МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА

Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»

Кафедра системного аналізу та кібербезпеки

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА «СИСТЕМНИЙ АНАЛІЗ»
Газузь знань 12 «Інформаційні технології»
Спеціальність 124 «Системний аналіз»

Форма навчання: денна

КВАЛІФІКАЦІЙНА МАГІСТЕРСЬКА РОБОТА

на тему: **«РОЗРОБКА ТА ВПРОВАДЖЕННЯ СИСТЕМИ ЗАХИСТУ ВІД ФІШИНГОВИХ АТАК В СОЦІАЛЬНИХ МЕРЕЖАХ»**

здобувачки Горової Аліни Володимирівни _____
(підпис)

Науковий керівник: к.е.н., професор Бєгун А.В.

_____
(підпис)

Робота допущена до захисту перед екзаменаційною комісією з атестації здобувачів вищої освіти (ЕК)

Завідувач кафедри доктор фізико-математичних наук,
професор Джалладова Ірада Агаверді-кизи

_____
(підпис)

Київ 2024

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ВАДИМА ГЕТЬМАНА
Навчально-науковий інститут
«Інститут інформаційних технологій в економіці»
Кафедра системного аналізу та кібербезпеки

ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА **«Системний аналіз»**
СПЕЦІАЛЬНІСТЬ **124 «Системний аналіз»**

ПОГОДЖЕНО
Керівник проєктної групи (гарант)
освітньо-професійної програми
д.фіз.-мат.н., проф. Джаладова І.А.

_____
(підпис)
_____2024

ЗАТВЕРДЖУЮ
Завідувач кафедри
д.фіз.-мат.н., проф. Джаладова І.А.

_____
(підпис)
_____2024

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
здобувачки вищої освіти Горової Аліни Володимирівни
очної(денної) форми навчання

на підготовку кваліфікаційної магістерської роботи
на тему «Розробка та впровадження системи захисту від фішингових атак в соціальних мережах»
Тему затверджено наказом ректора Університету від «_____» _____ 20___ р .№ _____

Кваліфікаційна магістерська робота виконується на основі наукових праць, інформації зібраної з соціальних мереж та даних з відкритих джерел

План кваліфікаційної магістерської роботи

Розділ 1	ТЕОРЕТИЧНІ ЗАСАДИ СИСТЕМИ ЗАХИСТУ В ІНТЕРНЕТІ
Розділ 2	СИСТЕМНИЙ АНАЛІЗ ФІШИНГОВИХ АТАК В СОЦІАЛЬНИХ МЕРЕЖАХ
Розділ 3	ПРОЄКТУВАННЯ ТА ВПРОВАДЖЕННЯ СИСТЕМИ ЗАХИСТУ ВІД ФІШИНГОВИХ АТАК В СОЦІАЛЬНИХ МЕРЕЖАХ

Об’єкт дослідження:	Процес виявлення фішингових атак в інформаційних системах.
Предмет дослідження:	Методи та інструменти для аналізу мережевого трафіку з мет виявлення фішингових атак.

Мета кваліфікаційної магістерської роботи:	Розробка та впровадження ефективного рішення для моніторингу і аналізу мережевого трафіку, що дозволить підвищити рівень безпеки інформаційних систем.

Конкретні завдання, які здобувач повинен виконати для досягнення поставленої мети:

У розділі 1	Здійснити огляд історичних відомостей про фішинг, визначити теоретичні засади формування фішингу та його характерні особливості, описати схему побудови фішингу, класифікувати різновиди фішингових атак та провести аналіз існуючих методів захисту від фішингових атак, оцінити їх ефективність та вразливості.

У розділі 2	Провести системний аналіз типів фішингових атак, що відбуваються в соціальних мережах, використати інструменти Wireshark для аналізу мережевого трафіку під час фішингових атак, виявити специфічні характеристики фішингового трафіку в соціальних мережах за допомогою аналізу пакетів даних, розглянути відомі приклади фішингових атак в соціальних мережах та проаналізувати їх наслідки для користувачів та компаній.

У розділі 3	Розробити та впровадити систему фільтрації електронної пошти для виявлення та блокування фішингових повідомлень в соціальних мережах, описати програмні компоненти, необхідні для захисту від фішингових атак в соціальних мережах, розглянути можливості захисту від фішингу за допомогою Cisco Email Security Appliance, створити програму захисту від фішингових атак в соціальних мережах.

Завдання підготував науковий керівник _____ _____ **Бєгун А.В.**

(підпис) *(ініціали, прізвище)*

«_____» _____ 2024 р.

Завдання одержав здобувач _____ **А.В. Горова**

(підпис) *(ініціали, прізвище)*

«_____» _____ 2024 р.

РЕФЕРАТ

Кваліфікаційна магістерська робота містить 91 сторінку, 1 таблицю, 11 рисунків, список літератури з 48 найменувань та 3 додатки.

РОЗРОБКА ТА ВПРОВАДЖЕННЯ СИСТЕМИ ЗАХИСТУ ВІД ФІШИНГОВИХ АТАК В СОЦІАЛЬНИХ МЕРЕЖАХ

Об’єктом дослідження є процес виявлення фішингових атак в інформаційних системах.

Предмет дослідження: методи та інструменти для аналізу мережевого трафіку з метою виявлення фішингових атак.

Метою кваліфікаційної магістерської роботи є розробка та впровадження ефективного рішення для моніторингу і аналізу мережевого трафіку, що дозволить підвищити рівень безпеки інформаційних систем.

Завдання кваліфікаційної магістерської роботи: провести огляд сучасних методів виявлення фішингових атак та дослідити їхні недоліки, визначити ключові характеристики та вразливості, які використовуються у фішингових атаках, розробити критерії для аналізу мережевого трафіку, що можуть свідчити про можливі фішингові атаки, вивчити різні інструменти та технології для детектування фішингових активностей у мережевому трафіку, розробити алгоритм для виявлення та аналізу фішингових атак, провести тестування розробленого алгоритму на реальних даних і оцінити його ефективність, сформулювати рекомендації для підвищення рівня безпеки інформаційних систем на основі результатів аналізу, розробити інструкції для фахівців з інформаційної безпеки щодо моніторингу мережевого трафіку та виявлення фішингових атак.

Наукова новизна одержаних результатів полягає у розробці та впровадженні інноваційної системи захисту від фішингових атак, спеціально адаптованої для соціальних мереж.

Практичне значення одержаних результатів. Запропоновані рекомендації можуть бути використані в практичній діяльності фахівців з інформаційної безпеки для підвищення рівня захисту інформаційних систем від фішингових атак.

Ключові слова: фішинг, фішингові атаки, соціальна інженерія, Інтернет-захист, системи заххисту, соціальні мережі, Wireshark

В і д г у к

про кваліфікаційну магістерську роботу здобувача навчально-наукового інституту «Інститут інформаційних технологій в економіці» освітньо-професійної програми « Системний аналіз» **Горової Аліни Володимирівни** на тему: «Розробка та впровадження системи захисту від фішингових атак в соціальних мережах»

Зміни у ставленні більшості країн світу, зокрема й України, до власної інформаційної й кібернетичної безпеки спонукають приділяти дедалі більшу увагу до розробки й пошуку систем захисту від атак в соціальних мережах. Як результат, користувач шукає варіанти вибору і впровадження системи з найбільш надійної структурою і раціональною вартістю.

На ринку інформаційних технологій присутні безліч варіантів впровадження систем захисту, в кожної з яких є свої негативні та позитивні сторони. Всі вони не схожі між собою за зовнішнім виглядом, складністю ведення та вартістю, але результат пропонують майже ідентичний – повна або часткова захищеність від фішингових атак. З цих аргументів тема магістерської дипломної роботи Горової А.В. безумовно є актуальною.

Робота складається з трьох розділів, кожний з яких має свою специфіку і особливості.

Розділ 1 присвячений аналізу теоретичних відомостей з проблеми виникнення, формування та впровадження фішингу, як об’єкту дослідження. Особлива увага приділяється аналізу різновиду фішингових атак та методам захисту від них.

У розділі 2 акцентується увага безпосередньо на системному аналізі застосування фішингових атак в соціальних мережах, застосуванню інструментів Wireshark для визначення особливостей фішингового трафіку, аналізу пакетів даних і потенційних наслідків від дії атак на користувачів.

Третій розділ пропонує постановку задачі впровадження в практичну діяльність засобів реалізації та налаштування електронної пошти проти фішингових атак в соціальних мережах. На основі системного аналізу рекомендуються програмні компоненти захисту продукту Cisco Email Security Appliance та їх інтеграція в соціум і бізнес.

Зауваження. Матеріал дипломної роботи бажано супроводжувати засобами функціонального призначення: математичними моделями фішингових атак в соціальних мережах, економічними розрахунками запропонованої й впровадженої системи захисту; слід звернути увагу до деяких семантичних оборотів в тексті.

В цілому дипломна робота виконана на рівні вимог до кваліфікаційних магістерських робіт, має практичне значення, може бути допущена до захисту, заслуговує позитивної оцінки, а здобувач Горова А.В. присвоєння ступеня магістр спеціальності «Системний аналіз».

Науковий керівник,

к.е.н., професор А.В. Бєгун

Рецензія
про кваліфікаційну магістерську роботу
здобувача навчально-наукового інституту
освітньо-професійної програми
«Системний аналіз»

Горової Аліни Володимирівни
(прізвище, ініціали)

на тему «Розробка та впровадження системи захисту від фішингових атак в соціальних мережах»
(назва теми)

1. Актуальність теми:
Тема магістерської роботи є важливою і актуальною в світі сучасних умов зростання кількості кібератак, особливо фішингових. Суми збитків користувачів та організацій, особливо з розвитком діяльності онлайн-сервісів, складають велики цифри. Тому формування системи захисту конфіденційних даних в соціальних мережах стає критично важливим, а ефективні методи виявлення та запобігання фішинговим атакам доповнюють сучасні інструменти безпеки.

2. Позитивні риси кваліфікаційної магістерської роботи:
Робота характеризується використанням сучасних методів аналізу та виявлення фішингових атак. Здобувач пропонує комплексний підхід та інноваційний алгоритм до захисту конфіденційної інформації, що підвищує ефективність протидії фішинговим загрозам.

3. Наявність самостійних розробок автора:
У дослідженні запропоновано розроблений алгоритм для виявлення фішингових атак у реальному часі. Програмний продукт було протестовано на реальних даних та показано високу ефективність функціонування. Здобувачем розроблено ряд мір та рекомендацій підвищення безпеки інформаційних систем складної структури.

4. Цінність теоретичних висновків та практичних рекомендацій:
Теоретичні засади дослідження містять системний аналіз виникнення, формування і розвитку терміну «фішинг», застосування сучасних методів виявлення фішингових атак, визначення їх характеристик та розробку критеріїв аналізу трафіку. Результати аналізу створюють умови продовження досліджень у сфері кібербезпеки соціальних мереж.
5. Наявність недоліків:
Основним недоліком роботи є обмеженість кількості тестових даних, які можуть вплинути на надійність запропонованого алгоритму. Тому потрібні додаткові дослідження програмного продукту для адаптації методів до нових фішингових атак.

6. Загальна оцінка кваліфікаційної магістерської роботи та її допущення до захисту перед ЕК:

В цілому кваліфікаційна магістерська робота відповідає встановленим вимогам і допускається до захисту перед ЕК.

Рецензент,
Зав. кафедрою «Системного аналізу»
Державного університету інформаційно-комунікаційних технологій
канд. фіз.-мат. наук, доцент Нафєєв Р. К.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

1. ICT - Information and Communication Technology (інформаційні та комунікаційні технології)

2.URL - Uniform Resource Locator (уніфікований покажчик ресурсу)

3. HTML - HyperText Markup Language (мова розмітки гіпертексту)

4. HTTP - Hypertext Transfer Protocol (протокол передачі гіпертексту)

5. HTTPS - Hypertext Transfer Protocol Secure (безпечний протокол передачі гіпертексту)

6. IP - Internet Protocol (інтернет-протокол)

7. DNS - Domain Name System (система доменних імен)

8. DDoS - Distributed Denial of Service (розподілена відмова в обслуговуванні)

9. SQL - Structured Query Language (структурована мова запитів)

10. XSS - Cross-Site Scripting (міжсайтове скриптування)

11. CSRF - Cross-Site Request Forgery (міжсайтове підроблення запитів)

12. VPN - Virtual Private Network (віртуальна приватна мережа)

13. Wi-Fi - Wireless Fidelity (бездротова точність)

14. IoT - Internet of Things (інтернет речей)

15. API - Application Programming Interface (інтерфейс програмування додатків)

16. SSL - Secure Sockets Layer (рівень захищених сокетів)

17. TLS - Transport Layer Security (захист транспортного рівня)

18. 2FA - Two-Factor Authentication (двухфакторна аутентифікація)

19. SMS - Short Message Service (служба коротких повідомлень)

20. OTP - One-Time Password (одноразовий пароль)

ВСТУП

Сучасний розвиток інформаційних технологій значно змінив спосіб, у який люди взаємодіють між собою, зокрема в контексті соціальних мереж. Платформи такі як Facebook, Instagram, Twitter та інші стали невід'ємною частиною повсякденного життя, забезпечуючи швидку комунікацію та обмін інформацією між користувачами. Однак, зростання популярності соціальних мереж супроводжується підвищенням ризиків, пов'язаних із кібербезпекою, особливо у контексті фішингових атак[1].

Фішинг, як один із найпоширеніших видів кіберзагроз, являє собою спробу зловмисників отримати конфіденційну інформацію (таку як логіни, паролі, банківські дані) шляхом маскування під надійні джерела. Особлива небезпека фішингу полягає в його здатності еволюціонувати, використовуючи все більш складні та підступні методи для обману користувачів. У соціальних мережах фішингові атаки набувають особливої актуальності через високу активність користувачів та велику кількість персональної інформації, яка може бути використана для підвищення ефективності атак[2].

Розробка та впровадження систем захисту від фішингових атак у соціальних мережах є надзвичайно важливим завданням. Відсутність адекватного захисту може призвести до серйозних наслідків як для окремих користувачів, так і для організацій, включаючи фінансові втрати, компрометацію особистих даних та підрив репутації. Зважаючи на масштаб використання соціальних мереж, ефективна боротьба з фішингом потребує комплексного підходу, який включає як технічні рішення, так і підвищення обізнаності користувачів[1].

Актуальність дослідження також обумовлена необхідністю підвищення рівня інформаційної безпеки в умовах зростаючих кіберзагроз. Розробка та впровадження ефективних систем захисту від фішингових атак сприятиме не тільки зниженню ризиків для користувачів, але й загальному

зміцненню інформаційної безпеки в суспільстві. Вивчення та застосування інструментів, таких як Wireshark, у поєднанні з іншими методами захисту, дозволяє створити комплексну систему, яка здатна ефективно протидіяти сучасним кіберзагрозам[3].

Таким чином, дослідження та розробка системи захисту від фішингових атак у соціальних мережах є надзвичайно актуальним завданням, яке має значний вплив на забезпечення безпеки особистих та корпоративних даних в умовах сучасного інформаційного середовища[1].

Мета дослідження: розробка та впровадження ефективного рішення для моніторингу і аналізу мережевого трафіку, що дозволить підвищити рівень безпеки інформаційних систем. У рамках даного дослідження передбачається створення інноваційного інструменту, який дозволить здійснювати комплексний аналіз мережевих даних у реальному часі, виявляти підозрілу активність та надавати рекомендації щодо запобігання потенційним загрозам.

Завдання дослідження:

1. Провести огляд сучасних методів виявлення фішингових атак та дослідити їхні недоліки.

2. Визначити ключові характеристики та вразливості, які використовуються у фішингових атаках.

3. Розробити критерії для аналізу мережевого трафіку, що можуть свідчити про можливі фішингові атаки.

4. Вивчити різні інструменти та технології для детектування фішингових активностей у мережевому трафіку.

5. Розробити алгоритм для виявлення та аналізу фішингових атак.

6. Провести тестування розробленого алгоритму на реальних даних і оцінити його ефективність.

7. Сформулювати рекомендації для підвищення рівня безпеки інформаційних систем на основі результатів аналізу.

8. Розробити інструкції для фахівців з інформаційної безпеки щодо моніторингу мережевого трафіку та виявлення фішингових атак.

Об'єкт дослідження: процес виявлення фішингових атак в інформаційних системах.

Предмет дослідження: методи та інструменти для аналізу мережевого трафіку з метою виявлення фішингових атак.

Методи дослідження. У роботі використані методи аналізу, синтезу, порівняння, а також емпіричні методи для тестування запропонованих рішень.

Наукова новизна одержаних результатів полягає у розробці та впровадженні інноваційної системи захисту від фішингових атак, спеціально адаптованої для соціальних мереж. Система включає вдосконалені методи аналізу мережевого трафіку з використанням Wireshark, що дозволяє виявляти та блокувати підозрілу активність у реальному часі. Запропоновані підходи до фільтрації та аналізу даних дозволяють швидко ідентифікувати фішингові атаки, забезпечуючи високий рівень безпеки користувачів соціальних мереж. Важливою складовою наукової новизни є інтеграція розробленої системи з існуючими інструментами моніторингу та аналізу, що підвищує її ефективність та знижує час реакції на інциденти безпеки.

Практичне значення одержаних результатів. Запропоновані рекомендації можуть бути використані в практичній діяльності фахівців з інформаційної безпеки для підвищення рівня захисту інформаційних систем від фішингових атак.

Структура роботи. Дипломна робота складається зі вступу, трьох розділів, висновків, списку використаних джерел та додатків.

Ключові слова: фішинг, фішингові атаки, соціальна інженерія, Інтернет-захист, системи заххисту, соціальні мережі, Wireshark

РОЗДІЛ 1
ТЕОРЕТИЧНІ ЗАСАДИ СИСТЕМИ ЗАХИСТУ ВІД ФІШИНГУ В ІНТЕРНЕТІ

1.1 Історія розвитку фішингу

Алгоритми фішингу вперше були детально описані в онлайн-стрічці новин американської медіа-корпорації AOL Inc. ("America Online") у 1987 році (термін з'явився після 2 січня 1996 року)[4].

У той час група шахраїв використовувала цю інтернет-платформу для розповсюдження піратського програмного забезпечення, проведення незаконних операцій з кредитними картками та інших онлайн-злочинів.

У 1995 році після того, як керівники AOL Inc. вжили заходів для боротьби з використанням підроблених номерів кредитних карток, злочинці стали агресивно використовувати фішинг, щоб отримати доступ до (справжніх) облікових записів інших людей[4].

Зловмисник представляється співробітником компанії та зв’язується з потенційними жертвами через миттєві повідомлення, запитуючи логін і паролі користувачів. Вони використовують такі вказівні фрази, як «Підтвердіть обліковий запис» і «Підтвердіть платіжну інформацію», щоб зміцнити довіру. Якщо відповідь позитивна, зловмисник отримав доступ до даних жертви та використав обліковий запис жертви для шахрайських цілей (незаконні фінансові операції, спам тощо)[4].

Ця «перша» афера була настільки популярною, що AOL Inc. змушений був додати таке речення: «Жоден співробітник AOL Inc. ніколи не спробує дізнатися Ваші логін і пароль або платіжну інформацію».

З 1997 року група зайняла рішучу позицію проти шахрайства та розробила систему запобігання роботі шахрайських облікових записів[4].

В результаті цих незаконних дій (взяття під контроль облікових записів користувачів AOL Inc., дало їм доступ до даних кредитних карток) злочинці зрозуміли технічну слабкість платіжної системи та психологічну вразливість користувача.

Перша відома спроба атаки на систему електронних платежів золота була зареєстрована в червні 2001 року, а друга атака відбулася після терористичних атак 11 вересня. Ця перша експериментальна атака вважається лише тестом для перевірки потенційних можливостей[5].

З 2004 року фішинг став загрозою номер один для бізнесу в усьому світі та продовжує зростати та збільшувати свій потенціал [4]. Наразі основними цілями шахраїв є клієнти банків та електронних платіжних систем.

У США хакери, які видавалися за податківців, зібрали велику кількість даних платників податків. Перші повідомлення надсилалися у випадковому порядку в надії досягти потрібного банку чи клієнта, який користується послугою[4]. Тепер шахраї можуть дізнатися, які служби використовують їхні жертви для надсилання цільових електронних листів. Кілька недавніх фішингових атак були спрямовані безпосередньо на керівників компаній та інших осіб на ключових посадах.

Соціальні мережі, які дозволяють збирати особисті дані користувачів, також викликають велику стурбованість шахраїв. У 2006 році за допомогою хробака (шкідливої інтернет-програми, яка самостійно розробилася та поширилася у кіберпросторі) на MySpace було розміщено велику кількість посилань на фішингові сайти, спрямовані на крадіжку записів даних[4].

Розглянемо коротку хронологію основних подій на рис.1.1 нижче.

Рисунок 1.1 Еволюція фішингу з 1987 до сьогодні"

Згідно з дослідженнями експертів, понад 70% фішингових атак у соціальних мережах є успішними[4].

Щороку фішингові атаки коштують мільярди доларів.

1.2 Теоретичний засади формування фішингу

1.2.1 Характерні особливості фішингу

Фішинг можна визначити як різновид соціальної інженерії, в якій метою зловмисника є отримання особистої інформації про жертву або встановлення шкідливого програмного забезпечення в системі жертви[5].

Це робиться шляхом обману змусити користувачів натиснути зловмисне посилання в електронному листі, яке перенаправляє користувача на підроблений веб-сайт, створений з єдиною метою збору конфіденційної інформації, такої як облікові дані автентифікації[5].

Ці підроблені фішингові веб-сайти часто нагадують добре відомі веб-сайти, такі як: веб-сайт банку, соціальні мережі та навіть веб-сайт цільової організації. Зловмисник робить це, щоб ціль не сумнівалася, що веб-сайт не підроблений. У деяких випадках явно підроблений веб-сайт може відображатися з неправильним доменним іменем або граматичними помилками.

Типові фішингові атаки масово надсилаються сотням і навіть тисячам одержувачів[5].

Ймовірність успіху простої фішингової атаки дуже різна. Щоб досягти вищих показників успіху, зловмисники можуть використовувати

цілеспрямовані фішингові атаки проти конкретних організацій, осіб або компаній.

Цільовий фішинг — це більш цілеспрямований тип фішингової атаки. Ця афера має свої цілі та свої жертви. У цільовому фішингу зловмисник витрачає час на дослідження та збір великої кількості інформації про ціль[3].

Ця інформація використовується, щоб зробити атаку більш легітимною та релевантною, змусивши одержувача розкрити інформацію або встановити шкідливе програмне забезпечення на своєму комп'ютері.

У цих цілеспрямованих фішингових атаках зловмисники можуть надсилати електронні листи, які виглядають легітимними, містять очікувані логотипи або навіть блоки підпису. Навіть шкідливі посилання та вкладення можуть бути замасковані під законні[5].

Цільові фішингові атаки користувачам важко виявити та пом'якшити, оскільки вони складні за дизайном і маскуванням.

У цьому випадку попереднє текстове повідомлення використовується для створення складного сценарію, щоб переконати цільову жертву розкрити інформацію або виконати дії, які зловмисник може використати на свою користь[3]. Тоді претекстинг дає перевагу соціальному інженеру.

Якщо зловмисник може надати достатньо правдивої інформації у фішинговому електронному листі з законного та надійного джерела, його шанси на успіх значно зростають.

Фішинг також можна розуміти як широкий термін, оскільки багато досліджень розглядають різні аспекти, засоби та стратегії. Типи фішингових атак, які найчастіше досліджуються, — веб-сайт і електронна пошта. Однак шахрайство також може відбуватися в багатьох інших медіа, таких як соціальні мережі та інші служби обміну повідомленнями[4]. Там, де є можливість цифрового спілкування, часто існує ймовірність фішингу.

Технічні рішення для зменшення загрози фішингу не завжди ефективні, оскільки вони покладаються на здатність кінцевого користувача знати, як

інтерпретувати кожне повідомлення та визначати, які повідомлення є шкідливими.

Фішинг варіюється від загального фішингу до цільового фішингу. Фішинг — це цілеспрямована комунікаційна атака, у якій контекст фішингового повідомлення є специфічним для певної цілі й тому не пов’язаний з іншими цілями[6]. Фішинг зазвичай відноситься до повідомлень, які націлені на ширшу аудиторію та не підходять для всіх цілей.

Водночас існує ще одна підкатегорія фішингу, яку ми можемо назвати «китобійний фішинг». Це той випадок, коли фішингові атаки спрямовані на керівників великих компаній. У разі зламу ці цілі розміром із кита можуть надати зловмисникам значні фінансові вигоди або додатковий доступ до їх організації[6].

Підсумовуючи, можна сказати, що фішинг — це унікальна загроза кібербезпеці, яка включає в себе поєднання технічних вразливостей і психологічних, соціологічних і комунікаційних ефектів[6].

1.2.2 Схема побудови фішингу

Тепер подивимося, як працюють фішингові веб-сайти. Приклад схеми фішингового веб-сайту показано на рис.1.2 нижче.

Рисунок 1.2 – Функціональна схема фішингового сайту

Отже, коли йдеться про створення фішингового веб-сайту, на цій діаграмі є три різні об’єкти, як показано на рис. 1.2. Це клієнт, шкідливий проксі-сервер зловмисника та справжній сайт.

Таким чином, було приховано всі введені користувачами данні після входу у систему на цьому проксі-сервері від зловмисника. Коли дані зареєстровані або отримані проксі-сервером зловмисника, проксі-сервер перенаправляє клієнта на справжній веб-сайт і відображає повідомлення «Ваш пароль неправильний» або «Сталася помилка. Повторіть спробу»[7].

Тому в більшості випадків фішингові сайти маніпулюють не тільки логінами та паролями, але й реквізитами банківських карток. Якщо ви введете всю інформацію, включаючи код CVC, на сайті з’явиться повідомлення про помилку, але вся введена при цьому інформація буде зареєстрована.

Що стосується фішингових атак, вам потрібно розуміти їх структуру. Розглянемо таблицю 1.1 побудови фішингової атаки:

Тип атаки	**Час життя (години)**	**Складність**	**Методи розповсюдження**	**Примітки**
Традиційний	62	Низька	Підроблена URL-адреса	Легко виявити і заблокувати
Rock-Phish	172	Середня	Короткі доменні імена, загальні сервери доменних імен	Використовує проксі-боти, що ускладнює виявлення
Fast-Flux	196	Висока	Динамічні IP-адреси, часті зміни	Найбільш складна і стійка до блокування

Таблиця 1.1 Порівняння різних видів фішингових атак

Давайте спочатку розглянемо традиційну схему. Ця схема зображена на рис.1.3 нижче.

Рисунок 1.3 – Діаграма традиційної фішингової атаки

Як видно на рис 1.3, схема складається з трьох елементів і трьох різних стрілок.

Перша стрілка – підроблена URL-адреса. Цю підроблену URL-адресу потрібно якимось чином повідомити жертві. Якщо жертва натисне цю URL-адресу, її буде переспрямовано. Спочатку URL-адреса розпізнається DNS-сервером, потім дозволена IP-адреса вказує на сервер зловмисника, а потім користувачеві повертається підроблена веб-сторінка[8].

Це відносно проста традиційна фішингова атака, яка не дуже складна та не потребує багато інструментів для виконання цієї атаки. Але більше того, вони вразливі до фішингових атак, оскільки їх можна легко вимкнути.

Щоб нейтралізувати такі атаки, ви повинні повідомити свого Інтернет-провайдера про те, що IP-адреса містить підроблену веб-сторінку або фішинговий веб-сайт, і ваш Інтернет-провайдер може провести дослідити та видалити цю IP-адресу[8]. Після цього фішинговий сайт перестає працювати.

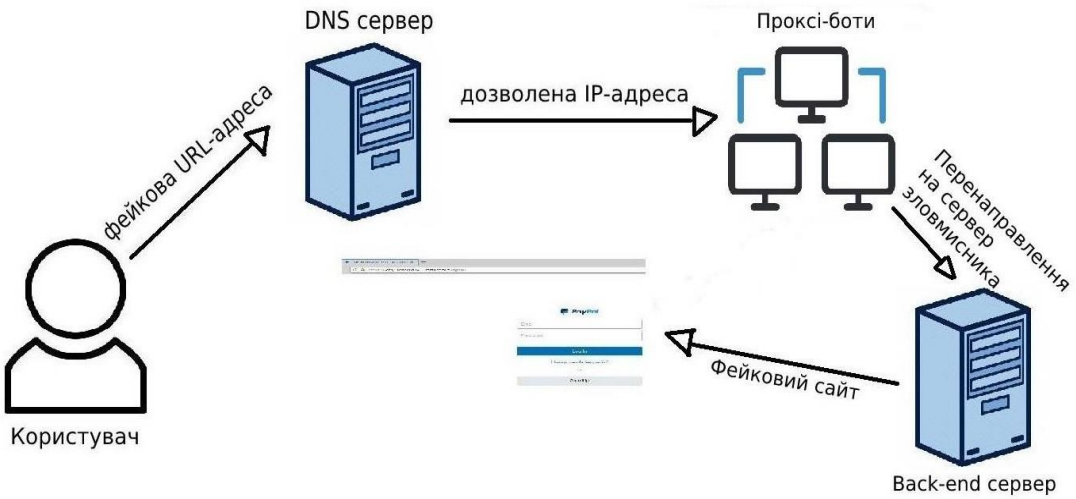


Рисунок 1.4 – Схема атаки Rock Phish

Тепер розглянемо схему Rock Phish. На рис. 1.4 вище, можна побачити, що Rock Phish є більш складною атакою. Ця стратегія атаки намагається уникнути виявлення фішингового веб-сайту та повністю використати його вразливі місця.

Таким чином, додавання надмірності ізолює елементи атаки, перед тим як вимагати видалення. Зловмисник починає з обробки низки коротких і здебільшого безглуздих доменних імен, таких як "loh80.info".

Потім, на етапі розповсюдження, ви отримуєте із цією довгою URL-адресою. Він поширюється, і перша частина показує, що цей веб-сайт є популярним.

Такі механізми, як загальні сервери доменних імен із підстановкою (DNS), можна використовувати для вирішення таких варіацій даної IP-адреси, які, у свою чергу, вказують на проксі-бота[8].

Таким чином, кожен домен, що міститься на цьому внутрішньому сервері, призначається динамічній групі скомпрометованих машин на основі серверів імен, контрольованих зловмисником[6].

Кожна скомпрометована машина запускає проксі-систему, яка підключає запити до серверної системи. Тоді цей внутрішній сервер завантажується багатьма фішинговими сайтами одночасно, іноді до 20 фішингових сайтів. І всі ці сайти доступні з будь-якої машини Rock Phish.

Однак, сторінка на яку ви потрапите, залежить від URL-адреси, переданої після першої косої риски.

Крім того, сайти Rockfish мають спільний хост.

Отже, якщо один із цих сайтів буде видалено, один із сайтів буде повідомлено, а його внутрішня IP-адреса сервера стане недійсною, сайт автоматично перемкнеться. Тому, щоб позбутися Rock-Phish, потрібно видалити всіх проксі-ботів[8].

Ця атака значно надійніша за традиційний фішинг з точки зору вразливостей.

Однак, якщо ви подивитеся на цих проксі-ботів, вони завжди мають певний набір IP-адрес, і якщо ви зможете ідентифікувати всіх, які мають до них доступ, наприклад, внутрішні сервери, ви можете перемогти цей тип фішингової атаки.

Однак існує також більш складна техніка, відома як «Fast-Flux Phishing». Це показано на рис 1.5 нижче.

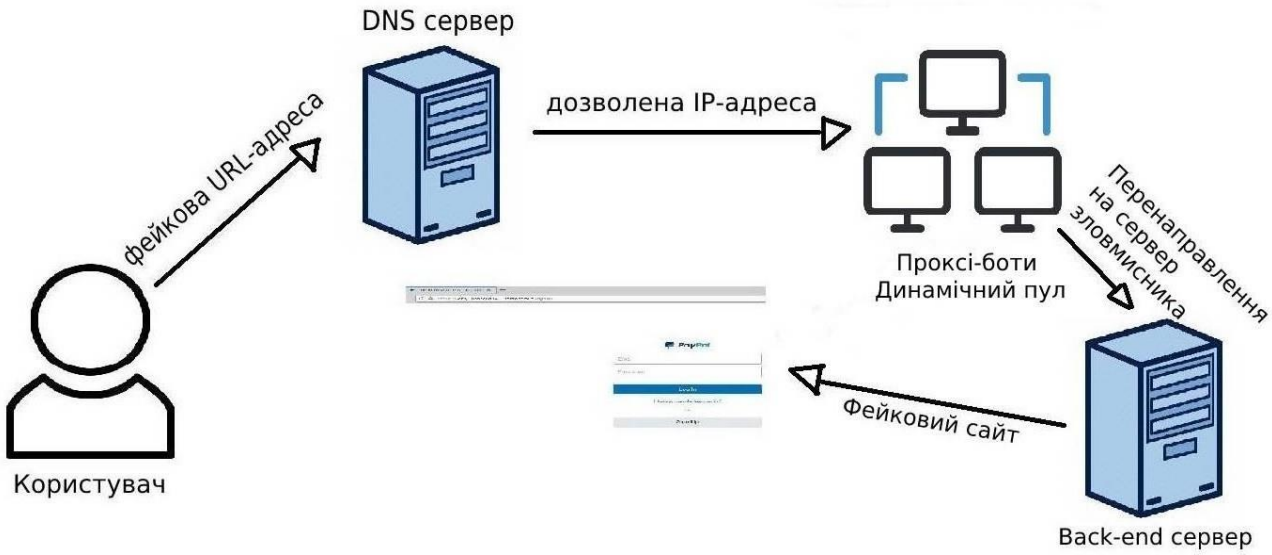


Рисунок 1.5 – Принципова схема атаки Fast-Flux

Отже, як показано на рисунку 1.5, основна ідея мережі Fast-Flux полягає в тому, щоб пов’язати кілька IP-адрес з доменним іменем і підтримувати їх швидку зміну.

Таким чином, комп’ютери в системі фішингу Fast-Flux, які утворюють цей тип мережі, насправді не відповідають за розміщення та завантаження шкідливого вмісту жертв. Завдання полягає в тому, щоб визначити кілька машин, які діють як хости для цього шкідливого вмісту[8].

Інші системи просто діють як перенаправлення, щоб допомогти приховати справжні фізичні адреси цих систем, від кримінального контролю. Зловмисники забезпечують максимальну доступність критично важливих систем у мережі та використовують балансувальники навантаження для обробки всіх запитів на завантаження шкідливого контенту.

Поширеним методом є перегляд стану мережі на загальному екрані брандмауера, щоб відкинути невикористані нотатки та переконатися, що шкідливий вміст усе ще є активним і завантажується.

Тому, порівнюючи фішингову схему Fast Flux із схемою Rock Phish, ми бачимо, що ця група динамічних проксі-ботів ускладнює роботу таких фішингових мереж. Це головним чином тому, що проксі-боти постійно розвиваються та змінюються.

Навіть якщо IP-адресу ідентифіковано та ви спробуєте її видалити, якщо один із цих проксі-ботів буде знесено, інша IP-адреса буде зареєстрована на його DNS-сервері.

Є цікавий приклад лавинної мережі. Це мережа, яка використовується як платформа доставки для запуску та управління глобальними атаками зловмисного програмного забезпечення та фішинговими компаніями[6]. Наприклад, у Німеччині збитки, завдані цією стрімкою фішинговою мережею після інтенсивної кібератаки на онлайн-банкінг, оцінили приблизно в 6 мільйонів євро. Ця фішингова мережа була закрита лише за рішучої підтримки прокурорів і слідчих з 30 країн. Затримано 5 осіб, обшукано 37 об'єктів, вилучено 39 серверів.

Тож цей приклад демонструє, скільки зусиль необхідно докласти для боротьби з такою швидкою фішинговою мережею.

Зауважте також, що існують різні типи високошвидкісних фішингових мереж. Фішингова мережа Fast Flux містить деяку унікальні нотатками про реєстрацію та скасування реєстрації IP-адрес як частини DNS-адреси одного доменного імені[7].

Ці реєстратори мають дуже короткий термін служби, в середньому 5 хвилин, і генерують постійно мінливий потік IP-адрес при спробі отримати доступ до певного домену. І є подвійні фішингові мережі Fast-Flux. Цей тип мережі використовує компоненти методу зв'язку між системою жертви та системою, контрольованою зловмисником.

Ddos системи в цій мережі є частиною ботнету та використовуються як проксі-сервери, щоб запобігти прямій взаємодії жертв із серверами, на яких розміщено шкідливе програмне забезпечення та фішингові веб-сайти.

1.3 Різновиди фішингових атак

Фішингова атака – це атаки, під час якої фішинговий сайт видає себе за законний веб-сайт, щоб викрасти конфіденційну інформацію.

Ця дія відбувається, коли зловмисник надсилає SMS або електронний лист зі шкідливою URL-адресою, обманом змушуючи жертву відкрити шкідливе посилання та перенаправляючи її на підроблений веб-сайт, який дуже схожий на справжній веб-сайт. Якщо жертву обманом змусять розкрити конфіденційну інформацію, зловмисник може використати цю інформацію для проведення атак соціальної інженерії або викрадення інформації пов'язаної з обліковим записом жертви.

Існує багато різних типів фішингових атак, коли зловмисник може націлитися на велику кількість цілей або лише на невелику кількість ретельно відібраних цілей, відомих як фішингова атака. Однак перш ніж зловмисники зможуть розпочати фішингову кампанію, вони повинні спочатку налаштувати

інфраструктуру для розміщення та розповсюдження даних або фішингових сайтів[5].

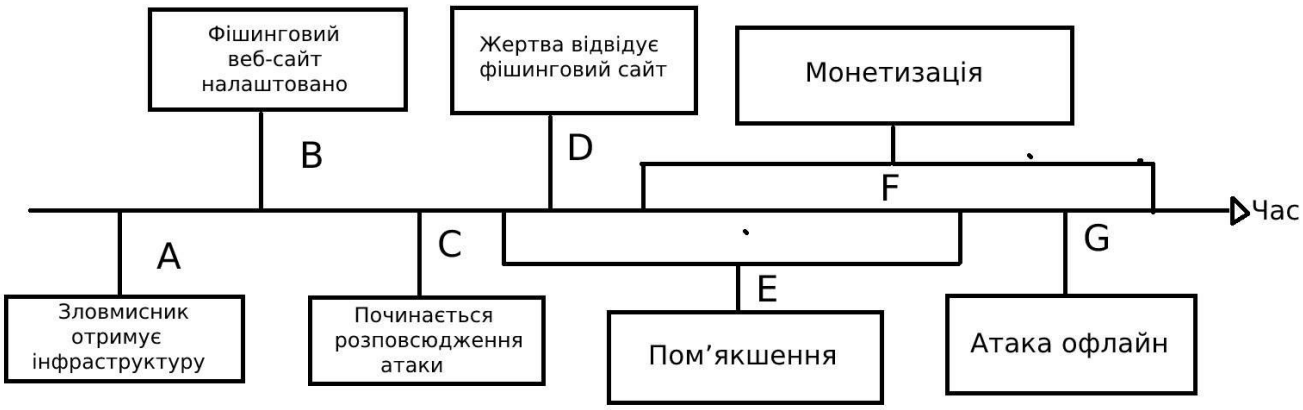


Рисунок 1.5 – Етапи високого рівня фішингової атаки

На малюнку 1.5 показано, що зловмисник спочатку отримує інфраструктуру (A) і створює фішинговий веб-сайт. Вони зазвичай мають набори для фішингу в даній інфраструктурі (B). Це використовується для збору облікових даних або надання зловмисного ПЗ для завантаження.

Коли веб-сайт працює, зловмисник починає розповсюджувати його жертвам, зазвичай електронною поштою (C). Потім жертва починає заходити на веб-сайт (D).

Залежно від того, чи може організація виявити фішингову кампанію, націлену на співробітників, наприклад, через звіти користувачів, організація може пом'якшити атаку (E). У найкращому випадку пом'якшення буде застосовано до (D), якщо користувач ще не відвідував фішинговий сайт, таким чином блокуючи майбутній трафік жертви.

Однак, якщо нічого не працює, це дає зловмиснику деякий час, щоб використати викрадені дані (F), щоб монетизувати атаку або закріпитися в мережі.

Зловмисники можуть видаляти або знищувати фішингові веб-сайти (G). Але в цьому випадку, як тільки зловмисник отримує дані, що впливають

на бізнес, будь то облікові дані чи доступ до мережі, він продовжує монетизувати свої дані, навіть якщо вихідну інфраструктуру закрито[4].

Беручи до уваги життєвий цикл фішингової атаки, він зазвичай складається з 3 періодів: період підготовки, активний період і неактивний період.

Фішингові атаки починаються з періоду підготовки та стають активними, коли фішингові посилання публікуються в соціальних мережах або надсилаються електронною поштою. Після виявлення або блокування атаки цей період стає неактивним. При виявленні або блокуванні атак цей період стає неактивним, однак це не означає, що атаки припинилися. Зловмисник може внести ряд змін, а потім продовжити атаку[4].

Далі ми розглянемо можливі фішингові атаки.

Спочатку розглянемо найпопулярнішу атаку «фішинг електронної пошти». Цей тип фішингу зазвичай називають «обманним фішингом» і він є одним із найвідоміших видів атак. По-перше, зловмисники надсилають електронні листи користувачам, видаючи себе за відомі бренди, і використовують тактику соціальної інженерії для підвищення ефективності[8,9]. Далі користувачеві буде запропоновано завантажити ресурс або натиснути на посилання. Ці посилання ведуть на шкідливі веб-сайти, які викрадають облікові дані або встановлюють шкідливий код (зловмисне програмне забезпечення) на пристрої користувача.

Щодо файлів, завантажені файли (зазвичай PDF-файли) містять шкідливий вміст, який встановлюючи зловмисне програмне забезпечення, щойно користувач відкриває документ.

Давайте розглянемо ознаки фішингового електронного листа.

1) Справжня інформація – у цьому випадку ви можете знайти контактну інформацію та іншу надійну інформацію про фальшиву організацію. Тоді помітите, що адреса електронної пошти відправника вказано в неправильному домені або написана з помилкою[8].

2) Шкідливий і безпечний код – будьте обережні з кодом, який намагається обдурити Exchange Online Protection (EOP). Наприклад, неправильно написані посилання або завантаження.

3) Скорочені посилання – рекомендуємо не натискати скорочені посилання, оскільки вони використовуються для обману безпечних шлюзів електронної пошти.

4) Підроблені логотипи брендів. Ці логотипи можуть містити підроблені та шкідливі атрибути HTML, тому вам слід шукати справжні логотипи у своїх повідомленнях.

5) Низький текст – електронні листи, які містять лише зображення та дуже мало тексту, слід ігнорувати, оскільки за зображеннями може ховатися шкідливий код[8].

Перейдемо до «HTTP fishing». Захист протоколу передачі гіпертексту (HTTPS) вважається безпечним посиланням, оскільки використовує шифрування для додаткової безпеки[9]. Багато законних організацій використовують HTTPS замість протоколу передачі гіпертексту (HTTP), оскільки це дозволяє їм встановити легітимність. Однак тепер зловмисники можуть використовувати HTTPS у посиланнях, які вони включають у фішингові електронні листи. Ця техніка є частиною фішингової атаки електронної пошти, але це дещо модернізований підхід.

Давайте розглянемо деякі ознаки, за якими можна розпізнати фішинг з допомогою захищеного протоколу передачі гіпертексту (HTTPS).

1) Гіпертекст — це інтерактивне посилання, вбудоване в текст, щоб приховати фактичну URL-адресу[9].

2) Скорочення посилання. Ви повинні переконатися, що це посилання має правильний формат довгого хвоста, а також містить усі частини URL-адреси[10].

І на кінець розглянемо атаку «Спливаючого фішингу».

Хоча багато людей використовують блокування спливаючих вікон, такі як «AdBlock», зловмисники можуть розмістити зловмисний код у невеликих

вікнах повідомлень, які з’являються під час відвідування веб-сайту. Новіші версії «спливаючого фішингу» використовують функції сповіщень браузера.

Наприклад, коли користувач відвідує веб-сайт, браузер відображає користувачеві «www.buysomestuff.com хоче показати вам повідомлення».

Якщо на цьому етапі користувач натисне кнопку «Дозволити», у спливаючому вікні буде встановлено шкідливий код.

Ознаки цього фішингу включають:

1) Повноекранний режим – автоматична зміна розміру екрана може бути індикатором, оскільки зловмисні спливаючі вікна зазвичай можуть спричинити перехід браузера в повноекранний режим[9].

2) Невідповідності – слід перевірити на наявність орфографічних помилок і нестандартних схем кольорів[12].

Це лише деякі з багатьох відомих фішингових атак.

1.4 Аналіз існуючих методів захисту від фішингових атак

У попередньому розділі було розглянуто деякі з основних фішингових атак. У цьому розділі докладніше розглядаються методи, якими можна скористатися для захисту від цих цілеспрямованих фішингових атак.

Захист від фішингу включає різноманітні передові інструменти і методи для виявлення та нейтралізації фішингових атак. Це також включає розширене навчання користувачів, встановлення спеціалізованих антифішингових рішень, інструментів і програм, а також впровадження інших заходів безпеки для захисту від фішингу, а також надання методів мінімізації впливу порушень безпеки. Це також може включати цільове та всебічне навчання користувачів.

Загалом фішингові атаки є більш поширеними в службах електронної пошти. Щоб захиститися від цієї атаки, ви можете навчити користувачів

розпізнавати спроби фішингу, такі як підозрілі адреси електронної пошти, прості привітання та незвичні повідомлення. Електронна пошта або миттєві повідомлення - повідомлення про граматичні помилки чи посилання за межами організації. Мета – викликати паніку та оперативні дії.

У цьому випадку ви також можна використати сегментацію мережі, для обмеження доступу до високозахищених. Це ускладнює проникнення фішингових атак. Тоді необхідний аудит середовища кібербезпеки для оцінки вразливостей, виявлення нових загроз і розробки стратегій захисту[8].

Найефективнішим методом захисту є встановлення технології, яка сканує всі вхідні електронні листи в режимі реального часу, не дозволяє користувачам переходити за посиланнями на підозрілі веб-сайти, пісочниці та сканувати всі вкладення на наявність потенційних загроз та виявити підозрілі URL-адреси до них[12].

Загрози електронної пошти розвиваються та потребують все більш потужних технологій для запобігання фішингу електронної пошти та захисту організацій від цілеспрямованих фішингових атак.

Багато компаній намагаються запобігти фішинговим електронним листам, навчаючи співробітників розпізнавати підозрілі електронні листи. Але незважаючи на те, що співробітники навчаються найпоширенішим методам фішинг, майже чверть фішингових листів все одно відкривається. Тобто, щоб справді запобігти даним атакам, ефективним захистом може бути технологія, яка нейтралізуватиме людські помилки та автоматично запобігати фішинговим атакам на службу електронної пошти користувача.

Іншим методом захисту від цих атак є вибір багаторівневого захисту для запобігання фішинговим атакам.

Оскільки фішингові шахрайства електронною поштою продовжують підривати захист, все більше організацій застосовують багаторівневий підхід до своїх стратегій безпеки, щоб запобігти атакам.

Крім того, інструменти автентифікації DNS, які використовують протоколи DMARC, SPF і DKIM для виявлення та блокування підозрілих

повідомлень, можна використовувати для захисту від фішингових атак електронної пошти[13]. Ви також можете використовувати протоколи двофакторної автентифікації, які не дають зловмисникам отримувати та використовувати викрадену інформацію, щоб отримати доступ до облікового запису користувача.

Тепер давайте розглянемо захист від фішингових атак HTTPS. У цьому випадку веб-сайт не вважається надійним, навіть якщо в адресному рядку відображається значок замка або HTTPS.

Також важливо переконатися, що URL-адреса вашого сайту точна. Наприклад, ви можете перевірити на наявність помилок або недійсних імен доменів, як-от домени .net(зазвичай домени .com). Рекомендується вводити URL-адресу веб-сайту, який користувач хоче відвідати, безпосередньо у своєму браузері, а не переходити за посиланням, отриманим в електронному листі.

Можна захистити себе від таких атак, використовуючи налаштуваннями таких інструментів, як менеджер паролів або програмне забезпечення безпеки.

Ці інструменти можуть включати функції, які попереджають користувачів, коли URL-адреса не збігається з законним веб-сайтом, або заважають відкривати фішингові веб-сайти. У більшості випадків увага користувача дуже допомагає в цих атаках. Тому, якщо користувач отримує підозрілий електронний лист із посиланням, необхідно сповістити його, зателефонувавши або написавши електронного листа безпосередньо цій особі та запитати, чи вона його надіслала чи ні.

У цьому випадку також може бути корисним уникнення використання сертифікатів із символами підстановки у виробничих системах, але це збільшує ризик і поверхню атак, якщо сервер або сертифікат зламано. Крім того, щоб захистити себе, компанія може мати навчальний розділ з кібербезпеки, який присвячений фішинговим атакам і вимагати перевірки

того, що цей розділ безпосередньо стосується шахрайства з підключенням до HTTPS[14].

Щоб краще захистити свою компанію від фішингових атак за допомогою сертифікатів SSL, важливо ознайомити своїх співробітників з кіберзагрозами HTTPS. Якщо користувачі не впевнені, чи веб-сайт є законним, вони можуть легко знайти доменне ім’я за допомогою пошуку «whois». Цей метод може бути використаним для перевірки на законність веб-сайту.

На сьогодні захист від фішингових сайтів інтегрований в сучасні браузери, і вони добре виконують своє завданням. Більшість антивірусних продуктів і пакетів безпеки мають власні функції захисту від фішингу[15]. Крім того, ви можете використовувати менеджер паролів, щоб захиститися від цієї атаки. Цей метод допомагає захистити користувачів від атак. Більшість із цих продуктів дозволяють користувачеві отримати доступ до закхищеного веб-сайту та ввійти на нього лише одним клацанням миші.

Тепер давайте розглянемо, як ви можете захистити себе від «цілеспрямованого фішингу». Ці методи включають захист електронної пошти для відстеження фішингових атак.

Компанії повинні впроваджувати передові рішення безпеки електронної пошти, включаючи безпечні шлюзи електронної пошти, які блокують зловмисне програмне забезпечення, спам і більшість атак електронною поштою[16]. Однак у цьому випадку фішингові атаки зазвичай не містять зловмисного програмного забезпечення, тому такі заходи безпеки, як DMARC («Доменна автентифікація пошти»), яка виявляє фішингові атаки шляхом перевірки автентичності електронних пошти за базою даних відправника. Для захисту в цьому випадку, можна використовувати шифрування конфіденційної інформації.

Коли компанія шифрує свої дані, зловмисники не можуть отримати доступ до будь-якої конфіденційної чи цінної інформації. У цьому випадку ви також можете використовувати багатофакторну автентифікацію. Крім

введення пароля, для входу в систему потрібні додаткові кроки, що ускладнює доступ зловмисників до ваших даних і систем[17].

Також регулярне оновлення системи безпеки зробить ваше перебування в інтернет просторі більш безпечним. Оновлення вашої системи безпеки до останніх версій допоможе блокувати будь-які відомі області, які можуть бути під загрозою[2].

Щоб захистити себе, ви можете організувати інструктаж техніки безпеки для своїх співробітників[18].

Ще один спосіб захистити себе від «цільового фішингу» - це використання першокласної служби безпеки електронної пошти, яка запобігає проникненню фішингових електронних листів в інфраструктуру електронної пошти компанії[1].

Щоб покращити захист від цілеспрямованого фішингу, багато компаній намагаються просвітити користувачів про небезпеку цих електронних листів, проте майже чверть усіх фішингових листів все ще відкривається.

Оскільки на сьогодні понад 90% усіх атак починаються з фішингових електронних листів, стає очевидним, що компаніям потрібен відмінний захист від фішингу, щоб захистити користувачів і свою компанію[1,19].

Давайте тепер розглянемо способи захисту від «китобійного фішингу». У цьому випадку створення протоколів для запобігання «китобійного фішингу» служить захистом. Типові приклади включають перевірку запитів конфіденційної інформації через інші канали, такі як телефонні дзвінки. Ви також можете використовувати програмне забезпечення DLP[19].

Програмне забезпечення «Data Leak Prevention» (DLP) може блокувати порушення встановлених протоколів. Ви також можете позначати електронні листи на основі домену чи віку, переглядати імена схожих відомих контактів і бачити підозрілі ключові слова, як-от «банківський переказ»[10].

Щоб захистити себе від таких атак, можна використовувати хороші програми для захисту від спаму та зловмисного програмного забезпечення, які

можуть запобігти атакам на шлюз електронної пошти для деяких електронних листів[20].

Крім того, можна використати служби перевірки DNS за допомогою DMARC, DKIM і SPF, щоб визначити, чи є листи, надіслані з домену, законними чи підробленими[20].

Іншим способом захисту від цієї атаки є використання програмного забезпечення для захисту від несанкціонованого доступу. Це програмне забезпечення може блокувати китобійні атаки, виявляючи методи соціальної інженерії, які зазвичай зустрічаються в «китобійних» електронних листах[17].

Щоб запобігти цій атаці, ви також можете використовувати технології захисту даних, які вимагають впровадження методів обробки даних, які ускладнюють завдання шкоди людині. Використовуйте елементи керування доступом, щоб обмежити root-доступ до бажаного рівня. Крім того, ви повинні переконатися, що всі дозволи користувача відповідні та необхідні для кожної посадової функції. Впровадити та підтримувати загальний набір інструментів IT-безпеки, брандмауерів, програмного забезпечення для виявлення вторгнень і утиліт для сканування шкідливих програм[17].

Давайте тепер розглянемо методи захисту, які можуть допомогти запобігти атакам «фішингових спливаючих вікон».

Щоб захистити себе в цьому випадку, ви також можете привернути увагу користувача. Крім того, налаштування браузера можуть забезпечити захист від цієї атаки. Для того, щоб ця атака не турбувала користувачів, потрібно посилити налаштування безпеки браузера. У цьому випадку антивірусне програмне забезпечення або комплексне рішення безпеки в Інтернеті може забезпечити належний захист[1].

У деяких випадках ви можете отримувати «фішингові спливаючі повідомлення» на певних сайтах, навіть якщо на комп’ютері встановлено антивірусне програмне забезпечення. Зазвичай це означає, що заражений не комп’ютер, а веб-сайт, який відвідує користувач. У цьому випадку необхідна увага користувача[1].

Узагальнюючи всі згадані фішингові атаки та методи захисту, можна зробити висновок, що найбільш вразливими є поштові служби та веб-сайти. Тому в цьому випадку слід розглянути, як виявити шахрайські повідомлення та веб-сайти.

Крім забезпечення безпеки, ці методи автоматизувати виявлення фішингових повідомлень і веб-сайтів, тим самим збільшуючи рівень виявлення.

Висновок до розділу 1

У першому розділі було докладно розглянуто історію виникнення фішингу, його еволюцію та основні теоретичні засади, що формують цю небезпечну кіберзагрозу. Виявлено, що фішинг як вид кіберзлочину зародився в 1990-х роках і з часом трансформувався, постійно адаптуючись до нових технологічних умов та соціальних середовищ.

Перші випадки фішингу були зафіксовані на платформі AOL Inc., де шахраї використовували підроблені облікові записи для незаконних операцій з кредитними картками та розповсюдження піратського програмного забезпечення. Надалі методи фішингу удосконалювалися, що дозволило зловмисникам проводити більш складні атаки, такі як цілеспрямовані фішингові атаки (спірінг) та фішингові атаки на керівників компаній (китобійний фішинг).

Фішинг визначено як різновид соціальної інженерії, метою якого є обманом змусити користувача розкрити конфіденційну інформацію або встановити шкідливе програмне забезпечення. Це досягається шляхом створення підроблених веб-сайтів, що нагадують легітимні платформи, або надсиланням електронних листів із шкідливими посиланнями чи вкладеннями. Такі атаки часто бувають масовими, проте значно небезпечнішими є

цілеспрямовані атаки, де зловмисник витрачає час на вивчення потенційної жертви, щоб зробити атаку більш переконливою.

Аналіз теоретичних засад фішингу показав, що технічні аспекти таких атак включають використання шкідливих проксі-серверів, методів перенаправлення та динамічних IP-адрес. Схеми побудови фішингових атак можуть варіюватися від простих традиційних методів до більш складних, таких як Rock Phish та Fast Flux Phishing. Ці методи дозволяють зловмисникам забезпечувати тривалу доступність фішингових веб-сайтів та ускладнюють їх виявлення і нейтралізацію.

Отже, фішинг є серйозною загрозою для сучасної кібербезпеки, оскільки поєднує технічні вразливості та психологічні аспекти впливу на користувачів. Ефективний захист від фішингових атак потребує комплексного підходу, який включає як технічні рішення, так і підвищення обізнаності користувачів щодо можливих ризиків. Важливим є не лише розуміння технологічних аспектів, але й розвиток навичок критичного мислення та уважності до деталей у цифровому середовищі.

РОЗДІЛ 2
СИСТЕМНИЙ АНАЛІЗ ФІШИНГОВИХ АТАК В СОЦІАЛЬНИХ МЕРЕЖАХ

2.1 Типи фішингових атак у соціальних мережах

Давайте розглянемо деякі з основних типів фішинговивих атак в соціальних мережах. Для початку ознайомимося з такою фішинговою атакою, як «кетфішинг».

1. «Кетфішинг (catfishing)» — це різновид онлайн-шахрайства, під час якого особа, також відома як **catfish (кетфішер),** створює фальшивий профіль у соціальній мережі чи на веб-сайті знайомств з метою шахрайства чи обману. Зазвичай метою зловмисників є фінансова вигода, але іноді методи «кетфішингу» використовуються для отримання конфіденційної інформації або приватного контенту[21].

Безліч профілів на сайтах знайомств і в соціальних мережах створюють ідеальне середовище для «кетфішингу». Шахраї часто користуються почуттям закоханості і роблять все можливе, щоб відносини дуже швидко розвинулися, щоб заплутати розум жертви. Проте є набір ознак, які дозволяють заздалегідь ідентифікувати профіль шахрая:

- ***Викрадені фото*. Використання чужої фотографії є серйозною ознакою того, що вас обманюють**. Наприклад, облікові записи, які містять лише професійні фотографії, будуть сприйматися з підозрою. На щастя, сьогодні існують такі інструменти, як зворотний пошук зображень Google, які дозволяють легко виявити, що фотографії були вкрадені. Щоб шукати схожі фотографії за допомогою Google, вам потрібно відкрити вкладку «зображення», натиснути маленький значок камери, щоб завантажити фотографію, або вставити її URL-адресу[22].

- ***Нещодавно створений профіль.*** На цей момент варто звернути увагу, якщо у профілю є значна кількість друзів або підписників, але в ньому все одно є щось підозріле. Звичайно, щойно створений профіль не означає, що користувач є новим на цьому сайті. Однак також може бути так, що шахрай часто створює нові профілі, щоб знайти нових жертв[23].
- ***Відсутність профілів у інших соціальних мережах.*** Сьогодні соціальні мережі є практично невід`ємною частиною життя більшості користувачів Інтернету. Не всі однаково активні на різних інтернет-платформах, але бувають випадки, коли це необхідно для кар'єрного росту. Наприклад, запитайте себе, чому успішний підприємець, якого ви щойно зустріли на Facebook, не має сторінки LinkedIn для просування свого бізнесу або пошуку нових партнерів і співробітників. Або чому у професійної моделі, яка спілкується з вами хусткою, немає прикладів своїх робіт в Instagram чи Facebook.
- ***Мала кількість друзів або підписників.*** ільшість людей використовують соціальні мережі для спілкування та спілкування з друзями та знайомими в Інтернеті. Тому природно мати багато друзів або підписників. У зв'язку з цим слід приділяти більше уваги користувачам з невеликою кількістю контактів. Звичайно, відсутність багатьох друзів може означати, що користувач новачок у соціальній мережі або просто не хоче мати багато контактів.
- ***Прохання надіслати гроші.*** Це одна з найяскравіших ознак того, що це «кетфішинг». Запитайте себе, чи ймовірно, що ви просите гроші в Інтернеті у когось, кого не знаєте, особливо якщо ви сподіваєтеся налагодити з ним стосунки.
- ***Уникнення зустрічей та відеодзвінків.*** Фішери, завжди намагаються уникати ситуацій, коли їхня справжня особистість може бути розкрита. Тому вони ніколи не погодяться на відеочат або особисту зустріч. Часто вони можуть відмовитися навіть від звичайних (не відео) дзвінків, тому що навіть їхній голос може розкрити їх особу.

- ***Прохання надіслати відверті фото або відео.*** Існує шахрайство під назвою «sextortion», яке зазвичай передбачає використання особистих документів, щоб погрожувати жертвам надати більш інтимний вміст або сплатити викуп.

2. Однією з атак, що впливають на цифрову екосистему, є «Angler Phishing». На відміну від традиційних фішингових атак, вони часто використовують підроблені електронні листи. Фішингові зловмисники діють як агенти служби підтримки. Вони використовують динамічний інтерфейс соціальної мережі, щоб обманом змусити користувачів надати конфіденційну інформацію або натиснути небажані посилання[24].

Враховуючи динамічний характер соціальних мереж, ризик кібератак сьогодні вищий, ніж будь-коли. Зараз кіберзлочинці використовують витончену тактику. Наприклад, зловмисник видає себе за представника служби підтримки або довіреної особи в соціальній мережі. Це робиться, щоб обманом змусити користувачів надати їм конфіденційну інформацію або встановити зловмисне програмне забезпечення[25].

Фішингові атаки «Angler» відрізняються від інших атак. Вони працюють, використовуючи взаємодію в реальному часі та довіру до соціальних мереж. Вони роблять це, видаючи себе за знайомі та авторитетні джерела. Кіберзлочинці користуються довірою користувачів до відомих організацій. Ця містифікація показує, що підприємствам необхідно покращити свою безпеку. Крім того, вони повинні інформувати користувачів про деталі цих атак[26].

Метою кіберзлочинців стають незадоволені клієнти компанії. Клієнти висловлюють невдоволення продуктами чи послугами в соціальних мережах. Зловмисники аналізують і відстежують ці гнівні повідомлення. Вони обирають користувачів, якими найімовірніше піддаються маніпулюванню через невдоволення[27].

Вони визначають свої потенційні цілі. Тоді вони діють як розуміючі представники підтримки. Здається, вони готові вирішити скарги клієнтів.

Обидві сторони продовжують спілкуватися у формі підтримки. Вони змушують жертв розкрити конфіденційну інформацію, таку як паролі або дані облікового запису, або переходити, здавалося б, на законні посилання, щоб отримати екстрену допомогу.

Отримавши цю конфіденційну інформацію, зловмисник розпочне більш небезпечні атаки. До них належать крадіжка особистих даних та фінансове шахрайство, а також інші злочини[7].

Розглянемо деякі типи фішингових тактик «Angler».

Видання за представника служби підтримки клієнтів. Однією з найпоширеніших тактик фішингу, яку використовують кіберзлочинці, є видавання себе за представника служби підтримки клієнтів. Цей підхід передбачає створення шахрайського фасаду. Він імітує реальну підтримку клієнтів від відомих брендів або організацій[8].

Після того, як пастку встановлено, жертву спонукатимуть розкрити конфіденційну інформацію або натиснути небажані посилання. Таким чином, цикл цифрового обману посилюється.

3. Шахрайство з криптовалютою – це система дій, яка включає різні оманливі та зловживаючі схеми, спрямовані на отримання несанкціонованого доступу до криптовалют або пов’язаних з ними коштів. Серед основних видів шахрайства можна виділити: обмінники, піратство гаманців та інші види кіберзлочинності.

Поширені випадки шахрайства з криптовалютою

Веб-сайти, що працюють з криптовалютою: Це підроблені біржі або інвестиційні платформи, які не мають децентралізованої економічної структури, але повністю контролюються злочинцями, а також зареєстровані облікові записи користувачів і кошти з метою викрадення більшої кількості грошей або криптовалюти у фізичних осіб.

Обмінники криптовалюти: Ненадійні обмінники криптовалюти переказують і конвертують валюту фізичних осіб з рахунків інших самозванців або жертв, які надали банківські картки (рахунки) для здійснення

цих операцій. Тобто відвідувач є анонімним на шкоду довіреним громадянам, які певним відсотком здійснюють операції з довіреними їм коштами.

Фішингові сайти (повні клони криптовалютних бірж, криптобірж, законних торгових платформ) і спам-атаки: Зловмисники надсилають жертвам електронні листи та повідомлення, які видають себе за офіційні повідомлення з бірж криптовалют або платіжних платформ. Потім людина натисне посилання та почне надавати свої дані (електронна адреса, телефон, пароль), гроші, криптоактиви злодієві, думаючи, що це реальні платформи для заробітку грошей або інвестування.

Жертвам намагаються пообіцяти значний дохід в обмін на особисті дані, такі як закриті ключі гаманця або інформацію, що дозволяє отримати доступ до фондової біржі для віддаленої торгівлі або використання аккаунт під % доходу.

Атаки на біржі: Зловмисники можуть використовувати біржі криптовалют або звичайні платформи грошових переказів для здійснення шахрайства. Зокрема, після завершення переказу вони звертаються до банку (обмінника) з повідомленням про ймовірне шахрайство з боку добросовісної особи з проханням повернути всі гроші, надіслані їм для купівлі криптовалюти від сумлінної особи. А жертва автоматично блокується до тих пір, поки не будуть надані всі докази та пояснення, а злодій отримає відшкодування, і більше того, він вже є власником криптовалюти.

Шахрайство з фальшивими гаманцями: Злочинці створюють фальшиві гаманці та програми, які генерують 10 000% доходу на місяць, і намагаються зняти гроші чи криптовалюту у нічого не підозрюючих зареєстрованих користувачів.

Користувачі повинні завжди перевіряти автентичність і історію гаманця або програми, перш ніж використовувати їх для зберігання криптовалюти або інвестицій.

Після того, як пастку встановлено, жертву спонукатимуть розкрити конфіденційну інформацію або натиснути на небажані посилання.

Інші злочини, під час яких звичайні гроші конвертуються в криптовалюту для захисту особи злочинця: Зловмисники, які користуються послугами онлайн обмінників котрі зараховують кошти на численні анонімні приватні гаманці, що належать невинним особам, з яких гроші виводять кошти на підставні рахунки аби зберегти анонімність.

Захист від даного типу атак полягає в:

Самостійній відповідальності:

- Користувачам слід бути обережними та уникати надання особистої інформації ненадійним джерелам.
- Зберігайте особисті ключі свого гаманця в надійному місці та не передавайте їх третім особам.

Безпека бірж:

- Обмінникам криптовалют необхідно покращити заходи безпеки за допомогою двофакторної автентифікації та інших технологій.
- Виконуйте регулярне тестування безпеки, щоб виявити потенційні вразливості.

Освітні кампанії:

- Освітні кампанії можуть розширити знання користувачів про ризики шахрайства та способи їх запобігання.
- Платформа може надавати рекомендації щодо заходів безпеки та вказівки щодо запобігання шахрайству.

Відповідальність:

- За шахрайство з криптовалютою передбачена кримінальна відповідальність за статтею 190 Кримінального кодексу України(див. додаток А1), відповідальність за такі дії передбачає до 12 років позбавлення волі з конфіскацією майна.
- Крім того, у разі таємного викрадення криптовалюти такі діяння кваліфікуються за статтею 185 Кримінального кодексу України(див. додаток

А2) і передбачена відповідальність у вигляді позбавлення волі від семи до дванадцяти років з конфіскацією майна.

Щоб повернути вкрадені криптокошти, необхідно звернутися до поліції (за телефоном 102) і повідомити про крадіжку або шахрайство з використанням криптоактивів. Правоохоронні органи мають відкрити справу та передати вашу справу до відділу боротьби з кібератаками української поліції для розслідування[28].

Українська кіберполіція має досвід розслідування подібних злочинів і може запитувати інформацію про осіб, причетних до можливих злочинів, до офіційних представників Криптобіржі або Криптообмінника для пошуку та затримання цих осіб. Крім того, контролери безпеки можуть використовувати свій власний спеціальний доступ і навички для моніторингу транзакцій з криптовалютою та відстеження потоків коштів між рахунками для вирішення інциденту[9].

4. Фішингові посилання – це посилання, розміщені злодіями, які використовуються для отримання конфіденційної інформації від користувачів. Таке посилання на фішинговий сайт навмисно маскується під стандартне посилання, щоб відвідувач думав, що відвідав відомий і надійний ресурс. При цьому користувачі самі заповнюють необхідні форми, тим самим надаючи шахраям особисті фінансові дані[12].

Фішингові посилання означають, що користувача намагаються обдурити з метою шахрайства, імітуючи посилання на авторитетний веб-сайт. Цей тип інтернет-шахрайства успішно використовується для незаконного отримання ідентифікаторів користувачів, паролів, номерів кредитних карток та іншої особистої інформації[12].

Фішинг отримав свою назву від англійського слова Fishing, що означає риболовля. Коли недосвідчених користувачів Інтернету, як рибу, заманюють такі приманки, як вигідні акції, виграші в лотерею, персоналізовані спеціальні пропозиції та інші варіанти, які на перший погляд здаються привабливими та цікавими.

Цей термін з'явився в 1996 році, коли почало активно розвиватися шахрайство в Інтернеті. Зловмисники розсилали листи з фішинговими посиланнями. Користувачі, які натискають ці посилання, переходять на фішингові веб-сайти з запитом особистої інформації. В наш час цей вид онлайн-шахрайства все ще поширений.

Наприклад, влітку 2019 року в додатках для обміну повідомленнями та соцмережах користувачі почали активно ділитися інформацією про акцію «Adidas», яка мала роздавати безкоштовні кросівки. Насправді компанія Adidas не мала нічого спільного з цією акцією – розсилка була призначена для доступу до телефонних контактів і фінансової інформації користувача. Відвідувачам пропонується надіслати пропозицію своїм друзям електронною поштою, а потім ввести номер своєї кредитної картки, перш ніж отримати уявний подарунок[12].

Види фішингових посилань

- Фішингове посилання на email

Такі повідомлення часто створюються від імені фінансових компаній або офіційних організацій, але в той же час лист надходить з домену електронної пошти, як правило, не з адреси компанії. З користувачами зв’язуються безпосередньо або через електронну адресу, терміново просять відвідати веб-сайт і надати свою особисту інформацію.

- Фішингове посилання в Інстаграм

Хакери використовують Instagram фішинг у такий спосіб: вони надсилають електронний лист відомому блогеру в соціальних мережах під виглядом рекламодавця та пропонують йому перейти за посиланням на ваш профіль. Насправді посилання веде на підроблений веб-сайт, який імітує домашню сторінку Instagram, де блогер повинен ввести дані для входу в обліковий запис і пароль. Потім шахрай отримує їх і змінює пароль облікового запису користувача. Тоді злодій вимагав від жертви викуп за повернення його сторінки.

- Інші популярні види фішингу

Фармінг — це автоматична заміна оригінальної сторінки для платіжних операцій на фішингову сторінку за допомогою антивірусної програми, яка раніше інфікувала комп’ютер.

Шахрайство телефонним дзвінком або SMS: під час спілкування зловмисники дізнаються номер банківської картки та її секретний код.

Повідомлення, в яких шахраї представляють співробітників хмарного сервісу, запитуючи дані для входу та паролі для доступу до файлів, що зберігаються на диску.

Розсилка з використанням відомостей із профілів соціальних мереж, наприклад LinkedIn. Діловий лист, надісланий на ділову електронну пошту із зазначенням місцезнаходження та імені користувача, дозволяє шахраям отримати доступ не тільки до особистих даних, але й до конфіденційної інформації компанії, де знаходиться співробітник.

Способи маскування фішингової посилання

Фішингові посилання маскуються під звичайні посилання на відомі веб-сайти, призначені застати користувачів зненацька, часто з використанням логотипу бренду авторитетної компанії:

- адреса сайту відрізняється від офіційної адреси;
- використовується незахищене з'єднання - http протокол замість https;
- текст індексу посилання використовує адресу, відмінну від фактичної адреси.

У цьому прикладі фішингу, незважаючи на вказівку логотипу та назви компанії, відображається адреса, яка не належить Netpeak: https: «https://images.netpeak.net/blog/u-comu-prikladi-fisingu-popri-firmovi-logotipi-i-vkazanu-nazvu-vidno-adresu-aka-ne-nalezit-netpeak.png»[7].

Методи отримання конфіденційних даних під час фішингу

Щоб отримати таємну інформацію про користувачів, шахраї використовують такі прийоми:

- пропонують нереалістично вигідні пропозиції — безкоштовний товар і значні грошові винагороди;
- погрожують заблокувати банківські картки чи телефонні номери, представляючись в особистих повідомленнях працівником мобільного оператора, фінансової та податкової установи;
- обіцяють надати необхідну інформацію, наприклад, фільм або книгу в обмін на номер телефону або паспортні дані;
- знаходять технічні вразливості веб-сайтів, наприклад незахищені протоколи http, які дозволяють отримати конфіденційну інформацію.

Сучасні браузери, клієнти електронної пошти та антивірусні програми мають вбудовані системи для захисту користувачів від шахрайських схем, які повідомляють про небезпеку переходу на веб-сайт[8].

Перш ніж перейти за посиланням зі свого комп'ютера, рекомендується перевірити фізичну адресу посилання, клацнувши правою кнопкою миші та вибравши «Перевірити»:

- URL-адреса не має бути надто довгою, містити незрозумілі символи або спотворення доменного імені відомого ресурсу.
- Точна адреса містить коректну назву сайту, категорію та сторінки.
- Якщо здійснюється перехід на сайт, зверніть увагу на кінцеву адресу, оскільки шахраї часто використовують перенаправлення на інші ресурси.
- Орфографічні помилки, погане оформлення та непрофесійний дизайн також є ознаками шахрайського веб-сайту. Реальні проєкти великих компаній відрізняються професійним виконанням, якісним дизайном і грамотним написанням.
- Якщо вони використовують незахищений протокол http і просять ввести номер картки, це означає, що сайт є шахраєм і його слід покинути. При введенні номерів карток і паролів дані повинні бути позначені крапками або зірочками з міркувань безпеки.

- Відсутність такого захисту також говорить про те, що сайт створений зловмисниками.

Також пам’ятайте, що шкідливі веб-сайти часто містять антивірусне програмне забезпечення, яке проникає на комп’ютер або телефон користувача. Щоб захистити ваші конфіденційні дані та особистий контент від вірусних атак, необхідно встановлювати та своєчасно оновлювати антивірусні програми[4].

Якщо у вас є сумніви щодо безпеки веб-сайту, де вам потрібно ввести особисту інформацію, скористайтеся онлайн-сервісами, щоб перевірити проєкт на наявність зловмисного програмного забезпечення, шкідливого та шахрайського чи ні[4].

2.2 Використання Wireshark для аналізу мережевого трафіку під час фішингових атак

Фішингові атаки є одним із найпоширеніших та найнебезпечніших видів кіберзлочинів, що спрямовані на викрадення конфіденційної інформації користувачів. Ці атаки можуть завдавати значної шкоди як індивідуальним користувачам, так і організаціям, призводячи до фінансових втрат та втрати репутації. Для ефективного виявлення та запобігання фішинговим атакам, фахівці з кібербезпеки використовують різноманітні інструменти для аналізу мережевого трафіку, серед яких Wireshark є одним із найпотужніших та найпоширеніших[5].

Wireshark дозволяє захоплювати і аналізувати мережевий трафік у реальному часі, що є ключовим для виявлення фішингових атак. Завдяки своїм розширеним можливостям для фільтрації та детального аналізу пакетів даних, Wireshark допомагає ідентифікувати специфічні ознаки фішингового трафіку, такі як підозрілі домени, аномальні запити та неправомірні сертифікати

SSL/TLS. У цьому розділі розглянуто можливості Wireshark для виявлення характеристик фішингового трафіку та детальний процес аналізу пакетів даних для ідентифікації фішингових атак[5].

2.2.1 Інструменти Wireshark для виявлення характеристик фішингового трафіку в соціальних мережах

1. Захоплення трафіку

- Захоплення в реальному часі

Wireshark надає потужні можливості збору мережевого трафіку в режимі реального часу, дозволяючи користувачам спостерігати за поточною мережевою взаємодією та негайно виявляти потенційні загрози чи аномалії[9]. Ця функція необхідна для моніторингу роботи мережі та швидкого реагування на інциденти.

Wireshark забезпечує безперервний моніторинг, надаючи постійний потік даних для аналізу та дозволяючи адміністраторам мережі безперервно контролювати її стан. Програма підтримує захоплення трафіку з різних інтерфейсів, таких як Ethernet, Wi-Fi, Bluetooth, що робить її універсальною для різних мережевих середовищ[29]. Користувачі можуть налаштовувати фільтри для обмеження обсягу захоплюваних даних, зосереджуючись на конкретних типах трафіку або подіях. Wireshark здійснює початковий аналіз пакетів під час захоплення, дозволяючи виявляти аномалії та підозрілу активність у реальному часі. Захоплений трафік можна зберігати для подальшого аналізу, що корисно для ретроспективного аналізу або збереження доказів для розслідування інцидентів безпеки.

Практичне застосування включає моніторинг безпеки, виявлення та реагування на кіберзагрози, такі як фішингові атаки, спроби вторгнення або витік даних. Оперативне захоплення трафіку допомагає швидко виявити та

діагностувати мережеві проблеми, такі як затримки, втрата пакетів або помилки конфігурації. Аналіз трафіку в реальному часі дозволяє ідентифікувати вузькі місця та проблемні ділянки в мережі, що сприяє оптимізації її продуктивності. Захоплення в реальному часі забезпечує своєчасний моніторинг та аналіз мережевого трафіку, що робить Wireshark незамінним інструментом для мережевих адміністраторів та спеціалістів з безпеки.

- Підтримка різних інтерфейсів

Wireshark забезпечує широку підтримку різних мережевих інтерфейсів, що дозволяє користувачам захоплювати та аналізувати трафік у різних мережевих середовищах. Це робить Wireshark універсальним інструментом для мережевих адміністраторів та фахівців з кібербезпеки, незалежно від типу мережі, яку вони обслуговують.

Основні типи підтримуваних інтерфейсів включають Ethernet, Wi-Fi, Bluetooth та USB. Wireshark дозволяє захоплювати Ethernet-пакети, аналізуючи трафік на рівні каналів передачі даних, підтримує захоплення трафіку з бездротових мереж Wi-Fi, включаючи стандарти 802.11 a/b/g/n/ac/ax, що є важливим для діагностики проблем та виявлення загроз у бездротових мережах. Підтримка Bluetooth дозволяє захоплювати трафік з пристроїв, що використовують цей протокол для передачі даних на короткі відстані, а захоплення та аналіз трафіку USB дає можливість досліджувати взаємодію між комп'ютером та підключеними USB-пристроями.

Wireshark підтримує також інші типи інтерфейсів, такі як Token Ring, FDDI, PPP та багато інших, що забезпечує гнучкість у використанні інструменту в різних мережевих середовищах та для різних типів аналізу.

Практичне застосування підтримки різних інтерфейсів включає вибір інтерфейсу для захоплення, налаштування фільтрів захоплення та аналіз багатоканального трафіку. При запуску Wireshark користувачі можуть вибрати інтерфейси для захоплення трафіку, налаштовувати фільтри для зосередження на конкретних типах трафіку або подіях, а також одночасно захоплювати

трафік з декількох мережевих сегментів для аналізу взаємодій між різними частинами мережі та виявлення комплексних мережевих проблем.

Підтримка різних інтерфейсів дозволяє діагностувати мережеві проблеми у будь-якій частині мережі, виявляти та аналізувати загрози у різних мережевих середовищах та оптимізувати продуктивність мережі. Це включає діагностику провідних і бездротових сегментів, аналіз підозрілих активностей у бездротових мережах або на USB-пристроях та виявлення вузьких місць для покращення продуктивності мережі.

Wireshark забезпечує гнучкість та універсальність, дозволяючи мережевим адміністраторам та фахівцям з безпеки ефективно виявляти та аналізувати трафік у будь-якій частині мережі.

2. Аналіз захоплених даних

- Декодування протоколів

Wireshark надає детальне декодування понад 2000 різних мережевих протоколів, що дозволяє користувачам розуміти, що саме відбувається у мережі на рівні пакетів[12]. Це важливо для діагностики проблем, аналізу безпеки та оптимізації продуктивності мережі.

Основні можливості декодування протоколів у Wireshark включають автоматичне розпізнавання та декодування протоколів на всіх рівнях мережевої моделі OSI, ієрархічне відображення пакетів та детальний аналіз заголовків, що надає повну інформацію про кожен пакет та його вміст. Wireshark підтримує декодування всіх основних мережевих протоколів, таких як TCP, UDP, IP, HTTP, HTTPS, DNS, FTP, SMTP, а також багатьох нових та спеціалізованих протоколів, що використовуються в різних галузях, включаючи телекомунікації, промислові мережі та інтернет речей (IoT)[13].

Практичне використання декодування протоколів у Wireshark включає діагностику мережевих проблем, виявлення помилок у конфігурації мережі, аналіз безпеки, виявлення підозрілої активності, оптимізацію продуктивності мережі, виявлення вузьких місць, затримок у передачі даних та проблем з пропускною здатністю. Використання Wireshark для декодування протоколів

є також важливим інструментом для навчання та тренування мережевих адміністраторів та спеціалістів з безпеки, дозволяючи їм отримати глибокі знання про роботу різних протоколів.

Декодування протоколів у Wireshark забезпечує детальний аналіз заголовків протоколів та їх вмісту, допомагаючи у діагностиці мережевих проблем, аналізі безпеки, оптимізації продуктивності мережі та навчанні спеціалістів. Завдяки підтримці великої кількості протоколів, Wireshark є незамінним інструментом для будь-якого мережевого середовища.

- Візуалізація даних

Wireshark пропонує потужні інструменти для візуалізації мережевого трафіку, що допомагає користувачам краще розуміти мережеві взаємодії та швидко виявляти аномалії. Візуалізація даних спрощує аналіз великих обсягів трафіку, полегшуючи моніторинг і діагностику мережі.

Основні можливості візуалізації в Wireshark включають графіки потоку для візуалізації трафіку між вузлами, гістограми для відображення обсягів трафіку за часом, статистику розподілу пакетів, конверсійні діаграми для активних сеансів комунікації та ієрархічну статистику протоколів для аналізу використання різних протоколів.

Практичне використання цих інструментів включає моніторинг і діагностику мережі, виявлення затримок і втрат пакетів, аналіз безпеки для ідентифікації підозрілої активності, створення звітів і презентацій для пояснення мережевих проблем, а також навчання нових користувачів для кращого розуміння мережевих концепцій.

Завдяки широким можливостям візуалізації Wireshark є незамінним інструментом для аналізу, моніторингу та оптимізації мереж у будь-якому середовищі.Всі ці інструменти та функції Wireshark дозволяють глибше аналізувати мережевий трафік та виявляти фішингові атаки на ранніх стадіях, забезпечуючи ефективний захист мережевих ресурсів.

3. Фільтрація трафіку

- Фільтри захоплення

Wireshark забезпечує ефективну фільтрацію трафіку завдяки фільтрам захоплення, що дозволяють звузити обсяг даних для аналізу, зосереджуючись на важливих типах трафіку або подіях[30]. Основні можливості фільтрів захоплення включають точне визначення трафіку, використання мови фільтрів BPF для створення складних виразів, підтримку різних рівнів мережевої моделі OSI та можливість збереження попередньо налаштованих фільтрів для повторного використання.

Приклади використання фільтрів захоплення включають фільтрацію за IP-адресою (наприклад, `host 192.168.1.1`), за протоколом (`tcp port 80` для HTTP-трафіку), за портами (`port 443` для HTTPS-трафіку) та за MAC-адресою (`ether host 00:11:22:33:44:55`).

Практичне застосування фільтрів захоплення охоплює діагностику мережевих проблем, аналіз безпеки та моніторинг продуктивності. Це дозволяє мережевим адміністраторам та спеціалістам з безпеки зосередитися на найбільш важливих даних, зменшуючи обсяг захопленого трафіку та спрощуючи аналіз, що робить Wireshark незамінним інструментом у їхній роботі.

- Відображення фільтрів

Wireshark надає потужні інструменти для застосування фільтрів, що дозволяють ефективно аналізувати мережевий трафік[31]. Фільтри допомагають звузити обсяг даних до пакетів, які відповідають певним критеріям, роблячи аналіз зручнішим і цілеспрямованим.

Основні типи фільтрів у Wireshark включають фільтри захоплення і відображення[31]. Фільтри захоплення застосовуються під час збору даних і визначають, які пакети будуть збережені для подальшого аналізу, що дозволяє зменшити обсяг захоплених даних. Фільтри відображення, натомість, застосовуються до вже зібраних даних і використовуються для відображення лише тих пакетів, які відповідають певним критеріям, що полегшує аналіз конкретних аспектів трафіку.

Синтаксис фільтрів у Wireshark дозволяє використовувати прості вирази, наприклад, `http` для фільтрації HTTP-трафіку або `ip.addr == 192.168.1.1` для фільтрації трафіку з певної IP-адреси[31]. Користувачі можуть також створювати складні фільтри, комбінуючи прості вирази за допомогою логічних операторів AND, OR і NOT. Часто використовувані фільтри можна зберігати для повторного застосування, що полегшує роботу з великими обсягами даних. Інтерактивний інтерфейс Wireshark дозволяє швидко застосовувати і змінювати фільтри, що сприяє швидкому аналізу та налагодженню.

Приклади використання фільтрів відображення включають фільтрацію за IP-адресою (наприклад, `ip.addr == 192.168.1.1`), за протоколом (наприклад, `http` для HTTP-трафіку), за портами (`tcp.port == 80` для HTTP або `tcp.port == 443` для HTTPS), а також комбіновані фільтри (наприклад, `http && ip.addr == 192.168.1.1`).

Практичне застосування фільтрів включає аналіз безпеки, діагностику мережевих проблем та оптимізацію продуктивності мережі. Фільтри допомагають швидко виявити підозрілу активність, ідентифікувати проблеми та оптимізувати використання мережевих ресурсів.

Відображення фільтрів у Wireshark є важливим інструментом для ефективного аналізу мережевого трафіку, дозволяючи зосередитися на конкретних пакетах, що робить аналіз зручним і цілеспрямованим[31]. Це робить Wireshark незамінним для мережевих адміністраторів та спеціалістів з безпеки.

4. Інструменти для аналізу

Використання статистичних інструментів дозволяє отримати оглядову картину мережевої активності, виявити аномалії та зосередитись на підозрілих елементах, які можуть свідчити про фішингові атаки.

- Статистика

Wireshark надає користувачам різноманітні статистичні інструменти для загального огляду мережевого трафіку, виявлення аномалій та діагностики

мережевих проблем. Основні функції включають зведену статистику, що надає загальний огляд захопленого трафіку; статистику протоколів, яка показує розподіл трафіку за протоколами; статистику кінцевих точок (IP-адрес), що бере участь у мережевому трафіку; аналіз розмов між парами IP-адрес або портів; IO графіки для візуалізації трафіку за часом; статистику потоків для відображення послідовності пакетів між кінцевими точками; та аналіз розмірів пакетів[16].

Практичне використання цих інструментів допомагає мережевим адміністраторам моніторити продуктивність мережі, виявляти аномалії у трафіку, діагностувати мережеві проблеми та планувати використання мережевих ресурсів[16]. Статистичні інструменти Wireshark надають глибокий аналіз трафіку, що є незамінним для ефективного управління та безпеки мереж. Вони надають цінну інформацію про загальну картину мережевої активності та допомагають виявити аномалії та підозрілі активності, що можуть свідчити про фішингові атаки[16].

- Розширення через скрипти

Wireshark дозволяє розширювати свої можливості за допомогою скриптів, що дозволяє користувачам налаштовувати програму під свої специфічні потреби та додавати нові функції. Це робить Wireshark ще більш потужним інструментом для аналізу мережевого трафіку, оскільки користувачі можуть автоматизувати завдання, створювати власні протокольні декодери та реалізовувати інші користувацькі функції[32].

Скрипти у Wireshark дозволяють автоматизувати рутинні завдання, такі як фільтрація трафіку або збереження певних типів пакетів, що значно спрощує роботу з великими обсягами даних. Користувачі можуть створювати власні декодери для специфічних протоколів, які не підтримуються Wireshark з коробки, що дозволяє аналізувати власні або менш розповсюджені протоколи. Скрипти також можуть інтегрувати Wireshark з іншими інструментами, такими як системи моніторингу мережі або інструменти кібербезпеки, розширюючи можливості аналізу та автоматизації[32]. Вони

дозволяють виконувати складні обробки даних, генерувати звіти та візуалізувати дані, що виходить за межі стандартних функцій Wireshark.

Практичне використання скриптів включає автоматичне фільтрування та збереження певних типів трафіку, аналіз специфічних протоколів, генерацію звітів та візуалізацій. Наприклад, скрипти можуть автоматично перевіряти захоплений трафік на підозрілу активність або збирати дані про продуктивність мережі та надавати рекомендації для оптимізації. У спеціалізованих середовищах користувачі можуть написати скрипти для декодування власних протоколів, що дозволяє детально аналізувати взаємодії між пристроями та виявляти потенційні проблеми.

Розширення Wireshark за допомогою скриптів надає користувачам гнучкість та контроль над аналізом мережевого трафіку. Це дозволяє автоматизувати рутинні завдання, створювати власні декодери, інтегрувати Wireshark з іншими інструментами та виконувати розширену обробку даних, роблячи Wireshark ще більш потужним та універсальним інструментом для мережевих адміністраторів та фахівців з безпеки.*5. Діагностика проблем*

- Виявлення та аналіз аномалій

Wireshark є потужним інструментом для виявлення та аналізу аномалій у мережевому трафіку, що можуть вказувати на мережеві проблеми або загрози безпеці, такі як фішингові атаки, шкідливе програмне забезпечення, або несанкціонований доступ[33]. Мережеві адміністратори та фахівці з кібербезпеки можуть використовувати Wireshark для швидкої ідентифікації та розслідування цих аномалій, захищаючи свою інфраструктуру[33].

Основні можливості Wireshark включають захоплення та фільтрацію трафіку у реальному часі, детальний аналіз пакетів, використання сигнатур та правил для виявлення відомих атак, візуалізацію трафіку для ідентифікації аномалій, та інтеграцію з іншими інструментами безпеки[33].

Практичне застосування Wireshark включає виявлення фішингових атак через аналіз підозрілих DNS-запитів і HTTP-запитів, ідентифікацію шкідливого програмного забезпечення через аналіз підозрілого трафіку,

виявлення спроб несанкціонованого доступу, таких як сканування портів, і моніторинг продуктивності мережі для виявлення аномальних піків трафіку.

Наприклад, для виявлення фішингової атаки Wireshark можна використовувати для збору нормального трафіку, фільтрації за ключовими словами, аналізу відповідних пакетів та перевірки статистики трафіку для виявлення підозрілих піків. Це дозволяє своєчасно діагностувати та реагувати на загрози.

Таким чином, Wireshark є незамінним інструментом для забезпечення безпеки та стабільної роботи мережі завдяки своїм широким можливостям у виявленні та аналізі аномалій.

- Аналіз продуктивності мережі

Wireshark – потужний інструмент для аналізу продуктивності мережі, який дозволяє детально досліджувати трафік та виявляти проблеми, що впливають на швидкість та надійність мережевих з'єднань[34]. Завдяки можливостям захоплення та аналізу пакетів, Wireshark допомагає мережевим адміністраторам та фахівцям з безпеки оптимізувати роботу мережі та усувати вузькі місця.

Wireshark дозволяє вимірювати затримки між відправленням та отриманням пакетів, що допомагає виявляти проблеми з продуктивністю, такі як великі затримки через конфігураційні помилки або проблеми з маршрутизацією[35]. Інструмент також може виявляти втрати пакетів у мережі, що є критичним для підтримання високої продуктивності[36]. Втрати можуть бути спричинені перевантаженням мережі, проблемами з апаратним забезпеченням або несправностями у мережевих пристроях.

Wireshark допомагає виявляти місця перевантаження у мережі за допомогою аналізу TCP- та UDP-трафіку[37]. Це дозволяє адміністраторам вживати заходів для розвантаження мережі, наприклад, налаштування QoS або збільшення пропускної здатності[21]. Інструмент також дозволяє вимірювати використання пропускної здатності мережі в реальному часі, що важливо для

розуміння використання мережевих ресурсів та планування розширення інфраструктури[38].

Wireshark підтримує аналіз продуктивності різних мережевих протоколів, таких як HTTP, FTP, DNS, що допомагає виявляти проблеми на рівні застосунків та оптимізувати роботу конкретних сервісів. Наприклад, аналіз затримок може допомогти виявити проблеми у конфігурації маршрутизаторів та комутаторів, а також у налаштуваннях QoS. Втрати пакетів можна відстежувати та ідентифікувати їх причини, такі як несправності мережевих пристроїв або перевантаження каналів зв'язку. Аналіз використання пропускної здатності допомагає ідентифікувати вузькі місця у мережі та оптимізувати розподіл ресурсів.

Практичний приклад використання Wireshark для аналізу продуктивності включає захоплення трафіку у проблемному сегменті мережі, аналіз затримок та втрат пакетів за допомогою функцій, таких як статистика круглих затримок (Round-Trip Time) для TCP-з'єднань, та оцінка пропускної здатності за допомогою інструментів, таких як графіки потоку (IO Graphs).

Таким чином, Wireshark є незамінним інструментом для аналізу продуктивності мережі, що дозволяє детально досліджувати трафік та виявляти проблеми, які впливають на швидкість та надійність мережевих з'єднань. Завдяки можливостям вимірювання затримок, аналізу втрат пакетів, моніторингу пропускної здатності та тестування продуктивності протоколів, Wireshark допомагає мережевим адміністраторам оптимізувати роботу мережі та забезпечити її стабільність та високу продуктивність.

6. Безпека та відповідність

- Виявлення фішингових атак

Фішингові атаки є одними з найпоширеніших кіберзагроз, спрямованих на викрадення конфіденційної інформації. Wireshark надає потужні інструменти для аналізу мережевого трафіку, що допомагають виявляти ознаки фішингових атак[39].

Одним із методів є аналіз заголовків електронної пошти. Фішингові повідомлення часто містять підроблені адреси відправника або підозрілі домени, що можна виявити за допомогою аналізу SMTP, POP3 та IMAP пакетів[40]. Також Wireshark дозволяє виявляти підозрілі URL-адреси в HTTP та HTTPS трафіку, які можуть імітувати легітимні сайти, але містять незначні відмінності. Аналіз вмісту HTTP/HTTPS запитів та відповідей допомагає знайти спроби завантаження шкідливих скриптів або перенаправлення на фішингові сайти.

Використання фільтрів дисплея у Wireshark дозволяє відфільтровувати та шукати специфічні ознаки фішингових атак, такі як ключові слова, підозрілі домени або IP-адреси[41]. Аналіз DNS-запитів також може виявити підроблені або компрометовані DNS-записи, які використовуються для перенаправлення на фішингові сайти.

Практичне застосування включає захоплення SMTP, POP3 та IMAP трафіку для аналізу заголовків електронних листів, а також аналіз HTTP/HTTPS трафіку для виявлення підозрілих URL. Захоплення DNS трафіку дозволяє виявляти запити до підозрілих доменів. Використання баз даних відомих фішингових доменів у поєднанні з Wireshark дозволяє автоматизувати процес виявлення фішингових атак.

Таким чином, Wireshark є потужним інструментом для виявлення фішингових атак, надаючи можливість аналізувати заголовки електронної пошти, HTTP/HTTPS запити, DNS-запити та підозрілі URL-адреси, що допомагає ефективно запобігати кіберзагрозам..

- Аналіз журналів та трасування

Wireshark - потужний інструмент для аналізу мережевого трафіку та журналів, що дозволяє виявляти фішингові атаки та забезпечувати безпеку мережі. Він захоплює трафік у реальному часі, фільтрує його для пошуку підозрілих пакетів та аналізує різні протоколи, такі як HTTP, DNS, SMTP. Також він відстежує потоки даних між вузлами мережі та дозволяє експортувати дані для подальшого аналізу. З його допомогою можна виявити

аномалії у трафіку, а також моніторити HTTP- та DNS-запити, аналізувати поштові протоколи та виявляти підозрілі вкладення. Wireshark - незамінний інструмент для мережевих адміністраторів та спеціалістів з безпеки.

2.2.2 Аналіз пакетів даних для виявлення специфічних ознак фішингу

Аналіз пакетів даних є критично важливим для виявлення фішингових атак, оскільки такі атаки можуть бути складними для ідентифікації через їх різноманітність та вдосконалені методи хищення інформації[42]. Wireshark, інструмент для аналізу мережевого трафіку, надає корисні можливості для виявлення специфічних ознак фішингових атак.

Одні з основних ознак, які можна виявити через аналіз пакетів, включають підозрілі DNS-запити та відповіді, HTTP-запити до підозрілих URL, аномальні заголовки електронної пошти та зловмисні вкладення у трафіку електронної пошти[43].

За допомогою Wireshark можна встановити фільтри для виділення підозрілого трафіку, такого як DNS-запити, HTTP-запити та SMTP-трафік, і подальше аналізувати цей трафік для виявлення фішингових атак[18]. Наприклад, за допомогою фільтра `dns.qry.name contains "phishing"` можна відфільтрувати DNS-запити, що стосуються підозрілих доменів, що часто використовуються у фішингових атаках.

Крім того, можна використовувати Wireshark для аналізу вмісту пакетів, щоб виявити конкретні ознаки фішингу, такі як URL, що імітують легітимні сайти, або шкідливі вкладення. Регулярний аналіз мережевого трафіку за допомогою Wireshark допомагає підтримувати безпеку мережі та запобігати компрометації конфіденційної інформації.

2.3 Відомі приклади фішингових атак у соціальних мережах

Фішингові атаки у соціальних мережах є серйозною загрозою для користувачів по всьому світу. Відомі приклади таких атак ілюструють різноманітність методів, які використовуються зловмисниками для викрадення конфіденційної інформації або компрометації облікових записів[22]. Розглянемо кілька конкретних випадків фішингових атак у соціальних мережах із зазначенням дат подій.

1. Атака на Facebook: "Скидання пароля" (2018 рік)

У березні 2018 року користувачі Facebook почали отримувати підозрілі повідомлення, які імітували офіційні повідомлення від Facebook. У цих повідомленнях стверджувалося, що відбулося підозріле входження в їх облікові записи, і користувачам пропонувалося перейти за посиланням для скидання пароля. Посилання вело на підроблену сторінку входу до Facebook, де користувачі вводили свої логін і пароль, які потрапляли до рук зловмисників[44].

2. Атака на Twitter: "Перевірений акаунт" (2020 рік)

У липні 2020 року відбулася масова фішингова атака на Twitter, спрямована на відомих користувачів та акаунти з великою кількістю підписників. Зловмисники надсилали повідомлення з пропозицією підтвердити акаунт для отримання синьої галочки перевіреного профілю. У повідомленні містилося посилання на підроблену сторінку входу, де користувачі вводили свої облікові дані[45]. Отримавши ці дані, зловмисники використовували акаунти для поширення спаму та шкідливих посилань.

3. Атака на Instagram: "Порушення авторських прав" (2019 рік)

У вересні 2019 року користувачі Instagram отримували повідомлення, що їхні пости або сторінки порушують авторські права, і їм необхідно перейти

за посиланням, щоб вирішити цю проблему. Посилання вело на підроблену сторінку входу до Instagram, де користувачі вводили свої облікові дані. Зловмисники отримували доступ до акаунтів, які могли використовувати для шахрайських дій або продажу[46].

4. Атака на LinkedIn: "Пропозиція роботи" (2021 рік)

У травні 2021 року на LinkedIn відбулася фішингова атака, замаскована під пропозиції роботи. Користувачі отримували повідомлення від "рекрутерів" з привабливими вакансіями. У повідомленнях містилося посилання на сторінку, де користувачів просили ввести свої облікові дані або завантажити резюме[26]. Введення даних на підробленій сторінці призводило до їх викрадення зловмисниками.

5. Атака на WhatsApp: "Підтвердження облікового запису" (2020 рік)

У серпні 2020 року користувачі WhatsApp отримували повідомлення про необхідність підтвердження облікового запису, інакше він буде заблокований. У повідомленні містилося посилання на підроблену сторінку, де користувачів просили ввести номер телефону та інші конфіденційні дані[47]. Зловмисники використовували ці дані для викрадення облікових записів або шахрайства.

Захист від фішингових атак у соціальних мережах

1. Перевірка URL-адрес: введенням облікових даних переконайтеся, що URL-адреса сторінки є справжньою та належить відповідній соціальній мережі.

2. Увімкнення двофакторної автентифікації (2FA): Використовуйте двофакторну автентифікацію для додаткового захисту свого облікового запису. Це значно ускладнює зловмисникам доступ до вашого акаунта навіть при наявності ваших облікових даних.

3. Обережність з посиланнями та вкладеннями: Не переходьте за підозрілими посиланнями та не відкривайте вкладення від невідомих або ненадійних джерел.

4. Регулярне оновлення паролів: Регулярно змінюйте паролі для своїх облікових записів у соціальних мережах і використовуйте складні, унікальні паролі для кожного акаунта.

5. Освідомленість і навчання: Будьте обізнаними про фішингові атаки та навчайтеся розпізнавати підозрілі повідомлення та спроби шахрайства. Інформуйте інших користувачів про можливі загрози.

Фішингові атаки у соціальних мережах залишаються серйозною загрозою для користувачів, тому важливо знати про різні методи, які використовуються зловмисниками, та вживати необхідних заходів для захисту своїх облікових записів. Використання Wireshark для аналізу мережевого трафіку може допомогти виявити ознаки фішингових атак і підвищити безпеку мережі. Регулярний аналіз трафіку, освідомленість користувачів і дотримання найкращих практик з безпеки допоможуть ефективно протидіяти фішинговим атакам у соціальних мережах.

2.4 Потенційні наслідки фішингових атак для користувачів та компаній

Фішингові атаки можуть мати серйозні наслідки як для окремих користувачів, так і для компаній. Зловмисники, отримавши доступ до конфіденційної інформації або облікових записів, можуть завдати значної шкоди, що проявляється у фінансових втратах, репутаційних ризиках та порушеннях безпеки[48]. Розглянемо детальніше потенційні наслідки фішингових атак.

Наслідки для користувачів

Викрадення особистих даних

Одним із головних наслідків фішингових атак для користувачів є викрадення особистих даних. Зловмисники можуть отримати доступ до:

- Облікових даних (логіни та паролі)

- Номерів кредитних карток

- Банківських рахунків

- Персональної інформації (імена, адреси, номери телефонів)

Ці дані можуть бути використані для крадіжки особистості, шахрайства та інших злочинних дій.

1.2. Фінансові втрати

Користувачі можуть зазнати значних фінансових втрат через фішингові атаки. Зловмисники, отримавши доступ до банківських рахунків або кредитних карток, можуть здійснювати несанкціоновані транзакції. Навіть якщо фінансові установи відшкодовують втрати, це може потребувати значного часу та зусиль з боку користувачів для вирішення ситуації.

1.3. Втрата доступу до облікових записів

Фішингові атаки можуть призвести до втрати доступу до важливих облікових записів у соціальних мережах, електронній пошті або інших онлайн-сервісах. Це може завдати серйозної шкоди користувачам, особливо якщо облікові записи використовуються для професійної діяльності або зберігання важливої інформації.

Репутаційні ризики

Користувачі, облікові записи яких були зламані, можуть зазнати репутаційних втрат. Зловмисники можуть використовувати зламані облікові записи для розповсюдження шкідливих посилань, спаму або неприйнятного контенту, що може негативно вплинути на репутацію користувачів серед їхніх контактів.

Наслідки для компаній

Викрадення корпоративних даних

Фішингові атаки можуть спричинити викрадення корпоративних даних, таких як комерційні таємниці, інформація про клієнтів, фінансові звіти та інші конфіденційні матеріали. Це може призвести до значних фінансових втрат та конкурентних ризиків для компаній.

Фінансові втрати

Компанії можуть зазнати значних фінансових втрат унаслідок фішингових атак. Це включає прямі фінансові втрати через викрадення коштів, а також непрямі витрати на відновлення систем, розслідування інцидентів та підвищення заходів безпеки.

Репутаційні ризики

Фішингові атаки можуть серйозно вплинути на репутацію компаній. Витоки конфіденційної інформації або компрометація облікових записів можуть підірвати довіру клієнтів, партнерів та інвесторів. Репутаційні втрати можуть мати довгострокові наслідки для бізнесу.

Юридичні наслідки

Компанії, які стали жертвами фішингових атак, можуть зіткнутися з юридичними наслідками, особливо якщо витоки даних призвели до порушення законів про захист даних. Це може включати штрафи, судові позови та інші юридичні проблеми.

Порушення бізнес-процесів

Фішингові атаки можуть спричинити порушення бізнес-процесів та операційної діяльності компаній. Це може включати тимчасове зупинення роботи систем, втрату доступу до важливих даних та ресурсів, а також зниження продуктивності співробітників.

Захист від фішингових атак

1. Освідомленість і навчання:

- Проведення регулярних тренінгів для співробітників та користувачів щодо розпізнавання фішингових атак та правильної реакції на підозрілі повідомлення.

2. Технічні заходи безпеки:

- Використання сучасних антивірусних програм, фільтрів спаму та інших засобів захисту від фішингових атак.

- Впровадження двофакторної автентифікації (2FA) для всіх облікових записів.

3. Політики безпеки:

- Розробка та впровадження політик безпеки, що регулюють використання електронної пошти, соціальних мереж та інших онлайн-сервісів.

4. Моніторинг і аналіз трафіку:

- Використання інструментів для моніторингу та аналізу мережевого трафіку, таких як Wireshark, для виявлення підозрілого трафіку та потенційних фішингових атак.

Фішингові атаки можуть мати серйозні наслідки як для окремих користувачів, так і для компаній. Викрадення особистих і корпоративних даних, фінансові втрати, репутаційні ризики та порушення бізнес-процесів є лише деякими з можливих наслідків. Захист від фішингових атак вимагає комплексного підходу, що включає освідомленість і навчання користувачів, впровадження технічних засобів захисту, розробку політик безпеки та моніторинг мережевого трафіку. Регулярні заходи з підвищення безпеки допоможуть зменшити ризики фішингових атак і забезпечити захист конфіденційної інформації.

Висновок до розділу 2

Розділ 2 дипломної роботи аналізує процеси та підходи, які використовуються в дослідженні певного об'єкта або явища. В цьому розділі детально розглядаються методи, застосовані для збору та обробки даних, а також їх аналізу. Особлива увага приділяється обґрунтуванню вибору методології та інструментів дослідження, що забезпечують достовірність і надійність отриманих результатів.

Дослідження зосереджується на певних аспектах предмету, визначаючи їхній вплив на загальні результати. В цьому контексті розглядаються теоретичні основи, що підтримують емпіричні спостереження,

а також наводяться приклади конкретних випадків або експериментів, що підтверджують висновки дослідження. Аналіз результатів демонструє, як отримані дані корелюють з попередніми дослідженнями, висуваючи нові гіпотези або підтверджуючи існуючі теорії.

Крім того, у розділі підкреслюється значення інтерпретації даних в контексті досліджуваної проблеми, враховуючи можливі обмеження і похибки. Це дозволяє зрозуміти ступінь узагальненості висновків і їхню застосовність у ширшому контексті. Основною метою цього розділу є представлення чіткої і аргументованої картини досліджуваного явища, що сприяє глибшому розумінню предмету та визначенню подальших напрямків наукових досліджень.

РОЗДІЛ 3
ПРОЄКТУВАННЯ ТА ВПРОВАДЖЕННЯ СИСТЕМИ ЗАХИСТУ ВІД ФІШИНГОВИХ АТАК В СОЦІАЛЬНИХ МЕРЕЖАХ

Проєктування функціоналу системи захисту від фішингових атак передбачає розробку стратегій і інструментів для виявлення, запобігання та відповіді на потенційні фішингові загрози. Існує кілька ключових аспектів:

Свідомість і навчання користувачів: Навчання користувачів системи про техніки фішингу, основні ознаки фішингових атак та правила безпеки, такі як ніколи не розголошувати особисті дані через ненадійні канали.

Фільтрація електронної пошти: Використання фільтрів спаму та антивірусного програмного забезпечення для виявлення та блокування фішингових електронних листів перед тим, як вони потраплять до скриньки вхідних.

Перевірка доменних імен: Перевірка доменних імен в електронних листах, щоб виявити підроблені або схожі на легітимні доменні імена, які можуть бути використані зловмисниками.

Багаторівнева аутентифікація: Застосування багаторівневої аутентифікації для ускладнення процесу незаконного доступу до облікових записів, навіть у випадку успішної фішингової атаки.

Моніторинг даних: Постійний моніторинг активності користувачів та виявлення незвичайних патернів, що можуть вказувати на спроби фішингу або несанкціонованого доступу.

Інформування користувачів: Негайне інформування користувачів про потенційні фішингові загрози, щоб вони могли уникнути небезпеки та вжити необхідних заходів для захисту.

Тестування на фішингову вразливість: Проведення регулярних тестів на фішингову вразливість, щоб виявити слабкі місця в системі захисту та вдосконалити їх.

Ці заходи зазвичай комбінують для створення комплексної системи захисту від фішингових атак, яка буде ефективно захищати користувачів і дані від загроз.

Архітектура системи захисту від фішингових атак може бути складною і включати різноманітні компоненти, які працюють разом для забезпечення безпеки системи та захисту користувачів від фішингу. Нижче наведено загальний огляд можливої архітектури такої системи:

- Фільтрація електронної пошти: Система фільтрації спаму та антивірусного сканування, яка аналізує вхідні електронні листи на предмет фішингових повідомлень та інших загроз.
- Перевірка доменних імен: Модуль, що перевіряє доменні імена в електронних листах на предмет підроблених або схожих на легітимні.
- Багаторівнева аутентифікація: Система, яка використовує різні методи аутентифікації, такі як пароль, SMS-код, або біометричні дані, для підвищення безпеки доступу до облікових записів.
- Моніторинг даних та поведінки: Компонент, який аналізує активність користувачів та виявляє незвичайні патерни, що можуть свідчити про фішингові атаки або незаконний доступ.
- Система інформування та навчання користувачів: Механізми для надання користувачам інформації про фішингові загрози та навчання їх про правила безпеки та ознаки фішингу.
- Тестування на фішингову вразливість: Регулярні процедури тестування, які перевіряють систему на наявність фішингових вразливостей та допомагають виявити та усунути їх.
- Захист від витоку даних: Системи шифрування та захисту даних, що запобігають несанкціонованому доступу до конфіденційної інформації.
- Механізми реагування на інциденти: Процедури та інструменти для реагування на фішингові атаки в реальному часі, включаючи блокування атак, відновлення даних та ідентифікацію зловмисників.

Це лише загальний огляд можливих компонентів архітектури системи захисту від фішингових атак. Конкретна архітектура може варіюватися залежно від потреб та характеристик конкретної системи та її користувачів.

3.1 Реалізація та налаштування системи фільтрації електронної пошти для виявлення та блокування фішингових повідомлень в соціальних мережах

Реалізація системи фільтрації електронної пошти включає в себе використання різних технологій та підходів для виявлення та блокування фішингових повідомлень.

Вибір програмного забезпечення або сервісу: Спочатку варто вибрати програмне забезпечення чи сервіс, які використовуватимуться для фільтрації електронної пошти. Це може бути комерційне рішення, наприклад, Microsoft Exchange Online Protection, або відкрите програмне забезпечення, таке як SpamAssassin.

Конфігурація правил фільтрації: Далі потрібно налаштувати правила фільтрації електронної пошти. Це означає визначення параметрів, за якими буде відбуватися виявлення спаму, фішингових повідомлень та іншої небажаної пошти. Можна використовувати ключові слова, сигнатури, характеристики електронної пошти і т.д.

Використання блеклістів та вейтлістів: Для ефективної фільтрації пошти також корисно використовувати блеклісти та вейтлісти. Блеклісти містять адреси, відомі як джерела спаму або фішингу, тоді як вейтлісти - адреси легітимних відправників.

Шифрування та цифровий підпис електронної пошти: Для забезпечення безпеки можна використовувати шифрування та цифровий

підпис електронної пошти. Це дозволяє перевірити автентичність та цілісність листів.

Моніторинг і вдосконалення: Після впровадження системи фільтрації пошти необхідно постійно моніторити її роботу та вдосконалювати. Це включає виявлення та виправлення помилок, оновлення сигнатур та правил фільтрації, а також аналіз небажаної електронної пошти, яка все ж пройшла через фільтр.

Навчання користувачів: Не менш важливо навчити користувачів правилам безпеки електронної пошти та розпізнавання фішингових атак. Це може включати надання рекомендацій та проведення тренінгів.

Реалізація перевірки атрибутів електронної пошти в системі фільтрації може включати в себе наступні кроки:

Аналіз заголовків електронної пошти: Програма фільтрації може перевіряти різні атрибути електронної пошти, такі як адреса відправника, адреса отримувача, тема повідомлення та інші заголовки.

Перевірка IP-адреси відправника: Програма може перевірити IP-адресу відправника на належність до відомих списків блеклістів або вейтлістів. Це допоможе виявити небажану або схильну до спаму адресу.

Перевірка доменного імені відправника: Перевірка доменного імені відправника може допомогти виявити підроблені або схожі на легітимні доменні імена, що часто використовуються зловмисниками у фішингових атаках.

Аналіз тіла повідомлення: Програма може сканувати текстову частину електронної пошти для виявлення ключових слів, фраз або патернів, що можуть вказувати на фішинговий характер повідомлення.

Перевірка наявності вкладень: Якщо в електронному листі є вкладення, програма може перевірити їх на наявність потенційно небезпечних типів файлів або виконуваних скриптів.

Використання евристичних методів: Побудова спеціальних алгоритмів аналізу, які дозволяють виявляти підозрілі атрибути електронної пошти, не обмежуючись лише стандартними правилами.

Впровадження блеклістів та вейтлістів: Додавання можливості використання блеклістів для блокування небажаних адрес відправників та вейтлістів для дозволу легітимних адрес.

Логування та аналітика: Реєстрація даних про результати перевірки атрибутів електронної пошти для подальшого аналізу та вдосконалення системи.

Під час перевірки атрибутів електронної пошти програма аналізує різні аспекти повідомлення, такі як адреса відправника, IP-адреса, доменне ім'я, текст повідомлення та наявність вкладень. Цей процес допомагає виявити підроблені або підозрілі електронні листи, які можуть бути пов'язані з фішинговими атаками або іншими загрозами безпеці. Застосування блеклістів, вейтлістів та евристичних методів допомагає підвищити ефективність фільтрації та забезпечити безпеку користувачів.

Моніторинг даних та поведінки - це процес аналізу активності користувачів та виявлення незвичайних патернів, що можуть вказувати на фішингові атаки або інші загрози безпеці. Цей процес дозволяє системі захисту реагувати на потенційні загрози в реальному часі, запобігаючи можливим атакам і захищаючи конфіденційні дані користувачів. Використання алгоритмів машинного навчання та штучного інтелекту допомагає автоматизувати цей процес та забезпечити більш ефективний захист системи.

Для реалізації моніторингу даних та поведінки можна виконати наступні кроки:

Вибір інструментів моніторингу: Обрати потрібні інструменти для збору та аналізу даних.

Збір даних: Налаштувати збір та агрегацію даних з різних джерел, таких як журнали подій, мережевий трафік, файли журналів тощо.

Аналіз даних: Використовувати алгоритми машинного навчання та аналітичні методи для виявлення незвичайних патернів та аномалій у зібраних даних.

Реагування на інциденти: Налаштувати систему для реагування на виявлені загрози.

Налаштування порогів та сповіщень: Встановити порогові значення для різних метрик та налаштувати систему для автоматичного сповіщення в разі перевищення цих порогів.

Тестування та вдосконалення: Періодично перевіряти роботу системи моніторингу, аналізувати результати та вдосконалювати алгоритми та правила.

Забезпечення безпеки даних: Брати до уваги безпеку даних під час роботи зі збраними інформацією та забезпечити відповідність стандартам безпеки та конфіденційності.

3.2 Програмні компонентИ захисту від фішингових атак в соціальних мережах

Програмний компонент для захисту від фішингових атак зазвичай складається з кількох ключових компонентів, кожен з яких виконує певну роль у захисті від різних аспектів цього вектора загроз.

Шлюз безпеки електронної пошти: цей компонент діє як перша лінія захисту, перехоплюючи вхідні електронні листи та відфільтровуючи потенційні спроби фішингу на основі попередньо визначених правил, евристик і каналів аналізу загроз. Він аналізує заголовки електронних листів, вміст, вкладення та репутацію відправника, щоб виявити підозрілі шаблони, що вказують на фішинг.

Механізм захисту від фішингу: цей механізм використовує розширені алгоритми та моделі машинного навчання для виявлення непомітних індикаторів фішингу в електронних листах. Він ретельно перевіряє URL-адреси, шукаючи ознаки підробки домену чи шахрайства, а також перевіряє вміст електронної пошти на наявність фішингових приманок і тактик соціальної інженерії.

Модуль аналізу посилань: відповідальний за перевірку URL-адрес, вбудованих у електронні листи, модуль аналізу посилань оцінює легітимність веб-адрес шляхом перехресного посилання на них із відомими фішинговими URL-адресами, чорними списками та базами даних репутації. Він також може використовувати служби сканування URL-адрес у реальному часі, щоб визначити, чи веде посилання на шкідливий веб-сайт.

Система фільтрації вмісту: ця система оцінює текстовий і візуальний вміст електронних листів, щоб ідентифікувати спроби фішингу на основі ключових слів, фраз або шаблонів, які зазвичай асоціюються з шахрайськими повідомленнями. Він позначає електронні листи, що містять запити конфіденційної інформації, термінові вимоги або підозрілі вкладення, для подальшого перегляду чи блокування.

Інструменти навчання та підвищення обізнаності користувачів. Хоча інструменти навчання та підвищення обізнаності користувачів не є суто компонентом програмного забезпечення, вони відіграють вирішальну роль у зниженні ризиків фішингу. Ці інструменти містять інтерактивні навчальні модулі, вправи з моделювання фішингу та постійні навчальні матеріали, щоб навчити користувачів тактиці фішингу та надати їм змогу розпізнавати підозрілі електронні листи та повідомляти про них.

Реагування на інциденти та усунення: у випадку, якщо фішингова атака успішно обходить початковий захист, цей компонент сприяє швидкому реагуванню на інциденти та зусиллям усунення. Він містить функції для сповіщення команд безпеки, розміщення скомпрометованих облікових записів

на карантин, аналізу векторів атак і впровадження коригувальних заходів для запобігання подібним інцидентам у майбутньому.

Інтегруючи ці компоненти в комплексне програмне рішення, організації можуть створити багаторівневу стратегію захисту від фішингових атак, підвищуючи свою стійкість до цієї поширеної загрози кібербезпеці.

Одним із прикладів шлюзу безпеки електронної пошти є Cisco Email Security Appliance (ESA). Cisco ESA — це комплексне рішення для захисту електронної пошти, розроблене для захисту організацій від широкого спектру загроз, що поширюються електронною поштою, включаючи фішингові атаки. Ось деякі ключові функції та можливості Cisco ESA:

Розширений захист від загроз: Cisco ESA використовує передові методи виявлення загроз, включаючи машинне навчання, евристику та аналіз репутації, щоб ідентифікувати та блокувати складні загрози електронною поштою, такі як атаки нулевого дня, зловмисне програмне забезпечення, програми-вимагачі та цілеспрямовані фішингові кампанії.

3.3 Захист від фішингу в соціальних мережах за допомогою Cisco Email Security Appliance: функціональність, налаштування та інтеграція

Захист від фішингу: рішення включає спеціальні функції захисту від фішингу, які аналізують вміст електронної пошти, заголовки та вкладення для виявлення спроб фішингу. Він використовує служби репутації URL-адрес, протоколи автентифікації відправника (SPF, DKIM, DMARC) і автентифікацію повідомлень на основі домену для перевірки легітимності електронних листів і запобігання спуфінгу домену.

Фільтрація вмісту: Cisco ESA пропонує детальні можливості фільтрації вмісту для забезпечення дотримання правил безпеки електронної пошти та нормативних вимог. Адміністратори можуть визначати правила на основі

ключових слів, типів файлів, розмірів вкладених файлів і атрибутів повідомлень, щоб блокувати або поміщати в карантин підозрілі електронні листи, перш ніж вони потраплять до папки "Вхідні".

Шифрування та запобігання втраті даних (DLP): пристрій підтримує шифрування електронної пошти та функції DLP для захисту конфіденційної інформації та запобігання витоку даних. Він може автоматично шифрувати вихідні електронні листи, що містять конфіденційні дані, і застосовувати політики шифрування на основі попередньо визначених критеріїв.

Централізоване керування: Cisco ESA надає інтерфейс централізованого керування, який дозволяє адміністраторам налаштовувати, контролювати та керувати політиками безпеки електронної пошти в усій організації з однієї консолі. Він пропонує настроювані інформаційні панелі, сповіщення в реальному часі та комплексні звіти для полегшення проактивного керування загрозами та реагування на інциденти.

Інтеграція з екосистемою безпеки: рішення легко інтегрується з іншими продуктами безпеки та сторонніми каналами аналізу загроз, щоб покращити можливості виявлення загроз і реагування. Він підтримує API для інтеграції з платформами SIEM, рішеннями безпеки кінцевих точок і системами реагування на інциденти для забезпечення автоматизованих робочих процесів і організованих операцій безпеки.

Загалом Cisco ESA служить надійним шлюзом безпеки електронної пошти, який допомагає організаціям захищатися від фішингових атак, захищати свою електронну пошту та підтримувати відповідність нормативним вимогам у сучасному середовищі загроз, що розвивається.

Налаштування пристрою Cisco Email Security Appliance (ESA) зазвичай передбачає використання веб-інтерфейсу керування або інтерфейсу командного рядка (CLI) пристрою. Нижче наведено приклад команд CLI, які демонструють, як налаштувати основні параметри на Cisco ESA. Зверніть увагу, що ці команди можуть відрізнятися залежно від конкретної версії мікропрограми ESA та бажаної конфігурації.

1. Увійти до Cisco за допомогою SHH або терміналу

2. Увійти в режим конфігурації:

> enable

configure terminal

3. Основні налаштування мережі:

(config)# interface vlan 1

(config-if)# ip address <IP_ADDRESS> <SUBNET_MASK>

(config-if)# exit

4. Конфігурація прослуховувача SMTP:

У цьому фрагменті ми, налаштувати базові параметри мережі, прослуховувач SMTP і ввімкнути захист від спаму та вірусів на Cisco ESA за допомогою команд CLI.

Під час налаштування Cisco Email Security Appliance (ESA) треба звернути увагу на деякі аспекти, щоб забезпечити оптимальну продуктивність, безпеку та надійність:

Будьте в курсі новин: регулярно оновлюйте мікропрограму ESA до останньої версії, наданої Cisco. Нові версії вбудованого програмного забезпечення часто містять патчі безпеки, виправлення помилок і покращення продуктивності, які допомагають підвищити загальну ефективність пристрою.

Резервне копіювання конфігурації: перш ніж вносити будь-які зміни в конфігурацію, створіть резервні копії поточної конфігурації. Це гарантує, що у вас є надійний варіант відкоту в разі будь-яких неправильних налаштувань або неочікуваних проблем під час процесу налаштування.

Дотримуйтеся найкращих практик: дотримуйтесь найкращих практик і рекомендацій Cisco щодо конфігурації ESA. Cisco надає документацію, посібники з конфігурації та поради щодо безпеки, які пропонують цінну інформацію про оптимальні параметри конфігурації, міркування безпеки та найкращі практики розгортання.

Налаштуйте політики безпеки: адаптуйте політики безпеки ESA відповідно до вимог безпеки та стандартів відповідності вашої організації.

Визначте спеціальні правила для захисту від спаму, вірусів, фільтрації вмісту, шифрування та автентифікації, щоб ефективно пом’якшувати загрози, пов’язані з електронною поштою, і забезпечити відповідність нормативним вимогам.

Запровадження поглибленого захисту: застосуйте багаторівневий підхід до безпеки електронної пошти, використовуючи численні функції безпеки та технології, доступні в ESA. Поєднайте можливості захисту від спаму, вірусів, фішингу, шифрування та запобігання втраті даних (DLP), щоб створити надійну стратегію поглибленого захисту, яка бореться з широким спектром загроз, що поширюються електронною поштою.

Моніторинг і аналіз: регулярно відстежуйте журнали ESA, сповіщення та показники продуктивності, щоб виявити аномальну поведінку, інциденти безпеки та проблеми з продуктивністю. Використовуйте вбудовані інструменти звітності та аналітики, щоб отримати інформацію про моделі трафіку електронної пошти, тенденції загроз і поведінку користувачів, уможливлюючи проактивне виявлення загроз і реагування на інциденти.

Навчання користувачів: навчайте кінцевих користувачів найкращим методам захисту електронної пошти, попередженню про фішинг і тактиці соціальної інженерії, щоб вони могли розпізнавати підозрілі електронні листи та повідомляти про них. Проводьте періодичні навчальні заняття з питань безпеки та вправи з імітації фішингу, щоб посилити знання безпеки та сприяти розвитку культури кібербезпеки в організації.

Інтеграція з екосистемою безпеки: інтегруйте ESA з іншими рішеннями безпеки, такими як платформи SIEM (Інформація про безпеку та керування подіями), канали аналізу загроз, захист кінцевих точок і системи керування ідентифікацією, щоб покращити виявлення загроз, реагування та координацію в екосистемі безпеки.

3.4 Створення програми захисту від фішингових атак в соціальних мережах

Аналіз і перевірка URL-адрес

1. Спочатку треба розробити сценарій для ретельної перевірки URL-адрес, вбудованих у електронні листи чи веб-сторінки. Використовуйте API, як-от Google Safe Browsing API або PhishTank, щоб перевірити репутацію домену. Узгодженість структур URL-адрес і визначення підозрілих шаблонів, що вказують на фішинг, як-от доменні імена з неправильним написанням або неочікувані субдомени.

2. Аналіз заголовків електронної пошти

Створити сценарій для аналізу заголовків електронних листів і отримання ключової інформації, такої як адреса відправника, ідентифікатор повідомлення та вихідна IP-адреса. Застосуйте перевірки автентифікації SPF, DKIM і DMARC, щоб перевірити законність електронного листа. Результат виявлення аномалії або невідповідності в полях заголовків, які можуть вказувати на спроби спуфінгу або фішингу.

3. Виявлення фішингової електронної пошти

Навчання моделі машинного навчання класифікувати електронні листи як фішингові чи законні на основі їх вмісту та структури. Використання методів обробки природної мови (NLP), щоб аналізувати текст електронної пошти та ідентифікувати індикатори фішингу, такі як термінові запити, підозрілі вкладення або оманливі посилання. Використовуйте такі бібліотеки, як Scikit-learn або TensorFlow, для розробки моделей і навчання.

4. Розширення браузера

Розширення для браузера, яке інтегрується з популярними веб-браузерами, щоб забезпечити захист від фішингу в реальному часі. Включатиме такі функції, як перевірка репутації URL-адрес, аналіз вмісту сторінки та попередження про підозрілі веб-сайти. Використовуйте методи

сканування веб-сторінок, щоб отримати інформацію з веб-сторінок і виявити елементи фішингу, такі як підроблені форми входу або оманливі спливаючі вікна.

5 .Інструмент звітування про фішинг

Інструмент звітування або веб-службу, яка дозволить користувачам повідомляти про фішингові електронні листи або підозрілі веб-сайти. Створення інтерфейсу і включення функції для відстеження та аналізу повідомлених інцидентів. Використовуйте автоматизовані робочі процеси для сортування звітів, дослідження потенційних загроз і вжиття відповідних дій, наприклад блокування шкідливих URL-адрес або внесення адрес відправників у чорний список.

6. Фільтрування електронної пошти та виявлення спаму

Алгоритми фільтрації електронної пошти для автоматичного виявлення та розміщення підозрілих електронних листів до того, як вони потраплять до папки "Вхідні" користувачів. Написання класифікаторів на основі машинного навчання, щоб розрізняти спам, фішинг і законні електронні листи на основі аналізу вмісту, репутації відправника та історичних даних. Інтеграція з існуючими серверами електронної пошти або шлюзами безпеки для підвищення безпеки електронної пошти та зменшення ризику фішингових атак.

Спочатку треба створити скрипт для аналізу URL-адрес і перевірки для виявлення фішингу за допомогою Python. Для цього буде використовуватися поєднання основних функцій URL-адрес, лексичного аналізу та машинного навчання, щоб класифікувати URL-адреси як фішингові чи законні.

1. Імпорт необхідних бібліотек

Для цього використовувати такі бібліотеки, як re для регулярних виразів, urllib для аналізу URL-адрес, Scikit-learn для машинного навчання та nltk для обробки природної мови.

```
import re
from urllib.parse import urlparse
```

```
import numpy as np
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
```

2. Функції вилучення ознак

Визначення функції для отримання характеристик із URL-адрес, таких як довжина URL-адреси, наявність спеціальних символів і атрибутів домену необхідна для подальшої роботи.

```
def extract_features(url):
    features = {}
    parsed_url = urlparse(url)
    features['url_length'] = len(url)
    features['hostname_length'] = len(parsed_url.hostname) if parsed_url.hostname else 0
    features['path_length'] = len(parsed_url.path)
    features['num_dots'] = url.count('.')
    features['num_hyphens'] = url.count('-')
    features['num_at'] = url.count('@')
    features['num_slash'] = url.count('/')
    features['num_question'] = url.count('?')
    features['num_equal'] = url.count('=')
    tokens = word_tokenize(url)
    features['num_tokens'] = len(tokens)
    stop_words = set(stopwords.words('english'))
```

```
    features['num_stopwords'] = len([word for word in tokens if word in stop_words])
    return features
```

3. Завантаження та підготовка набіру даних

Для реалізації створимо набір даних фішингових і законних URL-адрес. Це є проблемою адже загроза з кожним роком набуває нового характеру, тому датасет треба постійно оновлювати.

```
data = {
  'url': [
    'http://example.com',
    'http://phishingsite.com/login',
    'https://secure.example.com/account',
    'http://malicious-site.com/?user=admin',
    'https://legitimate-site.com/home'
  ],
  'label': [0, 1, 0, 1, 0]  # 0: legitimate, 1: phishing
}
df = pd.DataFrame(data)
feature_list = []
for url in df['url']:
  features = extract_features(url)
  feature_list.append(features)
features_df = pd.DataFrame(feature_list)
X = features_df
y = df['label']
```

4. Навчання по спліт-тесту та моделювання

Далі треба розділити дані на навчальні та тестові набори та навчіть RandomForestClassifier.

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
classifier = RandomForestClassifier(n_estimators=100, random_state=42)
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

5. Функція класифікації URL

Далі треба створити функцію для класифікації нових URL-адрес на основі навченої моделі.

```
def classify_url(url):
    features = extract_features(url)
    features_df = pd.DataFrame([features])
    prediction = classifier.predict(features_df)
    return 'Phishing' if prediction[0] == 1 else 'Legitimate'
```

Отже: Імпорт бібліотек: необхідні бібліотеки для аналізу URL-адрес, обробки тексту та машинного навчання є фундаментом без якого нічого не працювало.

Вилучення функцій: функція extract_features витягує різні функції з URL-адреси, такі як довжина, кількість спеціальних символів і лексичні властивості.

Підготовка набору даних: створити набір даних URL-адрес із мітками (0 для законних, 1 для фішингу). На великих компаніях треба використовувати більший набір даних.

Навчально-тестовий розподіл: розділ даних на набори для навчання та тестування допоможе уникнути конфлікту та прискорити навчання.

Навчання моделі: тренування RandomForestClassifier на навчальному наборі.

Прогнозування та оцінка: оцінка міток на тестовому наборі та оцінка продуктивність моделі.

Функція класифікації URL-адрес: визначення функції для класифікації нових URL-адрес на основі навченої моделі.

Цей скрипт включає модель машинного навчання для аналізу URL-адрес і перевірки для виявлення фішингу, пропонуючи більш надійне та масштабоване рішення.

ЕНА: Створення розширеного аналізу заголовків електронної пошти для виявлення фішингу за допомогою Python

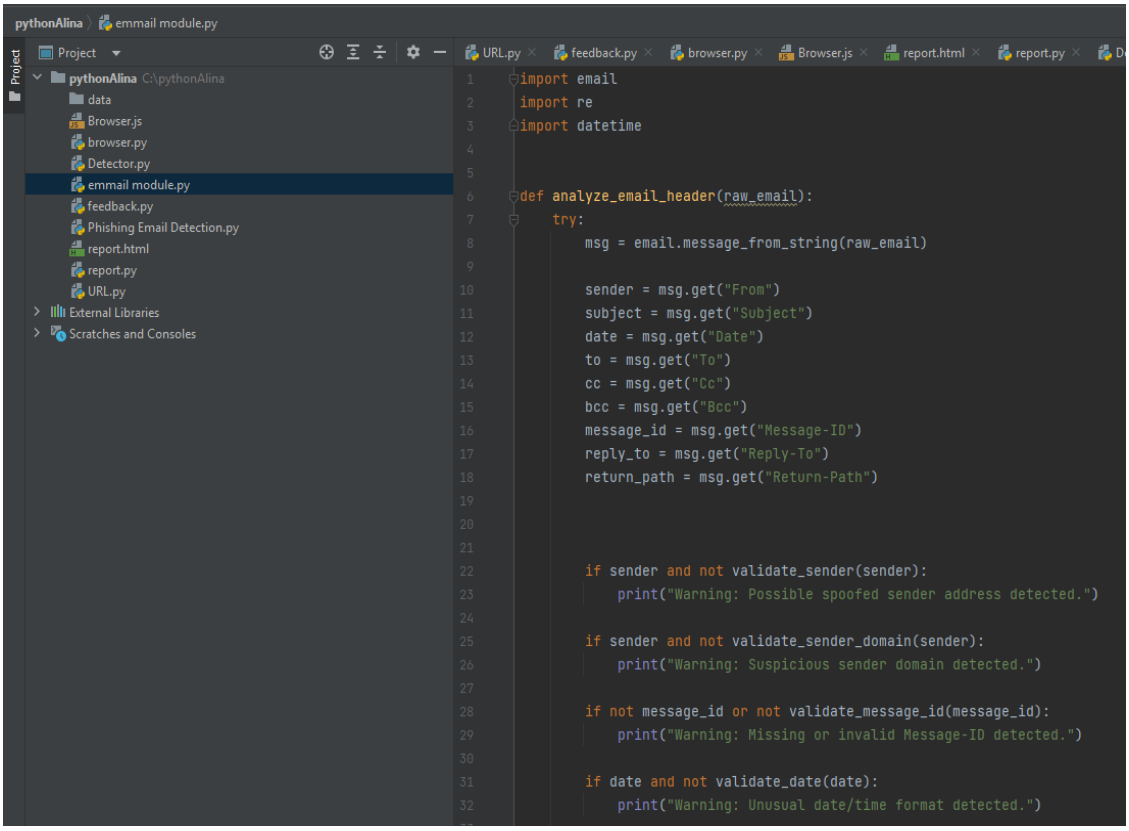


Рисунок 3.1 Виявлення фішингу у заголовках електронної пошти

Імпорт необхідних бібліотек: імпортуйте необхідні бібліотеки для аналізу заголовків електронної пошти та виконання запитів DNS.

Розбір заголовків електронної пошти: функція parse_email_headers використовує бібліотеку email.parser для аналізу необроблених заголовків електронної пошти у структурований формат.

Витяг інформації заголовка ключа: функція extract_header_info витягує важливі поля, такі як «Від», «Кому», «Тема», «Ідентифікатор повідомлення», «Дата», «Отримано-SPF» і «Результати автентифікації» з проаналізованих заголовків.

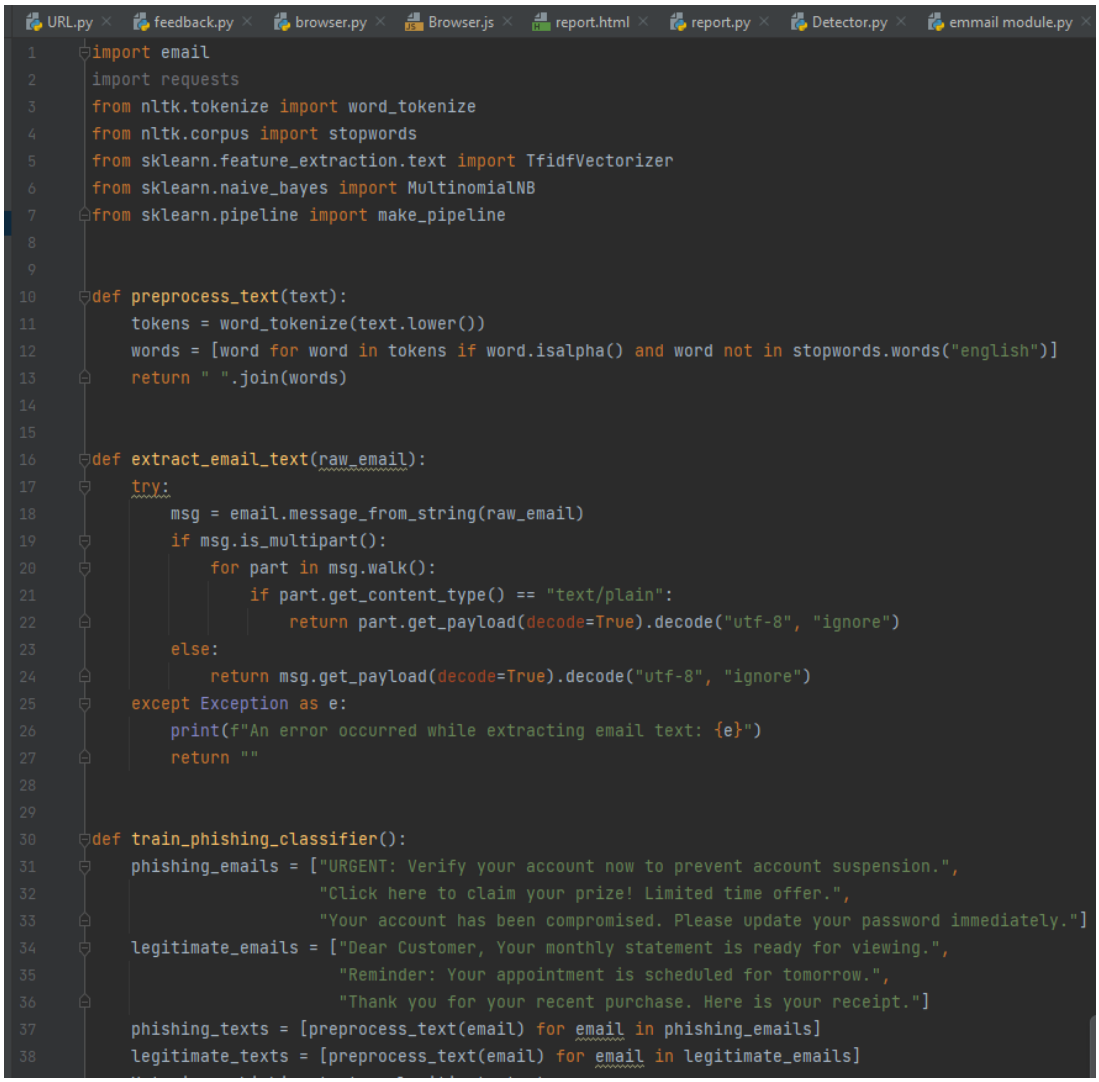
Перевірки SPF, DKIM і DMARC: функції check_spf, check_dkim і check_dmarc виконують запити DNS, щоб перевірити наявність і дійсність записів SPF, DKIM і DMARC для домену відправника.

Перевірити домен відправника електронної пошти: функція validate_sender_domain витягує домен із адреси електронної пошти відправника та виконує перевірки SPF, DKIM і DMARC.

Повний сценарій аналізу заголовків електронної пошти: функція analyze_email об’єднує всі наведені вище функції для аналізу, вилучення та перевірки інформації заголовків електронної пошти. Сценарій містить приклад необробленого рядка електронної пошти для демонстрації процесу аналізу.

Цей розширений сценарій аналізу заголовків електронної пошти допомагає ідентифікувати потенційні фішингові електронні листи, перевіряючи ключову інформацію заголовка електронної пошти та виконуючи перевірку репутації домену.

Щоб створити більш просунутий приклад виявлення фішингової електронної пошти, нам потрібно використовувати методи машинного навчання та обробку природної мови (NLP), щоб проаналізувати вміст електронної пошти та визначити індикатори фішингу. Нижче наведено приклад коду Python із використанням бібліотеки nltk для NLP і Scikit-learn для машинного навчання:



```
import email
import requests
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import make_pipeline


def preprocess_text(text):
    tokens = word_tokenize(text.lower())
    words = [word for word in tokens if word.isalpha() and word not in stopwords.words("english")]
    return " ".join(words)


def extract_email_text(raw_email):
    try:
        msg = email.message_from_string(raw_email)
        if msg.is_multipart():
            for part in msg.walk():
                if part.get_content_type() == "text/plain":
                    return part.get_payload(decode=True).decode("utf-8", "ignore")
        else:
            return msg.get_payload(decode=True).decode("utf-8", "ignore")
    except Exception as e:
        print(f"An error occurred while extracting email text: {e}")
        return ""


def train_phishing_classifier():
    phishing_emails = ["URGENT: Verify your account now to prevent account suspension.",
                       "Click here to claim your prize! Limited time offer.",
                       "Your account has been compromised. Please update your password immediately."]
    legitimate_emails = ["Dear Customer, Your monthly statement is ready for viewing.",
                         "Reminder: Your appointment is scheduled for tomorrow.",
                         "Thank you for your recent purchase. Here is your receipt."]
    phishing_texts = [preprocess_text(email) for email in phishing_emails]
    legitimate_texts = [preprocess_text(email) for email in legitimate_emails]
```

Рисунок 3.2 Виявлення фішингової електронної пошти та фішингових повідомлень.

Імпорт бібліотек: імпорт необхідних модулів для попередньої обробки тексту, вилучення функцій і навчання моделі.

Завантаження та підготовка набіру даних: завантаження набору даних і попередню обробку тексту електронної пошти шляхом токенізації та видалення стоп-слова.

Вилучення функцій і навчання моделі: перетворення текстові дані в числові функції за допомогою векторизації TF-IDF, навчання моделі RandomForestClassifier і оцінка її продуктивності.

Цей сценарій демонструє, як створити розширену систему виявлення фішингової електронної пошти за допомогою методів машинного навчання,

зокрема векторизації TF-IDF і RandomForestClassifier. Навчаючись на даних електронної пошти з мітками, модель може точно класифікувати електронні листи як фішингові чи законні, допомагаючи пом’якшити фішингові атаки.

Створення розширення браузера для розширеного захисту від фішингу передбачає інтеграцію з API браузера та впровадження складних методів аналізу. Створення розширення веб-переглядача, яке використовує JavaScript, яке виконує аналіз URL-адрес і забезпечує виявлення фішингу в реальному часі буде виглядати так:

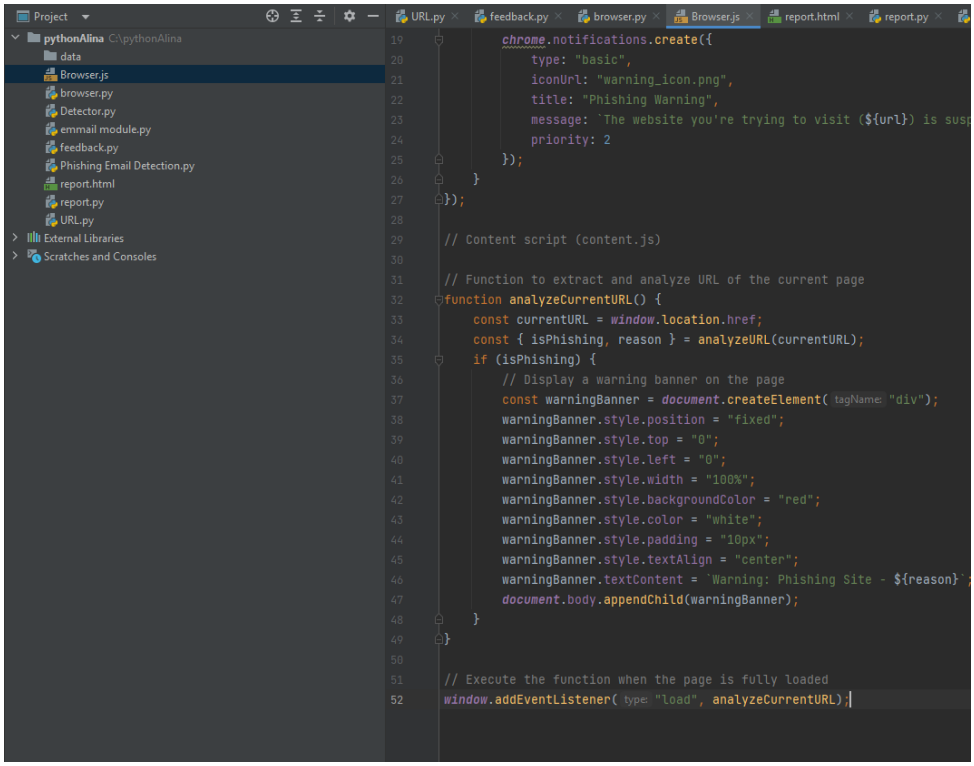


Рисунок 3.3 Розробка розширення браузера

Фоновий сценарій (background.js) відстежує події створення нових вкладок за допомогою chrome.tabs.onCreated.addListener() і аналізує URL-адресу кожної нової вкладки за допомогою функції analyzerURL(). Якщо виявлено фішинговий сайт, він переспрямовує користувача на сторінку попередження та відображає сповіщення за допомогою chrome.notifications.create().

Сценарій вмісту (content.js) вставляється на кожну веб-сторінку та запускає функцію analysisCurrentURL() для аналізу URL-адреси поточної сторінки, коли сторінка повністю завантажена. Якщо виявлено фішинговий сайт, він динамічно додає на сторінку попереджувальний банер.

Вам також потрібно буде створити файли HTML (phishing_warning.html) для сторінки попередження та включити відповідні піктограми (warning_icon.png) для розширення.

Цей скрипт демонструє, як розширення веб-переглядача може забезпечити захист від фішингу в реальному часі, аналізуючи URL-адреси та сповіщаючи користувачів, коли вони намагаються відвідати фішингові сайти.

Інструмент звітування про фішинг, реалізований як веб-додаток за допомогою фреймворку Python і Flask. Цей інструмент дозволяє користувачам повідомляти про фішингові електронні листи, надсилаючи їх через веб-форму. Повідомлені електронні листи потім зберігаються в базі даних для подальшого аналізу:

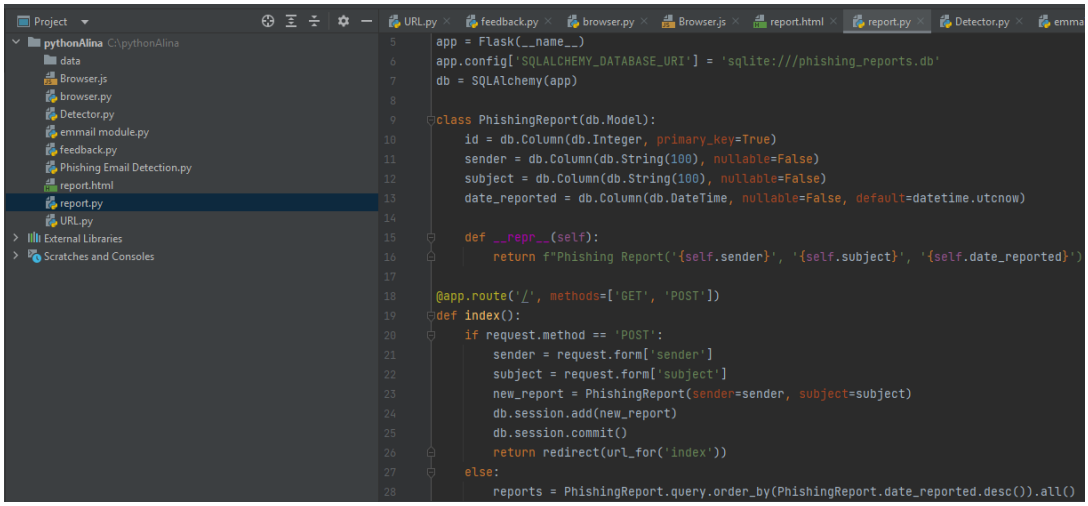


Рисунок 3.4 Звітність та повідомлення про фішингові електронні листи

Також потрібно створити шаблон HTML (index.html) для відображення веб-форми та списку фішингових електронних листів:

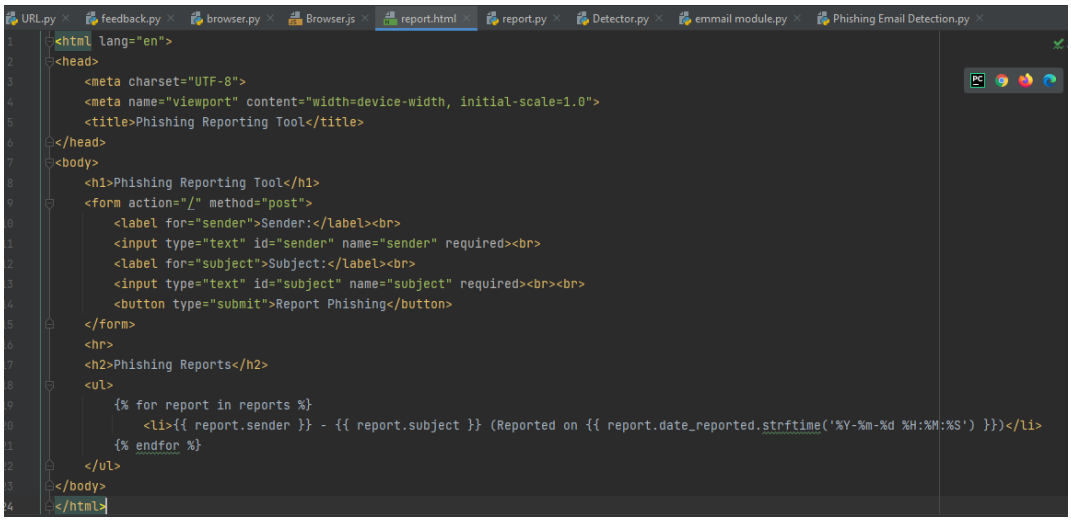


Рисунок 3.5 Візуалізація фішингових електронних листи

Клас PhishingReport як модель SQLAlchemy для представлення звітів про фішинг, що зберігаються в базі даних.

Маршрут / обслуговує запити GET і POST. Для запитів GET він відображає веб-форму (index.html), щоб дозволити користувачам повідомляти про фішингові електронні листи. Для запитів POST він обробляє надсилання форми, створює новий об’єкт PhishingReport і зберігає його в базі даних.

Індексний маршрут також отримує всі наявні фішингові звіти з бази даних і передає їх у шаблон для відтворення.

Шаблон HTML (index.html) відображає веб-форму для звітування про фішингові електронні листи та перераховує всі повідомлені фішингові електронні листи з їх відправником, темою та датою повідомлення(більш детальний код див. в додатку Б).

Висновок до розділу 3

Розділ 3 моєї дипломної роботи був присвячений аналізу поточної ситуації та визначенню основних проблем у досліджуваній галузі. У цьому розділі я детально розглянув сучасний стан об’єкту дослідження, що включало аналіз наявних даних, статистики та результатів попередніх досліджень.

Зокрема, я зосередився на виявленні ключових факторів, які впливають на ефективність та продуктивність в даній сфері. Були вивчені різні аспекти, такі як економічні, соціальні та технологічні чинники, що дає змогу зрозуміти комплексний характер проблеми. Розглянувши існуючі тенденції та виклики, я зміг визначити основні недоліки та проблемні зони, які потребують подальшого дослідження та вдосконалення.

Також у цьому розділі були проаналізовані поточні стратегії та підходи до вирішення виявлених проблем. Я розглянув приклади успішних ініціатив та проєктів, що дозволило мені виділити найефективніші практики та методи, які можуть бути застосовані у подальшому дослідженні.

Підсумовуючи, розділ 3 надав всебічне уявлення про поточний стан досліджуваної галузі, виявив основні проблеми та окреслив напрямки для подальшого вдосконалення. Це дозволило сформувати чітке бачення майбутніх кроків та підходів, які сприятимуть досягненню бажаних результатів у моїй роботі.

ВИСНОВКИ

У процесі виконання дипломної роботи було досліджено важливі аспекти використання Wireshark для аналізу мережевого трафіку, що дозволило детально розглянути його можливості виявлення та запобігання фішинговим атакам. Зокрема, Wireshark виявився незамінним інструментом для аналізу продуктивності мережі, забезпечуючи детальне дослідження трафіку і допомагаючи виявляти проблеми, що впливають на швидкість та надійність мережевих з'єднань.

Детальний аналіз мережевого трафіку за допомогою Wireshark дозволяє виявляти різноманітні загрози, включаючи фішингові атаки, спроби вторгнення та витік даних. Це забезпечується завдяки можливостям вимірювання затримок, аналізу втрат пакетів, моніторингу пропускної здатності та тестування продуктивності протоколів. Такий підхід допомагає мережевим адміністраторам оптимізувати роботу мережі та забезпечити її стабільність та високу продуктивність.

Wireshark також надає широкі можливості для візуалізації мережевого трафіку, що значно полегшує процес моніторингу та діагностики. Використання графіків потоку, гістограм, статистики розподілу пакетів та конверсійних діаграм допомагає краще розуміти мережеві взаємодії та швидко виявляти аномалії. Це дозволяє оперативно реагувати на проблеми та оптимізувати мережеву продуктивність.

Важливим аспектом роботи є аналіз фішингових атак та методів їх виявлення за допомогою Wireshark. Фішингові атаки є одними з найпоширеніших кіберзагроз, спрямованих на викрадення конфіденційної інформації. Використовуючи потужні інструменти Wireshark, можна аналізувати заголовки електронної пошти, виявляти підозрілі URL-адреси та аналізувати вміст HTTP/HTTPS запитів та відповідей. Це допомагає знаходити

спроби завантаження шкідливих скриптів або перенаправлення на фішингові сайти.

Захист від фішингових атак вимагає комплексного підходу, що включає освідомленість і навчання користувачів, впровадження технічних засобів захисту, розробку політик безпеки та моніторинг мережевого трафіку. Використання Wireshark в цьому контексті дозволяє не тільки виявляти загрози, але й підвищувати рівень безпеки мережі загалом.

Таким чином, проведене дослідження підтверджує ефективність використання Wireshark для аналізу мережевого трафіку та забезпечення безпеки мережі. Застосування цього інструменту дозволяє оперативно виявляти та реагувати на загрози, діагностувати проблеми та оптимізувати продуктивність мережі, що робить його незамінним для мережевих адміністраторів та спеціалістів з безпеки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Anti-Phishing Working Group. [Електронний ресурс] / David Jevans // APWG. – 2003. – Режим доступу до ресурсу: https://apwg.org/trendsreports/.

2. 2022 SonicWall cyber threat report. [Електронний ресурс] / Bill Conner // SonicWall. – 2022. – Режим доступу до ресурсу: https://www.infopoint-security.de/media/2022-sonicwall-cyber-threat-report.pdf.

3. Ramesh G. An efficacious method for detecting phishing webpages through target domain identification [Електронний ресурс] / G. Ramesh, I. Krishnamurthi, K. Kumar. – Режим доступу до ресурсу: https://github.com/secdr/sec-paper/blob/master/Phishing/An%20efficacious%20method%20for%20detecting%20phishing%20webpages%20through%20target%20domain%20identification.pdf.

4. Аракелова А.О. Спам. Історія виникнення. Методи боротьби // Інформатика та інформаційні технології: студ. наук. конф., 20 квітня 2015 р.: матер. конф. – Одеса: ОНЕУ, 2015. – С. 99-102.

5. Гудзь Н.О. Інтернет-дискурс як новий тип комунікації: структура, мовне оформлення, жанрові формати // Сучасні лінгвістичні студії: навч. посіб. – Житомир: ЖДУ ім. І. Франка, 2015. – С. 61-87.

6. Загнітко А.П. Основи дискурсології: науково-навчальне видання. – Донецьк: ДонНУ, 2008. – 194 с.

7. Краснобаєва-Чорна Ж.В. Дискурсологія: теоретико-прикладний

аспект: навч. посіб. – Вінниця, 2017. – 110 с.

8. Маринин С.А. Борьба со спамом и вирусами. – М.: НТ Пресс, 2007. – 48 с.

9. Селіванова О.О. Сучасна лінгвістика: напрями та проблеми: підручник. – Полтава: Довкілля-К, 2008. – 712 с.

10. Mohammad R. An assessment of features related to phishing websites using an automated technique [Електронний ресурс] / R. Mohammad, F. Thabtah, L. McCluskey. – 2012. – Режим доступу до ресурсу: https://www.researchgate.net/publication/261081735_An_assessment_of_features_related_to_phishing_websites_using_an_automated_technique.

11. Sheng S. An Empirical Analysis of Phishing Blacklists [Електронний ресурс] / S. Sheng, G. Warner, L. Cranor. – 2009. – Режим доступу до ресурсу: https://www.researchgate.net/publication/228932769_An_Empirical_Analysis_of_Phishing_Blacklists.

12. Phishing [Електронний ресурс]. – Режим доступу до ресурсу: http://www.technicalinfo.net/papers/Phishing.html.

13. Як працюють вбудовані фішинг та захист від шкідливих програм [Електронний ресурс] // Mozilla. – Режим доступу до ресурсу: https://support.mozilla.org/ru/kb/kak-rabotayut-vstroennye-fishing-i-zashita-ot-vred.

14. Яіцький А.О., Сахаров М.Г. Застосування хмарних технологій в космосі // Priority directions of science and technology development. – Київ, 2021. – С. 337-339.

15. The SSL Store. The 12 Most Costly Phishing Attack Examples

[Електронний ресурс] // SSL Store. – 2021. – Режим доступу до ресурсу: https://www.thesslstore.com/blog/the-dirty-dozen-the-12-most-costly-phishing-attack-examples/.

16. Cole M.L. The Complete Guide to Patents, Copyrights, and Trademarks: What You Need to Know Explained Simply. – США: Atlantic Publishing Group Inc., 2008. – 288 с.

17. Jüristo K. How to Conduct Email Phishing Experiments [Електронний ресурс] // Computer Science. – 2018. – Режим доступу до ресурсу: https://1library.net/document/q7x98rky-how-to-conduct-email-phishing-experiments.html.

18. Abu-Nimeh S. A Comparison of Machine Learning Techniques for Phishing Detection [Електронний ресурс] / S. Abu-Nimeh, D. Nappa, X. Wang, S. Nair. – 2017. – Режим доступу до ресурсу: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.114.1242&rep=rep1&type=pdf.

19. Kumaraguru P., Cranshaw J., Acquisti A., Cranor L., Hong J., Blair M.A., Pham T. School of phish: A real-world evaluation of anti-phishing training [Електронний ресурс] // DBLP. – 2009. – Режим доступу до ресурсу: https://www.heinz.cmu.edu/~acquisti/papers/Acquisti_RealWorld_Evaluation_of_Anti-Phishing_Training.pdf.

20. Cost of a Data Breach Report 2021 [Електронний ресурс] // IBM Security. – 2020. – Режим доступу до ресурсу: https://www.ibm.com/downloads/cas/OJDVQGRY.

21. Gupta D. Comparison of classification algorithms to detect phishing

web pages using feature selection and extraction [Електронний ресурс] / Dr. Rajendra Gupta. – 2016. – Режим доступу до ресурсу: https://pdfs.semanticscholar.org/fccd/8ff23734a1947d3efc14d3df9863a5efac6c.pdf.

22. Simpson D., Woodbury B. Phishing trends and techniques [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: https://docs.microsoft.com/en-us/windows/security/threat-protection/intelligence/phishing-trends.

23. Куликова Т., Щербакова Т., Сидорина Т. Спам и фишинг в I квартале 2021 года [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: https://securelist.ru/spam-and-phishing-in-q1-2021/101270/.

24. Symantec Internet Security Threat Report 2018 [Електронний ресурс]. – 2018. – Режим доступу до ресурсу: https://www.phishingbox.com/resources/phishing-stats.

25. Шелест О.І., Гавриленко А.В., Жаркова Г.М., Мурашко А.М., Бубенко П.Т. Атаки в інформаційних системах: підручник. – Київ: Вид-во Ліра-К, 2013. – 276 с.

26. Загальнодержавна програма адаптації законодавства України до законодавства Європейського Союзу [Електронний ресурс]. – 2022. – Режим доступу до ресурсу: https://zakon.rada.gov.ua/laws/show/1629-15#Text.

27. Guidelines on Cyber Security Measures [Електронний ресурс]. – European Union Agency for Cybersecurity. – 2021. – Режим доступу до ресурсу: https://www.enisa.europa.eu/publications/guidelines-on-cyber-security-measures.

28. Victim identity linked to initial access in phishing attacks

[Електронний ресурс] // Recorded Future. – 2022. – Режим доступу до ресурсу: https://www.recordedfuture.com/victim-identity-linked-initial-access-phishing-attacks.

29. Ватаманюк З. Лінгвістика фішингу: новий погляд на старі проблеми // Філологічні науки. – 2020. – № 3. – С. 19-24.

30. Баричев С. Г., Гончаров В. В., Серов Р. Е. Стандарт AES. Алгоритм Rijdael // Основы современной криптографии. – 3-е изд. – М.: Диалог-МИФИ, 2011. – С. 30–35. – 176 с. – ISBN 978-5-9912-0182-7.

31. Sa’id Abdullah Al-Saaidah. Detecting Phishing Emails Using Machine Learning Techniques. – 2017. – pp. 1-72.

32. Merritt D. SPEAR PHISHING ATTACK DETECTION / David Merritt. – 2011. – 128 с.

33. Abdullah Alnajim, Malcolm Munro. An Article of Anti-Phishing Approach that Uses Training Intervention for Phishing Websites Detection [Електронний ресурс] // Researchgate. – 2009. – Режим доступу до ресурсу: https://www.researchgate.net/publication/220841443_An_AntiPhishing_Approach_that_Uses_Training_Intervention_for_Phishing_Websites_Detection.

34. Viktor Krammer. Phishing Defense against IDN Address Spoofing Attacks. – 2006. [Електронний ресурс]. – Режим доступу: http://www.quero.at/papers/idn_spoofing.pdf.

35. Чепрасов Г.А. Методи та засоби захисту інформації в комп'ютерних системах: навч. посіб. – Харків: ХНУРЕ, 2017. – 128 с.

36. Symantec. Internet Security Threat Report 2019 [Електронний ресурс]. – 2019. – Режим доступу до ресурсу: [https://www.symantec.com/content/dam/symantec/docs/reports/istr-24-2019-

en.pdf](https://www.symantec.com/content/dam/symantec/docs/reports/istr-24-2019-en.pdf).

37. Keris V., Piltenko M. Modern Approaches to Phishing Detection // International Journal of Computer Applications. – 2020. – Vol. 175, No. 15. – P. 1-7.

38. IBM X-Force Threat Intelligence Index 2021 [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: https://www.ibm.com/security/data-breach/threat-intelligence.

39. Microsoft Security Intelligence Report Volume 25 [Електронний ресурс]. – 2020. – Режим доступу до ресурсу: https://info.microsoft.com/rs/157-GQE-382/images/EN-Microsoft-Intelligence-Report-Vol25.pdf.

40. Сідорова О.І., Кравченко Л.М. Інформаційна безпека: підручник. – Харків: ХНУРЕ, 2018. – 320 с.

41. Finjan. Web Security Trends Report Q2 2021 [Електронний ресурс]. – 2021. – Режим доступу до ресурсу: https://www.finjan.com/web-security-trends-q2-2021.

42. Pandey A., Mishra M. Machine Learning Techniques for Detecting Phishing Websites // International Journal of Engineering Research & Technology. – 2019. – Vol. 8, No. 3. – P. 120-123.

43. Niu Y., Liu L. What Makes a Phishing Website? // 2008 International Conference on Information Security. – 2008. – P. 132-144.

44. Goel S., Jain A. Mobile Phishing Attacks and Defense Mechanisms // International Journal of Network Security & Its Applications. – 2018. – Vol. 10, No. 2. – P. 53-61.

45. Ahmad I., Arshad J., Rehman H. M. The State of the Art in Phishing Attack Countermeasures // Computers & Security. – 2021. – Vol. 98. – P. 102-113.

46. Zielinska K., Mazurczyk W. Advanced Persistent Threats: The State of the Art and the Concept of Prevention // Journal of Cybersecurity. – 2021. – Vol. 7, No. 2. – P. 1-12.

47. Zuckerman E. Digital Security and Privacy for Human Rights Defenders // MIT Press. – 2020. – 256 с.

48. Ящук О. М., Білик С. В. Захист інформації в комп'ютерних системах: навч. посіб. – Львів: ЛНУ ім. І. Франка, 2019. – 310 с.

ДОДАТКИ

Додаток А1

Закон з Кримінального Кодексу України про шахрайство

Стаття 190. Шахрайство

1. Заволодіння чужим майном або придбання права на майно шляхом обману чи зловживання довірою (шахрайство) -

карається штрафом від двох тисяч до трьох тисяч неоподатковуваних мінімумів доходів громадян або громадськими роботами на строк від двохсот до двохсот сорока годин, або виправними роботами на строк до двох років, або обмеженням волі на строк до трьох років.

2. Шахрайство, вчинене повторно, або за попередньою змовою групою осіб, або таке, що завдало значної шкоди потерпілому, -

карається штрафом від трьох тисяч до чотирьох тисяч неоподатковуваних мінімумів доходів громадян або виправними роботами на строк від одного до двох років, або обмеженням волі на строк до п'яти років, або позбавленням волі на строк до трьох років.

3. Шахрайство, вчинене у великих розмірах, або шляхом незаконних операцій з використанням електронно-обчислювальної техніки, -

карається позбавленням волі на строк від трьох до восьми років.

4. Шахрайство, вчинене в особливо великих розмірах або організованою групою, -

карається позбавленням волі на строк від п'яти до дванадцяти років з конфіскацією майна.

{Стаття 190 із змінами, внесеними згідно із Законами № 270-VI від 15.04.2008, № 2617-VIII від 22.11.2018*}*

Додаток А2

Закон з Кримінального Кодексу України про крадіжку

Стаття 185. Крадіжка

1. Таємне викрадення чужого майна (крадіжка) -

карається штрафом від однієї тисячі до трьох тисяч неоподатковуваних мінімумів доходів громадян або громадськими роботами на строк від вісімдесяти до двохсот сорока годин, або виправними роботами на строк до двох років, або арештом на строк до шести місяців, або обмеженням волі на строк до п’яти років.

2. Крадіжка, вчинена повторно або за попередньою змовою групою осіб, -

карається арештом на строк від трьох до шести місяців або обмеженням волі на строк до п'яти років або позбавленням волі на той самий строк.

3. Крадіжка, поєднана з проникненням у житло, інше приміщення чи сховище або що завдала значної шкоди потерпілому, -

карається позбавленням волі на строк від трьох до шести років.

4. Крадіжка, вчинена у великих розмірах чи в умовах воєнного або надзвичайного стану, -

карається позбавленням волі на строк від п'яти до восьми років.

5. Крадіжка, вчинена в особливо великих розмірах або організованою групою, -

карається позбавленням волі на строк від семи до дванадцяти років з конфіскацією майна.

Примітка. 1. У статтях 185, 186 та 189-191 повторним визнається кримінальне правопорушення, вчинене особою, яка раніше вчинила будь-яке із кримінальних правопорушень, передбачених цими статтями або статтями 187, 262 цього Кодексу.

2. У статтях 185, 186, 189 та 190 цього Кодексу значна шкода визнається із врахуванням матеріального становища потерпілого та якщо йому

спричинені збитки на суму від ста до двохсот п’ятдесяти неоподатковуваних мінімумів доходів громадян.

3. У статтях 185-191, 194 цього Кодексу у великих розмірах визнається кримінальне правопорушення, що вчинене однією особою чи групою осіб на суму, яка в двісті п’ятдесят і більше разів перевищує неоподатковуваний мінімум доходів громадян на момент вчинення кримінального правопорушення.

4. У статтях 185-187 та 189-191, 194 цього Кодексу в особливо великих розмірах визнається кримінальне правопорушення, що вчинене однією особою чи групою осіб на суму, яка в шістсот і більше разів перевищує неоподатковуваний мінімум доходів громадян на момент вчинення кримінального правопорушення.

{Стаття 185 із змінами, внесеними згідно із Законами № 270-VI від 15.04.2008, № 1449-VI від 04.06.2009, № 2617-VIII від 22.11.2018, № 2117-IX від 03.03.2022}

Додаток Б

Код програми захисту від фішингу

```
import re
from urllib.parse import urlparse
import numpy as np
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords

nltk.download('punkt')
nltk.download('stopwords')


def extract_features(url):
    features = {}
    parsed_url = urlparse(url)

    features['url_length'] = len(url)
    features['hostname_length'] = len(parsed_url.hostname) if
parsed_url.hostname else 0
    features['path_length'] = len(parsed_url.path)

    features['num_dots'] = url.count('.')
    features['num_hyphens'] = url.count('-')
    features['num_at'] = url.count('@')
    features['num_slash'] = url.count('/')
    features['num_question'] = url.count('?')
    features['num_equal'] = url.count('=')

    tokens = word_tokenize(url)
    features['num_tokens'] = len(tokens)
    stop_words = set(stopwords.words('english'))
    features['num_stopwords'] = len([word for word in tokens if word in
stop_words])

    return features


# Example dataset
data = {
    'url': [
        'http://example.com',
        'http://phishingsite.com/login',
        'https://secure.example.com/account',
        'http://malicious-site.com/?user=admin',
        'https://legitimate-site.com/home'
    ],
    'label': [0, 1, 0, 1, 0]
}

df = pd.DataFrame(data)

feature_list = []
for url in df['url']:
```

```
    features = extract_features(url)
    feature_list.append(features)

features_df = pd.DataFrame(feature_list)
X = features_df
y = df['label']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

classifier = RandomForestClassifier(n_estimators=100, random_state=42)
classifier.fit(X_train, y_train)

y_pred = classifier.predict(X_test)

print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))


def classify_url(url):
    features = extract_features(url)
    features_df = pd.DataFrame([features])
    prediction = classifier.predict(features_df)
    return 'Phishing' if prediction[0] == 1 else 'Legitimate'


new_url = 'http://suspicious-site.com/verify'
result = classify_url(new_url)
print(f'The URL "{new_url}" is classified as: {result}')
```

```
import email

import re

import numpy as np

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.model_selection import train_test_split

from sklearn.naive_bayes import MultinomialNB

from sklearn.metrics import accuracy_score, classification_report



class EmailFilter:

    def __init__(self):

        self.vectorizer = TfidfVectorizer()

        self.classifier = MultinomialNB()



    def preprocess_text(self, text):

        tokens = re.findall(r'\b\w+\b', text.lower())
```

```
        words = [word for word in tokens if word.isalpha() and len(word) > 1]
        return " ".join(words)

    def extract_email_text(self, raw_email):
        try:
            msg = email.message_from_string(raw_email)
            if msg.is_multipart():
                for part in msg.walk():
                    if part.get_content_type() == "text/plain":
                        return part.get_payload(decode=True).decode("utf-8",
"ignore")
            else:
                return msg.get_payload(decode=True).decode("utf-8", "ignore")
        except Exception as e:
            print(f"An error occurred while extracting email text: {e}")
            return ""

    def train(self, X, y):
        X_tfidf = self.vectorizer.fit_transform(X)
        self.classifier.fit(X_tfidf, y)

    def predict(self, email_text):
        email_text_preprocessed = self.preprocess_text(email_text)
        email_tfidf = self.vectorizer.transform([email_text_preprocessed])
        return self.classifier.predict(email_tfidf)

with open("spam_emails.txt", "r") as f:
    spam_emails = [email.strip() for email in f.readlines()]
with open("ham emails.txt", "r") as f:
    ham_emails = [email.strip() for email in f.readlines()]

X = np.array(spam_emails + ham_emails)
```

```
y = np.array([1] * len(spam_emails) + [0] * len(ham_emails))



X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)



email_filter = EmailFilter()

email_filter.train(X_train, y_train)



y_pred = email_filter.predict(X_test)



print("Accuracy:", accuracy_score(y_test, y_pred))

print("Classification Report:\n", classification_report(y_test, y_pred))



while True:

    email_text = input("Enter the email text to classify (or 'exit' to quit):
")

    if email_text.lower() == "exit":

        break

    prediction = email_filter.predict(email_text)

    if prediction[0] == 1:

        print("Warning: This email is likely spam.")

    else:

        print("This email appears to be legitimate.")

    feedback = input("Was the classification correct? (yes/no): ")

    if feedback.lower() == "yes":

        continue

    elif feedback.lower() == "no":

        correct_label = int(input("Enter the correct label (1 for spam, 0 for
legitimate): "))

        X train = np.append(X train, email text)

        y_train = np.append(y_train, correct_label)

        email_filter.train(X_train, y_train)

        print("Thank you for your feedback. The model has been updated.")
```

```
    else:
        print("Invalid feedback. Please enter 'yes' or 'no'.")
```

```
from flask import Flask, render_template, request, redirect, url_for
from flask_sqlalchemy import SQLAlchemy
from datetime import datetime

app = Flask(__name__)
app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///phishing_reports.db'
db = SQLAlchemy(app)

class PhishingReport(db.Model):
    id = db.Column(db.Integer, primary_key=True)
    sender = db.Column(db.String(100), nullable=False)
    subject = db.Column(db.String(100), nullable=False)
    date_reported = db.Column(db.DateTime, nullable=False,
default=datetime.utcnow)

    def __repr__(self):
        return f"Phishing Report('{self.sender}', '{self.subject}',
'{self.date_reported}')"

@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        sender = request.form['sender']
        subject = request.form['subject']
        new_report = PhishingReport(sender=sender, subject=subject)
        db.session.add(new_report)
        db.session.commit()
        return redirect(url_for('index'))
    else:
        reports =
PhishingReport.query.order_by(PhishingReport.date_reported.desc()).all()
```

```
        return render_template('index.html', reports=reports)


if __name__ == "__main__":
    app.run(debug=True)
```

```
// Background script (background.js)


// Function to analyze URL and detect phishing
function analyzeURL(url) {
    // Perform URL analysis and phishing detection logic
    // For demonstration purposes, we'll assume all URLs are legitimate
    return { isPhishing: false, reason: "None" };
}


// Event listener for when a new tab is created
chrome.tabs.onCreated.addListener(function(tab) {
    // Analyze URL of the new tab
    const url = tab.url;
    const { isPhishing, reason } = analyzeURL(url);
    if (isPhishing) {
        // Redirect the user to a warning page
        chrome.tabs.update(tab.id, { url: "phishing_warning.html" });
        // Send a notification to the user
        chrome.notifications.create({
            type: "basic",
            iconUrl: "warning_icon.png",
            title: "Phishing Warning",
            message: `The website you're trying to visit (${url}) is
suspected to be a phishing site. Proceed with caution.`,
            priority: 2
        });
    }
});
```

```
// Content script (content.js)

// Function to extract and analyze URL of the current page
function analyzeCurrentURL() {
    const currentURL = window.location.href;
    const { isPhishing, reason } = analyzeURL(currentURL);
    if (isPhishing) {
        // Display a warning banner on the page
        const warningBanner = document.createElement("div");
        warningBanner.style.position = "fixed";
        warningBanner.style.top = "0";
        warningBanner.style.left = "0";
        warningBanner.style.width = "100%";
        warningBanner.style.backgroundColor = "red";
        warningBanner.style.color = "white";
        warningBanner.style.padding = "10px";
        warningBanner.style.textAlign = "center";
        warningBanner.textContent = `Warning: Phishing Site - ${reason}`;
        document.body.appendChild(warningBanner);
    }
}

// Execute the function when the page is fully loaded
window.addEventListener("load", analyzeCurrentURL);
```

```
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Phishing Reporting Tool</title>
</head>
```

```
<body>

    <h1>Phishing Reporting Tool</h1>

    <form action="/" method="post">

        <label for="sender">Sender:</label><br>

        <input type="text" id="sender" name="sender" required><br>

        <label for="subject">Subject:</label><br>

        <input type="text" id="subject" name="subject" required><br><br>

        <button type="submit">Report Phishing</button>

    </form>

    <hr>

    <h2>Phishing Reports</h2>

    <ul>

        {% for report in reports %}

            <li>{{ report.sender }} - {{ report.subject }} (Reported on {{
report.date_reported.strftime('%Y-%m-%d %H:%M:%S') }})</li>

        {% endfor %}

    </ul>

</body>

</html>
```

```
from flask import Flask, render_template, request, redirect, url_for

from flask_sqlalchemy import SQLAlchemy

from datetime import datetime



app = Flask(__name__)

app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///phishing_reports.db'

db = SQLAlchemy(app)



class PhishingReport(db.Model):

    id = db.Column(db.Integer, primary_key=True)

    sender = db.Column(db.String(100), nullable=False)

    subject = db.Column(db.String(100), nullable=False)
```

```
    date_reported = db.Column(db.DateTime, nullable=False,
default=datetime.utcnow)



    def __repr__(self):

        return f"Phishing Report('{self.sender}', '{self.subject}',
'{self.date_reported}')"



@app.route('/', methods=['GET', 'POST'])

def index():

    if request.method == 'POST':

        sender = request.form['sender']

        subject = request.form['subject']

        new_report = PhishingReport(sender=sender, subject=subject)

        db.session.add(new_report)

        db.session.commit()

        return redirect(url_for('index'))

    else:

        reports =
PhishingReport.query.order_by(PhishingReport.date_reported.desc()).all()

        return render_template('index.html', reports=reports)



if __name__ == "__main__":

    app.run(debug=True)
```

```
import email

import re

import numpy as np

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.model_selection import train_test_split

from sklearn.naive_bayes import MultinomialNB

from sklearn.metrics import accuracy_score, classification_report



def preprocess_text(text):

    tokens = re.findall(r'\b\w+\b', text.lower())
```

```
    words = [word for word in tokens if word.isalpha() and len(word) > 1]
    return " ".join(words)

def extract_email_text(raw_email):
    try:
        msg = email.message_from_string(raw_email)
        if msg.is_multipart():
            for part in msg.walk():
                if part.get_content_type() == "text/plain":
                    return part.get_payload(decode=True).decode("utf-8",
"ignore")
        else:
            return msg.get_payload(decode=True).decode("utf-8", "ignore")
    except Exception as e:
        print(f"An error occurred while extracting email text: {e}")
        return ""

with open("spam_emails.txt", "r") as f:
    spam_emails = [preprocess_text(email) for email in f.readlines()]
with open("ham_emails.txt", "r") as f:
    ham_emails = [preprocess_text(email) for email in f.readlines()]

X = np.array(spam_emails + ham_emails)
y = np.array([1] * len(spam_emails) + [0] * len(ham_emails))

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

vectorizer = TfidfVectorizer()
X_train_tfidf = vectorizer.fit_transform(X_train)
X_test_tfidf = vectorizer.transform(X_test)

classifier = MultinomialNB()
```

```
classifier.fit(X_train_tfidf, y_train)


y_pred = classifier.predict(X_test_tfidf)


print("Accuracy:", accuracy_score(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))


new_email = """
From: sender@example.com
To: recipient@example.com
Subject: Claim Your Prize Now!

Congratulations! You've won a prize! Click here to claim it.
"""
new_email_text = extract_email_text(new_email)
new_email_text_preprocessed = preprocess_text(new_email_text)
new_email_tfidf = vectorizer.transform([new_email_text_preprocessed])
prediction = classifier.predict(new_email_tfidf)
if prediction[0] == 1:
    print("Warning: This email is likely spam.")
else:
    print("This email appears to be legitimate.")
```

```
import email
import re
import datetime



def analyze_email_header(raw_email):
    try:
        msg = email.message_from_string(raw_email)
```

```
        sender = msg.get("From")
        subject = msg.get("Subject")
        date = msg.get("Date")
        to = msg.get("To")
        cc = msg.get("Cc")
        bcc = msg.get("Bcc")
        message_id = msg.get("Message-ID")
        reply_to = msg.get("Reply-To")
        return_path = msg.get("Return-Path")




        if sender and not validate_sender(sender):
            print("Warning: Possible spoofed sender address detected.")

        if sender and not validate_sender_domain(sender):
            print("Warning: Suspicious sender domain detected.")

        if not message_id or not validate_message_id(message_id):
            print("Warning: Missing or invalid Message-ID detected.")

        if date and not validate_date(date):
            print("Warning: Unusual date/time format detected.")

        print(f"From: {sender}")
        print(f"To: {to}")
        print(f"Cc: {cc}")
        print(f"Bcc: {bcc}")
        print(f"Subject: {subject}")
        print(f"Date: {date}")
        print(f"Message-ID: {message_id}")
        print(f"Reply-To: {reply_to}")
```

```
        print(f"Return-Path: {return_path}")

    except Exception as e:
        print(f"An error occurred while analyzing the email header: {e}")


def validate_sender(sender):

    return True


def validate_sender_domain(sender):

    return True


def validate_message_id(message_id):

    pattern = r"<.+@.+\..+>"
    return re.match(pattern, message_id)


def validate_date(date_str):
    try:
        parsed_date = email.utils.parsedate_to_datetime(date_str)
        now = datetime.datetime.now(parsed_date.tzinfo)
        delta = now - parsed_date
        return abs(delta.total_seconds()) < 24 * 3600  # Within 24 hours
    except Exception as e:
        return False
```

```
# Example usage

if __name__ == "__main__":

    # Example raw email message (put content under)

    raw_email = """

    From: sender@example.com

    To: recipient@example.com

    Subject: Example Email

    Date: Mon, 01 Jan 2024 12:00:00 +0000

    Message-ID: <1234567890@example.com>

    """

    analyze_email_header(raw_email)
import email

import requests

from nltk.tokenize import word_tokenize

from nltk.corpus import stopwords

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.naive_bayes import MultinomialNB

from sklearn.pipeline import make_pipeline




def preprocess_text(text):

    tokens = word_tokenize(text.lower())

    words = [word for word in tokens if word.isalpha() and word not in
stopwords.words("english")]

    return " ".join(words)




def extract_email_text(raw_email):

    try:

        msg = email.message_from_string(raw_email)

        if msg.is_multipart():

            for part in msg.walk():

                if part.get_content_type() == "text/plain":
```

```
                    return part.get_payload(decode=True).decode("utf-8",
"ignore")

        else:

            return msg.get_payload(decode=True).decode("utf-8", "ignore")

    except Exception as e:

        print(f"An error occurred while extracting email text: {e}")

        return ""




def train_phishing_classifier():

    phishing_emails = ["URGENT: Verify your account now to prevent account
suspension.",

                       "Click here to claim your prize! Limited time offer.",

                       "Your account has been compromised. Please update your
password immediately."]

    legitimate_emails = ["Dear Customer, Your monthly statement is ready for
viewing.",

                         "Reminder: Your appointment is scheduled for
tomorrow.",

                         "Thank you for your recent purchase. Here is your
receipt."]

    phishing_texts = [preprocess_text(email) for email in phishing_emails]

    legitimate_texts = [preprocess_text(email) for email in
legitimate_emails]

    X_train = phishing_texts + legitimate_texts

    y_train = ["phishing"] * len(phishing_texts) + ["legitimate"] *
len(legitimate_texts)

    model = make_pipeline(TfidfVectorizer(), MultinomialNB())

    model.fit(X_train, y_train)

    return model




def detect_phishing_email(raw_email, model):

    email_text = extract_email_text(raw_email)

    if not email_text:

        print("Error: Unable to extract email text.")
```

```
        return False

    processed_text = preprocess_text(email_text)

    phishing_probability = model.predict_proba([processed_text])[0][1]

    return phishing_probability > 0.5




if __name__ == "__main__":

    raw_email = """

    From: sender@example.com

    To: recipient@example.com

    Subject: Verify Your Account Now!


    Dear customer,


    Your account has been flagged for suspicious activity. Please click the
link below to verify your account and prevent account suspension:


    https://example.com/verify


    Thank you,

    Customer Support

    """

    phishing_classifier = train_phishing_classifier()

    is_phishing = detect_phishing_email(raw_email, phishing_classifier)

    if is_phishing:

        print("Warning: Phishing email detected!")

    else:

        print("Email appears to be legitimate.")
```

UNICHECK
by Turnitin

Ім'я користувача:
Комп'ютерної математики та інформаційної безпеки...

ID перевірки:
1016284162

Дата перевірки:
26.05.2024 16:43:31 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
26.05.2024 16:54:08 EEST

ID користувача:
100005746

Назва документа: Диплом_Духненко Анастасія

Кількість сторінок: 69 Кількість слів: 12382 Кількість символів: 94848 Розмір файлу: 2.32 MB ID файлу: 1016077983

2.88%
Схожість

Найбільша схожість: 0.32% з Інтернет-джерелом (https://learn.ztu.edu.ua/pluginfile.php/145514/mod_resource/content/.

2.08% Джерела з Інтернету | 137 ..... Сторінка 71

1.97% Джерела з Бібліотеки | 201 ..... Сторінка 72

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Модифікації

Виявлено модифікації тексту. Детальна інформація доступна в онлайн-звіті.

Замінені символи | 27