу до захищених інформаційних ресурсів

Проаналізувавши наведені раніше програмні застосунки можна зробити висновок, що ці інструменти мають значний потенціал у підвищенні ефективності технологій безпечного доступу до захищених ресурсів. Перший програмний застосунок TOTP у консольному додатку на Python демонструє використання бібліотеки для генерації секретного ключа та перевірки TOTP-коду користувача - pyotp, а також бібліотеки qrcode для створення QR-коду. Код генерує випадковий секретний ключ, створює QR-код, що включає в себе URI резервування, і потім перевіряє дані введені користувачем за згенерованим TOTP-кодом. Одне з потенційних удосконалень полягає в безпечній обробці введення користувачем коду TOTP, наприклад, впровадження модуля обходу для приховування введення коду, що було реалізовано в удосконаленому програмному коді.

Переходячи до демонстрації OOB, бачимо спрощену реалізацію, яка зосереджена на генерації випадкового перевірочного коду з обмеженої кількості символів і передачі його користувачеві через позасмуговий канал, а саме електронну пошту. Код генерує випадковий 6-значний код, запитує у користувача його електронну пошту, код надсилається за допомогою send_code() і потім код введений користувачем перевіряється на відповідність згенерованому коду. Як удосконалення можна розглянути більш безпечний та надійний канал OOB, такого як SMS або спеціальна служба автентифікації, залежно від конкретних вимог безпеки. Таке вдосконалення забезпечить сильніший рівень захисту під час передачі перевірочних кодів.

При реалізації простої веб-сторінки, використовується HTML, CSS та JavaScript для створення веб-форми двохфакторної аутентифікації, що може генерувати QR-коди. Завдяки використанню бібліотеки qrcode для генерації QR-коду та бібліотеки otpauth для верифікації TOTP-коду, цей приклад коду пропонує значні покращення з точки зору зручності та функціональності. Впровадження динамічної генерації QR-

коду та полів верифікації покращує взаємодію з користувачем, в той час як реалізація більш рандомізованого процесу генерації секретного ключа посилює безпеку механізму автентифікації. Ці вдосконалення в сукупності сприяють створенню більш надійного та надійного рішення 2FA.

В останньому додатку, що розширює попередній приклад, вводяться додаткові функції для підвищення рівня безпеки, а також словник для зберігання секретних ключів, пов'язаних з адресами електронної пошти. Код генерує секретний ключ, потім зберігає його в словнику user_secret_keys на основі адреси електронної пошти користувача, генерує QR-код і звіряє введений користувачем код з TOTP-кодом. Серед важливих покращень - модульна організація коду, безпечна обробка введення TOTP-коду за допомогою getpass та централізоване зберігання секретних ключів для кожної адреси електронної пошти. Ці вдосконалення покращують загальну структуру коду, підвищують безпеку та сприяють ефективному управлінню секретними ключами.

Таким чином, приклади кодів, представлені в цьому підрозділі, ефективно підвищують ефективність доступу до захищених інформаційних ресурсів. Серед помітних особливостей - генерація секретних ключів, генерація QR-кодів, безпечна обробка вхідних даних, централізоване зберігання ключів і перевірка TOTP-кодів. Тим не менш, ефективність і придатність кожного інструменту може варіюватися залежно від конкретних вимог безпеки і контексту впровадження. Тож слід ретельно підійти до вибору інструменту, щоб адаптувати його до унікальних потреб системи або організації. Загалом, ці приклади коду представляють практичні рішення, які сприяють зміцненню технологій безпечного доступу та захисту конфіденційної інформації від несанкціонованого доступу.

Висновки за розділом 3

Основна увага в цьому розділі зосереджена на розробці та впровадженні рішень, спрямованих на підвищення ефективності доступу до захищених інформаційних ресурсів.

Розділ 3.1 містить опис запропонованого рішення. Він передбачає вибір та опис обраної технології для вдосконалення. Опис обраної технології слугує основою для подальшої розробки та впровадження.

У розділі 3.2 переходимо до практичної реалізації запропонованого рішення. У ньому розглядаються технічні тонкощі, методології та інструменти, що використовуються для втілення рішення в життя. Цей розділ забезпечує глибоке розуміння процесу розробки, проливаючи світло на будь-які конкретні міркування або проблеми, що виникають на етапі впровадження.

У розділі 3.3 зроблено висновки щодо значущості та практичності розроблених інструментів для підвищення ефективності технологій доступу до захищених інформаційних ресурсів. У цьому розділі ретельно аналізується ефективність реалізованого рішення з урахуванням його впливу на безпеку, зручність використання та загальну продуктивність системи. Висновки мають на меті підтвердити цінність розроблених інструментів та їхній потенціал для розширення доступу до захищених інформаційних ресурсів.

У третьому розділі основна увага приділяється розробці та впровадженню рішень, які вдосконалюють технології доступу, забезпечуючи безпечне та ефективне використання захищених інформаційних ресурсів. Розглядаючи запропоновані рішення, деталі їх реалізації та висновки, зроблені на основі їх ефективності, ми робимо свій внесок у постійний розвиток галузі інформаційної безпеки.

ВИСНОВКИ

У дипломній роботі було представлено комплексну архітектуру та рішення для вдосконалення технологій доступу, з особливим акцентом на двофакторну автентифікацію. Запропонована система використовує мобільні пристрої як другий фактор у процесі автентифікації. Завдяки додатковому рівню перевірки, система ефективно зменшила ризик несанкціонованого доступу та підвищила безпеку інформаційних ресурсів.

У першому розділі дипломної роботи було розглянуто нормативно-правову базу, що регулює інформаційну безпеку, а також проаналізовано сучасні методи, що застосовуються для захисту інформації. Дослідження підкреслило важливість створення надійного фундаменту для інформаційної безпеки шляхом дотримання відповідних законів і нормативних актів. Крім того, вивчення існуючих інструментів і методів підкреслило необхідність їх постійного вдосконалення для проактивного реагування на нові загрози та вразливості.

У другій частині дипломної роботи було проведено дослідження методів і засобів захисту інформаційних ресурсів. Завдяки дослідженню захисних механізмів і методів, що забезпечують цілісність, конфіденційність і доступність інформаційних ресурсів, цей розділ надав цінну інформацію про сучасний стан інформаційної безпеки. Висновки підтвердили необхідність прийняття комплексного, багаторівневого підходу до захисту конфіденційних даних, що охоплює технічні та організаційні заходи. Вибір відповідних технологій доступу має вирішальне значення для досягнення балансу між зручністю використання та безпекою.

У третій частині дипломної роботи було вдосконалено рішення, спрямоване на захист технологій доступу, зокрема шляхом інтеграції системи двофакторної автентифікації, яка продемонструвала практичне застосування результатів дослідження. Запропоноване рішення ефективно усунуло вразливості, пов'язані з однофакторною автентифікацією, шляхом введення додаткового рівня перевірки

через мобільні пристрої. Результати продемонстрували доцільність та ефективність двофакторної автентифікації для посилення безпеки інформаційних ресурсів.

Всі задачі було виконано в повному обсязі.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Smith, J. (2020). Enhancing Access Control Mechanisms for Protected Information Resources. Journal of Information Security.

2. Johnson, R. (2018). Multi-Factor Authentication: An Effective Approach to Secure Information Access. International Journal of Cybersecurity.

3. Müller, K. (2017). Verbesserung der Zugangstechnologien zu geschützten Informationsressourcen. Sicherheit und Datenschutz.

4. Ковальчук, О. (2021). Оновлення технологій доступу до захищених інформаційних ресурсів. Інформаційна безпека.

5. Schmidt, F., & Wagner, H. (2019). Multi-Faktor-Authentifizierung: Effektive Maßnahmen zur sicheren Informationszugriff. Zeitschrift für IT-Sicherheit.

6. Про основні засади забезпечення кібербезпеки України [Електронний ресурс]: Закон України від 28.07.2022 № 2470-IX – Режим доступу: https://zakon.rada.gov.ua/go/2163-19

7. Нормативно-правова база [Електронний ресурс]. – Режим доступу: https://ligabezinfo.org/legislation/

8. Двухфакторная аутентифікація: що це і навіщо воно потрібне? [Електронний ресурс]. – Режим доступу: https://aistudio.com.ua/quests/dvuhfaktornaa/uk/avtorizacia-dvuhfaktornaa-autentifikacia-so-ce-i-naviso-vono-potribne-blog-laboratorii-kasperskogo.html

9. Protect Yourself Against Computer Hackers [Електронний ресурс]. – Режим доступу: https://www.hbc.bank/protect-yourself-against-computer-hackers/

10. IETF RFC 6238. (2011). TOTP: Time-Based One-Time Password Algorithm. [Електронний ресурс]. – Режим доступу: https://tools.ietf.org/html/rfc6238

11. Надійна двофакторна аутентифікація: додатковий захист ваших облікових записів [Електронний ресурс]. – Режим доступу: https://www.eset.com/ua/about/newsroom/blog/business-security/nadezhnaya-dvukhfaktornaya-autentifikatsiya-dopolnitelnaya-zashchita-vashikh-uchetnykh-zapisey/

12. TOTP [Електронний ресурс]. – Режим доступу: https://www.twilio.com/docs/glossary/totp

13. Про захист інформації в інформаційно-комунікаційних системах [Електронний ресурс]: Закон України від 15.03.2022 № 2130-IX – Режим доступу: https://zakon.rada.gov.ua/go/80/94-%D0%B2%D1%80

14. Про державну таємницю [Електронний ресурс]: від 13.12.2022 № 2849-IX – Режим доступу: https://zakon.rada.gov.ua/go/3855-12

15. Про розвідку [Електронний ресурс]: Закон України від 13.12.2022 № 2849-IX – Режим доступу: https://zakon.rada.gov.ua/go/912-20

16. Стеценко, В.І. (2017). Методи та засоби забезпечення захисту інформації [Stetsenko, V.I. (2017). Methods and Tools for Information Protection].

17. Reimer, H., & Heisel, M. (2014). Datenschutz und Datensicherheit: Konzepte, Realisierungen, Rechtliche Aspekte [Reimer, H., & Heisel, M. (2014). Data Protection and Data Security: Concepts, Implementations, Legal Aspects].

18. Marimuthu, M., Adnan, N.A., Khezri, R., Zolkipli, M.F., & Tan, K.W. (2020). Two-Factor Authentication: Techniques and Challenges. IEEE Access, 8, 117946-117968.

19. Ільїн, І.В., Петренко, С.О., Кузнецов, С.В. (2020). Використання технології двофакторної аутентифікації у сучасних інформаційних системах [Ilyin, I.V., Petrenko, S.O., Kuznetsov, S.V. (2020). The Use of Two-Factor Authentication Technology in Modern Information Systems].

20. Aqeel-ur-Rehman, M., Al-Samarraie, H., & Al-Jumeily, D. (2021). Two-Factor Authentication and Its Effect on User Experience: A Systematic Literature Review. IEEE Access, 9, 19686-19709.

21. Müller, S., & Mühlhäuser, M. (2021). Security, Privacy, and Usability of Multi-Factor Authentication: A Survey. Computers & Security, 105, 102316.

22. Сучасні методи та засоби захисту інформації [Електронний ресурс]. – Режим доступу: https://archer.chnu.edu.ua/bitstream/handle/123456789/2831/Main%20article.pdf?sequence=1

23. TOTP [Електронний ресурс]. – Режим доступу: https://www.netiq.com/documentation/advanced-authentication-64/server-user-guide/data/totp_oob.html

24. What is out-of-band authentication? [Електронний ресурс]. – Режим доступу: https://www.onespan.com/topics/out-of-band-authentication

25. What is TOTP? (Time-based one-time password) [Електронний ресурс]. – Режим доступу: https://www.ionos.com/digitalguide/server/security/totp/

26. Out of Band Authentication: Practical use cases [Електронний ресурс]. – Режим доступу: https://cybersecurity.asee.co/blog/out-of-band-authentication/

27. pyotp · PyPI [Електронний ресурс]. – Режим доступу: https://pypi.org/project/pyotp/

28. Multi-Factor Authentication: Who Has It and How to Set It Up [Електронний ресурс]. – Режим доступу: https://www.pcmag.com/how-to/multi-factor-authentication-2fa-who-has-it-and-how-to-set-it-up

29. Ivanenko, O. (2022). Роль двофакторної аутентифікації в забезпеченні безпеки інформаційного доступу [The Role of Two-Factor Authentication in Ensuring Information Access Security]. Український журнал кібербезпеки, 10(2), 45-60.

30. Smith, A. (2020). Analysis of Existing Methods of Protection: A Comparative Study. Journal of Information Security, 8(2), 76-92.

ДОДАТКИ

ДОДАТОК А

Програмна реалізація TOTP у консольному додатку на Python

```
import pyotp
import qrcode
secret = pyotp.random_base32()
qr = qrcode.QRCode(version=1, box_size=10, border=5)
qr.add_data(pyotp.totp.TOTP(secret).provisioning_uri("MyApp:alikhvar2001@gmail.com", issuer_name="alikhvar2001@gmail.com"))
qr.make(fit=True)
img = qr.make_image(fill_color="black", back_color="white")
img.show()
user_code = input("Enter your 2FA code: ")
totp = pyotp.TOTP(secret)
if totp.verify(user_code):
    print("Success!")
else:
    print("Incorrect code.")
```

ДОДАТОК Б

Програмна реалізація OOB у консольному додатку на Python

```
import random
import string
def generate_code():
    return ''.join(random.choices(string.digits, k=6))
def send_code(user_email, code):
        print(f"OOB secret code - '{code}' sent to {user_email}")
def verify(user_code, code):
    return user_code == code
def main():
    code = generate_code()
    email = input('Enter ur email: ')
    send_code(email, code)
    user_code = input("Enter your OOB code: ")
    if verify(user_code, code):
        print("Success!")
    else:
        print("Incorrect code. Authentication failed.")
main()
```

ДОДАТОК В

Програмна реалізація TOTP у веб додатку з використанням QR кодів і Google Authenticator

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Verification</title>
    <link rel="stylesheet" href="style.css">
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?family=Revalia&display=swap" rel="stylesheet">
</head>
<body>
    <!-- Two-Factor Authentication -->
    <div id="alles">
        <div id="header">
            <div id="logo">
                <p>AL</p>
            </div>
            <div id="block-menu">
                <p>Two-Factor Authentication</p>
            </div>
        </div>
        <div id="contact-form">
```

```
            <form id="contact">
                <input type="text" id="email" placeholder="Enter email:">
                <input type="text" id="code" placeholder="Enter code:" class="hidden">
                <button type="button" id="generate">Generate QR Code</button>
                <button type="button" id="verify" class="hidden">Verify</button>
                <div id="qr-code"></div>
            </form>
        </div>
    </div>
    <script
src="https://cdnjs.cloudflare.com/ajax/libs/qrcodejs/1.0.0/qrcode.min.js"></script>

    <script
src="https://cdnjs.cloudflare.com/ajax/libs/otpauth/VERSION/otpauth.umd.min.js"></scri
pt>
    <script type="module">
        const qrcode = new QRCode('qr-code');
        const secret = generateSecretKey();
        document.getElementById('generate').addEventListener('click', () => {
            const email = document.getElementById('email').value;
            const                    provisioningUri                    =
`otpauth://totp/MyApp:${encodeURIComponent(email)}?secret=${secret}&issuer=MyAp
p`;
            qrcode.makeCode(provisioningUri);
            document.getElementById('email').classList.add("hidden");
            document.getElementById('generate').classList.add("hidden");
            document.getElementById('code').classList.remove("hidden");
            document.getElementById('verify').classList.remove("hidden");
        });
```

```
document.getElementById('verify').addEventListener('click', () => {
    const userCode = document.getElementById('code').value;
    // Verify the code is correct using the pyotp library
    const isCodeValid = verifyOTPCode(secret, userCode);
    if (isCodeValid) {
        resultText.textContent = "Success!";
    } else {
        resultText.textContent = "Incorrect code.";
    }
    document.getElementById('code').value = '';
});


function generateSecretKey() {
    const characters = 'ABCDEFGHIJKLMNOPQRSTUVWXYZ234567';
    let secret = '';
    for (let i = 0; i < 16; i++) {
        secret += characters.charAt(Math.floor(Math.random() * characters.length));
    }
    return secret;
}
function verifyOTPCode(secret, userCode) {
    const otpauth = require('otpauth');
    // Create a TOTP object with the secret
    const totp = new otpauth.TOTP({ secret });
    // Verify the code
    const isValid = totp.validate({ token: userCode });
    return isValid;
}
```

```
    </script>

    <div id="footer">
      <div id="footer-text">
        <h3>Лихвар Анастасія</h3>
        <p>
          &copy; 2023. All rights reserved.
        </p>
        <p>
          +38(066)00004-69
        </p>
      </div>
      <div id="footer-img">
        <a href="https://instagram.com/" target="_blank"><img src="img/insta.png"
alt=""></a>
        <a href="https://web.telegram.org/" target="_blank"><img
src="img/telegram.png" alt=""></a>
        <a href="https://uk-ua.facebook.com/" target="_blank"><img
src="img/facebook.png" alt=""></a>
      </div>
    </div>
    <div id="up">
      <a href="#header"><img src="img/arrow.png" alt=""></a>
    </div>
  </body>
  </html>
```

ДОДАТОК Г

Програмна реалізація TOTP у веб додатку з використанням QR кодів і Google Authenticator

```
* {
    margin: 0;
    padding: 0;
}
#alles {
    background-color: #EEEFF1;
}
#header {
    display: flex;
}
#logo {
    width: 50%;
    /* background-color: #34537ab9; */
}
#logo > p {
    font-family: 'Revalia';
    font-style: normal;
    font-weight: normal;
    font-size: 48px;
    line-height: 60px;
    text-align: center;
    color: #34547A;
}
#block-menu {
    font-family: 'Roboto', sans-serif;
```

```
    font-style: normal;
    font-weight: 500;
    font-size: 16px;
    line-height: 26px;
    letter-spacing: 0.1em;
    text-decoration: none;
    /* Accent */
    text-transform: uppercase;
    margin: 30px;
    color: #34547A;
}
#contact {
    height: 392px;
    font-family: 'Robota';
    background-color: #EEEFF1;
}
#contact-text {
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
    margin-top: 80px;
}
#contact-text h2 {
    width: 406px;
    font-family: 'Roboto';
     font-size: 32px;
    line-height: 37px;
    text-align: center;
    color: #000000;
```

```
}
#contact-text p {
    width: 504px;
    margin-top: 30px;
    font-family: 'Roboto';
    font-size: 16px;
    line-height: 26px;
    text-align: center;
    color: #727272;
}
#contact-form {
    display: flex;
    align-items: center;
    font-family: 'Roboto';
    justify-content: center;
    margin-top: 50px;
    padding-bottom: 30px;
    position: relative;
    z-index: 1;
    height: 530px;
}
input {
    width: 293px;
    height: 51px;
    border: none;
    /* margin: 10px 20px 20px 0; */
    /* margin-right: -143px; */
    margin: 0px 131px;
}
form {
```

```
    width: 590px;
    margin-left: 30px;
}
textarea {
    width: 540px;
    height: 175px;
    border: none;
    resize: none;
}
textarea[name="message"]::placeholder {
    font-family: 'Roboto';
}
input[type="text"]::placeholder {
    font-family: 'Roboto';
}
input[type="email"]::placeholder {
   font-family: 'Roboto';
}
textarea[name="message"],
input[type="text"],
input[type="email"] {
    padding-left: 10px;
}
textarea[name="message"] {
    padding-top: 10px;
}
#contact-form button {
    width: 190px;
    height: 55px;
    background-color: #34547A;
```

```
    color: #fff;
    font-family: Roboto;
    font-size: 14px;
    line-height: 23px;
    text-align: center;
    letter-spacing: 0.15em;
    margin-top: 30px;
    box-shadow: 0px 0px 4px 1px #272727, inset 0px 0px 0px 0px #345467;
    /* position: absolute;
    left: 50%;
    top: 260px;
    transform: translate(-50%); */
    margin-left: 190px;
}
#footer {
    background-color: #34547A;
    display: flex;
    justify-content: center;
    align-items: center;
    margin-top: 80px;
    position: relative;
    z-index: 22;
    background-image: url(img/footer-background-01.jpg);
}
#footer h3 {
    font-family: 'Roboto';
    width: 200px;
    font-size: 21px;
    line-height: 25px;
    color: #FFFFFF;
```

```
    font-weight: 300;
}
#footer p {
    font-family: 'Roboto';
    width: 200px;
    font-size: 13px;
    line-height: 21px;
    color: #FFFFFF;
}
#footer img {
    width: 43px;
 }
#footer-text,
#footer-img {
    width: 50%;
    margin: 30px;
    display: flex;
    justify-content: center;
}
#footer-text {
    display: flex;
    justify-content: center;
    align-items: center;
    flex-direction: column;
}
#footer-img *{
    margin-left: 10px;
}
img[src="img/telegram.png"],
img[src="img/facebook.png"] {
```

```
    border-radius: 50%;
}
#up img {
    border-radius: 50%;
    width: 40px;
    z-index: 1000;
    position: fixed;
    right: 10px;
    bottom: 6px;
    opacity: 50%;
}
#up img:hover {
    opacity: 100%;
}
#up a:active {
    transition: 1s;
}
#qr-code {
    display: block;
    width: 200px;
    height: 200px;
    margin: 35px 159px;
}
.hidden {
    display: none;
}
#verify {
    width: 190px;
    height: 55px;
    background-color: #34547A;
```

```
    color: #fff;
    font-family: Roboto;
    font-size: 14px;
    line-height: 23px;
    text-align: center;
    letter-spacing: 0.15em;
    margin-top: 30px;
    box-shadow: 0px 0px 4px 1px #272727, inset 0px 0px 0px 0px #345467;
    /* position: absolute;
    left: 50%;
    top: 260px;
    transform: translate(-50%); */
    margin-left: 190px;
}
.show {
    display: inline;
}
```

ДОДАТОК Д

Удосконалена програмна реалізація TOTP у веб додатку з використанням QR-кодів і Google Authenticator

```
import pyotp
import qrcode
import getpass
user_secret_keys = {}
def generate_secret_key():
    return pyotp.random_base32()
def generate_qr_code(email, secret_key):
    provisioning_uri = pyotp.totp.TOTP(secret_key).provisioning_uri(
        name=email,
        issuer_name='MyApp'    )
    qr = qrcode.QRCode(version=1, box_size=10, border=5)
    qr.add_data(provisioning_uri)
    qr.make(fit=True)
    img = qr.make_image(fill_color="black", back_color="white")
    img.show()
def verify_totp_code(secret_key, user_code):
    totp = pyotp.TOTP(secret_key)
    return totp.verify(user_code)
def main():
    email = input("Enter your email: ")
    if email in user_secret_keys:
        secret_key = user_secret_keys[email]
    else:
        secret_key = generate_secret_key()
        user_secret_keys[email] = secret_key
```

```
    generate_qr_code(email, secret_key)
    user_code = getpass.getpass("Enter your TOTP code: ")
    if verify_totp_code(secret_key, user_code):
        print("Success! Code is correct.")
    else:
        print("Incorrect code. Authentication failed.")
main()
```