Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

МАГІСТЕРСЬКА КВАЛІФІКАЦІЙНА РОБОТА
на тему:
«МЕТОД ТА ЗАСІБ ВИЯВЛЕННЯ ФІШИНГОВИХ ВЕБ-САЙТІВ»

Виконав: студент 2 курсу, групи БС-23 м
спеціальності 125 Кібербезпека та захист
інформації
_____ Іван САВЧУК
Керівник: к. т. н., доцент каф. ЗІ
_____ Юрій БАРИШЕВ
«13» 12 2024 р.
Опонент: к. т. н., доцент, доцент каф. ПЗ
_____ Денис КАТЕЛЬНІКОВ
«13» 12 2024 р.

Допущено до захисту
Завідувач кафедри ЗІ
д. т. н., проф.
_____ Володимир ЛУЖЕЦЬКИЙ
«13» 12 2024 р.

Вінниця ВНТУ – 2024 року

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

ЗАТВЕРДЖУЮ
Завідувач кафедри ЗІ,
д. т. н., професор.
_____Володимир ЛУЖЕЦЬКИЙ
«18» 09 2024 року

З А В Д А Н Н Я
НА МАГІСТЕРСЬКУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТУ

Савчуку Івану Борисовичу

1. Тема роботи: «Метод та засіб виявлення фішингових веб-сайтів»

   керівник роботи: Баришев Юрій Володимирович, к. т. н., доцент кафедри ЗІ, затверджені наказом ректора ВНТУ від 17 вересня 2024 року №310.

2. Строк подання студентом роботи 23 грудня 2024 р.

3. Вихідні дані до роботи:

   – засіб виявлення фішингових веб-сайтів
   – метод виявлення фішингових веб-сайтів.

4. Зміст текстової частини: Вступ. 1. Аналіз джерел за напрямком фішингу. 2. Метод виявлення фішингових веб-сайтів. 3. Розробка засобу виявлення фішингових веб-сайтів. 4. Тестування методу та засобу виявлення фішингових веб-сайтів 5. Економічний розділ. Висновки. Список використаних джерел. Додатки.

5. Перелік ілюстративного матеріалу: Тенденції розвитку загрози фішингових веб-сайтів, Порівняльний аналіз методів виявлення фішингових веб-сайтів, Математичний опис процесу виявлення фішингових веб-сайтів, Метод виявлення фішингу, Результати експертнорго опитування, Результати блокового тестування засобу, Архітектура засобу, Узагальнений алгоритм роботи засобу, Результати експериментального дослідження засобу виявлення фішингових веб-сайтів, Показники економічної ефективності.

Результати експериментального дослідження засобу виявлення фішингових веб-сайтів, Показники економічної ефективності.

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		Завдання видав	Завдання прийняв
1	Юрій БАРИШЕВ , к.т.н., доц. каф.ЗІ	10.09.24	23.09.24
2	Юрій БАРИШЕВ , к.т.н., доц. каф.ЗІ	24.09.24	12.10.24
3	Юрій БАРИШЕВ , к.т.н., доц. каф.ЗІ	14.10.24	11.11.24
4	Юрій БАРИШЕВ, к.т.н., доц. каф.ЗІ	19.10.24	29.10.24
5	Ольга РАТУШНЯК, к. т. н., доц., доц. каф. ЕПВМ	09.11.24	14.11.24

7. Дата видачі завдання: 9 вересня 2024 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів магістерської кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Аналіз завдання. Вступ	1.09.2024 – 09.09.2024	
2	Аналіз інформаційних джерел за напрямком магістерської кваліфікаційної роботи	10.09.2024 – 16.09.2024	
3	Науково-технічне обгрунтування	17.09.2024 – 23.09.2024	
4	Розробка технічного завдання	24.09.2024 – 12.10.2024	
5	Аналіз стандартів виявлення фішингових сайтів	14.10.2024 – 11.11.2024	
6	Аналіз та формування вимог до методики	12.10.2024 – 18.10.2024	
7	Розробка методики аудиту	19.10.2024 – 29.10.2024	
8	Застосування методу виявлення фішингу	03.11.2024 – 08.11.2024	
9	Розробка розділу економічного обгрунтування	09.11.2024 – 14.11.2024	
10	Аналіз виконання ТЗ, висновки	15.11.2024-19.11.2024	
11	Оформлення пояснювальної записки	20.11.2024-26.11.2024	
12	Попередній захист та доопрацювання МКР	28.11.2024 – 1.12.2024	
13	Перевірка магістерської роботи на наявність текстових запозичень	02.12.2024 – 14.12.2024	
14	Представлення МКР до захисту, рецензування	11.12.2024-16.12.2024	
15	Захист МКР	17.12.2024-20.12.2024	

Студент _____ Іван САВЧУК

Керівник роботи _____ Юрій БАРИШЕВ

АНОТАЦІЯ

УДК 004.056.54

Савчук І. Метод та засіб виявлення фішингових веб-сайтів. Магістерська кваліфікаційна робота за спеціальністю 125 Кібербезпека та захист інформації, освітня програма – Безпека інформаційних і комунікаційних систем.

Вінниця: ВНТУ, 2024. – 98 с

Укр. Мовою. Бібліогр.: назв 27; рис.: 36; табл.: 19.

Магістерська кваліфікаційна робота присвячена розробці методу та засобу виявлення фішингових веб-сайтів. Здійснено аналіз загроз фішингових веб-сайтів, їх основних ознак, а також відомих методів захисту від фішингових атак. Здійснено визначення технічних ознак фішингових веб-сайтів. На основі отриманих даних здійснено математичний опис процесу виявлення фішингових веб-сайтів. Удосконалено метод виявлення фішингових веб-сайтів, який враховує обставини їх створення. Розроблено програмний засіб та протестовано на коректність роботи. Проведено оцінку економічної доцільності реалізації засобу.

Ілюстративна частина складається з 10 плакатів з демонстрацією проведених досліджень розроблених алгоритмів, тестування та визначення економічних доцільності

Ключові слова: фішинг, кібербезпека, соціальна інженерія, ознаки фішингу, веб-сайти, розширення браузера.

ABSTRACT

Savchuk I. Method and Means of Detecting Phishing Websites. Master's Qualification Thesis in the specialty 125 Cybersecurity and Information Protection, educational program – Information and Communication Systems Security.

Vinnytsa: VNTU, 2024. – 98 p

In Ukrainian language. Bibliographer: 27 titles; 36 fig; tabl.: 19.

The master's thesis is devoted to the development of a method and a tool for detecting phishing websites. It includes an analysis of phishing website threats, their key characteristics, and existing methods of protection against phishing attacks. The technical features of phishing websites were identified, and a mathematical description of the detection process was developed based on the obtained data. The detection method was improved by considering the circumstances of their creation. Additionally, a software tool was developed and tested for functionality, and an economic feasibility assessment of its implementation was conducted.

The illustrative part consists of 10 posters demonstrating the conducted research, developed algorithms, testing processes, and the assessment of economic feasibility.

Keywords: phishing, cybersecurity, social engineering, phishing indicators, websites, browser extension..

ЗМІСТ

ВСТУП

З-поміж кіберзагроз вирізняються своєю поширеністю загрози, що виникають при користуванні Інтернетом. Віруси, як правило, піддаються прогнозуванню завдяки наявності значних баз даних відомих шкідливих програм і ефективних засобів їхньої нейтралізації, таких як антивірусні програми. Однак окремо варто виділити загрози, що виникають через методи соціальної інженерії, зокрема фішингові вебсайти. Фішинговий сайт — це підроблений вебсайт, створений з метою обману користувачів, щоб змусити їх надати конфіденційну інформацію, таку як паролі, номери банківських карток або інші особисті дані, видаючи себе за легітимний ресурс [1]. Основна проблема полягає в тому, що людський фактор важко вирахувати, та, особливо важко, спрогнозувати. Суттю таких загроз є маніпулювання поведінкою користувачів, щоб отримати конфіденційну інформацію. Виявити та попередити їх складніше, ніж традиційні вірусні атаки.

У цьому контексті фішингові атаки становлять особливу небезпеку, оскільки їх складніше ідентифікувати за допомогою звичайних засобів кіберзахисту, оскільки вони експлуатують психологічні, а не технологічні вразливості. Від початку повномасштабного вторгнення у лютому 2022-го року кількість фішингових сайтів зросла майже удвічі згідно з даними Асоціації «ЄМА» та залишається одним з лідерів загроз для користувачів інтернету. Враховуючи, що данні містять лише виявлені та заблоковані ресурси, реальна цифра активних фішингових сайтів на сьогоднішній момент може бути значно більшою. Саме фішингові сайти становлять 88% заблокованих ресурсів [2]. Решта 12% припадають на шахрайські інтернет-крамниці, шахрайські схеми заробітку, шахрайство з «інвестиціями». Щодня у світі створюється близько 7,850 нових фішингових вебсайтів. Це означає, що новий фішинговий сайт з'являється кожні 11 секунд [3].

Таким чином, фішингові сайти продовжують залишатися однією з основних загроз в Інтернеті, постійно еволюціонуючи та створюючи серйозну небезпеку для

користувачів. Враховуючи швидкість їх створення та маніпулятивні методи соціальної інженерії, для ефективного захисту необхідно використовувати сучасні методи та засоби для вивлення фішингових веб-сайтів.

Об’єктом дослідження є процес захисту від методів соціальної інженерії.

Предметом дослідження є методи та засоби виявлення фішингових веб-сайтів.

Метою даної магістерської кваліфікаційної роботи є покращення захисту користувачів від фішингових веб-сайтів у мережі Інтернет.

Для досягнення мети потрібно розв’язати такі завдання:

- проаналізувати основні загрози, що виникають через фішингові сайти;
- виконати аналіз інформаційних джерел з метою дослідження наявних методів і стратегій фішингових атак та класифікації їх різновидів.
- ідентифікувати основні риси, притаманні фішинговим сайтам;
- розробити архітектуру системи захисту від фішингу;
- розробити алгоритм для автоматичного аналізу підозрілих сайтів;
- реалізувати розроблені алгоритми як засіб;
- експериментально оцінити показники ефективності засобу;
- визначити економічну доцільність розробки.

Наукова новизна полягає в удосконалені методу виявлення фішингового веб сайтів, який на відміну від відомих враховує не лише контент сайту але й обставини його створення, що дозволяє зменшити частку невиявлених фішингових сайтів.

Практична цінність роботи полягає в такому:

- розширені браузера для виявлення фішингових веб-сайтів;
- розробленому бек-енд модулі та його API для аналізу сайтів щодо наявності фішингу;
- результатах опрацювання опитування експертів щодо ознак фішингових сайтів станом на 2024 рік.

Публікації результатів магістерської кваліфікаційної роботи. Результати магістерської кваліфікаційної роботи доповідалися на 3 конференціях з публікацією тез доповідей:

- Міжнародна науково-практична інтернет-конференція Молодь в науці: дослідження, проблеми, перспективи (МН-2025) [4];
- онлайн-конференція Світ наукових досліджень 2024 [5]
- Результати роботи доповідались на LII Науково-технічній конференції факультету інформаційних технологій та комп'ютерної інженерії (2023) конференції та опубліковані як тези та матеріал доповіді [6].

1 АНАЛІЗ ФІШИНГОВИХ АТАК

1.1 Аналіз впливу загрози фішингових веб-сайтів на кібербезпеку

Загрози, спрямовані на технічну складову, такі як шкідливе програмне забезпечення, DDoS-атаки та зломи через уразливості програмного забезпечення, сьогодні краще контролюються завдяки сучасним інструментам кіберзахисту. Однак, людський фактор залишається складним для захисту, оскільки методи соціальної інженерії, зокрема фішингові атаки, експлуатують психологічні вразливості людей, що робить їх важчими для виявлення та попередження.

Фішингові атаки є однією з найпоширеніших кіберзагроз цього виду, оскільки їхня небезпека полягає не стільки в технічних аспектах, скільки у маніпулюванні користувачами. Зловмисники використовують методи соціальної інженерії для обману, змушуючи людей передавати конфіденційну інформацію, таку як паролі або номери банківських карток. Це відрізняє фішинг від традиційних технічних загроз, оскільки він спрямований безпосередньо на людей, а не на системи, що значно ускладнює його виявлення стандартними засобами кіберзахисту.

З початку 2022 року фішингові атаки та кількість фішингових сайтів почала стрімко зростати і у 2024 ця тенденція продовжується. За даними Anti-Phishing Working Group, кількість фішингових атак значно перевищила попередні показники, а кількість новостворених фішингових сайтів досягла рекордних значень [7]. Згідно з поквартальними звітами APWG, було виявлено суттєве зростання як у кількості атак, так і в кількості нових фішингових ресурсів. Таблиця 1.1 відображає кількість фішингових атак та створених фішингових сайтів по місяцях, наочно демонструючи масштаб загрози

Таблиця 1.1 – Кількість фішингових атак за 2022 рік

Місяць	Кількість фішингових сайтів	Кількість фішингових email-тем	Бренди-цілі
01-2022	331,698	15,275	608
02-2022	309,979	14,176	621
03-2022	384,291	24,187	673
04-2022	362,852	21,540	621
05-2022	353,242	20,339	612
06-2022	381,717	23,550	637
07-2022	425,112	64,696	621
08-2022	430,141	38,228	612
09-2022	415,630	23,994	637
10-2022	356,538	22,750	477
11-2022	350,776	24,621	442
12-2022	370,187	20,642	420
Загальна кількість	4,786,161		

Для визначення тенденцій у динаміці атак, а також виявлення пікових точок та подальшого детального аналізу, слід розглянути кожен тип фішингових атак окремо. Для більш детального аналізу кількості фішингових атак, був побудований графік на рис 1.1 .

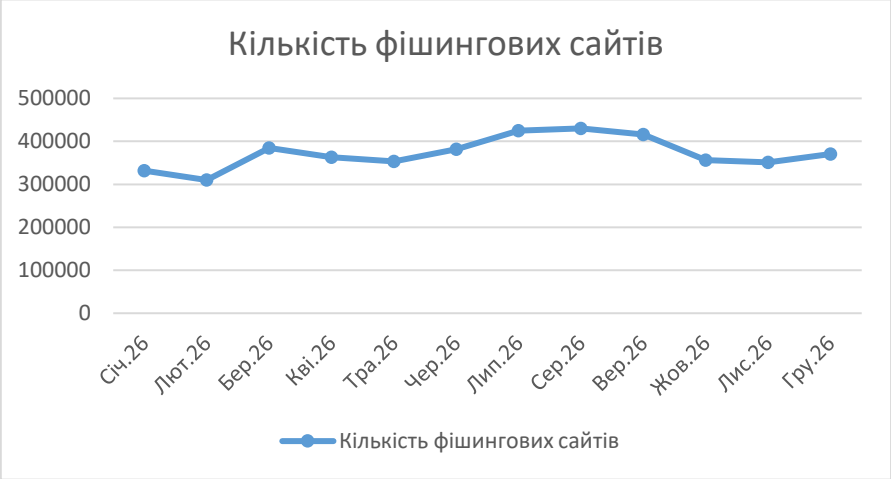


Рисунок 1.1 – Кількість фішингових сайтів за 2022 рік

Створений графік демонструє непостійність фішингових атак протягом року має як зростання, так і спад. Пік атак припадає на найтепліші пори року, саме період з червня по вересень. Це може свідчити про ймовірне підвищення використання вразливості в напрямку фішингу, через сезонні зростання онлайн активності. Зростання активності у даний період ймовірно пов’язане з підвищеною кількістю покупок, пов'язаних з туристичною та розважальною індустріями, коли користувачі активніше бронюють квитки, житло та інші послуги онлайн, що створює більше можливостей для кіберзловмисників використовувати фішингові атаки з метою викрадення конфіденційних даних [8]. Аналогічно до кількості фішингових сайтів, наведено графік кількості фішингових email-тем на рис 1.2.

Рисунок 1.2 – Графік кількості фішингових email-тем за 2022 рік

Як і у випадку з фішинговими сайтами, різке збільшення кількості шкідливих листів найбільш ймовірно пов’язане з сезоном відпусток та підвищеним попитом на розважальну індустрію. Схожа тенденція спостерігається також серед брендів, що стають цілями фішингових атак. Зокрема, відносно невеликий спад кількості брендів, під які маскуються фішингові сайти та листи, може бути пов'язаний із вжиттям додаткових засобів захисту зі сторони компаній.

Аналогічним чином слід розглянути кількість фішингових атак за 2023 рік. Для підвищення актуальності даних додано інформацію за перший квартал 2024 року. Кількість атак, здійснених методом фішингу за 2023 рік, наведено у таблиці 1.2.

Таблиця 1.2 – Кількість фішингових атак за 2023 рік

Місяць	Кількість фішингових сайтів	Кількість фішингових email-тем	Бренди-цілі
01-2023	495,690	40,863	561
02-2023	509,394	45,259	549
03-2023	619,060	40,742	576
04-2023	597,789	41,083	544
05-2023	381,572	30,717	521
06-2023	306,847	22,610	498
07-2023	327,294	29,110	477
08-2023	331,962	32,162	499
09-2023	340,700	32,740	508
10-2023	356,538	22,750	477
11-2023	350,776	24,621	442
12-2023	370,187	20,642	420
01-2024	358,107	50,837	314
02-2024	314,974	24,086	309
03-2024	290,913	41,550	301
Загальна кількість	6,448,238		

З наведеної інформації видно, що активність фішингових атак залишається значною протягом року, спостерігаються певні коливання в окремі періоди. Для кращого розуміння змін у динаміці фішингових сайтів, наведено графік за 2022 і 2023 роки на рис 1.3.

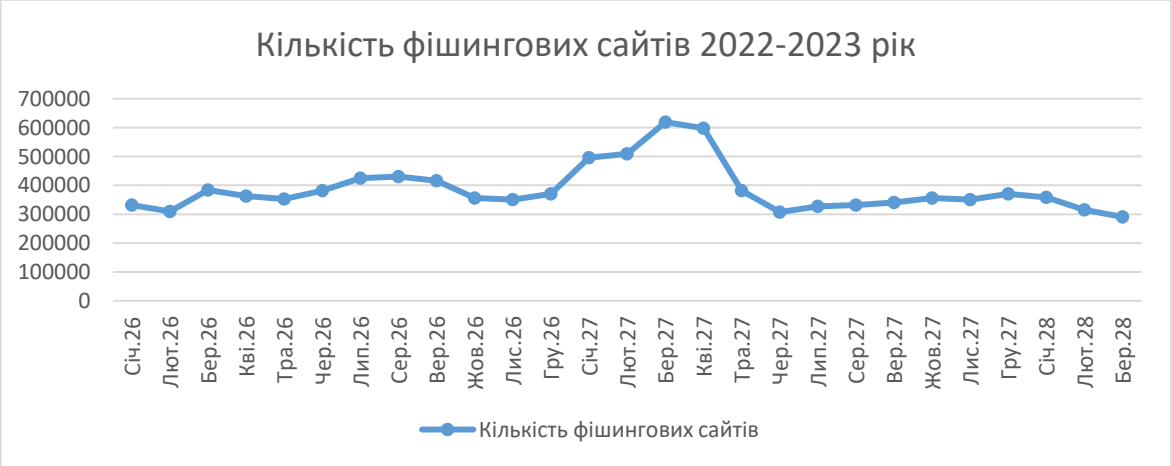


Рисунок 1.3 - Кількість фішингових сайтів за 2022-2023 роки

Аналіз графіку демонструє, що за період з 2022 року до першого кварталу 2024-го був відмічений різкий стрибок кількості як фішингових сайтів, так і email-тем. За даними Anti-Phishing Working Group, в період з лютого по квітень кількість фішингових сайтів досягла рекордних значень, а загальна кількість фішингових атак зросла на 34,5% [9]. Такий стрімкий ріст обумовлений популяризацією Phishing-as-a-Service (PaaS) сервісів, які дозволили менш досвідченим кіберзлочинцям здійснювати складні атаки без значних технічних знань. Велику роль у цьому процесі також відіграло широке розповсюдження доступу до систем штучного інтелекту, які допомагали кіберзлочинцям виконувати велику кількість рутинних завдань, таких як генерування контенту, текстових файлів, політик та інших медіа-файлів. Це дозволило швидше створювати та адаптувати фішингові сайти під різні географічні регіони [10]. Таким чином використання штучного інтелекту для розробки фішингових сайтів, особливо при копіюванні вже існуючих є актом порушення доброчесності по відношенню до власників легітимного сервісу [4].

Незважаючи на швидку реакцію компаній, які спеціалізуються на кіберзахисті, та вжиті заходи з протидії фішинговим атакам, графік показує, що ситуацію вдалося лише стабілізувати та повернути на середній рівень. Стабільність графіку може вказувати на те, що кіберзлочинцям вдалося адаптуватися до нових захисних механізмів, а також на відсутність подальших суттєвих змін у методах захисту, які могли б призвести до значного зниження кількості атак. Таким чином, загроза стати жертвою фішингу залишається на досить високому рівні. Шахраї використовують різні методики розробки фішингових сайтів, та мають за ціль не тільки звичайних користувачів, а й уповноважених осіб до прикладу, адміністраторів сайтів або баз даних тощо Отримання даних для входу до бази даних дає змогу шахраю здійснювати маніпуляції з великою кількістю конфіденційних даних, або ж зміною структури рольового розмежування [6]. Це означає, що навіть за наявності сильних технічних бар'єрів, ризик успішних атак залишається значним, а рівень загрози фішингових атак не знижується, вимагаючи

постійного вдосконалення захисних заходів та підвищення обізнаності користувачів.

1.2 Аналіз основних ознак фішингових веб-сайтів

Ще рік тому фішингові сайти легко визначались стандартним набором ознак, які не змінювались багато років поспіль. Дані ознаки мають свою актуальність нині, але варто враховувати активне розповсюдження готових наборів для фішингу, які внесли свої корективи в ідентифікацію веб-сайтів які підпадають під категорію фішингових. Зловмисники використовують різноманітні методи для надання фішинговим сайтам вигляду легітимних ресурсів, щоб ввести користувачів в оману та змусити їх розкрити свої дані.

Одна з найпоширеніших ознак фішингових сайтів – використання доменних імен, схожих на адреси відомих та надійних ресурсів. Зловмисники часто реєструють домени, що містять незначні відмінності від оригінальних адрес, наприклад, заміну одного символу, додавання префіксів або суфіксів. Це робиться для того, щоб користувачі не помітили підміни та вважали, що вони знаходяться на справжньому сайті. Підробка доменного імені зображена на рис. 1.4

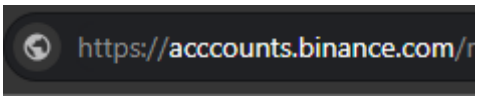


Рисунок 1.4 – Приклад підробки доменного імені

Іншою характерною рисою фішингових сайтів є дизайн, який копіює зовнішній вигляд легітимних ресурсів. Шахраї намагаються відтворити логотипи, кольорові схеми, розташування елементів та загальний стиль оформлення відомих сайтів, щоб створити ілюзію справжності, часто отримуючи медіа дані з легітимного сайту, порушуючи авторські права [5]. Однак, при уважному розгляді часто можна помітити незначні відмінності або недоліки в дизайні, які видають підробку. Фішингові сайти зазвичай орієнтовані на введення конфіденційних

даних, таких як логіни, паролі, та інша чутлива інформація. Ці форми можуть бути замасковані під процес входу в обліковий запис, оновлення платіжних даних або підтвердження особи. Важливо звертати увагу на URL-адресу сторінки та наявність захищеного з'єднання (HTTPS) перед введенням будь-яких персональних даних. Приклад скомпрометованого дизайну для входу на сторінку наведено на рис. 1.5

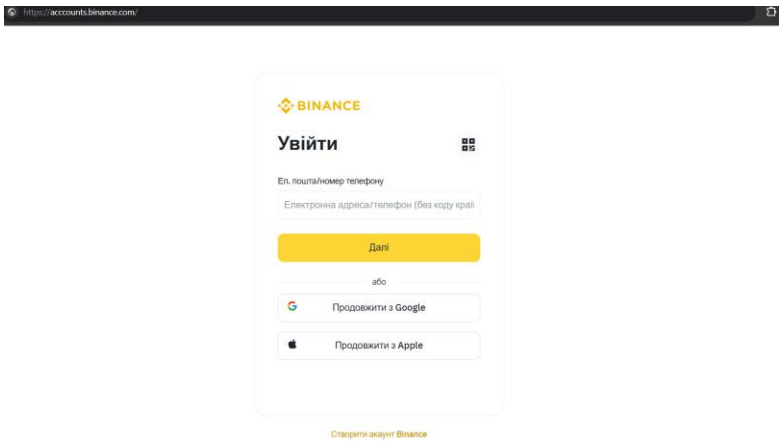


Рисунок 1.5 – Вигляд скомпрометованого сайту Binance

Ще однією ознакою фішингових сайтів є використання методів соціальної інженерії для маніпулювання емоціями та поведінкою користувачів. Зловмисники можуть створювати відчуття терміновості або загрози, стверджуючи, що обліковий запис користувача буде заблокований або він втратить доступ до важливих послуг, якщо негайно не введе свої дані. Також поширеними є обіцянки великих грошових виграшів, знижок або інших принад, які змушують користувачів діяти необдумано.

З технічної точки зору, фішингові сайти часто розміщуються на скомпрометованих серверах або використовують безкоштовні хостингові послуги. Це дозволяє зловмисникам швидко створювати та змінювати фішингові ресурси, ускладнюючи їх виявлення та блокування. Крім того, фішингові сайти можуть використовувати методи приховування свого справжнього розташування, такі як переадресація або використання служб скорочення URL-адрес.

Ще однією технічною особливістю фішингових сайтів є використання шкідливих скриптів та експлойтів для автоматичного збору введених користувачами даних або встановлення шкідливого програмного забезпечення на

їх пристрої. Фішингові сайти також часто використовують методи обфускації коду та шифрування для приховування своєї шкідливої діяльності від систем безпеки та аналізу. Це ускладнює виявлення та аналіз фішингових ресурсів, дозволяючи зловмисникам уникати виявлення та продовжувати свою шахрайську діяльність.

Крім того, фішингові сайти можуть використовувати вразливості в програмному забезпеченні та плагінах веб-браузерів для здійснення атак на користувачів. Наприклад, вони можуть експлуатувати недоліки в застарілих версіях Java для встановлення шкідливого коду на комп'ютери жертв.

Ще однією проблемою є використання фішинговими сайтами методів ухилення від виявлення, таких як використання унікальних URL-адрес для кожного відвідувача або динамічна зміна контенту сторінки в залежності від характеристик користувача. Це ускладнює автоматизоване виявлення та блокування фішингових ресурсів традиційними засобами безпеки.

Окрему увагу слід приділити мобільним фішинговим атакам, які стають все більш поширеними із зростанням використання смартфонів та планшетів. Зловмисники створюють фішингові сайти, оптимізовані для мобільних пристроїв, або розробляють шкідливі мобільні додатки, які маскуються під легітимні програми для викрадення даних користувачів.

Узагальнення основних ознак фішингових веб-сайтів продемонстровано у таблиці 1.4

Таблиця 1.4 – Основні ознаки фішингових веб-сайтів

Ознака	**Опис**
Схожі доменні імена	Використання доменних імен, схожих на адреси відомих сайтів, з незначними відмінностями
Дизайн, що імітує відомі сайти	Копіювання зовнішнього вигляду, логотипів та стилю оформлення легітимних сайтів
Форми для введення даних	Наявність форм для введення конфіденційних даних, замаскованих під процеси входу або оновлення інформації

Продовження таблиці 1.4

Ознака	**Опис**
Методи соціальної інженерії	Використання відчуття терміновості, загроз або обіцянок вигоди для маніпулювання діями користувачів
Розміщення на скомпрометованих серверах	Використання скомпрометованих серверів або безкоштовних хостингових послуг для розміщення фішингових сайтів
Шкідливі скрипти та експлойти	Використання шкідливих скриптів для автоматичного збору даних або встановлення шкідливого ПЗ
Обфускація та шифрування коду	Приховування шкідливої діяльності за допомогою обфускації та шифрування коду сайту
Використання вразливостей ПЗ	Експлуатація вразливостей в програмному забезпеченні та плагінах браузерів для здійснення атак
Методи ухилення від виявлення	Використання унікальних URL-адрес та динамічної зміни контенту для ускладнення виявлення
Мобільні фішингові атаки	Створення фішингових сайтів та додатків, оптимізованих для мобільних пристроїв

Розуміння основних ознак фішингових веб-сайтів є важливим кроком у розробці ефективних методів їх виявлення та протидії. Аналіз цих характеристик дозволяє створювати більш досконалі алгоритми та засоби захисту, здатні ідентифікувати та блокувати фішингові ресурси, забезпечуючи безпеку користувачів в мережі Інтернет.

Однак, постійна еволюція та адаптація методів фішингу вимагає безперервного вдосконалення та оновлення засобів захисту. Зловмисники постійно шукають нові шляхи обходу відомих механізмів безпеки, використовуючи більш складні та витончені техніки. Таким чином набори для фішингу типу EvilProxy дозволяють легко маскувати фішингові сайти під справжні, використовуючи HTTPS-з'єднання та точно копіюючи дизайн. Це значно ускладнює ідентифікацію загрози на ранніх етапах, оскільки навіть досвідчені користувачі можуть не відрізнити фішингову сторінку від справжньої [11]. За останні шість місяців дослідники Proofpoint спостерігали сплеск понад 100% випадків успішного захоплення облікових записів у хмарі, які вплинули на керівників провідних

компаній. Понад 100 організацій по всьому світу були об’єктом атак, що разом становить 1,5 мільйона співробітників [12]. Початковий етап роботи алгоритму EvilProxy зображений на рис. 1.6.

Рисунок 1.6 – Початкові етапи роботи алгоритму EvilProxy

Алгоритм атаки розпочинається з того, що зловмисники, використовуючи підроблені електронні листи, видають себе за легітимні сервіси, такі як Concur, DocuSign та інші. Ці листи містять шкідливі URL-адреси, спрямовані на фішингові сайти, що імітують платформи типу Microsoft 365. Після переходу на шкідливе посилання, відбувається перенаправлення трафіку через відкриті редиректори, по типу YouTube. Зловмисники за часту використовують Cloudflare для приховування справжньої IP-адреси, захисту від DDoS-атак та надання вигляду надійного сайту завдяки наявності HTTPS.

Cloudflare — це глобальна мережа доставки контенту (CDN), що надає послуги оптимізації та безпеки для вебсайтів та вебдодатків. Основною метою Cloudflare є підвищення швидкості доступу до контенту для користувачів з різних регіонів світу та забезпечення захисту від різноманітних інтернет-загроз, таких як DDoS-атаки, шкідливий трафік та спроби несанкціонованого доступу [13].

Таке перенаправлення дозволяє приховати справжній намір атаки та обійти системи виявлення загроз. Користувацький трафік проходить через кілька додаткових етапів перенаправлення, включаючи використання шкідливих файлів cookie та перенаправлення 404. Така кількість перенаправлень створюють

додаткові труднощі для виявлення атаки та розсіюють трафік непередбачуваним чином. Заключні етапи роботи алгоритму EvilProxy зображені на рис. 1.7.

Рисунок 1.7 – Заключні етапи роботи алгоритму EvilProxy

Електронна адреса користувача зазвичай кодується в параметрах URL, наприклад, через Base64 або інші схеми шифрування. Це дозволяє зловмисникам персоналізувати атаки та "запам'ятати" конкретного користувача для подальшого використання його даних на всіх етапах. Таким чином шкідливе посилання буде мати приблизний вигляд https://malicious-site.com/login?user=dXNlci52aWN0aW1AZ21haWwuY29t&redirect=true, де частина dXNlci52aWN0aW1AZ21haWwuY29t відповідатиме за прихований email, закодований Base64.

Наступним кроком відбувається перенаправлення трафіку користувача на фішингову платформу, таку як EvilProxy, яка функціонує як зворотний проксі-сервер. Використання Cloudflare на цьому етапі забезпечує захист фішингового сайту, приховуючи його справжню IP-адресу та надаючи SSL-сертифікат для шифрування трафіку. Зловмисники створюють підроблені сторінки входу, які імітують відомі сайти, такі як Microsoft 365. Ці сторінки можуть запитувати дані багатофакторної автентифікації (MFA), щоб отримати повний доступ до облікових записів користувача.

Отримані дані використовуються для викрадення сесій або доступу до конфіденційної інформації, а захист Cloudflare допомагає уникнути блокувань або

видалення фішингових ресурсів. Широке розповсюдження та популяризація даного підходу у вішингу в поєднанні з CloudFlare створює нову серйозну загрозу, та підвищує ризик проведення успішної атаки зловмисником. Так як використання CloudFlare допомагає модернізувати сайт з шкідливим контентом, забезпечивши імітацію захищеного та надійного сервісу, що з більшою ймовірністю викличе довіру у жертви.

Тому важливо врахувати нові тенденції та розробляти інноваційні підходи до виявлення та протидії фішинговим атакам. Ефективна стратегія боротьби з фішинговими веб-сайтами повинна включати комплексний підхід, який поєднує технічні засоби виявлення, освіту користувачів щодо безпечної поведінки в Інтернеті та співпрацю між організаціями та правоохоронними органами для оперативного реагування на загрози.

Таким чином, аналіз основних ознак фішингових веб-сайтів є фундаментальною основою для розробки ефективних методів та засобів захисту від фішингових атак. Врахування технічних особливостей та поведінкових аспектів фішингових ресурсів дозволяє створювати більш досконалі механізми виявлення та протидії, забезпечуючи безпеку користувачів та організацій в цифровому просторі.

1.3 Аналіз відомих методів захисту від фішингу

Проблема фішингових атак в мережі Інтернет стала настільки серйозною, що розробка ефективних засобів захисту від фішингу є пріоритетним завданням для фахівців з кібербезпеки. Існує широкий спектр методів та інструментів, спрямованих на виявлення та запобігання фішинговим атакам, кожен з яких має свої переваги та обмеження.

Одним з найпоширеніших засобів захисту від фішингу є використання списків відомих фішингових сайтів та доменів. Ці списки, які часто називають "чорними списками", містять адреси ресурсів, які були ідентифіковані як фішингові. Веб-браузери та системи безпеки використовують ці списки для

блокування доступу до шкідливих сайтів та попередження користувачів про потенційну загрозу. Однак, недоліком цього підходу є те, що він покладається на постійне оновлення списків та не може виявляти нові фішингові сайти, які ще не були додані до бази даних.

Іншим важливим інструментом захисту є евристичний аналіз веб-сайтів. Евристичні методи використовують алгоритми машинного навчання та статистичні моделі для аналізу характеристик сайтів та виявлення підозрілих ознак, притаманних фішинговим ресурсам. Перевагою евристичного аналізу є здатність виявляти нові та невідомі фішингові сайти, які ще не потрапили до чорних списків.

Також широко використовуються системи репутації веб-сайтів, які збирають та аналізують відгуки користувачів та експертів щодо доброчесності та безпеки різних ресурсів. Ці системи ставлять у відповідність сайтам рейтинги довіри на основі історії їх діяльності, відгуків користувачів та аналізу контенту. Користувачі та системи безпеки можуть звертатися до рейтингів репутації, щоб оцінити ризик взаємодії з конкретним сайтом. Однак, ці системи можуть бути вразливими до маніпуляцій з боку зловмисників, які намагаються штучно підвищити репутацію своїх фішингових сайтів.

Важливу роль у захисті від фішингу відіграють також розширення веб-браузерів та спеціалізовані плагіни. Ці інструменти інтегруються безпосередньо у веб-браузери користувачів і здійснюють перевірку відвідуваних сайтів на наявність ознак фішингу. Вони можуть використовувати комбінацію методів, таких як перевірка чорних списків, евристичний аналіз та репутаційні системи, для забезпечення комплексного захисту. Користувачі можуть встановлювати ці розширення, щоб отримувати попередження про потенційно небезпечні сайти та блокувати доступ до них.

Ще одним засобом захисту є двофакторна автентифікація (2FA). Цей метод вимагає від користувачів надання додаткового підтвердження своєї особи, окрім введення пароля, перед отриманням доступу до облікового запису або конфіденційної інформації. Зазвичай, другий фактор автентифікації може бути у

вигляді одноразового коду, надісланого на мобільний телефон, або біометричних даних, таких як відбиток пальця або розпізнавання обличчя. Використання 2FA значно ускладнює для зловмисників отримання несанкціонованого доступу до облікових записів користувачів, навіть якщо вони отримали паролі через фішингові атаки.

Освіта та інформування користувачів є важливим аспектом захисту від фішингу. Багато організацій проводять навчальні програми та кампанії з підвищення обізнаності, щоб навчити користувачів розпізнавати фішингові атаки та уникати їх. Ці програми можуть включати інформацію про поширені ознаки фішингових повідомлень, правила безпечної роботи в Інтернеті та процедури повідомлення про підозрілі дії. Користувачі, які мають знання та навички для ідентифікації фішингових атак, менш схильні до того, щоб стати жертвами шахраїв.

Організації також впроваджують системи моніторингу та виявлення фішингових атак на корпоративному рівні. Ці системи використовують комбінацію технологій, таких як аналіз мережевого трафіку, машинне навчання та евристичні алгоритми, для виявлення та блокування фішингових повідомлень та спроб несанкціонованого доступу до корпоративних ресурсів. Вони можуть інтегруватися з системами електронної пошти, брандмауерами та іншими засобами безпеки для забезпечення комплексного захисту.

Важливу роль у боротьбі з фішинговими атаками відіграють також організації та спільноти з кібербезпеки. Вони збирають та обмінюються інформацією про нові загрози, розробляють рекомендації та стандарти безпеки, а також співпрацюють з правоохоронними органами для виявлення та притягнення до відповідальності зловмисників. Такі організації, як Anti-Phishing Working Group (APWG) та Messaging, Malware and Mobile Anti-Abuse Working Group (M3AAWG), відіграють ключову роль у координації зусиль з протидії фішинговим атакам на глобальному рівні. Узагальнює основних засобів захисту від фішингу наведено у таблиці 1.5

Таблиця 1.5 – Узагальнення основних засобів захисту від фішингу

Метод захисту	**Опис**	**Засіб, що реалізує метод**
Списки відомих фішингових сайтів	Використання баз даних з адресами ідентифікованих фішингових сайтів для блокування доступу та попередження користувачів	Google Safe Browsing [14], PhishTank [15]
Евристичний аналіз	Застосування алгоритмів машинного навчання та статистичних моделей для виявлення підозрілих ознак фішингових сайтів	Google Safe Browsing, IBM Trusteer Rapport [16].
Системи репутації веб-сайтів	Збір та аналіз відгуків користувачів та експертів щодо надійності та безпеки веб-сайтів	Web of Trust, Norton Safe Web [17]
Розширення веб-браузерів	Інтеграція засобів захисту безпосередньо у веб-браузери для перевірки відвідуваних сайтів на наявність ознак фішингу	Microsoft Defender SmartScreen, Malwarebytes Browser Guard
Двофакторна автентифікація	Вимога додаткового підтвердження особи користувача, окрім введення пароля, для доступу до облікового запису	Google Authenticator, Microsoft Authenticator, Authy
Освіта та інформування користувачів	Проведення навчальних програм та кампаній з підвищення обізнаності користувачів щодо фішингових атак	Курси на онлайн-платформах штибу Coursera, Prometeus. Інформування через мас-медіа
Системи моніторингу та виявлення	Використання технологій аналізу мережевого трафіку та машинного навчання для виявлення фішингових атак на корпоративному рівні	Darktrace [18], Cisco Umbrella, Splunk [19].
Співпраця організацій з кібербезпеки	Обмін інформацією про загрози, розробка рекомендацій та співпраця з правоохоронними органами для протидії фішингу	Інформаційний обмін через Information Sharing and Analysis Centers, спільні ініціативи з правоохоронними органами, наприклад, Europol, CERT-UA.

Незважаючи на наявність різноманітних методів та засобів захисту від фішингу, ця проблема залишається актуальною через постійну еволюцію методів зловмисників. Фішингові атаки стають все більш складними та персоналізованими, використовуючи соціальну інженерію та маніпулятивні техніки для обману користувачів. Тому важливо постійно вдосконалювати та адаптувати засоби захисту, щоб встигати за новими загрозами.

Ефективний захист від фішингу вимагає комплексного підходу, який поєднує технічні рішення, освіту користувачів та співпрацю між організаціями. Лише поєднання різних методів та інструментів може забезпечити надійний захист від фішингових атак та мінімізувати ризики для користувачів та організацій.

Нарешті, ефективність засобів захисту від фішингу значною мірою залежить від обізнаності та поведінки самих користувачів. Навіть найдосконаліші технічні рішення не можуть гарантувати повний захист, якщо користувачі не дотримуються правил безпечної поведінки в Інтернеті та не звертають увагу на попередження та сигнали про потенційні загрози. Тому освіта та інформування користувачів залишаються критично важливими компонентами ефективної стратегії захисту від фішингу.

1.4 Постановка задачі

Проведений аналіз інформаційних джерел, дослідження основних ознак фішингових веб-сайтів та існуючих засобів захисту від фішингу виявив необхідність розробки нових, більш ефективних методів та засобів виявлення фішингових атак. Незважаючи на наявність різноманітних інструментів та підходів до боротьби з фішингом, проблема залишається актуальною в зв’язку з постійним оновленням підходу до розробки фішингових сайтів.

Створення шахраями сервісів, які забезпечують швидку реалізацію великої кількості фішингових сайтів за короткий період часу, наповнює мережу Інтернет новими шкідливими веб сайтами, сучасні методи здебільшого використовують застарілі методи, та не актуалізують роботу засобу належним чином відповідно до нових методів створення фішингових сайтів або ж швидкість адаптації занизька, здебільшого посилаючись на власні бази даних. Через це ідентифікація фішингових сайтів стає дедалі складнішим процесом для звичайного користувача.

Більшість сучасних сервісів які надають послуги з виявлення фішингових веб-сайтів використовують виключно технічні методи виявлення, не враховуючи обставини створення такі як: наявність сусідніх доменів на одному акаунті CloudFlare, свіжої дати реєстрації, підробки сертифікатів. Такий підхід призводить до неповної оцінки потенційних загроз і, як наслідок, вищій ймовірності класифікації фішингового сайту як легітимного. Таким чином постає задача врахування обставин створення сайту під час виявлення фішингових сайтів.

1.5 Висновки з розділу

Аналіз актуальності проблеми показав, що фішингові атаки становлять значну загрозу для користувачів та організацій в сучасному цифровому світі. Зловмисники використовують все більш складні та витончені методи для обману користувачів та викрадення їх конфіденційної інформації. Незважаючи на наявність різноманітних засобів захисту, кількість та ефективність фішингових атак продовжують зростати, що підкреслює необхідність розробки нових, більш досконалих методів виявлення та протидії фішингу.

Дослідження основних ознак фішингових веб-сайтів виявило, що вони мають низку характерних особливостей, які відрізняють їх від легітимних ресурсів. Зокрема, фішингові сайти часто використовують схожі доменні імена, імітують дизайн відомих сайтів, містять підозрілі форми для введення конфіденційних даних та застосовують методи соціальної інженерії для маніпулювання діями користувачів

Аналіз відомих методів та засобів захисту від фішингу показав, що вони включають використання списків відомих фішингових сайтів, евристичний аналіз, системи репутації веб-сайтів, розширення веб-браузерів, двофакторну автентифікацію, освіту користувачів та співпрацю організацій з кібербезпеки. Кожен з цих засобів має свої переваги та обмеження, і найбільш ефективний захист досягається шляхом їх комплексного використання. Однак, постійна еволюція методів фішингу вимагає безперервного вдосконалення та адаптації засобів захисту. Виконаний аналіз дозволив визначити задачі дослідження цієї магістерської роботи.

2 МЕТОД ВИЯВЛЕННЯ ФІШИНГОВИХ ВЕБ-САЙТІВ

2.1 Визначення ознак фішингових веб-сайтів на основі аналізу статистичних даних

Визначення основних ознак фішинових веб-сайтів формують основу для розуміння їх характерних особливостей. У сучасних умовах, коли фішингові ресурси стають дедалі витонченішими, статистичний аналіз стає необхідним інструментом для оцінки релевантності та вагомості кожної з ознак. Таким чином, для підвищення ефективності методу виявлення та захисту, важливо оцінити, важливість та вагомість кожної з ознак, а також визначити наскільки часто ці ознаки зустрічаються у реальних випадках та який вплив вони мають на ідентифікацію загроз. Ідентифікація первинних ознак фішингових сайтів дозволяє середньо-статистичному користувачу персонального комп'ютера при достатньому рівні обізнаності візуально ідентифікувати фішинговий веб-сайт. Однак ефективність даного методу визначається в більшій мірі виключно наявними знаннями користувача та людським фактором.

Для реалізації методу ідентифікації фішингових веб-сайтів, що не залежить від людського фактору, доцільно враховувати не лише базові ознаки, які є очевидними для користувачів, а й ті які залишаються поза увагою середньостатистичного відвідувача через обмеження часу, знань або навичок. Врахування таких малопомітних ознак в поєднанні з загальновідомими характеристиками надаватиме вищі показники якості перевірки сайту на наявність фішингу. Статистичне представлення допоможе визначити реальну поширеність та значущість кожної ознаки в умовах динамічного розвитку методів фішингу. Такий підхід дозволяє не лише підтвердити релевантність відомих характеристик, але й виявити тенденції, які раніше могли залишатися непомітними.

За даними робочої групи щодо боротьби із фішингом APWG було виявленно значну кореляцію між новостворенними доменними іменами та фішинговими атаками. На початку 2024 року спостерігалося зростання кількості унікальних доменів,

які використовуються для фішингу, на 77% порівняно з попереднім періодом [20]. Таким чином нещодавню дату реєстрації домена можна розглядати як важливу ознаку, яка може вказувати на підвищений ризик фішингу. Підробка домена шляхом використання бренд ключів у домені також є однією з видимих ознак. За даними ThreatLabz, використання брендових ключових слів у піддоменах залишається однією з ключових тактик зловмисників у фішингових атаках [21]. Статистичні дані розподілу найбільш популярних бренд-цілей доцільно зобразити у вигляді графіку продемонстрованому на рисунку 2.1.

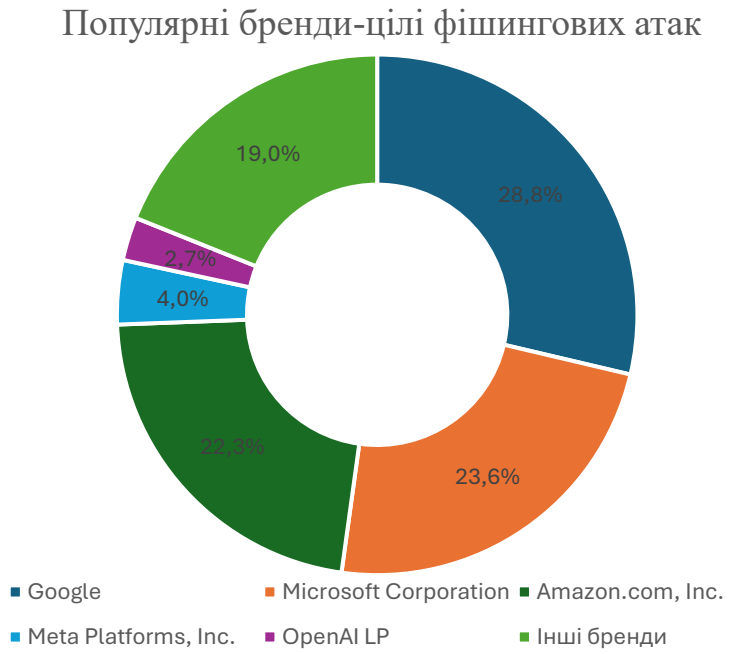


Рисунок 2.1 – Популярні бренди-цілі фішингових атак

Аналіз показав, що серед усіх зафіксованих випадків фішингових доменів 28,8% були спрямовані на імітацію Google, 23,6% – Microsoft, а 22,3% – Amazon. Зловмисники використовують ці бренди через їхню глобальну базу користувачів та високу довіру споживачів. Наведені дані підтверджують потребу врахування факту підробки домену при аналізі сайту на фішинг.

Використання HTTPS стало частою практикою для створення ілюзії легітимності фішингового сайту. Зловмисники все частіше користуються послугами

центрів сертифікації. У звітах ThreatLabz відмічено що вибір центру сертифікації зловмисниками обгрунтовується такими ключовими факторами [21]:

– простота отримання сертифікату;
– низька вартість;
– довіра користувачів;
– широка доступність;
– мінімальні перевірки;
– швидкість видачі;

Відповідно до наведених факторів найчастіше зловмисники використовують такі сервіси як: Let's Encrypt, Google Trust Services, GoDaddy. Частоту їх використання доцільно представити у вигляді діаграми яка зображена на рисунку 2.2.

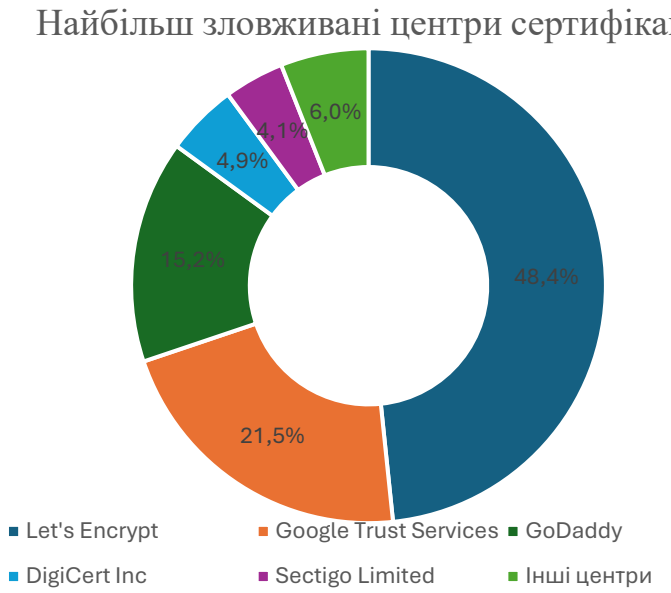


Рисунок 2.2 – Центри сертифікації, що зазнають найбільших зловживань

Аналіз демонструє, що найбільшу частку займає Let's Encrypt – 48,4% сертифікатів фішингових доменів було видано цим центром. Це зумовлено безкоштовністю сервісу, а також простотою у використанні та мінімальними вимогами до перевірки безпеки, що робить його вдалим вибором для зловмисників. Google Trust Services із часткою 21,5% є одним з лідерів серед центрів сертифікації які обирають зловмисники, популярність якого пояснюється високим рівнем довіри до бренду

Google, що додає фішинговим сайтам довіри в очах користувачів. GoDaddy, на який припадає 15,2% сертифікатів, також є одним із найбільш використовуваних центрів сертифікації SSL TLS високий рівень популярності зумовлений простотою та автоматизованим підходом до отримання сертифікату. Інші центри сертифікації, також фігурують у статистиці, але посідають менш значущі позиції у рейтингу. Аналіз демонструє схильність зловмисників до центрів які поєднують в собі найбільшу кількість ключових факторів вибору.

Також важливим фактором у загальній структурі доменного імені є домен верхнього рівня (TLD). Зловмисники використовують різні підходи та оперуються залежністю ціна-якість. Якщо потенційно небезпечні сайти створюються великою кількістю до прикладу з використанням сервісів для фішингу, обирають менш популярні TLD такі як .top, .site, .xyz та ін. Але найбільшу частку займають TLD з високим рівнем довіри. Відсотковий розподіл доменів верхнього рівня представлений у вигляді діаграми на рисунку 2.3.

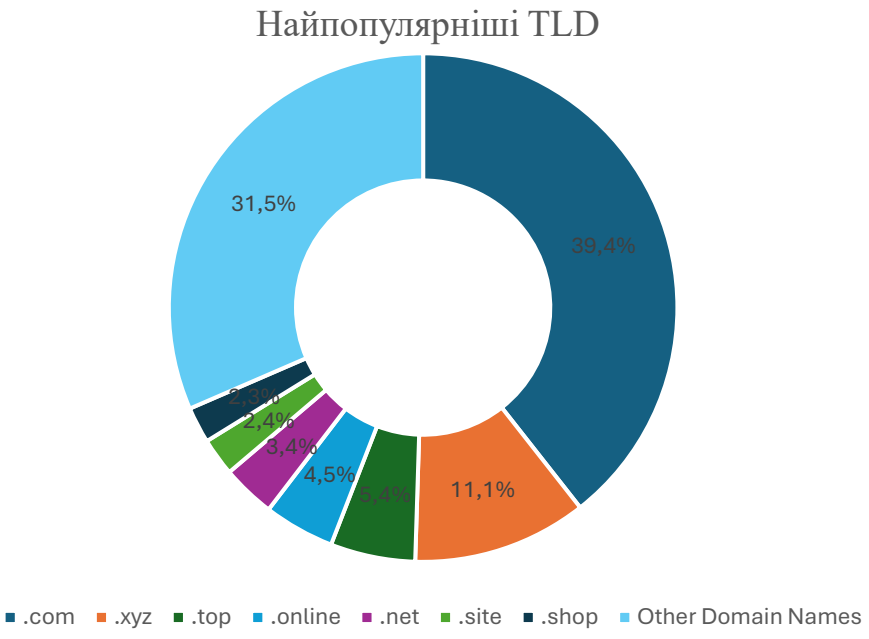


Рисунок 2.3 – Найпопулярнії TLD

Аналіз демонструє що домени .com займають найбільшу частку що становить 39,4%. Це зумовлено їхньою популярністю серед користувачів як наслідок сайти з

таким доменом верхнього рівня інтуїтивно викликають більшу довіру. Домени .xyz становить 11,1%, це пов’язано з його відносно низькою вартістю реєстрації та доступністю для створення великої кількості сайтів. Решта доменів по типу: .top .online, .net, користуються нижчою популярністю хоча не є виключенням з переліку. Така статистика демонструє потребу проведення перевірок не тільки дешевих доменів верхнього рівня, а й довірених які потребують більших фінансових затрат з боку зловмисника. Це також підтверджує доцільність врахування цієї ознаки під час аналізу сайту на наявність фішингу.

Перехід за посиланням є однією з ключових задач, до яких спонукає зловмисник у фішингових атаках. Використання перенаправлення при переході за посилання є поширеною тактикою, яка дозволяє шахраям приховати шкідливі наміри. Цей підхід ефективний у випадках, коли необхідно приховати шкідливий ресурс або обійти системи кіберзахисту. Редіректи (перенаправлення на інші сайти) стають інструментом для маніпуляції користувачами, створюючи ілюзію довіри до сторінки, що відкривається, і, відповідно, зменшуючи рівень настороженості зловмисники використовують шкідливі посилання в 43% випадків про що свідчать данні Positive Technologies [22]. Користувач бачить знайому URL-адресу, переходить за нею, однак автоматично перенаправляється на шкідливу сторінку. Це перенаправлення часто реалізується через уразливості типу Open Redirect, які можуть бути присутні навіть на авторитетних веб-ресурсах. Після перенаправлення користувач може потрапити на ресурс на якому будуть виконані небезпечні дії. Таким чином перенаправлення з одного посилання на інше можна ідентифікувати як ознаку фішингового сайту.

Часто зловмисники не обмежуються перенаправленнями, даний метод також може включати відкриття домену з різних геолокацій. Такий підхід ґрунтується на аналізі IP-адреси користувача. Це дає змогу серверу визначити географічну локацію запиту. В наслідок цього, фішинговий сайт може надавати різний контент, підлаштовувати мову, валюту, що підвищує імовірність успішної атаки. Близько 10-15% фішингових сайтів використовують геолокаційні перенаправлення або

динамічні зміни контенту, залежно від місця розташування користувача [8]. Такий підхід націлений на підвищення довіри у жертви, що робить даний підхід характерною ознакою фішингового сайту.

Альтернативою цій практиці є використання сусідніх доменів, розміщених на одному обліковому записі сервісу CloudFlare. У цьому випадку один обліковий запис може містити кілька доменів, кожен із яких служить різним цілям однієї атаки. З точки зору стратегії зловмисника, один домен може виглядати, як сторінка входу до банківського сервісу, другий як система технічної підтримки, а третій як підроблена сторінка активації облікового запису. Використання однієї інфраструктури дозволяє зловмисникам централізовано керувати кількома компаніями, такий підхід дозволяє швидко змінювати налаштування та уникати блокувань. Така тактика часто супроводжується динамічним підлаштуванням вмісту залежно від географічного положення користувача, що дозволяє обійти регіональні фільтри або імітувати локальні сервіси.

Зловмисники використовують кілька доменів, кожен із яких виконує конкретну роль у фішинговій компанії. Також декілька доменів можуть містити безпечну інформацію для розбавлення вмісту, що ускладнює ідентифікацію шкідливого сайту. Під час перевірки підозрілого веб-сайту доцільно здійснювати перевірку сусідніх доменів, щоб знизити ймовірність помилкового спрацювання. Як наслідок потреба перевірки сусідніх доменів пов’язаних небезпечним сайтом можна віднести до ознаки фішингового сайту.

На фішингових сайтах з технічної точки зору є також і велика кількість слідів непомітних користувачеві за якими можна ідентифікувати фішинговий сайт. Під час створення фішингового веб-сайту зловмисники разом з ним створюють шкідливі файли, такі як підроблені логотипи, скрипти для збору даних чи документи, які потім розміщують на фішингових сайтах. Кожен із цих файлів має унікальний MD5-геш, такі геші можна ідентифікувати за допомогою бази даних відомих загроз. Збіг геш файлу з гешем у базі, означає, що файл вже використовувався в інших атаках, і його можна класифікувати як небезпечний. Однак є випадки, коли

зловмисники не обмежуються прямим копіюванням файлів. Розповсюдженою методикою обходу захисту є внесення зміни в код або метадані, щоб створити новий MD5-геш, обходячи системи безпеки. Проте, фішингові ресурси зазвичай мають схожі елементи, і навіть мінімальні відмінності не завжди заважатимуть виявленню загрози. Наприклад, підроблений логотип банку, навіть злегка змінений, все одно залишає сліди у вигляді схожих гешів або схожості контенту.

Для ускладнення виявлення шкідливого вмісту, зловмисники також досить часто використовують практику обфускації. Основна мета обфускації – приховування шкідливого коду від автоматизованих систем безпеки. Одним із поширених способів обфускації є додавання надмірної кількості непотрібного коду, який не впливає на функціональність, але ускладнює читання.

Наприклад, додаються зайві змінні, функції, що викликають самі себе, або шифруються критичні частини коду, які відповідають за шкідливі дії. У випадку з фішинговими сайтами це може бути приховування частин DOM-структури, таких як підроблені форми входу, скрипти для крадіжки облікових даних або посилання на зовнішні сервери. За даними компанії Akamai, близько 60% фішингових сайтів використовують обфускацію JavaScript-коду для приховування своїх шкідливих дій [23]. Основна проблема ідентифікації фішингових сайтів за наявністю обфускованого коду полягає у тому,що велика кількість легітимних сервісів також використовують обфускацію. Основна причина обфускації у випадку легітимних сервісів приховування коду з комерційних поглядів. Даний аспект стає потужним інструментом для зловмисників, який дозволяє їм не лише приховувати свої шкідливі дії, але й уникати виявлення системами захисту. Таким чином обфускацію коду можна ідентифікувати як ознаку, але її врахування можливе лише у випадку поєднання з іншими ознаками. На основі лиш цієї ознаки ідентифікація фішингового веб-сайту буде недоцільна.

Також використання сторонніх ресурсів доцільне для визначення фішингового сайту. Перевага таких сервісів полягає у великих базах даних, які зберігають велику кількість інформації стосовно шкідливого коду. Таким сервісом може бути

один з найпопулярніших — VirusTotal. Даний сканер дозволяє проводити аналіз файлів, URL-адрес та IP-адрес у режимі реального часу, використовуючи дані з численних антивірусних систем і спеціалізованих інструментів. Для перевірки фішингового сайту на наявність шкідливого вмісту можна передати його URL-адресу. VirusTotal аналізує вміст, зіставляючи його з відомими сигнатурами загроз, які вже були виявлені іншими аналітиками чи системами. Таким чином попередження від сервісів сканування можна ідентифікувати як ознаку. Узагальнену ідентифікацію ознак фішингового веб-сайту наведено у таблиці 2.2.

Таблиця 2.2 – Узагальнені ознаки фішингового веб-сайту

Ознака фішингового сайту	Короткий опис
Редірект з одного посилання на інше	Використання автоматичного перенаправлення для приховування шкідливого ресурсу.
Використання брендових ключових слів у піддоменах	Зловмисники додають популярні бренди в піддомени для створення довіри у користувача.
Відкриття домену з різних геолокацій	Адаптація контенту сайту на основі IP-адреси користувача (мова, валюта, регіон).
Свіжа дата реєстрації домену	Часто фішингові домени створюються незадовго до запуску атаки.
Попередження від сервісів сканування	Антивірусні сервіси або сервіси перевірки URL (наприклад, VirusTotal) позначають ресурс як небезпечний.
Наявність MD5-гешів файлів у базі даних	Порівняння файлів сайту із відомими зразками шкідливого коду.
Обфускація контенту сайту	Приховування HTML-коду шляхом його шифрування або ускладнення для аналізу.
Обфускація на сайті	Використання заплутаних JavaScript-кодів для приховування шкідливої активності.
Сусідні домени на одному акаунті CloudFlare	Розміщення кількох шкідливих сайтів на одному обліковому записі для централізованого керування.

Узагальнена структура ознак фішингового веб-сайту дозволяє виділити основні характеристики які використовуються зловмисниками під час створення таких ресурсів. Виділення ключових ознак фішингового веб-сайту дозволить в подальшому провести детальне визначення вагомості кожної ознаки, для підвищення ефективності методу виявлення та захисту. Наведені ознаки формують

основу для автоматизованих систем виявлення фішингових веб-ресурсів, дозволяючи підвищити безпеку користувачів у мережі, знижуючи ризик атак та крадіжки даних. Інтеграція та поєднання різних методів аналізу допоможе створити ефективну багаторівневу систему захисту для виявлення фішингових загроз.

2.2 Визначення важливості ознак фішингових веб-сайтів на основі опитування експертів

Ідентифікація ознак демонструє велику варіативність можливих комбінацій під час створення фішингового сайту. Кожна з ознак має власну специфіку, яка впливатиме на ймовірність успішного виявлення фішингового ресурсу. Зловмисники рідко обмежуються використанням однієї ознаки. Стандартним підходом вважається комбінація декількох методів, щоб підвищити ефективність своїх атак та знизити ймовірність їх виявлення захисними системами. Аналіз кожної ознаки окремо дозволяє оцінити її значення у загальній картині виявлення фішингового сайту. Так деякі ознаки, можуть бути більш критичними для ідентифікації, тоді як інші, можуть мати менший вплив у певних сценаріях. Таким чином, для підвищення точності аналізу необхідно ввести вагові коефіцієнти, які відображатимуть відносну важливість кожної ознаки.

Якими б актуальними не були б результати статистичних досліджень на момент створення методу виявлення фішингу, внаслідок зміни векторів атак вони втрачають актуальність, тому поточний стан може відрізнятись від стану, який був на момент збору статистичних даних. Для компенсування цього недоліку прийнято рішення з залучення експертів у галузі кібербезпеки для проведення оцінювання актуальності кожної з визначених ознак. Опитування дозволяє отримати думку спеціалістів, та врахувати сучасні тенденції у методах, які використовують зловмисники для створення фішингових сайтів. Залучення декількох експертів дозволяє сформувати загальну статистику на основі відповідей. Це сприяє більш точному розумінню, які ознаки є критичними для виявлення фішингових ресурсів та які з них мають другорядний характер. Результати аналізу отриманих даних дозволять

ідентифікувати найбільш вагомі ознаки, враховуючи їхній вплив як окремо, так і у взаємозв'язку з іншими характеристиками. Такий підхід забезпечує побудову більш точної та адаптивної моделі для виявлення фішингових сайтів, яка може використовуватися в автоматизованих системах безпеки.

Оцінку кожної ознаки необхідно проводити за рейтинговою системою, яка дозволить кількісно виміряти окремі суб'єктивні судження експертів. У рейтингових системах використовуються шкали розділені на різну кількість рівнів. Найпоширенішими шкалами є трьохбальні, п'ятибальні, семибальні та десятибальні [24]. Для визначення найкращої шкали для проведення рейтингового оцінювання ознак фішингового сайту, необхідно проаналізувати кожну з шкал. Першим кроком у виборі шкали є визначення можливих сценаріїв, які експерт враховуватиме під час оцінки критичності ознаки. Ймовірні сценарії опитування наведено в таблиці 2.4.

Таблиця 2.4 – Ключові питання для визначення критичності ознаки

Кейс	Опис
Вираженість ознаки	Рівень чіткості та помітності ознаки, що визначає, наскільки легко її можна розпізнати серед інших характеристик сайту.
Наслідки для ідентифікації	Рівень, до якого ознака сприяє точній ідентифікації ресурсу як фішингового, чи може викликати певні сумніви замість однозначних висновків.
Реальні приклади застосування	Частота виникнення ознаки в реальних фішингових сайтах, яка впливає на її важливість для аналізу та класифікації.
Помилковість інтерпретації ознаки	Ймовірність того, що ознака буде неправильно інтерпретована на легітимному ресурсі, що може призводити до хибних оцінок.

Визначення приблизних кейсів на які експерти будуть зважати під час визначення критичності ознаки дозволяє визначити, що використання трибальної шкали недоцільне. Трибальна шкала забезпечує можливість базової оцінки ознаки, проте має недостатній рівень деталізації. Це значно обмежує можливість врахування нюансів. Використання трибальної шкали може обмежити експерта та призвести до прийняття спрощених рішень, що вплинуть на точність оцінки. Таким чином використання трибальної шкали створює ризик отримання узагальнених даних, які важче використовувати для глибокого аналізу. Для визначення шкали за якою

експерти проводитимуть оцінку ознак, необхідно врахувати потребу введення проміжних станів для збільшення рівня деталізації. Семибальна шкала забезпечує високу деталізацію за рахунок більшої кількості проміжних станів, що дозволяє респондентам більш точно оцінити ознаку. Проте, використання семибальної шкали може створити когнітивне навантаження, змушуючи респондента витрачати більше часу на прийняття рішення. Також велика кількість проміжних станів недоцільна через часте ігнорування їх. Респонденти можуть не використовувати усі доступні рівні шкали, що теж значно знижає ефективність шкали. Дослідження Джона Доуса зазначають, що збільшення шкали за рахунок додавання великої кількості проміжних станів знижують ефективність знижаючи середні значення. Таким чином використання шкали з великою кількістю проміжних станів недоцільне, що автоматично виключає з вибірки десятибальну шкалу.

На основі проаналізованих аспектів, визначено, що використання п’ятибальної шкали є найкращим рішенням для даного дослідження. Дана шкала забезпечить достатній рівень деталізації для ознак фішингових сайтів, та дозволить респондентам врахувати різні рівні ступенів критичності ознаки. Це дозволить отримати найбільш точні та зрозумілі значення, які будуть використовуватись для подальших розрахунках та визначення вагових коефіцієнтів, що допоможуть в процесі визначення наявності загрози фішингу на веб-сайті.

Оцінка критичності ознаки буде відбуватися шляхом анкетування. Експертам буде представлено на розгляд 9 ознак які вони оцінять за п’ятибальною шкалою. Градація рівня критичності ознаки за п’ятибальною шкалою представлена у вигляді таблиці 2.5.

Таблиця містить п’ять рівнів критичності ознак фішингових сайтів. Вона чітко структурує ознаки, починаючи від "дуже слабкої" яка дорівнює 1 балу до "дуже сильної" яка оцінюється в 5 балів. Для забезпечення однозначності і зручності інтерпретації критичності ознак до кожного рівня наведений опис та пояснення.

Таблиця 2.5 – Шкала оцінювання критичності ознак фішингових сайтів

Бал	Опис	Пояснення
1	Дуже слабка ознака	Ознака має мінімальний зв'язок із фішинговими сайтами або майже не зустрічається. Ймовірність мінімальна.
2	Слабка ознака	Ознака рідко зустрічається у фішингових сайтах. Ймовірність є низькою.
3	Релевантна ознака	Ознака відіграє роль у визначенні фішингових сайтів, проте не є вирішальною для ідентифікації.
4	Сильна ознака	Ознака часто трапляється у фішингових сайтах і вказує на значний ризик.
5	Дуже сильна ознака	Ознака є критично важливою та практично гарантовано вказує на фішинговий сайт.

Опитування проводилось серед 5 експертів. Результатом опитування стали заповнені анкети (додаток Б). Для забезпечення структурованого аналізу результатів, впорядкування загроз відбувалось за середнім балом де перша ознака є найвагомішою на думку експертів. Сортування за середнім балом дозволить попередньо визначити пріоритетні ознаки, які на думку експертів є найбільш або найменш вагомими. Такий підхід дає змогу створити базу для подальшого аналізу, впорядкування дає змогу чітко визначити відносну важливість кожної ознаки та є ключовим етапом для розрахунку вагових коефіцієнтів. Узагальнені результати опитування представлено у вигляді таблиці 2.6.

Таблиця 2.6 – Узагальнені результати опитування експертів

Ознака	Середній бал	Розподіл оцінок
Використання брендових ключових слів у піддоменах	4.8	4 (20.0%), 5 (80.0%)
Наявність MD5-гешів файлів у базі даних	4.8	4 (20.0%), 5 (80.0%)
Попередження від сервісів сканування	4.2	4 (80.0%), 5 (20.0%)
Відкриття домену з різних геолокацій	3.8	1 (20.0%), 2 (20.0%), 3 (20.0%), 4 (20.0%), 5 (40.0%)
Редірект з одного посилання на інше	3.6	3 (40.0%), 4 (60.0%)
Свіжа дата реєстрації домену	3.6	3 (40.0%), 4 (60.0%)
Сусідні домени на одному акаунті	3.2	3 (80.0%), 4 (20.0%)
Обфускація контенту сайту	2.4	2 (60.0%), 3 (40.0%)
Обфускація скриптів	2.0	1 (20.0%), 2 (60.0%), 3 (20.0%)

За результататми опитування було визначено, що найкритичнішими ознаками виявилися «Використання брендових ключових слів у піддоменах» та

«Наявність MD5-гешів файлів у базі даних», які отримали найвищий середній бал. Дані ознаки будуть найвагомішими під час визначення наявності фішингу на сайті, у той час коли такі ознаки як «Обфускація скриптів на сайті» та «Обфускація контенту сайту», які отримали найнижчі середні бали будуть відігравати меншу роль.

2.3. Математичний опис методу процесу аналізу веб сайтів

Для кращого подання процесу визначення фішингового вебсайту, його доцільно представити у вигляді загального математичного опису. Таким чином математичний опис методу виявлення фішингових веб-сайтів можна представити у вигляді формули 2.1

$$M = \{I, S, F, O\} \tag{2.1}$$

де

I – Множина вхідних даних;

S – множина проміжних станів;

F – множина функцій;

O – множина вихідних даних;

Множина вхідних даних I представляє собою початковий етап, на якому користувач надає адресу потенційно інфікованого сайту з метою перевірки його на наявність фішингу. Множина вхідних даних може бути представлена у вигляді формути 2.2.

$$I = \{URL, x_i, w_i, s_i\} \tag{2.2}$$

Множина проміжних станів описує ті характеристики які враховуються під час аналізу веб-сайту. Це включає виявлені ознаки (x_i), які відповідають за ідентифікацію специфічних характеристик фішингового веб-сайту. Вагові коефіцієнти (w_i) які завчасно визначені на основі експертного оцінювання відповідно до кожної ознаки. А також показник вагомості (s_i), який підвищує точність за допомогою розрахунку на основі кількості помилкових спрацювань та хибних пропусків. Проміжні стани дозволяють никопичити максимально-необхідну кількість параметрів

для обчислення рівня ризику. Математичний опис множини проміжних даних наведено у формулі 2.3.

$$S = \{x_i, w_i, s_i\} \tag{2.3}$$

Множина функцій (F) визначає яким чином засіб обробляє дані. На початковому етапі відбувається функція збору даних $f_{collect}$, яка виконує автоматичний аналіз даних отриманих адреси вебсайту аналізуючи технічні та контентні характеристики сайту. Після цього на основі отриманих даних відбувається ідентифікація ознак функція визначає які ознаки ідентифіковані на основі отриманих даних. Після цього функція обчислення ризику , інтегрує всі параметри у формулу ризику та розраховує рівень ризику у відсотковому форматі. На завершальному етапі функція , перетворює відсоткове значення у категоріальний формат для полегшення сприйняття рівня ризику користувачем. Таким чином математичний опис множин функцій наведено у формулі 2.4.

$$F = \{f_{collect}, f_{ident}, f_R, f_{ter}\} \tag{2.4}$$

Вихідними даними (O), є спрощений для сприйняття рівень ризику ($R_{phishing}$). Знчення інтерпертоване у категоріальну шкалу. На основі цієї інформації користувач буде приймати відповідне рішення. Множина вихідних даних наведена у формулі 2.5

$$O = \{R_{phishing}\} \tag{2.5}$$

Таким чином, даний метод забезпечує послідовний процес аналізу, де вхідні дані автоматично обробляються для отримання надійного і чіткого результату.

2.4. Метод аналізу веб-сайтів

Опис методу виявлення фішингових веб-сайтів має включати максимальну кількість ключових аспектів, таких як вхідні та вихідні дані, проміжні етапи аналізу та механізми обчислення вагових коефіцієнтів ознак. Вагові коефіцієнти відіграють ключову роль у формуванні інтегральної оцінки кожного ресурсу. Це дозволить

оцінити відносну важливість кожної ознаки. Такий підхід дозволяє чітко визначити всі параметри методу, їх вплив на кінцевий результат і способи оптимізації, а також надасть представлення про те які параметри можуть бути змінені для підвищення точності ідентифікації. Визначення вагових коефіцієнтів представлено у формулі 2.6

$$w_i = \frac{\frac{\sum_{j=1}^{n} x_{ij}}{n}}{B_{max}} \tag{2.6}$$

w_i – ваговий коефіцієнт для конкретної ознаки.

x_{ij} – оцінка j-го експерта для ознаки і.

n – кількість експертів, які брали участь в оцінюванні ознаки.

B_{max} – максимальний можливий бал в шкалі оцінювання.

Таким чином відбувається розрахунок вагового коефіцієнта для кожної ознаки. Але врахування лише вагових коефіцієнтів не надає повної картини та робить модель суровою класифікуючи велику кількість сайтів як фішингові навіть за мінімальними ознаками.

Будь яка система не є ідеальною, і може бути лише наближеною до істини. Таким чином потрібно врахувати, що деякі ознаки можуть бути ідентифіковані помилково, так наприкладу ознака відсутності HTTPS не завжди вказує на фішинговий веб сайт, адже легітимні сайти також можуть використовувати HTTP. Модель має враховувати коефіцієнт помилкових спрацювань на основі вірних та хибних спрацювань. Визначення коефіцієнту помилкових спрацювань слід розпочати з класифікації сайту як фішингового або легітимного. Дану класифікацію представлено в таблиці 2.7.

Таблиця 2.7 – Матриця класифікації сайтів

	Фішинговий (P)	Легітимний (N)
Класифіковано як фішинговий (P)	True Positive (TP)	False Positive (FP)
Класифіковано як легітимний (N)	False Negative (FN)	True Negative (TN)

Для зручності в подальшому математичному представлені для кожного елементу введено умовне позначення у вигляді скорочення. Умовні позначення відповідають таким станам:

TP – сайт правильно класифіковано як фішинговий.

FP – легітимний сайт помилково класифіковано як фішинговий.

FN – фішинговий сайт помилково класифіковано як легітимний.

TN – сайт правильно класифіковано як легітимний.

Помилкові спрацювання розраховуються як частка легітимних сайтів, які були помилково класифіковані як фішингові. Математичний опис визначення коефіцієнта помилкових спрацювань представлений у формулі 2.7.

$$\text{FPR} = \frac{FP}{FP+TN} \tag{2.7}$$

Аналогічно до розрахунку коефіцієнту помилкових спрацювань, необхідно реалізувати коефіцієнт хибних пропусків, який буде відображати частку фішингових сайтів, які система помилково класифікує як легітимні. Математичний опис визначення коефіцієнту хибних пропусків наведено у формулі 2.8.

(2.8)

$$\text{FNR} = \frac{FN}{FN+TP}$$

Оскільки різні ознаки мають різний ступінь точності, їхній внесок у загальну оцінку ризику повинен бути пропорційним до їхньої ваомості. Математичний опис показника вагомості ознак наведено у формулі 2.9

(2.9)

$$a_i = 1 - FPR_i + (1 - FNR_i)$$

Показники враховуються в парі для оцінки надійності враховуючи здатність не пропускати фішингові сайти та знизити частоту випадків визначення легітимних сайтів як фішингових.

Визначення ймовірності загрози фішингу на основі ідентифікованих ознак має включати в себе всі описані аспекти, це дозволить комплексно оцінити ймовірність наявності фішингу на веб-сайті. Кожна ідентифікована ознака впливає на

результат за рахунок ваги, яка обчислюється на основі корегуючого коефіцієнта та вагових коефіцієнтів. Модель враховує всі доступні дані про виявлену ознаку, що забезпечує зменшення впливу менш надійних елементів, а також підвищує значущість критичних факторів збільшуючи точність розрахунку рівня ризику.

Розрахунок ймовірності загрози наявності фішингу наведено у формулі 2.10

$$R = \frac{\sum_{i \in V} a_i \cdot w_i \cdot x_i}{\sum_{i \in V} a_i \cdot w_i} \quad (2.10)$$

Таким чином, застосування цього методу забезпечить гнучкість, дозволяючи враховувати як сильні так і слабкі ознаки. Окремою перевагою є можливість інтеграції додаткових ознак та зміни коефіцієнтів у випадку потреби актуалізації даних або статистичних показників. Перед визначенням ймовірності наявності фішингу метод аналізу веб сайтів використовує послідовні перевірки потенційно шкідливого сайту. Дані перевірки стають основою для ідентифікації наявних ознак та розрахунків. Даний метод покроково виконує наступні функції.

Крок 1.

Визначення чи відомий даний веб сайт. Здійснюється преевірка наявності домену сайту у списку раніше тестованих.

Крок 2.

Відбувається перевірка домену за допомогою довірених веб-сервісів, які здійснюють визначення основної інформації про домен, сканують на наявність шкідливого вмісту, та перевіряють наявність сертифікатів.

Крок3

Відбувається ряд внутрішніх перевірок, які на основі отриманих даних від зовнішніх веб-сервісів та внутрішніх перевірок здійснюють аналіз домену та ідентифікують.

Крок 4

Відбувається аналіз сформованих звітів, та визначення кількості ознак. За кожною з ознак закріплюється коефіцієнт вагомості, який визначає наскільки сильно впливатиме наявність ідентифікованої ознаки на ймовірність наявності фішингу.

Крок 5

Розрахунок ймовірності ймовірності наявності фішингу на веб-сайті, результат якого інтерпретується у категоріальну шкалу.

Цей метод має стати основою для розробки засобу і забезпечити його ефективну роботу шляхом поєднання зовнішніх даних з внутрішніми алгоритмами аналізу. Такий підхід дозволить оперативно виявляти фішингові сайти, враховуючи як технічні ознаки домену, так і його поведінку. Крім того, використання вагових коефіцієнтів підвищить точність оцінки, дозволяючи системі адаптуватися до нових загроз та мінімізувати помилкові спрацювання.

2.5 Висновки з розділу

Аналіз фішингових веб-сайтів дозволив визначити найбільш популярні аспекти, які використовуються під час проведення фішингових атак. На основі отриманих даних виділено основні ознаки, які стали фундаментом для побудови найбільш частих фішингових ознак. Ці ознаки дозволяють враховувати як технічні характеристики, так і поведінкові особливості сайтів, що значно розширює можливості класифікації. Отримані дані представлені у вигляді анкетного опитування для експертів з метою визначення значущості окремих ознак. Анкетування охоплювало найбільш релевантні ознаки, які можуть бути застосовані у різних сценаріях аналізу. На основі отриманих даних з анкетування розраховано вагові коефіцієнти, що дозволяють більш точно визначати ймовірність виявлення фішингових сайтів. Вагові коефіцієнти визначають ступінь впливу кожної ознаки на загальний результат, зокрема враховуються як частота, так і значущість кожної ознаки.

Удосконалений метод містить вхідні дані, зокрема URL, який користувач надає засобу для перевірки, проміжні дані, коефіцієнти та кількість виявлених ознак.

3 ЗАСІБ ВИЯВЛЕННЯ ФІШИНГОВИХ ВЕБ-САЙТІВ

3.1 Архітектура засобу виявлення фішингових веб-сайтів

Засіб виявлення фішингових сайтів базується на клієнт-серверній архітектурі, що передбачає взаємодію двох ключових елементів. Клієнтська частина реалізована у вигляді розширення у веб-браузері, що значно спрощує використання засобу. Для виконання перевірки не потрібно додатково відвідувати сторонні веб-сторінки чи вводити вручну адреси сервісів перевірки, вся функціональність доступна у інтерфейсі, вбудованому у браузер. Серверна частина реалізована у вигляді веб-сервера, на якому сконцентрована основна логіка перевірки. При використанні розширення, адреса потенційно шкідливого сайту буде надсилатися на веб-сервер.

Алгоритми перевірки реалізовані на серверній частині виконають первинний аналіз посилання, також за допомогою сторонніх веб-сервісів таких як VirusTotal та WHOIS отримають додаткову інформацію стосовно доменної адреси. Відповідно до зібраної інформації буде проведено перевірку на наявність ознак фішингового сайту, їх кількість та розрахують рівень ризику відповідно до удосконаленого методу. Після чого сервер поверне дані стосовно рівня ризику, та рекомендації які може виконати користувач для запобігання потраплянню на фішингові сайти чи компрометації його персональної інформації, тим самим допомагаючи прийняти обґрунтоване рішення щодо подальшої взаємодії з ресурсом. Схематичне представлення архітектури засобу представлено на рисунку 3.1.

Окрім взаємодії з розширенням користувач має змогу відвідати веб-сайт на якому реалізована форма ручної перевірки посилань. Результатом користувач отримує відповідний рівень ризику, чи є сайт фішинговим, у випадку використання веб-сайту також можна ознайомитись з додатковою інформацією про домен. Таким чином, користувач має можливість здійснювати перевірку потенційно небезпечних веб-сайтів без потреби встановлювати розширення.

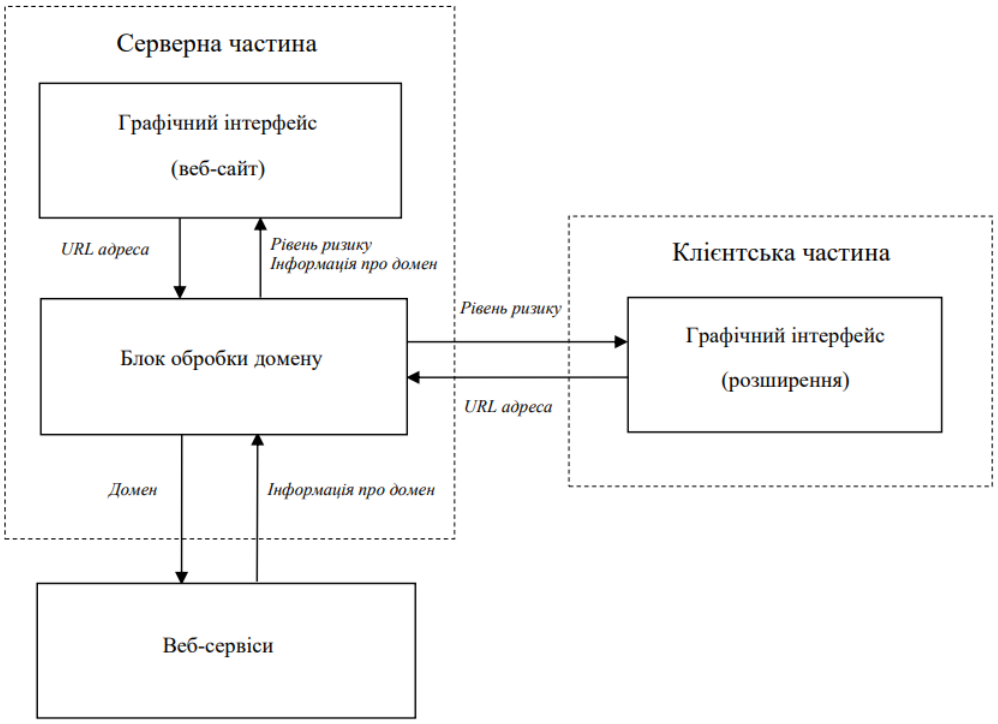


Рисунок 3.1 – Схематичне представлення структури засобу виявлення фішингових веб-сайтів

Використання, клієнт-серверної архітектури забезпечує поєднання зручності та ефективності. Користувач, завдяки розширенню, отримує миттєвий доступ до перевірки будь-якої сторінки, не відволікаючись від поточних дій і не докладаючи зайвих зусиль. Інтеграція серверної логіки з зовнішніми сервісами, дозволяє отримувати додаткову інформацію про сайт. Це створює багаторівневу систему перевірки, де кожен компонент доповнює загальну картину і дозволяє більш точно оцінити рівень ризику. Проведення аналізу на серверній стороні дозволяє уникнути навантаження на пристрій користувача, зберігаючи високу швидкість роботи на слабких пристроях.

Після отримання URL-адреси від клієнтської частини або через веб-сайт сервер ініціює процес аналізу, який складається з кількох етапів. На першому етапі URL-адреса передається до блоку, який відповідає за інтеграцію із зовнішніми веб-сервісами, такими як VirusTotal і WHOIS. В результаті VirusTotal перевіряє факт наявності URL у базах шкідливих ресурсів, визначення оцінок антивірусних

двигунів, а також для аналізу MD5-гешів файлів, розташованих на сайті, шляхом порівняння їх із відомими фрагментами шкідливого коду. WHOIS надає дані про дату реєстрації домену, а також інформацію про реєстратора, власника домену та його геолокацію. Після отримання даних від зовнішніх сервісів сервер переходить до виконання внутрішнього аналізу URL-адреси здійснюючи додаткові перевірки на наявність ознак фішингового сайту. Усі отримані дані передаються до блоку обробки, де вони об’єднуються у єдиний профіль перевірюваного сайту. Наступним кроком дані надходять до блоку розрахунку рівня ризику, де виконується фінальна оцінка на основі розрахунку за формулою, яка враховує всі виявлені ознаки, їхню вагу та контекст повертаючи результат представлений у вигляді категорії ризику.

На основі розрахованого рівня ризику в блоці представлення результату відбувається формування рекомендацій для користувача, які містять рівень ризику та практичні поради щодо подальшої взаємодії з ресурсом. У випадку використання сайту додатково надається пояснення щодо виявлених ознак. Завершальним етапом є передача результатів перевірки до клієнтської частини або графічного інтерфейсу веб-сайту, де вони відображаються користувачу у зручному вигляді. Схематичне представлення структури серверу наведено на рисунку 3.2.

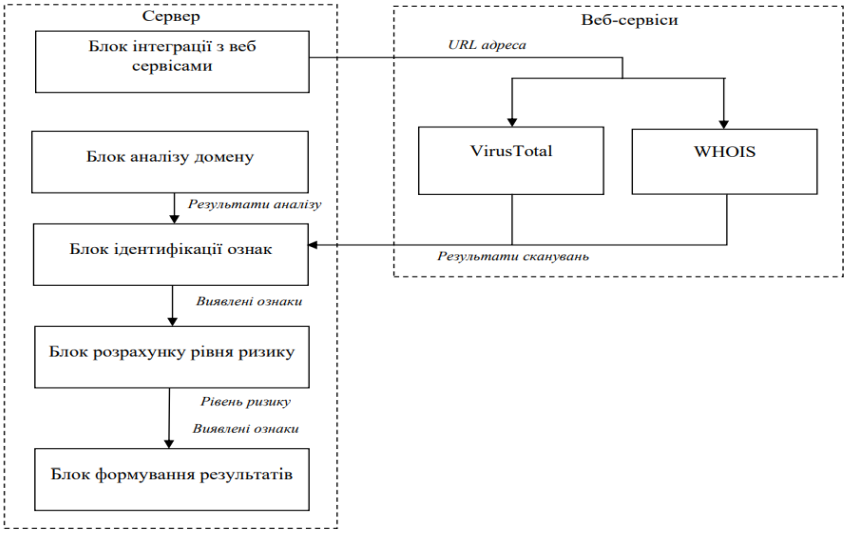


Рисунок 3.2 – Схематичне архітектури серверної частини засобу виявлення фішингових веб-сайтів

Розроблена архітектура серверу поєднує дані зовнішніх веб-сервісів із внутрішнім аналізом, що дозволяє швидко отримувати ключову інформацію про домен. Завдяки багаторівневій перевірці враховуються як зовнішні дані, так і результати внутрішньої ідентифікації ознак. Це забезпечує можливість виявляти не лише відомі фішингові сайти, але й аналізувати нові загрози, підвищуючи точність результатів.

3.2 Узагальнений алгоритм роботи засобу

Узагальнений алгоритм роботи засобу виявлення фішингових веб-сайтів складається з трьох основних етапів. кожен з яких включає ряд кроків. На першому етапі відбувається перевірка чи наявні дані про досліджуваний веб-сайт та збір даних, який включає дані отримані від зовнішніх джерел та власні внутрішні алгоритми перевірок. Другий етап містить алгоритм оцінок характеристик на основі даних отриманих під час аналізу. Третій етап є завершальним та реалізує формування звіту де кожна ознака має свій внесок у загальний ризик. Таким чином узагальнений алгоритм роботи засобу у вигляді блок-схеми зображено на рисунку (додаток В).

Як тільки отримується URL адреса веб-сайту, засіб генерує унікальний MD5-хеш для створення ідентифікатора, що спрощує обробку та скорочує довгі посилання. Далі перевіряється наявність цього хешу у базі даних попередніх перевірок. Якщо хеш вже існує, дані про сайт актуалізуються для виявлення можливих змін. У разі відсутності хешу запускається алгоритм аналізу даних з використанням зовнішніх джерел, таких як WHOIS та VirusTotal, а також внутрішніх ознак веб-сайту. Усі отримані дані записуються у JSON-файли, які згодом використовуються для ідентифікації ознак фішингу. На основі цих ознак засіб розраховує рівень ризику у числовому форматі. Завершальним етапом є формування звіту, який надається користувачу у числовому та категоріальному форматі на окремій сторінці веб-сервісу. Таким чином засіб після проведення аналізу надасть користувачу оцінку ризику та зафіксує у власній базі даний веб-

сайт, у випадку повторного тестування система зможе оперативно надати актуальні дані про веб-сайт, використовуючи вже збережену інформацію та за необхідності оновивши її. Це зменшує час на аналіз вже відомих веб-ресурсів, а також підвищує точність завдяки регулярному оновленню результатів. Реалізація такого типу дозволить не лише швидко ідентифікувати фішингові веб-сайти, а й підтримувати актуальну базу даних фішингових ресурсів.

3.3 Процедури аналізу на основі зовнішніх джерел

Процедури аналізу на основі зовнішніх джерел включають в себе визначення базових характеристик домену за допомогою довірених сервісів. Отримана ключова інформація стає основою для проведення подальших перевірок, а також спрощує ідентифікацію аномалій у поведінці веб-сайту. Основними зовнішніми сервісами є WHOIS та VirusTotal. Аналіз за допомогою сервісу WHOIS надає засобу ключову інформацію про домен, включаючи дані про реєстратора, дати створення, останнього оновлення та закінчення реєстрації. Також повертається список серверів імен, які використовує домен, статус домену з вказівкою можливих обмежень, таких як заборона видалення чи передачі, а також контактну email-адресу. Окрім цього, WHOIS надає за наявності додаткові технічні дані, як специфікації адміністративних чи технічних контактів, які іноді включають інформацію про організації або країни реєстрації. Алгоритм роботи процедури аналізу доменного імені за допомогою веб сервісу Whois наведено у вигляді блок-схеми на рисунку 3.3

Таким чином, після отримання адреси, засіб виконує перший етап аналізу даних, а саме відправку запиту на веб-сервіс Whois, що містить адресу потенційно шкідливого сайту. Після цього здійснюється перевірка чи приходить відповідь від сервісу. У випадку якщо відповідь не надійшла формується відповідне повідомлення про помилку та алгоритм закінчує свою роботу.

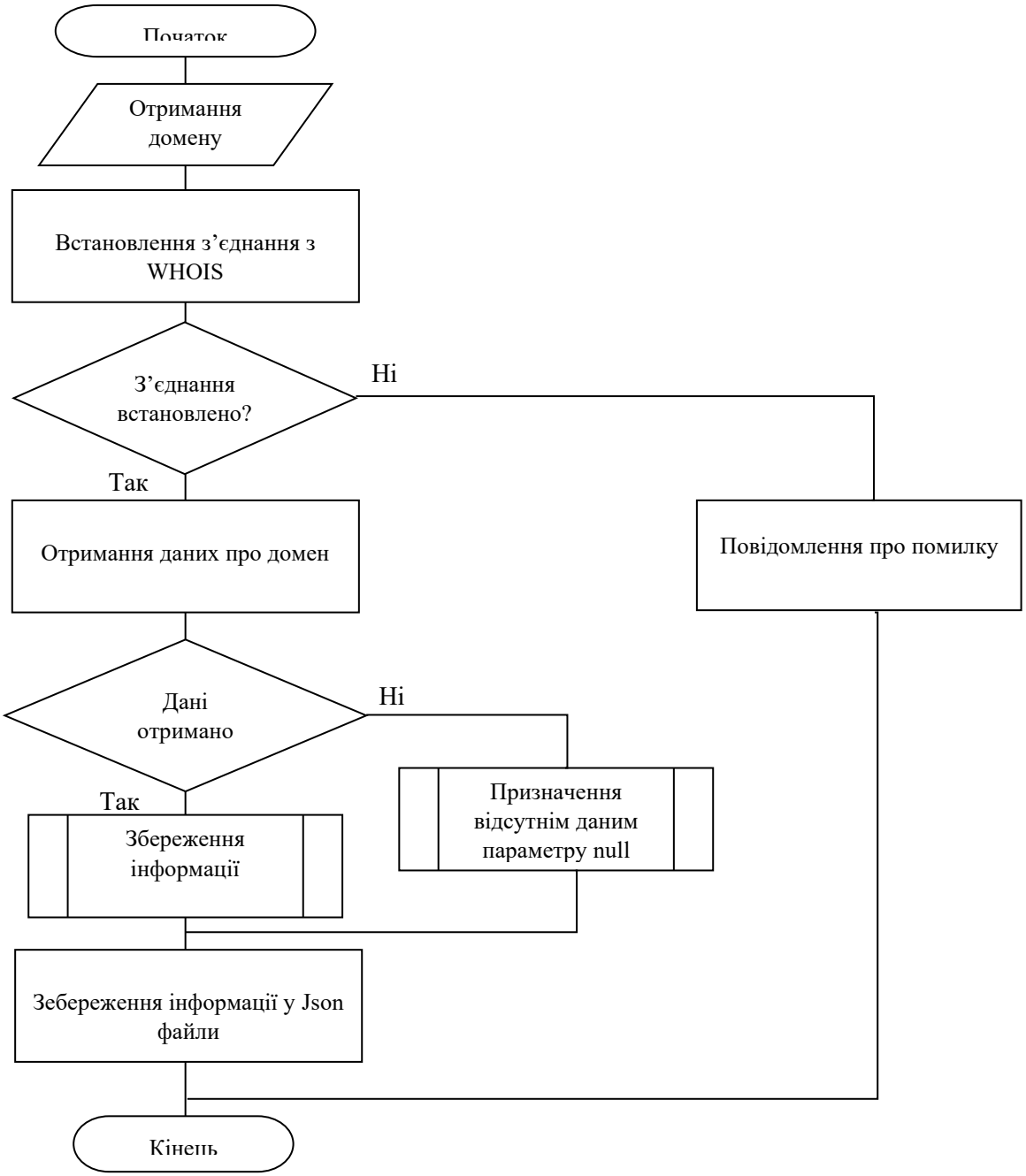


Рисунок 3.3 – Алгоритм роботи процедури аналізу домену за допомогою Whois

Важливо зазначити, що у випадку передчасного завершення роботи алгоритму через помилку, робота засобу не завершується, та переходить до наступних функцій. Така логіка реалізована на випадок, якщо один з сервісів по технічній причині не буде доступний, тобто не прийматиме запити. У випадку якщо сервіс надає у відповідь дані відбувається перевірка чи отримані всі очікувані дані.

Якщо сервіс по якійсь причині надав не всі дані, то пропущеним даним призначається параметр “NULL”. Дані з цим параметром ігноруються при визначені рівня ризику фішингу, але будуть знову перевірені у випадку повторної перевірки. Завершальним етапом є збереження необхідних даних, та запис їх у файли json.

Аналіз домену за допомогою VirusTotal здійснюється з метою отримати у відповідь дані стосовно кількості шкідливих та підозрілих елементів. Даний веб-сервіс використовує велику кількість антивірусних двигунів, на основі яких відбуваються перевірки підозрілих посилань. В залежності від відповіді антивірусів, сервіс категорізує загрози за типом походження. З’єднання з сервісом здійснюється за допомогою API. Алгоритм роботи процедури аналізу домена за допомогою веб-сервіса VirusTotal наведено у вигляді блок-схеми на рисунку (додаток Г).

Робота алгоритму розпочинається з того, що сервер надсилає до VirusTottal Post запит з адресою підозрюваного сайту, використовуючи API. Після надсилання через деякий час відбувається перевірка чи сервіс надіслав відповідь.. У випадку якщо за цей час сервіс не надсилає відповідь, формується повідомлення про помилку. Як і у випадку з Whois якщо формується повідомлення про помилку сервер переходить до виконання наступних функцій. Якщо ж дані від сервісу надходять, виконується функція, яка підраховує кількість шкідливих елементів після чого відбувається запис даних у Json.

Процедура визначення сертифікату працює за рахунок прямої взаємодії з публічним реєстром сертифікатів ctr.sh. Даний сервіс містить записи про всі сертифікати видані для певного домену. Виконання процедури розпочинається з формування запиту який надсилається до crt.sh. Результатом повернення структуровані дані про сертифікати, включаючи їх статус, дати дії, сертифікаційний центр, який видав сертифікат, та перелік доменів, для яких сертифікат видано. Алгоритм роботи процедури визначення сертифікатів представлено у вигляді блок-схеми на рисунку 3.4.

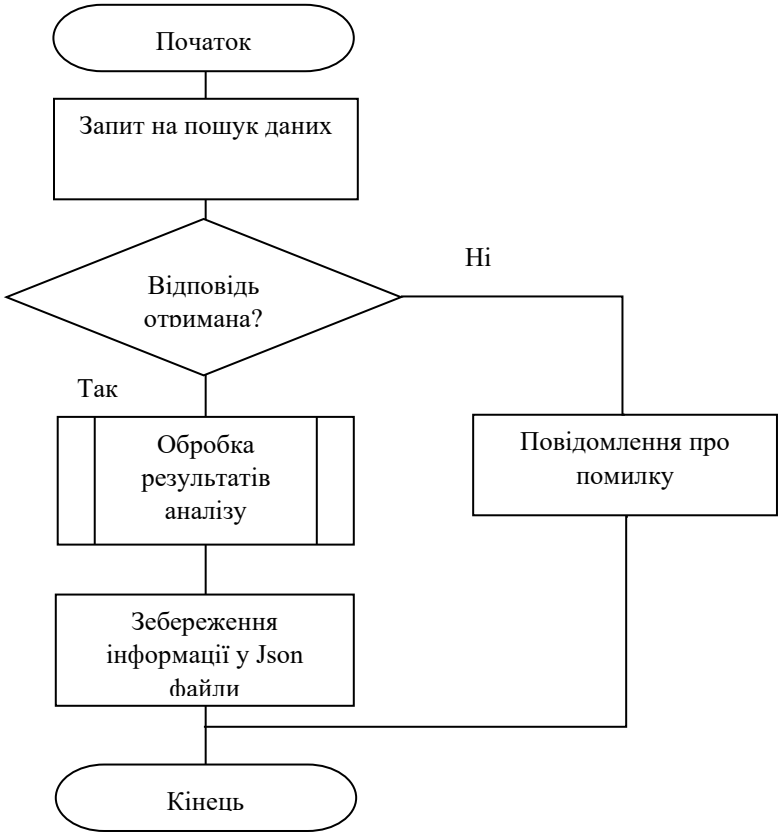


Рисунок 3.4 – Алгоритм роботи процедури аналізу сертифікатів за допомогою crt.sh

Отримана інформація дає можоливість визначити, які саме сертифікати видані, чи є сертифікати надійними, чи були видані з порушеннями. Виконання цих процедур надає можливість здійснити базовий аналіз потенційно шкідливого посилання та сформувати звіти даних з ключових параметрів отриманих під час аналізу за допомогою зовнішніх джерел. На основі отриманих даних відбувається частковий аналіз, та перевірка наявності ознак фішингових веб-сайтів.

3.4 Процедури аналізу на основі внутрішніх ознак

Процедури аналізу на основі внутрішніх ознак працюють у поєднанні з аналізом за зовнішніми ознаками. Ці процедури здійснюють як власні перевірки, так і доопрацьовують отримані дані. Оскільки частина отримання даних покладається на зовнішні джерела, аналіз на основі внутрішніх ознак виконує допоміжні функції. Основна процедура яка визначає вектор аналізів є перевірка чи

є тестована адреса у списку раніше перевірених веб-сайтів на наявність фішингу. Алгоритм роботи процедури наведено у вигляді блок-схеми на рисунку 3.5.

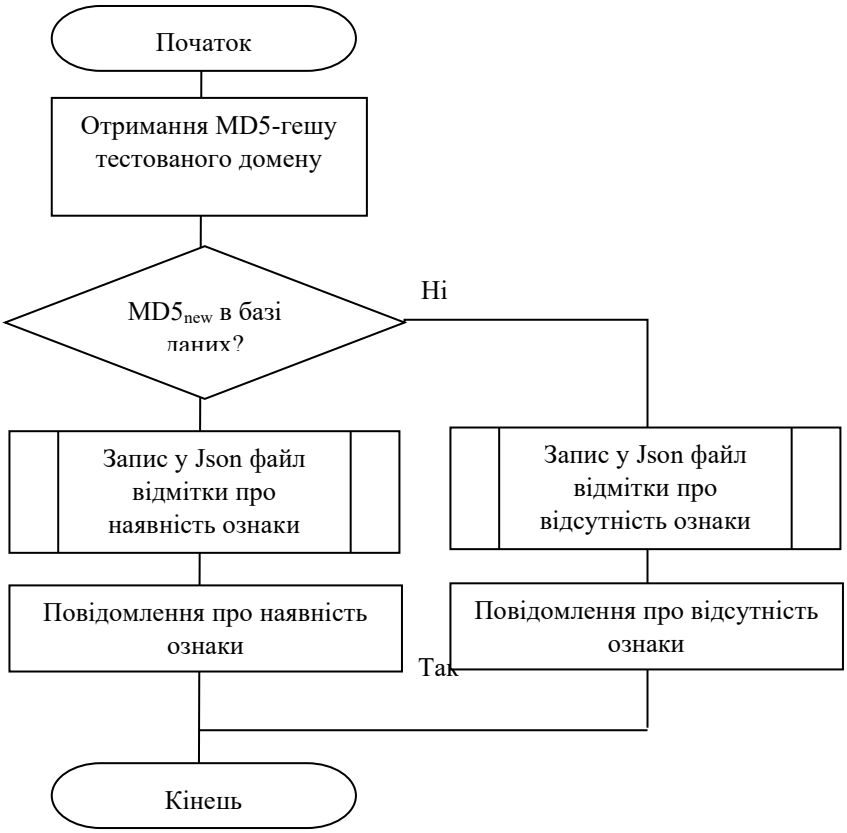


Рисунок 3.5 – Алгоритм роботи процедури перевірки наявності MD5-гешу в базі

Після отримання адреси веб-сайту у вигляді MD5 гешу відбувається порівняння з наявними хешами у базі даних. Якщо таких хеш вже наявний, то розрахунок ймовірності наявності фішингу буде відбуватись з врахуванням того, що тестований сайт вже був перевірений засобом. В інакшому випадку, перевірка веб-сайту буде відбуватись за стандартим алгоритмом. Процедура визначення наявності захищеного з’єднання перевіряє отриману адресу, інтерпретуючи її як рядок, і визначає, з чого починається цей рядок. Якщо користувач надав посилання без приставки http або https, визначення наявності захищеного з’єднання здійснюється на основі результатів аналізу сертифікатів. Алгоритм роботи процедури представлений у вигляді блок-схеми на рисунку 3.6

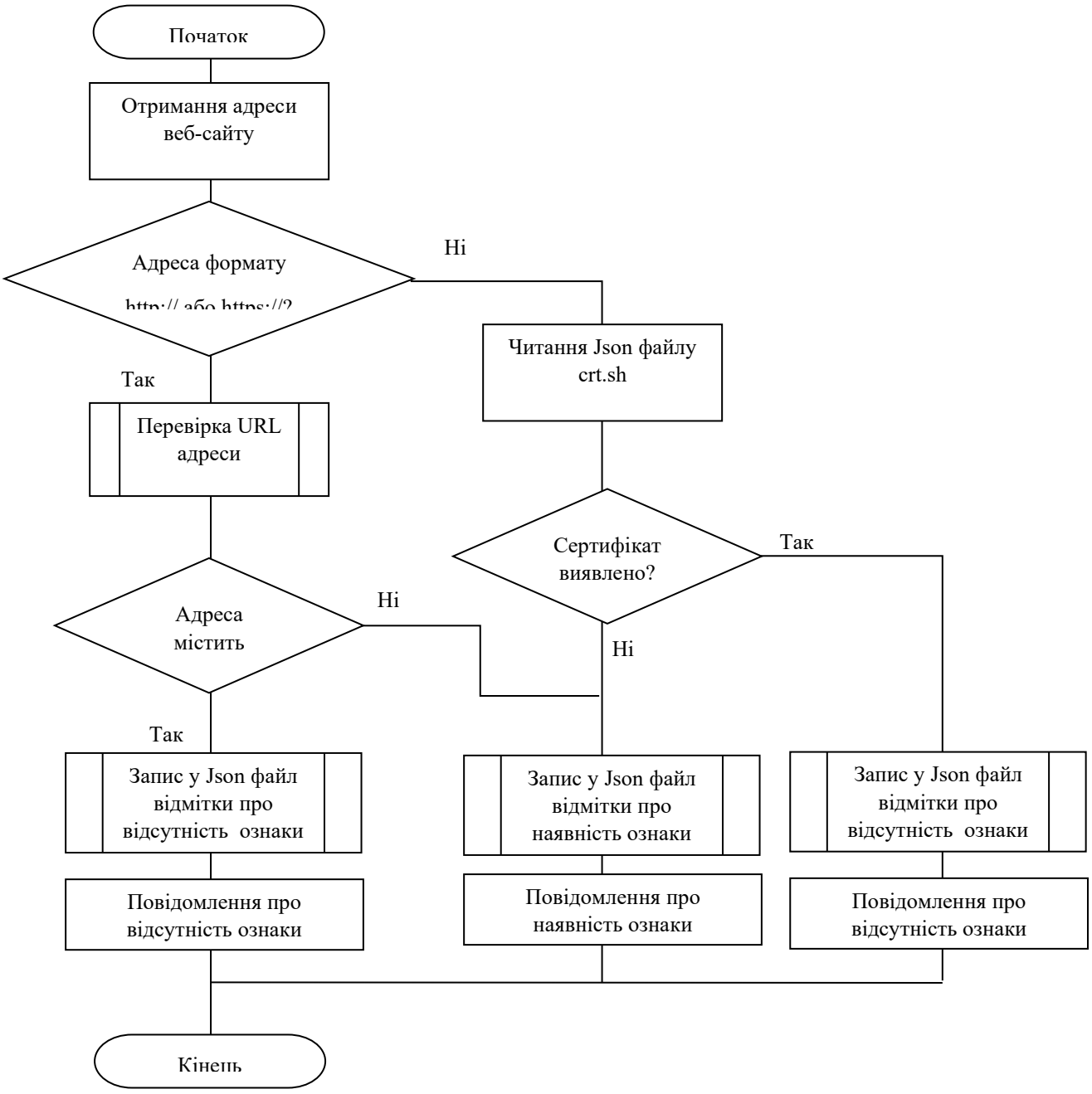


Рисунок 3.6 – Алгоритм роботи процедури перевірки наявності https

Таким чином після отримання адреси сайту здійснюється перевірка чи наявний протокол у форматі введення. Якщо протокол наявний процедура визначає початкові значення читаючи формат посилання, та визначає чи присутній протокол https, у випадку, якщо результатом перевірки є http:// відбувається відмітка про наявність ознаки незахищеного з’єднання, як наслідок це буде враховуватись при розрахунку ймовірності ризику. В іншому випадку це буде визначено як відсутність ознаки. Наступна процедура перевіряє наявність ключових слів в

домені. Для цього на сервері міститься список популярних бренд-цілей, які часто використовуються в доменах фішингових сайтів. Після отримання домену, відбувається читання словника та перевіряється наявність ключового слова по даному словнику. Алгоритм роботи наведено у вигляді блок-схеми на рисунку 3.7

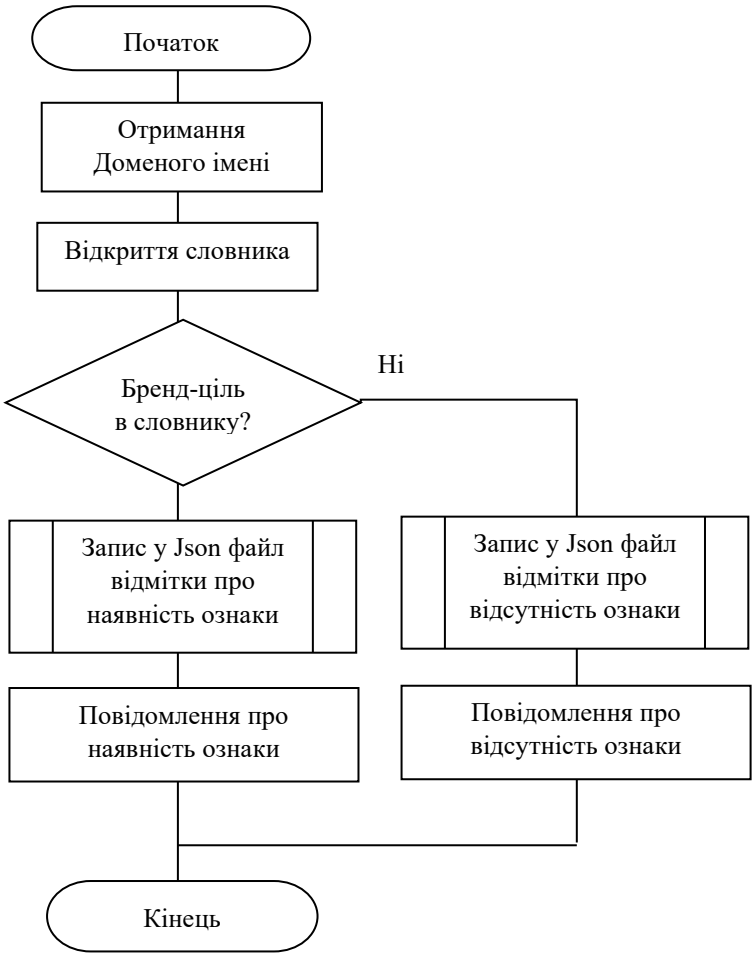


Рисунок 3.7 – Алгоритм роботи процедури визначення ключових слів у домені

Наявність обфускації коненту відбувається шляхом перевірки декількох елементів. Аналіз відбувається на основі пошуку прихованих та закодованих елементів. Для цього отримується DOM структура сторінки. Після чого відбувається перевірка на наявність довгих рядків, даних у вигляді Base64 включаючи медіа дані, а також відбувається пошук прихованих елементів, які приховані за допомогою атрибутів none, hidden, opacity 0. Якщо під час перевірки підтверджується хоча б одна умова, в файл звіту відмічається позначка, що ознака ідентифікована.

В іншому випадку викликається наступна процедура, яка перевіряє структуру сайту на наявність скриптів із шкідливим вмістом. Аналіз відбувається шляхом отримання HTML-коду сторінки та перевірки наявності підозрілих JavaScript-конструкцій. Скрипти з шкідливим вмістом як правило використовують функції eval, Function, setTimeout, а також функції декодування, такі як decodeURIComponent, atob та btoa. Ці елементи часто використовуються для динамічного виконання або приховування шкідливого коду. У випадку якщо хоча б один з перерахованих елементів наявний в коді це вважається виявленою ознакою, та слугує підставою для підвищення ймовірноссті наявності фішингу.

Наступна процедура відповідає за перевірку редиректу шляхом заборони слідувати за редиректами. Якщо редирект відбудеться сервер надішле відповідь який вказує на спробу редиректу що буде ідентифіковано, як наявність ознаки.В іншому випадку сервер не передаватиме ніякої відповіді що буде вказувати на те, що редирект не відбувся. Завершальною процедурою є перевірка чи змінюється вміст сайту , використовуючи різні заголовки User-Agent для імітації різних браузерів. Запити також виконуються через проксі, щоб перевірити, чи змінюється відповідь залежно від маршруту за якими передаються дані.

Після збору всіх відповідей відбувається їх порівняння. Якщо хоча б дві відповіді відрізняються, це свідчить про адаптацію веб-сайту до різних User-Agent або проксі, що є ознакою потенційного фішингу. Далі перевіряються усі ідентифіковані ознаки та проводиться розрахунок ймовірності наявності фішингу. Згенеровані JSON-файли містять відповідні записи про ознаки. Якщо певна ознака виявлена, вона враховується у фінальному розрахунку ризику.

3.5 Висновки з розділу

Розроблена архітектура засобу дозволяє поєднувати різні підходи для використання засобу виявлення фішингових сайтів. Користувач має змогу здійснювати перевірку за допомогою веб-сервісу, який відіграє роль серверу, а також за допомогою розширення для браузера що дозволяє швидко отримати дані

про проаналізований домен без потреби переходу на веб-сервіс. Єдина відмінність у використанні веб-сервісу та розширенні полягає у тому, що при скануванні за допомогою веб сервісу, користувачу надається розгорнута інформація про просканований домен, у той час коли сканування через розширення повертає лише ймовірність наявності фішингу.

Реалізована архітектура серверу дозволяє поєднувати результати аналізу з зовнішніх веб-сервісів із внутрішнім аналізом який виконує сам сервер. Довірені веб-сервіси надають можливість швидко отримувати ключову інформацію про домен, здійснювати ряд перевірок по власним базам надаючи оперативно інформацію про наявності вже виявлених ознак, а внутрішній аналіз домену та ідентифікація ознак дозволяють серверу виконувати багаторівневу перевірку, враховуючи як зовнішні дані, так і результати внутрішнього аналізу. Це забезпечує можливість не лише виявляти відомі фішингові сайти, але й аналізувати нові, враховуючи обставини створення сайту. Запам’ятовування MD5 гешу домену надає можливість поповнювати список вже просканованих сайтів, що пришвидшує роботу, та частково автоматизує процес виявлення фішингових сайтів. Чим більша кількість перевірок відбувається, тим більше ідентифікованих сайтів наявно. Важливим аспектом є те, що навіть якщо сайт вже виявлений у базі даних, засіб актуалізує його дані для перерахування ризику. Такий підхід дозволяє не лише збільшувати власну базу даних, а й залишати її постійно актуальною.

4 ТЕСТУВАННЯ МЕТОДУ ТА ЗАСОБУ ВИЯВЛЕННЯ ФІШИНГОВИХ ВЕБ-САЙТІВ

4.1 Обґрунтування засобів розробки

Для реалізації засобу виявлення фішингових веб-сайтів необхідно створити клієнт-серверну архітектуру, яка включатиме фронтенд-частину та сервер. Фронтенд забезпечуватиме взаємодію користувача із засобом, а серверна частина відповідатиме за обробку даних та виконання основної логіки перевірок. Перед вибором інструментів розробки варто визначити вимоги, враховуючи специфіку реалізації засобу виявлення фішингових веб-сайтів. Таким чином основні вимоги до засобів розробки включають:

- забезпечення кросплатформеності для підтримки різних операційних систем;
- підтримка інтеграції з популярними інструментами та бібліотеками;
- висока продуктивність при роботі з великими обсягами даних;
- доступність бібліотек для роботи з HTTP-запитами, кешуванням;
- зручність розробки та налагодження;

Для реалізації клієнтської та серверної частини були розглянуті такі мови програмування: Python, PHP, JavaScript. Python є гнучкою мовою з великою кількістю бібліотек для роботи з HTTP-запитами. Завдяки доступності Redis Python забезпечує швидке кешування результатів перевірок, що знижує навантаження на сервери та покращує продуктивність [25]. Проте Python використовується здебільшого на серверній стороні та не забезпечує взаємодії з клієнтською частиною без інтеграції інших мов. PHP має широке використання для веб-розробки, легко інтегрується з HTML5 і JavaScript. Однак ця мова менш продуктивна для обробки великих обсягів даних і не має такої гнучкості, як Python, при роботі з API-сервісами та кешуванням. JavaScript є універсальною мовою, яка використовується як на клієнтській стороні, так і на серверній. Вона підтримує асинхронність і має доступ до багатьох бібліотек для роботи з HTTP-запитами.

Проте, JavaScript поступається Python у зручності обробки складних операцій на сервері.

З огляду на вимоги, мову Python було обрано для серверної частини засобу, а для фронтенду поєднання HTML5, CSS і JavaScript. Python забезпечує легкість інтеграції з Redis, обробку HTTP-запитів і роботу з API, тоді як JavaScript відповідає за динамічність клієнтської частини.

Для забезпечення зручності роботи над проектом важливим є вибір інтегрованого середовища розробки, яке дозволить знизити ймовірність помилок та оптимізувати процес розробки. Враховуючи специфіку проєкту, було розглянуто декілька середовищ: PyCharm, Visual Studio Code та Sublime Text.

Хоча Visual Studio Code має великий функціонал завдяки плагінам, його використання вимагає ручного налаштування що значно уповільнює процес [26]. Sublime Text, є більш обмеженим за функціоналом і не забезпечує повної підтримки для Python-проектів. PyCharm, розроблений спеціально для Python, є найбільш оптимальним середовищем для реалізації. Дане середовище розробки забезпечує автоматичне заповнення коду, перевірку помилок у реальному часі, підтримку Redis і можливість інтеграції з тестовими інструментами [27]. PyCharm також має зручні інструменти для налагодження, що дозволяють ефективно тестувати програму.

Таким чином, розробка засобу буде виконуватись в середовищі PyCharm із використанням таких технологій, як Python, Redis, HTML5, JavaScript і CSS. Обрані інструменти забезпечать продуктивність, масштабованість і легкість розробки як клієнтської, так і серверної частин системи. Це гарантує швидкий процес розробки, зручну підтримку та можливість подальшого розширення проекту.

4.2 Блокове тестування засобу виявлення фішингових веб-сайтів

Блокове тестування виконується з метою перевірки роботи окремих функцій, визначення помилок у їх виконанні та забезпечення коректності взаємодії між компонентами. Такий підхід дозволяє ізолювати кожен модуль і протестувати його

незалежно від інших частин системи. Це дає змогу швидко виявити недоліки у функціональності та виправити їх без впливу на інші частини програми.

Таким чином блокове тестування виконується для основних компонентів програми. Тестування роботи з Redis необхідне для забезпечення коректності взаємодії системи з чергою доменів. Робота з чергою є основою для обробки даних у засобі. Якщо взаємодія з Redis не буде працювати належним чином, це призведе до помилок у процесі аналізу та втрати даних. Таким чином необхідно перевірити такі правильність виконання:

- робота з порожньою чергою;
- додавання до черги;
- отримання домену з черги;
- обробка кількох доменів у черзі.

Перевірка поведінки з порожньою чергою забезпечує коректність роботи системи, коли в черзі немає жодного домену. При такій реалізації засіб знаходитиметься в постійному стані очікування. Це забезпечує стабільну роботу засобу та уникнення помилок у разі відсутності даних для обробки. Додавання та вилучення домену з черги реалізоване для перевірки правильності зчитування домену. Це забезпечує цілісність даних і підтверджує, що доданий запис доступний для зчитування. Тестування роботи з кількома доменами здійснює перевірку можливості додавання декількох записів до черги. Дане тестування підтверджує, що черга підтримує порядок записів, а всі домени зберігаються належним чином і доступні для подальшої обробки. Результати блокового тестування наведені на рисунку 4.1

З отриманих результатів можна зробити висновок, що усі описані тести пройдені, що свідчить про коректну роботу засобу з Redis. Тестування основних функцій спрямоване на перевірку основного функціоналу аналізу доменів який знаходиться в класі SiteChecker.

```
(.venv) PS C:\Users\Savch\PycharmProjects\DiplomaWork> python -m unittest test_phishing.py -v
test_empty_redis_queue (test_phishing.TestRedisOperations)
Test behavior with an empty Redis queue ... ok
test_multiple_domains (test_phishing.TestRedisOperations)
Test handling multiple domains in Redis ... ok
test_redis_add_and_retrieve (test_phishing.TestRedisOperations)
Test adding and retrieving a domain from Redis ... ok
test_retrieve_domain_from_redis (test_phishing.TestRedisOperations)
Test retrieving a domain from Redis ... ok

----------------------------------------------------------------------
Ran 4 tests in 0.024s

OK
```

Рисунок 4.1 – Результати блокового тестування Redis

Некоректне виконання даних функцій призведе до невірних результатів, таким чином важливо перевірити такі функції:

- перевірка визначення протоколу;
- генерація MD5-гешу ;
- виявлення ключових слів;
- інтеграція з довіреними веб-сервісами;
- перевірка SSL сертифікатів;
- формування фінального звіту.

Перевірка визначення протоколу спрямована на коректну ідентифікацію яка необхідна у разі якщо користувач ввів лише домене ім’я. Тестування генерації гешу необхідне для перевірки точності створення унікальних ідентифікаторів доменів а також правильності їх під час порівняння з вже наявним у базі. Виявлення ключових слів перевіряє здатність засобу знаходити підозрілі назви доменів які можуть маскуватись під легітимні. Тестування інтеграції з довіреними веб-сервісами націлене на перевірку встановлення з’єднання, передачі запиту, та отримання відповіді. Тестування знаходдження SSL сертифікатів реалізоване з метою підтвердження того, що система правильно перевіряє їх наявність та підраховує кількість. Тестування правильності формування звіту гарантує, що дані отримані під час всіх перевірок правильно об’єднуються та подаються у

коректному вигляді. Результати блокового тестування CheckSite наведені на рисунку 4.2

```
(.venv) PS C:\Users\Savch\PycharmProjects\DiplomaWork> python -m unittest test_phishing.py -v
test_certificate_check (test_phishing.TestSiteChecker)
Test SSL certificate existence ... ok
test_final_report_generation (test_phishing.TestSiteChecker)
Test final report generation ... ok
test_https_check (test_phishing.TestSiteChecker)
Test the correctness of the HTTPS check ... ok
test_keyword_detection (test_phishing.TestSiteChecker)
Test keyword detection in the domain name ... ok
test_md5_hash_generation (test_phishing.TestSiteChecker)
Test MD5 hash generation for a domain ... ok
test_virustotal_integration (test_phishing.TestSiteChecker)
Test integration with the VirusTotal API ... ok
test_whois_lookup (test_phishing.TestSiteChecker)
Test WHOIS data processing ... ok

----------------------------------------------------------------------
Ran 7 tests in 2.536s
```

Рисунок 4.2 – Результати блокового тестування SiteChecker

З результатів тестування можна зробити висновок що всі тести виконались без помилок, що свідчить про коректність їх роботи. Таким чином виконання блокового тестування для засобу виявлення фішингу можна вважати успішним та переходити до наступних типів тестування

4.3 Інтеграційне тестування засобу виявлення фішингових веб-сайтів

Метою проведення інтеграційного тестування, є перевірка коректності взаємодії між окремими компонентами засобу для забезпечення їх узгодженої роботи Таке тестування дозволяє виявити помилки, що виникають у процесі обміну даними або взаємодії між окремими частинами програми, які вже пройшли етап модульного тестування. Важливо врахувати не тільки роботу серверу і сайту, а й розширення, яке також надсилає та отримує дані. Для проведення тестування в даному випадку достатньо надіслати запит з клієнтської частини на сервер. В свою

чергу сервер поверне дані в консолі. Таким чином інтеграційне тестування роботи сайту та сервера наведено на рисунку 4.3

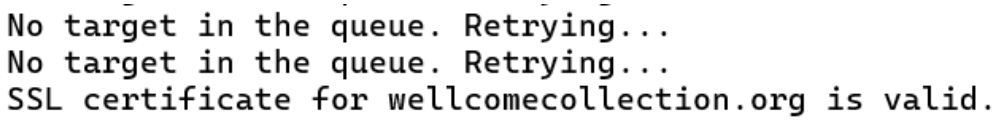


Рисунок 4.3 – Результат тестування взаємодії сайту з сервером

Після отримання запиту сервер починає здійснювати аналіз який займає певний час, після чого на клієнт надсилаються дані. З результату проведеного тестування можна зробити висновок, що сервер коректно обробляє запити які надходять з сайту. Наступним етапом необхідно перевірити коректність надсилання запитів з розширення на сервер, та їх повернення. Аналогічним чином необхідно здійснити запит, та перевірити консоль, в якій буде наведено статус отриманого запиту сервером, та статус надсилання відповіді на розширення. Результати тестування взаємодії розширення та серверу наведено на рисунку 4.4

Рисунок 4.4 – Результат тестування взаємодії сайту з розширенням

Результат проведеного тестування свідчить про коректне отримання запиту від розширення до серверу. На основі інтеграційного тестування можна зробити висновок, що сервер отримує запит від клієнтської частини та повертає відповідь, що підтверджує коректність роботи клієнт-серверної архітектури, та забезпечує узгоджену роботу між окремими компонентами.

4.4 Мануальне тестування інтерфейсу виявлення фішингових веб-сайтів

Мета мануального тестування полягає у забезпеченні перевірки функціональності без залучення автоматизованих інструментів. Здійснення

тестування відбувається від лиця користувача, таким чином можна перевірити нестандартні сценарії які можуть бути реалізовані під час використання засобу виявлення фішингових сайтів. Для отримання задовільного результату під час тестування необхідно слідувати таким етапам:

- Перейти на сайт https://phishscan.info/
- В поле введення адреси сайту ввести валідну адресу, або валідне ім’я домену.
- Дочекатись завершення тестування, у випадку, якщо сторінка реультатів з’явилась швидше за результат, зачекати. Така поведінка обумовлена рівнем завантаженості сторонніх веб серверів.
- Отримати результати тестування у вигляді заповнених секцій відповідно до кожної.

За даними етапами, визначено вимоги яким має відповідати засіб для успішного тестування. Результати тестування мають відповідати таким вимогам:

- Введення адреси доступне у будь якому форматі;
- сторінка результатів з’являється лише після завершення процесу сканування;
- визначення ймовірності наявності фішингу у категоріальному форматі;
- дані отримані під час сканування заповнені по секціям відповідно до заголовків.

Таким чином, необхідно ввести відповідну адресу домену, та натиснути кнопку «Сканувати». Введення адреси в поле різними форматами зображено на рисунку 4.6.

Як і очікувалось у всіх трьох випадках створюється запит на сервер, та сервер починає обробку, з врахуванням можливості введення даних різними форматами. Наступним етапом відбувається процес сканування, після чого з’являється вікно з відповідною інформацією. Результат сканування представлено на рисунку 4.7

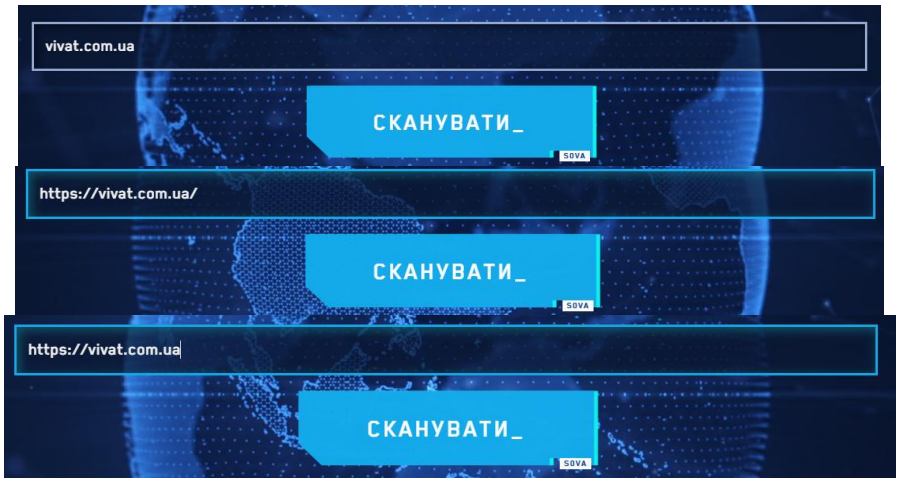


Рисунок 4.6 – Приклади форматів введених адрес в поле введення

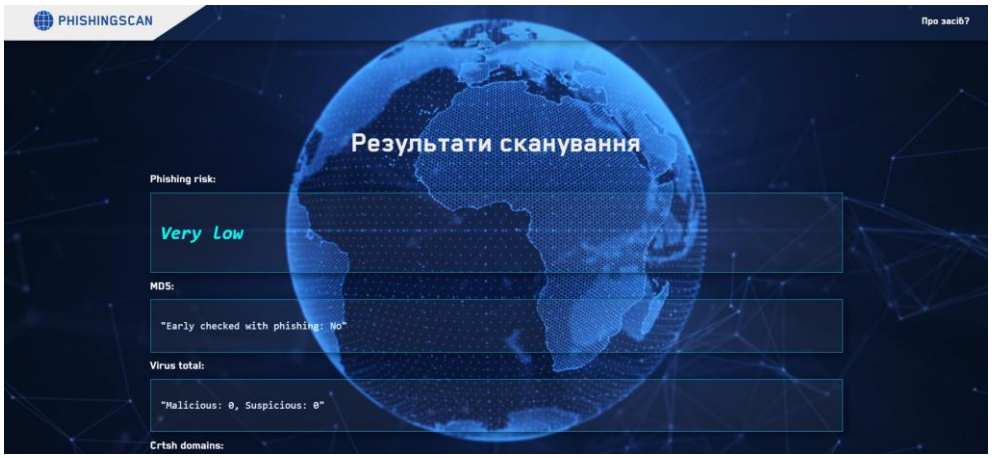


Рисунок 4.7 – Результат сканування введених адрес

Таким чином було протестовано 3 варіації адрес. У всіх спробах тестування результат був однаковий. З даних результатів можна зробити висновок, що введення адрес в різних форматах, не спричиняє виявлення помилок.

За аналогією було протестовано фішинговий сайт, який вже містився в базі даних засобу адреси вводились у таких самих форматах, як і для легітинмного. Результат тестування наведено на рисунку 4.8.

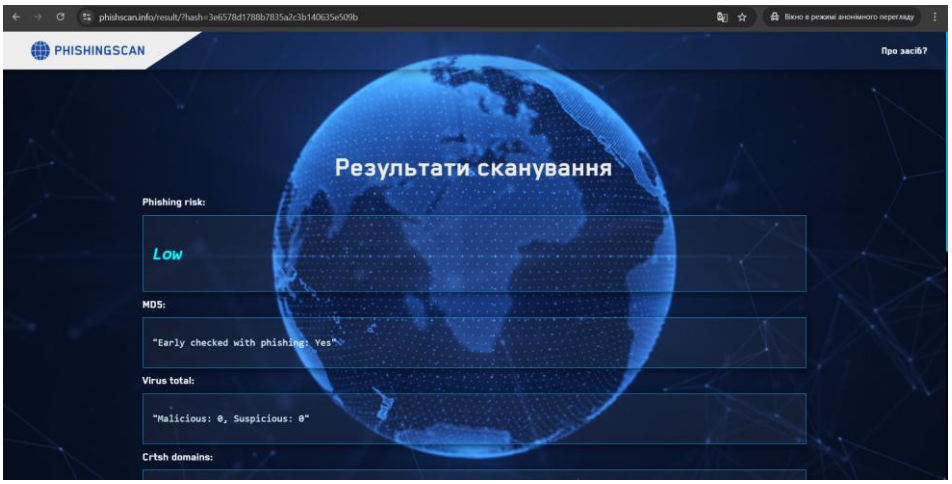


Рисунок 4.8 – Результат сканування введених адрес фішингового сайту

Результати відповідні до процедури тестування легітимного сайту та свідчать, що незалежно від формату посилання сайту, сервер сприймає адресу вірно та надсилає коректну відповідь розташовуючи дані у секції які відповідають заголовкам. На основі цього можна зробити висновок, що тестування сайту вдале та відповідає визначеним вимогам. Аналогічно, необхідно провести тестування розширення. Для цього необхідно провести тестування розширення. Так як функціонал націлений на портативність та зручність, для успішного сканування сайту необхідно виконати такі дії:

– Встановити розширення
– В поле введення адреси сайту ввести валідну адресу, або валідне ім’я домену.
– Дочекатись завершення тестування.
– Отримати результати тестування у вигляді категоріального рівня ризику.

Введені дані в поле адреси в розширені, наведено на рисунку 4.9

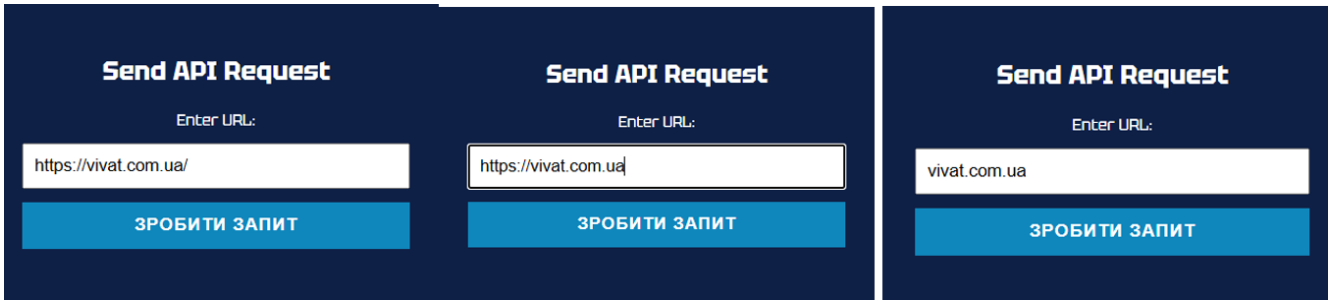


Рисунок 4.9 – Приклади форматів введених адрес в поле введення розширення

Після надсилання запиту, відбудеться сканування домену сервером, результатом якого буде передача, ймовірності ризику, яке буде передане в розширення. Результат виконання запиту наведено на рисунку 4.10.

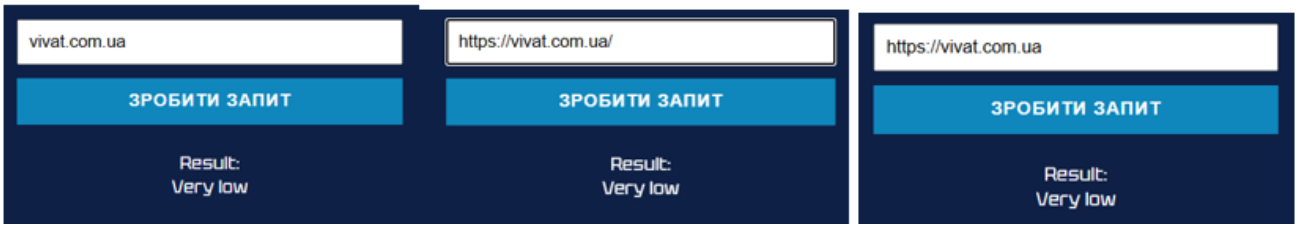


Рисунок 4.10 – Результат сканування з використанням розширення

Як і очікувалось сканування повертає однаковий рівень ризику для сайту адреса якого введена в різних форматах. Таким чином мануальне тестування виконане відповідно до поставлених вимог.

4.5 Експериментальне дослідження засобу виявлення фішингових веб-сайтів

Проведення експериментального дослідження має на меті визначення, здатності системи класифікувати сайти на довірені та фішингові, використовуючи задані ознаки та критерії аналізу. Оцінити коректність роботи алгоритмів у реальних умовах, включаючи взаємодію з зовнішніми сервісами.

Так, як всі етапи сканування, записуються у json файли, доцільно перевірити наявність в них інформації та правильності їх введення, для цього було обрано 2 фішингових сайти, які в поєднанні містили б максимальну кількість всіх можливих ознак та 2 легітимних. Першим етапом сканування створюється файл check_md5.json у якому міститься основна інформація така як MD5 хеш, наявність в базі вже ідентифікованих як фішинговий та статусу. Результат тестування продемонстровано на рисунку 4.11

Рисунок 4.11 – Результат запису даних у JSON-файл check_md5.json для фішингового сайту.

Файл зберігається у каталозі звіту, при цьому сайт уже був ідентифікований як фішинговий, що підтверджується статусом already_present = true, який вказує на його наявність у базі. Легітимні сайти не додаються до бази даних під час тестування, а результати їх сканування зберігаються окремо для демонстрації записів. Результат перевірки легітимного сайту із збереженням MD5-гешу наведено на рисунку 4.12

Рисунок 4.12 – Результат запису даних у check_md5.json для легітимного сайту.

Наступним кроком здійснюється перевірка запису даних, які надсилають зовнішні веб сервіси для початку здійснюється запит до сервісу Whois, який повертає дані серверу, в результаті чого формується json файл. Результат запису інформації від сервісу наведено на рисунку 4.13.

Рисунок 4.13 – Результат запису у Whois.json отриманого від Whois.

В результаті тесту створюється файл із записом інформації про домен, що включає неймсервери, реєстратора та дату реєстрації. Отримані дані враховуються під час перевірок умов ідентифікації ознак. У файлі check_date.json зберігається інформація про те, чи був домен зареєстрований або поновлений за останні 6 місяців. Результат запису даних наведено на рисунку 4.14.

Рисунок 4.14 – Результат запису у check_date.json

Результат свідчить, що засіб коректно визначає факт раннього строку життя домену. Наступний веб-сервіс здійснює перевірку сертифікатів, на основі яких отримує список доменів для яких були видані SSL/TLS сертифікати. Результат запису відповіді від сервісу наведено на рисунку 4.15.

```
rclient.rpush( name: "target:domain",  *values: "https://jetiq.vntu.edu.ua/")
```

```
{
  "sertificates": [
    "aiit.vntu.edu.ua",
    "aivt.vntu.edu.ua",
    "aspir.vntu.edu.ua",
    "docs.vntu.edu.ua",
    "iq.vntu.edu.ua",
    "jetiq.vntu.edu.ua",
    "wiki.vntu.edu.ua"
  ]
}
```

Рисунок 4.15 – Результат запису у sertificate_data.json списку доменів для яких був виданий сертифікат

Наступним кроком, здійснюється перевірка на наявність редіректів на сервісі. Для цього функція перевіряє, чи перенаправляє сайт на іншу сторінку. Якщо є перенаправлення, вона визначає кінцеву адресу, на яку веде редирект. Тестувння проводилось на сайті, на якому встановлений редірект та записує у файл redirect.json. Результати тестування наведені на рисунку 4.16.

```
{
  "destination": "https://www.amazon.com/ap/signin?z=signin",
  "redirect": true
}
```

Рисунок 4.16 – Результат запису у redirect.json наявності редіректу

Результат свідчить про наявність редиректу, що підтверджується практичною перевіркою, тому даний запис є коректним. Наступним етапом здійснюється перевірка сервісом VirusTotal. Результат записується у файл vt_result.json, який містить кількість знайденого шкідливого вмісту на основі сканування антивірусами. Результати запису відповіді від сервісу наведено на рисунку 4.17

Рисунок 4.17 – Результат запису у vt_result.json для фішингвого сайту

Таким чином можна зробити висновок, що даний сервіс містить велику кількість загроз різних типів. Аналогічно проведено перевірку для легітимного веб-сайту. Результати сканування, наведено у вигляді рисунку 4.18

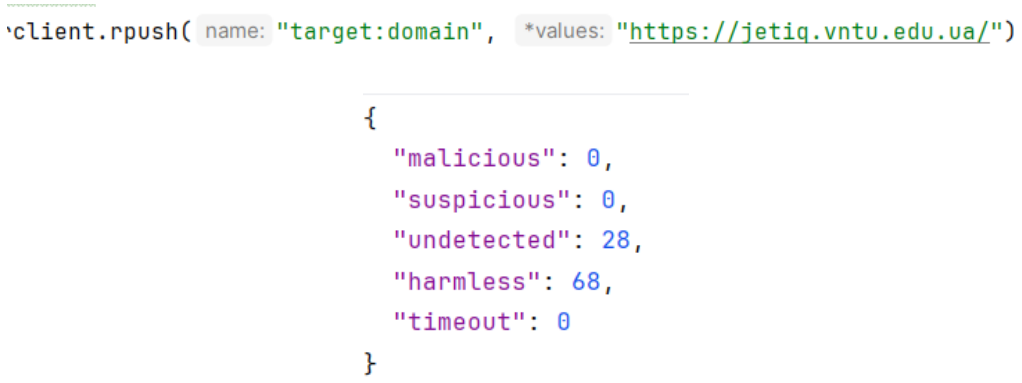


Рисунок 4.18 – Результат запису у vt_result.json для легітимного сайту

Перевірка веб-сайту на наявність обфускації скриптів шляхом читання коду та виявлення в ньому шкідливих елементів за визначеними параметрами. Отримані дані записуються у файл obfuscation.json. Результати тестування наведено на рисунку 4.19.

При проведені ручної перевірки виявлено скрипт який, ймовірно є шкідливим. Скрипти побудовані такою структурою як правило мають на меті приховати свій справжній функціонал.

```
rclient.rpush( name: "target:domain",  *values: "https://steeamcommnity.com/sielit/vlonet/cicul")
```


```
"matches": [
  {
    "pattern": "eval\\(",
    "matches": 0
  },
  {
    "pattern": "Function\\(",
    "matches": 1
  },
  {
    "pattern": "setTimeout\\(",
    "matches": 1
  },
  {
    "pattern": "decodeURIComponent|atob|btoa",
    "matches": 2
  }
```


Рисунок 4.19 – Результат запису у файл obfuscation.json

Структура скрипта містить велику кількість нелогічних елементів, зашифрованих даних та функцій виклику ззовні, а також затримок наявність яких підвищує ризик завантаження динамічного коду. Тому даний скрипт можна вважати шкідливим.

Наступним кроком здійснюється тестування формування результату. Всі дані, необхідні для визначення ймовірності фішингу тестованого сайту, зберігаються у файл result.json. Файлі містить коефіцієнти ймовірності фішингу, розраховані на основі отриманих даних під час сканування. Результат запису розрахунку ймовірності фішингу наведений на рисунку 4.20.

```
rclient.rpush( name: "target:domain",  *values: "https://steeamcommnity.com/sielit/vlonet/cicul")
```


```
  "vt_scan": {
    "coef": 1.764,
    "data": "Malicious: 24, Suspicious: 1"
  },
  "md5_hash": {
    "coef": 1.1039999999999999,
    "data": "Early checked with phishing: Yes"
  },
  "content_obfuscation": {
    "coef": 0.44799999999999995,
    "data": "Content obfuscation: Yes"
  },
  "script_obfuscation": {
    "coef": 0.324,
    "data": "Script obfuscation: Yes"
  },
  "negh_domains": {
    "coef": 0.576,
    "data": "Present neighbors on CloudFlare: Yes'
  },
  "general": 0.5819961938697006
}
```


Рисунок 4.20 – Результат запису розрахунку ймовірності фішингового сайту

Аналогічно проведено перевірку запису для легітимного сайту. Результат запису розрахунку ймовірності наведено на рисунку 4.21

Рисунок 4.21 – Результат фрагменту формування загального звіту

Таким чином засіб визначає ймовірність фішингу, після чого на клієнтській частині отримане значення інтерпретується у категоріальний показник порівняльним методом. Користувач отримає у відповідь в першому випадку значення Height а у другому Low.

Програма повинна бути здатною виявляти фішингові сайти на основі аналізу характеристик. Перш за все необхідно врахувати корегуючі коефіцієнти які розраховуються на основі матриці класифікації сайтів. Таким чином для тестування роботи без врахування коефіцієнтів було сформовано вибірку з 10 сайтів, 4 з яких завчасно відомі як фішингові, відповідно до бази даних PhishTank. Результати тестування наведено в таблиці 4.1.

Таблиця 4.1 – Результати помилкових спрацювань

Ознака	**FP**	**FN**	**TP**	**TN**
Перенаправлення	2	1	3	4
Ключові слова	2	2	2	4
Відкриття з різних локацій	1	2	2	5
Відсутність сертифікатів	0	0	4	5
Свіжа дата реєстрації домену	1	1	3	5
Попередження від сервісів сканувань	0	0	4	6
Обфускація контенту	1	2	2	5
Обфускація скриптів	2	2	2	4
Виявлені сусідні домени	0	1	3	6

:

TP – сайт правильно класифіковано як фішинговий.

FP – легітимний сайт помилково класифіковано як фішинговий.

FN – фішинговий сайт помилково класифіковано як легітимний.

TN – сайт правильно класифіковано як легітимний.

На основі отриманих даних розраховуються, корегуючі коефіцієнти для кожної з ознак. Приклад розрахунку корегуючого коефіцієнта для здійснений для ознаки наявності ключових слів у домені. Першим етапом розраховується частка помилково позитивних спрацювань.

$$FPR=2/(2+4)=0.3$$

Наступним етапом відбувається розрахунок частки помилково негативних спрацювань

$$FNR=2/(2+2)=0.5$$

Отримання цих данних дозволяє розрахувати корегуючий коефіцієнт

$$a_{keyword} = 1\text{-}0.25 + (1\text{-}0.5) = 1.2$$

Аналогічним чином проведені розрахунки для решти ознак. Після інтеграції корегуючих коефіцієнтів проведено повторне тестування вибірки, для створення порівняльного аналізу. Оновлені результати повторного тестування наведено у таблиці 4.2

Таблиця 4.2 – Результати помилкових спрацювань з врахуванням корегуючого коефіцієнта

Ознака	**FP**	**FN**	**TP**	**TN**
Перенаправлення	1	0	4	5
Ключові слова	1	1	4	5
Відкриття з різних локацій	0	0	5	6
Відсутність сертифікатів	0	0	4	5
Свіжа дата реєстрації домену	0	1	4	6
Попередження від сервісів сканувань	0	0	4	6
Обфускація контенту	1	1	4	5
Обфускація скриптів	1	0	4	6
Виявлені сусідні домени	0	0	4	6

Результати повторного тестування демонструють значне покращення точності виявлення фішингових сайтів, що свідчить про підвищення чутливості системи до ознак фішингових сайтів. Ознаки, які раніше демонстрували неоднозначність у класифікації, стали більш надійними, забезпечуючи підвищення точності виявлення фішингових сайтів. З отриманих даних можна зробити висновок, що точність виявлення покращилась на 0.17.

4.6 Висновки з розділу

Проведення різних типів тестування дозволило впевнитись, що засіб для виявлення фішингових сайтів працює коректно, згідно з визначеними вимогами. Результати блокового тестування продемонструвало правильність роботи окремих частин засобу, коректність роботи черги, здатність черги знаходитись тривалий час в стані очікування, приймати та віддавати дані, а також створювати чергу відповідно до порядку сканувань. Тестування основних функцій вказує на коректність роботи окремих модулів програми результат яких продемонстрував вірну перевірку визначення протоколу, генерацію гешу домену, що тестується та перевірку наявності в базі, виявлення ключових слів, інтеграцію з довіреними вебсервісами, правильне визначення SSL сертифікатів, та коректне формування загального звіту. Результати експериментального дослідження демонструють

здатність системи коректно інтерпретувати отримані шляхом аналізу дані, проводити маніпуляції з ними у вигляді запису та розрахунків, а також формувати коректні звіти. Визначення корегуючих коефіцієнтів дозволило збільшити чутливість засобу до фішингових сайтів підвищивши ймовірність правильного виявлення фішингового веб-сайту. Результатом інтеграційного дослідження є успішна перевірка зв’язку засобу між різними компонентами такими як веб-сайт та сервер, а також розширення і сервер, в результаті тестування в обох випадках сервер приймав та надсилав коректні дані. Мануальне тестування дозволило перевірити роботу засобу у випадку виконання непередбачуваних сценаріїв які можуть бути виконані з боку користувача.

Як результат отримано протестований засіб, з клієнт-серверною архітектурою, який виконує коректний, та правильний аналіз згідно з ідентифікованими ознаками, формує звіти, та подає їх у вигляді різного обсягу відповідно до середовища з якого здійснено запит.

5 ЕКОНОМІЧНА ЧАСТИНА

5.1 Комерційний та технологічний аудит науково-технічної розробки

Технологічний аудит методу та засобу виявлення фішингових сайтів має на меті повну оцінку його ефективності для визначення науково-технічного рівня та рівня комерційного потенціалу розробки, створеної в результаті науково-технічної діяльності.

Найбільш схожим за функціоналом та реалізацією є сервіс CheckPhish від американської компанії Bolster. Вартість пакетів рівних розробленому засобу становить приблизно 4145 грн. Критерії оцінювання комерційного потенціалу розробки та їх оцінка представлені в таблиці (додаток Д).

Оцінювання науково-технічного аудиту відбувається з участю трьох незалежних експертів: Ляшко П.О – директор ТОВ «Українські кібернетичні розробки»,Кореньков А.О – спеціаліст з інформаційної безпеки ТОВ «Траффік.ван»,Творун Р.П – спеціаліст з інформаційної безпеки ТОВ «Українські кібернетичні розробки»

Відповідно до критеріїв результати оцінювання комерційного потенціалу роботи представлено у таблиці 5.1.

Таблиця 5.1 – Результати оцінювання науково-технічного рівня і комерційного потенціалу розробки

Критерії оцінювання	Експерти		
	Творун Р.П	Ляшко П.О	Кореньков А.О
	Бали		
Технічна здійсненність концепції	3	4	4
Наявність аналогів на ринку	3	3	3
Цінова політика	3	5	4
Технічні та споживчі властивості виробу	3	3	4
Експлуатаційні витрати	4	3	3
Ринок збуту	3	4	3
Конкурентоспроможність	3	3	4
Фахівці з технічної і комерційної реалізації	3	3	3
Фінансування	4	3	3
Матеріально-технічна база	4	3	4
Термін реалізації ідеї	4	5	3
Супровідна документація	3	3	4
Сума	40	42	42
Середньоарифметична сума балів	41		

Визначення рівня комерційного потенціалу методу та засобу виявлення фішингових сайтів відбувається відповідно до рекомендацій представлених в таблиці 5.2.

Таблиця 5.2 – Рекомендації визначення рівня комерційного потенціалу розробки

Середньоарифметична сума балів СБ , розрахована на основі висновків експертів	Рівень комерційного потенціалу розробки
0-10	Низький
11-20	Нижче середнього
21-30	Середній
30-40	Вище середнього
41-48	Високий

Відповідно до середньоарифметичної суми балів та рекомендаціям визначення рівня комерційного потенціалу, рівень комерційного потенціалу розроблюваного засобу є високим. Такий результат пов’язаний з відносно низькою кількістю аналогів, більшій гнучкості у використанні а також поєднанням одразу декількох функціональних можливостей засобу.

Наступним кроком необхідно оцінити конкурентоспроможність засобу, для цього залучено трьох експертів, які брали участь в оцінці науково-технічного рівня. Результати оцінювання відбуваються за 10 бальною шкалою. Результати оцінювання відображають якісні характеристики продукту та слугують основою для подальшого аналізу його конкурентоспроможності. Для зручності подання оцінка буде окремо розділена для розроблюваного засобу PhishScan (PS) та аналогу CheckPhish (CP). Таким чином дані експертних оцінок подано у вигляді таблиці 5.3.

Таблиця 5.3 – Порівняльна характеристика засобу і його аналогу

Критерії оцінювання	Експерти					
	Творун Р.П		Ляшко П.О		Кореньков А.О	
	Бали					
	CP	PS	CP	PS	CP	PS
Зручність використання	6	8	7	8	6	10
Функціональність	7	9	8	9	6	8
Швидкість роботи	8	8	7	8	7	8

Для спрощення подальших розрахунків, необхідно отримати середні значення по кожній з критеріїв для засобів. Таким чином середнє значення оцінок експертів наведено у таблиці 5.4.

Таблиця 5.4 – Середні значення оцінок.

Критерії оцінювання	Бали	
	CheckPhish	PhishScan
Зручність використання	6.33	8.67
Функціональність	7	8.67
Швидкість роботи	7.33	8

Наступним кроком необхідно оцінити конкурентоспроможність засобів. Для цього розраховуються одиничні параметричні індекси, які дозволяють порівняти кожен критерій окремо. Якщо збільшення величини параметра свідчить про підвищення якості нової розробки, одиничний параметричний індекс розраховується за формулою 5.1

$$q_i = \frac{P_i}{P_{\text{баз } i}} \tag{5.1}$$

де

q_i – одиничний параметричний індекс, розрахований за і-м параметром;

P_i – значення і-го параметра розробки;

$P_{\text{базі}}$ – аналогічний параметр базової розробки-аналога, з якою проводиться порівняння.

Якщо зменшення величини параметра свідчить про підвищення якості нової розробки, то одиничний параметричний індекс розраховується за оберненою формулою 5.2

$$q_i = \frac{P_{\text{баз } i}}{P_i} \tag{5.2}$$

Також важливо зазначити розрахувати такий параметр, як «ціна». Для цього є відома прогнозована ціна розроблюваного засобу яка дорівнює 2300 грн та ціна аналогу рівна 4145 грн.

Розрахунок параметричних індексів для кожного критерія наведено у вигляді таблиці 5.5.

Таблиця 5.5 - Розрахунок параметричних індексів

Параметр	Одиниця виміру	Аналог	Нова розробка	q_i	Коефіцієнт вагомості
Зручність використання	Бали	6.33	8.67	1.37	0.30
Функціональність	Бали	7	8.67	1.24	0.35
Швидкість роботи	Бали	7.33	8	1.09	0.35
Ціна	грн	4145	2300	0.55	1

Також експертною оцінкою визначено коефіцієнт вагомості аі який вказує на важливість параметра при оцінці конкурентоспроможності важливо щоб виконувалась умова сума аі=1. Для параметру ціни використовується коефіцієнт економічної вагомості. Так як це єдиний економічний параметр, його вага буде дорівнювати β_i =1. Наступним етапом необхідно виконати розрахунок групових параметричних індексів за технічними параметрами на основі часткових індексів та коефіцієнтів вагомості за формулою 5.3

$$I_{\text{ТП}} = \sum_{i=1}^{n} q_i \cdot a_i \tag{5.3}$$

$$I_{\text{ТП}} = (1.37 \cdot 0.30) + (1.24 \cdot 0.35) + (1.09 \cdot 0.35) = 1.2265$$

Маючи лише один економічний параметр, його ціна не потребує розрахунку і дорівнює параметричному індексу

$$I_{\text{ЕП}} = 0.55$$

Завершальним етапом є розрахунок інтегрального показника розрахунок якого відбувається за формулою 5.4.

$$K_{IHT} = I_{H\text{П}} \cdot \frac{I_{\text{ТП}}}{I_{\text{ЕП}}} \tag{5.4}$$

Враховуючи що розроблюваний засіб відповідає всім нормам і параметрам $I_{H\text{П}}$=1.

$$K_{IHT} = 1 \cdot \frac{1.2265}{0.55} = 2.23$$

На основі інтегрального показника відомо, що K_{IHT}>1. Відповідно до рекомендації можна зробити висновок що розроблюваний метод та засіб виявлення

фішингових сайтів є дуже перспективною розробкою та є конкурентоспроможним на ринку.

5.2 Розрахунок прогнозованих витрат на реалізацію науково-дослідної роботи

Першим кроком необхідно визначити галузі у яких здійснюються витрати під час проведення науково-дослідної розробки. Статті витрат враховують будь-які аспекти що стосуються фінансової складової роботи з врахуванням затраеного часу, обладнання, бонусів, тощо. Витрати на реалізацію науково-дослідної роботи групуються за такими статтями:

- витрати на оплату праці;
- відрахування на соціальні заходи;
- матеріали;
- паливо та енергія для науково-виробничих цілей;
- витрати на службові відрядження;
- спецустаткування для наукових (експериментальних) робіт;
- програмне забезпечення для наукових (експериментальних) робіт;
- витрати на роботи, які виконують сторонні підприємства, установи організації;
- інші витрати;
- накладні (загальновиробничі) витрати.

Після визначення статтей за якими буде проведено розрахунок прогнозованих витрат на розробку, доцільно розглянути кожну статтю в окремому порядку з проведенням відповідних розрахунків.

5.2.1 Витрати на оплату праці

Витрати на оплату праці є однією з найважливіших статей у науково-дослідній роботі, адже від кваліфікації та продуктивності персоналу залежить якість та терміни виконання завдань. Розрахунок витрат на виконання науково-

дослідної роботи розраховується відповідно до посадових окладів працівників. Розрахунок окладів працівників виконується за формулою 5.5.

$$З_о = \sum_{i=1}^{k} \frac{M_{ni} \cdot t_i}{T_p} \quad (5.5)$$

де

k – кількість посад дослідників, залучених до процесу досліджень;

M_{ni} – місячний посадовий оклад конкретного дослідника, грн;

t_i – кількість днів роботи конкретного дослідника, дн.;

T_p – середня кількість робочих днів в місяці, Тр=21…23 дні.

Розрахунок витрат на заробітню платню дослідників представлено в таблиці (додаток Е)

Розрахунок витрат на заробітну плату визначався відповідно до мінімального та середнього рівня заробітку для кожної посади вказаного на вакантних пропозиціях. Наступним етапом необхідно визначити основну заробітну плату робітників. Для розрахунку основної заробітної плати робітникам використовується формула 5.6

$$З_р = \sum_{i=1}^{n} C_i \cdot t_i \quad (5.6)$$

де

C_i – погодинна тарифна ставка робітника відповідного розряду, за виконану відповідну роботу, грн/год;

t_i – час роботи робітника при виконанні визначеної роботи, год.

Погодинна тарифна ставка робітника відповідного розряду C_i розраховується відповідно до формули 5.7.

$$З_р = \frac{M_m \cdot K_i \cdot K_C}{T_p \cdot t_{зм}} \quad (5.7)$$

де

M_m– розмір прожиткового мінімуму працездатної особи або мінімальної місячної заробітної плати, грн. На момент 2024 року M_m= 8000,00 грн;

K_i – коефіцієнт міжкваліфікаційного співвідношення для встановлення тарифної ставки робітнику відповідного розряду;

K_C – мінімальний коефіцієнт співвідношень місячних тарифних ставок робітників першого розряду з нормальними умовами праці виробничих об'єднань і підприємств до законодавчо встановленого розміру мінімальної заробітної плати.

T_p – середня кількість робочих днів в місяці, приблизно Тр = 22 дні;

$t_{ЗМ}$ – тривалість зміни, год.

Для зручності сприйняття та упорядкованого подання, розрахунок велечини витрат на основну заробітну платню доцільно реалізувати у вигляді таблиці (додаток Д).

Після розрахунку виплат та заробітної плати необхідно врахувати наявність додаткової заробітної плати, яка може бути інтерпретована у вигляді різних бонусів які отримують робітники під час робочого процесу. Додаткова заробітна плата розраховується за формулою 5.8

$$З_{дод} = \left(З_о + З_р\right) \cdot \frac{Н_{дод}}{100\%} \quad (5.8)$$

де

$Н_{дод}$ – норма нарахування додаткової заробітної плати. Норма може знаходитись у діапазоні 10%-12%. Врахування відбуватиметься по середньому допустимому значеню. Таким чином $Н_{дод}$=11%

$$З_{дод} = (53955.7 + 27985.8) \cdot \frac{11\%}{100\%} = 9013.6$$

Таким чином додаткова заробітна плата буде становити 9013.6 грн. Використання цих значень дозволяє провести розрахунок відрахувань на соціальні заходи.

5.2.2 Відрахування на соціальні заходи

До статті «Відрахування на соціальні заходи» належать відрахування внеску на загальнообов'язкове державне соціальне страхування та для здійснення заходів щодо соціального захисту населення (ЄСВ – єдиний соціальний внесок).

Нарахування на заробітну плату дослідників та робітників розраховується як 22% від суми основної та додаткової заробітної плати дослідників і робітників. Розрахунок суми нарахувань відбувається з використанням формули 5.9

$$З_{дод} = (З_о + З_р + З_{дод}) \cdot \frac{Н_{зп}}{100\%} \quad (5.9)$$

де

$Н_{зп}$ – норма нарахування додаткової заробітної плати, тому $Н_{зп} = 22\%$

$$З_{дод} = (53955.7 + 27985.8 + 9013.6) \cdot \frac{22\%}{100\%} = 20010$$

Таким чином розмір відрахувань на соціальні виплати становить 20010 грн.

5.2.3 Сировина і матеріали.

Витрати на матеріали у вартісному вираженні розраховуються окремо для кожного виду матеріалів. Розрахунок відбувається за формулою 5.10.

$$M = \sum_{j-1}^{n} H_j \cdot Ц_j \cdot K_j - \sum_{j=1}^{n} B_j \cdot Ц_{вj} \quad (5.10)$$

де

H_j – норма витрат матеріалу j-го найменування, кг;

n – кількість видів матеріалів;

$Ц_j$ – вартість матеріалу j-го найменування, грн/кг;

$К_j$ – коефіцієнт транспортних витрат, (Кj = 1,1 … 1,15);

$В_j$ – маса відходів j-го найменування, кг;

$Ц_{вj}$ – вартість відходів j-го найменування, грн/кг.

Розрахунок витрат на матеріали представлено в таблиці 5.6

Таблиця 5.6 – Витрати на матеріали

Найменування матеріалу, марка, тип, сорт	Ціна за 1 кг, грн	Норма витрат, кг	Величина відходів, кг	Ціна відходів, грн/кг	Вартість витраченого матеріалу, грн
Бумага А4, 80 гр/м^2	71	2.5	0.2	14	195.25
Блокнот А5, 80 гр/м^2	320	0.54	0.06	19.2	188.9
Epson 108 EcoTank L8050/L18050 Black	7842	0.06	–	–	517.5
Ручка, Piano Maxriter, шарикова	687.5	0.048	–	–	36.3
Всього					938.95

Таким чином сума витрат на матеріали становить 938.95 грн.

5.2.4 Розрахунок витрат на комплектуючі.

Розроблюваний засіб не потребує додаткових комплектуючих. Таким чином витрати на матеріали та комплектуючі можна прирівняти до нуля, оскільки всі необхідні елементи для його функціонування вже наявні.

5.2.5 Програмне забезпечення для наукових (експериментальних) робіт.

Наступним кроком необхідно визначити ціну середовищ які задіяні при розробці засобу. Балансова вартість програмного забезпечення розраховується за формулою 5.11.

$$В_{прг} = \sum_{j=1}^{k} Ц_{іпрг} \cdot С_{прг.і} \cdot K_j \qquad (5.11)$$

де

$Ц_{іпрг}$ – ціна придбання одиниці програмного засобу цього виду, грн;

$С_{прг.і}$ – кількість одиниць програмного забезпечення відповідного найменування, які придбані для проведення досліджень, шт.;

K_j – коефіцієнт, що враховує інсталяцію, налагодження програмного засобу тощо, (Кі = 1,10…1,12);

k – кількість найменувань програмних засобів.

Для розробки необхідне лише одне середовище розробки програм - PyCharm від компанії JetBrains. Розрахунок витрати на придбання програмних засобів наведено в таблиці 5.7

Таблиця 5.7 – Розрахунок витрат на придбання програмних засобів

Найменування програмного засобу	Кількість, шт	Ціна за одиницю, грн	Вартість, грн
PyCharm Professional	1	4140	4140
Всього			4140

Мала кількість одиниць програмних засобів обумовлена використанням безкоштовних версій для інших необхідних засобів.

5.2.6 Амортизація обладнання, яке використовувалось для проведення розробки.

Амортизація обладнання, що використовувалась для розробки розраховується за формулою 5.12

$$A = \frac{Ц}{T_в} \cdot \frac{t_{вик}}{12} \quad (5.12)$$

де

Ц – балансова вартість обладнання, грн.;

$T_в$ – термін корисного використання обладнання згідно податкового законодавства, років

$t_{вик}$ – термін використання під час розробки, місяців

Розрахунки амортизації обладнання яку використовувалось під час розробки засобу, представлені в таблиці 5.8.

Таблиця 5.8 – Розрахунки амортизації обладнання

Найменування обладнання	Балансова вартість, грн.	Строк корисного використання, років	Термін використання обладнання, місяців	Амортизаційні відрахування, грн
Приміщення	330000	20	1	1375
Меблі (офісне обладнання)	22000	4	1	458.33
Комп’ютер (включно з периферією)	31000	2	1	1,291.67
Всього				3125

Таким чином сума амортизаційних відрахувань складає 3125грн.

5.2.7 Витрати на електроенергію.

Тарифи на електроенергію для підприємств відрізняються від тарифів, призначених для населення. Вартість розподілу електроенергії залежить від енергорозподільної компанії, яка надає послуги, і може змінюватися в залежності від постачальника. Також тарифна ставка визначається класом напруги споживача (1-й чи 2-й клас). Регулювання тарифів на розподіл здійснює Національна комісія з регулювання енергетики та комунальних послуг (НКРЕКП). Витрати на електроенергію розраховуються з використанням формули 5.13.

$$В_е = В \cdot П \cdot Ф \cdot К_п \quad (5.13)$$

де

В – вартість 1 кВт-години електроенергії для 1 класу підприємства, В = 11 грн./кВт;

П – встановлена потужність обладнання, кВт. П = 0,4 кВт;

Ф – фактична кількість годин роботи обладнання, годин.

$К_п$ – коефіцієнт використання потужності, Кп = 0,9.

Таким чином відповідно до формули витрати на електроенергію будуть становити:

$$В_е = 11 \cdot 0.4 \cdot 8 \cdot 30 \cdot 0.9 = 950 \text{ (грн)}$$

5.2.8 Службові відрядження

Даний тип виконуваної роботи не передбачає наявності службових відряджень.

5.2.9 Витрати на роботи, які виконують сторонні підприємства, установи і організації

Підприємство забезпечує усі етапи виконання роботи та не потребує залучення інших підприємств тому розрахунок витрат на роботи сторонніх підприємств непотрібен

5.2.10 Інші витрати та загальновиробничі витрати.

До статті «Інші витрати» включаються витрати, які не були враховані в інших статтях витрат, але можуть бути безпосередньо віднесені до собівартості досліджень на основі прямих ознак. Розрахунок витрат за цією статтею здійснюється у розмірі 50–100% від суми основної заробітної плати дослідників наведено у формулі 5.14

$$I_{\text{в}} = (З_{\text{о}} + З_{\text{р}}) \cdot \frac{Н_{\text{ів}}}{100\%} \quad (5.14)$$

де

$Н_{\text{ів}}$ – норма нарахування за статтею «Інші витрати». Таким чином $Н_{\text{ів}}$=50%

$$I_{\text{в}} = (53955.7 + 27985.8) \cdot \frac{50}{100} = 40970.75$$

До статті «Накладні (загальновиробничі) витрати» відносяться такі види витрат: витрати на управління організацією, витрати на винахідницьку та раціоналізаторську діяльність, витрати на підготовку, перепідготовку та навчання персоналу, витрати, пов’язані з набором працівників, оплата банківських послуг, витрати на освоєння виробництва продукції, а також витрати на науково-технічну інформацію, рекламу та інші подібні витрати. Розрахунок витрат за цією статтею здійснюється у розмірі 100–150% від суми основної заробітної плати дослідників та розраховується за формулою 5.15

$$Н_{\text{нзв}} = (З_{\text{о}} + З_{\text{р}}) \cdot \frac{Н_{\text{нзв}}}{100\%} \quad (5.15)$$

де

$Н_{\text{нзв}}$ – норма нарахування за статтею «Накладні (загальновиробничі) витрати» $Н_{\text{нзв}}$=100%.

$$Н_{\text{нзв}} = (53955.7 + 27985.8) \cdot \frac{100}{100} = 81941.5$$

Для отримання суми витрат на проведення науководослідної роботи необхідно врахувати всі попередні статті витрат.

$$В_{заг} = 53955.7 + 27985.8 + 9013.6 + 20010 + 938.95 + 4140 + 3125 + 950 + 40970.75 + 81941.5 = 243031.3$$

5.2.12 Розрахунок загальних витрат на науково-дослідну (науковотехнічну) роботу та оформлення її результатів.

Загальні витрати на завершення науково-дослідної (науково-технічної) роботи та оформлення її результатів розраховуються за формулою 5.16

$$ЗВ = \frac{В_{заг}}{\eta} \quad (5.16)$$

де

η – коефіцієнт, який характеризує етап (стадію) виконання науководослідної роботи. Так як розробка, на даний момент, знаходиться на стадії дослідного зразка то η=0.5.

$$ЗВ = \frac{243031.3}{0.5} = 486062.6$$

5.3 Розрахунок економічної ефективності науково-технічної розробки за її можливої комерціалізації потенційним інвестором

У ринкових умовах основним позитивним результатом, який може отримати потенційний інвестор від впровадження тієї чи іншої науково-технічної розробки, є збільшення чистого прибутку. Це зростання прибутку дозволить інвестору отримати додаткові кошти, поліпшити фінансові результати його діяльності, підвищити конкурентоспроможність і, в кінцевому підсумку, може сприяти ухваленню рішення щодо комерціалізації розробки.

Для розрахунку можливого зростання чистого прибутку у потенційного інвестора від можливого впровадження науково-технічної розробки необхідно врахувати низку факторів:

1) Вказати, з якого моменту можуть бути впроваджені результати науково-технічної розробки;
2) Визначити, через скільки років після впровадження розробки очікуються основні позитивні результати для потенційного інвестора;
3) Кількісно оцінити поточний і майбутній попит на цю або подібні науково-технічні розробки, а також визначити основних учасників (категорію населення) цього попиту;
4) Оцінити ціну реалізації на ринку для науково-технічних розробок з аналогічними чи схожими функціями.

При розрахунку економічної ефективності обов'язково слід враховувати зміну вартості грошей у часі, оскільки між інвестуванням та отриманням прибутку проходить значний час. Оцінка ефективності інноваційних проектів передбачає розрахунок таких ключових показників:

- абсолютного економічного ефекту (чистого дисконтованого доходу);
- внутрішньої економічної дохідності (внутрішньої норми дохідності);
- терміну окупності (дисконтованого терміну окупності).

Аналізуючи напрямки науково-технічних розробок, розрахунок економічної ефективності для можливості комерціалізації таких розробок потенційним інвестором можна узагальнити, враховуючи визначені умови та ситуації.

5.3.1 Розробка програмного засобу для використання масовим споживачем.

Маючи орієнтовну вартість продукту який складає 1750грн за копію, таермін збільшення прибутку від продажу складе 3 роки. Вдосконалення розробки дозволить змінити ціну продукту піднявши його вартість на 600 грн. Таким чином кількість реалізованої продукції у перший рік збільшиться на 6000 копій на другий рік прогнорзується більшенням на 5000 копій, а третій на 4000 копій відповідно. Важливо зазначити, що до моменту впровадження результатів наукової розробки реалізації продукту не було для розрахунку майбутнього економічного ефекту слід використовувати формулу 5.17

$$\Delta П_i = (\pm \Delta Ц_o \cdot N + Ц_0 \cdot \Delta N)_i \cdot \lambda \cdot \rho \cdot (1 - \frac{\vartheta}{100}) \quad (5.17)$$

де

$\pm \Delta Ц_o$ – зміна вартості програмного продукту (зростання чи зниження) від впровадження результатів науково-технічної розробки в аналізовані періоди часу;

N – кількість споживачів які використовували аналогічний продукт у році до впровадження результатів нової науково-технічної розробки;

$Ц_o$– основний оціночний показник, який визначає діяльність підприємства у даному році після впровадження результатів наукової розробки.

$Ц_б$ – вартість програмного продукту у році до впровадження результатів розробки;

ΔN – збільшення кількості споживачів продукту, в аналізовані періоди часу, від покращення його певних характеристик;

λ – коефіцієнт, який враховує сплату податку на додану вартість. Ставка податку на додану вартість дорівнює 20%, а коефіцієнт λ = 0,8333.

р – коефіцієнт, який враховує рентабельність продукту;

ϑ – ставка податку на прибуток, у 2024 році ϑ =18%.

$$\Delta П_1 = (0 \cdot 600 + 2350 \cdot 6000) \cdot 0.8333 \cdot 0.25 \cdot (1 - 0.18) = 2407452.45$$

$$\Delta П_2 = (0 \cdot 600 + 2350 \cdot 11000) \cdot 0.8333 \cdot 0.25 \cdot (1 - 0.18) = 4413592.28$$

$$\Delta П_3 = (0 \cdot 600 + 2350 \cdot 15000) \cdot 0.8333 \cdot 0.25 \cdot (1 - 0.18) = 6021817.43$$

$$\Delta П_{заг} = 2407452.45 + 4413592.28 + 6021817.43 = 12842862.16$$

Таким чином комерційний ефект від реалізації результатів розробки засобу виявлення фішингових сайтів протягом трьох років складає 12842862грн.

5.3.2 Розрахунок ефективності вкладених інвестицій та періоду їх окупності.

Наступним етапом необхідно здійснити розрахунок приведеної вартості загального чистого прибутку, який потенційний інвестор може отримати від

впровадження та комерціалізації науково-технічної розробки. Розрахунок вібувається за формулою 5.18.

$$ПП = \sum_1^T \frac{\Delta П_i}{(1+\tau)^t} \quad (5.18)$$

де

$\Delta П_i$ – збільшення чистого прибутку у кожному із років, протягом яких виявляються результати виконаної та впровадженої науково-дослідної (науково-технічної) роботи, грн;

Т – період часу, протягом якою виявляються результати впровадженої науково-дослідної (науково-технічної) роботи, роки;

τ – ставка дисконтування, за яку можна взяти щорічний прогнозований рівень інфляції в країні, τ = 0,05…0,15;

t – період часу (в роках).

ПП= (2407452.45/(1+0,1)[1])+(4413592.28/(1+0,1)[2])+(6021817.43/(1+0,1)[3])= 2188593,13+3648835,92+4529514,15 = 10306943,20 грн

Далі обчислюється розмір початкових інвестицій PV, які потенційний інвестор повинен вкласти для реалізації та комерціалізації науково-технічної розробки. Розрахунок відбувається за формулою 5.19.

$$PV = k_{інв} * ЗВ \quad (5.19)$$

де

$k_{інв}$ – коефіцієнт, що враховує витрати інвестора на реалізацію науково-технічної розробки та її комерціалізацію. Це можуть бути витрати на підготовку інфраструктури, розробку технологій, навчання персоналу, маркетинг та інші заходи. Зазвичай kінв варіюється від 2 до 5.

ЗВ – загальні витрати на проведення науково-технічної розробки та оформлення її результатів, грн

PV=2*486062.6 = 972125.2

Розрахунок абсолютного економічного ефекту $Е_{абс}$ для потенційного інвестора від впровадження та комерціалізації розраховується як різниця

приведеної вартості загального чистого прибутку та розміру початкових інвестицій. Розрахунок відбувається за формулою 5.20

$$\text{Е}_{\text{абс}} = \text{ПП} - PV \quad (5.20)$$

$$\text{Е}_{\text{абс}} = 10306943{,}20 - 972125.2 = 9334818 \text{ грн.}$$

Таким чином $\text{Е}_{\text{абс}} > 0$ що свідчить про доцільність вкладання коштів на виконання даної науково-дослідної (науково-технічної) роботи.

Для прийняття остаточного рішення необхідно розрахувати внутрішню економічну дохідність (IRR) вкладених інвестицій та порівняти її з бар'єрною ставкою дисконтування. Розрахунок відбувається за формулою 5.21

Таким чином $\text{Е}_{\text{абс}} > 0$ що свідчить про доцільність вкладання коштів на виконання даної науково-дослідної (науково-технічної) роботи.

Для прийняття остаточного рішення необхідно розрахувати внутрішню економічну дохідність (IRR) вкладених інвестицій та порівняти її з бар'єрною ставкою дисконтування. Розрахунок відбувається за формулою 5.21

$$\text{Е}_{\text{в}} = \sqrt[\text{Т}_{\text{ж}}]{1 + \frac{\text{Е}_{\text{абс}}}{PV}} - 1 \quad (5.21)$$

Якщо внутрішня норма дохідності виявиться нижчою за цю ставку, інвестиції в науково-технічну розробку будуть абсолютно економічно неефективними.

$$\text{Е}_{\text{в}} = \sqrt[3]{1 + 9334818/972125.2} - 1 = 1.19$$

Визначення мінімальної ставку дисконтування, виконується за формулою 5.22

$$\tau_{min} = d + f \quad (5.22)$$

де

d – середньозважена ставка за депозитними операціями в комерційних банках. Таким чином d = (0,09…0,14);

f – показник, що характеризує ризикованість вкладень. Зазвичай, величина f коливається від 0,05 до 0,5.

$$\tau_{min} = 0{,}14 + 0{,}05 = 0.19$$

Результат $E_в > \tau_{min}$ вказує на те, що інвестор може бути зацікавлений у фінансуванні даного засобу. Завершальним етапом є розрахунок термінів окупності. Даний розрахунок відбувається за формулою 5.23

$$T_{ок} = \frac{1}{E_в} \quad (5.23)$$

$$T_{ок} = 1/1.19 = 0.84$$

Таким чином $T_{ок}$ = 0.84 що рівне близько 10 місяців враховуючи, що $T_{ок}$ < 3 років можна зробити висновок, що фінансування розробки засобу для виявлення фішингових сайтів є доцільним.

5.4 Висновки з розділу

Проведення комерційного та технологічного аудиту науково-технічної розробки методу і засобу виявлення фішингових атак визначило, що рівень комерційного потенціалу розроблюваного засобу є високим. На основі цього була підтверджена доцільність проведення розрахунку прогнозованих витрат на реалізацію науково-дослідної роботи, за яким були визначені основні статті витрат які необхідні для реалізації засобу. Результатом розрахунків прогнозованих витрат визначено що загальні витрати становитимуть близько 243031 грн.

На основі отриманих результатів розрахунків проведено розрахунок економічної ефективності результат якої вказує на те, що внутрішню економічну дохідність більша за мінімальну ставку дисконтування. З цього можна зробити висновок, що в таких умовах інвестор може бути зацікавлений у фінансуванні даного проекту. Період окупності даної розробки становить 0.84 року, що дорівнює приблизно 10 місяцям. На основі отриманих результатів можна стверджувати, що розроблений програмний продукт є більш доступним за ціною порівняно з аналогами. Це свідчить про те, що фінансування розробки засобу для виявлення фішингових сайтів є доцільним.

ВИСНОВКИ

Зростання кількості атак з використанням фішингових веб сайтів в період з 2022 по 2024 роки обумовило актуальність досліджень в галузі кібербезпеки, що спрямовані на протидію цього виду соціальної інженерії. Аналіз впливу загроз фішингових веб-сайтів дозволив визначити їх ознаки. При цьому аналіз відомих методів захисту від фішингу показав їх недостатню ефективність на тлі зміни вектору цієї атаки. З урахуванням цього було поставлено задачу дослідження направлену на врахування обставин створення сайту як ознак потенційного фішингу.

На основі статистичних досліджень фішингових атак, доступних у відкритих джерелах було визначено низку технічних показників, які опосередковано можуть свідчити про фішинг. Для визначення вагомості кожної з ознак проведено опитування експертів з кібербезпеки завдяки якому було визначено вагові коефіцієнти кожної з ознак. Для формалізації було виконано математичний опис процесу аналізу веб-сайтів, який дозволив визначити основні вхідні та вихідні параметри цього процесу, а також використовуванні перетворення. Спираючись на математичний опис було розроблено метод виявлення фішингових веб сайтів, який на відміну від відомих окрім технічних параметрів сайту, враховує обставини його створення.

Для реалізації запропонованого методу, було розроблено архітектуру програмного засобу на основі клієнт-серверної архітектури. Особливістю засобу є його можливість використання, як незалежного веб-сервіса або, як розширення для браузера яке може бути встановлене користувачем.

Розроблений узагальнений алгоритм дозволив показати роботу запропонованої архітектури. В подальшому були наведені окремі процедури роботи цього алгоритму, які поєднували використання зовнішніх сервісів, а внутрішнього аналізу властивостей досліджуваного сайту. Важливим аспектом є визначення ймовірності фішингу на сайті не тільки на основі технічних аспектів

контенту сайту а, й врахування умов його створення, таких як рання реєстрація сайту, попереднє запам’ятовування сусідніх доменів розташованих на одному акаунті CloudFlare.

Для реалізації засобу було проведено аналіз та обґрунтування засобів розробки. Після розробки засобу здійснено блокове тестування в результаті показало коректну роботу реалізованої черги реалізованої за допомогою Redis, здатної постійно моніторити наявність запитів до неї, а також виконувати послідовне сканування отриманих запитів, тестування окремих функцій показало коректність запису, обробки та представлення інформації яка використовується засобом. Інтеграційне тестування підтвердило коректність зв'язку клієнта з сервером, та правильну передачу даних між ними. На основі мануального тестування виявлено та враховано можливі сценарії поведінки користувача під час використання засобу. Під час проведення експериментального тестування , було визначено необхідність введення коефіцієнтів для корегування відповідей експертів, для досягнення більшої точності аналізу.

Проведення комерційного та технологічного аудиту науково-технічної розробки методу і засобу виявлення фішингових атак визначило, що рівень комерційного потенціалу розроблюваного засобу є високим та має підстави для фінансування що робить його економічно вигідним для потенційних інвесторів.

Розроблені метод та засіб доцільно використовувати, як звичайним користувачам в побуті під час веб-серфінку так і комерційних підприємствах, для захисту працівників. Удосконалений метод дозволяє досягти мети дослідження, забезпечуючи користувачу швидкий та точний доступ до інформації про підозрілі адреси. Метод, є гнучким та дозволяє швидко адаптувати засіб під зміни тенденцій в особливостях фішингових веб-сайтів, шляхом додавання більшої кількості нових ознак, врахуванням коефіцієнтів вагомості та корегування.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Phishing and Countermeasures M. Jakobsson, S. Myers. Hoboken, NJ, USA : John Wiley & Sons, Inc., 2006. URL: https://doi.org/10.1002/0470086106 (accessed: 15.09.2024).
2. Шахрайські та фішингові сайти. Аналіз, тренди та рекомендації для клієнтів, 2022/2023 – Асоціація ЄМА. URL: https://www.ema.com.ua/citizens/cyber-safety-school/shahrajski-ta-fishingovi-sajti-analiz-trendi-ta-rekomendacii-dlja-kliientiv-2022-2023/ (дата звернення: 16.09.2024).
3. Фішинг: як розпізнати підроблений сайт. URL: https://minfin.com.ua/ua/2021/01/14/58552994/. (дата звернення: 16.09.2024).
4. Насталенко Я. І., Савчук І. Б., Радченко Є. В., Залюбівська О. Б. Використання штучного інтелекту в освітньому процесі у світлі академічної доброчесності. Матеріали LIII НТКП ВНТУ-2024 Вінниця: ВНТУ 2024. С. 2694–2697. URL: https://press.vntu.edu.ua/index.php- /vntu/catalog/view/832/1453/2726-1
5. Майданевич Л. О., Насталенко Я. І., Савчук І. Б. Особливості технічного захисту авторських прав за допомогою використання стеганографії (на прикладі комп’ютерних ігор). Світ наукових досліджень : матеріали інтернет-конф. URL: https://www.economy-confer.com.ua/full-article/5577/
6. Савчук І. Б. Модель рольового розмежування прав доступу для бази даних графічної новели. м. Вінниця / Наук. керівник Ю. В. Баришев. Вінниця, ВНТУ 2023.
7. APWG | About the APWG. APWG | Unifying The Global Response To Cybercrime. URL: https://apwg.org/about-us/ (дата звернення: 18.09.2024)
8. APWG | Phishing Attack Trends Report – 3Q 2022. APWG URL: https://docs.apwg.org/reports/apwg_trends_report_q3_2022.pdf (date of access: 18.09.2024)

9. APWG | Phishing Attack Trends Report – 1Q 2023. APWG URL: https://docs.apwg.org/reports/apwg_trends_report_q1_2023.pdf (date of access: 18.09.2024)
10. Ilca F., Balan T. Phishing as a Service Campaign using IDN Homograph Attack. 2021 International Aegean Conference on Electrical Machines and Power Electronics (ACEMP) & 2021 International Conference on Optimization of Electrical and Electronic Equipment (OPTIM), м. Brasov, Romania, 2–3 верес. 2021 р. 2021. URL: https://doi.org/10.1109/optim-acemp50812.2021.9590028 (date of access: 21.09.2024)
11. Resecurity | EvilProxy Phishing-as-a-Service with MFA Bypass Emerged in Dark Web.URL: https://www.resecurity.com/blog/article/evilproxy-phishing-as-a-service-with-mfa-bypass-emerged-in-dark-web (date of access: 24.09.2024)
12. EvilProxy Phishing Used for Cloud Account Takeover Campaign URL: https://www.proofpoint.com/us/blog/email-and-cloud-threats/cloud-account-takeover-campaign-leveraging-evilproxy-targets-top-level (date of access: 22.10.2024).
13. Cloudflare. URL: https://www.cloudflare.com/about-overview/ (date of access: 02.11.2024).
14. Google Safe Browsing. Safe Browsing – Google Safe Browsing . URL: https://safebrowsing.google.com/ (дата звернення: 03.11.2024).
15. PhishTank > Frequently Asked Questions (FAQ). PhishTank | Join the fight against phishing . URL: https://phishtank.org/faq.php#whatisphishtank (date of access: 03.11.2024).
16. IBM Trusteer Rapport | Malware and Phishing Attacks - IBM. IBM - United States. URL: https://www.ibm.com/products/trusteer-rapport (date of access: 03.11.2024).
17. Safeweb. Safeweb . URL: https://safeweb.norton.com/help/about (date of access: 08.11.2024).

18. Darktrace | Cyber security that learns you. URL: https://darktrace.com/ (дата звернення: 03.11.2024).
19. About Splunk What is Splunk? URL: https://www.splunk.com/en_us/about-splunk.html (date of access: 03.11.2024).
20. APWG | Phishing ended 2023 with a bang. *APWG | Unifying The Global Response To Cybercrime*. URL: https://apwg.org/phishing-ended-2023-with-a-bang/ (date of access: 30.11.2024).
21. Dixit S., Ramanukolanu J. Phishing Via Typosquatting and Brand Impersonation: Trends and Tactics. Cybersecurity and Zero Trust Leader | Zscaler. URL: https://www.zscaler.com/blogs/security-research/phishing-typosquatting-and-brand-impersonation-trends-and-tactics?utm_source=chatgpt.com (date of access: 06.11.2024).
22. Trends in phishing attacks on organizations in 2022–2023. *Positive Technologies*. URL: https://global.ptsecurity.com/analytics/trends-in-phishing-attacks-on-organizations-in-2022-2023 (date of access: 06.11.2024).
23. Phishing JavaScript Obfuscation Techniques Soars. *akamai.com*. URL: https://www.akamai.com/blog/security/phishing-javascript-obfuscation-techniques-soars (date of access: 07.11.2024)
24. Dawes J. Do Data Characteristics Change According to the Number of Scale Points Used? International Journal of Market Research. URL: https://doi.org/10.1177/147078530805000106 (date of access: 18.11.2024).
25. Redis - The Real-time Data Platform. *Redis*. URL: https://redis.io/ (date of access: 22.11.2024).
26. Microsoft. Visual Studio Code - Code Editing. Redefined. URL: https://code.visualstudio.com/ (accessed: 17.11.2024).
27. JetBrains. PyCharm: the Python IDE for data science and web development. *JetBrains*. URL: https://www.jetbrains.com/pycharm/ (date of access: 15.12.2024).

ДОДАТКИ

ПРОТОКОЛ ПЕРЕВІРКИ
НАВЧАЛЬНОЇ (КВАЛІФІКАЦІЙНОЇ) РОБОТИ

Назва роботи: Метод та засіб виявлення фішингових веб-сайтів

Автор роботи: Савчук Іван Борисович

Тип роботи: магістерська кваліфікаційна робота

Підрозділ кафедра захисту інформації ФІТКІ

Керівник Баришев Юрій Володимирович, к.т.н., доцент

Показники звіту подібності

Turnitin	
Оригінальність	97 %
Загальна схожість	3 %

Аналіз звіту подібності (відмітити потрібне):

√ 1. Запозичення, виявлені у роботі, оформлені коректно і не містять ознак плагіату.

☐ 2. Виявлені у роботі запозичення не мають ознак плагіату, але їх надмірна кількість викликає сумніви щодо цінності роботи і відсутності самостійності її автора. Роботу направити на доопрацювання.

☐ 3. Виявлені у роботі запозичення є недобросовісними і мають ознаки плагіату та/або в ній містяться навмисні спотворення тексту, що вказують на спроби приховування недобросовісних запозичень.

Заявляю, що ознайомлений з повним звітом подібності, який був згенерований Системою щодо роботи.

Автор роботи _____ (підпис) Іван САВЧУК (прізвище, ініціали)

Особа, відповідальна за перевірку _____ (підпис) Валентина КАПЛУН (прізвище, ініціали)

Керівник роботи _____ (підпис) Юрій БАРИШЕВ (прізвище, ініціали)

Додаток Б. ТЕХНІЧНЕ ЗАВДАННЯ

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

УЗГОДЖЕНО	ЗАТВЕРДЖУЮ
директор ТОВ "Укр.кібер.розробки"	
посада, підприємство	д. т. н., проф.
______ Ященко П.О	______ Володимир ЛУЖЕЦЬКИЙ
підпис І. І. П.	
"27" вересня 2024 р.	"27" вересня 2024 р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання магістерської кваліфікаційної роботи

на тему: «МЕТОД ТА ЗАСІБ ВИЯВЛЕННЯ ФІШИНГОВИХ ВЕБ-САЙТІВ»
08-53.МКР.01500.000 ТЗ

Керівник магістерської кваліфікаційної роботи

доц. каф. ЗІ, к. т. н.

______ **Юрій БАРИШЕВ**

Вінниця 2024

1 Підстави для проведення робіт

Робота проводиться на підставі наказу ректора ВНТУ від 9 вересня 2024 року № 207.

2 Мета та призначення МКР

Мета – покращення захисту користувачів від фішингових веб-сайтів у мережі Інтернет.

Об'єктом дослідження є процес захисту від методів соціальної інженерії.

Предметом дослідження є методи та засоби виявлення фішингових веб-сайтів.

Робота призначена для захисту структурованих даних, що часто оновлюються.

3 Вихідні дані для проведення МКР

МКР проводиться вперше і вхідними даними для проведення МКР є:

3.1 APWG | About the APWG. APWG | Unifying The Global Response To Cybercrime. URL: https://apwg.org/about-us/.

3.2 EvilProxy Phishing Used for Cloud Account Takeover Campaign URL: https://www.proofpoint.com/us/blog/email-and-cloud-threats/cloud-account-takeover-campaign-leveraging-evilproxy-targets-top-level.

3.3 PhishTank – Frequently Asked Questions (FAQ). PhishTank | Join the fight against phishing . URL: https://phishtank.org/faq.php#whatisphishtank.

3.4 Trends in phishing attacks on organizations in 2022–2023. *Positive Technologies*. URL: https://global.ptsecurity.com/analytics/trends-in-phishing-attacks-on-organizations-in-2022-2023.

4 Виконавці МКР

Студент групи 1БС-23м Савчук Іван Борисович

5 Вимоги до виконання МКР

Для досягнення мети необхідно виконати такі завдання:

- проаналізувати загрози, що виникають через фішингові сайти;
- виконати аналіз інформаційних джерел з метою дослідження наявних методів і стратегій фішингових атак та класифікації їх різновидів;
- ідентифікувати основні риси, притаманні фішинговим сайтам;
- розробити архітектуру системи захисту від фішингу;
- розробити алгоритм для автоматичного аналізу підозрілих сайтів;
- реалізувати розроблені алгоритми як засіб;
- експериментально оцінити показники ефективності засобу;
- визначити економічну доцільність розробки.

6 Вимоги до супровідної документації

Графічна і текстова документація повинна відповідати стандартам: ДСТУ 3008:2015, ДСТУ 8302:2015.

7 Етапи МКР

Робота з теми виконується за такими етапами:

Зміст етапу	Початок - закінчення		Очікувані результати	Звітна документація
Аналіз завдання. Вступ	2.09.2024	09.09.2024	Вступ	Чернетка вступу
Аналіз інформаційних джерел за напрямком магістерської кваліфікаційної роботи	10.09.2024	16.09.2024	Перший розділ	Чернетка першого розділу
Науково-технічне обґрунтування	17.09.2024	23.09.2024	Аналіз існуючих режимів блокового шифрування та методів автентифікованго шифрування	Чернетка першого розділу
Розробка технічного завдання	24.09.2-24	12.10.2024	Технічне завдання	Технічне завдання
Аналіз методів виявлення фішингу	12.10.2024	14.10.2024	Аналіз існуючих методик тестування випадкових послідовностей	Чернетка першого розділу
Математичний опис методів виявлення фішингу	15.10.2024	19.10.2024	Другий розділ	Чернетка другого розділу
Розробка методу виявлення фішингу	21.10.2024	25.10.2024	Другий розділ	Чернетка другого розділу
Реалізація методу виявлення фішингу	28.10.2024	29.11.2024	Третій розділ	Чернетка третього розділу
Тестування ефективності методу виявлення фішингу	02.11.2024	08.11.2024	Четвертий розділ	Чернетка четвертого розділу
Розробка розділу економічного обґрунтування доцільності розробки	09.11.2024	14.11.2024	Економічний розділ	Чернетка економічного розділу
Аналіз виконання ТЗ, висновки	15.11.2024	19.11.2024	Готові висновки	Чернетка висновків
Оформлення пояснювальної записки	20.11.2024	26.11.2024	Пояснювальна записка	Пояснювальна записка

8 Очікувані результати та порядок реалізації МКР

Передбачається розробка нових (удосконалення існуючих) методів які спрямовані на покращення ефективності виявлення фішингових веб-сайтів.

Запланована реалізація прототипу для дослідження показників якості розробки, а також потенційного використання для розв'язання виробничих задач.

9 Матеріали які подаються після закінчення МКР

По завершенню роботи подається пояснювальна записка та ілюстративна частина.

10 Порядок приймання МКР та її етапів

Апробація на науково-технічних конференціях та семінарах. Результати роботи будуть розглядатися на засіданні ДЕК із захисту магістерських кваліфікаційних робіт.

11 Вимоги щодо технічного захисту інформації з обмеженим доступом

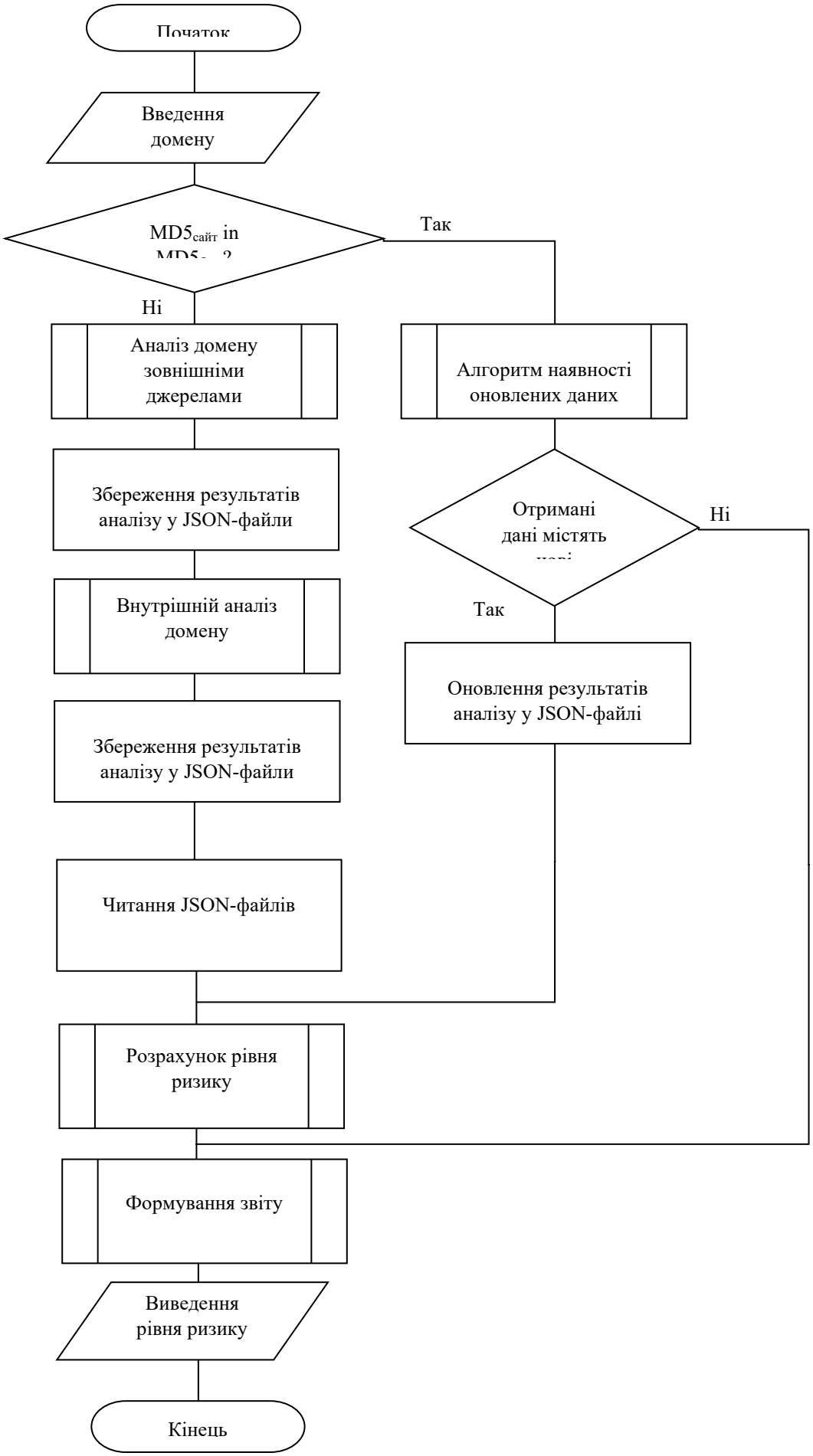
У зв'язку з тим, що дана робота не містить інформації, що потребує захисту у відповідності до законів України, заходи з її технічного захисту не передбачаються.

Розробив студент групи 1БС-23м _____ Іван САВЧУК

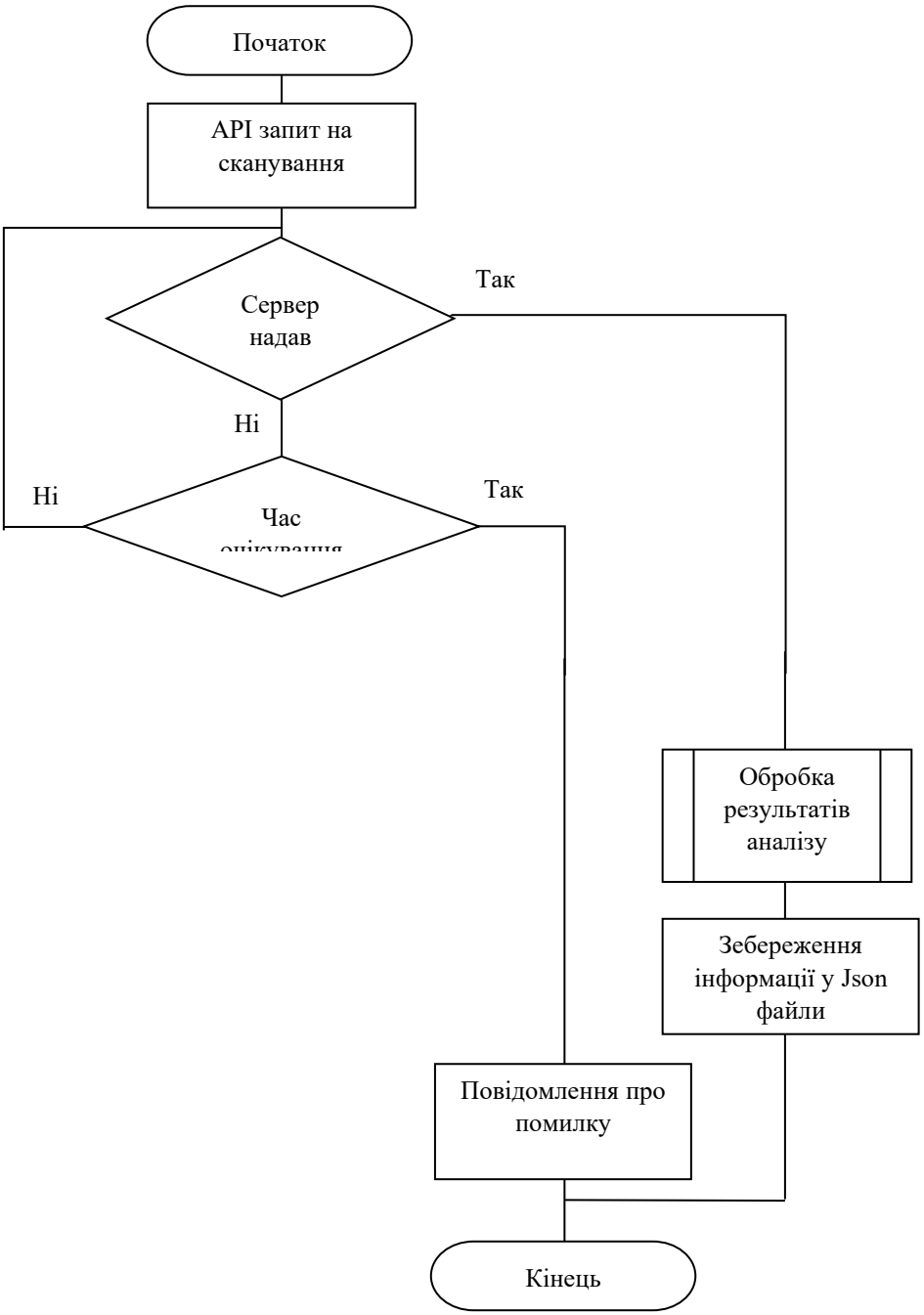
Додаток В. РЕЗУЛЬТАТ ОПИТУВАННЯ ЕКСПЕРТІВ

Ознака	Експерти				
	Баришев Ю.В	Куперштейн Л.М	Войтович О. П	Дронь А.А	Кореньков А.О
Редірект з одного посилання на інше	3	4	3	4	4
Використання брендових ключових слів у піддоменах	4	5	5	5	5
Відкриття домену з різних геолокацій	5	3	2	5	4
Свіжа дата реєстрації домену	4	3	4	4	3
Попередження від сервісів сканування	4	5	4	4	4
Наявність MD5-хешів файлів у базі даних	5	5	5	5	4
Обфускація контенту сайту	2	2	4	2	2
Обфускація скриптів на сайті	1	2	3	2	2
Сусідні домени на одному акаунті CloudFlare	3	3	4	3	3

Додаток Г. УЗАГАЛЬНЕНИЙ АЛГОРИТМ РОБОТИ ЗАСОБУ



Додаток Д. АЛГОРИТМ РОБОТИ ПРОЦЕДУРИ АНАЛІЗУ ДОМЕНУ ЗА ДОПОМОГОЮ VIRUSTOTAL



Додаток Е. КРИТЕРІЇ ОЦІНЮВАННЯ НАУКОВО-ТЕХНІЧНОГО РІВНЯ І КОМЕРЦІЙНОГО ПОТЕНЦІАЛУ РОЗРОБКИ ТА БАЛЬНА ОЦІНКА

Бали (за п’ятибальною шкалою)					
	0	1	2	3	4
1	2	3	4	5	6
Технічна здійсненність концепції					
1	Достовірність концепції не підтверджена	Концепція підтверджена експертними висновками	Концепція підтверджена розрахунками	Концепція перевірена на практиці	Перевірено роботоздат–ність продукту в реальних умовах
2	Багато аналогів на малому ринку	Мало аналогів на малому ринку	Кілька аналогів на великому ринку	Один аналог на великому ринку	Продукт не має аналогів на великому ринку
3	Ціна продукту значно вища за ціни аналогів	Ціна продукту дещо нижча за ціни аналогів	Ціна продукту приблизно дорівнює цінам аналогів	Ціна продукту дещо нижча за ціни аналогів	Ціна продукту значно нижча за ціни аналогів
4	Технічні та споживчі властивості продукту значно гірші ніж в аналогів	Технічні та споживчі властивості і продукти трохи гірші, ніж в аналогів	Технічні та споживчі властивості продукту на рівні аналогів	Технічні та споживчі властивості продукту трохи кращі ніж в аналогів	Технічні та споживчі властивості продукту значно кращі, ніж в аналогів
5	Експлуатаційні витрати значно вищі, ніж в аналогів	Експлуатаційні витрати дещо вищі, ніж в аналогів	Експлуатаційні витрати на рівні експлуатацій-них витрат аналогів	Експлуатаційні витрати трохи нижчі, ніж в аналогів	Експлуатаційні витрати значно нижчі, ніж в аналогів
Ринкові перспективи					
6	Ринок малий і не має позитивної динаміки	Ринок малий але має позитивну динаміку	Середній ринок з позитивною динамікою	Великий стабільний ринок	Великий ринок з позитивною динамікою
7	Активна конкуренція великих компаній на ринку	Активна конкуренція	Помірна конкуренція	Незначна конкуренція	Конкурентів немає
Практична здійсненність					
8	Відсутні фахівці як з технічної, так і з комерційної реалізації ідеї	Необхідно наймати фахівців або витрачати значні кошти	Необхідне незначне навчання фахівців та збільшення їх штату	Необхідне незначне навчання фахівців	Є фахівці з питань як з технічної, так і з комерційної реалізації ідеї

		та час на навчання наявних фахівців			
9	Потрібні значні фінансові ресурси, які відсутні. Джерела фінансування ідеї відсутні	Потрібні незначні фінансові ресурси. Джерела фінансування відсутні	Потрібні значні фінансові ресурси. Джерела фінансування є	Потрібні незначні фінансові ресурси. Джерела фінансування є	Не потребує додаткового фінансування
10	Необхідна розробка нових матеріалів	Потрібні матеріали, що використову-ються у військово-промисловому комплексі	Потрібні дорогі матеріали	Потрібні досяжні та дешеві матеріали	Всі матеріали для реалізації ідеї відомі та давно використов-уються у виробництві
11	Термін реалізації ідеї більший за 10 років	Термін реалізації ідеї більший за 5 років. Термін окупності інвестицій більше 10-ти років	Термін реалізації ідеї від 3-х до 5-ти років. Термін окупності інвестицій більший 5-ти років	Термін реалізації ідеї менший 3-х років. Термін окупності інвестицій від 3-х до 5-ти років	Термін реалізації ідеї менший 3-х років. Термін окупності інвестицій менший 3-х років
12	Необхідна розробка регламентних документів та отримання великої кількості дозвільних документів на виробництво та реалізацію продукту	Необхіднот отримання великої кількості дозвільних документів на виробництво та реалізацію продукту, що потребує значних коштів та часу	Процедура отримання дозвільних документів для виробництва та реалізації продукту потребує незначних коштів та часу	Необхідно тільки повідомлення відповідним органам про виробництво та Реалізацію продукту	Відсутні будь-які регламентні обмеження на виробництво та реалізацію продукту

Додаток Ж. РОЗРАХУНОК ВИТРАТ НА ЗАРОБІТНЮ ПЛАТНЮ

Найменування посади	Місячний посадовий оклад, грн	Оплата за робочий день, грн.	Число днів роботи	Витрати на заробітну плату, грн.
Керівник проекту	14700	668.2	20	13364
Науковий співробітник	10500	477.3	25	11932.5
Програміст	24500	1113.6	15	16704
Frontend розробник	19500	886.4	10	8864
Тестувальник	8500	386.4	8	3091.2
Сума витрат (грн)	53955.7			

Додаток К. ТЕКСТ ЗАСОБУ

Scratch.py

```
import hashlib
import json
import os.path
import re
import socket
import ssl
import time
from datetime import datetime

import redis
import requests
import tldextract
import urllib3
import whois
from bs4 import BeautifulSoup
from dateutil.relativedelta import relativedelta
from dotenv import load_dotenv
from requests.adapters import HTTPAdapter
from urllib3.util.retry import Retry


class siteChecker():
    def __init__(self):
        self.rclient = redis.Redis(host='localhost', port=6379, decode_responses=True)
        urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)
        load_dotenv()
        self._domain = ""
        self._url = ""
        self.without_prefix = False
        self._md5_hash = hashlib.md5(self._url.encode()).hexdigest()
        self._api_key = os.getenv("APIKEY")
        self.USER_AGENTS = [
            "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/96.0.4664.110 Safari/537.36",
        ]
        self.PROXY = {
            "http": "http://proxy",
            "https": "https://proxy"
        }
        with open("./brandlist.txt", "r", encoding="utf-8") as f:
            self.BRAND_KEYWORDS = [keyword.replace("\n", "").strip() for keyword in
list(f.readlines())]

    def set_target(self):
        target = self.rclient.lpop('target:domain')

        if not target:
            print("No target found in the queue.")
            return
        if target.startswith("http://") or target.startswith("https://"):
            self._domain = target.split("/")[2]
            self._url = target
            self._md5_hash = hashlib.md5(self._url.encode()).hexdigest()
        else:
            self._url = target
            self._domain = target
            self._md5_hash = hashlib.md5(self._url.encode()).hexdigest()

    def analize(self):
```

```
        self.set_target()
        self.write_json("https", self.check_https())
        self.write_json("check_md5", self.check_md5())
        self.write_json("check_keywords", self.check_domain_keywords(self._domain))
        self.write_json("whois", self.domain_whois_lookup(self._domain))
        self.write_json("certificate_data", self.fetch_certificate_data(self._domain))
        self.write_json("content_obfuscation", self.detect_content_obfuscation(self._url))
        self.write_json("js_obfuscation", self.detect_js_obfuscation(self._url))
        self.write_json("diff_geo", self.compare_responses(self._url))
        self.write_json("redirect", self.check_redirect(self._url))
        self.write_json("result_vt", json.dumps(self.result_vt(self._api_key)))
        report = self.form_report()
        self.write_json("result", report)


    def check_https(self):
        try:
            if self._url.split("/")[0] in "https:":
                self._domain = self._url.split("/")[2]
                self._url = self._url.split("/")[2]
            if self._url.startswith("https://") or self._url.startswith("http://"):
                print({"https": self._url.startswith("https://")})
                return {"https": self._url.startswith("https://")}
            else:
                try:
                    context = ssl.create_default_context()
                    with socket.create_connection((self._url, 443)) as sock:
                        with context.wrap_socket(sock, server_hostname=self._url) as ssock:
                            cert = ssock.getpeercert()
                           # print(f"SSL certificate for {self._url} is valid.")
                            self._url = "https://" + self._url
                            return {"https": True}

                except Exception as e:
                    print(f"HTTPS failed for {self._url}: {e}")
                    self._url = "http://" + self._url
                    return {"https": False}
        except Exception as e:
            print(f"Помилка: {e}")
            return {"https": False}

    def check_md5(self):
        if self._md5_hash in self.rclient.lrange("md5:hashes", 0, -1):
            print({"md5": self._md5_hash, "already_present": True, "phishing ": True})
            return json.dumps({"md5": self._md5_hash, "already_present": True,
                               "phishing ": True})  # Повертаємо словник без json.dumps()
та json.loads()
        else:
            print({"md5": self._md5_hash, "already_present": False, "phishing": None})
            return json.dumps({"md5": self._md5_hash, "already_present": False,
                               "phishing": None})  # Повертаємо словник без json.dumps() та
json.loads()

    def check_redirect(self, url):
        try:
            result = {
                "destination": url,
                "redirect": False
            }
            response = requests.get(url, allow_redirects=False, verify=False)
            if 300 <= response.status_code < 400:
                result["destination"] = response.headers.get('Location')
                result["redirect"] = True
               # print(result)
```

```
                return json.dumps(result, default=self.datetime_handler)
            else:
                # print(result)
                return json.dumps(result, default=self.datetime_handler)
        except requests.exceptions.RequestException as e:
            return {
                "error": str(e),
                "status": "failed",
                "domain": url
            }

    def check_domain_keywords(self, domain):
        domain_keywords_info = dict()
        for i in range(len(self.BRAND_KEYWORDS)):
            domain_keywords_info[f'{self.BRAND_KEYWORDS[i]}'] = False
        try:
            # Витягуємо основний домен
            extracted = tldextract.extract(domain)
            domain = extracted.domain.lower()

            # Перевіряємо основний домен на брендові ключові слова
            for keyword in self.BRAND_KEYWORDS:
                if keyword in domain:
                    domain_keywords_info[keyword] = True
          #  print(f"Keywords count: {len([i for i in domain_keywords_info.values() if
i])}")
            return json.dumps(domain_keywords_info, default=self.datetime_handler)
        except Exception as e:
            return {
                "error": str(e),
                "status": "failed",
                "domain": domain
            }

    def form_report(self):
        base = f"/var/www/phishscan.info/result/{self._md5_hash}"

        categories = [
            "https", "cert", "redirect", "keywords", "geo",
            "domain_reg", "vt_scan", "md5_hash",
            "content_obfuscation", "script_obfuscation", "negh_domains"
        ]

        ai = {
            "https": 1.12, "cert": 2, "redirect": 1.15, "keywords": 1.2, "geo": 1.25,
            "domain_reg": 1.5, "vt_scan": 2.1, "md5_hash": 1.15,
            "content_obfuscation": 1.5, "script_obfuscation": 1.35, "negh_domains": 1.8
        }
        wi = {
            "https": 0.64, "cert": 0.64, "redirect": 0.72, "keywords": 0.92, "geo": 0.68,
            "domain_reg": 0.68, "vt_scan": 0.84, "md5_hash": 0.96,
            "content_obfuscation": 0.32, "script_obfuscation": 0.24, "negh_domains": 0.48
        }

        def calc(category):
            return wi[category] * ai[category]

        file_map = {
            "cert": "certificate_data.json",
            "https": "https.json",
            "vt_scan": "result_vt.json",
            "redirect": "redirect.json",
            "keywords": "check_keywords.json",
            "geo": "diff_geo.json",
```

```
            "domain_reg": "check_date.json",
            "md5_hash": "check_md5.json",
            "content_obfuscation": "content_obfuscation.json",
            "script_obfuscation": "js_obfuscation.json",
            "negh_domains": "neighbors.json",
        }

        res = {category: {"coef": 0, "data": ""} for category in categories}
        data = {}

        # Load data from files
        for key, file_name in file_map.items():
            file_path = f"{base}/{file_name}"
            if not os.path.exists(file_path):
                print(f"File {file_path} not found.")
                continue
            try:
                with open(file_path, "r", encoding="utf-8") as file:
                    content = file.read().strip()
                    if content:
                        data[key] = json.loads(content)
                    else:
                        print(f"File {file_path} is empty.")
            except json.JSONDecodeError as e:
                print(f"JSON decode error in file {file_path}: {e}")
            except Exception as e:
                print(f"Error reading file {file_path}: {e}")

        # Conditions for categories
        conditions = {
            "https": lambda d: d.get('http', False),
            "cert": lambda d: len(d.get('http', [])) > 0,
            "vt_scan": lambda d: d.get('malicious', 0) > 0 or d.get('suspicious', 0) > 0,
            "redirect": lambda d: d.get('redirect', False),
            "keywords": lambda d: any(d.values()),
            "geo": lambda d: not d.get('Same_answers', True),
            "domain_reg": lambda d: not d.get('pass_six_month', True),
            "md5_hash": lambda d: d.get('already_present', False),
            "content_obfuscation": lambda d: any(d.values()),
            "script_obfuscation": lambda d: sum(pattern.get('matches', 0) for pattern in
d["matches"]) > 0,
            "negh_domains": lambda d: d.get('neighbors', 0) > 0,
        }

        # Evaluate conditions and calculate coefficients
        for category in categories:
            if category in data and conditions[category](data[category]):
                res[category]["coef"] = calc(category)
            res[category]["data"] = self.format_data(category, data[category])
        chisel = sum(ai[category] * wi[category] * res[category]["coef"] for category in
categories)
        znamenn = sum(ai[category] * wi[category] for category in categories)
        res["general"] = chisel / znamenn if znamenn != 0 else 0
        if res["general"] > 0.31:
            self.rclient.lpush("md5:hashes", self._md5_hash)
            with open(f"/var/www/phishscan.info/result/{self._md5_hash}/whois.json","r",
encoding="utf-8") as file:
                content = json.loads(file.read().strip())
                for item in content["name_servers"]:
                    self.rclient.lpush("name_servers", item)
        return res

    def format_data(self, category, content):
        if category == "cert":
```

```
            return ", ".join(content.get("sertificates", []))
        elif category == "vt_scan":
            return f"Malicious: {content.get('malicious', 0)}, Suspicious:
{content.get('suspicious', 0)}"
        elif category == "redirect":
            return f"Redirect: {'Yes' if content.get('redirect') else 'No'}"
        elif category == "keywords":
            return f"Strange keywords: {'Yes' if any(content.values()) else 'No'}"
        elif category == "geo":
            return f"Different answers on different GEO: {'Yes' if not
content.get('Same_answers', True) else 'No'}"
        elif category == "domain_reg":
            return f"Domain registered in last six months: {'Yes' if not
content.get('pass_six_month', True) else 'No'}"
        elif category == "md5_hash":
            return f"Early checked with phishing: {'Yes' if content.get('already_present',
False) else 'No'}"
        elif category == "content_obfuscation":
            return f"Content obfuscation: {'Yes' if any(content.values()) else 'No'}"
        elif category == "script_obfuscation":
            return f"Script obfuscation: {'Yes' if sum(pattern['matches'] for pattern in
content['matches']) > 0 else 'No'}"
        elif category == "negh_domains":
            return f"Present neighbors on CloudFlare: {'Yes' if content.get('neighbors', 0)
> 0 else 'No'}"
        elif category == "https":
            return f"Https: {'Yes' if content.get('https', False) else 'No'}"
        return ""

    def get_report(self, hash):
        with open(f"/var/www/phishscan.info/result/{hash}/result.json") as file:
            return file

    def domain_whois_lookup(self, domain):
        """
        Perform a WHOIS lookup for a domain and return results in JSON format.

        Args:
            domain (str): Domain name to lookup

        Returns:
            dict: WHOIS information in JSON-compatible dictionary format
        """
        try:
            # Perform WHOIS lookup
            w = whois.whois(domain)

            # Extract relevant WHOIS information
            whois_info = {
                "domain_name": w.domain_name if w.domain_name else "",
                "registrar": w.registrar if w.registrar else "",
                "creation_date": w.creation_date if w.creation_date else "",
                "expiration_date": w.expiration_date if w.expiration_date else "",
                "updated_date": w.updated_date if w.updated_date else "",
                "name_servers": w.name_servers if w.name_servers else "",
                "status": w.status if w.status else "",
                "emails": w.emails if w.emails else "",
                "registrant": {
                    "name": w.name if w.name else "",
                    "organization": w.org if w.org else "",
                    "country": w.country if w.country else ""
                },
                "admin": {
                    "email": w.admin_email if w.admin_email else "",
```

```
                    "country": w.admin_country if w.admin_country else ""
                },
                "tech": {
                    "email": w.tech_email if w.tech_email else "",
                    "country": w.tech_country if w.tech_country else ""
                }
            }
            #print(f'{whois_info}')
            # перевіряємо дату реєстрації
            self.write_json("check_date", self.check_date(whois_info["creation_date"]))
            # Завантажуємо список всіх відомих name_servers
            # name_servers_data = self.rclient.get("name_servers")
            self.write_json("neighbors", self.check_neighbors(whois_info['name_servers']))
            # Записуємо значення поточного домену і відправляємо в редіс
            # retrieved_data[f"{self._md5_hash}"] = f"{whois_info['name_servers']}"
            # self.rclient.set(f"name_servers", json.dumps(retrieved_data))

            return whois_info

        except Exception as e:
            whois_info = {
                "domain_name": '',
                "registrar": '',
                "creation_date": '',
                "expiration_date": '',
                "updated_date": '',
                "name_servers": '',
                "status": '',
                "emails": '',
                "registrant": {
                    "name": '',
                    "organization": '',
                    "country": ''
                },
                "admin": {
                    "email": '',
                    "country": ''
                },
                "tech": {
                    "email": '',
                    "country": ''
                }
            }
            self.write_json("check_date", self.check_date(whois_info["creation_date"]))
            self.write_json("neighbors", self.check_neighbors(whois_info['name_servers']))

            return whois_info

    def check_neighbors(self, name_servers):
        if name_servers:
            neighbors = {"neighbors": 0}
            name_servers_data = self.rclient.lrange("name_servers",0,-1)
            if name_servers[0] in name_servers_data and name_servers[1] in
name_servers_data and "cloudflare" in name_servers[0]:
                neighbors["neighbors"] += name_servers_data.count(name_servers[0])
                return dict({'neighbors': neighbors["neighbors"]})
            else:
                return dict({'neighbors': 0})
        else:
            print("No nameservers")
            return dict({'neighbors': 0})

    def detect_content_obfuscation(self, url):
        try:
```

```
            # Отримання HTML-коду
            response = requests.get(url, verify=False)
            html = response.text

            # Пошук ознак обфускації
            obfuscation_content_signs = {
                "Довгі рядки без пробілів": False,
                "Base64-закодовані дані": False,
                "Unicode або Hex-коди": False,
                "Приховані елементи": 0
            }

            # 1. Довгі рядки без пробілів
            if re.search(r'[a-zA-Z0-9]{100,}', html):
                obfuscation_content_signs['Довгі рядки без пробілів'] = True

            # 2. Base64-кодовані дані
            if re.search(r'[A-Za-z0-9+/]{50,}={0,2}', html):
                obfuscation_content_signs['Base64-закодовані дані'] = True

            # 3. Unicode або hex-коди
            if re.search(r'\\u[0-9a-fA-F]{4}|&#x[0-9a-fA-F]+;', html):
                obfuscation_content_signs['Unicode або Hex-коди'] = True

            # 4. Приховані елементи
            soup = BeautifulSoup(html, 'html.parser')
            hidden_elements =
soup.find_all(style=re.compile(r'display:\s*none|opacity:\s*0'))
            if hidden_elements:
                obfuscation_content_signs['Приховані елементи'] = len(hidden_elements)
            print(obfuscation_content_signs)
            return json.dumps(obfuscation_content_signs, default=self.datetime_handler)

        except requests.exceptions.RequestException as e:
            return {
                "error": str(e),
                "status": "failed",
                "url": url
            }

    def check_date(self, creation_date):
        try:
            if creation_date != None:
                if len(creation_date) > 0:
                    creation_date = creation_date[0]
                else:
                    raise ValueError("Список creation_date порожній")

                # Перевірка, чи є creation_date рядком
                if not isinstance(creation_date, str):
                    raise TypeError("Очікується, що creation_date буде рядком")

                # Перевірка формату дати creation_date
                try:
                    earlier_date = datetime.strptime(creation_date, "%Y-%m-%d %H:%M:%S")
                except ValueError:
                    raise ValueError("Формат creation_date не відповідає '%Y-%m-%d
%H:%M:%S'")

                # Отримання поточного часу
                now = datetime.now()
                later_date = now  # Поточна дата та час

                # Обчислення дати через пів року від меншої
```

```
            six_months_later = earlier_date + relativedelta(months=6)
            pass_six_month = later_date >= six_months_later

            # Логування результату
            result = {
                "creation_date": creation_date,
                "check_date": now,
                "pass_six_month": pass_six_month
            }
            # print(result)
            return result
        else:
            now = datetime.now()
            return {
                "creation_date": now,
                "check_date": now,
                "pass_six_month": False
            }
    except Exception as e:
        now = datetime.now()
        print({
            "creation_date": now,
            "check_date": now,
            "pass_six_month": False
        })

        return {
            "creation_date": now,
            "check_date": now,
            "pass_six_month": False
        }

def detect_js_obfuscation(self, url):
    try:
        response = requests.get(url, timeout=10, verify=False)
        html = response.text

        # Пошук використання eval, Function, setTimeout
        suspicious_js_patterns = [
            r'eval\(',  # Виклики eval
            r'Function\(',  # Виклики Function
            r'setTimeout\(',  # Таймери для динамічного виконання
            r'decodeURIComponent|atob|btoa'  # Декодування даних
        ]

        suspicious_matches = {"matches": []}
        for pattern in suspicious_js_patterns:
            matches = re.findall(pattern, html)
            suspicious_matches["matches"].append({
                "pattern": pattern,
                "matches": len(matches)
            })

        # Якщо збігів не знайдено, додаємо нульові значення для всіх патернів
        if len(suspicious_matches) == 0:
            suspicious_matches["matches"] = {
                {"pattern": r'eval\(', "matches": 0},
                {"pattern": r'Function\(', "matches": 0},
                {"pattern": r'setTimeout\(', "matches": 0},
                {"pattern": r'decodeURIComponent|atob|btoa', "matches": 0},
            }

        # Не перетворюйте список об'єктів на словник, поверніть список
        print(suspicious_matches)
```

```
            return suspicious_matches

        except Exception as e:
            print(f"Error in detect_js_obfuscation: {e}")
            return {"matches": []}

    def compare_responses(self, url):
        responses = []

        for user_agent in self.USER_AGENTS:
            try:
                headers = {
                    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/96.0.4664.110 Safari/537.36"}
                # Виконання запиту через проксі
                response_proxied = requests.get(url, headers=headers, timeout=10,
verify=False)
                response = requests.get(url, headers=headers, timeout=10, verify=False)
                # Збереження тексту відповіді
                responses.append(response.text)
                responses.append(response_proxied.text)

            except requests.exceptions.RequestException as e:
                return {
                    "error": str(e),
                    "user_agent": user_agent,
                    "url": url
                }

        # Порівняння всіх отриманих відповідей
        for i in range(len(responses) - 1):
            if responses[i] != responses[i + 1]:
                res = {"Same_answers": False}
                print(res)
                return json.loads(json.dumps(res, default=self.datetime_handler))

        res = {"Same_answers": True}
        print(res)
        return json.dumps(res, default=self.datetime_handler)

    def scan_vt(self, target, apikey):
        url = "https://www.virustotal.com/api/v3/urls"
        payload = {"url": target}
        headers = {
            "accept": "application/json",
            "x-apikey": f"{apikey}",
            "content-type": "application/x-www-form-urlencoded"
        }
        try:
            response = requests.post(url, data=payload, headers=headers, timeout=15)
            response.raise_for_status()  # Перевіряє статус-код відповіді (викликає помилку
для статусів 4xx і 5xx)

            try:
                json_response = response.json()  # Спроба розпарсити JSON
               # print("Результат API запиту:", json_response)
                return json_response
            except ValueError as e:
                print("Помилка парсингу JSON:", e)
                return {"error": {"code": "InvalidJSON", "message": "Відповідь не є
коректним JSON"}}

        except requests.Timeout:
            print("Помилка: Таймаут запиту до API.")
```

```
            return {"error": {"code": "Timeout", "message": "Запит перевищив ліміт часу"}}

        except requests.RequestException as e:
            print(f"Помилка запиту до API scan_vt: {e}")
            return {"error": {"code": "RequestException", "message": str(e)}}

    def result_vt(self, apikey):
        try:
            # Отримуємо посилання на результат сканування
            scan_result = self.scan_vt(self._url, apikey)
            if 'error' in scan_result:
                raise ValueError(f"Помилка сканування: {scan_result['error']['message']}")

            api_link = scan_result['data']['links']['self']
            headers = {
                "accept": "application/json",
                "x-apikey": f"{apikey}"
            }

            # Очікуємо завершення обробки в черзі
            while True:
                try:
                    response = requests.get(api_link, headers=headers, timeout=15,
verify=False)
                    response.raise_for_status()  # Перевіряємо статус-код відповіді

                    json_response = response.json()  # Парсимо JSON
                    if 'data' not in json_response or 'attributes' not in
json_response['data']:
                        raise ValueError("Некоректна структура відповіді API.")

                    # Перевірка статусу
                    status = json_response['data']['attributes'].get('status')
                    if status == 'queued':
                        time.sleep(1)  # Затримка перед повторним запитом
                        continue

                    # Повертаємо статистику, якщо сканування завершено
                    stats = json_response['data']['attributes'].get('stats')
                    if stats is None:
                        raise ValueError("Статистика не знайдена у відповіді API.")

                    print(stats)
                    return stats

                except requests.Timeout:
                    print("Таймаут під час запиту до API.")
                    time.sleep(1)  # Затримка перед повторною спробою
                except requests.RequestException as e:
                    print(f"Помилка запиту до API result_vt: {e}")
                    return {"error": {"code": "RequestException", "message": str(e)}}

        except KeyError as e:
            print(f"Ключ {e} не знайдено у відповіді.")
            return {"error": {"code": "KeyError", "message": str(e)}}
        except ValueError as e:
            print(f"Помилка обробки даних: {e}")
            return {"error": {"code": "ValueError", "message": str(e)}}
        except Exception as e:
            print(f"Непередбачена помилка: {e}")
            return {"error": {"code": "Exception", "message": str(e)}}

    def write_json(self, file_name, data):
        if not os.path.exists(f"/var/www/phishscan.info/result/{self._md5_hash}"):
```

```
            os.makedirs(f"/var/www/phishscan.info/result/{self._md5_hash}")
        if isinstance(data, dict):
            data = json.dumps(data, default=self.datetime_handler)
        with open(f'/var/www/phishscan.info/result/{self._md5_hash}/{file_name}.json', "w",
encoding="utf-8") as file:
            file.write(data)

    def datetime_handler(self, obj):
        if isinstance(obj, datetime):
            return obj.isoformat()
        return str(obj)

    def fetch_certificate_data(self, domain: str):

        # Setup request parameters
        url = "https://crt.sh"
        params = {
            "q": domain,
            "output": "json"
        }
        headers = {
            "User-Agent": "Mozilla/5.0",
            "Accept-Encoding": "gzip, deflate"
        }

        # Make the request

        session = requests.Session()
        retries = Retry(total=8, backoff_factor=1, status_forcelist=[500, 502, 503, 504])
        session.mount('https://', HTTPAdapter(max_retries=retries))

        response = session.get(url, params=params, headers=headers, timeout=30,
verify=False)

        response.raise_for_status()  # Raise an exception for bad status codes

        # Parse JSON response
        data = response.json()

        # Extract common_name and name_value fields
        results = set()  # Using set to automatically remove duplicates
        for cert in data:
            if "common_name" in cert:
                results.add(cert["common_name"])
            if "name_value" in cert:
                results.add(cert["name_value"])

        ans = {"sertificates": sorted(results)}
       # print(f"Count of certificate: {len(ans['sertificates'])}")
        return json.dumps(ans, default=self.datetime_handler)


if __name__ == "__main__":
    n = siteChecker()
    rclient = redis.Redis(host='localhost', port=6379, decode_responses=True)

    while True:
        domain_length = rclient.llen("target:domain")
        if domain_length == 0:
            print("No target in the queue. Retrying...")
            time.sleep(1)
            continue
        else:            n.analize()
```

ЗАТВЕРДЖУЮ

Директор ТОВ «УКРАЇНСЬКІ КІБЕРНЕТИЧНІ РОЗРОБКИ»

Павло ЛЯШКО

«04» квітня 2024 р.

АКТ

про впровадження результатів магістерської кваліфікаційної роботи

Савчука Івана Борисовича

Комісія у складі: голова комісії – директор Ляшко П. О., спеціаліст з інформаційної безпеки Творун Р. П., розглянувши матеріали бакалаврської роботи Савчука І. Б. на тему “Метод та засіб виявлення фішингових веб-сайтів”, склала цей акт про те, що у ТОВ «УКРАЇНСЬКІ КІБЕРНЕТИЧНІ РОЗРОБКИ» впроваджено результати цієї роботи, а саме:

- метод ідентифікації ознак фішингових веб-сайтів.
- Результат експертного опитування щодо ознак фішингу

Впроваджені результати дозволяють здійснювати розробку нових алгоритмів захисту конфіденційних даних користувачів, зокрема працівників підприємства, що дає змогу підвищити рівень захисту персоналу компанії.

Голова комісії:
Директор Павло ЛЯШКО

Члени комісії:
Спеціаліст з інформаційної безпеки Руслан ТВОРУН

ІЛЮСТРАТИВНА ЧАСТИНА

МЕТОД ТА ЗАСІБ ВИЯВЛЕННЯ ФІШИНГОВИХ ВЕБ-САЙТІВ

Виконав: студент 2 курсу, групи БС-23 м
спеціальності 125 Кібербезпека та захист
інформації
_____ Іван САВЧУК
«13» 12 2024 р.

Керівник: к. т. н., доцент, доцент каф. ЗІ
_____ Юрій БАРИШЕВ
«13» 12 2024 р.

ТЕНДЕНЦІЇ РОЗВИТКУ ЗАГРОЗИ ФІШИНГОВИХ ВЕБ-САЙТІВ

ПОРІВНЯЛЬНИЙ АНАЛІЗ МЕТОДІВ ВИЯВЛЕННЯ ФІШИНГОВИХ ВЕБ-САЙТІВ

Метод захисту	Опис	Засіб, що реалізує метод
Списки відомих фішингових сайтів	Використання баз даних з адресами ідентифікованих фішингових сайтів для блокування доступу та попередження користувачів	Google Safe Browsing [14], PhishTank [15]
Евристичний аналіз	Застосування алгоритмів машинного навчання та статистичних моделей для виявлення підозрілих ознак фішингових сайтів	Google Safe Browsing, IBM Trusteer Rapport [16].
Системи репутації веб-сайтів	Збір та аналіз відгуків користувачів та експертів щодо надійності та безпеки веб-сайтів	Web of Trust, Norton Safe Web [17]
Розширення веб-браузерів	Інтеграція засобів захисту безпосередньо у веб-браузери для перевірки відвідуваних сайтів на наявність ознак фішингу	Microsoft Defender SmartScreen, Malwarebytes Browser Guard
Двофакторна автентифікація	Вимога додаткового підтвердження особи користувача, окрім введення пароля, для доступу до облікового запису	Google Authenticator, Microsoft Authenticator, Authy
Освіта та інформування користувачів	Проведення навчальних програм та кампаній з підвищення обізнаності користувачів щодо фішингових атак	Курси на онлайн-платформах штибу Coursera, Prometeus. Інформування через мас-медіа
Системи моніторингу та виявлення	Використання технологій аналізу мережевого трафіку та машинного навчання для виявлення фішингових атак на корпоративному рівні	Darktrace [18], Cisco Umbrella, Splunk [19].
Співпраця організацій з кібербезпеки	Обмін інформацією про загрози, розробка рекомендацій та співпраця з правоохоронними органами для протидії фішингу	Інформаційний обмін через Information Sharing and Analysis Centers, спільні ініціативи з правоохоронними органами, наприклад, Europol, CERT-UA.

МАТЕМАТИЧНИЙ ОПИС ПРОЦЕСУ ВИЯВЛЕННЯ ФІШИНГОВИХ ВЕБ-САЙТІВ

$$M = \{I, S, F, O\}$$

$$I = \{URL, x_i, w_i, s_i\}$$

$$S = \{x_i, w_i, s_i\}$$

$$F = \{f_{collect}, f_{ident}, f_R, f_{ter}\}$$

$$O = \{R_{phishing}\}$$

МЕТОД ВИЯВЛЕННЯ ФІШИНГУ

$$w_i = \frac{\frac{\sum_{j=1}^{n} x_{ij}}{n}}{B_{max}}$$

$$\text{FPR} = \frac{FP}{FP+TN}$$

$$\text{FNR} = \frac{FN}{FN+TP}$$

$$a_i = 1 - FPR_i + (1 - FNR_i)$$

$$R = \frac{\sum_{i \in V} a_i \cdot w_i \cdot x_i}{\sum_{i \in V} a_i \cdot w_i}$$

РЕЗУЛЬТАТИ ЕКСПЕРТНОГО ОПИТУВАННЯ

Ознака	Середній бал	Розподіл оцінок
Використання брендових ключових слів у під-доменах	4.8	4 (20.0%), 5 (80.0%)
Наявність MD5-гешів файлів у базі даних	4.8	4 (20.0%), 5 (80.0%)
Попередження від сервісів сканування	4.2	4 (80.0%), 5 (20.0%)
Відкриття домену з різних геолокацій	3.8	1 (20.0%), 2 (20.0%), 3 (20.0%), 4 (20.0%), 5 (40.0%)
Редірект з одного посилання на інше	3.6	3 (40.0%), 4 (60.0%)
Свіжа дата реєстрації домену	3.6	3 (40.0%), 4 (60.0%)
Сусідні домени на одному акаунті	3.2	3 (80.0%), 4 (20.0%)
Обфускація контенту сайту	2.4	2 (60.0%), 3 (40.0%)
Обфускація скриптів	2.0	1 (20.0%), 2 (60.0%), 3 (20.0%)

Ознака	Експерти				
	Баришев Ю.В	Куперштейн Л.М	Войтович О. П	Дронь А.А	Кореньков А.О
Редірект з одного посилання на інше	3	4	3	4	4
Використання брендових ключових слів у піддоменах	4	5	5	5	5
Відкриття домену з різних геолокацій	5	3	2	5	4
Свіжа дата реєстрації домену	4	3	4	4	3
Попередження від сервісів сканування	4	5	4	4	4
Наявність MD5-хешів файлів у базі даних	5	5	5	5	4
Обфускація контенту сайту	2	2	4	2	2
Обфускація скриптів на сайті	1	2	3	2	2
Сусідні домени на одному акаунті CloudFlare	3	3	4	3	3

РЕЗУЛЬТАТИ БЛОКОВОГО ТЕСТУВАННЯ ЗАСОБУ

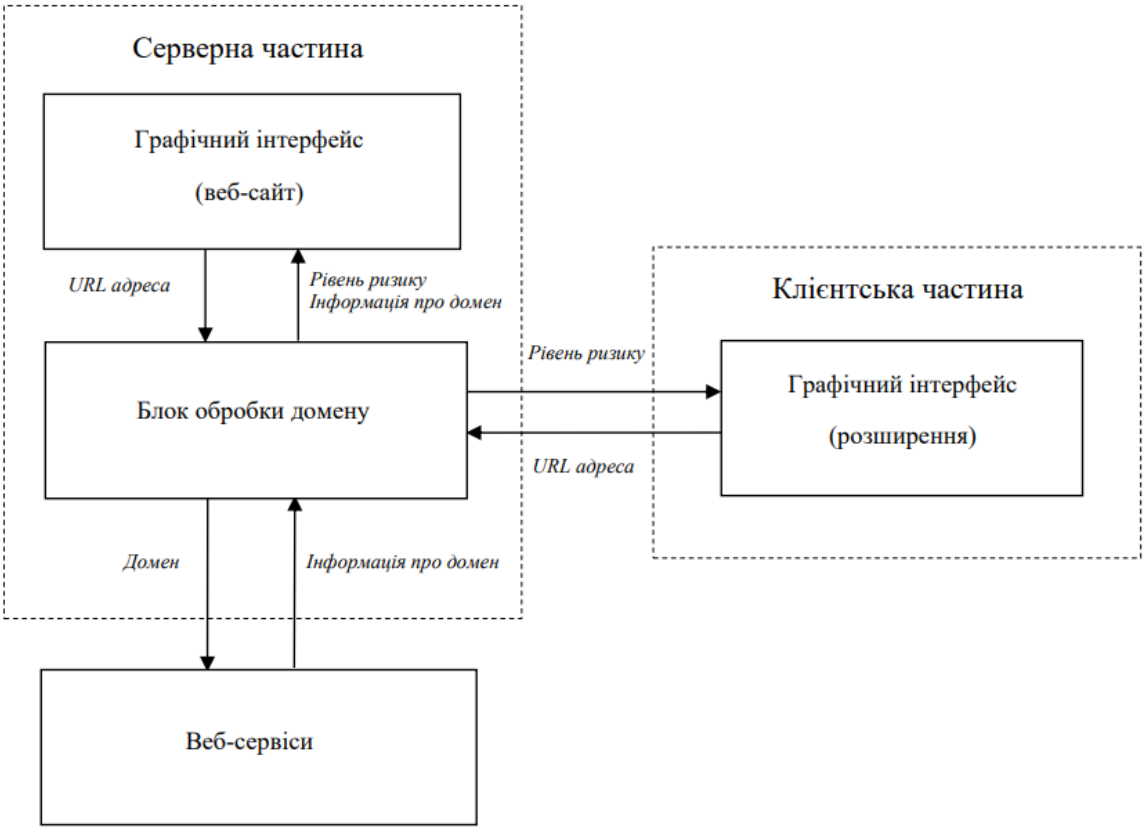
```
(.venv) PS C:\Users\Savch\PycharmProjects\DiplomaWork> python -m unittest test_phishing.py -v
test_empty_redis_queue (test_phishing.TestRedisOperations)
Test behavior with an empty Redis queue ... ok
test_multiple_domains (test_phishing.TestRedisOperations)
Test handling multiple domains in Redis ... ok
test_redis_add_and_retrieve (test_phishing.TestRedisOperations)
Test adding and retrieving a domain from Redis ... ok
test_retrieve_domain_from_redis (test_phishing.TestRedisOperations)
Test retrieving a domain from Redis ... ok

----------------------------------------------------------------------
Ran 4 tests in 0.024s

OK
```

Результати блокового тестування для Redis

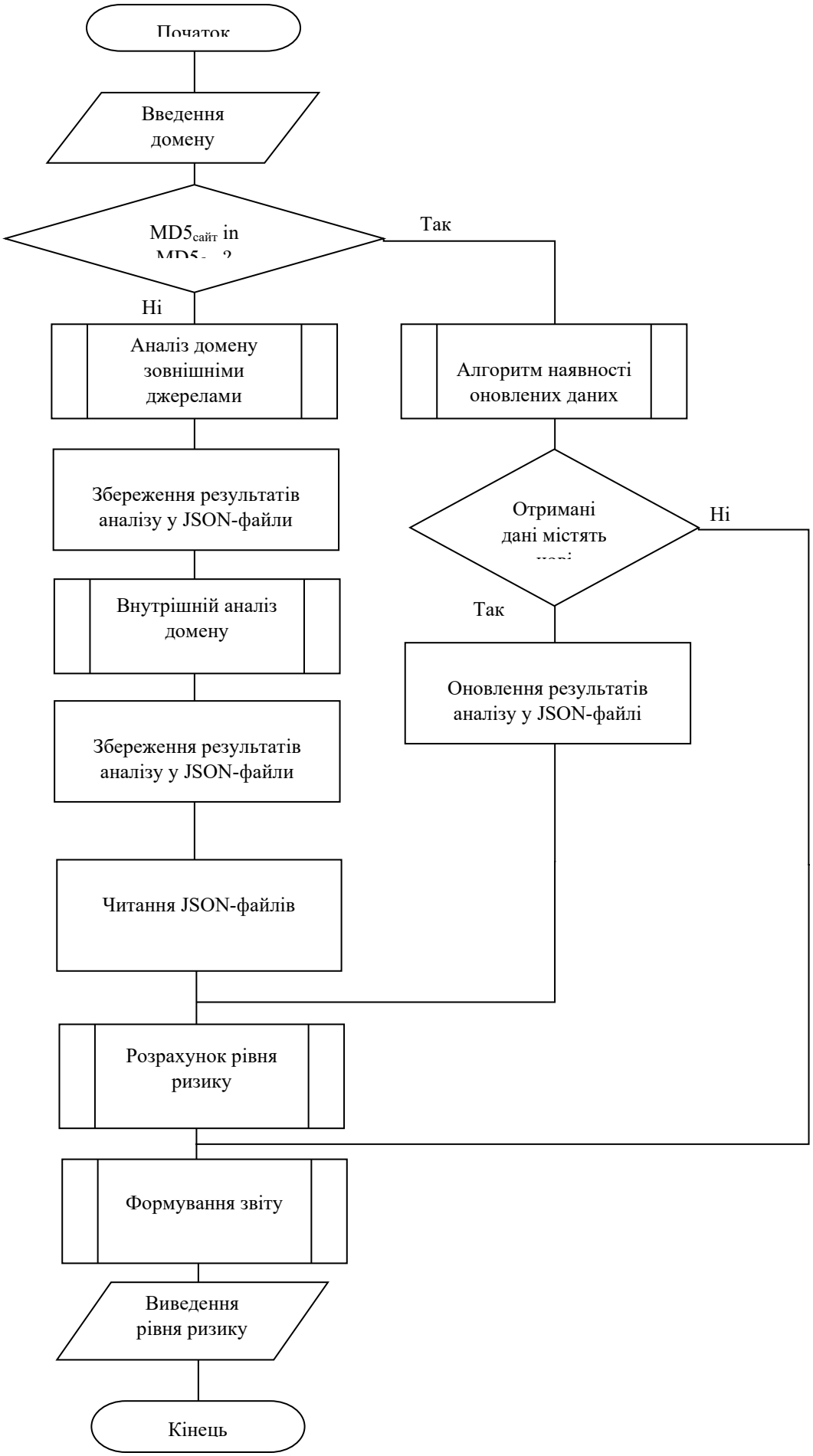
```
(.venv) PS C:\Users\Savch\PycharmProjects\DiplomaWork> python -m unittest test_phishing.py -v
test_certificate_check (test_phishing.TestSiteChecker)
Test SSL certificate existence ... ok
test_final_report_generation (test_phishing.TestSiteChecker)
Test final report generation ... ok
test_https_check (test_phishing.TestSiteChecker)
Test the correctness of the HTTPS check ... ok
test_keyword_detection (test_phishing.TestSiteChecker)
Test keyword detection in the domain name ... ok
test_md5_hash_generation (test_phishing.TestSiteChecker)
Test MD5 hash generation for a domain ... ok
test_virustotal_integration (test_phishing.TestSiteChecker)
Test integration with the VirusTotal API ... ok
test_whois_lookup (test_phishing.TestSiteChecker)
Test WHOIS data processing ... ok


----------------------------------------------------------------------
Ran 7 tests in 2.536s
```

Результати блокового тестування SiteChecker

АРХІТЕКТУРА ЗАСОБУ



УЗАГАЛЬНЕНИЙ АЛГОРИТМ РОБОТИ ЗАСОБУ



РЕЗУЛЬТАТИ ЕКСПЕРИМЕНТАЛЬНОГО ДОСЛІДЖЕННЯ ЗАСОБУ ВИЯВЛЕННЯ ФІШИНГОВИХ ВЕБ-САЙТІВ

Ознака	**FP**	**FN**	**TP**	**TN**
Перенаправлення	2	1	3	4
Ключові слова	2	2	2	4
Відкриття з різних локацій	1	2	2	5
Відсутність сертифікатів	0	0	4	5
Свіжа дата реєстрації домену	1	1	3	5
Попередження від сервісів сканувань	0	0	4	6
Обфускація контенту	1	2	2	5
Обфускація скриптів	2	2	2	4
Виявлені сусідні домени	0	1	3	6

Результат до інтеграції корегуючих коефіцієнтів

Ознака	**FP**	**FN**	**TP**	**TN**
Перенаправлення	1	0	4	5
Ключові слова	1	1	4	5
Відкриття з різних локацій	0	0	5	6
Відсутність сертифікатів	0	0	4	5
Свіжа дата реєстрації домену	0	1	4	6
Попередження від сервісів сканувань	0	0	4	6
Обфускація контенту	1	1	4	5
Обфускація скриптів	1	0	4	6
Виявлені сусідні домени	0	0	4	6

Результат після інтеграції корегуючих коефіцієнтів

ПОКАЗНИКИ ЕКОНОМІЧНОЇ ЕФЕКТИВНОСТІ

Критерії оцінювання	Експерти		
	Творун Р.П	Ляшко П.О	Кореньков А.О
	Бали		
Технічна здійсненність концепції	3	4	4
Наявність аналогів на ринку	3	3	3
Цінова політика	3	5	4
Технічні та споживчі властивості виробу	3	3	4
Експлуатаційні витрати	4	3	3
Ринок збуту	3	4	3
Конкурентоспроможність	3	3	4
Фахівці з технічної і комерційної реалізації	3	3	3
Фінансування	4	3	3
Матеріально-технічна база	4	3	4
Термін реалізації ідеї	4	5	3
Супровідна документація	3	3	4
Сума	40	42	42
Середньоарифметична сума балів	41		

$$K_{ІНТ} = 1 \cdot \frac{1.2265}{0.55} = 2.23$$

$$ЗВ = \frac{243031.3}{0.5} = 486062.6$$

$$Т_{ок} = 1/1.19 = 0.84$$