Міністерство освіти і науки України
Київський Національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ПОЯСНЮВАЛЬНА ЗАПИСКА
Дипломної роботи

магістра
(назва освітньо-кваліфікаційного рівня)

галузь знань *12 Інформаційні технології*
(шифр і назва галузі знань)

спеціальність *125 Кібербезпека*
(код і назва спеціальності)

освітній рівень ***магістр***
(назва освітнього рівня)

кваліфікація
(код і назва кваліфікації)

на тему: *Методи захисту від фішингу*

Виконавець: студент *2* **курсу, групи** *КБм-21*

Романюк Валентин Вікторович
(підпис) (прізвище ім'я по-батькові)

	Прізвище, ініціали	Оцінка	Підпис
Науковий керівник	*Бабенко Т. В.*		
Рецензент			
Нормоконтроль			

Київ
2021

Міністерство освіти і науки України

Київський Національний університет імені Тараса Шевченка

Факультет інформаційних технологій
Кафедра кібербезпеки та захисту інформації

ЗАТВЕРДЖЕНО:
завідувач кафедри
кібербезпеки та захисту інформації

_____Лукова-Чуйко Н. В.

«_____»_____20___року

ЗАВДАННЯ
на виконання дипломної роботи

спеціальності *125 Кібербезпека*
(код і назва спеціальності)

студенту *КБм-21* *Романюку Валентину Вікторовичу*
(група) (прізвище ім'я по-батькові)

Тема дипломного роботи *Методи захисту від фішингу*

1. ПІДСТАВИ ДЛЯ ПРОВЕДЕННЯ РОБОТИ

Рішення засідання кафедри кібербезпеки та захисту інформації факультету інформаційних технологій протокол № 2 від 08.10.2020

2. МЕТА ТА ВИХІДНІ ДАНІ ДЛЯ ПРОВЕДЕННЯ РОБІТ

Об'єкт досліджень *Процес захисту від фішингових атак.*

Предмет досліджень *Методи захисту від фішингу, що забезпечують виявлення фішингових вкладень.*

Мета *Підвищення рівня виявлення фішингових вкладень.*

Вихідні дані для проведення роботи *Методи захисту від фішингу.*

3. ОЧІКУВАНІ НАУКОВІ РЕЗУЛЬТАТИ

Наукова новизна *удосконалення методів захисту від фішингу за допомогою синтезу нової моделі виявлення фішингових вкладень*

Практична цінність *покращення системи захисту від фішингових атак.*

4. ВИМОГИ ДО РЕЗУЛЬТАТІВ ВИКОНАННЯ РОБОТИ

Робота виконана у повному обсязі відповідно до теми.

5. ЕТАПИ ВИКОНАННЯ РОБОТИ

Найменування етапів робіт	Строки виконання робіт (початок-кінець)
Розробка плану для досягнення мети роботи	20.09.2020 – 19.10.2020
Аналіз літературних джерел	20.10.2020 – 14.01.2021
Розробка методів захисту від фішингу	15.01.2021 – 06.05.2021
Оформлення і друк пояснювальної записки	07.05.2021 – 13.05.2021

6. РЕАЛІЗАЦІЯ РЕЗУЛЬТАТІВ ТА ЕФЕКТИВНІСТЬ

Економічний ефект *Зниження збитків через викрадення даних*

Соціальний ефект *Покращення технологій забезпечення захисту інформації*

7. ДОДАТКОВІ ВИМОГИ

Завдання видав _____ _____
(підпис) (прізвище, ініціали)

Завдання прийняв до виконання _____ _____
(підпис) (прізвище, ініціали)

Дата видачі завдання: _____

Термін подання дипломної роботи до ЕК _____

УДК 004.492.2

РЕФЕРАТ

Пояснювальна записка до дипломної роботи «Методи захисту від фішингу»: 58 сторінок, 10 рисунків, 1 додаток та 3 таблиці. 31 літературне джерело.

Мета роботи – підвищення рівня виявлення фішингових вкладень.

Предмет дослідження – методи захисту від фішингу, що забезпечують виявлення фішингових вкладень.

Об’єкт дослідження - процес захисту від фішингових атак.

Методи дослідження – аналіз методів захисту від фішингу, аналіз фішингових атак, метод дерева рішень, моделювання, синтез, наукова абстракція, індукція та дедукція.

У роботі досліджено сучасні загрози та методи протидії фішинговим атакам. Запропоновано модель системи виявлення фішингових вкладень. Досліджено ефективність системи захисту від фішингу на базі запропонованої моделі.

Наукова новизна: удосконалено методику захисту від фішингу за допомогою використання методу дерева рішень, дана методика відрізняється від існуючих тим, що застосовує для виявлення фішингових вкладень уперше синтезовану для цього модель дерева рішень.

Актуальність теми: Фішингові атаки є досить поширеним інструментом для вторгнення (особливо в періоди кризи) в інформаційні системи, які відповідно змінюються для протидії новим засобам виявлення фішингових атак. У зв’язку з цим зростає значення заходів, які здійснюються для запобігання реалізації фішингових атак та пом’якшення їх наслідків.

Ключові слова: фішинг, фішингові атаки, система захисту інформації, дерево рішень.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ ТА СКОРОЧЕНЬ

ПК – Персональний комп'ютер

CSV – comma-separated values

DKIM – DomainKeys Identified Mail

DOM – Document Object Model

GNU – GNU's Not Unix

HTTP – Hypertext Transfer Protocol

IP – Internet Protocol address

IQ – intelligence quotient

PDF – Portable Document Format

SSL – Secure Sockets Layer

TLS – Transport Layer Security

URL – Uniform Resource Locator

VBA – Visual Basic for Applications

ЗМІСТ

ВСТУП

Фішинг - це кіберзлочин та техніка отриманння конфіденційних даних, яка становить небезпеку як для окремих осіб (крадіжка банківських даних та персональних даних) так і для великих і малих компаній (компрометація інформаційних систем компанії та крадіжка конфіденційної інформації чи крадіжка грошових коштів з рахунків компанії).

В 2020 році 95% організацій піддавались фішинговим атакам. 22% успішних витоків даних з організацій включали в себе фішингові атаки[1]. Оскільки фішинг є надзвичайно поширеним інструментом атаки і його доля серед інших типів атак застосованих в успішних витоках даних є доволі високою – даний тип атак потребує значних зусиль та відповідних механізмів для протидії.

Зловмисники можуть надсилати фішинг-повідомлення, щоб отримати доступ до систем жертв. Усі форми фішингу є соціальною інженерією, що передається в електронному вигляді. Фішинг може бути націлений, відомий як цільовий фішинг. Під час цільового фішингу зловмисник буде націлений на конкретну особу, компанію чи галузь. Більш загально, зловмисники можуть проводити нецільовий фішинг, наприклад у масових спам-кампаніях зловмисного програмного забезпечення[2].

Фішингові атаки спрацьовують, оскільки люди допитливі від природи і, як правило, не очікують, що з ними траплятимуться погані речі, коли вони виконують свої типові задачі. Фішинг, або соціальна інженерія насправді, є одним із найшвидших способів скомпрометувати мережу. Іноді найпростіший спосіб - це просто попросити про доступ.

Найбільш успішні фішингові атаки націлені на одну людину і персоналізовані для цієї людини таким чином, що це зовсім не схоже на атаку. Насправді спроба фішингу буде більше схожа на типову особисту або ділову взаємодію.

Інші, більш узагальнені, фішингові атаки потребують менше зусиль, але тим не менш залишаються ефективними. Це електронні листи, що попереджають про

пропущені відправлення або проблеми з доставкою пошти. Їх надсилають тисячам людей за день, і, можливо, піввідсотка або менше потраплять на гачок цього шахрайства.

Злочинці покладаються на обман і створюють відчуття терміновості для досягнення успіху за допомогою своїх фішинг-кампаній. Такі кризи, як пандемія коронавірусу, дають цим злочинцям велику можливість заманити жертв на фішинг. В 2020 році компанія Google блокувала 18 мільйонів фішингових листів на день, пов’язаних з темою коронавірусу[3].

Під час кризи люди перебувають на межі. Вони хочуть отримати інформацію та шукають вказівок від своїх роботодавців, уряду та інших відповідних органів влади. Електронний лист, який, як видається, надійшов від однієї з цих організацій і обіцяє нову інформацію або вказує одержувачам швидко виконати завдання, швидше за все, буде перевірений менше, ніж до кризи. Імпульсивний клік і перехід за шкідливим посиланням і пристрій жертви інфікований або обліковий запис зламано.

.

РОЗДІЛ 1
СУЧАСНІ ЗАГРОЗИ ТА МЕТОДИ ПРОТИДІЇ ФІШИНГОВИМ АТАКАМ

1.1 Фішингові атаки

Фішинг - це кібератака, яка використовує замаскований електронний лист як зброю. Мета полягає в тому, щоб обдурити одержувача електронної пошти щоб він повірив, що повідомлення - це те, що він хоче або очікує - наприклад, запит від його банку або повідомлення від когось із їхньої компанії, для того, щоб він натиснув посилання або завантажив вкладення.

Що насправді відрізняє фішинг від інших кібератак, то це форма повідомлення: зловмисники маскуються як довірений суб'єкт свого роду, часто це реальна або вигадана правдоподібна особа або компанія, з якою жертва може вести бізнес. Це один з найдавніших типів кібератак, починаючи з 1990-х років, який і досі залишається одним із найпоширеніших та згубних, а фішингові повідомлення та техніки стають все більш досконалими[4].

1.1.1. Класифікація фішингових атак відповідно до спрямування атаки

Масова фішингова розсилка. Найбільш поширеною формою фішингу є загальний тип масової розсилки коли хтось надсилає електронне повідомлення, прикидаючись кимось іншим, і намагається обдурити одержувача, щоб щось зробити, як правило, увійшовши на веб-сайт або завантаживши шкідливе програмне забезпечення. Атаки часто покладаються на підробку адреси електронної пошти, коли заголовок електронної пошти - поле "from" підробляється, щоб повідомлення виглядало так, ніби його надіслав надійний відправник.

Однак фішингові атаки не завжди виглядають як електронний лист із повідомленням про доставку, попереджувальне повідомлення від клієнт-банку про закінчення терміну дії паролів або електронне повідомлення Office 365 про квоти на зберігання. Деякі атаки розроблені спеціально для організацій та приватних осіб, а інші покладаються на інші методи, окрім електронної пошти[5].

Цільовий фішинг. Фішингові атаки отримали свою назву від концепції, що шахраї ловлять випадкових жертв, використовуючи підроблену або шахрайську електронну пошту як приманку. Цільові фішингові атаки продовжують фішингову аналогію, оскільки зловмисники спеціально націлюються на жертв та організації, що мають велику цінність. Замість того, щоб намагатись отримати банківські реквізити для 1000 користувачів, зловмиснику може бути вигідніше орієнтуватися на декілька підприємств. Зловмисник може націлитись на працівника, який працює в державному відомстві, або урядовця, щоб викрасти державну таємницю.

Цільові фішингові атаки є надзвичайно успішними, оскільки зловмисники витрачають багато часу на пошук інформації, що стосується одержувача, наприклад, посилання на конференцію, на якій одержувач міг щойно бути присутнім, або надсилання шкідливого вкладення, де назва файлу посилається на тему, яка зацікавить одержувача[5].

У фішинговій кампанії 2017 року група 74 (вона ж Sofact, APT28, Fancy Bear) націлилась на фахівців з кібербезпеки через електронну пошту, яка прикидалась пов’язаною з конференцією з кіберконфліктів у США, заходом, що був організованим Інститутом кібернетики армії Військової академії США, Кібервійськовою академією НАТО та Кооперативним центром кіберзахисту НАТО. Хоча CyCon є справжньою конференцією, вкладення насправді було документом, що містив шкідливий макрос Visual Basic for Applications (VBA), який завантажував і виконував розвідувальну шкідливу програму під назвою Seduploader[6].

Вейлінг (Whaling). Різні жертви - різні зарплати. Фішингова атака, спеціально націлена на топ-менеджерів підприємства називається вейлінгом (англ. whaling), оскільки жертва вважається цінною, а викрадена інформація буде ціннішою, ніж та, яку можна вкрасти у звичайного працівника. Облікові дані, що належать

генеральному директору відкриють більше дверей, ніж ті, якими володіють співробітники початкового рівня. Мета - викрасти дані, інформацію про співробітників та готівку.

Вейлінг також потребує додаткових досліджень, оскільки зловмисник повинен знати з ким обмінюється листами жертва та тип обговорень. Приклади можуть включати посилання на скарги клієнтів, судові повістки або навіть проблеми у виконавчому пакеті. Зазвичай зловмисники починають із соціальної інженерії, щоб зібрати інформацію про жертву та компанію, перш ніж формувати фішингове повідомлення, яке буде використано під час вейлінгової атаки[5].

Компрометація ділової електронної пошти. Окрім масових розподілених загальних фішингових кампаній, злочинці націлюються на ключових осіб у фінансових та бухгалтерських відділах через шахрайство з діловою електронною поштою та шахрайство з електронною поштою генерального директора. Видаючи себе за фінансових працівників та виконавчих директорів ці злочинці намагаються обдурити жертв, щоб вони ініціювали грошові перекази на несанкціоновані рахунки.

Як правило, зловмисники компрометують обліковий запис електронної пошти вищого керівника чи фінансового директора, використовуючи наявну інфекцію або через цілеспрямовану фішингову атаку. Зловмисник відстежує діяльність електронної пошти керівника протягом певного періоду, щоб дізнатись про процеси та процедури в компанії. Фактична атака набуває форми фальшивого електронного листа, який, схоже, надходить із скомпрометованого облікового запису виконавця, який надсилається комусь, хто є звичайним одержувачем. Електронний лист видається важливим і терміновим, і він просить одержувача надіслати банківський переказ на зовнішній або незнайомий банківський рахунок. Зрештою гроші потрапляють на банківський рахунок зловмисника[5].

Клонувальний фішинг. Клонувальний фішинг вимагає від зловмисника створити майже ідентичну копію легітимного повідомлення, щоб обдурити жертву, видаючи її за справжній лист. Електронне повідомлення надсилається з адреси, схожої на легітимного відправника, а тіло повідомлення виглядає так само, як і попереднє повідомлення. Єдина відмінність полягає в тому, що вкладення чи

посилання в повідомленні замінено на шкідливе. Зловмисник може сказати щось на зразок необхідності повторно надіслати оригінал або оновлену версію, щоб пояснити, чому жертва знову отримувала “те саме” повідомлення.

Ця атака базується на раніше побаченому, легітимному повідомленні, що робить більш імовірним, що користувачі потраплять під вплив атаки. Зловмисник, який уже заразив одного користувача, може використовувати цей прийом проти іншої особи, яка також отримала повідомлення, яке клонується. В іншому варіанті зловмисник може створити клонований веб-сайт із підробленим доменом, щоб обдурити жертву[5].

Вішинг (Vishing): Фішинг по телефону. Вішинг (англ. vishing) означає «голосовий фішинг» (англ. voice phishing), і він передбачає використання телефону. Як правило, жертва отримує дзвінок із голосовим повідомленням, замаскованим під комунікацію від фінансової установи. Наприклад, повідомлення може попросити одержувача зателефонувати за номером та ввести дані свого облікового запису або PIN-код для забезпечення безпеки чи інших офіційних цілей. Однак номер телефону дзвонить прямо зловмиснику через послугу передачі голосу за IP-адресою[5].

Смішинг (Smishing): фішинг за допомогою текстового повідомлення. Смішинг (англ. Smishing) - це поєднання слів "фішинг" (англ. phishing) та "SMS", останнє з яких означає протокол, що використовується більшістю служб обміну текстовими повідомленнями, - це кібератака, яка використовує оманливі текстові повідомлення для обману жертв. Мета полягає в тому, щоб обдурити вас, спонукати повірити, що надійшло повідомлення від довіреної особи або організації, а потім переконати вас вчинити дії, які надають зловмисникові інформацію, яку можна використати (наприклад, облікові дані для входу в клієнт-банк для маніпуляцій з банківським рахунком) або доступ до вашого мобільного пристрою.

Обсяг смішингу зростає, тому що люди частіше читають і відповідають на текстові повідомлення, ніж на електронну пошту: 98% текстових повідомлень читаються і на 45% повідомлень надаються відповіді, тоді як еквівалентні цифри для електронної пошти складають відповідно 20% та 6%. Користувачі часто менше стежать за підозрілими повідомленнями на своїх телефонах, ніж на своїх

комп’ютерах, а їх особисті пристрої, як правило, не мають такого захисту, який доступний на їх корпоративних ПК[5].

Короткострокове поширення шкідливих повідомлень. Фішинг з малим обсягом повідомлень з одного домену та адреси (snowshoeing) вимагає від зловмисників надсилання повідомлень з кількох доменів та IP-адрес. Кожна IP-адреса надсилає невеликий обсяг повідомлень, тому технології фільтрації спаму на основі репутації або обсягу не можуть відразу розпізнати та заблокувати шкідливі повідомлення. Деякі повідомлення надходять у скриньки електронної пошти, перш ніж фільтри навчаться їх блокувати.

Кампанії з швидким надсилання великої кількості повідомлень (hailstorm) працюють так само, як і фішинг з малим обсягом повідомлень з одного домену та адреси, за винятком того, що повідомлення розсилаються за надзвичайно короткий проміжок часу. Деякі з таких атак закінчуються в той час коли засоби боротьби зі спамом вловлюють та оновлюють фільтри для блокування майбутніх повідомлень, але зловмисники в цей час вже переходять до наступної фішингової кампанії[5].

1.1.2. Класифікація фішингових атак відповідно до застосованих методів

Зловмисники можуть надсилати фішингові повідомлення, щоб отримати доступ до систем жертв. Усі форми фішингу є електронною соціальною інженерією. Фішинг може бути цільовим. При цільовому фішингу зловмисник буде націлений на конкретну особу, компанію чи галузь. Більш загально, зловмисники можуть проводити нецільовий фішинг, наприклад у масових спам-кампаніях зловмисного програмного забезпечення.

Зловмисники можуть надсилати жертвам електронні листи, що містять шкідливі вкладення або посилання, як правило, для виконання шкідливого коду в системах жертв або для збору облікових даних для використання дійсних облікових записів. Фішинг також може проводитися через сторонні служби, такі як платформи соціальних медіа[2].

Фішинг з використанням шкідливих вкладень. Зловмисники можуть надсилати електронні листи з шкідливими вкладеннями, намагаючись отримати доступ до систем жертв. Шкідливе вкладення - це специфічний варіант фішингу. Фішинг з вкладеннями відрізняється від інших форм фішингу тим, що він використовує шкідливе програмне забезпечення, прикріплене до електронного листа. Усі форми фішингу - це соціальна інженерія, спрямована на особу, компанію чи галузь. У цьому випадку зловмисники прикріплюють файл до фішингової електронної пошти з приводом для відкриття вкладення і зазвичай покладаються на дії користувача, щоб файл був відкритий жертвою.

Існує багато варіантів вкладень, таких як документи Microsoft Office, виконувані файли, PDF-файли або заархівовані файли. Після відкриття вкладення (і, можливо, ігнорування попереджень про потенційну небезпеку), корисне навантаження зловмисника використовує вразливість або безпосередньо запускається в системі користувача. Текст електронного листа із приводом для відкривання вкладення зазвичай намагається вказати правдоподібну причину того, чому файл слід відкривати його і може пояснити, як обійти захист системи, щоб зробити це. Електронний лист може також містити вказівки щодо того, як розшифрувати вкладення, наприклад, пароль ZIP-файлу, щоб уникати засоби захисту електронної пошти. Зловмисники часто маніпулюють розширеннями файлів та піктограмами, щоб зробити прикріплені виконувані файли файлами документів, або файли, що використовують одну програму, здавалися файлами для іншої програми[7].

Фішинг з використанням шкідливих посилань. Зловмисники можуть надсилати електронні листи зі шкідливим посиланням, намагаючись отримати доступ до систем жертв. Фішинг із посиланням - це специфічний варіант фішингу. Він відрізняється від інших форм фішингу тим, що використовує посилання для завантаження шкідливого програмного забезпечення, що міститься в електронній пошті, замість того, щоб прикріплювати шкідливі файли до самого електронного листа, для уникання засобів захисту, які можуть перевіряти вкладення електронної пошти.

У цьому випадку шкідливі електронні листи містять посилання. Як правило, посилання будуть супроводжуватися текстом з використанням соціальної інженерії, і користувач повинен натискати або копіювати та вставляти URL-адресу в браузер, здійснюючи виконання шкідливого вмісту. Відвіданий веб-сайт може зламати веб-браузер, використовуючи експлойт, або користувачеві буде запропоновано завантажити програми, документи, zip-файли або навіть виконувані файли, в залежності від приводу електронної пошти. Зловмисники можуть також включати посилання, призначені для безпосередньої взаємодії з програмою зчитування електронної пошти, включаючи вбудовані зображення, призначені для безпосереднього використання кінцевою системою або перевірки отримання електронного листа (тобто веб-помилок/веб-маяків). Посилання можуть також спрямовувати користувачів на шкідливі програми, призначені для викрадення маркерів доступу до програм, як-от маркери OAuth, щоб отримати доступ до захищених програм та інформації[8].

Фішинг з використанням сервісів. Зловмисники можуть надсилати фішнгове повідомлення через сторонні служби, намагаючись отримати доступ до систем жертв. Фішинг через службу - це специфічний варіант фішингу. Він відрізняється від інших форм фішингу тим, що використовує сторонні послуги, а не безпосередньо корпоративні канали електронної пошти.

У випадку використання фішингу через сторонні служби зловмисники надсилають повідомлення через різні сервіси соціальних медіа, особисту веб-пошту та інші непідприємницькі служби. Ці служби, швидше за все, мають менш жорстку політику безпеки, ніж у підприємств. Як і у більшості видів фішингу, метою є створення взаємозв'язку з ціллю або якимось чином зацікавити ціль. Зловмисники створюватимуть підроблені облікові записи в соціальних мережах та запитуватимуть працівників про потенційні можливості роботи. Це дозволяє запитати про послуги, політики та програмне забезпечення, яке працює в середовищі з непідозрілою та правдоподібною причиною. Потім зловмисник може надсилати шкідливі посилання або вкладення через ці служби.

Поширеним прикладом є створення зв'язку з ціллю через соціальні медіа, а потім надсилання шкідливого вмісту в особисту службу веб-пошти, яку ціль використовує на своєму робочому комп'ютері. Це дозволяє зловмиснику обійти деякі обмеження електронної пошти на робочому обліковому записі, і ціль швидше відкриє файл, оскільки це те, на що вона очікувала. Якщо корисне навантаження не працює належним чином, зловмисник може продовжувати звичайний зв’язок та вирішувати проблеми з ціллю щодо того, як змусити його працювати[9].

1.2 Засоби захисту від фішингу

На сьогодні запропоновано досить багато рішень проти фішингу. Загалом, засоби протидії фішингу можна класифікувати на три основні категорії: засоби протидії фішингу на стороні сервера, засоби протидії фішингу на стороні браузера та засоби навчання персоналу.

1.2.1 Засоби протидії фішингу на стороні сервера

До засобів, що працюють на стороні сервера відносяться засоби які вимагають автентифікації сервера для захисту від фішингових атак. Прямий підхід полягає в тому, щоб клієнти перевіряли облікові дані, представлені веб-сервером. Облікові дані зазвичай видаються довіреною третьою стороною та надають впевненість в особі її пред'явника. Як правило, він відображається у вигляді логотипів, піктограм, печаток бренду у вікні браузера, щоб привернути увагу користувачів. Один із таких підходів намагається усунути проблему фішингу на стороні сервера, намагаючись запобігти фішинговим повідомленням електронною поштою, щоб вони не потрапили до потенційних жертв. Підходи на основі електронної пошти зазвичай використовують фільтри та аналіз вмісту.

Крім того, засоби захисту від спаму не завжди є повсюдно доступними. Якими б ефективними не були анти-фішингові рішення, але деякі фішингові електронні листи все одно можуть потрапити до потенційних жертв. Microsoft і Yahoo також визначили протоколи автентифікації електронної пошти (тобто Sender ID та DKIM), які можна використовувати для перевірки справжності отриманого електронного листа. Хоча протоколи автентифікації можуть вирішити проблему фішингу, більшість користувачів Інтернету в даний час їх не використовують. На жаль, широкомасштабне розгортання протоколів автентифікації вимагало б модифікації (або адаптації) існуючої інфраструктури.

Деякі антивірусні програми спеціально призначені для вирішення проблеми фішингових вкладень, оскільки фішинові атаки часто супроводжують віруси, вони можуть бути корисними для виявлення та блокування таких атак. Листи, що містять шкідливі вкладення можуть бути заблоковані такими рішеннями, що дозволяє попередити зараження комп'ютерів, яке може відбутися через відкриття шкідливого вкладення.

1.2.2 Засоби протидії фішингу на стороні браузера

Засоби протидії фішингу на стороні браузера, вбудовують антифішингову логіку у модулі веб-браузерів. Браузери регулюють візуальну поведінку веб-сторінок, щоб запобігти обману. На основі підходів, що використовуються в існуючих засобах боротьби з фішингом у вигляді плагінів, можна виділити три категорії: підходи на основі чорного списку, підходи на основі веб-сторінок, підходи на основі потоків інформації.

Підходи, засновані на чорному списку. Найпопулярніші та широко застосовувані методи боротьби з фішингом засновані на використанні чорних списків фішингових доменів. Недоліком підходу є те, що фішингові сайти, що не внесені в чорний список, не розпізнаються, нові фішинг-сайти, яких немає в базі даних, можуть бути не розпізнані. Основна проблема чорного списку полягає в

тому, що антифішингові організації опиняються в гонці проти зловмисників. На жаль, завжди є вікно вразливості, під час якого користувачі сприйнятливі до атак. Крім того, підходи є настільки ж ефективними, як і якість підтримуваних списків.

Підходи на основі веб-сторінок. Відоме рішення в літературі - SpoofGuard. SpoofGuard використовує перевірку доменних імен, URL-адрес, посилань та зображень, щоб оцінити ймовірність того, що дана сторінка є частиною підступної атаки. Наприклад, сторінка з підозрілою URL-адресою, такою як etrade-maintenance.suspicious.org або www.etrade.com@129.170.213.101/maintainance.aspand з логотипом E*Trade, матиме більший індекс підробки, ніж сторінка, на якій немає жодної з цих характеристик. SpoofGuard також використовує історію, наприклад, чи відвідував користувач цей домен раніше, і чи посилальна сторінка була з веб-сайту електронної пошти, такого як Hotmail або Yahoo! Mail. Найголовніше, що SpoofGuard перехоплює та оцінює дописи користувачів з урахуванням відповідної історії та підробленого індексу сторінки. SpoofGuard перевіряє поля імені користувача та пароля та порівнює розміщені дані з раніше введеними паролями з різних доменів. Цей механізм застерігає користувача від надсилання пароля E*Trade на сайт із логотипом E*Trade, але за межами домену etrade.com. Спочатку механізм розкладає веб-сторінки на помітні блоки відповідно до «візуальних підказок».

Підходи, засновані на потоці інформації. PwdHash - це рішення проти фішингу. Він створює специфічні для домену паролі, які стають марними, якщо вони надсилаються в інший домен (наприклад, пароль для www.hotmail.com буде іншим, якщо буде надісланий на www.attacker.com). Для порівняння, AntiPhish застосовує інший підхід і відстежує місце подання конфіденційної інформації. Тобто, якщо він виявляє, що конфіденційна інформація, така як пароль, вводиться у форму на підробленому веб-сайті, генерується попередження та очікувана операція скасовується. Основним недоліком AntiPhish є те, що він вимагає взаємодії користувача, щоб вказати, яку конфіденційну інформацію слід збирати та контролювати.

1.2.3 Засоби навчання персоналу

Навчання користувачів з питань фішингу зосереджено на навчальних матеріалах в Інтернеті, тестуванні та навчанні на місцях. Навчальні матеріали в мережі Інтернет публікують державні організації, некомерційні організації та підприємства. Ці матеріали пояснюють, що таке фішинг, та містять поради, щоб запобігти впливу фішингових атак на користувачів. Тестування використовується, щоб продемонструвати, наскільки сприйнятливі люди до фішингових атак, та навчити їх, як їх уникнути.

1.3 Аналіз методів захисту від фішингу

1.3.1 Машинне навчання

Цей тип фокусується на застосуванні методів машинного навчання та інтелектуального аналізу даних для виявлення фішингу. Пов’язані з цим методи класифікуються на три основні категорії: класифікація, кластеризація та виявлення аномалій.

Методи класифікації. Методи класифікації намагаються зіставити вхідні дані (ознаки чи змінні) з бажаними результатами (відповіддю) за допомогою певної функції. У разі класифікації фішингових електронних листів створюється модель для класифікації електронного листа як фішингового або легітимного шляхом вивчення певних характеристик електронного листа. Контрзаходи, засновані на класифікації, покладаються на використання маркованих наборів даних фішингу та легітимних екземплярів (наприклад, електронної пошти чи веб-сторінок). Навчальна модель m засвоює зразки $t = |f_1, \dots, f_n|$ з навчальних даних, використовуючи вектор відповідних ознак, який складається із вмісту та/або на основі URL-адрес. Деякі показники якості використовуються для оцінки класифікаційних характеристик навченої моделі m на випробувальних зразках e. Більшість моделей виявлення

фішингу застосовують статистичні класифікатори, які використовують функцію $f(t,\gamma)$ для класифікації екземплярів e таким чином, щоб визнати взаємозв'язок між t і e, використовуючи деякі критерії оптимізації, де γ є вектором регульованих параметрів. Значення γ визначаються з використанням обраних критеріїв оптимізації. На основі типів функцій, які використовуються для виявлення фішингу, класифікатори фішингу можна згрупувати у три основні категорії:

- Класифікатори на основі властивостей URL-адреси, таких як доменне ім'я, характеристики IP-адреси та географічні властивості. Функції, пов'язані з URL-адресами, використовувались як вхідні дані для кількох методів класифікації для виявлення фішингу, таких як метод опорних векторів, наївний баєсів класифікатор та метод k-найближчих сусідів. Серед них метод k-найближчих сусідів дає найкращу точність[10].
- Класифікатори на основі текстових ознак. Ці підходи вивчають вміст підозрілого матеріалу, щоб визначити, чи є він легітимним чи фішинговим. Наприклад, виявлення фішингу на веб-сайті може оперувати властивостями, витягнутими з текстового вмісту головної сторінки, файлів, її компонентів та структури DOM[11].
- Класифікатори на основі гібридних ознак. Кілька класифікаторів побудовані на гібридних характеристиках, які витягуються із вмісту та URL-адрес на веб-сторінках для виявлення фішингу. Деякі методи цієї категорії зосереджені на створенні динамічних, адаптивних або ансамблевих класифікаторів. Порівняно зі статичними класифікаторами, динамічні класифікатори орієнтовані на адаптацію правил класифікації. Деякі класифікатори використовували метод опорних векторів онлайн, який використовує теорію ігор та попередні знання для створення класифікатора виявлення фішингу. Подібна класифікація на основі адаптивної теми запропонована для виявлення фішингу в середовищі електронної пошти. Більшість підходів до класифікації застосовано до виявлення фішингу на веб-сайтах, а деякі - до електронних листів та голосу за допомогою гауссової суміші[12].

Методи кластеризації. Кластерні методи розділяють набір екземплярів на фішинг та легітимні кластери. Метою кластеризації є групування об’єктів на основі їх подібності. Якщо кожен об’єкт представлений як вузол, і подібність між різними об’єктами вимірюється на основі їх загальних спільних ознак, тоді алгоритм кластеризації може бути використаний для ідентифікації груп (вузлів) подібних спостережень. Кількість груп можна вибрати так, щоб вузли в одній групі мали більшу схожість, ніж вузли в різних групах. Інформація про таку структуру кластеризації, в свою чергу, використовується для призначення нових об'єктів правильному кластеру. Нові об'єкти призначаються кластеру на основі їх подібності з іншими аналізованими екземплярами.

Можна припустити, що $p = |p_1, \dots, p_n|$ представляє набір веб-сторінок, де кожна сторінка p_i представлена у вигляді елемента вектору $(f_{i_1}, \dots, f_{i_m})$, в якому f_{i_j} є або об’ємом або властивістю URL-адреси. Мета кластеризації - створити структуру, яка найкраще відокремлює фішинг від легітимних сторінок, а потім використовувати таку структуру для кластеризації нових сторінок. Двома важливими компонентами методу кластеризації є показник подібності (відстані) між двома вибірками даних (наприклад, сторінками p_i, p_j) та алгоритмом кластеризації. Різні показники подібності / відстані можуть призвести до різних результатів кластеризації. Знання домену можуть бути використані для керівництва формулюванням міри подібності/відстані. Для даних з великими розмірами метрика Мінковського є популярною:

$$d(p_i, p_j) = (\sum_{k=1}^{m} |p_{i,m} - p_{j,m}|^d)^{\frac{1}{d}}$$

де m – це розмірність даних.

Для виявлення фішингу було використано кілька алгоритмів кластеризації, таких як DBscan, метод k-середніх та самоорганізаційні карти Кохонена. DBscan був використаний для виявлення фішингових цілей шляхом кластеризації набору веб-сторінок, що складається з даної веб-сторінки p_i та всіх пов'язаних з нею веб-сторінок. Взаємозв'язки між веб-сторінками p_i та пов'язаними з ними веб-сторінками

визначаються на основі посилань, ранжирування, схожості тексту та схожості макета веб-сторінки, які використовуються як вхідні функції для кластеризації. Метод кластеризації має на меті виявити кластер, сформований навколо p_i, щоб визначити p_i як фішинг, що, у свою чергу, спричинить процес виявлення легітимної веб-сторінки p_l, на яку здійснюють атаку шляхом створення фальшивої версії p_i. В іншому випадку сторінка визначається як легітимна. Як і класифікація, методи боротьби з фішингом, засновані на кластеризації, залучають різні функції введення та засоби комунікації. Окрім функцій, заснованих на URL-адресах та функцій вмісту, кластеризація фішингу також включає властивості, вилучені із зображень веб-сайтів. Кластеризація застосовується при виявленні атак у декількох комунікаційних носіях, таких як фішинг за допомогою електронних листів , підроблених веб-сайтів та фішингу за допомогою голосової комунікації [13, 14].

Виявлення аномалій. Аномалія - це модель даних, яка не узгоджується зі схемами нормальної поведінки. Підходи до виявлення фішингу, засновані на аномаліях, по суті розглядають спроби фішингу як непередбачувані. Кожен веб-сайт має унікальну ідентичність у віртуальному просторі явно чи неявно. Коли фішинговий сайт зловмисно заявляє про помилкову ідентичність, він завжди демонструє ненормальну поведінку порівняно із легітимним сайтом, що виявляється в деяких об'єктах DOM на веб-сторінках та транзакціях HTTP. Алгоритми виявлення аномалій виявляють фішингові веб-сайти, фіксуючи ці аномалії. Методи виявлення аномалій присвоюють оцінку підозрілому матеріалу, що аналізується, порівнюючи ознаки фішингового матеріалу з характеристиками одного чи кількох найближчих сусідів. Якщо показник аномалії перевищує граничну точку, веб-сторінка буде класифікована як фішинг[15].

Альтернативно, однокласне виявлення аномалії передбачає, що всі навчальні зразки належать до одного класу (тобто легітимної електронної пошти). Відповідно, це створює помітну границю навколо випадків, які відповідають легітимному класу. Для виявлення фішингу застосовується однокласний SVM . Він розглядає джерело як єдиного члена другого класу (потенційний фішинговий лист). Якщо $e_1, e_2, \ldots, e_n$ - це навчальні електронні листи, які належать до легітимного класу E, де E -

компактна підмножина R^E, то $\Phi: E \to H$ це відображення ядра, яке перетворює властивості електронної пошти в E у простір H. Для того, щоб відокремити набір даних від джерела, потрібно вирішити проблему квадратичного програмування. Параметри рішення встановлюють верхню межу на частку фішингових листів і нижню межу на кількість тренувань із легітимними електронними листами, що використовуються як опорний вектор[16].

Більшість підходів до виявлення аномалій для виявлення фішингу на підроблених веб-сайтах зосереджуються на виявленні аномальних ознак, які, швидше за все, присутні в URL-адресах. Деякі моделі видобувають властивості, засновані на регулярності шаблонів надсилання повідомлень, часу між миттєвими повідомленнями, шаблонах аномалій в URL-адресах та тимчасовій регулярності поведінки відправника для виявлення фішингу. Класифікація є домінуючим методом виявлення фішингу, а моделі класифікації, як правило, черпають ознаки із вмісту та/або URL-адрес веб-сторінок або електронних листів. Виявлення фішингу в електронних листах здебільшого покладається на функції, що базуються на вмісті, а на веб-сайтах - на функціях на основі URL. Порівняно невелика кількість досліджень застосовувала методи кластеризації до фішингу, але тим не менш, деякі функції, такі як зображення та голос, досі досліджувались лише у кластеризації. Крім того, для виявлення фішингу в чаті, а також веб-сайтах та електронних листах використовувались методи виявлення аномалій.[17]

1.3.2 Добування тексту

Добування тексту стосується використання методів добування даних та машинного навчання для виявлення тенденцій, закономірностей або корисних знань із тексту. Добування тексту визначає спроби фішингу, аналізуючи шаблони підозрілих матеріалів, які включають, але не обмежуються, вмістом електронних листів, веб-сайтів, URL-адрес, миттєвих повідомлень та SMS. Для виявлення фішингу застосовано такі типи методів видобування тексту: частота терміну зі

зворотною частотою документа (TF-IDF), регулярні вирази (RE) та латентно-семантичний аналіз та тематичні моделі. Хоча такі методи можна згрупувати за категорією машинного навчання, вони зазвичай використовуються як етапи попередньої та/або подальшої обробки при створенні інших рішень для виявлення фішингу. По суті, схема зважування TF-IDF виявляє вагу слова в наборі документів, знаходячи його відносну частоту в одному документі порівняно з його оберненою пропорцією щодо набору документів, на який посилаються. TF-IDF інтуїтивно визначає вагу даного терміну щодо певного документа (наприклад, веб-сторінки або електронного листа). Наприклад, терміни, які зазвичай використовуються в підроблених електронних листах, як правило, мають вищі ваги TF-IDF, ніж їхні легітимні аналоги. Враховуючи набір електронних листів $E = \{e_1, \dots, e_n\}$, та термінів $T = \{t_1, \dots, t_n\}$ вага TF-IDF терміна t_i в електронному листі e_j обчислюється наступним чином:

$$w_{t_i,i_j} = f_{t_i,i_j} \times \log\left(\frac{|E|}{f_{t_iE}}\right)$$

де f_{t_i,i_j}- кількість випадків, коли термін t_i зустрічається в електронному листі e_j, $|E|$ - це загальна кількість електронних листів, що позначає кількість електронних листів, що містять термін t_i. Підхід TF-IDF в основному використовується для веб-сайтів. Існує методика під назвою CANTINA, яка використовує TF-IDF замість URL-адрес та доменних імен для виявлення спроб фішингу. Деякі моделі використовують TF-IDF та пошукові системи для виявлення фактичного домену сторінки, аналізуючи особливості її оголошеної ідентичності. Витягнуті властивості заявленого домену використовуються для запуску запиту в пошукових системах. Якщо запит отримує результати - дві ідентичності вважатимуться подібними, і відповідно, сторінки будуть класифіковані як легітимні. Цей напрямок досліджень продовжується шляхом використання інших властивостей сторінки для визначення її ідентичності, таких як властивості, вилучені зі структури DOM, вузли елементів, що представляють назву бренда сайту та ключові слова сторінки[18].

Регулярні вирази забезпечують гнучкі засоби для узгодження рядків тексту. При виявленні фішингу регулярні вирази використовувались для створення

шаблонів фішингових URL-адрес із існуючих сторінок. Ці шаблони в свою чергу можуть бути використані для виявлення нових фішингових URL-адрес. Регулярні вирази є корисними для створення баз даних чорного списку та для обробки частих незначних змін у шаблонах фішингу. LSA покладається на виявлення прихованих взаємозв'язків між ключовими словами, такими як синоніми та омоніми, а отже є корисним для виявлення споріднених слів в тому ж контексті. Моделі LSA та тем використовувались у багатьох додатках для аналізу тексту. LSA базується на принципі, згідно з яким слова, що вживаються в одному контексті, мають подібне значення. Моделювання тем розглядає документи як суміш прихованих тем, а теми, у свою чергу, представляються як розподіл ймовірностей за словами у наборі даних навчального матеріалу. Такі теми використовуються як функції при виявленні фішингу на основі класифікації[19].

1.3.3 Користувачі

Контрзаходи, що стосуються людей та поведінкові фактори, що характеризують тих, хто потрапляє у фішинг, дуже важливі для запобігання фішинговим атакам. Дослідження користувачів, спрямовані на вимірювання реакції користувачів на фішингові матеріали, враховували поведінкові фактори, а також демографічні фактори. Ці дослідження зазвичай залучають користувачів для ідентифікації фішингу. Більшість досліджень користувачів проводяться із застосуванням антифішингових систем; тим не менш, деякі з них забезпечують механізми або для підвищення обізнаності користувачів, коли стикаються з фішинговими атаками, або залучають їх до виявлення фішингу[20].

Підвищення обізнаності користувачів. Недостатній рівень обізнаності щодо безпеки може бути використаний зловмисниками, щоб обдурити жертв. Встановлено, що різноманітні фактори впливають на обізнаність людини про безпеку, включаючи фактори досвіду, такі як знання безпеки, досвід використання веб, ефективність використання комп'ютера та такі фактори як довірливість

користувача і сприйнятий ризик. Можна припустити, що такі фактори досвіду, як знання безпеки, веб-досвід та ефективність використання комп'ютера знижують ймовірність людини бути ошуканою фішинговими електронними листами. Існує два основних підходи до підвищення рівня обізнаності щодо безпеки: навчання та тренування користувачів та тестування IQ[21].

Навчання та тренування здійснюється шляхом навчання користувачів, як виявляти фішингові атаки під час регулярних дій у своїх електронних системах, або уникнення стати жертвою фішингу. Однією з форм такого навчання є надсилання користувачам певних повідомлень про безпеку щодо фішингових атак. Навчання, вбудоване в електронні листи з текстовими та графічними примітками про фішинг, є більш ефективним, ніж традиційне сповіщення про безпеку, що надсилається користувачам. Користувацькі підходи до виявлення фішингу в основному застосовуються до середовищ електронної пошти та веб-сайтів. З огляду на обмеження безпеки мобільного середовища, мобільним додаткам не вистачає захищених індикаторів ідентифікації (наприклад, інформація про сертифікат, піктограми блокування та вибір шифру). Більше того, зловмисники можуть пов'язувати мобільні програми з підробленим вмістом або підробленими веб-сайтами, що збільшує проблему для користувачів при розпізнанні фальшивих та легітимних URL-адрес. Враховуючи відсутність технічних рішень проблем фішингу на мобільних пристроях, підвищення обізнаності зацікавлених сторін стає ще більш важливим для виявлення фішингу на цих пристроях. Подібним чином покращення обізнаності користувачів також рекомендується для запобігання фішингу на основі голосових комунікацій. Навчання користувачів розпізнаванню фішингових атак включає також використання методів PHaaS (Фішинг як послуга), при якому організації імітують реальні сценарії фішингу своїх користувачів для відстеження сприйнятливості до фішингу в експериментальному безпечному середовищі. Основна мета цих експериментів - зрозуміти, наскільки організація сприйнятлива до фішингу, та підвищити обізнаність щодо фішингових атак.

Тестам на IQ зазвичай передують надання користувачам навчальних матеріалів про фішинг у конкретному контексті. Ці тести розроблені на основі

відомих служб, якими користується група користувачів, виключаючи елемент недосвідчених послуг. Одне з досліджень представляє стратегію навчання на основі підсвідомості в галузі цільового фішингу на основі IQ. Запропонована техніка показує користувачам як легітимні електронні листи, так і фішингові, і пропонує користувачам класифікувати електронні листи. Метод допомагає користувачам розпізнавати фішинг та зосереджуватись на важливих функціях при отриманні підозрілих електронних листів.

Залучення користувачів до виявлення фішингового матеріалу. Завдяки участі в ідентифікації легітимних та фішингових матеріалів користувачі, як очікується, можуть вручну ідентифікувати нові спроби фішингу. Крім того, досвідчені користувачі, можуть навіть брати участь у створенні фішингових наборів даних, які також сприяють виявленню на основі голосування користувачів.

Ручна автентифікація. Цей підхід сповіщає користувачів про виявлення підозрілої інформації про ознаки фішингу. Один з підходів представляє шлях електронної пошти на географічній карті, використовуючи інформацію із заголовків електронної пошти. Цей підхід інформує користувачів про шлях повідомлення, сценарій, коли відправник електронної пошти стверджує, що електронне повідомлення надіслано від довіреної організації, але фактична IP-адреса може не підтверджувати таке твердження. Участь користувача у виявленні фішингових атак має дві основні переваги: 1) вона ефективно сприяє підвищенню рівня обізнаності людини про фішинг та 2) корисна при створенні фішингових наборів даних шляхом голосування користувачів на підозрілих сторінках[22].

Голосування користувачів. Phish tank - це найпопулярніша база даних про фішингові веб-сайти, про які повідомляється. База даних пропонує систему перевірки фішингу на основі спільноти, де користувачі подають підозрілі матеріали, а інші користувачі "голосують" за те, чи є такі подання фішинговими чи легітимними. Подібним чином була розроблена техніка виявлення фішингу, що покладається на навчених учасників, які голосують за підозрілі URL-адреси.

1.3.4 Відповідність профілів

Методи щодо відповідності профілів використовують інформацію про доменне ім’я, URL-адреси доменів, до яких нещодавно отримували доступ користувачі, їх облікові дані в цих доменах та інші характеристики доменів, до яких здійснюється доступ (наприклад, макет та зображення), для створення профілів на основі властивостей та використання їх для виявлення фішингу. Компоненти зіставлення профілів можуть бути простими (наприклад, відповідність URL-адресам) або включати складні методи (наприклад, відповідність зображень). Попередження браузера та більшість фішинг-панелей підпадають під цю категорію. Доступно кілька інструментів для децентралізованого управління профілями користувачів та консолідації їхніх ідентичностей (наприклад, OpenID, SAML Liberty Alliance, WS Microsoft). Методи, які спираються на стратегію відповідності профілів, можна згрупувати за чотирма категоріями: відповідність історії використання, відповідність шаблонів, візуальна та структурна відповідність та відповідність білого та чорного списків[23].

Відповідність історії використання. Профілі користувачів зберігають інформацію про ресурс та аутентифікацію користувача, що використовується для кожного ресурсу. Коли така інформація запитується на конкретному ресурсі, який претендує на те, що він є одним із легітимних, що зберігаються в профілі користувача, антифішинговий компонент використовує інформацію, що зберігається в профілі, для виявлення спроби фішингу. Цей тип підходу, який в основному застосовується для виявлення фішингу на веб-сайтах, складається з двох підкатегорій. Перша категорія розробляє засоби розширення браузера для відстеження активності користувачів в мережі, таких як його облікові дані та веб-сторінки, які він відвідував. Ці інструменти генерують попередження кожного разу, коли користувач намагається передати інформацію ненадійним шляхом на основі історично відстеженої інформації. Друга категорія вимагає від користувачів створення своїх профілів вручну, які, в свою чергу, будуть використовуватися для виявлення фішингу.

Відповідність шаблонів: Замість запису інформації про діяльність користувачів цей тип підходів створює профілі про інші сутності (наприклад, легітимні веб-сторінки, легітимні шаблони електронної пошти). Наприклад, плагін браузера Spoof Guard екранує сторінки, що вимагають облікові дані користувача, перевіряючи історію перегляду користувачів. Якщо користувач вводить свої збережені дані на невідомій цільовій сторінці, оцінка аномалії обчислюється за допомогою процедури збігу шаблонів. На основі оцінки сторінку класифікують як фішингову або легітимну. Методи узгодження зразків також використовувались для виявлення клонованих профілів у соціальних мережах.

Візуальне узгодження: візуальна схожість обчислюється на основі візуальних аспектів веб-інтерфейсів, таких як зображення, блоки та макет, щоб розрізняти фішинг та легітимні сторінки. Кілька підходів вводять заходи візуальної схожості для виявлення фішингових атак, такі як візуальна схожість на основі сегментації, схожість дерева DOM, яка виявляє фішингові веб-сторінки, порівнюючи легітимні та підозрілі сторінки на основі схожості графів, списку схожості символів Unicode та вимірювання гістограми, яка вилучає ключові особливості узгодження зразків у реальному часі. Деякі підходи візуального узгодження використовують більше одного типу міри подібності, наприклад, подібність сторінок на рівні блоку, макет та загальну схожість при порівнянні веб-сторінок, фрагменти тексту, стиль веб-сторінки та зображення, вбудовані в сторінки.

Відповідність білого та чорного списків: Цей тип контрзаходів робить акцент на створенні бази даних відомих довірених та підозрілих доменів. Як тільки аномалії виявляються за допомогою техніки фільтрації доменів, можна здійснити зіставлення з чорним та/або білим списком. Встановлення відповідності білого та чорного списків є одним із найефективніших підходів до виявлення фішингу. Чорні списки браузерів є основним механізмом захисту від фішингових атак. Google надає службу безпечного перегляду, яка дозволяє клієнтській програмі перевіряти підозрілі URL-адреси на основі постійно оновлюваних списків підозрілих сайтів. Виходячи з того, як створюються чорні списки, існуючі браузери можна класифікувати на три категорії:

1. Браузери, які використовують безпечний перегляд Google, такі як Chrome, Firefox та Safari.

2. Браузери, які використовують власні чорні списки, такі як Internet Explorer та Edge, які використовують SmartScreen - власний чорний список компанії Microsoft.

3. Браузери, які складають чорні списки, використовуючи сторонні засоби. Наприклад, Opera використовує чорні списки Phishtank та Netcraft для створення власного списку підозрілих URL-адрес.

Більшість підходів з чорними списками не виявилися ефективними для обробки фішингу нульового дня/години. Існує алгоритм фільтрації чорного списку, який можна застосувати до проксі-сервера без накладних витрат. Ідея полягає в очищенні проксі-системи шляхом блокування всіх частин вмісту веб-сторінок, що містять шкідливий код, включаючи форми імені користувача та пароля. Одне обмеження цього підходу полягає в тому, що він вимагає зусиль для підтримання чорного списку. Інший полягає у накладних витратах на продуктивність, що виникають, коли веб-форми блокуються на підозрілих сторінках.

1.3.5 Браузери, які використовують безпечний перегляд Google, такі як Chrome, Firefox Та Safari

До даної категорії методів боротьби з фішингом, віднесено: онтологію, Honeypots («Пастки»), пошукові системи та клієнт-серверну аутентифікацію.

Онтологія. Онтологія моделює набір понять у певній галузі, а також семантичні асоціації між цими поняттями. Нові терміни, фрази або вирази, що використовуються у фішингових повідомленнях, можна визначити, моделюючи їх як поняття та семантичні взаємозв'язки в онтології. Спроби фішингу стають складними. Зокрема, текстовий вміст, який використовується для ініціалізації атак, змінюється, що ускладнює їх класифікацію за допомогою звичайних методів боротьби з фішингом. Наприклад, зловмисники зазвичай змінюють фішинговий вміст електронної пошти, щоб уникнути виявлення, коли стикаються із звичайними

контрзаходами на основі вмісту. Однак, якщо семантичні взаємозв'язки між поняттями визначені належним чином, ймовірність виявлення нових форм фішингових електронних листів може зрости. Онтологічна семантика може покращити розуміння природної мови, виявляючи засновані на значенні підказки, що вказують на фішинг та докази фішингу. На сьогоднішній день дуже мало методів боротьби з фішингом включають онтологію. Одне дослідження пропонує підхід на основі онтології для підвищення точності методів боротьби з фішингом на основі класифікаторів. Спочатку метод витягує функції з електронного повідомлення, аналізуючи його текст, і якщо витягнуті об’єкти відповідають характеристикам відомих фішингових листів, електронне повідомлення передається для онтологічного аналізу, який потім включає набір пов’язаних понять у процес виявлення. Відповідно до цього, можливим є створення системи подання знань для розмежування декількох видів шахрайства, включаючи фішинг[24].

Honeypots (Пастки) Honeypots (Пастки) - це захисні пристрої, цінність яких полягає в тому, що їх досліджують та компрометують. Пастки, як правило, налаштовані на збір підозрілих даних. Вони налаштовані на збір даних про зловмисників, створення баз даних чорного списку зловмисників та/або блокування підозрілих доменів. Для протидії фішинговим атакам було запропоновано декілька систем, заснованих на методі "пасток". Ключова ідея таких підходів полягає в тому, щоб активно забезпечувати зловмисників хибними даними, які, здаються даними автентифікації (наприклад, відбитки пальців). Хибні дані можуть існувати практично в будь-якій формі, від мертвого, підробленого облікового запису до запису в базі даних, який буде обраний лише за допомогою зловмисних запитів - будь-яке їх використання за своєю суттю є підозрілим, якщо не є зловмисним. Іншим прикладом хибних даних є підроблена електронна адреса, яка використовується для відстеження того, чи вкрадено список розсилки. Підходи, засновані на хибних даних, можуть допомогти відстежувати фішингову діяльність, яка ініціює зупинку сайту, і таким чином стати популярним активним фішинговим контрзаходом. HoneyBuddy - це підхід для виявлення підозрілої діяльності. Метод виявляє контакти та включає їх у свої месенджери, подаючи запити пошуковим

системам для виявлення нових контактів та розширення бази даних. Крім того, він може використовувати сайти пошуку контактів для пошуку нових потенційних жертв обміну миттєвими повідомленнями. Одне з обмежень підходів з пастками полягає в легкості їх виявлення зловмисниками. Таким чином, основною проблемою такого типу підходу є продовження терміну служби хибних даних[25].

Пошукові системи. Пошукові системи поєднуються з іншими методами виявлення фішингу. Як правило, якщо сторінка є легітимною, її слід проіндексувати та присвоїти рейтинг пошуковою системою. На відміну від них, фішингові домени не користуються популярністю, і відповідно їх рейтинг у пошуковій системі, як правило, дуже низький. Ще гірше те, що більшість фішингових доменів не індексуються пошуковими системами. Один з підходів використовує пошукові системи для перевірки URL-адрес, розміщених на сторінках соціальних мереж. Евристичний підхід розробляється шляхом аналізу повідомлень в Facebook, які містять URL-адреси. Виділено кілька функцій, щоб відрізнити дійсні URL-адреси від фішингу, наприклад, кількість тире в імені хосту, існування доменного імені при запиті в пошуковій системі та вік домену. Рекомендується використовувати результати рейтингу в пошукових системах як вхідні дані для створення фішингових класифікаторів - техніки, яка, як стверджується, є ефективною у зменшенні помилкових спрацьовувань[26].

Клієнт-серверна автентифікація. Клієнт-серверна автентифікація покладається на взаємну автентифікацію між клієнтами та серверами. Ключі сайту, довірені пристрої, схема підпису на основі ідентифікації, динамічний індивідуальний інтерфейс та аутентифікація на основі ідентифікатора каналу є основними методами автентифікації, що використовуються для виявлення та запобігання фішинговим атакам. В інтерактивній перевірці ключів сайту користувачеві потрібно виконати лише одне графічне співставлення для автентифікації зображень, які він обрав для певного сайту. Цей підхід має переваги у своїй простоті та надійності. Довірений пристрій (наприклад, смарт-телефон) може використовуватися для здійснення взаємної автентифікації. Цей підхід не тільки зменшує залежність від користувачів під час процесу перевірки, але також

може запобігти іншим типам атак, таким як атаки «людина посередині». Подібним чином, схема підписів, що базується на особі, була використана, щоб зробити спілкування електронною поштою надійним. На відміну від типових цифрових підписів, цей підхід не вимагає заздалегідь створеної інфраструктури відкритих ключів. Також не потрібна співпраця між доменами електронної пошти. Натомість кожен домен електронної пошти є незалежним, і контролюючий орган на основі ідентифікації видаватиме ключі. Крім того, головні відкриті ключі, що відповідають кожному домену, повинні бути розподілені та сертифіковані[27].

На відміну від типових пар ключів, секретні ключі на основі ідентифікації обчислюються контролюючим органом, а потім надсилаються користувачам. Після відправки ключів використовується підписна схема на основі групи, яка дозволяє відправникам ініціювати підпис для повідомлення за допомогою обраної групи підписантів. Ця схема автентифікації вимагає, щоб відправник був частиною групи, а відкриті ключі інших членів були доступними. Будь-яка людина в групі може підтвердити, що підпис обчислюється з розкриттям особи підписувача. Отже, сам відправник та одержувачі його повідомлення повинні бути в одній групі. Динамічний індивідуальний інтерфейс - це ще один підхід, який лише просить користувачів розпізнати зображення, сформоване сервером, замість будь-яких статичних показників безпеки, якими спільно користується сервер. Кілька методів автентифікації використовують протоколи транспортного рівня (TLS) і SSL, щоб забезпечити певну впевненість у тому, що користувач є легітимним, а не шахрайським веб-сайтом. SSL і TSL базуються на криптографії відкритих ключів. Під час процесу автентифікації клієнт TLS/SSL надсилає повідомлення на сервер TLS/SSL. В результаті сервер пройде автентифікацію клієнта. Потім здійснюється обмін ключами автентифікації між сервером і клієнтом. Після обміну ключами та завершення перевірки можна встановити захищений зв’язок між клієнтом та сервером. Аутентифікація на основі ідентифікатора каналу була розроблена, щоб зірвати тип атак "людина посередині". Коли користувач намагається вперше увійти до свого облікового запису з браузера, веб-сервер вимагає від користувача самостійної автентифікації, використовуючи пристрій двохфакторної

автентифікації, як у телефонній автентифікації та протоколах Universal 2nd Factor (U2F).

Як частина протоколу автентифікації, двохфакторний пристрій порівнює ідентифікатор каналу браузера з ідентифікатором з'єднання TLS, якому свідчить сервер. Якщо вони рівні, то браузер безпосередньо підключений до веб-сервера; інакше всередині здійснюється атака, і пристрій скасовує протокол автентифікації, щоб зупинити атаку. Сервер може створити файл cookie, прив'язаний до каналу, щоб захистити подальшу взаємодію із сервером від цього браузера. Існують інші категорії механізмів автентифікації, такі як автентифікація електронної пошти. Microsoft застосовує фільтр SenderID, протокол автентифікації електронної пошти, для вирішення проблеми підробки домену.

1.4 Постановка задачі

Об’єктом дослідження в даній роботі є процес класифікації фішингових вкладень за допомогою інтелектуальних моделей.

Дана задача потребує виконання різноманітних робіт, які орієнтовані на покращення результатів виявлення фішингових вкладень для попередження фішингових атак.

Для досягнення цілі даної роботи необхідно виконати наступні завдання:

- збір даних про фішингові вкладення, які використовуються при здійсненні фішингових атак;
- компонування даних та приведення їх до узагальненого вигляду для подальшого їх аналізу;
- розробка моделі аналізу даних;
- аналіз даних та розробка моделі класифікації фішингових вкладень;
- навчання та перевірка адекватності моделі класифікації фішингових вкладень;
- розробка моделі системи захисту від фішингових HTML-вкладень;

- перевірка адекватності роботи розробленої моделі;

Висновки за розділом 1

В даному розділі було надано визначення фішингу. Надано класифікацію фішингових атак відповідно до спрямування атаки, в межах якої виділено масову фішингову розсилку, цільовий фішинг, вейлінг (whaling), компрометацію ділової електронної пошти, клонувальний фішинг, вішинг (vishing) або ж фішинг по телефону, смішинг (smishing) або ж фішинг за допомогою текстового повідомлення та короткострокове поширення шкідливих повідомлень.

Також в даному розділі було описано та прокласифіковано фішингові атаки відповідно до застосованих методів: фішинг з використанням шкідливих вкладень, фішинг з використанням шкідливих посилань, фішинг з використанням сервісів.

В цьому розділі було виконано аналіз методів захисту від фішингу та описано принцип їх дії.

В межах даного розділу було розділено методи захисту від фішингу на 5 категорій, серед яких виділено:

- Машинне навчання, яке в свої чергу розділено на:

a) Методи класифікації
b) Методи кластеризації
c) Виявлення аномалій

- Добування тексту
- Користувачі, що в свою чергу розділено на:

a) Підвищення обізнаності користувачів
b) Залучення користувачів до виявлення фішингового матеріалу

- Відповідність профілів
- Інші методи, що в свою чергу розділено на:

a) Онтологію

b) Honeypots (Пастки)

c) Пошукові системи

d) Клієнт-серверна автентифікація

В кінці даного розділу визначено основні завдання, які необхідно виконати для досягнення мети даної дипломної роботи.

Таким чином в даному розділі було виконано перший етап розробки моделі захисту від фішингу.

РОЗДІЛ 2
АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ. РОЗРОБКА МОДЕЛІ КЛАСИФІКАЦІЇ ФІШИНГОВИХ ВКЛАДЕНЬ

2.1 Опис понять предметної області та теоретичних основ предметної області

2.1.1 Машинне навчання

Машинне навчання (machine learning) - це підмножина штучного інтелекту, яка будує математичну модель на основі зразкових даних, відомих як "дані навчання", для того, щоб робити прогнози або приймати рішення, не будучи явно запрограмованими на виконання завдання.

При застосуванні машинного навчання, нейронних мереж та еволюційних обчислень зазвичай використовують навчальний набір та тестовий набір. Під набором для тренувань розуміють об'єднання маркованого набору та немаркованого набору прикладів, доступних при машинному навчанні. Для порівняння, набір тестів складається з прикладів, які не використовувалися раніше.

Нехай $(X_l, Y_l) = \{(x_1, y_1),..., (x_l, y_l)\}$ позначають мічений набір, де $x_i \in RD$ - це i-й D-вимірний вектор даних, а $y_i \in R$ або $y_i \in \{1, ..., M\}$ - відповідна мітка вектора даних x_i. У задачах регресії y_i - це регресія або підгонка x_i. У задачах класифікації y_i є відповідною міткою класу x_i серед M класів цілей. Помічені дані x_i, $i = 1, ..., l$ спостерігаються користувачем, тоді як y_i, $i = 1, ..., l$ позначаються експертами або наглядачами з маркування даних. Немаркований набір складається з векторів даних і позначається $X_u = \{x_1, ..., x_u\}$.

Машинне навчання має на меті вставлення регресії або класифікатора шляхом вивчення навчального набору, а потім оцінити ефективність регресора або

класифікатора через тестовий набір. Відповідно до характеру навчальних даних, ми можемо класифікувати машинне навчання наступним чином:

1. Регулярне або евклідове вивчення структурованих даних:

- навчання з учителем
- навчання без учителя
- напівавтоматичне навчання
- навчання з підкріпленням
- Передавальне навчання

2. Машинне навчання на основі графів є нерегулярним або неевклідовим структурованим вивченням даних, і вивчає структуру графу, яка також називається побудовою графа, шляхом навчання зразків у напівнаглядових та неконтрольованих випадках навчання.

2.1.2 Дерева рішень

Контрольоване навчання на основі дерева рішень визначається як техніка побудови двійкового дерева на основі правил, яке можна інтерпретувати як ієрархічну техніку поділу доменів . Тому дерево рішень визначається як контрольована модель навчання, яка ієрархічно відображає домен даних на набір відповідей. Воно розділяє домен даних (вузол) рекурсивно на два субдомени, так що субдомени мають більший приріст інформації, ніж будучи розділений. Метою контрольованого навчання є класифікація даних, а отже, отримання інформації означає простоту класифікації в субдоменах, створених поділом. Пошук найкращого розподілу, який дає максимальний приріст інформації (тобто простота класифікації), є метою алгоритму оптимізації для навчання з учителем на основі дерева рішень.

В навчанні з учителем дерево рішень ділиться на класифікаційні дерева та дерева регресії. Простіше кажучи, дерево класифікації допомагає передбачити мітку класу для змінної у відповіді, тоді як дерево регресії допомагає передбачити

значення для змінної у відповіді. На рис. 2.1 процеси дерева класифікації та дерева регресії проілюстровані з використанням ієрархічної структури, щоб показати еволюцію властивостей поділу доменів[28].

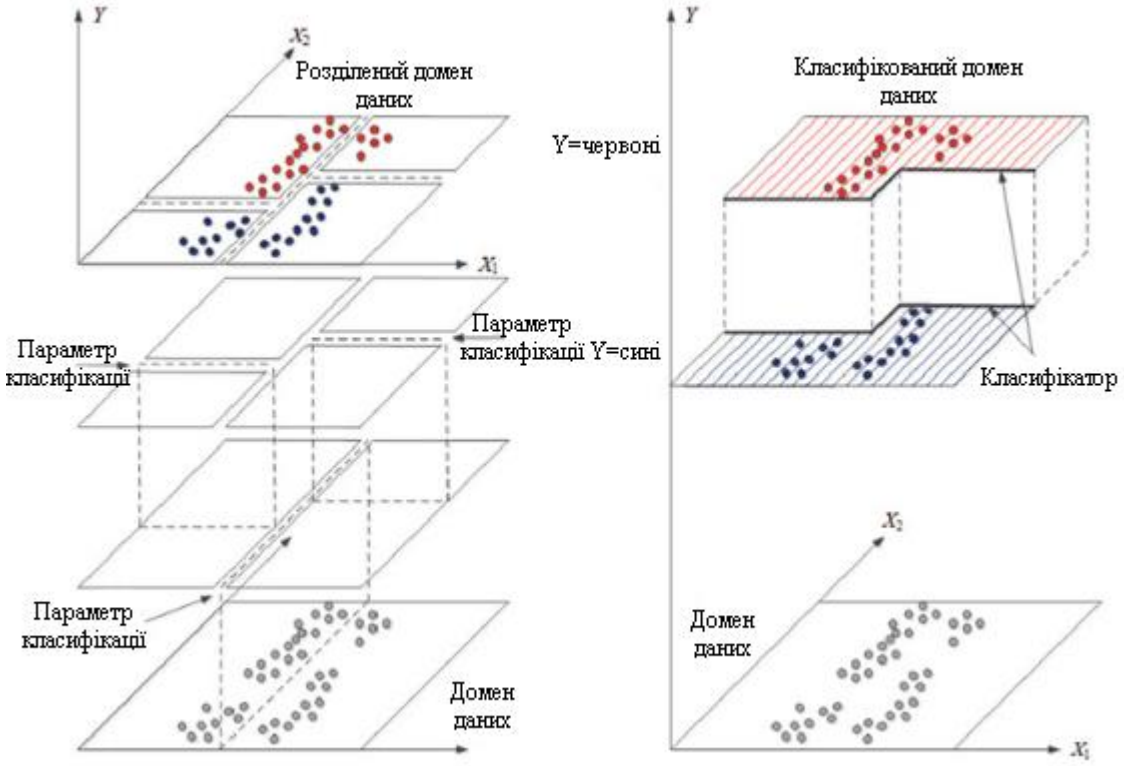

Рисунок 2.1 − Ілюстрація дерева класифікації у 3D із використанням двох класів із властивостями поділу доменів.

Дерево класифікації допомагає призначити мітку новому набору даних. Наприклад, це може допомогти нам вирішити, чи нове спостереження належить до класу 1 або класу 0. Концепція дерева класифікації пояснюється за допомогою ілюстрації на рис. 10.6, де використовуються два класи (червоний та синій). Його зміст полягає у побудові дерева класифікації на етапі навчання, і воно дає наочний приклад будови вирішального дерева. Воно також використовує домен даних та його зміни під час побудови дерева рішень. На лівій діаграмі показано розподіл розгалуження дерева, що розділяє домен даних і створює субдомени для відповідної класифікації на основі отриманих міток класів. Це можна розглядати як генерацію простих множинних тонких шарів доменів даних із розділеними областями. Перший

тонкий шар показує даний домен даних та умову розділення, яка застосовується до першої функції. Домен розділений на два субдомени, і це показано у другому тонкому шарі. Ці субдомени додатково поділяються на чотири субдомени на основі другої ознаки, і вони показані в третьому тонкому шарі. Нарешті, ярлики класів виділяються, і ми можемо бачити, що вони класифікуються на різні нероз'єднані субдомени.

2.2 Синтез моделі

Для виконання синтезу та аналізу моделі класифікації фішингових вкладень необхідно підготувати навчальні та тестові дані.

Модель процесу класифікації фішингових вкладень представлено у вигляді схеми на рисунку 2.2.

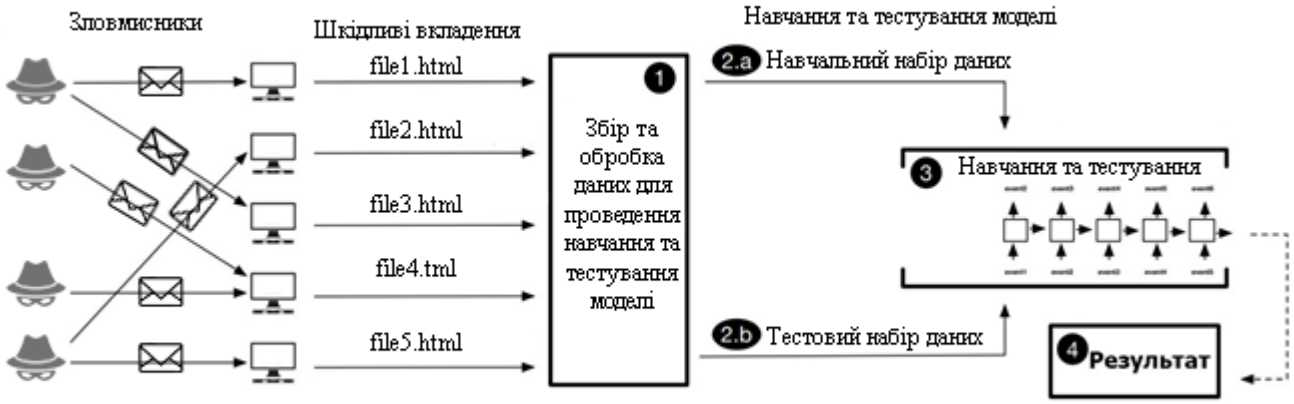

Рисунок 2.2 − Схема процесу обробки навчальних та тестових даних

Модель даного процесу складається з 4-ох етапів:

1) Збір зразків фішингових вкладень, що використовуються для здійснення фішингових атак.

2) Обробка зразків та отримання даних з подальшим приведенням їх до потрібного вигляду, що потрібен для їх логічної обробки.

3) Тренування та тестування моделі.

4) Аналіз результатів, отриманих в результаті виконання попередніх етапів.

Зазначені етапи описують схему, що буде використана в подальшому для проектування моделі виявлення фішингових вкладень. У даному розділі виконується опис моделі виявлення фішингових вкладень. В залежності від отриманих результатів буде здійснено оцінювання можливості застосування даної моделі.

2.3 Підготовка вхідних даних

Для проведення навчання та перевірки (тестування) інтелектуальної моделі потрібно мати достатню кількість вхідних даних. Кращі результати можна отримати при наявності великої кількості вхідних даних, що стосуються різних фішингових атак. Для цього необхідно провести збір даних, метою якого є формування набору даних для навчання і тестування, що включає дані про фішингові вкладення, які дають достатньо інформації для прийняття рішення стосовно того, чи є вкладення фішинговим.

Готових наборів даних (datasets), що стосуються фішингових вкладень не було знайдено в публічному доступі під час написання даної роботи. Через відсутність таких даних було здійснено створення набору таких даних в результаті обробки фішингових вкладень та проведення ручного аналізу цих вкладень з подальшим виділенням даних, які можуть бути використані для ідентифікації фішингових вкладень, а також проведено класифікацію цих вкладень та поділ на 3 класи, серед яких виділено безпечні HTM-вкладення (клас non-phishing), фішингові вкладення, що здійснюють маскування посилань за допомогою HTM-вкладеннь (клас redirector), та фішингові вкладення, що являють собою форми для збору даних аутентифікації (клас form).

Безпечні HTM-вкладення (клас non-phishing). До даного класу віднесено вкладення, які не є фішинговими та не є частиною фішингових атак. Вони не виконують перенаправлення на фішингові сайти та не здійснюють збір даних з подальшою їх відправкою на фішинговий сайт. В основному це статичні HTM-сторінки, що містять інформацію для отримувача листа.

Фішингові вкладення, що здійснюють маскування посилань за допомогою HTM-вкладеннь (клас redirector). До даного класу віднесено вкладення що містять посилання, яке перенавправляє користувача на фішинговий сайт за допомогою функцій «href» у явному вигляді та замаскованих посилань за допомогою додавання функцій JavaScript, таких як unescape().

Фішингові вкладення, що являють собою форми для збору даних аутентифікації (клас form). До даного класу фішингових вкладень віднесено великі HTM-файли, що копіюють дизайн форм аутентифікації олегітимних сервісів та здійснюють відправку даних на сервери зловмисників за допомогою запитів POST та використання відповідних функцій для цього.

2.4 Підготовка даних для навчання

Аналіз HTM-вкладень здійснювався з класифікацією цих вкладень та збором даних про ці вкладення, які будуть використані для навчання моделі. Аналіз та збір даних здійснювався вручну.

Зібрані дані були зібрані у форматі CSV з 5-ма полями даних для навчання моделі та 6-м значенням, що відповідало класу цих вкладень відповідно до проведеного їх аналізу, серед яких наявні наступні поля даних: кількість рядків коду у вкладеннях (поле Lines), використання в файлі функції unescape() (поле Unescape), використання функцій перенаправлення на сторінку в формані HTM (поле href), використання функцій, що дозволяють використовувати HHTP-метод POST (поле POST), використання функцій для кодування даних в форматі base64 (поле Base64) та поле даних з класом, до якого віднесено дане вкладення (поле Class).

Таблиця 2.1 − Опис полів вхідних даних

Поле	Значення	Опис поля
Lines	1-∞	Кількість рядків коду у файлі
Unescape	True,False	Наявність чи відсутність функції unescape() у файлі

href	True,False	Наявність чи відсутність функції href у файлі
POST	True,False	Наявність чи відсутність функції POST у файлі

Продовження таблиці 2.1

Base64	True,False	Наявність чи відсутність функцій для декодування base64 у файлі
Class	redirector, form, non-phishing	Клас, до якого віднесено даний файл

2.5 Підготовка моделі навчання

Синтез моделі виконано на базі алгоритму j48, що є реалізацією алгоритму C4.5 на мові Java.

Алгоритм C4.5 будує дерево рішень із стратегією поділу. У C4.5 кожен вузол у дереві пов'язаний з набором випадків. Також випадкам присвоюються ваги з урахуванням невідомих значень атрибутів. На початку присутній лише корінь, який асоціюється з усім навчальним набором TS та з усіма вагами кожного випадку, рівними 1:0. На кожному вузлі виконується наступний алгоритм поділу, намагаючись використати найкращий локальний вибір без зворотного відстеження, який можна представити у вигляді псевдокоду[29]:

(1) Compute Class Frequency(T);

(2) if One Class or Few Cases

return a leaf;

create a decision node N;

(3) For Each Attribute A

Compute Gain (A);

(4) N.test = Attribute With Best Gain;

(5) if N.test is continuous

find Threshold;

(6) For Each T` in the splitting of T

(7) if T` is Empty

Child of N is a leaf

else

(8) Child of N = Form Tree (T`);

(9) Compute Errors of N;

return N

Для підготовки моделі в даній роботі використане програмне забезпечення Weka.

Weka - це організована колекція алгоритмів машинного навчання та інструментів попередньої обробки даних. Основний спосіб взаємодії з цими методами шляхом їх виклику з командного рядка. Однак, крім цього, надаються зручні інтерактивні графічні користувальницькі інтерфейси для дослідження даних, для проведення масштабних експериментів на розподілених обчислювальних платформах і для проектування конфігурацій для потокової обробки даних. Ці інтерфейси є вдосконаленим середовищем для експериментального аналізу даних. Система написана на Java та поширюється на умовах Загальної публічної ліцензії GNU[30].

2.6 Навчання моделі

В процесі виконання даної роботи дані для навчання були завантажені в програмі Weka та проведено розбір параметрів даних.

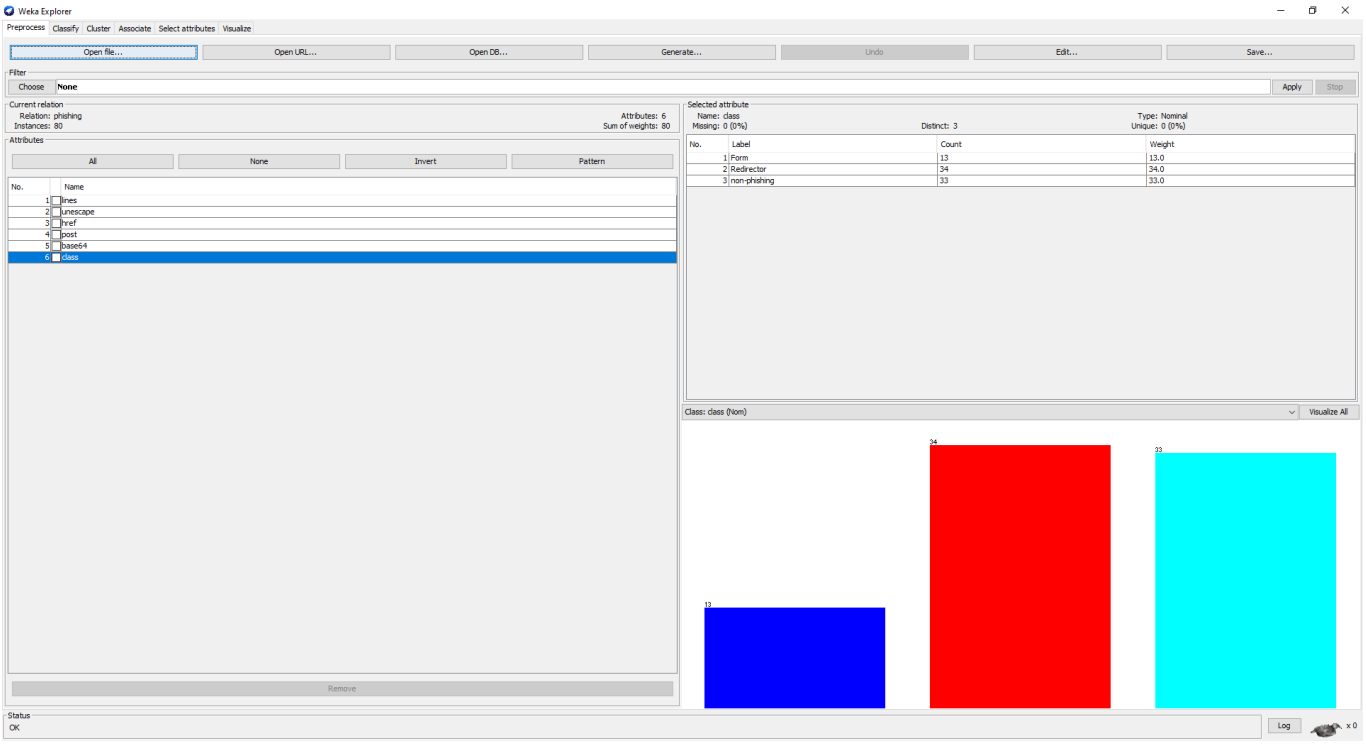

Рисунок 2.3 – Набір даних для навчання моделі виявлення фішингових HTM-вкладень.

В результаті обробки даних програмою Weka за допомогою алгоритму J48 було отримано результати представлені на рисунку 2.4, які мають точність класифікації в 98,75% для тестового набору даних.

Рисунок 2.4 – Результат створення моделі виявлення фішингових HTM-вкладень з застосуванням алгоритму J48.

Матриця невідповідностей, що показує наявність помилок при обробці показана в таблиці 2.2.

Таблиця 2.2 – Матриця невідповідностей

Form	Redirector	Non-phishing	
12	0	0	Класифіковано як Form
1	34	0	Класифіковано як Redirector
0	0	33	Класифіковано як Non-phishing

В результаті даних дій було отримано дерево рішень, яке можна використовувати для написання алгоритму виявлення фішингових HTM-вкладень. Візуалізація даного дерева рішень представлена на рисунку 2.5

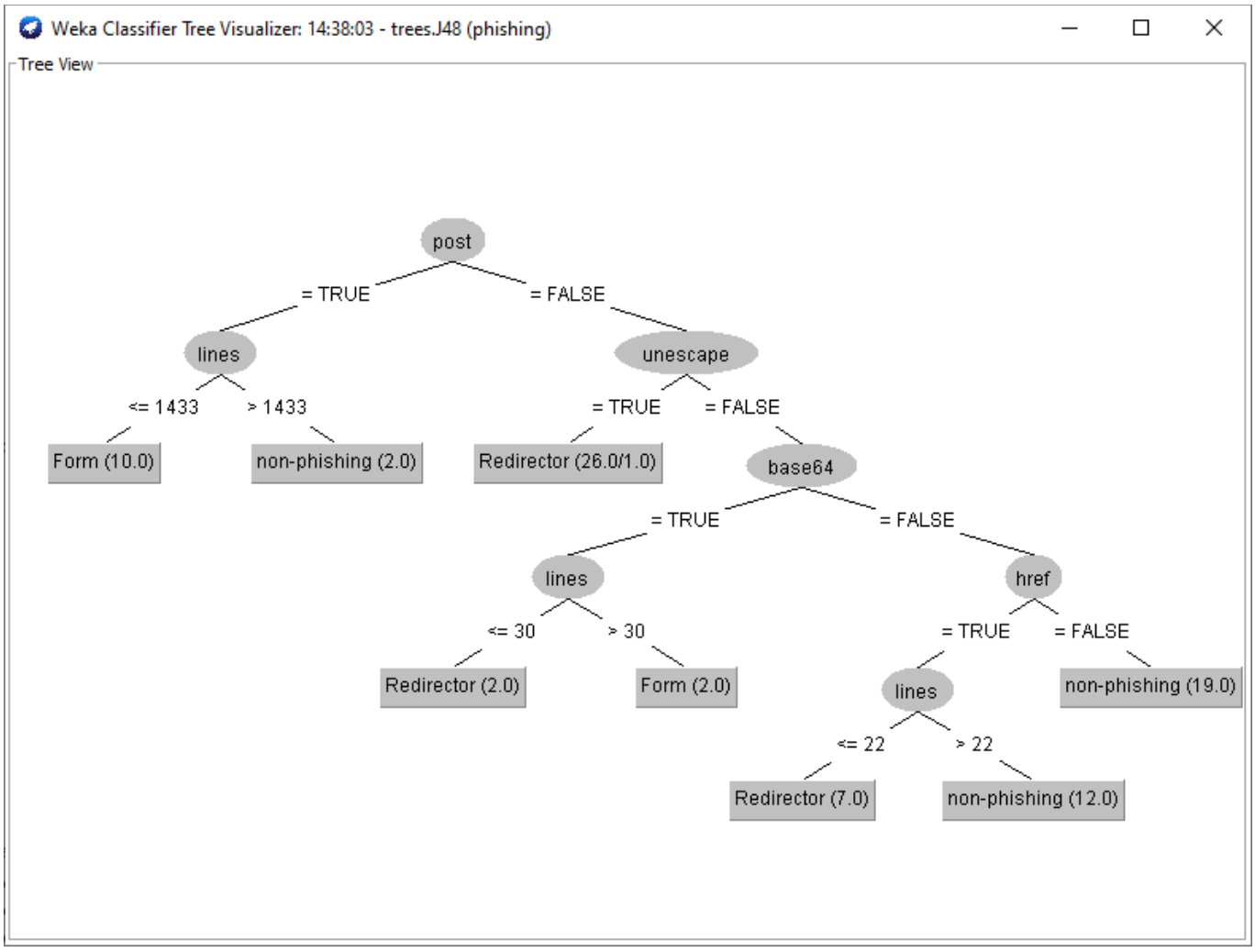

Рисунок 2.5 – Візуалізація дерева рішень для моделі виявлення фішингових HTM-вкладень, що було отримано в результаті застосування алгоритму J48.

2.7 Аналіз адекватності роботи моделі

Для тестування адекватності моделі використано тестовий набір даних, який складається з 20% даних, зібраних при аналізі фішингових вкладень. Результат аналізу представлено на рисунку 2.6.

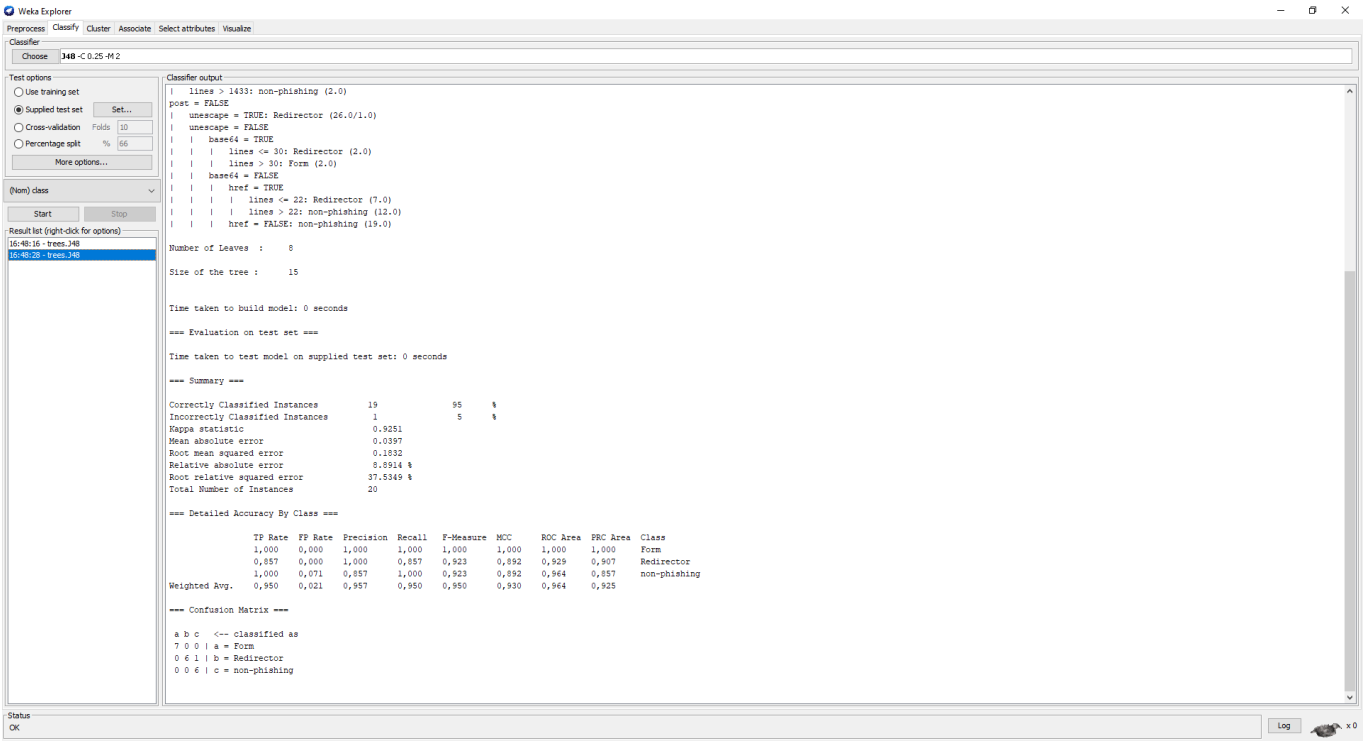

Рисунок 2.6 – Результат перевірки адекватності моделі

Результати навчання та перевірки адекватності представлено в таблиці 2.3.

Таблиця 2.3 – Відсотки правильної та неправильної класифікації моделі

Набір даних	Відсоток правильної класифікації	Відсоток неправильної класифікації
Навчальний	98,75%	1,25%
Тестовий	95%	5%

Висновки за розділом 2

В даному розділі дипломної роботи виконано аналіз предметної області та розробку моделі класифікації фішингових вкладень. В межах даного розділу було

виконано опис понять предметної області та теоретичних основ предметної області, серед яких здійснено опис машинного навчання та дерев рішень.

Для виконання мети роботи здійснено сентез моделі аналізу фішингових вкладень, підготовлено вхідні дані для навчання та тестування, виконано навчання моделі та побудову дерева рішень даної моделі, а також проведено аналіз адекватності даної моделі з використанням даних, що не були використані при навчанні моделі.

Таким чином в даному розділі було підготовлено модель класифікації фішингових вкладень, яка може бути використана для побудови ситеми захисту, що дозволяє виявляти шкідливі HTM-вкладення за допомогою розробленого дерева рішень.

РОЗДІЛ 3
ПОБУДОВА ТА РЕАЛІЗАЦІЯ СИСТЕМИ ЗАХИСТУ ВІД ФІШИНГОВИХ ВКЛАДЕНЬ

3.1 Побудова та реалізація системи захисту від фішингових вкладень

3.1.1 Побудова моделі системи

Для реалізації системи захисту від фішингових вкладень необхідно побудувати модель даної системи. Для спрощення перевірки роботи даної системи вона буде виконана у вигляді скрипта. Даний скрипт буде виконувати перевірку файлів за допомогою отримання метаданих файлу, таких як кількість рядків коду у файлі та використання певних функцій у ньому. Візуалізацію схеми наведено на рисунку 3.1.

Рисунок 3.1 – Схема роботи тестової системи класифікації фішингових вкладень

3.1.2 Розробка системи

Скрипт для аналізу даних буде виконаний на мові Python. Даний скрипт буде здійснювати збір метаданих про потенційно фішингові файли та здійснювати аналіз відповідно до дерева рішень, що було розроблене в розділі 2 даної дипломної роботи. Лістінг коду даного скрипта наведено в Додатку А до даної дипломної роботи.

Даний скрипт дозволяє отримати інформацію про класифікацію файлу як безпечного (non-phishing) чи фішингового (form, redirector).

3.1.3 Перевірка роботи системи

Скрипт script.py видає інформацію в консолі, яка в залежності від того, до якого класу був віднесений файл буде виглядати як показано на рисунках 3.2, 3.3, 3.4.

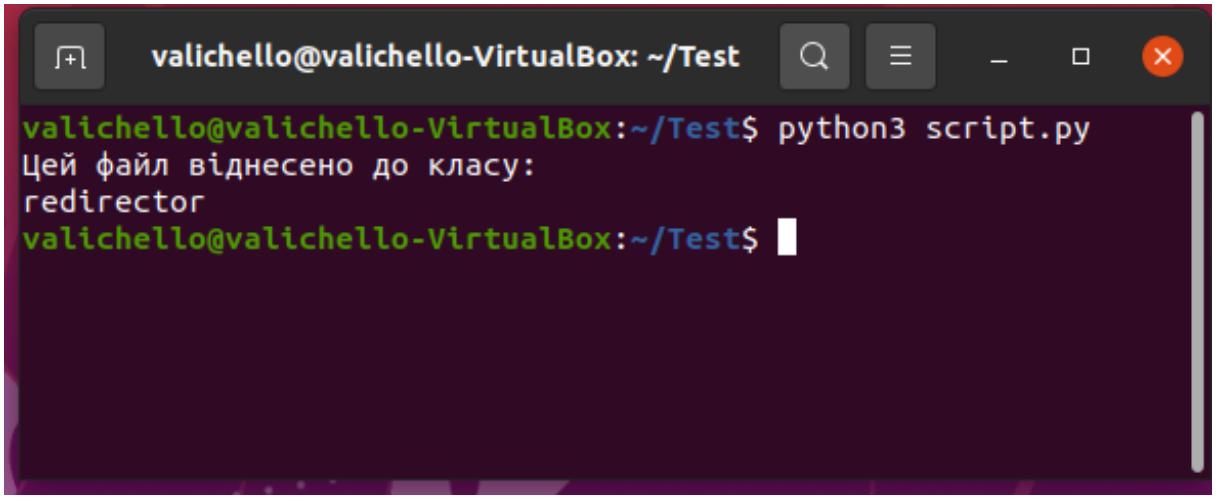

Рисунок 3.2 – Результат роботи скрипта при класифікації файла як redirector

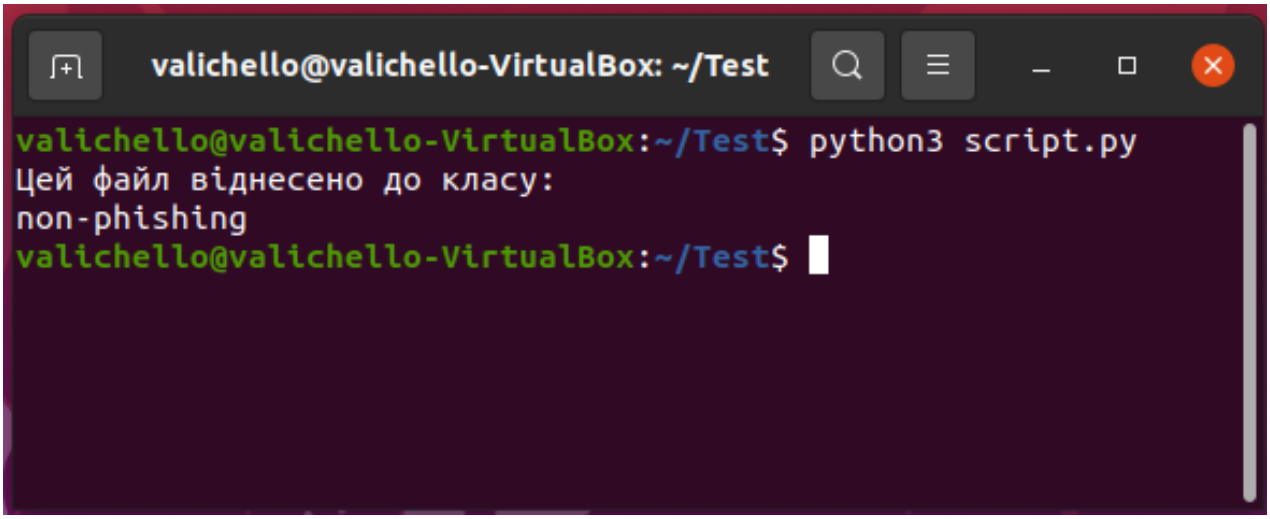

Рисунок 3.3 – Результат роботи скрипта при класифікації файла як non-phishing

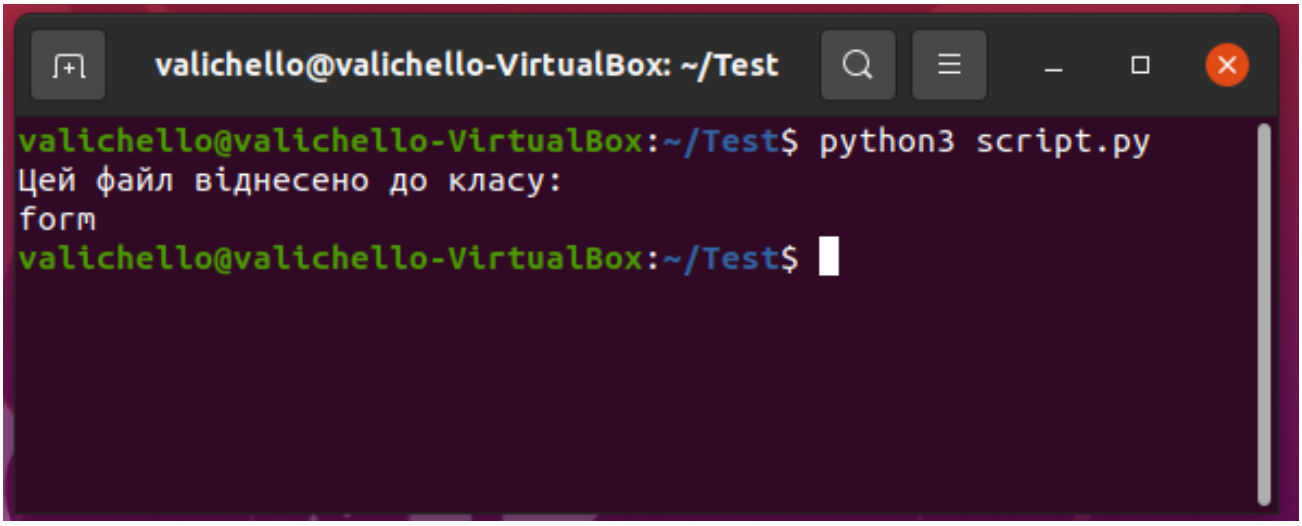

Рисунок 3.3 – Результат роботи скрипта при класифікації файла як form

3.2 Тестування ефективності побудованої системи захисту від фішингових вкладень.

Фішингові вкладення можуть бути виявлені антивірусними програмами. Для тестування ефективності даної моделі було використано тестування шкідливих

файлів на 3-х найпопулярніших антивірусних продуктах (Windows Defender, ESET-NOD32, Avast)[31].

При тестуванні шкідливих файлів за допомогою скрипта script.py було виявлено 80% протестованих файлів. При тестуванні файлів через антивірусні програми було виявлено 60% протестованих файлів принаймні однією антивірусною програмою. Таким чином це дозволяє виявляти 20% шкідливих вкладень, які не були виявлені антивірусними програмами.

Висновки за розділом 3

Таким чином, в даному розділі було розроблено схему системи захисту від фішингових вкладень, розроблено скрипт для перевірки роботи даної системи, а також виконано тестування та порівняння роботи даної системи з іншим підходом виявлення фішингових вкладень. В результаті створення тестової реалізації системи виявлення фішингових вкладень та порівняння результатів виявлення фішингових вкладень було виявлено 20% фішингових вкладень, які не були виявлені антивірусними програмами.

ВИСНОВКИ

В даній дипломній роботі було виконано аналіз фішингових атак відповідно до спрямування атаки, в межах якої виділено масову фішингову розсилку, цільовий фішинг, вейлінг (whaling), компрометацію ділової електронної пошти, клонувальний фішинг, вішинг (vishing) або ж фішинг по телефону, смішинг (smishing) або ж фішинг за допомогою текстового повідомлення та короткострокове поширення шкідливих повідомлень.

Також в даній роботі було описано та прокласифіковано фішингові атаки відповідно до застосованих методів: фішинг з використанням шкідливих вкладень, фішинг з використанням шкідливих посилань, фішинг з використанням сервісів. Було виконано аналіз методів захисту від фішингу та описано принцип їх дії. Було розділено методи захисту від фішингу на 5 категорій.

В даній роботі здійснено розробку моделі класифікації фішингових вкладень.

Для виконання мети роботи здійснено сентез моделі аналізу фішингових вкладень, підготовлено вхідні дані для навчання та тестування, виконано навчання моделі та побудову дерева рішень даної моделі, а також проведено аналіз адекватності даної моделі з використанням даних, що не були використані при навчанні моделі.

В результаті виконання даної дипломної роботи було розроблено схему системи захисту від фішингових вкладень, розроблено скрипт для перевірки роботи даної системи, а також виконано тестування та порівняння роботи даної системи з іншим підходом виявлення фішингових вкладень. В результаті створення тестової реалізації системи виявлення фішингових вкладень та порівняння результатів виявлення фішингових вкладень було виявлено 20% фішингових вкладень, які не були виявлені антивірусними програмами.

Розроблена система дозволяє збільшити ефективність виявлення фішингових вкладень до рівня 80%.

Подальшою роботою в даній сфері є розробка моделей та систем для виявлення більшого спектру фішингових вкладень, а також збільшення ефективності роботи вже наявних моделей та систем.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Security Boulevard [Електронний ресурс] / Staggering phishing statistics in 2020 Режим доступу: https://securityboulevard.com/2020/12/staggering-phishing-statistics-in-2020/

2. MITRE ATT&CK [Електронний ресурс] / Офіційний веб-сайт корпорації MITRE / Phishing –Режим доступу: https://attack.mitre.org/techniques/T1566/

3. BBC News [Електронний ресурс] / Офіційний веб-сайт BBC / Google blocking 18m coronavirus scam emails every day – Режим доступу: https://www.bbc.com/news/technology-52319093

4. CSO [Електронний ресурс] / Офіційний веб-сайт CSO / What is phishing? How this cyber attack works and how to prevent it – Режим доступу: https://www.csoonline.com/article/2117843/what-is-phishing-how-this-cyber-attack-works-and-how-to-prevent-it.html

5. CSO [Електронний ресурс] / Офіційний веб-сайт CSO / 8 types of phishing attacks and how to identify them – Режим доступу: https://www.csoonline.com/article/3234716/8-types-of-phishing-attacks-and-how-to-identify-them.html

6. Cisco Talos Intelligence Blog [Електронний ресурс] / Офіційний веб-сайт Cisco Talos Intelligence / “Cyber Conflict” Decoy Document Used In Real Cyber Conflict – Режим доступу: https://blog.talosintelligence.com/2017/10/cyber-conflict-decoy-document.html

7. MITRE ATT&CK [Електронний ресурс] / Офіційний веб-сайт корпорації MITRE / Phishing: Spearphishing Attachment –Режим доступу: https://attack.mitre.org/techniques/T1566/001/

8. MITRE ATT&CK [Електронний ресурс] / Офіційний веб-сайт корпорації MITRE / Phishing: Spearphishing Link –Режим доступу:

https://attack.mitre.org/techniques/T1566/002/

9. MITRE ATT&CK [Електронний ресурс] / Офіційний веб-сайт корпорації MITRE / Phishing: Spearphishing via Service –Режим доступу: https://attack.mitre.org/techniques/T1566/003/

10. Improved Phishing Detection using Model-Based Features. / Bergholz, A., Chang, J. H., Paass, G., Reichartz, F., Strobel, S. // CEAS. – 2008

11. A content-based approach to detecting phishing web sites. / Zhang, Y., Hong, J., & Cranor, L. Cantina// 16th international conference on World Wide Web, Banff, Alberta, Canada – 2007

12. A Real-Life Study in Phishing Detection. / Andre, B., Gerhard, P., Luigi, D., & Domenico, D. // Conference on Email and Anti-Spam (CEAS), Redmond, Washington. – 2010

13. Data clustering: a review. / Jain, A. K., Murty, M. N., & Flynn, P. J. // ACM computing surveys (CSUR) – 1999

14. A survey of recent advances in hierarchical clustering algorithms. / Murtagh, F. // The Computer Journal – 1983

15. Anomaly detection: A survey. / Chandola, V., Banerjee, A., & Kumar, V. // ACM computing surveys (CSUR) – 2009

16. Estimating the support of a high-dimensional distribution. / Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. // Neural computation – 2001

17. Anomaly based malicious URL detection in instant messaging. / Guan, D., Chen, C. M., & Lin, J. B. // Joint Workshop on Information Security (JWIS), Kaohsiung, Taiwan – 2009

18. Survey of text mining. / Berry, M. W., & Castellanos, M. // Computing Reviews – 2004

19. Latent semantic analysis and keyword extraction for phishing classification. / L'Huillier, G., Hevia, A., Weber, R., & Rios, S. // Intelligence and Security Informatics (ISI), Vancouver, BC, Canada – 2010

20. For fake: four studies on how we fall for phish. / Blythe, M., Petrie, H., & Clark, J. A. F // SIGCHI Conference on Human Factors in Computing Systems – 2011

21. What instills trust? a qualitative study of phishing. / Jakobsson, M., Tsow, A., Shah, A., Blevis, E., & Lim, Y.-K. // Financial Cryptography and Data Security – 2007

22. Phishing for user security awareness. / Dodge, R. C., Carver, C., & Ferguson, A. J. // Computers & Security – 2007

23. A new anti-phishing method in OpenID. / Lee, H., Jeun, I., Chun, K., & Song, J. // Second International Conference on Emerging Security Information, Systems and Technologies – 2008

24. Ontological Semantic Technology Goes Phishing / Taylor, J. M., Raskin, V., & Spafford, E. H. // CERIAS Security Seminar Presentation – 2011

25. A systematic characterization of im threats using honeypots. / Antonatos, S., Polakis, I., Petsas, T., & Markatos, E. P. // The Network and Distributed System Security Symposium(NDSS), San Diego, California, USA – 2010

26. Automatic detection of phishing target from phishing webpage. / Liu, G., Qiu, B., & Wenyin, L. // 20th International Conference on Pattern Recognition (ICPR'10) – 2010

27. PhorceField: a phish-proof password ceremony. / Hart, M., Castille, C., Harpalani, M., Toohill, J., & Johnson, R. // 27th Annual Computer Security Applications Conference – 2011

28. Machine Learning. / Zhang XD. // A Matrix Algebra Approach to Artificial Intelligence, Springer, Singapore. – 2020

29. Efficient C4.5 [classification algorithm]. / Ruggieri, S. // IEEE Transactions on Knowledge and Data Engineering – 2002

30. Weka-A Machine Learning Workbench for Data Mining. / Frank, E., Hall, M., Holmes, G., Kirkby, R., Pfahringer, B., Witten, I. H., & Trigg, L. // Data Mining and Knowledge Discovery Handbook – 2009

31. ZDNET [Електронний ресурс] / Веб-сайт ZDNET / Which is the most popular antivirus software? –Режим доступу: https://www.zdnet.com/article/which-is-the-most-popular-antivirus-software/

ДОДАТКИ

ДОДАТОК А

Лістінг коду скрипта script.py

```
filename  = "1.html"

fileHandler  = open(filename,  "r")

line_count = 0

for line  in fileHandler:

    if line != "\n":

        line_count += 1

unescape=False

post=False

href=False

base64=False

with open(filename)  as f:

    if 'unescape(' in f.read():

        unescape = True
```

```
    if 'href' in f.read():
        href = True
    if 'POST' in f.read():
        post = True
    if 'base64' in f.read():
        base64 = True

print('Цей файл віднесено до класу:')
if (post==True):
        if (lines >= 1433):
                print('non-phishing')
        else:
                print('form')
else:
        if (unescape==True):
```

```
            print('redirector')

        else:

            if (base64==True):

                if (lines>30):

                    print('form')

                else:

                    print('redirector')

            else:

                if (href==False):

                    print('non-phishing')

                else:

                    if (lines>22):

                        print('non-phishing')

                    else:

                        print('redirector')
fileHandler.close()
```