Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

Бакалаврська кваліфікаційна робота на тему:

«Засіб для перевірки повідомлень електронної пошти щодо ознак фішингової атаки»

Виконав: студент 4 курсу групи 1БС-21б
спеціальності 125 Кібербезпека
Володимир РОСТЕЦЬКИЙ

Керівник: к. т. н., доц., доцент каф. ЗІ
Олеся ВОЙТОВИЧ
«16» червня 2025 р.

Рецензент: к. т. н. доц., доцент каф. ПЗ
Володимир МАЙДАНЮК
«16» червня 2025 р.

Допущено до захисту
Завідувач кафедри ЗІ
д. т. н., проф.
Володимир ЛУЖЕЦЬКИЙ
«16» червня 2025 р.

Вінниця ВНТУ – 2025 року

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації
Рівень вищої освіти I (бакалаврський)
Галузь знань – 12 Інформаційні технології
Спеціальність – 125 Кібербезпека
Освітньо-професійна програма – Безпека інформаційних і комунікаційних систем

ЗАТВЕРДЖУЮ
Завідувач кафедри ЗІ, д. т. н., проф.
_____ Володимир ЛУЖЕЦЬКИЙ
«21» березня 2025 року

ЗАВДАННЯ
на бакалаврську кваліфікаційну роботу студенту
Ростецькому Володимиру Богдановичу

1. Тема роботи: «Засіб для перевірки повідомлень електронної пошти щодо ознак фішингової атаки»,
   керівник роботи: Олеся ВОЙТОВИЧ, к. т. н., доцент кафедри ЗІ,
   затверджені наказом ректора ВНТУ від 20 березня 2025 року №97.
2. Строк подання студентом роботи 16 червня 2025 р.
3. Вихідні дані до роботи:
   - Формат додатку – Програмний застосунок
   - Метод перевірки – Перевірка вмісту, адресата та заголовка
   - Мова програмування – Python.
4. Зміст текстової частини: Вступ. Аналіз літературних джерел. Розробка засобу виявлення фішингових атак. Реалізація та тестування засобу. Висновки. Перелік використаних джерел. Додатки.
5. Перелік ілюстративного матеріалу: Результати порівняння існуючих засобів захисту від фішингових атак. Результати порівняння технічних методів захисту від фішингових атак. Основні ознаки фішингових листів. Схема процесу перевірки на ознаки фішингового листа. Структурна схема архітектури модулів. Алгоритм перевірки посилань. Алгоритм аналізу фішингових фраз. Алгоритм класифікації листа за рівнем загроз. Алгоритм взаємодії користувача з програмою. Результат перевірки 100 безпечних листів. Результат перевірки 100 фішингових листів. Результати роботи засобу.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийн
1	Войтович О. П., к.т.н., доц. каф.ЗІ	24.02.25	10.03.25
2	Войтович О. П., к.т.н., доц. каф.ЗІ	14.05.25	20.05.25
3	Войтович О. П., к.т.н., доц. каф.ЗІ	30.05.25	10.06.25

7. Дата видачі завдання 21 березня 2025 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської кваліфікаційної роботи	Строк виконання етапів роботи	Прим
1	Аналіз завдання. Вступ	24.03.25 – 30.03.25	
2	Аналіз літературних за напрямком бакалаврської кваліфікаційної роботи	31.03.25 – 13.04.25	
3	Розробка рішень, моделей, алгоритмів	14.04.25 – 04.05.25	
4	Практична реалізація, моделювання, експериментування, результати	05.05.25 – 14.05.25	
5	Аналіз виконання ПЗ, висновки	15.05.25 – 18.05.25	
6	Оформлення пояснювальної записки	19.05.25 – 26.05.25	
7	Попередній захист БКР	27.05.25 – 30.05.25	
8	Виправлення зауважень, підготовка ілюстративного матеріалу	31.05.25 – 10.06.25	
9	Представлення БКР до захисту, рецензування	11.06.25 – 13.06.25	

Студент _____ Володимир РОСТЕЦЬКИ

Керівник роботи _____ Олеся ВОЙТОВИ

АНОТАЦІЯ

Ростецький Володимир Богданович. Засіб для перевірки поштових повідомлень на ознаки фішингової атаки. Бакалаврська кваліфікаційна робота. Вінниця, ВНТУ, 2025. – 71 ст.

Бакалаврська кваліфікаційна робота складається з 71 сторінки формату А4, на яких є 30 рисунків, 7 таблиць, список використаних джерел містить 56 найменувань.

Бакалаврська кваліфікаційна робота присвячена розробці програмного засобу для автоматизованої перевірки електронних листів на наявність ознак фішингових атак. У ході дослідження проведено детальний аналіз існуючих методів виявлення фішингу – перевірки автентичності відправника за протоколами SPF, DKIM та DMARC, семантичного аналізу тексту, перевірки URL на предмет підробок і сканування вкладень на наявність шкідливого коду – що дозволило відокремити ключові критерії оцінки ризику кожного повідомлення.

Розроблений засіб реалізовано мовою Python із використанням модульної архітектури: окремі компоненти відповідають за інтеграцію з поштовими серверами, аналіз заголовків, контенту, посилань та вкладень, а також за класифікацію повідомлень за категоріями «безпечні», «підозрілі» та «фішингові». Оптимізовані алгоритми та асинхронна обробка даних забезпечують високу швидкодію й гнучкість системи, що робить засіб зручним для використання. Тестування на реальних та змодельованих вибірках підтвердило ефективність запропонованого підходу, що свідчить про доцільність впровадження рішення в практику кіберзахисту.

Ключові слова : фішингові атаки, виявлення фішингу, безпека електронної пошти, перевірка повідомлень, перевірка посилань, перевірка вкладень, текстовий аналіз.

ABSTRACTS

Rostetskyi Volodymyr Bohdanovych. A Tool for Checking Email Messages for Phishing Attack Indicators. Bachelor’s Thesis. Vinnytsia: VNTU, 2025. – 71 pp.

The bachelor's thesis consists of 71 A4 pages, which have 30 figures, 7 tables, the list of sources used contains 56 items.

This bachelor’s thesis is devoted to the development of a software tool for the automated inspection of electronic mail messages to identify signs of phishing attacks. The study involved a comprehensive analysis of existing phishing-detection methods–including sender authentication via SPF, DKIM and DMARC protocols, semantic text analysis, URL spoofing checks, and attachment scanning for malicious code–which made it possible to isolate key risk-assessment criteria for each message.

The proposed tool is implemented in Python using a modular architecture: separate components handle integration with mail servers, header analysis, content inspection, link and attachment verification, and message classification into “safe,” “suspicious,” and “phishing” categories. Optimized algorithms and asynchronous data processing ensure high performance and flexibility, making the solution user-friendly and easy to deploy. Testing on both real and simulated datasets confirmed the effectiveness of the approach, indicating its suitability for integration into practical cybersecurity workflows.

Keywords: phishing attacks, phishing detection, email security, message verification, link verification, attachment verification, text analysis.

ЗМІСТ

ВСТУП ... 3
1 АНАЛІЗ ЛІТЕРАТУРНИХ ДЖЕРЕЛ ... 5
1.1 Аналіз проблеми фішингових листів ... 5
1.2 Аналіз методів та засобів боротьби з фішинговими листами ... 13
1.3 Постановка завдання ... 22
2 РОЗРОБКА ЗАСОБУ ВИЯВЛЕННЯ ФІШИНГОВИХ АТАК ... 25
2.1 Ознаки фішингових атак ... 25
2.2 Обґрунтування вибору методів для реалізації ... 28
2.3 Розробка узагальненої схеми засобу ... 38
2.4 Варіанти реалізації засобу ... 40
3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ЗАСОБУ ... 42
3.1 Обґрунтування вибору мови програмування та середовища розробки ... 42
3.2 Розробка модулів для функціонування засобу ... 43
3.3 Алгоритм функціонування засобу ... 47
3.4 Інтерфейс засобу ... 51
3.5 Експериментальні дослідження ... 54
ВИСНОВКИ ... 64
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ ... 65
ДОДАТКИ ... 72
Додаток А ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ ... 73
Додаток Б ЛІСТИНГ ПРОГРАМИ ... 74
Додаток В ІЛЮСТРАТИВНА ЧАСТИНА ... 83

ВСТУП

Сучасний розвиток інформаційних технологій значно розширив можливості комунікації, проте одночасно створив нові виклики в сфері кібербезпеки. Однією з найпоширеніших та найнебезпечніших загроз сьогодення є фішингові атаки - методи соціальної інженерії, спрямовані на викрадення конфіденційної інформації шляхом надсилання підроблених електронних листів [1]. З кожним роком фішинг стає все більш досконалішим, використовуючи технології штучного інтелекту для створення персоналізованих та замаскованих повідомлень, що значно ускладнює їх виявлення традиційними засобами захисту.

Зростання кількості фішингових атак негативно впливає як на звичайних користувачів, так і на бізнес-структури, що призводить до значних фінансових втрат та втрати репутації. За статистикою, у 2023–2024 роках кількість фішингових листів зросла на понад 20%, а більшість із них містять ознаки автоматизованого створення за допомогою штучного інтелекту [2]. Це підкреслює необхідність розробки ефективних засобів для автоматичної перевірки поштових повідомлень на ознаки фішингових атак, що дозволить своєчасно виявляти загрози та мінімізувати ризики.

Об’єктом бакалаврської роботи є процеси виявлення фішингових листів.

Предметом бакалаврської роботи є методи і засоби виявлення фішингових листів.

Метою дослідження є підвищення ефективності перевірки поштових повідомлень на ознаки фішингових атак шляхом розробки засобу, який аналізує вміст поштових повідомлень, адресата та заголовків.

Для досягнення мети необхідно:

- провести аналіз літературних джерел за темою БКР;
- дослідити існуючі методи та засоби виявлення фішингових листів та виявити основні проблеми та недоліки;
- розробити структуру, модель та алгоритм роботи програмного засобу, який позволить виявляти фішингові атаки у електронних листах.
- розробити програмний засіб та протестувати його.

Результати дослідження засобу перевірки фішингових атак були апробовані на LIV всеукраїнській науково-технічній конференції факультету інформаційних технологій та комп'ютерної інженерії, що проходила у Вінницькому національному технічному університеті (Вінниця, 2025 р.) [3].

1 АНАЛІЗ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

Електронна пошта залишається одним із найпоширеніших і зручних способів обміну інформацією у сучасному світі. Однак разом із широким використанням даного способу комунікації зростає й кількість кіберзагроз, серед яких особливо небезпечними є фішингові атаки.

З огляду на постійне ускладнення методів фішингу, актуальним залишається дослідження існуючих способів виявлення та запобігання таким атакам. Аналіз їх ефективності та розробка нових засобів для автоматичної перевірки поштових повідомлень на ознаки фішингу є важливим напрямом у сфері кібербезпеки.

1.1 Аналіз проблеми фішингових листів

Фішингові листи – це шахрайські електронні повідомлення, які надсилаються з метою викрадення конфіденційної інформації користувачів шляхом обману. Вони маскуються під офіційні листи від відомих організацій, банків або популярних сервісів і можуть містити посилання на підроблені сайти, запити на введення особистих даних або шкідливий вміст. Такі листи часто створюють із високою професійністю, через що навіть досвідченим користувачам важко відрізнити їх від офіційних повідомлень. Фішинг є однією з найпоширеніших форм кіберзлочинності, що постійно розвивається та ускладняється [1].

Основна мета фішингових листів – змусити користувача перейти за шкідливим посиланням, ввести логіни, паролі, банківські реквізити або інші персональні дані, які зловмисники використовують для шахрайства або несанкціонованого доступу до систем. Відсутність достатньої обізнаності користувачів та постійне вдосконалення методів атак роблять фішинг надзвичайно ефективним інструментом кіберзлодіїв. Крім того, масштабність розповсюдження таких листів дозволяє зловмисникам досягати великої кількості потенційних жертв за короткий час, що підсилює рівень загрози [2].

В основі фішингових атак лежить соціальна інженерія – набір психологічних прийомів, спрямованих на маніпулювання поведінкою людини. Зловмисники використовують страх, терміновість, авторитетність джерела або обіцянки вигоди, щоб змусити жертву діяти необачно і не перевіряти достовірність інформації. Наприклад, у листі може міститися попередження про блокування банківського рахунку або повідомлення про виграш призу, що змушує користувача діяти швидко і необдумано.

На рис.1.1 зображено приклад фішингової атаки через SMS-повідомлення.

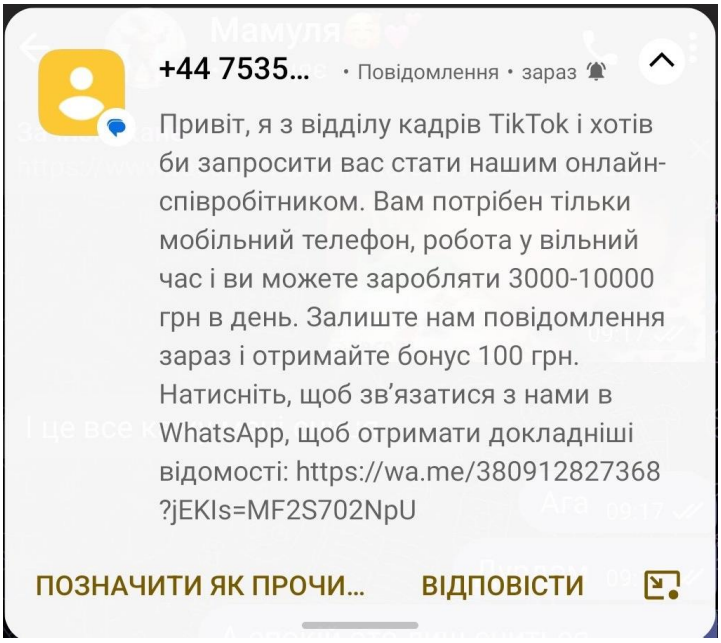


Рисунок 1.1 – Фішингове повідомлення

Дана фішингова атака почала швидко поширюватися навесні 2025 року, зокрема завдяки популярності платформи TikTok. Велика кількість користувачів отримувала подібні повідомлення не лише у вигляді SMS, а й через різноманітні соціальні мережі та месенджери, зокрема TikTok, Viber, Instagram. Такий багатоканальний підхід значно підвищує охоплення жертв і ускладнює їх виявлення, адже шахрайські повідомлення маскуються під офіційні сповіщення різних сервісів у звичних для користувачів середовищах.

Залежно від способу реалізації та цільової аудиторії, фішинг поділяється на велику кількість видів, кожен з яких має свої механізми реалізації, особливості та цілі. Для зручності їх можна умовно розділити на категорії.

1) За каналом поширення:

– Поштовий фішинг (Email phishing) є найпоширенішим типом фішингових атак, в яких зловмисники надсилають масові електронні листи, які імітують повідомлення від відомих компаній, банків або сервісів. У листах містяться посилання на підроблені сайти або шкідливі вкладення, які при відкритті можуть викрасти дані або заразити пристрій. Цей тип часто використовує розсилку великої кількості однакових листів без персоналізації, сподіваючись на випадковий успіх [4].

– Вішинг (Vishing) це фішинг через телефонні звінки. Зловмисники, видаючи себе за співробітників банку, правоохоронних органів чи технічної підтримки, намагаються отримати конфіденційну інформацію [5].

– Смішинг (Smishing) використовує SMS-повідомлення. Зловмисники надсилають текстові повідомлення з проханням перейти за посиланням або зателефонувати на фальшивий номер. Повідомлення можуть містити загрози блокування рахунку або обіцянки виграшу [6].

– Фішинг у соціальних мережах. Зловмисники створюють фальшиві акаунти або зламують справжні профілі, щоб розсилати шкідливі посилання чи виманювати дані через особисті повідомлення [2].

– Календарний фішинг. Шахраї надсилають фішингові запрошення у календарі, які містять посилання на шкідливі ресурси [7].

Для умовного порівняння різних типів фішингових атак використаємо три рівні ризику: високий, середній та низький. Високий рівень ризику означає, що атака може призвести до значних фінансових втрат, витоку конфіденційної інформації або швидкої компрометації облікових записів і потребує негайного реагування. Середній рівень ризику характерний для поширених атак, які можуть спричинити втрату даних або тимчасові проблеми, але їх можна відносно легко виявити і блокувати за допомогою стандартних засобів захисту. Низький рівень ризику властивий менш поширеним або менш ефективним атакам, які рідко призводять до серйозних наслідків, але все ж потребують базового рівня захисту.

У табл. 1.1 наведено короткий опис проаналізованих атак.

Таблиця 1.1 – Аналіз фішингових атак за каналом поширення

Тип атаки	Короткий опис	Канал поширення	Рівень ризику
Поштовий фішинг	Масова розсилка підроблених листів із посиланнями або вкладеннями	Електронна пошта	Середній
Вішинг	Фішинг через телефонні дзвінки, видача себе за довірену особу	Телефон	Високий
Смішинг	Фішинг через SMS із проханням перейти за посиланням або зателефонувати	SMS	Середній
Фішинг у соціальних мережах	Використання фальшивих або зламаних акаунтів для обману	Соціальні мережі	Середній
Календарний фішинг	Фішингові запрошення з посиланнями на шкідливі ресурси	Календарні сервіси	Низький

2) За цільовою аудиторією

– Масовий фішинг (Mass Phishing). Розсилка однакових листів багатьом користувачам без персоналізації, наприклад, зі зверненням «Шановний клієнт». Зловмисники сподіваються, що хоча б частина отримувачів, не перевіряючи деталі, дадуть відповідь на лист і розкриють особисту інформацію, що робить цей метод простим, але досить ефективним для отримання великої кількості даних через великий об’єм листів [2].

– Цільовий фішинг (Spear Phishing). Атака на конкретну особу або організацію. Зловмисник попередньо збирає інформацію про жертву, щоб зробити повідомлення максимально правдоподібним і персоналізованим. Часто використовується для отримання доступу до корпоративних систем або фінансових ресурсів [8].

– Китовий фішинг (Whaling). Підвид цільового фішингу, спрямований на топ-менеджерів, керівників компаній чи інших осіб із високим рівнем доступу до важливої інформації. Повідомлення можуть виглядати як офіційні запити від партнерів, державних органів тощо [9].

У табл. 1.2 наведено короткий опис проаналізованих атак.

Таблиця 1.2 – Аналіз фішингових атак за цільовою аудиторією

Тип атаки	Короткий опис	Цільова аудиторія	Рівень ризику
Масовий фішинг	Розсилка однакових листів або повідомлень великій кількості людей без персоналізації.	Широка аудиторія	Середній
Цільовий фішинг	Атака на конкретну особу або організацію з персоналізованими повідомленнями, підготовленими на основі зібраної інформації.	Конкретні особи або компації	Високий
Китовий фішинг	Підвид цільового фішингу, спрямований на топ-менеджерів, керівників компаній чи осіб із високим рівнем доступу.	Топ-менеджери, керівники компаній	Високий

3) За технікою реалізації

– Фармінг (Pharming). Перенаправлення користувача на підроблену веб-сторінку через маніпуляції з DNS або зараження пристрою. Навіть якщо користувач вводить правильну адресу, він потрапляє на фішинговий сайт, де вводить свої дані, які потім потрапляють до зловмисників [10].

– Клон-фішинг (Clone Phishing). Зловмисник створює копію офіційного листа, змінюючи посилання або вкладення на шкідливі. Такий лист може виглядати як повторне надсилання важливого повідомлення [11].

– Фальшиві веб-посилання (Deceptive Phishing). Використання підроблених або дуже схожих на справжні URL-адреси, скорочених посилань або доменів із помилками, щоб змусити жертву не помітити підміну та ввести свої дані на фальшивому сайті [12].

– Angler Phishing. Використання підроблених акаунтів у соціальних мережах або службах підтримки, які імітують офіційні сторінки компаній. Зловмисники заманюють жертв у спілкування, щоб отримати довіру та змусити надати конфіденційну інформацію, перейти за фальшивими посиланнями або відкрити заражений файл [13] .

У табл. 1.3 наведено короткий опис проаналізованих атак.

Таблиця 1.3 – Аналіз фішингових атак за цільовою аудиторією

Тип атаки	Короткий опис	Особливості реалізації	Рівень ризику
Фармінг	Перенаправлення користувача на підроблену веб-сторінку через зміну DNS або зараження пристрою.	Користувач вводить правильну адресу, але потрапляє на фальшивий сайт.	Високий
Клон-фішинг	Створення копії офіційного листа з заміною посилань або вкладень на шкідливі.	Лист виглядає як повторне надсилання важливого повідомлення.	Середній
Фальшиві веб-посилання	Використання підроблених або схожих URL, скорочених посилань або доменів із помилками.	Маскування адрес для обману користувача.	Середній
Angel Phishing	Використання підроблених акаунтів у соцмережах або службах підтримки, що імітують офіційні сторінки.	Залучення жертви до спілкування для отримання довіри.	Середній

4) За цілями атак

– Крадіжка особистих даних. Мета полягає у викраденні логінів, паролів, номерів кредитних карток та інших конфіденційних даних, які використовуються для несанкціонованого доступу до акаунтів чи ідентифікації жертви [14].

– Отримання фінансової вигоди. Злочинці спрямовують зусилля на безпосереднє заволодіння коштами жертв через шахрайські транзакції, перекази або використання викрадених платіжних реквізитів [14].

– Встановлення шкідливого ПЗ. Фішингові повідомлення містять посилання або вкладення, що інсталюють на пристрій жертви шкідливі програми для подальшого контролю чи викрадення даних [15].

– Шпигунство. Отримання несанкціонованого доступу до конфіденційної корпоративної інформації, комерційних секретів, технологічних розробок або стратегічних планів. Зловмисники використовують фішингові атаки для збору цінних даних, які можуть бути використані конкурентами або продані на чорному ринку [16].

Згідно проаналізованих атак, найбільш популярним каналом поширення фішингу є електронна пошта через легку реалізацію, недостатній рівень захисту та несвідомість користувачів.

Фішинг залишається однією з найпоширеніших і найнебезпечніших форм кіберзлочинності, що постійно зростає як за кількістю атак, так і за масштабами завданих збитків. Щодня у світі надсилається понад 3,4 мільярда шкідливих електронних листів, що містять фішингові атаки [17]. При цьому Google блокує близько 100 мільйонів таких листів щодня, що свідчить про масштабність і постійну активність кіберзлочинців [18].

Зростання кількості фішингових атак у 2023 році склало понад 40% у порівнянні з 2022 роком. Цей сплеск пов’язаний із сезонними факторами, наприклад, початком літніх відпусток, коли користувачі активніше користуються інтернетом для бронювання подорожей [19]. Крім того, зловмисники розширюють канали поширення атак, активно використовуючи месенджери, платформи зі штучним інтелектом, соціальні мережі та криптобіржі [20].

В Україні ситуація також залишається напруженою. За даними Держспецзв’язку, у 2024 році кількість кібератак на український кіберпростір зросла майже на 70% у порівнянні з 2023 роком і досягла 4315 інцидентів. Серед найпоширеніших типів атак – фішинг, розповсюдження шкідливого програмного забезпечення, компрометація облікових записів та систем. Основними цілями зловмисників є державні установи, енергетичний сектор, комерційні організації та телекомунікації [21].

На рис.1.2 зображено статистику кількості унікальних фішингових атак по всьому світу за останні кілька років [22].

Рисунок 1.2 – Статистика унікальних фішингових атак за останні кілька років

Цю інформацію було опубліковано групою по боротьбі з фішингом (AWPG) наприкінці 2024 року [22]. Дані на графіку чітко демонструють стрімке зростання кількості фішингових атак у світі з 2014 по 2023 рік, із піком у 2023-му. Незважаючи на незначне зниження у 2024 році, загальний рівень залишається надзвичайно високим, що свідчить про постійну актуальність цієї кіберзагрози та необхідність впровадження сучасних заходів захисту.

Еволюція фішингових атак демонструє поступове ускладнення методів злочинців. Відомі масштабні кампанії минулих років, як-от злам RSA у 2011 році [2], коли кіберзлочинці змогли обійти двофакторну автентифікацію та викрасти корпоративні дані, або злив даних Sony Pictures у 2014 році, що спричинило збитки понад 100 мільйонів доларів, свідчать про високий рівень організації і технічної складності атак [23].

У 2013–2015 роках Google і Facebook втратили близько 100 мільйонів доларів через фішингові афери, де зловмисники надсилали підроблені інвойси співробітникам компаній [2]. Це показує, що навіть найбільші технологічні гіганти не застраховані від фішингових атак.

З 2021 року LinkedIn став ідеальною платформою для фішингових атак, на нього припадає понад половину виявлених інцидентів, випереджаючи Facebook і Twitter. Це пов’язано з професійною орієнтацією LinkedIn і великою кількістю

корпоративних користувачів, що робить платформу привабливою для зловмисників [24].

Кількість великих фішингових атак зростає не лише кількісно, але й якісно. За перші півроку 2023 року було зафіксовано більше великих інцидентів, ніж за весь 2020 рік, що свідчить про зростаючу активність і складність атак [22].

Варто також відзначити, що з 2018 року кількість атак шкідливого програмного забезпечення зменшується, що пов’язано з покращенням засобів захисту і зміною тактики кіберзлочинців. Водночас фішинг набирає обертів, оскільки він поєднує технічні та психологічні методи обману, що робить його ефективним і відносно простим у реалізації [22].

Прогнози на найближчі роки передбачають подальше зростання кількості фішингових атак, особливо з використанням штучного інтелекту для персоналізації повідомлень і підвищення їх переконливості. Це створює додаткові виклики для систем захисту, які повинні адаптуватися до нових загроз у реальному часі [25].

Таким чином, статистика і еволюція фішингових атак демонструють, що ця форма кіберзлочинності залишається однією з найактивніших і найнебезпечніших. Постійне зростання кількості атак, збільшення їх складності та використання новітніх технологій зловмисниками вимагають комплексного підходу до протидії, що включає технічні заходи, підвищення обізнаності користувачів і розвиток законодавства.

1.2 Аналіз методів та засобів боротьби з фішинговими листами

У зв’язку з активним зростанням фішингових атак розроблено велику кількість програмних засобів, спрямованих на виявлення таких загроз в електронній пошті. Ці рішення відрізняються за функціональністю, способом інтеграції та цільовою аудиторією. Проаналізовано та представлено найпоширеніші інструменти, які використовуються для захисту користувачів від фішингових повідомлень.

Microsoft Defender for Office 365 є корпоративним засобом безпеки, вбудованим у хмарну платформу Microsoft 365, який забезпечує багатоетапний захист електронної пошти. Основними механізмами є перевірка SPF, DKIM і DMARC для виявлення підміни адрес, аналіз фішингових URL-адрес, динамічне сканування вкладень у sandbox-середовищі, а також поведінковий аналіз для виявлення атак типу BEC (Business Email Compromise). Програма інтегрується безпосередньо з Exchange Online і надає адміністраторам розширені можливості моніторингу, звітності та реагування. Обмеженням є прив'язка до екосистеми Microsoft, що унеможливлює її використання з іншими платформами [26].

Google Workspace Email Protection – це вбудований у Gmail механізм захисту електронної пошти, який використовує алгоритми штучного інтелекту Google для виявлення фішингових загроз. Захист охоплює перевірку автентичності відправника, аналіз вкладень на наявність шкідливого вмісту, виявлення фішингових посилань, а також визначення шаблонів небажаної поведінки. Завдяки повній інтеграції у середовище Google Workspace, захист активується автоматично та не потребує додаткової конфігурації. Водночас система не підтримує інші поштові сервіси, окрім Gmail [27].

Proofpoint Essentials є одним із провідних рішень для безпеки електронної пошти у малому та середньому бізнесі. Цей шлюз фільтрації працює як проміжний рівень між зовнішнім поштовим трафіком і корпоративною поштовою системою, перевіряючи кожен лист на відповідність політикам безпеки. Програма виконує перевірку SPF, DKIM та DMARC, сканує вкладення й URL-адреси, ідентифікує фішингові шаблони, а також реалізує виявлення BEC-атак на основі поведінкових аномалій. Підтримка Microsoft 365, Gmail і локальних поштових серверів робить її універсальним рішенням для гібридних середовищ [28].

Mimecast Email Security – це комплексний шлюз безпеки електронної пошти, що поєднує фільтрацію контенту, захист бренду, перевірку вкладень та URL-адрес у реальному часі. Програма виявляє фішингові повідомлення за допомогою аналізу шаблонів, спуфінгу, QR-кодів, а також sandbox-механізмів для динамічного аналізу вмісту. Mimecast підтримує інтеграцію як із хмарними, так і з локальними

сервісами електронної пошти, що забезпечує високу гнучкість у застосуванні. Завдяки широкому набору функцій і стабільній інфраструктурі, рішення підходить як для середнього бізнесу, так і для великих корпоративних клієнтів [29].

IRONSCALES – це автоматизована платформа захисту поштових скриньок, яка працює безпосередньо на стороні користувача, аналізуючи листи вже після їх надходження в inbox. Програма використовує самообучувані алгоритми для виявлення фішингових атак, включаючи підміну адрес, фальшиві форми, URL та поведінкові шаблони. Особливістю є підтримка інтерактивного реагування: користувачі можуть вручну позначати фішинг, а система на цій основі покращує свої моделі. Платформа інтегрується з Gmail і Microsoft 365, але не підтримує інші поштові сервіси [30].

Barracuda Email Protection – хмарне рішення для захисту пошти, що забезпечує багатошарову перевірку листів на предмет фішингу, шкідливих вкладень та атак із підміною особи. Засіб підтримує аналіз URL, sandbox-сканування вкладень, виявлення шаблонів атак та інтеграцію з механізмами антиспаму. Завдяки підтримці всіх основних поштових платформ, включаючи локальні сервери, Gmail і Microsoft 365, Barracuda є універсальним вибором для організацій з різною інфраструктурою [31].

Abnormal Security – це сучасна хмарна система захисту поштового середовища, яка зосереджується на поведінковому аналізі вхідної пошти. Вона вивчає звичні шаблони комунікації кожного користувача й виявляє аномальні повідомлення, що можуть бути ознакою BEC або соціальної інженерії. Платформа виявляє підміну адрес, фішингові шаблони, відхилення в структурі повідомлень. Водночас інструмент орієнтований переважно на корпоративних користувачів і працює лише з хмарними платформами Microsoft 365 і Google Workspace [32].

Rspamd – це безкоштовний поштовий фільтр з відкритим кодом, який інтегрується у локальні поштові сервери (Postfix, Exim, Sendmail) як Milter-фільтр. Засіб підтримує перевірку SPF і DKIM, аналіз URL-адрес, використання байєсівських моделей та евристик для виявлення фішингових шаблонів. Завдяки високій продуктивності та гнучкому конфігуруванню, Rspamd підходить для

серверів з великим обсягом поштового трафіку та дає змогу налаштувати політики фільтрації відповідно до потреб організації [33].

Проаналізовані програмні засоби демонструють різний рівень ефективності та доступності, проте більшість з них мають певні обмеження. Серед основних недоліків виявлено недостатній функціонал, відсутність підтримки усіх поштових сервісів або ж надмірно високу вартість, що ускладнює їх використання для індивідуальних користувачів чи малих організацій.

У таблиці 1.4 описані проаналізовані існуючі засоби.

Таблиця 1.4 – Результати порівняння існуючих засобів захисту від фішингових атак

Засіб	**Опис**	**Методи перевірки**	**Переваги**	**Недоліки**
Defender for Office 365	Корпоративна безпека в Microfost 365	SPF, DKIM, DMARC, аналіз посилань та вкладень	Інтеграція з M365, звітність	Працює лише з M365
Google Workspace Email Protection	Вбудований у Gmail захист із ШІ-аналітикою	SPF, DKIM, аналіз вкладень і URL, ШІ-аналіз поведінки	Автоматична активація, простота	Тільки для Gmail
Proofpoint Essentials	Шлюз фільтрації між зовнішнім трафіком і поштою	SPF, DKIM, DMARC, сканування вкладень та посилань	Підтримка великої кількості поштових сервісів	Платний зовнішній шлюз
Mimecast Email Security	Комплексна платформа з контент-фільтрацією	Фільтрація контенту, перевірка посилань та вкладень, QR-аналіз	Підтримує хмару та локальні сервери	Висока вартість
IRON-SCALES	Платформа для аналізу повідомлень із самонавчальними моделями	Аналіз підміни адрес, фальшивих форм і URL, поведінковий аналіз	Адаптивні моделі, інтерактивність	Лише для M365 та Gmail
Barracuda Email Protection	Хмарне рішення з антифішингом і антиспамом	Перевірка посилань та вкладень, антиспам	Підтримка всіх популярних платформ	Залежить від Barracuda Cloud
Abnormal Security	Платформа для перевірки повідомлень штучним інтелектом	Поведінковий аналіз, шаблонний аналіз	Висока точність у виявленні BEC	Тільки для cloud-платформ
Rspamd	Безкоштовний Milter-фільтр для локальних SMTP-серверів	SPF, DKIM, аналіз URL	Безкоштовний, гнучке налаштування	Потребує ручної конфігурації

Фішингові атаки залишаються однією з найпоширеніших і найнебезпечніших кіберзагроз, що щоденно впливають як на приватних користувачів, так і на бізнес. Ефективний захист від фішингу вимагає комплексного підходу, який поєднує технічні засоби, організаційні заходи та навчання персоналу. З огляду на стрімке зростання кількості фішингових інцидентів в Україні та світі, впровадження надійних методів захисту є критично важливим для збереження конфіденційності даних та безпеки інформаційних систем.

Технічні засоби, такі як багатофакторна автентифікація (MFA), антивірусне програмне забезпечення з антифішинговими модулями, поштові фільтри та системи аутентифікації електронної пошти (SPF, DKIM, DMARC), забезпечують перший рівень оборони, автоматично виявляючи і блокуючи більшість фішингових листів. Однак людський фактор залишається вразливим місцем, тому навчання користувачів і регулярні тренінги зі зростанням обізнаності про фішинг є невід'ємною частиною ефективної стратегії захисту.

Організаційні заходи, включаючи впровадження політик безпеки, централізований моніторинг інцидентів і багаторівневі системи захисту, допомагають знизити ризики та оперативно реагувати на загрози. Сучасні рішення дозволяють не лише виявляти атаки, а й залучати користувачів до їхнього блокування, підвищуючи загальний рівень безпеки.

1.2.1 Технічні методи захисту

Технічні засоби є основою ефективного захисту від фішингових атак, оскільки дозволяють автоматично виявляти, блокувати та запобігати проникненню шкідливих листів і сайтів. Серед найпоширеніших і найефективніших методів варто виділити кілька ключових.

Перш за все, двофакторна автентифікація (2FA) або багатофакторна автентифікація (MFA) додає додатковий рівень безпеки. Навіть якщо зловмисник отримує пароль користувача, без другого фактора (наприклад, одноразового коду з мобільного додатка або SMS) доступ до акаунту залишається заблокованим. Це один із найбільш надійних способів знизити ризики компрометації [17].

Важливу роль відіграють антивірусні програми з антифішинговими модулями. Сучасні рішення, такі як Bitdefender, ESET, які мають вбудовані механізми, які автоматично сканують вхідні листи й посилання, блокуючи відомі фішингові ресурси. Регулярне оновлення таких програм дозволяє враховувати нові загрози [17].

Поштові фільтри – ще один ефективний інструмент. Сервіси Google Workspace, Microsoft 365 та інші мають вбудовані фільтри, що аналізують вхідні повідомлення, виявляючи підозрілі листи і зменшуючи ймовірність їх потрапляння до користувача. Вони використовують алгоритми машинного навчання для виявлення нових типів атак [26].

Для перевірки оригінальності листів застосовуються стандарти аутентифікації електронної пошти – SPF, DKIM, DMARC. Вони допомагають виявляти підроблені адреси відправників і блокувати листи, які не відповідають політикам безпеки домену.

SPF (Sender Policy Framework) - це протокол, який дозволяє власникам доменів визначати, які сервери мають право надсилати листи від імені їхнього домену. Для цього в DNS домену створюється спеціальний TXT-запис із переліком дозволених IP-адрес поштових серверів. Коли поштовий сервер отримує лист, він перевіряє IP-адресу відправника і порівнює її зі списком у SPF-записі. Якщо IP не входить до списку, лист може бути відхилений або позначений як спам [27].

DKIM (DomainKeys Identified Mail) додає до листа цифровий підпис, який гарантує цілісність повідомлення і підтверджує, що воно дійсно надійшло від власника домену. Відправник підписує лист приватним ключем, а отримувач перевіряє підпис за допомогою публічного ключа, розміщеного в DNS. Якщо підпис співпадає, лист вважається автентичним і незмінним [28].

DMARC (Domain-based Message Authentication, Reporting & Conformance) об’єднує SPF і DKIM, встановлюючи політику, як обробляти листи, що не пройшли перевірку. Власник домену задає правила: не робити нічого (none), відправляти в спам (quarantine) або відхиляти (reject). Після перевірки SPF і DKIM сервер

отримувача дивиться політику DMARC і приймає рішення, що робити з листом, а також надсилає звіти власнику домену про результати перевірок [28].

Ще одним корисним технічним засобом є браузерні розширення та інструменти для перевірки URL-адрес. Вони автоматично попереджають користувача про потенційно небезпечні сайти, що значно знижує ризик переходу на фішингові ресурси [29].

Також варто звернути увагу на менеджери паролів, які не лише створюють надійні унікальні паролі, а й не автозаповнюють форми на підозрілих сайтах, що допомагає уникнути випадкового введення даних на фішингових сторінках [30].

Регулярне оновлення операційних систем, браузерів та програмного забезпечення є важливим, оскільки багато фішингових атак використовують відомі вразливості, які виправляються оновленнями.

Для корпоративного захисту застосовуються також системи моніторингу доменів і мережевого трафіку (наприклад, Cisco Umbrella), які відстежують активність підозрілих ресурсів і блокують доступ до них [39].

У сучасних умовах активно впроваджуються рішення на основі штучного інтелекту і машинного навчання, які аналізують поведінку користувачів і аномалії трафіку, що дозволяє виявляти навіть нові, раніше невідомі фішингові загрози [40].

У таблиці 1.5 наведено коротке порівняння технічних методів захисту від фішингових атак.

Таблиця 1.5 – Порівняннях технічних методів захисту фішингових атак

Метод захисту	**Принцип роботи**	**Переваги**	**Недоліки**
Двофакторна автентифікація (2FA/MFA)	Додатковий рівень підтвердження (код, апаратний ключ)	Дуже високий рівень захисту навіть при викраденні пароля	Потребує додаткових дій від користувача
Антивірус з антифішингом	Сканування листів і посилань, блокування відомих загроз	Простота використання, регулярні оновлення	Може пропускати нові або складні цільові атаки
Поштові фільтри	Аналізують вхідні листи, блокують підозрілі повідомлення	Зменшують кількість фішингових листів, використовують машинне навчання	Можливі помилкові спрацьовування, залежність від налаштувань

Продовження табл.1.5

SPF	Перевірка IP відправника за списком у DNS	Захищає від підробки адреси відправника (спуфінгу)	Не працює при пересиланні, не захищає вміст листа
DKIM	Додає цифровий підпис для підтвердження цілісності листа	Гарантує, що лист не змінено і надійшов від власника домену	Вимагає налаштування ключів, не працює при зміні листа
DMARC	Встановлює політику обробки листів, що не пройшли SPF/DKIM	Дозволяє контролювати підроблені листи, отримувати звіти	Потрібна правильна конфігурація SPF і DKIM, ризик блокування офіційних листів
Браузерні розширення та URL-фільтри	Попереджають про небезпечні сайти, блокують переходи	Знижують ризик переходу на фішингові ресурси	Не гарантують повний захист, можуть блокувати офіційні сайти
Менеджери паролів	Генерують і зберігають складні паролі, не автозаповнюють форми на підозрілих сайтах	Підвищують кібергігієну, знижують ризик компрометації паролів	Не захищають від фішингу напряму
Моніторинг доменів і мережевого трафіку	Відстеження підозрілих доменів і блокування доступу	Проактивний захист на рівні мережі, виявлення нових загроз	Вартість, потреба у кваліфікованому адмініструванні
Антифішингові платформи на основі ШІ	Використання машинного навчання для виявлення нових і складних атак	Висока точність, автоматизація реагування	Висока вартість, потреба у налаштуванні і підтримці

1.2.2 Організаційні заходи

Організаційні заходи є невід'ємною складовою комплексного захисту від фішингових атак, оскільки навіть найсучасніші технічні рішення не здатні повністю усунути людський фактор - основну вразливість у системі безпеки. Вони включають впровадження політик безпеки, навчання персоналу, моніторинг і реагування на інциденти.

Перш за все, важливо розробити і впровадити чіткі політики безпеки, які регламентують порядок роботи з електронною поштою, правила доступу до інформаційних систем, а також процедури реагування на підозрілі листи. Політика має передбачати обмеження доступу до критично важливої інформації на основі ролей співробітників, а також вимоги щодо використання безпечних протоколів зв'язку (наприклад, HTTPS) [41].

Одним із ключових напрямів є навчання і підвищення обізнаності співробітників. Регулярні тренінги, семінари та симуляції фішингових атак допомагають персоналу розпізнавати підозрілі листи та уникати необачних дій. За даними Microsoft, обізнаність користувачів є найкращим способом захисту від фішингу, оскільки більшість атак базуються на соціальній інженерії.

Для оперативного виявлення і мінімізації наслідків атак необхідно впроваджувати системи моніторингу та реагування на інциденти. Централізований збір інформації про загрози, автоматичне оновлення захисних систем і координація дій між IT-відділом, службою безпеки та керівництвом дозволяють швидко локалізувати інциденти і зменшувати їхній вплив.

Важливо також використовувати інструменти багаторівневого захисту, які поєднують фільтрацію спаму, перевірку безпечності вебсайтів, блокування підозрілих посилань і автоматичне інформування користувачів. Наприклад, сервіси Office 365, Cisco Umbrella та спеціалізовані рішення на кшталт PhishMe допомагають знизити ризики успішних фішингових атак.

Не менш важливо мати план реагування на інциденти, який включає етапи ідентифікації загрози, ізоляції скомпрометованих систем, інформування відповідальних осіб і правоохоронних органів, а також відновлення нормальної роботи. Такий план дозволяє мінімізувати збитки і швидко повернути контроль над ситуацією.

1.2.3 Заходи підвищення обізнаності

Освітні та поведінкові методи є критично важливими у боротьбі з фішинговими атаками, оскільки людський фактор залишається однією з головних вразливостей у системах безпеки. Навчання користувачів допомагає розпізнавати ознаки фішингу, уникати необачних дій і своєчасно реагувати на загрози.

Регулярні тренінги для співробітників організацій допомагають підвищити їхню обізнаність і сформувати навички критичного мислення. Важливо навчати розпізнавати підозрілі листи за такими ознаками, як підроблені адреси, граматичні помилки, тиск на швидку реакцію, підозрілі посилання та вкладення. Працівники мають чітко розуміти правила безпечної роботи з електронною поштою: не

відкривати сумнівні вкладення, не переходити за неперевіреними посиланнями, користуватися офіційними каналами для підтвердження запитів [42].

Проведення симуляцій фішингових атак дозволяє практично перевірити готовність персоналу і виявити слабкі місця у знаннях. Такі тренінги підвищують пильність і знижують ризик успішних атак.

Крім того, важливо формувати культуру кібергігієни серед усіх користувачів - від школярів до дорослих. Навчальні матеріали, інтерактивні курси і просвітницькі кампанії допомагають розвивати навички безпечної поведінки в інтернеті і відповідальності за власну кібербезпеку.

1.3 Постановка завдання

Після аналізу сучасних джерел і досліджень у сфері кібербезпеки, зокрема щодо фішингових атак через електронну пошту, стає очевидним, що ця проблема залишається однією з найактуальніших і найсерйозніших загроз для користувачів і організацій. З кожним роком кількість фішингових повідомлень зростає, а методи їх реалізації стають дедалі витонченішими. Зловмисники активно використовують різноманітні техніки – від масових розсилок до цільових атак, застосовуючи соціальну інженерію, підробку адрес, шкідливі посилання та вкладення. Це ускладнює процес виявлення таких листів і знижує ефективність традиційних засобів захисту.

Існуючі технології, такі як SPF, DKIM і DMARC, а також популярні антивірусні та антиспам-фільтри, хоч і забезпечують певний рівень безпеки, не завжди здатні своєчасно і точно виявити всі види фішингових загроз. Особливо це стосується новітніх методів атак, які використовують штучний інтелект, персоналізовані повідомлення та складні схеми маскування. Крім того, відсутність зручних і доступних інструментів для автоматичної перевірки поштових повідомлень підвищує ризик того, що користувачі стануть жертвами шахрайства.

Враховуючи ці виклики, виникає потреба у створенні ефективного засобу, який би дозволяв автоматизувати процес аналізу електронних листів, виявляти

ознаки фішингових атак і надавати користувачам зрозумілі рекомендації щодо безпечної взаємодії з поштовими повідомленнями. Такий засіб має бути адаптивним, здатним враховувати різноманітні параметри листів – від перевірки автентичності відправника до аналізу посилань, вкладень і текстового вмісту, а також легко інтегруватися з існуючими поштовими сервісами.

Отже, основною метою цієї роботи є розробка програмного засобу для перевірки поштових повідомлень на ознаки фішингової атаки, який підвищить рівень захисту користувачів і організацій від кіберзагроз, знизить ризики компрометації даних і фінансових втрат.

Перший етап розробки полягає у формулюванні чітких вимог до системи на основі проведеного аналізу джерел і вивчення предметної області. На цьому етапі визначаються основні функціональні можливості, які має забезпечувати засіб, а також технічні та експлуатаційні характеристики. Особлива увага приділяється вибору критеріїв для виявлення фішингових ознак, визначенню форматів підтримуваних поштових повідомлень, вимогам до сумісності з різними поштовими сервісами і протоколами, а також до продуктивності і зручності користування. Важливо також врахувати потенційні обмеження та особливості впровадження системи в реальних умовах, що дозволить уникнути проблем на наступних етапах розробки.

Наступним кроком є проектування архітектури системи. На цьому етапі розробляється загальна структура програмного засобу, яка включає ключові компоненти: модуль інтеграції з поштовими серверами, модуль аналізу та виявлення фішингових ознак, модуль класифікації повідомлень, а також інтерфейс. Особлива увага приділяється забезпеченню масштабованості системи, її гнучкості та можливості подальшого розширення функціоналу.

Далі розробляється модуль експорту та інтеграції поштових повідомлень, який відповідає за підключення до поштових серверів, аутентифікацію користувача та отримання списку листів. Цей модуль забезпечує збір необхідної інформації для подальшого аналізу, включаючи тему листа, адресу відправника, дату надходження

та розмір. Коректна робота цього модуля є критичною для забезпечення своєчасного і повного доступу до поштових повідомлень.

Ключовим етапом є реалізація модуля виявлення фішингових ознак. Цей компонент здійснює глибокий аналіз кожного листа, перевіряючи автентичність відправника за допомогою протоколів SPF, DKIM, DMARC, аналізує URL-адреси на предмет підробок, перенаправлень або шкідливих посилань, виявляє потенційно небезпечні вкладення. Крім того, застосовується семантичний аналіз текстового вмісту листа для виявлення ознак соціальної інженерії, таких як маніпулятивні фрази, заклики до термінових дій або підроблені підписи.

Після цього розробляється модуль аналізу та класифікації, який на основі результатів роботи модуля виявлення формує остаточний висновок щодо рівня ризику кожного листа. Цей модуль класифікує повідомлення за категоріями – безпечні, підозрілі або фішингові – і виконує дії відповідно до категорії листа.

Паралельно з цим розробляється інтерфейс, який має бути інтуїтивно зрозумілим і зручним для користувачів різного рівня підготовки. Інтерфейс повинен надавати швидкий доступ до результатів перевірки, чітко інформувати про потенційні загрози і пропонувати рекомендації щодо подальших дій. Важливо забезпечити можливість налаштування рівня деталізації інформації та інтеграції з поштовими клієнтами або корпоративними системами.

Завершальним етапом є тестування та перевірка розробленого засобу. Проводиться комплексне тестування на реальних і змодельованих даних, що дозволяє оцінити точність виявлення фішингових листів, кількість хибних спрацьовувань, продуктивність і стабільність роботи системи. Результати тестування аналізуються для виявлення слабких місць і подальшого вдосконалення алгоритмів. Важливим аспектом є також перевірка сумісності з різними поштовими сервісами та оцінка зручності користувацького інтерфейсу.

Такий поетапний підхід до розробки забезпечує створення надійного, ефективного і зручного у використанні засобу, який зможе ефективно протистояти сучасним фішинговим загрозам і значно підвищити рівень захисту користувачів електронної пошти.

2 РОЗРОБКА ЗАСОБУ ВИЯВЛЕННЯ ФІШИНГОВИХ АТАК

Виявлення фішингових листів є надзвичайно важливим завданням у сфері кібербезпеки, яке забезпечує захист користувачів та організацій від несанкціонованого доступу, фінансових втрат і компрометації конфіденційної інформації. З огляду на постійне ускладнення методів фішингових атак, розробка та вдосконалення ефективних методів і засобів для автоматичного виявлення таких загроз стає першочерговим завданням для забезпечення інформаційної безпеки.

У даному розділі буде розглянуто різні підходи та алгоритми, які дозволяють аналізувати поштові повідомлення на предмет фішингових ознак. Особлива увага приділяється методам перевірки автентичності відправника, аналізу структури листа, перевірці посилань і вкладень, а також застосуванню правил і евристик для підвищення точності виявлення загроз. На основі цього аналізу буде розроблена узагальнена структура засобу, що відповідає сучасним вимогам безпеки та зручності використання.

2.1 Ознаки фішингових атак

Фішингові атаки, що здійснюються через електронну пошту, є однією з найбільш поширених і небезпечних форм кіберзлочинності, яка продовжує еволюціонувати. Однією з ключових задач у боротьбі з фішингом є виявлення фішингових листів на ранньому етапі, що дозволяє запобігти небажаним наслідкам. Існують кілька типових ознак, які можуть свідчити про наявність фішингової атаки, і розпізнавання цих ознак є важливим для ефективного захисту.

Однією з перших ознак фішингових листів є підозріла або неправдоподібна адреса відправника. Зловмисники часто використовують адреси, які виглядають схожими на офіційні, але мають незначні відмінності. Наприклад, адреса може бути написана з помилкою, додаванням зайвих символів або зміненою літерою, що робить її схожою на офіційну, але не зовсім правильною [43]. Така підробка може бути складною для виявлення, особливо коли відправник маскується під відомий

бренд чи організацію. Для виявлення таких підробок використовуються модулі перевірки автентичності відправника, зокрема реалізації протоколів SPF, DKIM та DMARC, які аналізують відповідність IP-адреси відправника і домену, а також перевіряють цифрові підписи листів. Якщо адреса не проходить ці перевірки або домен викликає підозру, система підвищує ризик листа, що може призвести до його маркування як фішингового або автоматичного блокування.

Ще однією характерною ознакою фішингових листів є наявність нетипових або неправильних граматичних конструкцій, що часто свідчить про непрофесійність відправника. Офіційні компанії, як правило, ретельно перевіряють свої повідомлення на наявність граматичних і стилістичних помилок, тому їхня присутність у листі є серйозним індикатором потенційної загрози. Для виявлення таких ознак застосовуються модулі обробки природної мови (NLP), які аналізують структуру тексту, виявляють орфографічні та синтаксичні помилки, а також оцінюють стилістичну відповідність повідомлення офіційним стандартам [44]. Відсутність персоналізації, наприклад, звернення на ім’я одержувача, також фіксується цими модулями як додатковий фактор ризику. У разі виявлення подібних аномалій система підвищує ймовірність класифікації листа як фішингового, що може призводити до його маркування, блокування або інформування користувача.

Однією з найбільш тривожних ознак фішингового листа є запит на введення конфіденційної інформації, оскільки офіційні компанії ніколи не вимагають від користувачів надсилати паролі, номери кредитних карток або інші чутливі дані через електронну пошту. Для виявлення цієї ознаки використовуються модулі контентного аналізу, які сканують текст листа на наявність ключових фраз і запитів, що стосуються особистих або фінансових даних, а також аналізують посилання, що містяться у повідомленні. Якщо система виявляє такі запити, особливо якщо вони супроводжуються посиланнями на підозрілі або фальшиві веб-сайти, це суттєво підвищує ризик класифікації листа як фішингового. Вплив цієї ознаки на прийняття рішення є критичним: листи з подібними запитами зазвичай блокуються або позначаються з відповідним попередженням для користувача, оскільки їхня мета -

викрадення конфіденційної інформації, що може призвести до значних фінансових втрат і шахрайства.

Іншою важливою ознакою фішингових атак є наявність у листах підозрілих посилань, які часто маскуються під офіційні URL, але містять незначні відмінності, що можуть бути складними для виявлення. Зловмисники використовують схожі доменні імена або змінюють окремі символи в адресах, щоб перенаправити користувачів на фальшиві вебсайти, де відбувається крадіжка особистих даних [45]. Для автоматичного виявлення таких посилань застосовуються модулі перевірки URL, які аналізують структуру адреси, порівнюють її з базами відомих фішингових ресурсів і оцінюють репутацію домену. Якщо посилання виявляється підозрілим або невідповідним офіційному сайту, система підвищує ризик повідомлення, що може призвести до блокування листа або відображення попередження для користувача.

Використання тактики терміновості або погроз є однією з найпоширеніших і найефективніших психологічних методик у фішингових атаках. Зловмисники свідомо створюють у повідомленнях відчуття нагальної необхідності негайних дій, наприклад, повідомляють, що акаунт користувача буде заблоковано протягом 24 годин або що оплата не пройшла і потрібно терміново оновити дані. Такий підхід спрямований на те, щоб викликати у жертви паніку або страх втрати доступу до важливих ресурсів, що змушує її діяти імпульсивно, не перевіряючи достовірність інформації та ігноруючи інші підозрілі ознаки листа. Для виявлення цієї ознаки застосовуються модулі семантичного аналізу тексту, які ідентифікують ключові слова та фрази, що створюють тиск терміновості або погроз, а також оцінюють тональність повідомлення. Виявлення таких елементів підвищує ризик класифікації листа як фішингового, що впливає на рішення системи про його блокування або інформування користувача з метою запобігання необдуманим діям.

Фішингові листи часто містять вкладення зі шкідливим програмним забезпеченням, які маскуються під звичні документи, такі як рахунки чи квитанції, але насправді можуть містити віруси, троянські програми або інші види шкідливого ПЗ. Відкриття таких вкладень призводить до зараження комп’ютера, що дає

зловмисникам можливість викрадати конфіденційну інформацію, контролювати систему або здійснювати інші шкідливі дії. Для виявлення таких загроз використовуються антивірусні сканери(наприклад, VirusTotal), які автоматично перевіряють вкладення на наявність шкідливого коду [46]. У разі виявлення небезпеки лист блокується, а користувач отримує відповідне попередження.

У боротьбі з фішинговими атаками, важливо вчасно розпізнати характерні ознаки підозрілих листів. У таблиці 2.1 наведені основні проаналізовані ознаки фішингових листів.

Таблиця 2.1 – Основні ознаки фішингових листів

Ознака	**Короткий опис**	**Метод виявлення**
Підозріла адреса	Адреса містить помилки чи змінені символи	SPF/DKIM/DMARC; перевірка IP-домену
Граматичні помилки та відсутність персоналізації	Орфографічні/стилістичні помилки, загальне звернення	NLP-аналіз помилок; виявлення імені
Запит конфіденційної інформації	Заклики надати паролі чи дані карток	Контент-аналіз; пошук ключових фраз; перевірка URL
Підозрілі посилання	URL із незначними змінами, що ведуть на фейкові сайти	Аналіз структури URL; база фішингових ресурсів; перевірка URL
Приховані посилання	Текст посилання не відповідає фактичному посиланню	Перевірка тексту проти href
Невідповідність тексту та посилання	Посилання не відповідає тексту в листі	Порівняння тексту на посиланню
Терміновість або погрози	Фрази з погрозами або терміновими закликами	Пошук ключових слів
Вкладення з шкідливим програмним забезпеченням	Вкладення містять шкідливе ПЗ	Антивірусне сканування
Замасковане вкладення	Вкладення має подвійне або хибне розширення	Антивірусне сканування; перевірка розширення

2.2 Обґрунтування вибору методів для реалізації

Для розробки засобу виявлення фішингових листів було проведено ретельний аналіз існуючих методів, алгоритмів, структур та засобів, які застосовуються для ідентифікації фішингових загроз у електронній пошті. Вибір конкретних рішень

базувався на їхній ефективності, простоті реалізації, можливості інтеграції з поштовими сервісами, а також здатності адаптуватися до сучасних технік фішингу. Особливу увагу приділено поєднанню класичних підходів, таких як перевірка SPF, з сучасними методами аналізу вмісту листів і поведінковими ознаками, що дозволяє підвищити точність виявлення фішингових атак і знизити кількість хибних спрацьовувань.

2.2.1 Перевірка автентичності відправника за допомогою SPF, DKIM, DMARC

Автентифікація електронної пошти – це комплекс заходів, які дозволяють переконатися, що вхідний лист дійсно надійшов від заявленого домену, а не є підробкою або фішинговою атакою. Автентифікація електронної пошти базується на комплексі протоколів, серед яких SPF, DKIM і DMARC є ключовими для забезпечення достовірності відправника та захисту від підробки листів.

SPF (Sender Policy Framework) - це протокол, який дає можливість власнику домену визначити, які сервери мають право надсилати електронні листи від імені цього домену. Це реалізується через спеціальний запис у DNS, де вказуються IP-адреси або діапазони адрес серверів, які вважаються довіреними відправниками. Коли поштовий сервер отримує лист, він перевіряє IP-адресу сервера-відправника, порівнюючи її з вказаними у SPF-записі. Якщо IP-адреса співпадає з дозволеними, лист вважається авторизованим. Якщо ні – лист може бути відхилений або позначений як потенційно шкідливий [27].

SPF значно підвищує рівень безпеки електронної пошти, обмежуючи коло серверів, які можуть надсилати листи від імені домену, що зменшує ризик спуфінгу – підробки адреси відправника [47]. Проте варто враховувати, що SPF не є універсальним рішенням, оскільки він перевіряє лише IP-адресу сервера-відправника і не гарантує захисту від підробки заголовка «From», який відображається користувачу. Крім того, при пересиланні листів через проміжні сервери IP-адреса може змінюватися, що призводить до некоректної роботи SPF і потенційного блокування легітимної кореспонденції.

Важливо також зазначити, що правильне налаштування SPF-запису вимагає ретельного планування та регулярного оновлення. Невірно сформований SPF-запис може призвести до помилкових відхилень або втрати важливої кореспонденції. Крім того, деякі поштові сервери можуть ігнорувати або по-різному інтерпретувати результати SPF-перевірки, тому важливо впроваджувати SPF у рамках комплексної стратегії безпеки електронної пошти.

На рис. 2.1 зображено схему роботи SPF.

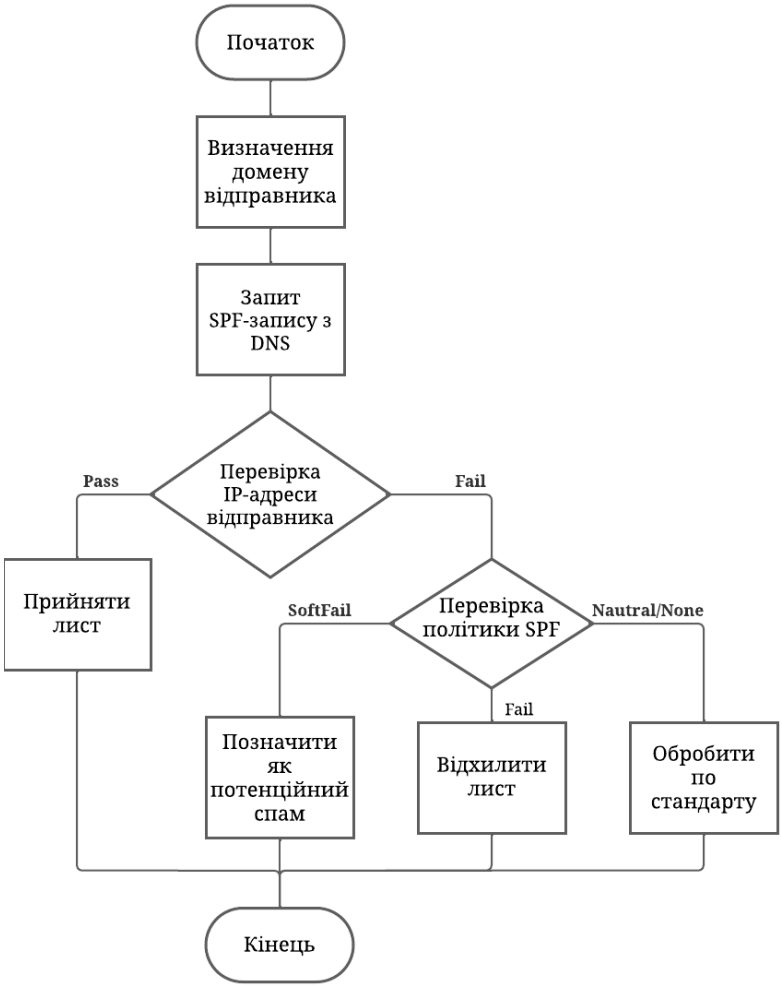


Рисунок 2.1 – Схема роботи SPF

DKIM (DomainKeys Identified Mail) – це протокол, який додає до листа цифровий підпис, що дозволяє підтвердити його цілісність і походження. Він забезпечує захист електронної пошти від підробки, фішингу та спаму, дозволяючи поштовим серверам одержувачів перевірити, що лист надійшов від заявленого домену і не був змінений під час передачі. Для цього у DNS домену відправника

публікується відкритий ключ, який використовується для перевірки підпису. Наявність валідного DKIM-підпису дозволяє поштовим серверам підтвердити достовірність відправника і зберегти цілісність повідомлення, що позитивно впливає на репутацію домену та підвищує ймовірність доставки листа.

На рис. 2.2 зображено схему роботи підписання DKIM [28].

Рисунок 2.2 – Схема роботи підписання DKIM

Під час відправлення електронного листа сервер відправника створює цифровий підпис, обираючи певні поля повідомлення (наприклад, тіло листа, заголовки) і генеруючи хеш-рядок із їхнього вмісту. Цей хеш шифрується за допомогою закритого криптографічного ключа, який зберігається лише у відправника, і додається до заголовка листа у вигляді DKIM-підпису. Одночасно у DNS домену відправника публікується відповідний відкритий ключ, який використовується для перевірки підпису.

Поштовий сервер одержувача отримує лист із DKIM-підписом і витягує з DNS відкритий ключ домену відправника. Потім він розшифровує підпис за допомогою цього ключа, отримуючи хеш-рядок, і паралельно генерує власний хеш із відповідних полів листа. Якщо обидва хеші співпадають, це означає, що лист не

був змінений під час передачі разі невідповідності підпис вважається недійсним, і лист може бути відхилений або позначений як підозрілий. На рис. 2.3 зображено схему роботи перевірка DKIM.

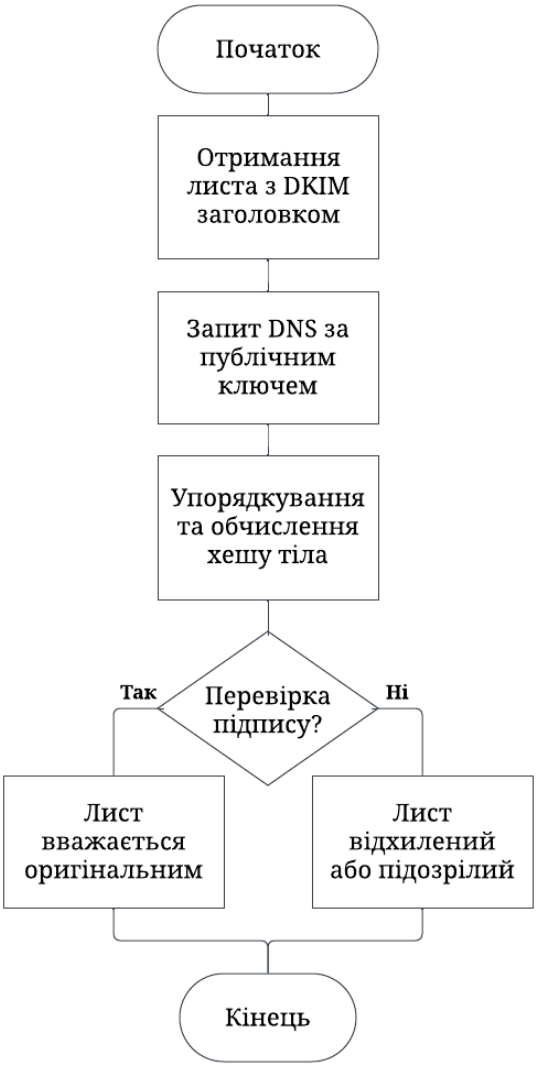


Рисунок 2.3 – Схема роботи перевірки DKIM

Цей механізм забезпечує додатковий рівень захисту, оскільки навіть якщо хтось спробує змінити вміст листа або підробити його, перевірка DKIM виявить таку зміну і позначить лист як підозрілий. DKIM також допомагає підвищити довіру до листів. Проте DKIM вимагає правильної настройки і підтримки, а також може бути недійсним, якщо лист змінюється проміжними серверами (наприклад, додаються рекламні банери).

DMARC (Domain-based Message Authentication, Reporting and Conformance) – це протокол, який працює на основі SPF і DKIM і дозволяє власнику домену

визначити політику обробки листів, які не проходять перевірки автентичності. DMARC встановлюється через DNS-запис і вказує, чи слід приймати, поміщати в карантин або відхиляти такі листи. Крім того, DMARC забезпечує механізми звітування, що дозволяє власникам доменів отримувати інформацію про спроби підробки їхніх листів і аналізувати джерела загроз [28].

Цей протокол дає змогу не лише автоматично захищати домен від шахрайських листів, але й отримувати статистику та аналітику, що допомагає у виявленні і реагуванні на атаки. DMARC вимагає, щоб SPF і DKIM були налаштовані коректно, і часто впроваджується поступово – спочатку у режимі моніторингу, а потім із суворішими політиками. На рис. 2.4 зображено схему роботи DMARC.

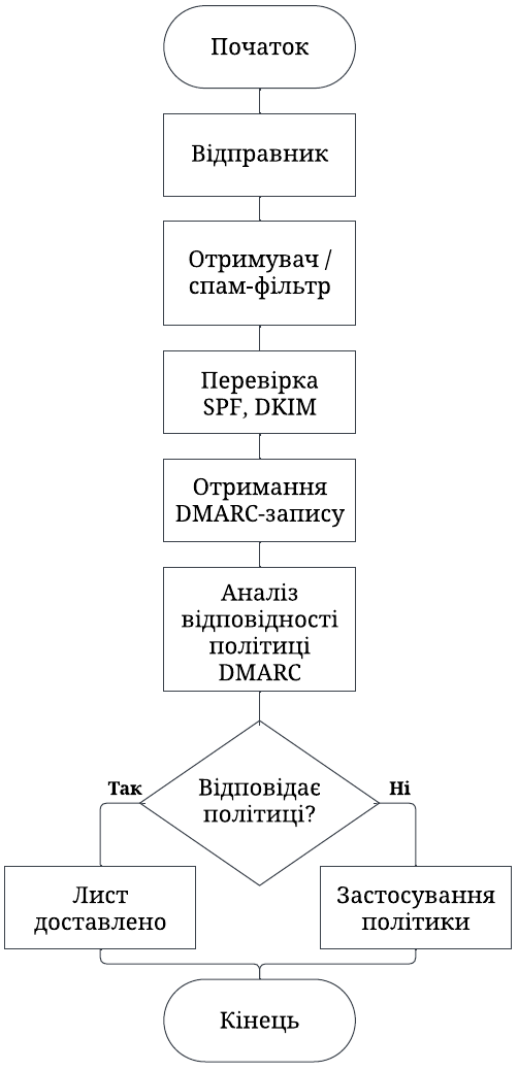


Рисунок 2.4 – Схема роботи DMARC

У комплексі протоколи SPF, DKIM і DMARC створюють багаторівневий захист електронної пошти. Коли лист надходить на поштовий сервер або у систему перевірки, спочатку визначається IP-адреса сервера, що надіслав лист, і здійснюється перевірка SPF – чи входить ця IP-адреса до списку дозволених для домену відправника. Паралельно перевіряється цифровий підпис DKIM, щоб переконатися в цілісності листа і його справжності. Потім аналізується політика DMARC домену відправника, яка визначає, що робити з листами, що не пройшли SPF і DKIM або мають невідповідність доменів у заголовках. Залежно від політики DMARC лист може бути прийнятий, відхилений або поміщений у карантин. Власник домену отримує звіти про всі невдалі спроби автентифікації, що дозволяє відслідковувати і реагувати на потенційні загрози. Ця система допомагає значно знизити кількість підроблених листів і фішингових атак, підвищити довіру до електронної пошти і покращити якість доставки офіційних повідомлень.

Впровадження SPF, DKIM і DMARC є стандартом для сучасних поштових сервісів і організацій, які прагнуть захистити свої домени і користувачів від шахрайства. Для ефективної роботи цих протоколів необхідно правильно налаштувати DNS-записи SPF, DKIM і DMARC, постійно моніторити звіти DMARC і коригувати політики. Політику DMARC рекомендується поступово посилювати – від режиму моніторингу до відхилення підроблених листів. Важливо враховувати особливості пересилання листів і можливі проблеми з доставкою. Загалом, ці протоколи створюють надійний фундамент для захисту електронної пошти, роблячи її більш безпечною і довіреною для користувачів і організацій.

2.2.2 Аналіз текстового вмісту на типові фішингові фрази і заклики до термінових дій

Фішингові листи часто містять стандартні фрази, які створюють у користувача відчуття терміновості або загрози, наприклад: «Терміново оновіть пароль», «Ваш акаунт буде заблоковано», «Підтвердіть особисті дані» та інші подібні заклики. Аналіз текстового вмісту листа базується на наборі правил і шаблонів, які дозволяють виявити ці типові фрази та оцінити їхню частоту і контекст використання.

У процесі роботи система витягує текст листа і здійснює пошук ключових слів і фраз, характерних для фішингових повідомлень. Далі оцінюється, наскільки часто і в якому контексті ці фрази використовуються, що допомагає сформувати рейтинг підозрілості листа. Лист із великою кількістю таких фраз або з контекстом, що викликає занепокоєння, позначається як потенційно фішинговий і може бути додатково перевірений або заблокований.

Цей метод аналізу допомагає виявити багато видів фішингових атак, оскільки шахраї часто використовують психологічний тиск і термінові заклики, щоб змусити користувачів діяти необачно. Використання правил і шаблонів для пошуку таких фраз є ефективним інструментом захисту від шахрайства в електронній пошті.

2.2.3 Перевірка IP-адрес в DNS Blacklist

Перевірка IP-адрес відправника за допомогою DNSBL (DNS Blacklist) є важливим механізмом захисту від фішингових та спам-листів. DNSBL – це спеціалізовані бази даних, які містять IP-адреси серверів, помічених у розсилці небажаних або шкідливих повідомлень. Під час отримання листа поштовий сервер автоматично звертається до цих списків, щоб визначити, чи не входить IP-адреса відправника до чорного списку.

Процедура починається з ідентифікації IP-адреси сервера, що надіслав лист, після чого вона перевіряється у кількох авторитетних DNSBL. Ці бази даних постійно оновлюються, адже нові загрози з’являються щодня. Важливо враховувати не лише факт наявності IP у чорному списку, а й кількість таких збігів, що може свідчити про активну шкідливу діяльність з цієї адреси.

Якщо IP-адреса виявляється у кількох DNSBL, поштовий сервер підвищує рівень підозри щодо листа, що може призвести до його блокування, маркування як спам або відправлення на додаткову перевірку. Для підвищення оперативності та точності перевірок сучасні системи інтегрують API популярних DNSBL-сервісів, таких як Spamhaus, Barracuda чи SpamCop, що дозволяє миттєво реагувати на нові загрози.

Загалом, використання DNSBL для перевірки IP-адрес є одним із найефективніших способів зменшити кількість фішингових і спам-листів,

підвищуючи безпеку електронної пошти та захищаючи користувачів від потенційних атак.

2.2.4 Перевірка посилань у листі

Фішингові листи часто містять посилання, які на перший погляд виглядають офіційно, але фактично ведуть на шкідливі або підроблені сайти, створені для викрадення особистих даних, фінансової інформації або зараження пристроїв шкідливим програмним забезпеченням. Тому ретельний аналіз усіх посилань у листі є критично важливим для виявлення потенційних загроз і запобігання фішинговим атакам.

Процес перевірки починається з автоматичного витягання всіх URL-адрес із тексту повідомлення. Далі виконується порівняння доменів цих посилань із базами даних відомих фішингових і шкідливих ресурсів, які регулярно оновлюються. Особливу увагу приділяють порівнянню видимого тексту посилання, який бачить користувач, і фактичної URL-адреси, на яку воно веде. Якщо ці адреси не співпадають, або якщо посилання є, це викликає підозру і вимагає додаткової перевірки [45].

Скорочені посилання особливо небезпечні, оскільки вони приховують справжню адресу, що ускладнює їх ідентифікацію. Для їх аналізу застосовують спеціальні сервіси або API, які розкривають кінцевий URL і дозволяють оцінити його безпеку. Також звертають увагу на нетипові або підозрілі домени, які можуть містити помилки в написанні, зайві символи або імітувати відомі бренди (метод typosquatting).

Інтеграція таких перевірок у засоби захисту може здійснюватися через API популярних сервісів безпеки, які надають інформацію про репутацію доменів і URL. Наприклад, VirusTotal, Google Safe Browsing, PhishTank та інші пропонують API для автоматизованої перевірки посилань у реальному часі. Це дозволяє оперативно виявляти і блокувати шкідливі URL, навіть якщо вони з’явилися недавно [45].

Якщо в процесі аналізу виявляються аномалії або збіги з чорними списками, лист позначається як підозрілий і може бути заблокований або направлений на

додаткову перевірку. Такий багаторівневий підхід значно підвищує ефективність захисту від фішингових атак, оскільки більшість шахрайських повідомлень базуються саме на обмані через підроблені або приховані URL.

Таким чином, аналіз посилань у листах із використанням сучасних технологій і API є одним із найефективніших способів виявлення та запобігання фішинговим атакам, що значно підвищує загальний рівень безпеки електронної пошти.

2.2.5 Аналіз вкладень у листі

Вкладення у електронних листах можуть містити різноманітні файли, серед яких часто трапляються шкідливі програми, віруси, трояни або скрипти, які можуть завдати шкоди комп'ютеру користувача або викрасти конфіденційну інформацію. Тому аналіз вкладень є важливою складовою системи захисту від фішингу та інших кіберзагроз.

Перший етап перевірки полягає у визначенні типу файлів, що додаються до листа. Особливу увагу приділяють потенційно небезпечним форматам, таким як .exe, .js, .scr, .bat, .vbs та іншим виконуваним або скриптовим файлам, які можуть містити шкідливий код. Такі вкладення зазвичай блокуються або поміщаються у карантин.

Далі для більш глибокого аналізу використовують інтеграцію з антивірусними сканерами та системами виявлення шкідливого програмного забезпечення. Ці інструменти виконують перевірку вмісту файлів на наявність відомих вірусів, троянів, руткітів, програм-вимагачів та інших загроз. У разі виявлення шкідливого коду лист позначається як небезпечний і може бути автоматично заблокований або відправлений на додатковий аналіз.

Інтеграція таких перевірок у поштові системи або спеціалізовані засоби безпеки часто здійснюється через API провайдерів антивірусних сервісів, що дозволяє автоматизувати процес і забезпечити оперативний захист користувачів. Наприклад, популярні сервіси, як VirusTotal, пропонують API для сканування файлів і отримання детальних звітів про їхній стан [45].

Таким чином, комплексний аналіз вкладень, що поєднує визначення типу файлів і антивірусного сканування є ключовим елементом у запобіганні поширенню шкідливого ПЗ через електронну пошту і забезпечує додатковий рівень безпеки для користувачів і організацій.

2.3 Розробка узагальненої схеми засобу

Узагальнена структура засобу для перевірки поштових повідомлень на ознаки фішингової атаки передбачає поетапну обробку кожного листа, що надходить у систему. Такий підхід дозволяє забезпечити комплексний аналіз як технічних, так і змістовних характеристик повідомлення, а також оперативно реагувати на виявлені загрози.

Перший етап – отримання вхідного поштового повідомлення. На цьому кроці система автоматично приймає електронний лист, який надходить у поштову скриньку користувача або організації. Відразу після отримання листа він передається на обробку для проведення всіх необхідних перевірок.

Далі здійснюється аналіз заголовків повідомлення. Цей етап є одним із найважливіших, оскільки саме заголовки містять інформацію про відправника, маршрутизацію листа, проміжні сервери та інші технічні деталі. Аналіз заголовків дозволяє виявити підробки адреси відправника (спуфінг) та невідповідності між адресою у полі «From» та фактичним шляхом доставки.

Наступний етап – аналіз вмісту листа. Тут система досліджує текст повідомлення на наявність типових фішингових фраз, закликів до термінових дій, психологічних прийомів впливу, а також перевіряє вкладення та гіперпосилання. Особлива увага приділяється пошуку підозрілих вкладень (наприклад, виконуваних файлів, документів із макросами) та прихованих або скорочених посилань, які можуть вести на шкідливі ресурси.

Після цього всі знайдені у листі домени, IP-адреси та URL-посилання проходять перевірку з базами фішингових адрес. Для цього використовуються актуальні чорні списки, які містять інформацію про відомі фішингові та шкідливі

ресурси. Така перевірка дозволяє оперативно виявити листи, що містять посилання на небезпечні сайти, навіть якщо текст листа виглядає правдоподібно.

Далі система переходить до оцінки рівня фішингової атаки. На цьому етапі всі зібрані дані – результати аналізу заголовків, вмісту, перевірки вкладень і посилань – обробляються за допомогою спеціальних алгоритмів. Система визначає, наскільки лист є підозрілим, небезпечним чи безпечним, і присвоює йому відповідний статус.

Після визначення статусу відбувається позначення повідомлення як підозріле, безпечне або шкідливе. Це дозволяє користувачеві або адміністратору швидко зорієнтуватися щодо рівня ризику, який несе даний лист.

Далі формується звіт для користувача який відображений у спеціальному інтерфейсі У разі необхідності, якщо лист визначено як підозрілий або шкідливий, система може автоматично застосувати фільтрацію або блокування повідомлення. Це дозволяє запобігти потраплянню фішингових листів до основної поштової скриньки, мінімізувати ризик інфікування пристроїв чи викрадення особистих даних.

На рис. 2.5 зображено схему процесу перевірки на ознаки фішингового листа.

Рисунок 2.5 – Схема процесу перевірки на ознаки фішингового листа

Завдяки такій структурі забезпечується багаторівневий та комплексний підхід до аналізу електронної пошти. Кожен етап є важливим для виявлення різних типів фішингових атак, а їх послідовне виконання дозволяє значно підвищити рівень безпеки користувачів. Модульність такої системи дає змогу легко розширювати функціонал, інтегрувати нові методи аналізу та адаптувати засіб до сучасних кіберзагроз.

2.4 Варіанти реалізації засобу

Існує багато варіантів реалізації засобу виявлення фішингових листів, кожен із яких має свої особливості, переваги та виклики. Фішинг – це одна з найпоширеніших загроз у кібербезпеці, що постійно розвивається і ускладнюється, тому системи захисту мають бути гнучкими, ефективними та здатними швидко адаптуватися до нових методів атак. Для цього застосовують різні підходи, які відрізняються за місцем обробки пошти, способом взаємодії з користувачем, рівнем інтеграції з інфраструктурою та можливостями оновлення і навчання моделей.

– Застосунок з офлайн роботою це програмний засіб, який встановлюється безпосередньо на комп'ютер користувача і працює автономно, без постійного підключення до Інтернету. Основна функція такого застосунку – аналіз локально збережених електронних листів. Він розбирає структуру повідомлень, перевіряє тексти, посилання та вкладення на наявність ознак фішингу, використовуючи внутрішні алгоритми та бази даних. Оновлення правил виявлення загроз можуть здійснюватися під час тимчасового підключення до мережі, але в основному програма функціонує незалежно. Такий підхід забезпечує повний контроль над обробкою інформації, оскільки всі дані залишаються на пристрої користувача. Проте для отримання нових листів застосунок потребує мережевого підключення, тому він не може працювати з поштою в режимі реального часу.

– Застосунок з онлайн роботою це локальна програма, яка регулярно або постійно підключається до Інтернету для взаємодії з поштовими серверами. Вона завантажує нові повідомлення через протоколи IMAP або POP3, проводить їх аналіз на предмет фішингових ознак і може виконувати активні дії, такі як видалення листів, позначення їх як безпечних або спам, безпосередньо на сервері. Вся обробка відбувається локально, що дозволяє швидко реагувати на загрози, а синхронізація з сервером забезпечує актуальність даних. Такий застосунок має власний інтерфейс, де користувач може налаштовувати параметри, переглядати результати перевірок і

керувати поштою. Цей варіант поєднує автономність локальної обробки з можливістю онлайн-взаємодії, що робить його гнучким і ефективним.

– Плагін або модуль для поштового сервера інтегрується безпосередньо у поштову інфраструктуру організації. Він працює на рівні сервера, перехоплюючи всі вхідні повідомлення ще до їх доставки користувачам. Модуль виконує аналіз у режимі реального часу, застосовуючи різні методи виявлення фішингу. Після оцінки ризиків модуль може автоматично блокувати небезпечні листи, позначати їх або дозволяти доставку. Такий підхід забезпечує централізований контроль і захист на рівні всієї організації, знижуючи навантаження на кінцеві пристрої і підвищуючи загальний рівень безпеки.

– Плагін для поштового клієнта це розширення, яке встановлюється у поштовий клієнт користувача, наприклад Outlook або Thunderbird. Воно отримує доступ до поштових повідомлень через внутрішні API клієнта, що дозволяє аналізувати листи безпосередньо в середовищі програми. Плагін виконує перевірку вмісту листів, посилань і вкладень, відображає попередження про потенційні загрози і надає користувачу інструменти для додаткового аналізу. Такий компонент працює локально, інтегруючись у звичний інтерфейс поштового клієнта, що підвищує зручність і оперативність реагування на фішингові атаки.

– Розширення для веб-браузера це додаток, який інтегрується у веб-браузер і працює з веб-інтерфейсами поштових сервісів, таких як Gmail, Yahoo Mail та інші. Розширення перехоплює вміст сторінок, на яких користувач переглядає пошту, і аналізує тексти листів, посилання та вкладення на предмет фішингових ознак. Воно відображає попередження і рекомендації безпосередньо у браузері, допомагаючи користувачу уникнути небезпечних дій. Розширення працює у контексті веб-сторінки, не потребує встановлення окремого поштового клієнта і може звертатися до зовнішніх сервісів для додаткової перевірки. Це робить його зручним інструментом для захисту під час роботи з веб-поштою.

3 РЕАЛІЗАЦІЯ ТА ТЕСТУВАННЯ ЗАСОБУ

Розробка програмного засобу для виявлення фішингових листів є важливим напрямом у забезпеченні кібербезпеки, особливо з огляду на зростаючу кількість та складність фішингових атак у сучасному цифровому середовищі. Ефективні інструменти автоматичного виявлення допомагають своєчасно ідентифікувати загрози, мінімізуючи ризики викрадення особистих і фінансових даних користувачів.

Програмний засіб, що розробляється, спрямований на комплексний аналіз електронних листів із застосуванням сучасних методів перевірки автентичності, аналізу вмісту та відправника. Його створення дозволить підвищити рівень захисту користувачів, зменшити кількість успішних фішингових атак і сприятиме подальшому розвитку технологій кібербезпеки.

У цьому розділі буде детально описано процес розробки, вибір технологій, а також тестування і впровадження запропонованого рішення.

3.1 Обґрунтування вибору мови програмування та середовища розробки

Для розробки програмного засобу виявлення фішингових листів було обрано мову програмування Python. Цей вибір базується на широких можливостях мови у сфері обробки текстів, роботи з мережевими протоколами та інтеграції з численними зовнішніми API. Python має великий набір бібліотек, що значно спрощують роботу з електронною поштою (наприклад, imaplib, email), а також реалізацію перевірок автентичності повідомлень (бібліотеки DKIM, SPF). Окрім зручності та багатофункціональності, Python має активну спільноту розробників, що забезпечує постійне оновлення та підтримку необхідних інструментів і бібліотек. Завдяки цьому Python є ідеальним інструментом для швидкої розробки гнучких та масштабованих рішень у сфері кібербезпеки [48].

В якості середовища розробки було обрано PyCharm – одну з найпопулярніших інтегрованих середовищ розробки (IDE) для Python. PyCharm надає широкий спектр інструментів, які значно підвищують продуктивність розробника: інтелектуальне автодоповнення коду, зручний відладчик, інтеграція з системами контролю версій (Git, Mercurial), можливість роботи з віртуальними середовищами та пакетними менеджерами. Це дозволяє ефективно управляти залежностями, швидко знаходити та виправляти помилки, а також підтримувати якість коду на високому рівні [49].

Використання PyCharm також спрощує командну роботу над проєктом, оскільки середовище підтримує інструменти для кодування за стандартами PEP8, рефакторинг та інші функції, що допомагають підтримувати код чистим і зрозумілим. Завдяки цим можливостям розробка, тестування та подальше вдосконалення програмного засобу проходять ефективно і з мінімальними витратами часу.

Поєдання мови програмування Python та середовища PyCharm забезпечує оптимальний баланс між швидкістю розробки, гнучкістю реалізації та можливістю масштабування. Це дозволяє створити надійний і сучасний інструмент для виявлення фішингових листів, що відповідає вимогам сучасної кібербезпеки та здатен адаптуватися до нових викликів у цій сфері.

3.2 Розробка модулів для функціонування засобу

У цьому розділі представлено опис основних модулів, які складають структуру розробленого програмного засобу. Кожен модуль виконує свою ключову роль у забезпеченні функціональності системи та відповідає за окремі аспекти її роботи. Розглянуті компоненти включають модуль ініціалізації програми, графічний інтерфейс користувача, моніторинг електронної пошти, взаємодію із зовнішніми сервісами та редагування фішингових фраз. Такий поділ дозволяє чітко окреслити відповідальність кожного модуля і створити основу для подальшого детального вивчення алгоритмів та взаємодії між ними.

Структурна схема взаємодії модулів між собою зображена на рисунку 3.1

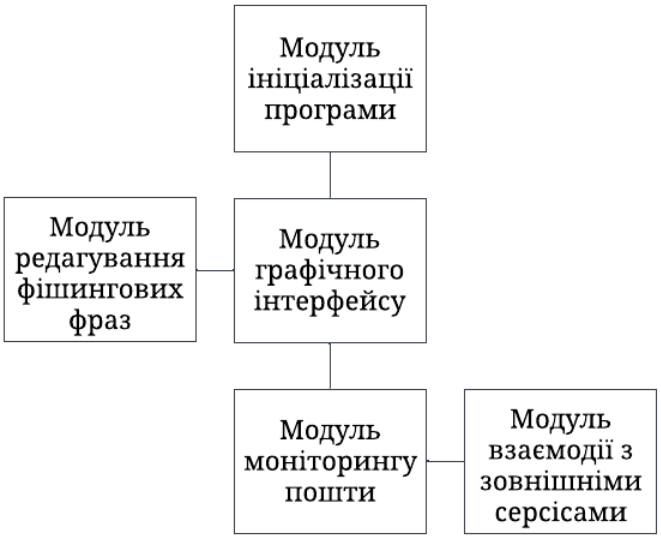


Рисунок 3.1 – Структурна схема архітектури модулів

3.2.1 Модуль ініціалізації програми

Даний модуль відповідає за старт роботи всього програмного засобу. Його основна функція полягає у запуску головного графічного інтерфейсу користувача, який забезпечує подальшу взаємодію з користувачем. Цей модуль не містить складної логіки або алгоритмів, а виконує роль точки входу, ініціалізуючи основні компоненти системи.

Під час запуску модуль створює екземпляр головного вікна програми та запускає його основний цикл подій, що дозволяє відображати інтерфейс і реагувати на дії користувача.

3.2.2 Модуль графічного інтерфейсу

Модуль графічного інтерфейсу користувача реалізує візуальну оболонку програми, що забезпечує зручну та інтуїтивно зрозумілу взаємодію користувача із системою. Для створення інтерфейсу використовується стандартна бібліотека Python – tkinter [50], яка містить набір класів графічних компонентів (віджетів), таких як кнопки, мітки, поля введення, таблиці та інші елементи керування.

Робота модуля починається зі створення головного вікна програми – базового контейнера для всіх елементів інтерфейсу. Далі у вікно додаються вкладки, які логічно розділяють функціонал на блоки: керування поштовими акаунтами,

налаштування методів захисту, перегляд результатів моніторингу та журналу подій. Кожна вкладка містить відповідні віджети, які дозволяють вводити дані, запускати процеси, переглядати статистику та вести журнал.

Для розміщення елементів у вікні застосовуються менеджери розташування, що дають змогу гнучко контролювати позицію і розмір віджетів.

Щоб уникнути блокування інтерфейсу під час виконання тривалих операцій (наприклад, моніторингу пошти), у модулі використовується стандартний модуль threading [51], який запускає ці процеси у фонових потоках. Завдяки цьому користувач може одночасно працювати з інтерфейсом і спостерігати за ходом перевірок у реальному часі.

Модуль також відповідає за збереження та оновлення списку фішингових фраз, що редагуються користувачем, з подальшим записом у локальний файл. Це забезпечує збереження налаштувань між сесіями роботи програми.

3.2.3 Модуль моніторингу електронної пошти

Цей модуль ключовим компонентом системи, який відповідає за безпосередню взаємодію з поштовими серверами та аналіз вхідних повідомлень. Його робота починається із встановлення захищеного з’єднання з поштовим сервером через протокол IMAP, що реалізується за допомогою стандартної бібліотеки Python – imaplib [52]. Цей протокол дозволяє працювати з поштовими скриньками без необхідності завантажувати всі листи на локальний пристрій, зберігаючи їх на сервері.

Після встановлення з’єднання модуль виконує авторизацію користувача, використовуючи надані облікові дані. Для забезпечення безпеки з’єднання застосовується SSL/TLS шифрування. Далі відбувається вибір папки для моніторингу, зазвичай це папка INBOX, яка містить нові вхідні листи.

Для виявлення нових повідомлень модуль періодично виконує пошук непрочитаних листів за допомогою IMAP-команди search з параметром UNSEEN. Важливо, що для унікальної ідентифікації листів використовується їх UID – незмінний ідентифікатор, що дозволяє відстежувати повідомлення навіть при зміні

порядку в папці. Це забезпечує коректну обробку нових листів і запобігає повторній обробці вже перевірених.

Отримані листи розбираються за допомогою стандартного модуля email, який дозволяє аналізувати структуру повідомлення, отримувати заголовки, текстові частини та вкладення. Для перевірки автентичності листів застосовуються бібліотеки dkim та spf, які аналізують відповідні заголовки та політики безпеки, що допомагає виявити підроблені або шкідливі повідомлення.

Модуль також виконує перевірку IP-адреси відправника через DNS-запити до чорних списків (DNSBL) з використанням бібліотеки dnspython [53]. Це дозволяє оперативно ідентифікувати потенційні загрози на основі репутації відправника.

Для додаткового аналізу вмісту листа модуль інтегрується з функціями зовнішніх сервісів, які перевіряють URL-адреси та вкладення на наявність шкідливого коду або фішингових посилань. Крім того, здійснюється пошук фішингових фраз у тексті листа, що дає змогу виявляти підозрілі повідомлення навіть за відсутності інших явних ознак загрози.

У разі виявлення небезпечного або підозрілого листа модуль може автоматично видалити його або позначити для подальшого розгляду. Для забезпечення стабільної роботи передбачено збереження інформації про вже оброблені листи, а також механізми автоматичного відновлення з’єднання у разі його втрати.

3.2.4 Модуль взаємодії з зовнішніми сервісами

Модуль взаємодії з зовнішніми сервісами відповідає за інтеграцію програмного засобу з різноманітними сторонніми API, які надають інформацію про безпеку URL-адрес, вкладень і IP-адрес. Для реалізації цієї взаємодії використовується бібліотека requests, що забезпечує надійне надсилання HTTP-запитів та отримання відповідей від сервісів, таких як VirusTotal, Google Safe Browsing та OpenPhish.

Першим кроком у роботі модуля є завантаження та кешування актуального списку фішингових URL з OpenPhish, що дозволяє зменшити кількість запитів і підвищити швидкодію. Для перевірки URL-адрес модуль надсилає запити до API

VirusTotal та Google Safe Browsing, отримує результати аналізу і визначає, чи є адреса безпечною. Аналогічно здійснюється перевірка вкладень, які надсилаються на сканування VirusTotal.

Перевірка IP-адрес виконується через DNS-запити до чорних списків (DNSBL), для чого використовуються стандартні бібліотеки socket і dnspython. Це дозволяє оперативно визначати репутацію відправника листа.

Модуль також реалізує кешування результатів перевірок за допомогою декоратора lru_cache з модуля functools [54], що знижує навантаження на мережу і пришвидшує повторні звернення. У разі виникнення помилок під час взаємодії з зовнішніми сервісами модуль коректно обробляє їх і повертає спеціальні значення, що дозволяє основній логіці програми адекватно реагувати на відсутність або затримку даних.

3.2.5 Модуль редагування фішингових фраз

Модуль редагування фішингових фраз забезпечує користувача інструментом для гнучкого керування списком ключових слів і виразів, які використовуються для виявлення підозрілих повідомлень у тексті електронних листів. За допомогою цього модуля користувач може додавати нові фрази, редагувати існуючі або видаляти непотрібні, що дозволяє оперативно адаптувати систему захисту до нових видів фішингових атак.

Інтерфейс модуля реалізований із використанням бібліотеки tkinter, що забезпечує простоту і зручність у роботі. Вікно редактора містить список поточних фраз, а також кнопки для додавання, редагування та видалення елементів. Зміни автоматично зберігаються і передаються основній системі для подальшого використання під час аналізу листів.

3.3 Алгоритм функціонування засобу

Робота програми починається з ініціалізації головного вікна графічного інтерфейсу, яке є основним засобом взаємодії користувача із системою. На цьому етапі відбувається підготовка середовища для подальшої роботи: створюється

вікно з усіма необхідними елементами управління, такими як вкладки, кнопки, поля введення та таблиці, що дозволяють користувачу зручно налаштовувати параметри моніторингу. Завдяки використанню бібліотеки tkinter, інтерфейс має кросплатформенний характер, що забезпечує стабільну роботу на різних операційних системах. Крім того, для підтримки плавності роботи інтерфейсу та уникнення його блокування під час виконання ресурсомістких операцій, таких як перевірка пошти, застосовується багатопоточність, що дозволяє запускати моніторинг у фоновому режимі.

Після запуску інтерфейсу користувач вводить необхідні налаштування, серед яких – додавання поштових акаунтів, введення API-ключів для взаємодії з зовнішніми сервісами безпеки, а також формування та редагування списку фішингових фраз, які будуть використовуватися для аналізу тексту листів. Ці налаштування зберігаються локально у відповідних файлах, що дозволяє зберігати стан системи між сесіями роботи та забезпечує безперервність моніторингу.

Далі програма встановлює захищене з’єднання з поштовими серверами через протокол IMAP, використовуючи SSL/TLS для шифрування даних. Авторизація відбувається на основі облікових даних, введених користувачем. Після успішного підключення система обирає папку для моніторингу, зазвичай це INBOX, і починає періодично перевіряти наявність нових непрочитаних листів. Для унікальної ідентифікації кожного повідомлення використовується UID, що дозволяє відстежувати оброблені листи та уникати їх повторної перевірки.

Кожен новий лист проходить багаторівневий аналіз. Спершу виконується перевірка автентичності за допомогою SPF, DKIM та DMARC – стандартів, які допомагають визначити, чи було повідомлення надіслано з дозволеного джерела і чи не було воно змінене в процесі передачі. Після цього відбувається оцінка репутації IP-адреси відправника через DNS-запити до чорних списків, що дає змогу швидко виявляти потенційно небезпечних відправників.

Наступним етапом є сканування URL-адрес, які містяться у листі, за допомогою зовнішніх сервісів безпеки, таких як VirusTotal і Google Safe Browsing.

Ці сервіси надають детальний аналіз на наявність фішингових посилань чи інших загроз. На рисунку 3.2 Алгоритм перевірки посилань у листі

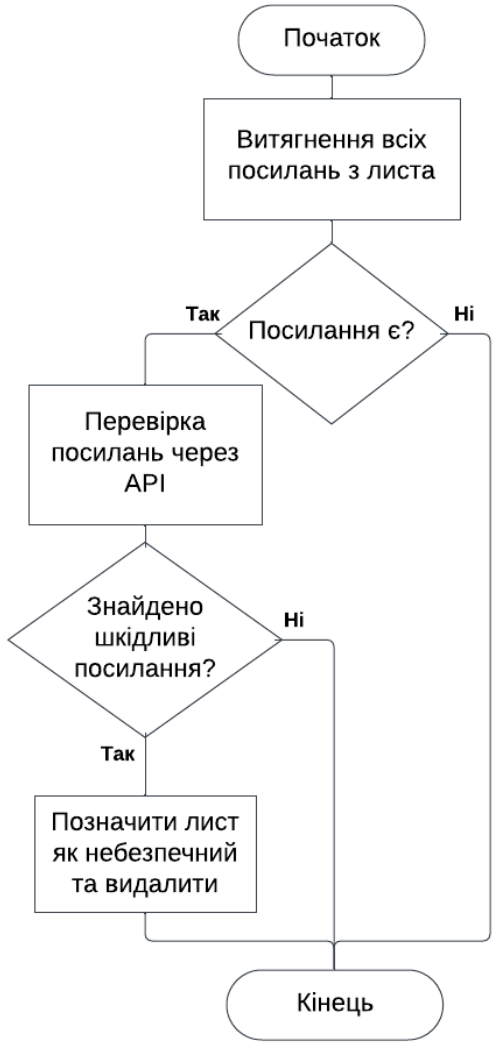


Рисунок 3.2 – Алгоритм перевірки посилань

Одразу після посилань здійснюється перевірка вкладень. Засіб здійснює пошук вкладень в листі, якщо вкладення найдене, то його зміст декодується у байтове значення та це значення відправляється у Virustotal, де здійснюється його перевірка.

Паралельно з цим виконується пошук фішингових фраз у тексті листа, що дозволяє виявляти підозрілі повідомлення навіть у випадках, коли інші ознаки загрози відсутні. Засіб шукає всі слова у листі, та зрівнює їх з фразами у файлі. Якщо знайдено фразу то лист позначається як підозрілий.

Алгоритм аналізу фішингових фраз зображено на рисунку 3.3

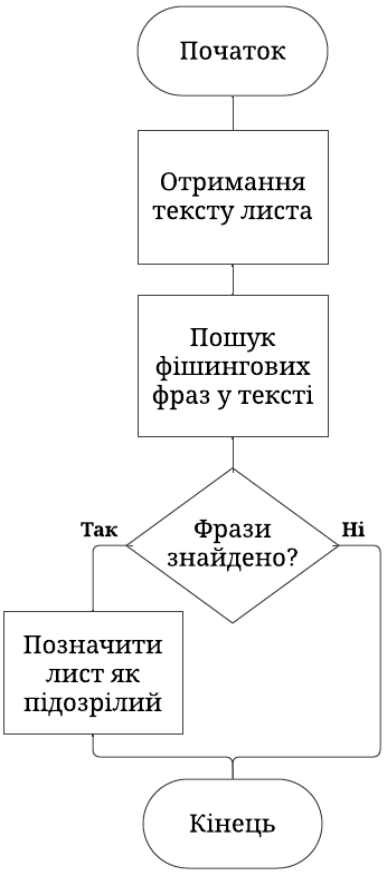


Рисунок 3.3 – Алгоритм аналізу фішингових фраз

У разі виявлення потенційної загрози система приймає рішення про автоматичне видалення листа, позначення для подальшого розгляду користувачем або позначення листа як безпечного. На рисунку 3.4 зображено алгоритм класифікації листа за рівнем загроз.

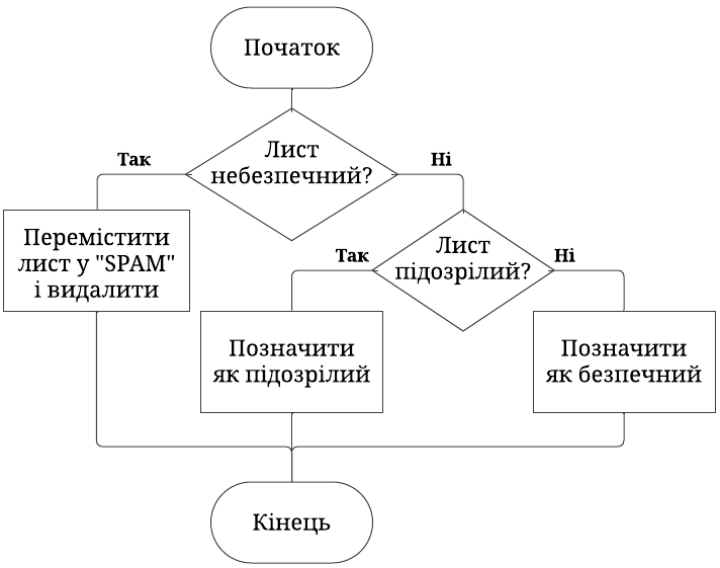


Рисунок 3.4 – Алгоритм класифікації листа за рівнем загроз

Водночас інформація про оброблені листи зберігається в локальних файлах, що запобігає повторній обробці і підвищує ефективність роботи. Особлива увага приділяється стабільності роботи – у разі втрати з’єднання з поштовим сервером програма автоматично намагається відновити зв’язок, забезпечуючи безперервність моніторингу.

Алгоритм взаємодії користувача з програмою зображено на рисунку 3.5

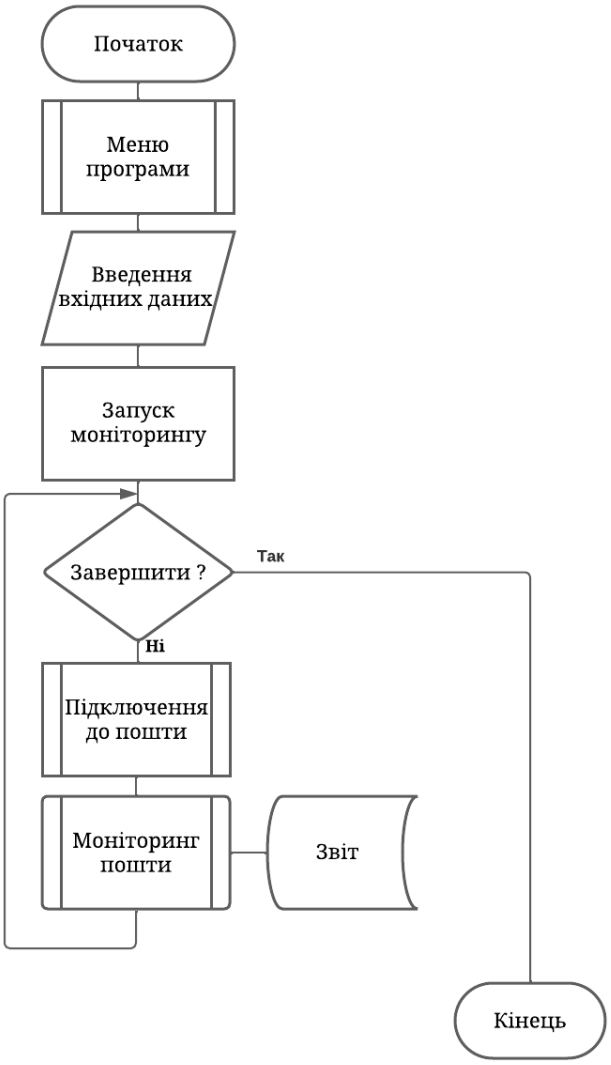


Рисунок 3.5 – Алгоритм взаємодії користувача з програмою

3.4 Інтерфейс засобу

Інтерфейс засобу побудований таким чином, щоб забезпечити максимальну зручність і прозорість взаємодії користувача з системою, незалежно від рівня його технічної підготовки. Завдяки логічному поділу на три вкладки, користувач

отримує чітку структуру роботи, що дозволяє швидко орієнтуватися у функціоналі, легко налаштовувати параметри та контролювати процес моніторингу.

Першою вкладкою є розділ для роботи з обліковими записами електронної пошти. Саме тут користувач вводить усі необхідні параметри для підключення до поштової скриньки: адресу електронної пошти, сервер, пароль і унікальну назву акаунта для зручної ідентифікації. Інтерфейс цієї вкладки інтуїтивно зрозумілий, що дозволяє швидко додавати нові акаунти або видаляти непотрібні. Користувач може запускати або зупиняти процес моніторингу у будь-який зручний час. Це дозволяє гнучко керувати перевіркою пошти, оперативно реагувати на зміни та контролювати безпеку кожної скриньки окремо. На рис.3.6 зображено інтерфейс вкладки «Акаунти»

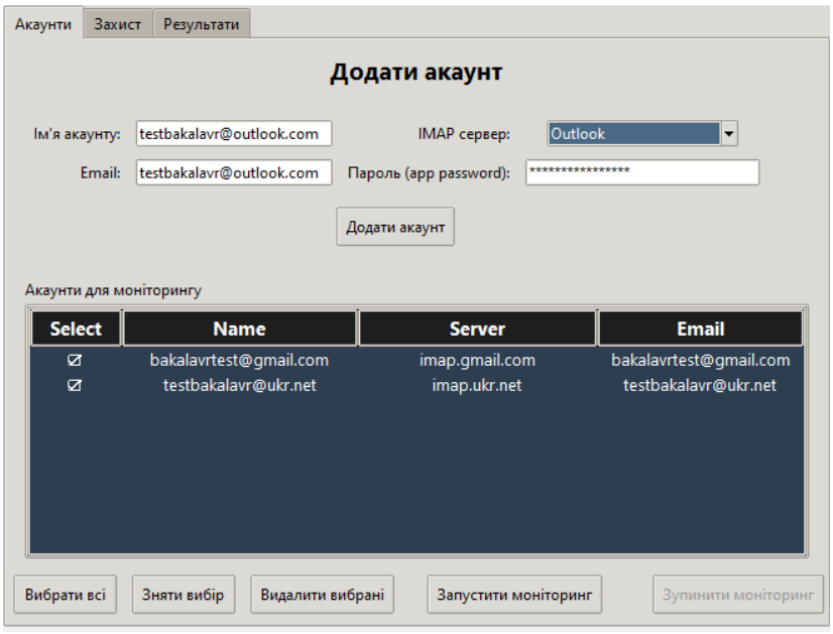


Рисунок 3.6 – Інтерфейс вкладки «Акаунти»

Друга вкладка відповідає за налаштування захисту. На цій сторінці користувач визначає, які саме перевірки будуть застосовуватися до вхідної пошти. Можна активувати аналіз фішингових фраз, перевірку автентичності відправника за допомогою SPF, DKIM та DMARC, а також перевірку посилань у листах через зовнішні сервіси, такі як VirusTotal, Google Safe Browsing чи OpenPhish. Окрім цього, тут вводяться ключі API для інтеграції із зовнішніми сервісами, налаштовується автоматичне видалення або переміщення підозрілих листів, а

також редагується список фішингових фраз, які використовуються для пошуку потенційно небезпечних повідомлень. Гнучкість налаштувань дозволяє адаптувати рівень захисту під конкретні потреби користувача чи організації, забезпечуючи баланс між безпекою та зручністю. На рис.3.7 зображено інтерфейс вкладки «Захист»

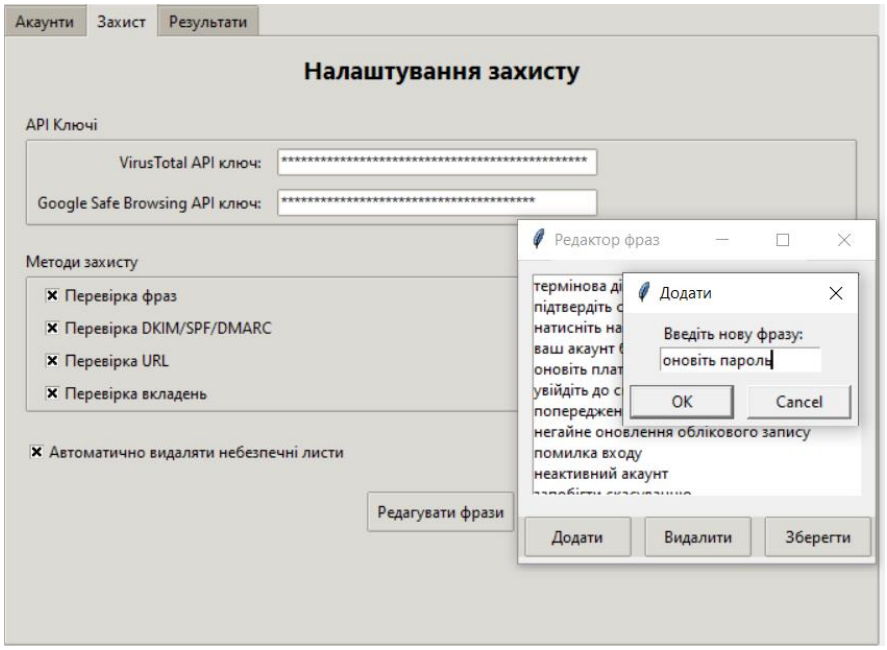


Рисунок 3.7 – Інтерфейс вкладки «Захист»

Третя вкладка зосереджена на відображенні результатів роботи засобу та моніторингу поточного стану системи. На цій сторінці користувач бачить детальний журнал подій із інформацією про всі оброблені листи: тему, відправника, статус перевірки, причину спрацювання захисту, виконані дії над повідомленнями, а також час, витрачений на перевірку кожного окремого листа. Також на цій вкладці є результати перевірки для кожного акаунту окремо, наприлад кількість перевірених, підозрілих або видалених листів. Такий підхід дозволяє не лише швидко оцінити поточний стан безпеки пошти, а й детально аналізувати ефективність роботи засобу та своєчасно виявляти потенційні проблеми. Вся інформація оновлюється у реальному часі, що дає змогу оперативно реагувати на виявлені загрози та контролювати процес моніторингу без зайвих зусиль.

На рис.3.8 зображено інтерфейс вкладки «Результати»

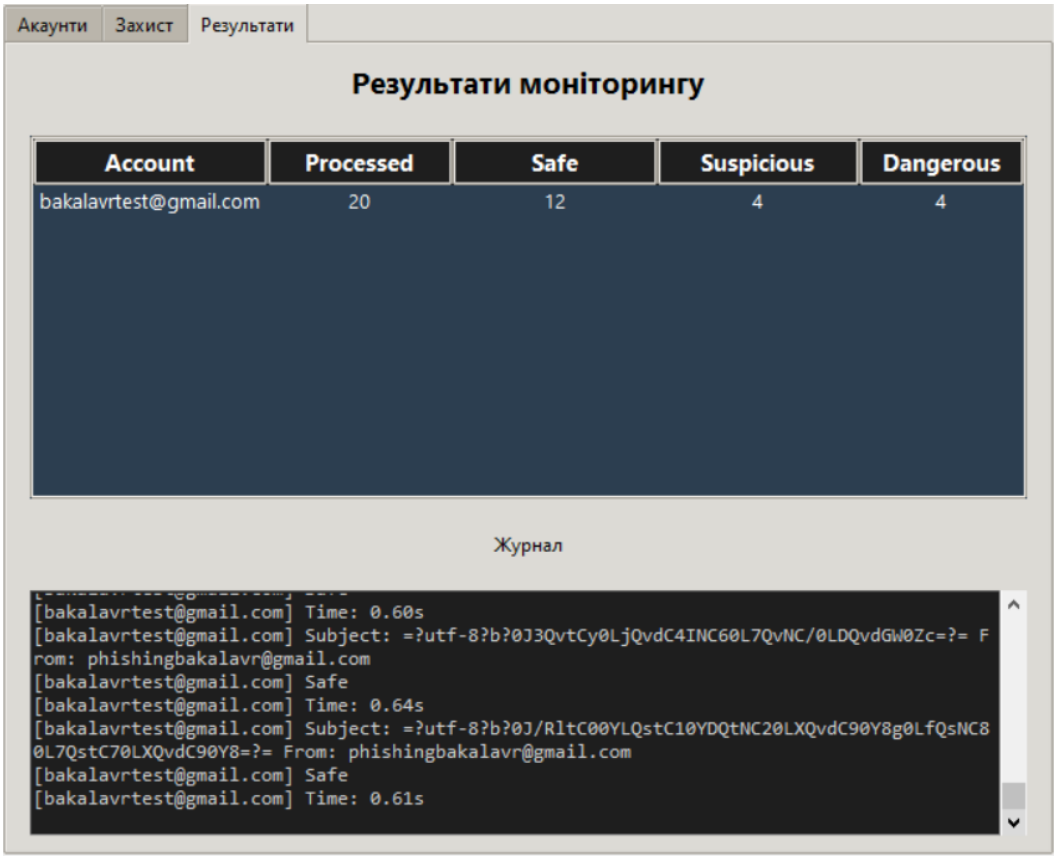


Рисунок 3.8 – Інтерфейс вкладки «Результати»

3.5 Експериментальні дослідження

Для комплексної перевірки ефективності розробленого засобу планується послідовне надсилання тестових листів, кожен з яких міститиме конкретний вид атаки. Такий поетапний підхід дозволяє детально оцінити здатність системи виявляти різноманітні загрози, що характерні для фішингових кампаній та інших видів шкідливої електронної кореспонденції.

Кожен отриманий лист проходить повний цикл перевірки, під час якого фіксується детальна інформація про результати кожного етапу аналізу, виконані автоматичні дії (наприклад, видалення або позначення листа), а також час, витрачений на обробку. Ця інформація дає змогу не лише виявити потенційні слабкі місця в системі, але й оцінити продуктивність та швидкодію засобу в реальних умовах.

Першим етапом після запуску засобу буде додавання акаунту поштової скриньки. Інформація, яку необхідно ввести включає ім’я акаунту для зручності і зрозуміння, електронну поштову адресу, IMAP сервер пошти та пароль від акаунту або пароль додатку (app password). Після введення цих даних, потрібно натиснути на кнопку «Додати аккаунт», після чого акаунт буде доданий в список всіх акаунтів та поля будуть очищені (рис.3.9).

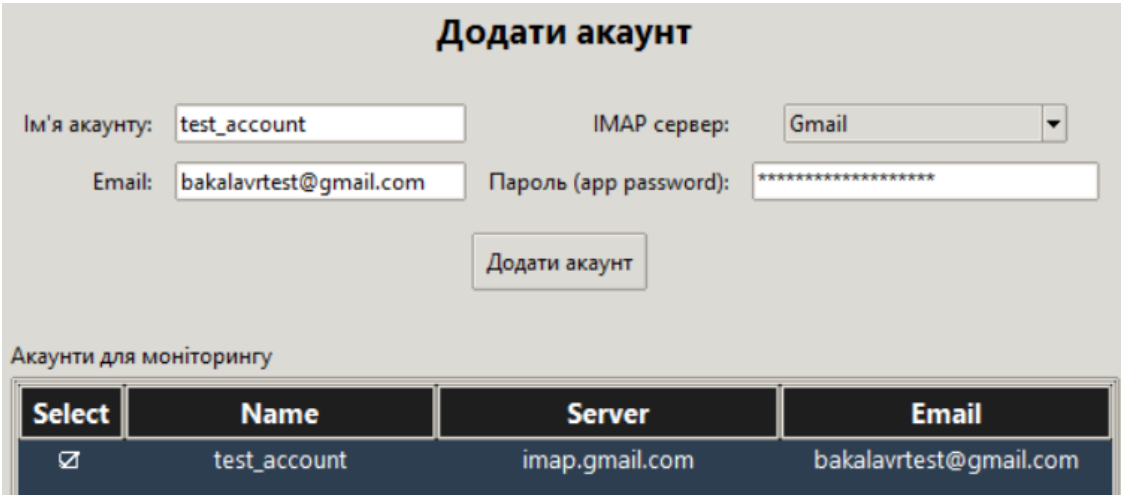


Рисунок 3.9 – Доданий новий акаунт

IMAP-сервер можна обрати з спливаючого списку, або при необхідності, можна обрати варіант «custom», після чого з’явиться додаткове поле для введення іншого IMAP-серверу (рис.3.10).

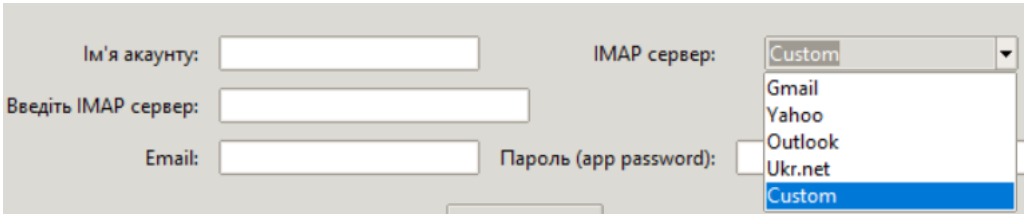


Рисунок 3.10 – Додавання кастомного IMAP-серверу

Також, можна додати декілька акаунтів та при необхідності видалити їх. Можна обрати всі акаунти, або тільки необхідні для подальших дій (рис.3.11).

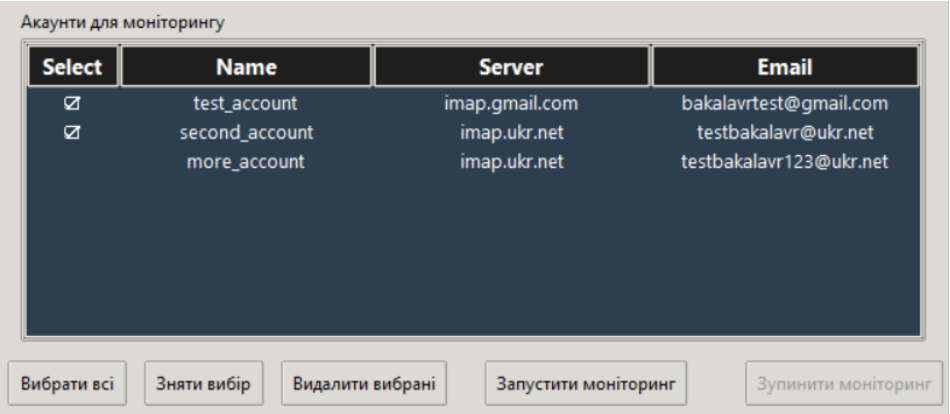


Рисунок 3.11 – Додано та обрано декілька акаунтів

Наступні кроки потрібно виконувати у вкладці «Захист». У поля з API ключами потрібно ввести API-ключі цих сервісів. Всі введені тут дані, відображаються у вигляді зірочок (*), для захисту від копіювання та для безпеки. (рис.3.12)

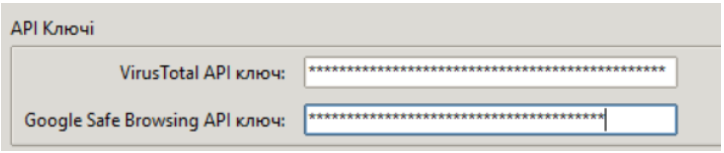


Рисунок 3.12 – Введені API-ключі

Для зручності та для використання необхідних перевірок, у засобі передбачені прапорці вибору необхідних перевірок. Кожну перевірку можна включити або виключити. Також присутня кнопка яка відповідає за автоматичне видалення небезпечних листів, якщо користувачу не потрібно видаляти небезпечні повідомлення (рис.3.13).

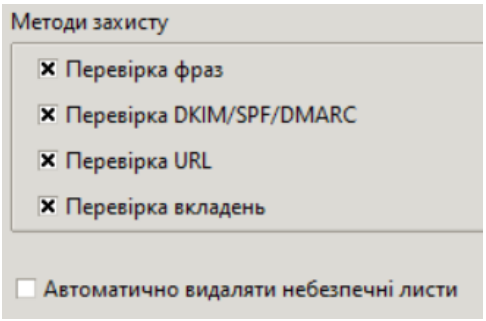


Рисунок 3.13 – Прапорці вибору перевірок та видалення листів

При необхідності, можна додати або видалити фішингову фразу, якщо натиснути на відповідну кнопку. Після натискання, з’являється окреме вікно де можна видалити або додати нову фішингову фразу (рис.3.13).

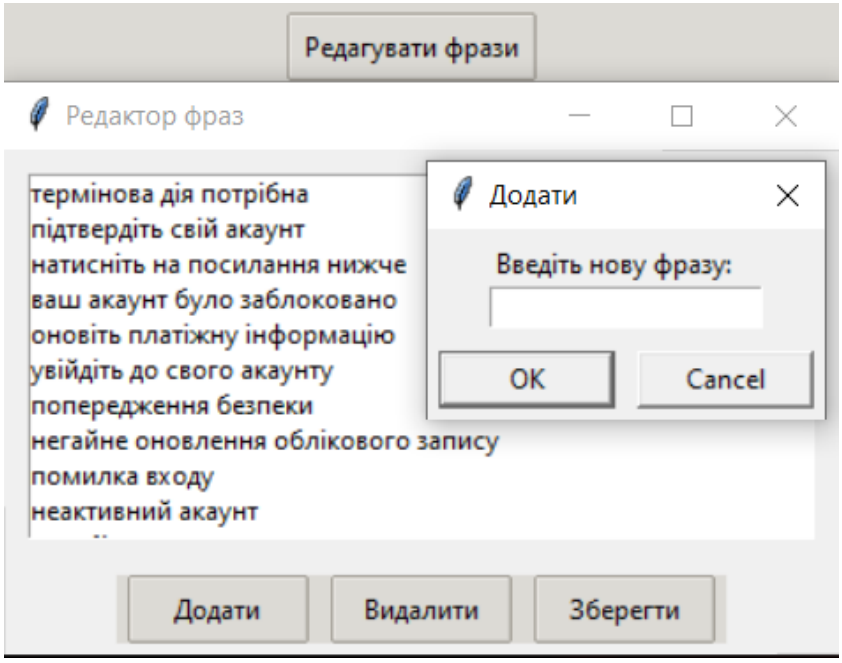


Рисунок 3.13 – Вікно редагування фішингових фраз

Після введення всіх необхідних даних, потрібно натиснути «Запустити моніторинг» для початку перевірки поштових повідомлень на поштовому сервісі.

Для початку, відправимо звичайний лист, для перевірки засобу. Текст в листі буде без фішингових ознак, наприклад «Привіт, як справи» (рис.3.14).

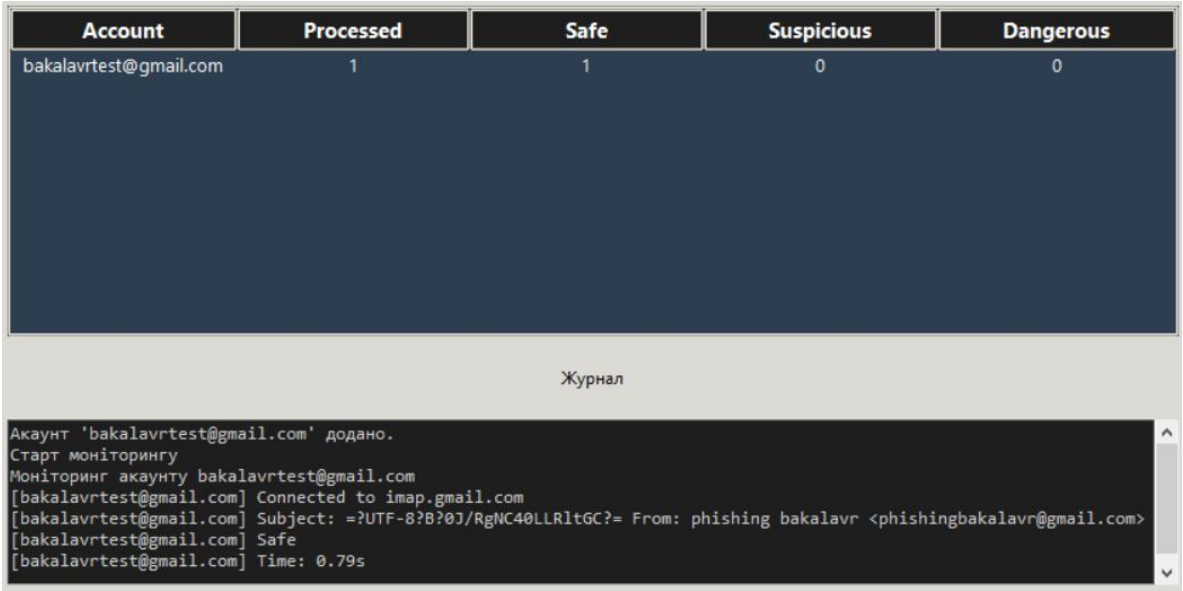


Рисунок 3.14 – Повідомлення без фішингових ознак

Провівши тестування, засобом було знайдено лист, перевірено його та позначено як безпечний (safe), і перевірка листа відбувалась 0,79 секунд. Також у статистиці акаунту видно, що засобом було опрацьовано 1 лист, та 1 лист був безпечним.

Наступної перевіркою буде лист з фішинговою фразою «помилка входу» (рис.3.15).

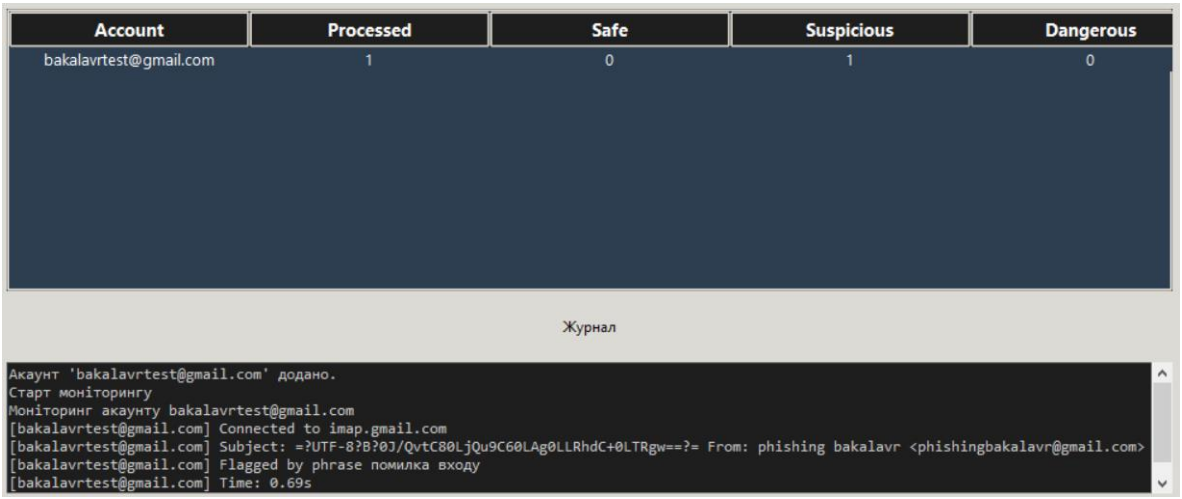


Рисунок 3.15 – Повідомлення з фішинговою атакою

Дане повідомлення було знайдено та перевірено, і під час перевірки засіб виявив фішингову фразу. Лист позначився як «Suspicious», та час перевірки був 0.69 секунд. На самій поштовій скриньці, лист був також позначений зіркою, що означає що він підозрілий (рис.3.16).

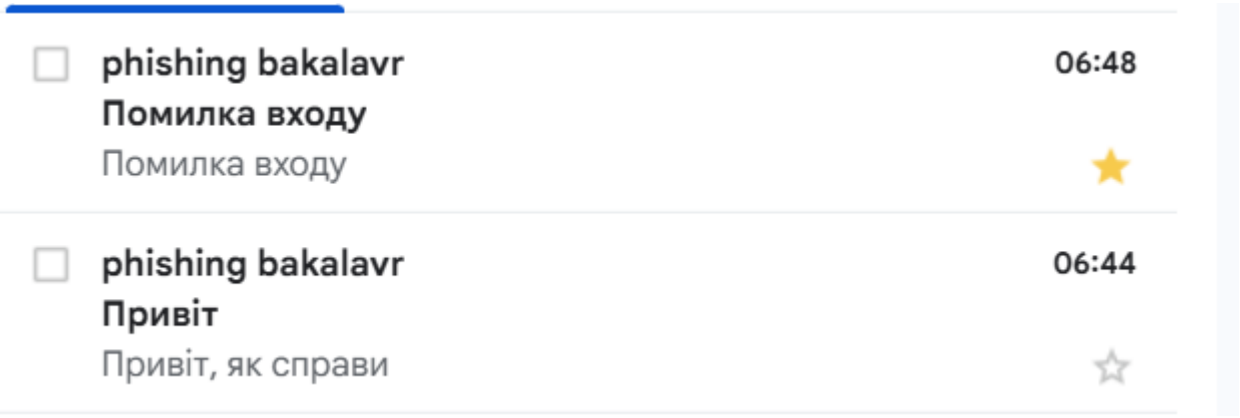


Рисунок 3.16 – Поштова скринька після перевірки

Ще однією з важливих перевірок є перевірка фішингових посилань. Фішингові посилання перевіряються на 3 різних сервісах, і якщо всі сервіси повернуть повідомлення що посилання безпечне, то лист позначиться як безпечний, в іншому ж випадку, лист буде видалений. Для перевірки посилань, потрібно додати API ключі даних сервісів та запустити перевірку. На поштову скриньку надійде два листа, один з безпечним посиланням, інший з небезпечним (рис.3.17).

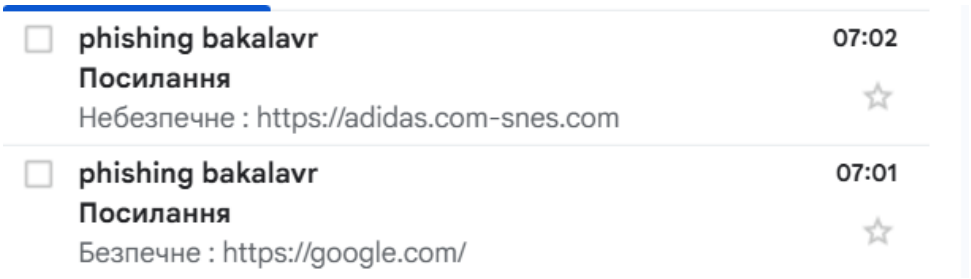


Рисунок 3.17 – Поштова скринька до перевірки посилань

Запустимо засіб та здійснимо перевірку посилань (рис. 3.18).

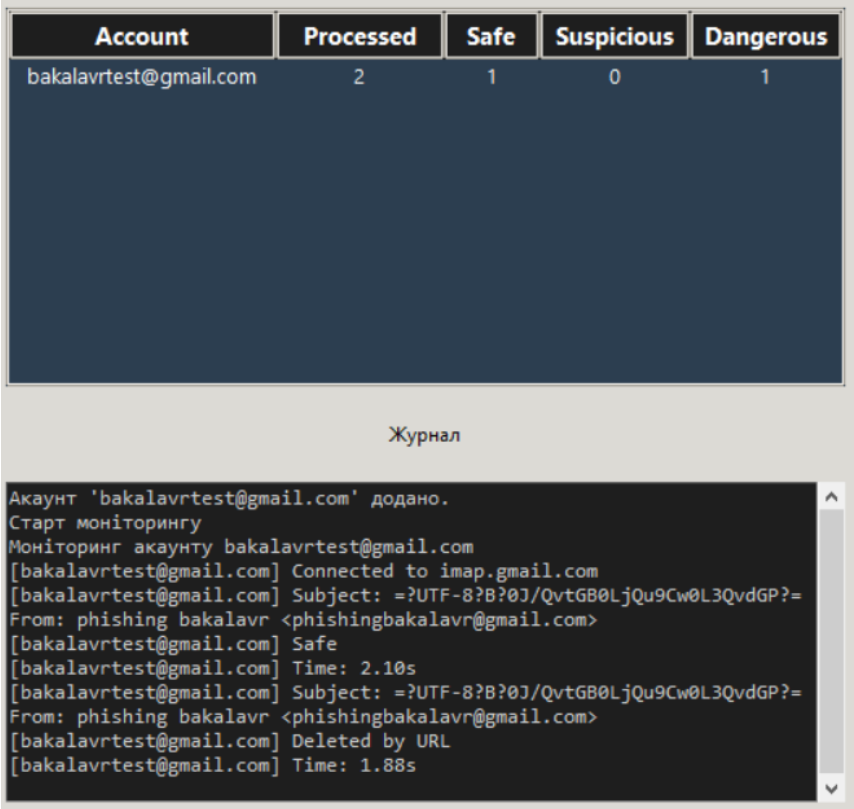


Рисунок 3.18 – Перевірка посилань засобом

При моніторингу, засіб виявив два листа з посиланнями та перевірив їх у зовнішніх сервісах. По результатам перевірки, безпечне посилання було перевірено за 2.10 секунд та позначено як безпечне, в той час як лист з небезпечним посиланням був видалений через фішингове посилання, та час на перевірку зайняв 1.88 секунд. У результатах моніторингу було позначено що перевірено два посилання, одне з яких було безпечним, інше небезпечним. У поштовій скриньці небезпечне посилання було видалено (рис. 3.19).

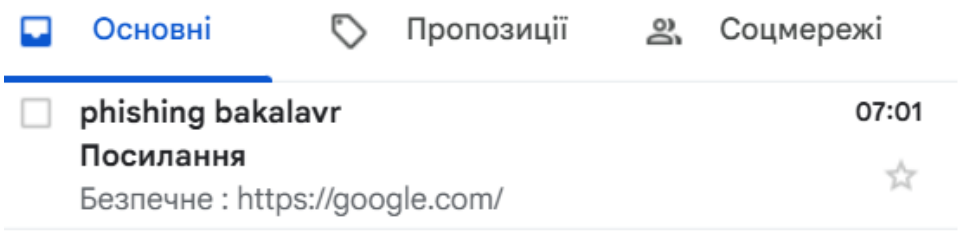


Рисунок 3.19 – Поштова скринька після перевірки посилань

Для перевірки DNSBL можна замінити функцію витягування IP адреси з листа на готову небезпечну тестову IP адресу «127.0.0.2». Після заміни ділянки коду, надсилаємо звичайний лист та запускаємо моніторинг пошти (рис. 3.20).

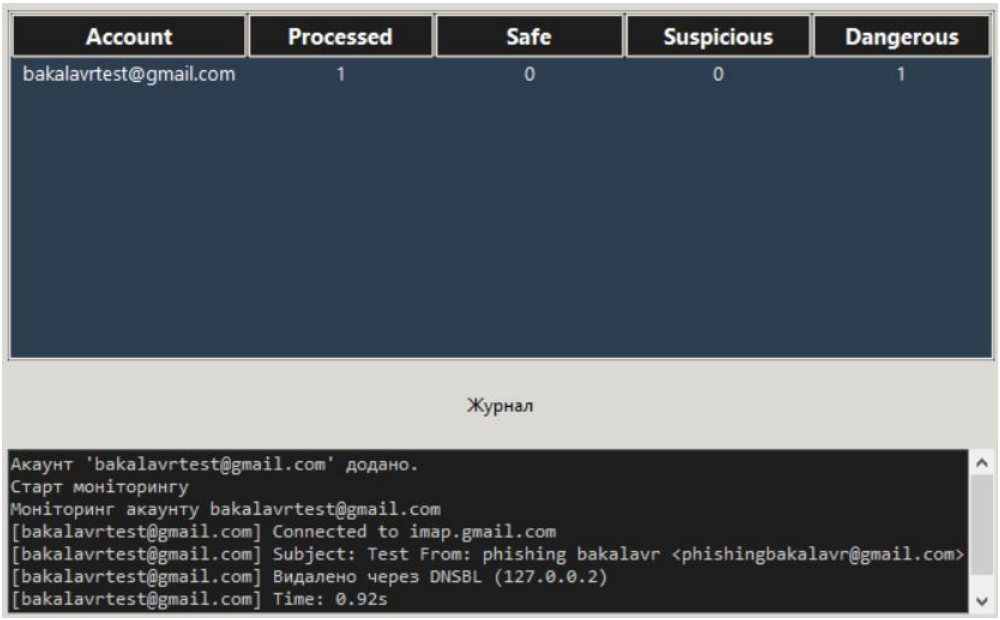


Рисунок 3.20 – Результати перевірки DNSBL

Моніторинг виявив IP та при зверненні до зовнішніх сервісів, отримав що ця IP-адреса знаходиться у чорних списках, та видалив повідомлення. Повідомлення було перевірено за 0.93 секунди, та за результатами перевірки було видалене.

Для оцінювання достовірності розробленого засобу було згенеровано за допомогою штучного інтелекту ChatGPT сто електронних повідомлень, що імітували справжній діловий трафік: у текстах поєднувалися українська та англійська мови, змінювалися заголовки та стиль викладу, а кожне третє повідомлення містило посилання для перевірки механізмів аналізу URL [55]. Усі листи надсилалися на єдиний тестовий обліковий запис із ввімкненими усіма перевірками, а додаткову перевірку проводили через інтеграцію із сервісами VirusTotal та Google Safe Browsing. Така методика дала змогу змоделювати реальні умови обміну повідомленнями й підтвердити, що інструмент коректно розпізнає безпечні електронні листи й надійно ігнорує безпечний контент. На рис. 3.21 зображено результати перевірки засобом 100 безпечних листів

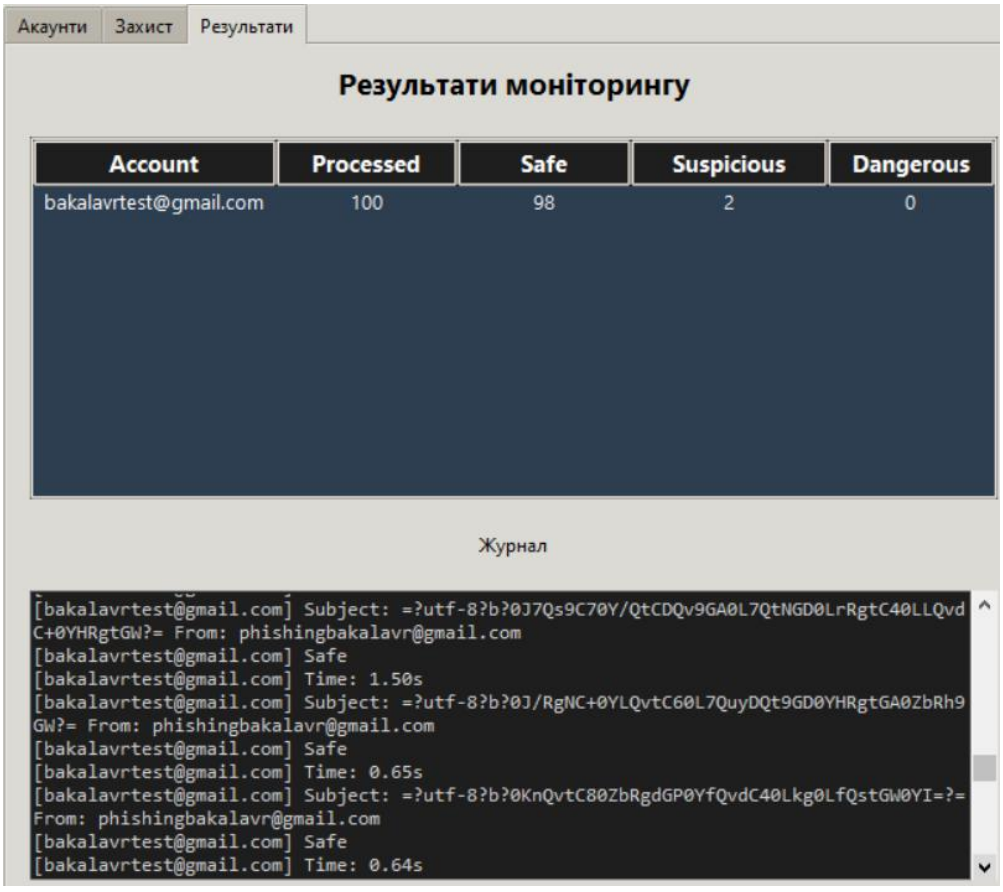


Рисунок 3.21 – Результати перевірки 100 безпечних повідомлень

По результатам перевірки засіб коректно класифікував усі 100 повідомлень, позначивши 98 як безпечні та лише у двох випадково виявив фіктивні фішингові фрази, що свідчить про високий рівень точності системи і водночас підкреслює незначну ймовірність хибних спрацьовувань. Час на перевірку 100 повідомлень зайняв близько 180 секунд, на звичайне повідомлення було витрачено в середньому 0.7 секунд, на повідомлення з посиланням – 3 секунди.

У межах тестування було згенеровано та відправлено 100 електронних повідомлень, кожне з яких містило ознаки потенційної загрози – або фішингові фрази зі словника, або підозрілі URL на фейкові ресурси, інколи доповнені DNSBL-повідомленнями. Усі листи надійшли на один контрольний акаунт із ввімкненими всіма перевірками і були перевірені через VirusTotal та Google Safe Browsing. Таким чином система працювала в умовах, де кожне з 100 повідомлень вимагало класифікації як «небезпечне» або «підозріле». На рис. 3.22 зображено результати перевірки засобом 100 фішингових листів.

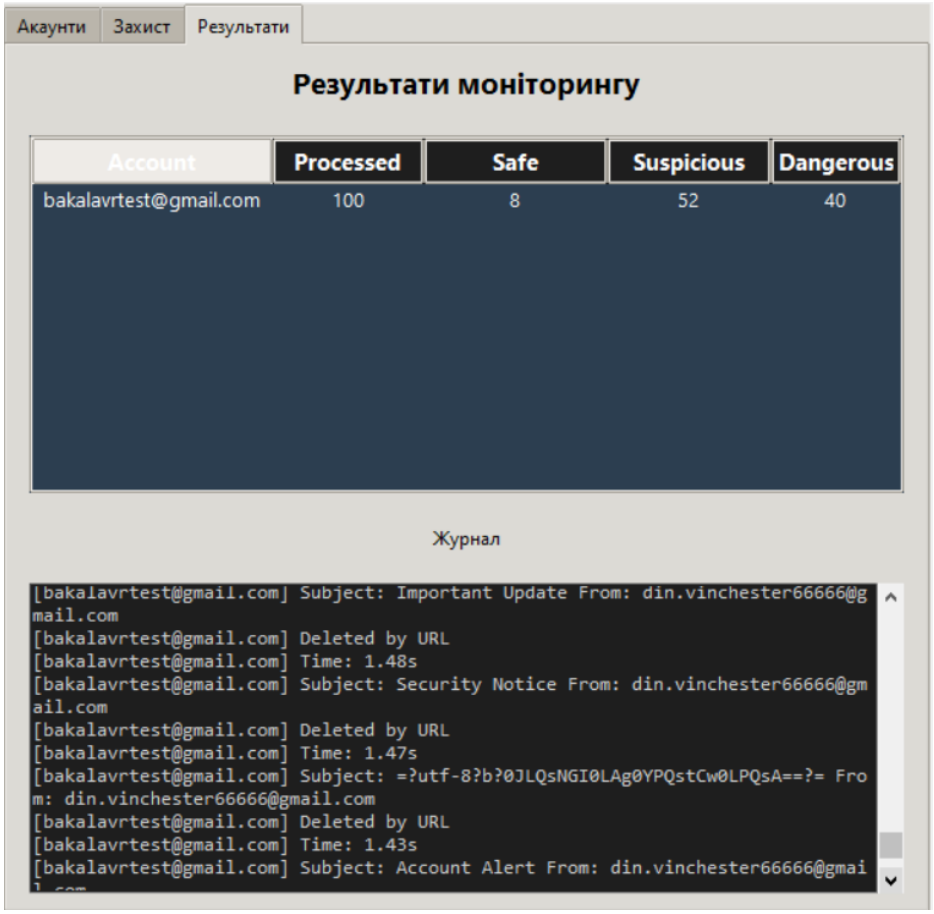


Рисунок 3.22 – Результати перевірки 100 фішингових листів

Під час тестування засіб опрацював 100 електронних повідомлень: 40 із них класифіковано як небезпечні через небезпечні посилання або DNSBL, 52 визнано підозрілими через фішингові фрази, а 8 повідомлень система віднесла до безпечних. Засіб не зміг знайти фішингові фрази у 8 повідомлень через недостатню кількість фішингових фраз у базі. Така статистика свідчить про високу точність класифікації, здатність чітко відокремлювати прямі загрози та підтримувати низький рівень хибних спрацьовувань.

У табл. 3.1 наведено результати роботи засобу.

Таблиця 3.1 – Результати роботи засобу

	Надіслані	**Визначені як безпечні**	**Визначені як підозрілі**	**Визначені як небезпечні**	**Виявлення**
Надіслані безпечні	100	98	2	0	98%
Надіслані небезпечні	100	8	52	40	92%

Прототип системи був розроблений на мові Python з використанням модульної архітектури та відкритих бібліотек для роботи з поштовими протоколами, текстовим аналізом та обробкою мережевих запитів. Кожен компонент інтегровано в єдину систему, де взаємодія між модулями забезпечується через чітко визначені інтерфейси. У складі рішення також є модулі для логування та генерації звітів, що спрощують аналіз результатів і відстеження процесу перевірок.

Тестування було проведено на наборах змодельованих фішингових листів, що використовують різні методи атак. Результати аналізу підтвердили високу точність у виявленні фішингових повідомлень із мінімальними хибними спрацьовуваннями. Тестування часу обробки показало, що система ефективно працює навіть при середньому навантаженні.

ВИСНОВКИ

У ході виконання бакалаврської кваліфікаційної роботи було проведено вивчення наукових та технічних джерел із теми виявлення фішингових атак зокерма у електронній пошти. Проаналізовано класичні механізми захисту, зокрема перевірку автентичності відправника за протоколами SPF, DKIM і DMARC, сигнатурні фільтри, а також підходи до синтаксичного й семантичного аналізу тексту та сканування вкладень.

На основі отриманих висновків розроблено архітектуру програмного засобу, що передбачає багатоетапну обробку електронних повідомлень. Спочатку здійснюється збір та нормалізація вхідних даних із поштових серверів, далі – перевірка автентичності відправника, після чого проводиться лінгвістичний аналіз тексту з урахуванням семантичних ознак фішингу. На завершальному етапі оцінюється безпека URL-адрес та вкладень, а повідомлення класифікуються за рівнем ризику, що забезпечує гнучке налаштування під різноманітні варіанти атак.

Реалізація прототипу виконана на мові Python із використанням модульного підходу та відкритих бібліотек для роботи з поштовими протоколами, текстовим аналізом і обробкою мережевих ресурсів. Кожен компонент інтегровано в єдину систему обробки, де взаємодія між модулями відбувається через чітко розмежовані інтерфейси. До складу рішення увійшли модулі логування та формування звітів, що спрощують аналіз результатів і відстеження історії перевірок.

Тестування програмного засобу проводилося на наборах змодельованих листів із різними видами фішингових технік. Аналіз результатів підтвердив високу точність виявлення фішингових повідомлень при мінімальному рівні хибних спрацьовувань. Вимірювання часу обробки вхідного потоку листів показало, що система здатна працювати навіть на середніх навантаженнях.

Отже, створено готовий інструмент для автоматизованої перевірки електронних повідомлень на ознаки фішингових атак, що успішно поєднує класичні методи та сучасний лінгвістичний аналіз.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Phishing. What is Phishing. URL: https://www.phishing.org/what-is-phishing (Дата звернення: 03.05.2025).

2. Whitepay. Що таке фішинг та як від нього захиститися. 2024. URL : https://whitepay.com/uk/news/shho-take-fishing-ta-yak-vid-nogo-zahistitisya (дата звернення 03.05.2025)

3. Ростецький В.Б. Аналіз методів виявлення фішингових атак у електронних листах. LIV Всеукраїнська науково-технічна конференція факультету інформаційних технологій та комп'ютерної інженерії (ВНТКП ВНТУ) : тези доп., Вінниця, 2025. URL : https://conferences.vntu.edu.ua/index.php/all-fitki/all-fitki-2025/paper/view/23629/19516 (дата звернення : 03.05.2025)

4. Almomani, A., Gupta, B. B., Atawneh, S., Meulenberg, A., & Almomani, E. (2013). A survey of phishing email filtering techniques. IEEE communications surveys & tutorials, 15(4), 2070-2090. URL : https://ieeexplore.ieee.org/abstract/document/6489877 (дата звернення 05.05.2025)

5. Terranovasecurity. What is Vishing?. URL : https://www.terranovasecurity.com/solutions/security-awareness-training/what-is-vishing (дата звернення 05.05.2025)

6. Yeboah-Boateng, E. O., & Amanor, P. M. (2014). Phishing, SMiShing & Vishing: an assessment of threats against mobile devices. Journal of Emerging Trends in Computing and Information Sciences, 5(4), 297-307. URL : https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=7a271a3ff90b2a19d6b4f4ecc800e0aebdcda063 (дата звернення 05.05.2025)

7. NordVPN. Що таке фішинг і як не потрапити на гачок. 2024. URL : https://nordvpn.com/uk/blog/shcho-take-fishynh/ (дата звернення 05.05.2025)

8. Hackyourmom. Targeted phishing. 2023. URL : https://hackyourmom.com/en/kibervijna/czilovyj-fishyng/ (дата звернення 05.05.2025)

9. Kalaharsha, P., & Mehtre, B. M. (2021). Detecting Phishing Sites--An Overview. arXiv preprint arXiv:2103.12739. URL : https://arxiv.org/abs/2103.12739 (дта звернення 05.05.2025)

10. Brody, R. G., Mulig, E., & Kimball, V. (2007). PHISHING, PHARMING AND IDENTITY THEFT. Academy of Accounting & Financial Studies Journal, 11(3). URL : https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=c3f9bedf86614a9ca7cf18197bfbb978f56e6e56 (дата звернення 11.05.2025)

11. Hackyoumom. What is clone phishing?. 2023. URL : https://hackyourmom.com/en/kibervijna/shho-take-fishyng-kloniv/ (дата звернення 11.05.2025)

12. Zhylin, A., & Shevchuk, O. (2022). Метод аналізу фішингових повідомлень. Collection" Information Technology and Security", 10(1), 72-82. URL : http://its.iszzi.kpi.ua/article/view/261176 (дата звернення 11.05.2025)

13. Itgovernance. What Is Angler Phishing? Definition, Examples & Prevention. 2019. URL : https://www.itgovernance.co.uk/blog/beware-of-angler-phishing (дата звернення 11.05.2025)

14. Гаць, Д. М., & Куперштейн, Л. М. (2023). Засіб попередження фішингових атак на основі гейміфікації (Doctoral dissertation, ВНТУ). URL : https://www.researchgate.net/profile/Leonid-Kupershtein/publication/380571650_Privention_of_phishing_attacks_based_on_gamification/links/6643d66c08aa54017a0b63a8/Privention-of-phishing-attacks-based-on-gamification.pdf (дата звернення 13.05.2025)

15. Fenix. Фішинг – що це таке?. 2024. URL : https://fnx.com.ua/ua/articles/publications/25 (дата звернення 13.05.2025)

16. Go deal brokers. Фішинг: Як захистити себе від онлайн-шахрайства. 2024. URL : https://business-broker.com.ua/blog/fishynh-iak-zakhystyty-sebe-vid-onlajn-shakhrajstva/ (дата звернення 13.05.2025)

17. Гривня. Як розпізнати та захиститися від фішингу в інтернеті: інструкція на 2025 рік. 2025. URL : https://www.grivnya.in.ua/bezpeka/phishing-security/ (дата звернення 13.05.2025)

18. H-X. Прогноз кіберзагроз 2024. 2023. URL : https://www.h-x.technology/ua/blog-ua/cyber-threats-forecast-2024-ua (дата звернення 13.05.2025)

19. 10guards. У 2023 році кількість фішингових атак зросла на 40%. 2024. URL : https://10guards.com/ua/blog/2024/04/08/phishing-attacks-increased-by-40-in-2023/ (дата звернення 13.05.2025)

20. Fintech insider. Аналітики зафіксували зростання активності криптовалютного фішингу у 2023 році. 2024. URL : https://fintechinsider.com.ua/u-2023-roczi-analityky-zafiksuvaly-zrostannya-aktyvnosti-kryptovalyutnogo-fishyngu/ (дата звернення 20.05.2025)

21. Сас, А. І., & Лях, Ю. І. Державне регулювання захисту інформації на об'єктах критичної інформаційної інфраструктури. економіка та публічне управління: нові виклики та рішення, 228. URL : https://khai.edu/assets/documents/7582/%D0%9A%D0%9E%D0%9D%D0%A4%D0%95%D0%A0%D0%95%D0%9D%D0%A6%D0%86%D0%AF_%D0%A5%D0%90%D0%86_601_23-24_01_2025.pdf#page=228 (дата звернення 20.05.2025)

22. Statista. Number of phishing attacks detected worldwide from 3rd quarter 2013 to 4th quarter 2024. 2024. URL : https://www.statista.com/statistics/266155/number-of-phishing-attacks-worldwide/ (дата звернення 20.05.2025)

23. SIPA. The Hacking of Sony Pictures: A Columbia University Case Study. 2015. URL : https://www.sipa.columbia.edu/sites/default/files/2022-11/Sony%20-%20Written%20Case.pdf (дата звернення 20.05.2025)

24. Terazus. В LinkedIn відбулася потужна хвиля фішингових атак. 2025. URL : https://terazus.com/uk/3717-v-linkedin-vidbulasja-potuzhna-xvilja-fishingovix-atak (дата звернення 20.05.2025)

25. Nadiyno. До яких кіберзагроз готуватися українцям у 2025 році. 2025. URL : https://nadiyno.org/do-yakyh-kiberzagroz-gotuvatysya-ukrayinczyam-u-2025-roczi/ (дата звернення 20.05.2025)

26. Wozny, H. (2023). *Microsoft 365 customizable and resilient to ransomware* (Bachelor's thesis, NTNU). URL : https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/3081000 (дата звернення : 20.05.2025)

27. gPanel. Google Workspace Email Security: Gmail & Google Workspace Protection. 2023. URL : https://gpanel.io/blog/google-workspace-email-security/ (дата звернення : 20.05.2025)

28. Ascic, H. J. (2023). *Effectiveness of cybersecurity awareness training in lowering the risks of email-borne attacks for Irish SME* (Doctoral dissertation, Dublin, National College of Ireland). URL : https://norma.ncirl.ie/7112/ (дата звернення : 20.05.2025)

29. Lankeshwar, S., & Rangaswamy, S. (2020). Mimecast: Cloud Email Security Management. URL : https://www.academia.edu/download/64729432/IRJET_V7I8513.pdf (дата звернення : 20.05.2025)

30. Saleem, B. M. (2021). *The p-fryer: Using machine learning and classification to effectively detect phishing emails* (Doctoral dissertation, Marymount University). URL : https://search.proquest.com/openview/71843d15fee62354e0d6bc3e039b2d7e/1?pq-origsite=gscholar&cbl=18750&diss=y (дата звернення : 20.05.2025)

31. Reese, B., & Chen, W. F. (2019, November). Cybersecurity: Phishing Attack Mitigation and Prevention for Higher Education. In *eLearn: World Conference on EdTech* (pp. 955-959). Association for the Advancement of Computing in Education (AACE). URL : https://www.learntechlib.org/p/211178/ (дата звернення : 20.05.2025)

32. BusinessWire. Abnormal AI Launches Breakthrough AI Agents to Reimagine Security Awareness Training and Provide Instant Board-Ready Data Insights 2025. URL : https://www.businesswire.com/news/home/20250428226412/en/abnormal-

ai-launches-breakthrough-ai-agents-to-reimagine-security-awareness-training-and-provide-instant-board-ready-data-insights (дата звернення : 20.05.2025)

33. Catteau, G. Rspamd: logiciel anti-spam opensource, performant, évolutif et personnalisable. In *JRES (Journées réseaux de l'enseignement et de la recherche) 2019*. URL : https://hal.science/hal-04807095/ (дата звернення : 20.05.2025)

34. Spaceship. Різні обличчя загроз електронної пошти. 2025. URL : https://www.spaceship.com/uk/blog/email-threats-phishing-whaling-scams/ (дата звернення 20.05.2025)

35. Cyberset. SPF-запис: Що це таке, як працює і навіщо потрібен?. 2024. URL : https://cyberset.com.ua/network/attacks-vs-defense/spf-record/ (дата звернення 23.05.2025)

36. Stripo. Як покращити доставлюваність за допомогою email-автентифікації. 2023. URL : https://stripo.email/ua/blog/how-to-improve-deliverability-with-email-authentication/ (дата звернення 23.05.2025)

37. Галицький, В. В. (2025). Система виявлення шкідливих вебсайтів на основі аналізу URL-адрес. URL : https://elar.khmnu.edu.ua/items/e76d019d-40e4-4f62-9a0a-1a3ee152fdcb (дата звернення 23.05.2025)

38. Hackyourmom. Password manager, what it is and for what purposes it is needed. 2023. URL : https://hackyourmom.com/en/pryvatnist/menedzher-paroliv-shho-cze-take-i-dlya-yakyh-czilej-vin-potriben/ (дата звернення 23.05.2025)

39. Megatrade. Cisco Umbrella для захисту трафіку від кібератак в перевантажених мережах. 2021. URL : https://megatrade.ua/news/reviews/cisco-umbrella-dlya-zakhistu-trafiku-vid-kiberatak-v-perevantazhenikh-merezhakh/ (дата звернення 23.05.2025)

40. Mohammad, R. M., Thabtah, F., & McCluskey, L. (2014). Predicting phishing websites based on self-structuring neural network. *Neural Computing and Applications*, *25*, 443-458. URL : https://link.springer.com/article/10.1007/s00521-013-1490-z (дата звернення 23.05.2025)

41. ESET. Безпека пошти: основні рекомендації для захисту скриньок від онлайн-загроз. 2019. URL : https://www.eset.com/ua/about/newsroom/press-

releases/security-tips/bezopasnost-pochty-vneshniye-i-vnutrenniye-faktory-zashchity-pochty/ (дата звернення 23.05.2025)

42. ESKA. Фішинг та ентерпрайз : як навчати співробітників протидії складним атакам?. 2024. URL : https://eska.global/blog/fishing-ta-enterprajz-yak-navchati-spivrobitnikiv-protidiyi-skladnim-atakam (дата звернення 23.05.2025)

43. Salloum, S., Gaber, T., Vadera, S., & Shaalan, K. (2022). A systematic literature review on phishing email detection using natural language processing techniques. IEEE Access,10, 65703-65727. URL : https://ieeexplore.ieee.org/abstract/document/9795286/ (дата звернення 23.05.2025)

44. IBM. What is NLP (natural language processing)? 2024. URL : https://www.ibm.com/think/topics/natural-language-processing (дата звернення 23.05.2025)

45. Lee, J., Lee, Y., Lee, D., Kwon, H., & Shin, D. (2021). Classification of attack types and analysis of attack methods for profiling phishing mail attack groups.IEEE Access,9, 80866-80872. URL : https://ieeexplore.ieee.org/abstract/document/9444397/ (дата звернення 27.05.2025)

46. Zhylin, A., & Shevchuk, O. (2022). Метод аналізу фішингових повідомлень. Collection" Information Technology and Security", 10(1), 72-82. URL : http://its.iszzi.kpi.ua/article/view/261176 (дата звернення 27.05.2025)

47. NinjaOne. Understanding and Preventing Email Spoofing Attacks. 2025. URL : https://www.ninjaone.com/blog/understanding-and-preventing-email-spoofing-attacks/ (дата звернення 27.05.2025)

48. Python. Підручник з Python. URL : https://docs.python.org/uk/3.13/tutorial/index.html (дата звернення 27.05.2025)

49. JetBrains. Learn PyCharm. URL : https://www.jetbrains.com/pycharm/learn/ (дата звернення 27.05.2025)

50. Lundh, F. (1999). An introduction to tkinter. URL: www. pythonware. com/library/tkinter/introduction/index. htm, 539, 540. URL : http://jgaltier.free.fr/Terminale_S/ISN/TclTk_Introduction_To_Tkinter.pdf (дата звернення 27.05.2025)

51. Matloff, N., & Hsu, F. (2007). Tutorial on Threads Programming with Python. University of California. URL : http://www.cs.ucdavis.edu/~matloff/matloff/public_html/Python/PyThreadsS2010.pdf (дата звернення : 30.05.2025)

52. Cesarini, F., Pappalardo, V., & Santoro, C. (2008, September). A comparative evaluation of imperative and functional implementations of the imap protocol. In Proceedings of the 7th ACM SIGPLAN workshop on ERLANG (pp. 29-40). URL : https://dl.acm.org/doi/abs/10.1145/1411273.1411279 (дата звернення : 30.05.2025)

53. Pinto, J. C., Scheid, E. J., Franco, M. F., & Granville, L. Z. (2024, June). Eeny, Meeny, Miny, Moe: Analyzing and Comparing the Selection of DNS Lookup Tools. In 2024 IEEE Symposium on Computers and Communications (ISCC) (pp. 1-6). IEEE. URL : https://ieeexplore.ieee.org/abstract/document/10733718/ (дата звернення : 30.05.2025)

54. Rajalakshmi, B., Rani, C. N., Vijayasekaran, G., Kumar, S. M., Hegde, S. B., & Srikanth, S. (2024, August). Optimized NLP Model with Caching, Parallel Processing and Streamlined I/O. In 2024 7th International Conference on Circuit Power and Computing Technologies (ICCPCT) (Vol. 1, pp. 619-623). IEEE. URL : https://ieeexplore.ieee.org/abstract/document/10672788/ (дата звернення : 03.06.2025)

55. Куперштейн, Л. М., & Примаков, Б. С. (2023). *Про використання ChatGPT в кібербезпеці* (Doctoral dissertation, ВНТУ). URL : https://www.researchgate.net/profile/Leonid-Kupershtein/publication/380571213_About_using_ChatGPT_in_cyber_security/links/6643d21f08aa54017a0b568a/About-using-ChatGPT-in-cyber-security.pdf (дата звернення 17.06.2025)

56. Методичні вказівки до виконання бакалаврських кваліфікаційних робіт зі спеціальності 125 «Кібербезпека» освітньо-професійні програми «Безпека інформаційних і комунікаційних систем» та «Кібербезпека критичних систем» / Уклад. В. А. Каплун, О. П. Войтович. Вінниця: ВНТУ, 2023. 61 с

ДОДАТКИ

Додаток А

ПРОТОКОЛ ПЕРЕВІРКИ КВАЛІФІКАЦІЙНОЇ РОБОТИ

Назва роботи: Засіб для перевірки повідомлень електронної пошти щодо ознак фішингової атаки

Автор роботи: Ростецький Володимир Богданович

Тип роботи: бакалаврська кваліфікаційна робота

Підрозділ кафедра захисту інформації ФІТКІ, група 1 БС-21б

Коефіцієнт подібності текстових запозичень, виявлених у роботі системою StrikePlagiarism **1,07 %**

Висновок щодо перевірки кваліфікаційної роботи (відмітити потрібне)

- [√] Запозичення, виявлені у роботі, є законними і не містять ознак плагіату, фабрикації, фальсифікації. Роботу прийняти до захисту
- [ ] У роботі не виявлено ознак плагіату, фабрикації, фальсифікації, але надмірна кількість текстових запозичень та/або наявність типових розрахунків не дозволяють прийняти рішення про оригінальність та самостійність її виконання. Роботу направити на доопрацювання.
- [ ] У роботі виявлено ознаки плагіату та/або текстових маніпуляцій як спроб укриття плагіату, фабрикації, фальсифікації, що суперечить вимогам законодавства та нормам академічної доброчесності. Робота до захисту не приймається.

Експертна комісія:

Завідувач кафедри ЗІ д. т. н., проф. _____ Володимир ЛУЖЕЦЬКИЙ

Гарант освітньої програми «Безпека інформаційних і комунікаційних систем» к. т. н., доц. _____ Леонід КУПЕРШТЕЙН

Особа, відповідальна за перевірку _____ Валентина КАПЛУН

З висновком експертної комісії ознайомлений(-на)

Керівник _____ Олеся ВОЙТОВИЧ

Здобувач _____ Володимир РОСТЕЦЬКИЙ

Додаток Б

ЛІСТИНГ ПРОГРАМИ

main.py

```
from gui import EmailMonitorApp

def main():
    EmailMonitorApp().run()

if __name__ == "__main__":
    main()
```

email_monitor.py

```
import email
import time
import json
import os
import socket
import re
import hashlib
import ssl
import imaplib
import dkim
import spf
import dns.resolver
from apis import (check_url_virustotal, check_url_gsb, check_url_openphish,
check_attachment_virustotal, check_dnsbl)

def load_phishing_phrases(file_path="phishing_phrases.json"):
    try:
        with open(file_path, "r", encoding="utf-8") as f:
            data = json.load(f)
            return data.get("phishing_phrases", [])
    except:
        return []

class EmailMonitor:
    def __init__(
        self,
        account_info,
        vt_api_key,
        gsb_api_key,
        check_phrases=True,
        check_dkim_spf=True,
        check_url=True,
        check_attachments=True,
        auto_delete=False,
        phishing_phrases=None,
        log_func=print,
        stats_callback=None
    ):
        self.account = account_info
        self.account_domain = account_info["email"].split("@")[1]
        self.vt_key = vt_api_key
        self.gsb_key = gsb_api_key
        self.check_phrases = check_phrases
        self.check_dkim_spf = check_dkim_spf
        self.check_url = check_url
        self.check_attachments = check_attachments
        self.auto_delete = auto_delete

        self.log = log_func
```

```
        self.stats_callback = stats_callback

        if phishing_phrases is not None:
            self.phrases = phishing_phrases
        else:
            self.phrases = load_phishing_phrases()

        self.processed_file = f"{self.account['name']}_processed.json"
        if os.path.exists(self.processed_file):
            try:
                with open(self.processed_file, "r", encoding="utf-8") as pf:
                    self.processed = set(json.load(pf))
            except:
                self.processed = set()
        else:
            self.processed = set()

        self.running = True

    def save_processed(self):
        try:
            with open(self.processed_file, "w", encoding="utf-8") as pf:
                json.dump(list(self.processed), pf)
        except:
            pass

    def stop(self):
        self.running = False

    def _sleep_with_check(self, total_seconds):
        interval = 0.5
        loops = int(total_seconds / interval)
        for _ in range(loops):
            if not self.running:
                return False
            time.sleep(interval)
        return True

    def extract_ip(self, msg):
        for hdr in msg.get_all("Received", []):
            m = re.search(r"\[([0-9]+\.[0-9]+\.[0-9]+\.[0-9]+)\]", hdr)
            if m:
                return m.group(1)
        return None

    def run(self):
        ctx = ssl.create_default_context()
        ctx.check_hostname = True
        ctx.verify_mode = ssl.CERT_REQUIRED
        try:
            imap = imaplib.IMAP4_SSL(self.account["server"], ssl_context=ctx)
            sock = imap.socket()
            sock.setsockopt(socket.SOL_SOCKET, socket.SO_KEEPALIVE, 1)
            imap.login(self.account["email"], self.account["password"])
            imap.select("INBOX")
            self.log(f"[{self.account['name']}] Connected to {self.account['server']}")
        except Exception as e:
            self.log(f"[{self.account['name']}] Login error: {e}")
            return

        while self.running:
            try:
                imap.noop()
                status, data = imap.uid("search", None, "UNSEEN")
            except Exception as e:
```

```
                self.log(f"[{self.account['name']}] Connection lost: {e}")
                time.sleep(1)
                try:
                    imap = imaplib.IMAP4_SSL(self.account["server"], ssl_context=ctx)
                    sock = imap.socket()
                    sock.setsockopt(socket.SOL_SOCKET, socket.SO_KEEPALIVE, 1)
                    imap.login(self.account["email"], self.account["password"])
                    imap.select("INBOX")
                    self.log(f"[{self.account['name']}] Reconnected")
                except Exception as e2:
                    self.log(f"[{self.account['name']}] Reconnect error: {e2}")
                    break
                continue

            if status != "OK" or not data or not data[0]:
                if not self._sleep_with_check(10):
                    break
                continue

            for raw_uid in data[0].split():
                if not self.running:
                    break
                uid = raw_uid.decode()
                start = time.time()

                try:
                    st, parts = imap.uid("fetch", uid, "(BODY.PEEK[])")
                except Exception as e:
                    self.log(f"[{self.account['name']}] Fetch error: {e}")
                    time.sleep(1)
                    break

                raw_bytes = None
                if parts and isinstance(parts, list):
                    for item in parts:
                        if isinstance(item, tuple):
                            raw_bytes = item[1]
                            break
                if not raw_bytes:
                    continue

                h = hashlib.md5(raw_bytes).hexdigest()
                if h in self.processed:
                    continue

                msg = email.message_from_bytes(raw_bytes)
                subj = msg.get("Subject", "(No subject)")
                frm = msg.get("From", "(No from)")
                self.log(f"[{self.account['name']}] Subject: {subj} From: {frm}")

                spf_pass = None
                dkim_pass = None
                if self.check_dkim_spf:
                    m = re.search(r"@([\w\.-]+)", frm)
                    dom = m.group(1) if m else ""
                    if dom == self.account_domain:
                        spf_pass = dkim_pass = True
                    else:
                        recs = msg.get_all("Received", [])
                        ip_addr = None
                        for r in recs:
                            m2 = re.search(r"\[([0-9]+\.[0-9]+\.[0-9]+\.[0-9]+)\]", r)
                            if m2:
                                ip_addr = m2.group(1)
                                break
```

```
            if ip_addr:
                try:
                    res, _ = spf.check2(ip_addr, dom, None)
                    spf_pass = (res.lower() == "pass")
                except:
                    spf_pass = None
            try:
                if msg.get("DKIM-Signature"):
                    dkim_pass = dkim.verify(msg.as_bytes())
            except:
                dkim_pass = None

        if spf_pass is False and dkim_pass is False:
            status = "dangerous"
            self.log(f"[{self.account['name']}] Deleted by SPF/DKIM")
            if self.auto_delete:
                imap.uid("copy", uid, "Spam")
                imap.uid("store", uid, "+FLAGS", "\\Deleted")
                imap.expunge()
            self.processed.add(h); self.save_processed()
            if self.stats_callback:
                self.stats_callback(self.account["name"], status)
            self.log(f"[{self.account['name']}] Time: {time.time()-start:.2f}s")
            continue

        m = re.search(r"@([\w\.-]+)", frm)
        dom = m.group(1) if m else ""
        try:
            ans = dns.resolver.resolve(f"_dmarc.{dom}", "TXT")
            txt = "".join(r.to_text().strip('"') for r in ans)
        except:
            txt = ""
        policy = {}
        for p in txt.split(";"):
            if "=" in p:
                k, v = p.split("=",1)
                policy[k.strip()] = v.strip()
        action = policy.get("p","none")
        if action == "reject":
            status = "dangerous"
            self.log(f"[{self.account['name']}] Deleted by DMARC")
            if self.auto_delete:
                imap.uid("copy", uid, "Spam")
                imap.uid("store", uid, "+FLAGS", "\\Deleted")
                imap.expunge()
            self.processed.add(h); self.save_processed()
            if self.stats_callback:
                self.stats_callback(self.account["name"], status)
            self.log(f"[{self.account['name']}] Time: {time.time()-start:.2f}s")
            continue
        if action == "quarantine":
            status = "suspicious"
            self.log(f"[{self.account['name']}] Flagged by DMARC")
            imap.uid("store", uid, "+FLAGS", "\\Flagged")
            imap.uid("store", uid, "-FLAGS", "\\Seen")
            if self.stats_callback:
                self.stats_callback(self.account["name"], status)

        ip = self.extract_ip(msg)
        if ip and not check_dnsbl(ip):
            status = "dangerous"
            self.log(f"[{self.account['name']}] Видалено через DNSBL ({ip})")
            if self.auto_delete:
                imap.uid("copy", uid, "Spam")
                imap.uid("store", uid, "+FLAGS", "\\Deleted")
```

```
                        imap.expunge()
                    self.processed.add(h);
                    self.save_processed()
                    if self.stats_callback:
                        self.stats_callback(self.account["name"], status)
                    self.log(f"[{self.account['name']}] Time: {time.time() - start:.2f}s")
                    continue

                if self.check_url:
                    body = ""
                    if msg.is_multipart():
                        for part in msg.walk():
                            if part.get_content_type()=="text/plain":
                                try:
                                    body += part.get_payload(decode=True).decode()
                                except:
                                    pass
                    else:
                        try:
                            body = msg.get_payload(decode=True).decode()
                        except:
                            pass
                    urls = re.findall(r"https?://[^\s'\"<>]+", body)
                    bad = False
                    for u in urls:
                        vok = True
                        if self.vt_key: vok = check_url_virustotal(self.vt_key,u)
                        gok = True
                        if self.gsb_key: gok = check_url_gsb(self.gsb_key,u)
                        oop = check_url_openphish(u)
                        if vok is not True or gok is not True or oop is not True:
                            bad = True; break
                    if bad:
                        status = "dangerous"
                        self.log(f"[{self.account['name']}] Deleted by URL")
                        if self.auto_delete:
                            imap.uid("copy", uid, "Spam")
                            imap.uid("store", uid, "+FLAGS", "\\Deleted")
                            imap.expunge()
                        self.processed.add(h); self.save_processed()
                        if self.stats_callback:
                            self.stats_callback(self.account["name"], status)
                        self.log(f"[{self.account['name']}] Time: {time.time()-
start:.2f}s")
                        continue

                if self.check_attachments and msg.is_multipart():
                    bada = False
                    for part in msg.walk():
                        fn = part.get_filename()
                        if fn:
                            fb = part.get_payload(decode=True)
                            ok = True
                            if self.vt_key:
                                ok = check_attachment_virustotal(self.vt_key,fn,fb)
                            if not ok:
                                status="dangerous"
                                self.log(f"[{self.account['name']}] Deleted by attach
{fn}")
                                if self.auto_delete:
                                    imap.uid("copy", uid, "Spam")
                                    imap.uid("store", uid, "+FLAGS", "\\Deleted")
                                    imap.expunge()
                                self.processed.add(h); self.save_processed()
                                if self.stats_callback:
```

```
                                    self.stats_callback(self.account["name"], status)
                                self.log(f"[{self.account['name']}] Time: {time.time()-
start:.2f}s")
                                bada=True
                                break
                    if bada:
                        continue

                if self.check_phrases:
                    body2=""
                    if msg.is_multipart():
                        for part in msg.walk():
                            if part.get_content_type()=="text/plain":
                                try:
                                    body2+=part.get_payload(decode=True).decode()
                                except:
                                    pass
                    else:
                        try:
                            body2=msg.get_payload(decode=True).decode()
                        except:
                            pass
                    flagg=False
                    for ph in self.phrases:
                        if ph.lower() in body2.lower():
                            status="suspicious"
                            self.log(f"[{self.account['name']}] Flagged by phrase {ph}")
                            imap.uid("store", uid, "+FLAGS", "\\Flagged")
                            imap.uid("store", uid, "-FLAGS", "\\Seen")
                            self.processed.add(h); self.save_processed()
                            if self.stats_callback:
                                self.stats_callback(self.account["name"], status)
                            self.log(f"[{self.account['name']}] Time: {time.time()-
start:.2f}s")
                            flagg=True
                            break
                    if flagg:
                        continue

                status="safe"
                self.log(f"[{self.account['name']}] Safe")
                imap.uid("store", uid, "+FLAGS", "\\Answered")
                imap.uid("store", uid, "-FLAGS", "\\Seen")
                self.processed.add(h); self.save_processed()
                if self.stats_callback:
                    self.stats_callback(self.account["name"], status)
                self.log(f"[{self.account['name']}] Time: {time.time()-start:.2f}s")

            if not self._sleep_with_check(10):
                break

        try:
            imap.logout()
        except:
            pass
        self.log(f"[{self.account['name']}] Monitoring stopped")
```

apis.py

```
import socket
import requests
import time
from functools import lru_cache

@lru_cache(maxsize=1)
def load_openphish_feed():
```

```
    try:
        resp = requests.get("https://openphish.com/feed.txt", timeout=30)
        resp.raise_for_status()
        lines = resp.text.splitlines()
        return set(line.strip() for line in lines if line.strip())
    except Exception:
        return set()

def check_url_virustotal(api_key, url, timeout=60, interval=5):

    if not api_key:
        return True
    headers = {"x-apikey": api_key}
    try:

        resp = requests.post(
            "https://www.virustotal.com/api/v3/urls",
            headers=headers,
            data={"url": url}
        )
        resp.raise_for_status()
        url_id = resp.json()["data"]["id"]
        waited = 0
        while waited < timeout:
            rpt = requests.get(
                f"https://www.virustotal.com/api/v3/analyses/{url_id}",
                headers=headers
            )
            rpt.raise_for_status()
            data = rpt.json()
            if data["data"]["attributes"]["status"] == "completed":
                stats = data["data"]["attributes"]["stats"]
                return stats.get("malicious", 0) == 0
            time.sleep(interval)
            waited += interval
        return True
    except Exception:
        return None

def check_url_gsb(api_key, url):

    if not api_key:
        return True
    api_url = f"https://safebrowsing.googleapis.com/v4/threatMatches:find?key={api_key}"
    body = {
        "client": {"clientId": "email_monitor", "clientVersion": "1.0"},
        "threatInfo": {
            "threatTypes": ["MALWARE", "SOCIAL_ENGINEERING", "UNWANTED_SOFTWARE"],
            "platformTypes": ["ANY_PLATFORM"],
            "threatEntryTypes": ["URL"],
            "threatEntries": [{"url": url}]
        }
    }
    try:
        resp = requests.post(api_url, json=body)
        resp.raise_for_status()
        result = resp.json()

        return not bool(result.get("matches"))
    except Exception:
        return None

def check_url_openphish(url):
```

```
    try:
        feed = load_openphish_feed()
        return url not in feed
    except Exception:
        return None

def check_attachment_virustotal(api_key, filename, file_bytes, timeout=60, interval=5):

    if not api_key:
        return True
    headers = {"x-apikey": api_key}
    try:
        files = {"file": (filename, file_bytes)}
        resp = requests.post(
            "https://www.virustotal.com/api/v3/files",
            headers=headers,
            files=files
        )
        resp.raise_for_status()
        file_id = resp.json()["data"]["id"]
        waited = 0
        while waited < timeout:
            rpt = requests.get(
                f"https://www.virustotal.com/api/v3/analyses/{file_id}",
                headers=headers
            )
            rpt.raise_for_status()
            data = rpt.json()
            if data["data"]["attributes"]["status"] == "completed":
                stats = data["data"]["attributes"]["stats"]
                return stats.get("malicious", 0) == 0
            time.sleep(interval)
            waited += interval
        return True
    except Exception:
        return None

@lru_cache(maxsize=1024)
def check_dnsbl(ip, zones=None):

    if zones is None:
        zones = ["zen.spamhaus.org", "bl.spamcop.net", "dbl.spamhaur.org"]
    rev = ".".join(ip.split(".")[::-1])
    for zone in zones:
        try:
            socket.gethostbyname(f"{rev}.{zone}")
            return False
        except socket.gaierror:
            continue
    return True
```

phrases_editor.py

```
import tkinter as tk
from tkinter import ttk, simpledialog

class PhrasesEditor(tk.Toplevel):
    def __init__(self, parent, phrases_list, callback):
        super().__init__(parent)
        self.title("Редактор фраз")
        self.geometry("400x400")
        self.callback = callback
        self.phrases = phrases_list.copy()
        self.listbox = tk.Listbox(self)
        self.listbox.pack(fill=tk.BOTH, expand=True, padx=10, pady=10)
        for phrase in self.phrases:
```

```
            self.listbox.insert(tk.END, phrase)
        self.listbox.bind("<Double-Button-1>", self.edit_phrase)
        btn_frame = ttk.Frame(self)
        btn_frame.pack(pady=5)
        ttk.Button(btn_frame, text="Додати", command=self.add_phrase).grid(row=0, column=0,
padx=5)
        ttk.Button(btn_frame, text="Видалити", command=self.delete_phrase).grid(row=0,
column=1, padx=5)
        ttk.Button(btn_frame, text="Зберегти", command=self.save_phrases).grid(row=0,
column=2, padx=5)

    def add_phrase(self):
        new_phrase = simpledialog.askstring("Додати", "Введіть нову фразу:", parent=self)
        if new_phrase:
            self.phrases.append(new_phrase)
            self.listbox.insert(tk.END, new_phrase)

    def delete_phrase(self):
        sel = self.listbox.curselection()
        if sel:
            idx = sel[0]
            self.listbox.delete(idx)
            del self.phrases[idx]

    def edit_phrase(self, event):
        sel = self.listbox.curselection()
        if sel:
            idx = sel[0]
            cur = self.phrases[idx]
            new = simpledialog.askstring("Редагувати", "Змініть фразу:", initialvalue=cur,
parent=self)
            if new:
                self.phrases[idx] = new
                self.listbox.delete(idx)
                self.listbox.insert(idx, new)

    def save_phrases(self):
        self.callback(self.phrases)
        self.destroy()
```

Додаток В

ІЛЮСТРАТИВНА ЧАСТИНА

ЗАСІБ ДЛЯ ПЕРЕВІРКИ ПОВІДОМЛЕНЬ ЕЛЕКТРОННОЇ ПОШТИ ЩОДО ОЗНАК ФІШИНГОВОЇ АТАКИ

Виконав: студент 4 курсу групи 1БКС-21б
спеціальності 125 Кібербезпека
Володимир РОСТЕЦЬКИЙ

Керівник: к. т. н., доцент каф. ЗІ
Олеся ВОЙТОВИЧ

«16» червня 2025 р.

Результати порівняння існуючих засобів захисту від фішингових атак

Засоби	**Опис**	**Методи перевірки**	**Переваги**	**Недоліки**
Defender for Office 365	Корпоративна безпека в Microfost 365	SPF, DKIM, DMARC, аналіз посилань та вкладень	Інтеграція з M365, звітність	Працює лише з M365
Google Workspace Email Protection	Вбудований у Gmail захист із ШІ-аналітикою	SPF, DKIM, аналіз вкладень і URL, ШІ-аналіз поведінки	Автоматична активація, простота	Тільки для Gmail
Proofpoint Essentials	Шлюз фільтрації між зовнішнім трафіком і поштою	SPF, DKIM, DMARC, сканування вкладень та посилань	Підтримка великої кількості поштових сервісів	Платний зовнішній шлюз
Mimecast Email Security	Комплексна платформа з контент-фільтрацією	Фільтрація контенту, перевірка посилань та вкладень, QR-аналіз	Підтримує хмару та локальні сервери	Висока вартість
IRONSCALES	Платформа для аналізу повідомлень із самонавчальними моделями	Аналіз підміни адрес, фальшивих форм і URL, поведінковий аналіз	Адаптивні моделі, інтерактивність	Лише для M365 та Gmail
Barracuda Email Protection	Хмарне рішення з антифішингом і антиспамом	Перевірка посилань та вкладень, антиспам	Підтримка всіх популярних платформ	Залежить від Barracuda Cloud
Abnormal Security	Платформа для перевірки повідомлень з штучним інтелектом	Поведінковий аналіз, шаблонний аналіз	Висока точність у виявленні BEC	Тільки для cloud-платформ
Rspamd	Безкоштовний Milter-фільтр для локальних SMTP-серверів	SPF, DKIM, аналіз URL	Безкоштовний, гнучке налаштування	Потребує ручної конфігурації

Результати порівняння технічних методів захисту від фішингових атак

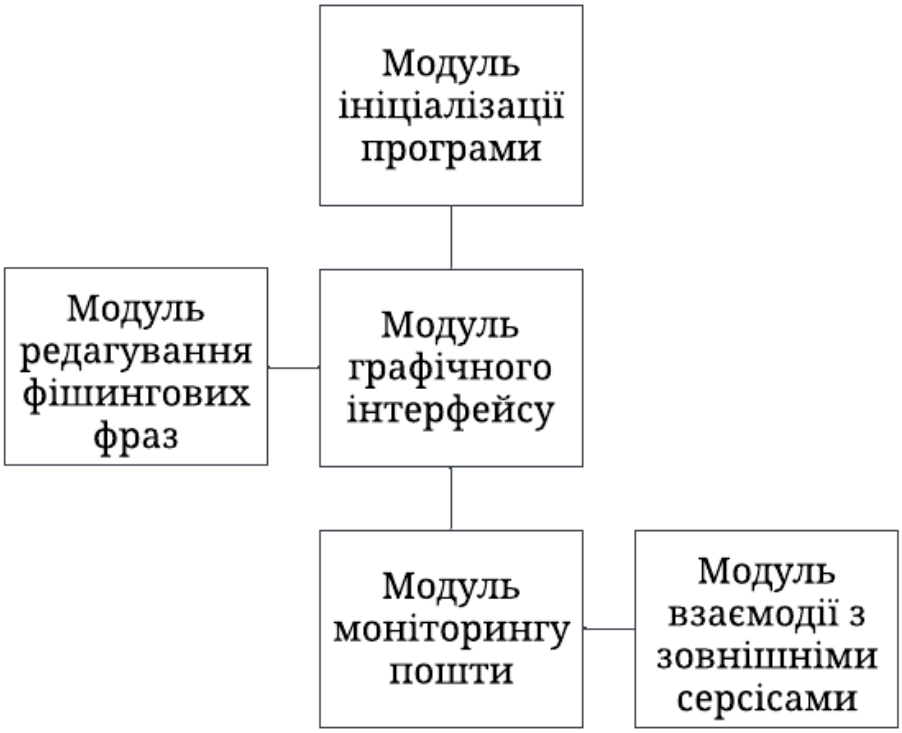
Метод захисту	**Принцип роботи**	**Переваги**	**Недоліки**
Двофакторна автентифікація (2FA/MFA)	Додатковий рівень підтвердження (код, апаратний ключ)	Дуже високий рівень захисту навіть при викраденні пароля	Потребує додаткових дій від користувача
Антивірус з антифішингом	Сканування листів і посилань, блокування відомих загроз	Простота використання, регулярні оновлення	Може пропускати нові або складні цільові атаки
Поштові фільтри	Аналізують вхідні листи, блокують підозрілі повідомлення	Зменшують кількість фішингових листів, використовують машинне навчання	Можливі помилкові спрацьовування, залежність від налаштувань
SPF	Перевірка IP відправника за списком у DNS	Захищає від підробки адреси відправника (спуфінгу)	Не працює при пересиланні, не захищає вміст листа
DKIM	Додає цифровий підпис для підтвердження цілісності листа	Гарантує, що лист не змінено і надійшов від власника домену	Вимагає налаштування ключів, не працює при зміні листа
DMARC	Встановлює політику обробки листів, що не пройшли SPF/DKIM	Дозволяє контролювати підроблені листи, отримувати звіти	Потрібна правильна конфігурація SPF і DKIM, ризик блокування офіційних листів
Браузерні розширення та URL-фільтри	Попереджають про небезпечні сайти, блокують переходи	Знижують ризик переходу на фішингові ресурси	Не гарантують повний захист, можуть блокувати офіційні сайти
Менеджери паролів	Генерують і зберігають складні паролі, не автозаповнюють форми на підозрілих сайтах	Підвищують кібергігієну, знижують ризик компрометації паролів	Не захищають від фішингу напряму
Моніторинг доменів і мережевого трафіку	Відстеження підозрілих доменів і блокування доступу	Проактивний захист на рівні мережі, виявлення нових загроз	Вартість, потреба у кваліфікованому адмініструванні
Антифішингові платформи на основі ШІ	Використання машинного навчання для виявлення нових і складних атак	Висока точність, автоматизація реагування	Висока вартість, потреба у налаштуванні і підтримці

Основні ознаки фішингових листів

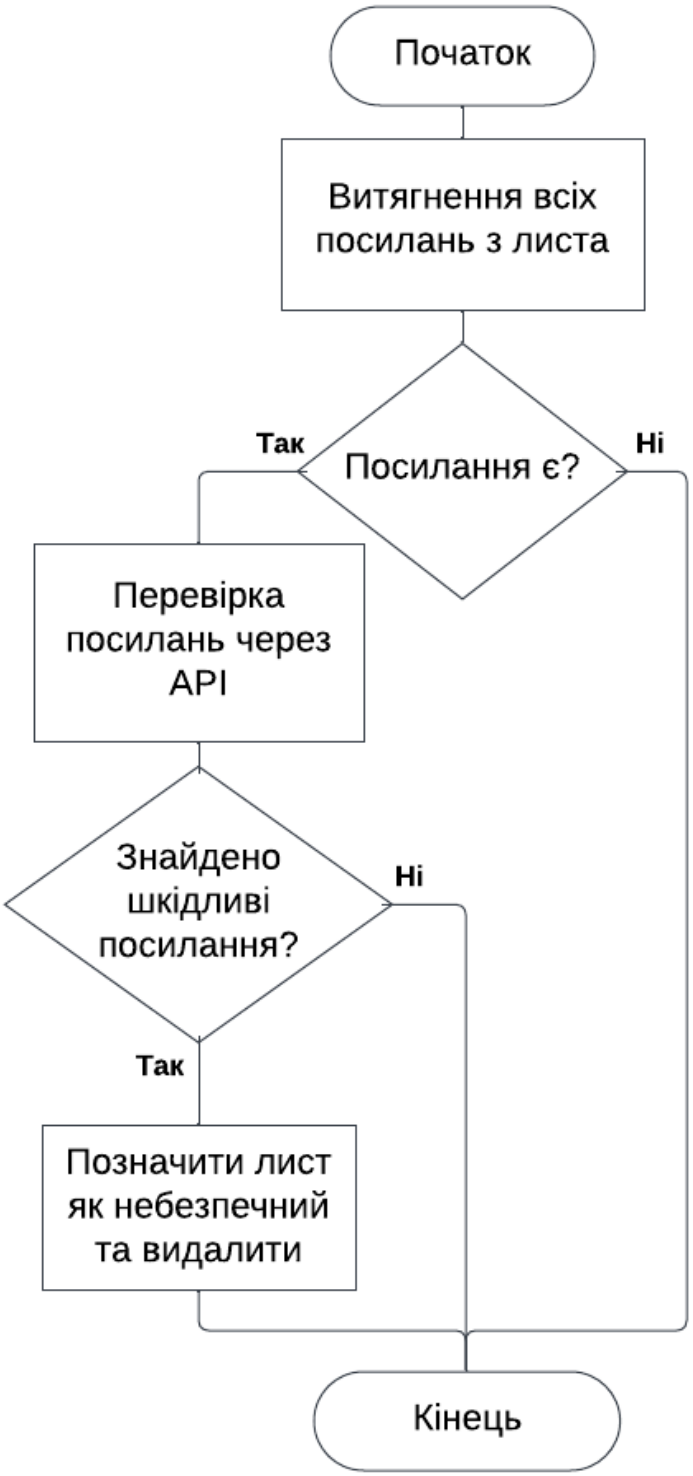
Ознака	**Короткий опис**	**Метод виявлення**
Підозріла адреса	Адреса містить помилки чи змінені символи	SPF/DKIM/DMARC; перевірка IP-домену
Граматичні помилки та відсутність персоналізації	Орфографічні/стилістичні помилки, загальне звернення	NLP-аналіз помилок; виявлення імені
Запит конфіденційної інформації	Заклики надати паролі чи дані карток	Контент-аналіз; пошук ключових фраз; перевірка URL
Підозрілі посилання	URL із незначними змінами, що ведуть на фейкові сайти	Аналіз структури URL; база фішингових ресурсів; перевірка URL
Приховані посилання	Текст посилання не відповідає фактичному посиланню	Перевірка тексту проти href
Невідповідність тексту та посилання	Посилання не відповідає тексту в листі	Порівняння тексту на посиланню
Терміновість або погрози	Фрази з погрозами або терміновими закликами	Пошук ключових слів
Вкладення з шкідливим програмним забезпеченням	Вкладення містять шкідливе ПЗ	Антивірусне сканування
Замасковане вкладення	Вкладення має подвійне або хибне розширення	Антивірусне сканування; перевірка розширення

Схема процесу перевірки на ознаки фішингового листа

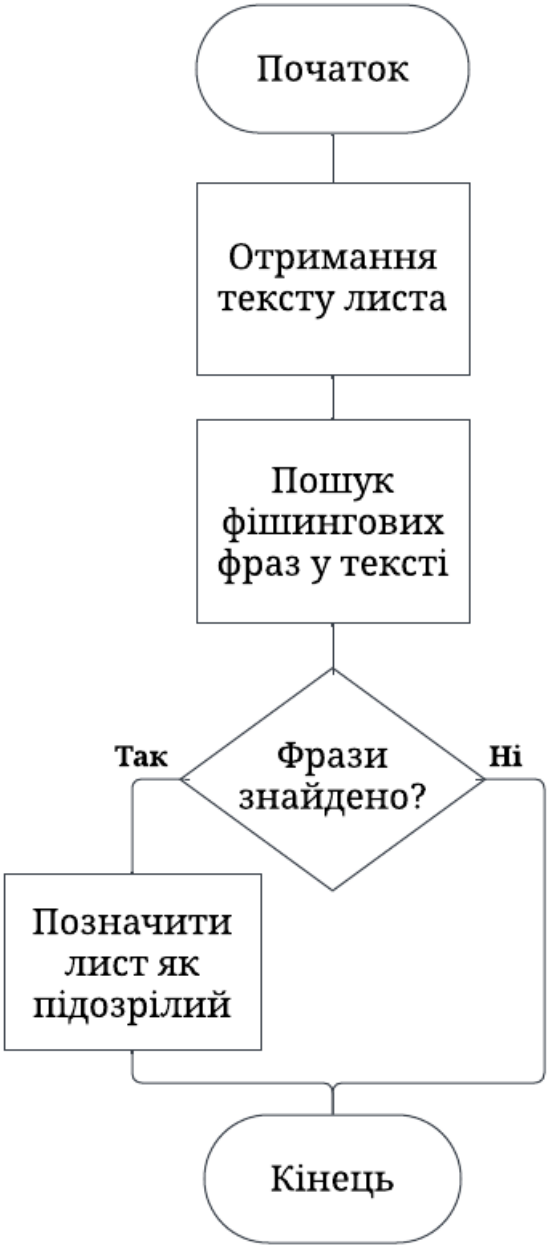
Структурна схема архітектури модулів



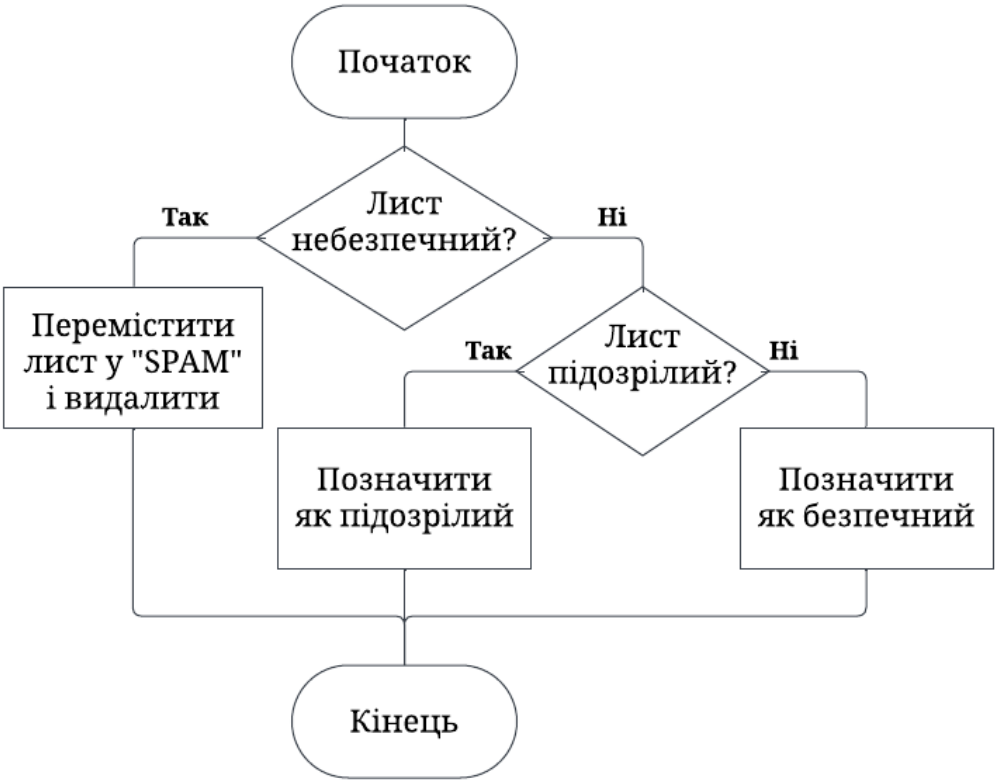
Алгоритм перевірки посилань



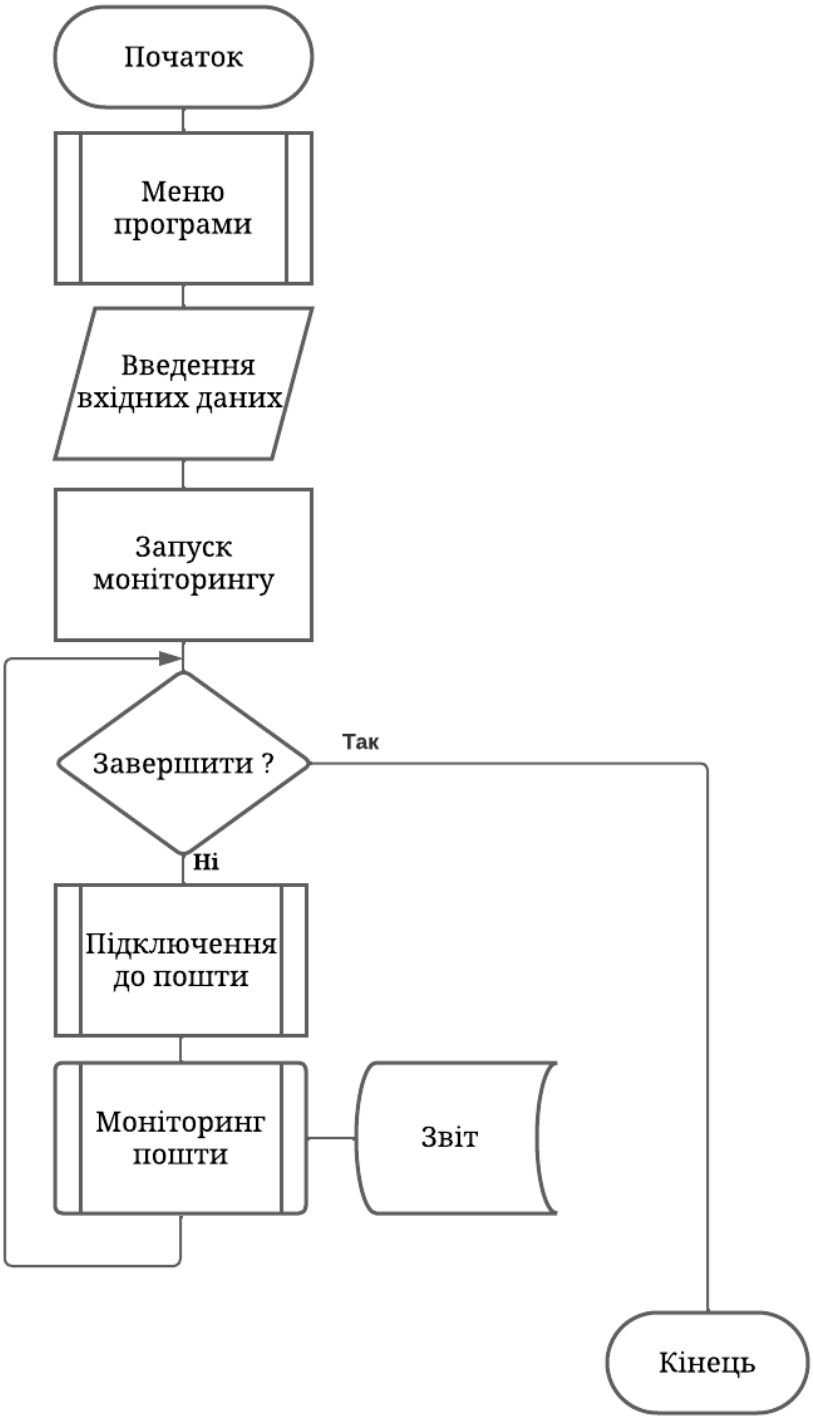
Алгоритм аналізу фішингових фраз



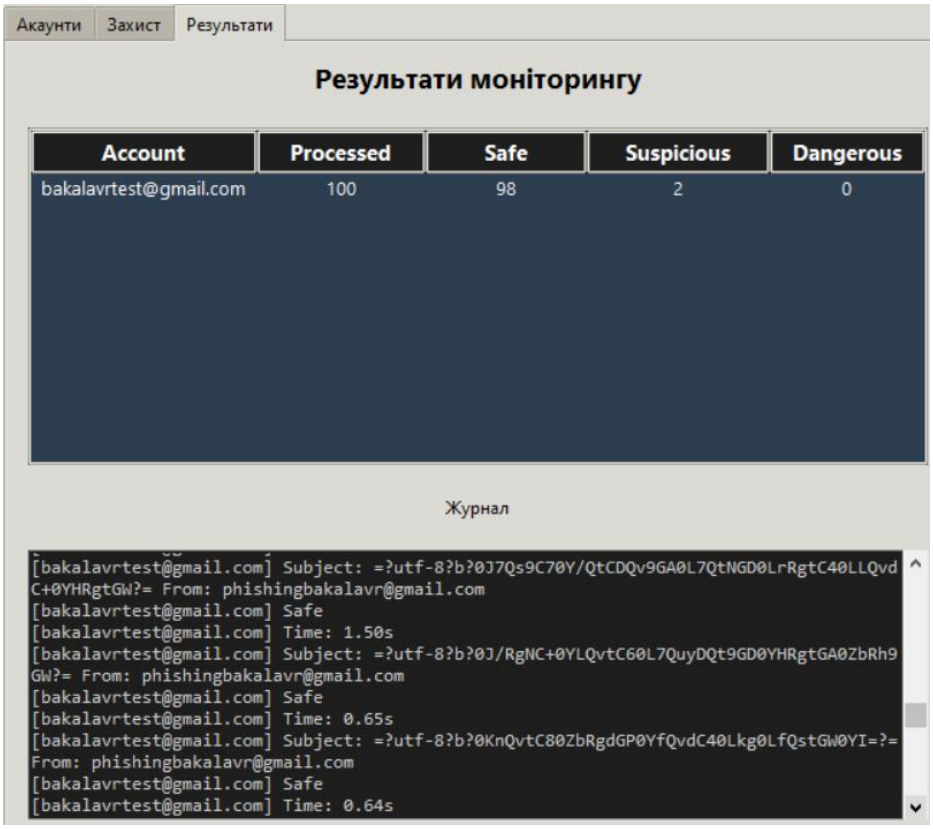
Алгоритм класифікації листа за рівнем загроз



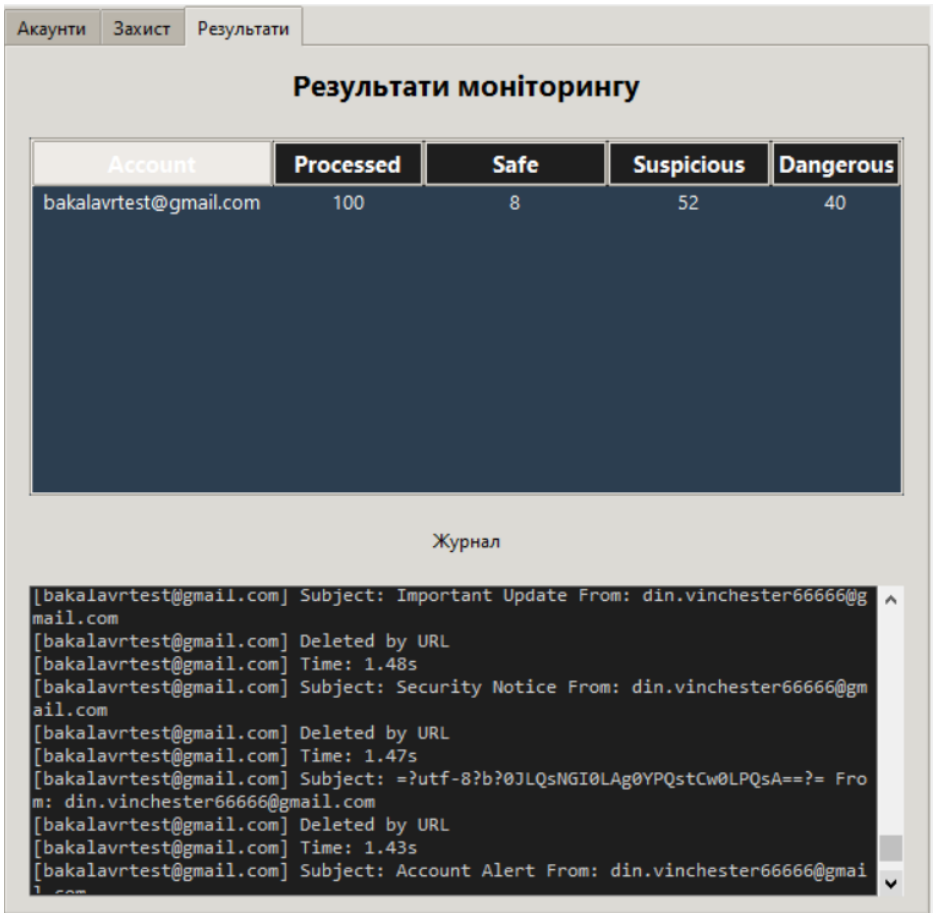
Алгоритм взаємодії користувача з програмою



Результат перевірки 100 безпечних листів



Результат перевірки 100 фішингових листів



Результати роботи засобу

	Надіслані	**Визначені як безпечні**	**Визначені як підозрілі**	**Визначені як небезпечні**	**Ефективність виявлення**
Надіслані безпечні	100	98	2	0	98%
Надіслані небезпечні	100	8	52	40	92%