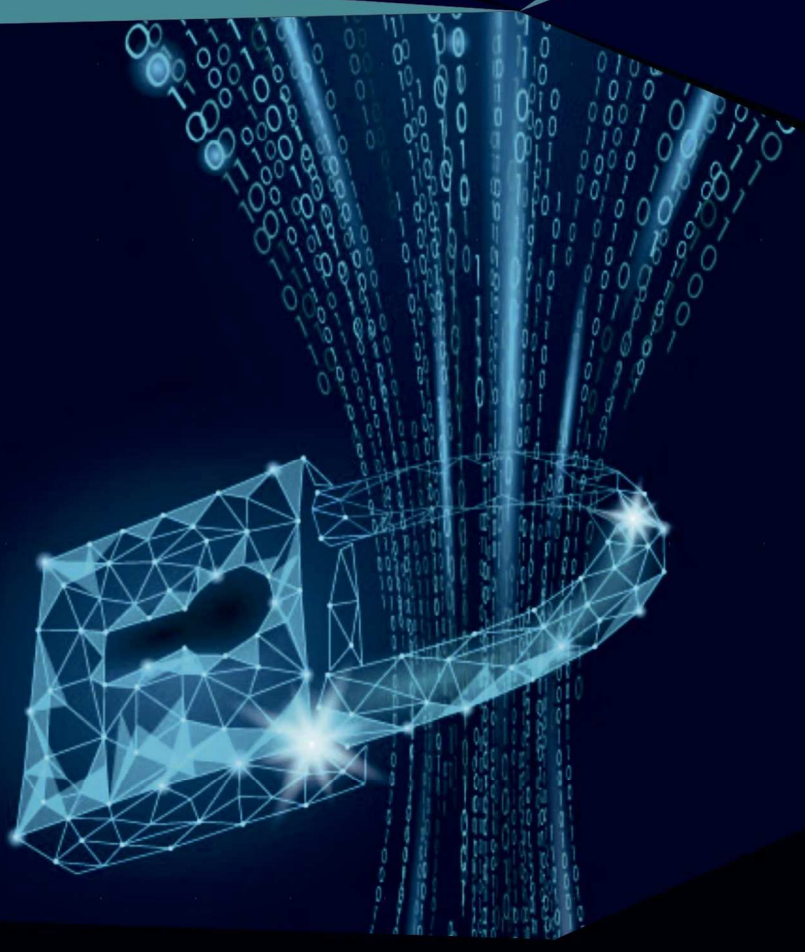
КІБЕРБЕЗПЕКА ТА КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ



2025

науково-практична конференція молодих вчених, аспірантів та студентів

ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ПОЛІТЕХНІКА»
ГАЛИЦЬКИЙ ФАХОВИЙ КОЛЕДЖ ІМ. В'ЯЧЕСЛАВА ЧОРНОВОЛА

КІБЕРБЕЗПЕКА ТА КОМП'ЮТЕРНО-ІНТЕГРОВАНІ ТЕХНОЛОГІЇ (КБКІТ – 2025)

науково-практична конференція
молодих вчених, аспірантів та студентів

28–29 серпня 2025
Тернопіль

Збірник матеріалів науково-практичної конференції молодих вчених, аспірантів та студентів «Кібербезпека та комп'ютерно-інтегровані технології» (КБКІТ - 2025), Тернопіль, 2025. - 154 с.

Адреса редакції:
Західноукраїнський національний університет, кафедра кібербезпеки,
вул. Олени Теліги 8, м. Тернопіль 46003
Контакти:
*e-mail:*conferencekb@gmail.com

ЗМІСТ

СИСТЕМИ ТА ТЕХНОЛОГІЇ КІБЕРБЕЗПЕКИ

КРИПТОГРАФІЧНІ МЕТОДИ ЗАХИСТУ ІНФОРМАЦІЇ

СПЕЦІАЛІЗОВАНІ КОМП'ЮТЕРНІ СИСТЕМИ ТА ТЕХНОЛОГІЇ

УДК 004.056.53

Інна ЯРОВА, Аліса ВЛАСОВА, Наталія КУШНІРЕНКО

Національний університет «Одеська політехніка»

АНАЛІЗ НОРМАТИВНОЇ БАЗИ ДЛЯ СТВОРЕННЯ МОДЕЛІ ПОРУШНИКА ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Вступ. Забезпечення захисту інформації в кіберпросторі ґрунтується в першу чергу на державній нормативно-правовій базі. Правова частина цієї бази складається із законів України, постанов Кабінету Міністрів України та деяких документів нижчого рівню. Нормативна частина (або нормативна база) являє собою сукупність ДСТУ – державних стандартів, та НД ТЗІ – нормативних документів системи технічного захисту інформації, створених ДССЗЗІ України. Одним з етапів процесу створення комплексних систем захисту інформації є розробка моделі порушника інформаційної безпеки. Цей етап є важливим з точки зору подальшого управління ризиками, адже для ефективного вибору тактик зниження ризиків та подолання наслідків реалізованих загроз необхідно чітко розуміти походження джерела загрози та вектор створюваної ним небезпеки. Визначальними вимогами до моделі порушника є адекватність, тобто відповідність моделі реальному об'єкту, та ступінь формалізації.

Мета: Дослідження нормативної бази технічного захисту інформації та визначення системи нормативних вимог для побудови адекватної формалізованої моделі порушника інформаційної безпеки.

Аналіз нормативних вимог для створення моделі порушника

Згідно із загальноприйнятою термінологією, порушник – користувач, який здійснює несанкціонований доступ до інформації [1]. Використовувана в процесі управління ризиками модель порушника – це абстрактний формалізований або неформалізований опис порушника [1]. Ступінь формалізації в даному документі не обговорюється. Таке визначення, з одного боку, позбавляє розробників КСЗІ необхідності жорсткої регламентації своїх дій. З іншого боку, формалізація процесів дозволяє узагальнювати певний досвід, підвищувати деталізацію, адекватність і точність моделі порушника, що розробляється. Аналіз державних стандартів в сфері кібербезпеки показав, що термін «порушник інформаційної безпеки» в них відсутній, принципи побудови моделі порушника в процесах управління ризиками не є стандартизованими. Єдиним державним стандартом, що містить згадку про джерела загроз, які визначаються на етапі аналізу загроз, є ДСТУ 3396.0-96. Це можна вважати непрямим вказанням на порушника інформаційної безпеки. Джерелами загроз за стандартом є «діяльність розвідок іноземних держав, а також навмисні або ненавмисні дії юридичних і фізичних осіб» [2]. Але в документі немає уточнень щодо реєстрації юридичних осіб або громадянства фізичних осіб, або будь-яких інших ознак. Отже, методологічні засади побудови моделі порушника спираються на документи більш низького рівня – відомчі НД ТЗІ. Згідно із [3], порушник – суб'єкт, який вчиняє навмисні

або випадкові дії, що створюють загрозу для інформації, або випадкова подія, внаслідок настання якої можуть реалізуватися загрози для автоматизованої системи. В процесі моделювання загроз рекомендується використовувати найнесприятливішу комбінацію ознак порушника: вважати, що порушник-суб'єкт є кваліфікованим фахівцем, який має повний обсяг інформації про систему, на яку він здійснює атаку, в тому числі про заходи її захисту. Для випадку, коли в якості порушника моделюється випадкова подія, рекомендується обирати найгірший закон розподілу відносно до системи, яка потребує захисту. Слід зазначити, що введення випадкової події в якості порушника інформаційної безпеки протирічить визначенню порушника, запровадженому в [1]. Положення [4] звужує поняття моделі порушника, концентруючись тільки на характеристиках його дій: це абстрактний формалізований або неформалізований опис дій порушника, який відображає його практичні та теоретичні можливості, апріорні знання, час та місце дії і т. ін. Цей перелік ознак, якими можуть бути описані дії порушника, фактично не завершений і протирічить подальшим вимогам до моделі порушника, наведеним далі в цьому документі, адже вона повинна визначати: можливу мету порушника, із градацією за ступенями небезпечності для автоматизованої системи; категорії осіб, з числа яких може бути порушник; припущення про кваліфікацію порушника; припущення про характер дій порушника. Для кожного параметру моделі порушника [4] наводить характеристики у вигляді рекомендацій. Це дозволяє створити вербально-інформаційну модель порушника, яка має доволі узагальнений вигляд внаслідок малої кількості класифікаційних ознак. Використання положення [3] дозволяє створити модель порушника з більшим ступенем деталізації, оскільки бере до уваги ймовірні дії порушника, його ресурсні можливості, рівень повноважень в системі, використовувані програмні та апаратні засоби. Але даний документ має доволі вузьку сферу застосування (призначений для систем захисту інформації для АТС), розроблений доволі давно і тому певною мірою морально застарів.

Висновок. Незважаючи на актуальність впровадження ризик-орієнтованих підходів в процеси управління інформаційною безпекою, нормативна база в сфері технічного захисту інформації демонструє неузгодженість в питаннях розробки моделі порушника інформаційної безпеки та низький рівень вимог щодо формалізації результатів.

Перелік використаних джерел.

1. НД ТЗІ 1.1-003-99. Термінологія в галузі захисту інформації в комп'ютерних системах від несанкціонованого доступу. [Електронний ресурс]. - Режим доступу: https://cip.gov.ua/ua/news/normativni-dokumenti-sistemi-tzi2024

2. ДСТУ 3396.0-96. Захист інформації. Технічний захист інформації. Основні положення. [Чинний від 01.01.1997]. Вид. офіц. Київ, 1996. 6 с.

3. НД ТЗІ 1.1-001-99. Технічний захист інформації на програмно-керованих АТС загального користування. Основні положення. [Електронний ресурс]. - Режим доступу: https://usts.kiev.ua/wp-content/uploads/2020/07/nd-tzi-1.1-001-99.pdf

4. НД ТЗІ 1.4-001-2000 Типове положення про службу захисту інформації в автоматизованій системі. [Електронний ресурс]. - Режим доступу: https://www.tzi.com.ua/downloads/1.4-001-2000.pdf

УДК 004.9

Юр'єв Д.А., Тимошенко Л.М.

Національний університет «Одеська політехніка»

КІБЕРСИТУАЦІЙНА ОБІЗНАНІСТЬ СПІВРОБІТНИКІВ ОБ'ЄКТУ КРИТИЧНОЇ ІНФРАСТРУКТУРИ

Вступ. В Україні в умовах воєнного стану питання людського фактору є особливо актуальним, оскільки об'єкти критичної інфраструктури постійно перебувають під загрозою кібератак. Необізнаність або помилки співробітників можуть проявлятись у різних формах, серед яких відзначають використання слабких паролів, нехтування правилами кібергігієни, наприклад, перехід за фішинговими посиланнями або встановлення вірусного програмного забезпечення. Відсутність знань або навичок у реагуванні на кіберзагрози, зокрема, фішинг, DDoS-атаки, втрата пристроїв, що містять конфіденційну інформацію, або їх неналежне зберігання призводять до отримання зловмисниками доступу до систем, що за інших обставин було б неможливим.

Одеська обласна державна (військова) адміністрація (ОВА) – ключова державна інституція виконавчої влади і важлива ланка в системі державного управління, особливо в умовах воєнного стану. ОВА виступає координатором роботи органів влади, органів місцевого самоврядування, критичних підприємств державного і приватного сектора в регіоні.

Перехід війни в кіберпростір сприяє зростанню кількості практичних розробок і наукових праць з проблематики кіберситуаційної обізнаності, визначає актуальність досліджень з оцінки й моніторингу ситуаційної обізнаності та інформаційної безпеки для усіх сфер державного управління та держави в цілому.

Мета: Дослідження кіберситуаційної обізнаності співробітників об'єкту критичної інфраструктури, інструментів для опису стану ситуаційної обізнаності, та розробка програмної системи моніторингу кіберситуаційної обізнаності.

Основна частина

Особливу роль ОВА відіграє у підтримці обороноздатності регіону [1], зокрема через мобілізацію ресурсів, активну взаємодію з військовими підрозділами, військовим командуванням, Радою національної безпеки і оборони України, Кабінетом Міністрів України.

Серед основних задач ОВА під час воєнного стану визначають забезпечення безперебійного і стабільного функціонування об'єктів критичної інфраструктури, зокрема, залізничних вузлів, портів, медичних закладів, об'єктів енергетичної інфраструктури вищих навчальних закладів, базових станцій операторів стільникового зв'язку. Таким чином ОВА виступає стратегічним центром управління та захисту, поєднуючи адміністративні, оборонні та координаційні функції для підтримки регіональної стабільності.

Завдання системи із ситуаційною обізнаністю полягає у забезпеченні повністю автономного прийняття рішення інтелектуальною системою у динамічному середовищі. Зі стрімким розвитком інформаційних технологій термін «ситуаційна обізнаність», що з'явився у військовій галузі, набуває

подальшого розвитку. Це означає можливість отримання досить повного і точного набору необхідної для прийняття рішення інформації про ситуації в реальному часі [2]. Такий комплексний підхід у володінні ситуацією актуальний в різних областях, де є великий обсяг інформаційних потоків і високий ступінь ризику, зокрема, в кіберпросторі. Побудова автоматизованого механізму виявлення загроз створить картину ситуаційної обізнаності в кіберпросторі для керівників різних рівнів управління критичної інфраструктури.

Вимоги до кіберзахисту державних електронних інформаційних ресурсів та інформаційної інфраструктури, критичної інфраструктури, до якої належать органи державного управління, встановлені законом. Вони передбачають, насамперед, підвищення обізнаності працівників держорганів у сфері інформаційної безпеки та кібербезпеки шляхом проведення різних тренінгів і навчань, та проведення моніторингу їх результатів. Для оцінки кіберситуаційної обізнаності у цій роботі використано метод тестування. Для визначення експертних оцінок, що є вагами кожної відповіді, використано метод аналізу ієрархій (MAI) - математичний інструмент системного підходу до різних складних проблем прийняття рішень [3]. Згідно MAI складено матрицю порівняння критеріїв, матриці порівняння альтернатив за кожним критерієм, матриці ваги альтернатив і ваги критеріїв, з'ясовано та узгоджено експертні оцінки кіберситуаційної обізнаності співробітників. Враховуючи одержані ваги відповідей у тестах, використовуючи шкалу Лайкерта рівнів обізнаності, розраховано рівень кіберситуаційної обізнаності.

Для реалізації системи використано Python та Michrosoft Excel. Виконано експериментальне дослідження тестування ситуаційної обізнаності та аналіз одержаних результатів моніторингу.

Дане дослідження мотивоване до кіберситуаційної обізнаності співробітників держадміністрації. Співробітники приходять з різних галузей та верств суспільства, володіють різними звичками, досвідом та інтелектом, що в підсумку впливає на їх обізнаність про інформаційну безпеку на робочому місці. З іншого боку, органи державного управління все ще не можуть гарантувати, що у співробітників знання, ставлення і поведінка відповідні і вони автоматично дотримуються правил. Їх кіберситуаційна обізнаність може бути меншою за очікувану або недостатньою настільки, що це загрожуватиме цілісності органу.

Дослідження характеризується кількісною оцінкою індивідуальної обізнаності про інформаційну безпеку. Шляхом її вимірювання можна інтерпретувати, які проблеми чи загрози інформаційної безпеки слід очікувати, використовуючи інформаційні програми обізнаності та кібергігієни в майбутньому. Цей вимір враховує три основні етапи.

1. Побудова кіберситуаційних параметрів обізнаності. Даний етап спрямований на визначення сфери ситуаційного дослідження обізнаності. Він встановлює відповідні критерії вимірювання для контексту співробітника. Критерії охоплюють три виміри моделі AIU (Awareness, Ignorance, Uncertainty), що ґрунтуються на дослідженнях Ханша і Бененсона в [1].

2. Оцінка кіберситуаційної обізнаності. На цьому етапі використовується шкала Лайкерта для вивчення того, як співробітники узгоджуються із твердженнями з кіберситуаційної обізнаності. Цей етап формує середній бал за

відсотковою шкалою. На основі агрегованих оцінок респонденти поділяються на п'ять рівнів обізнаності.

3. Кіберситуаційний аналіз причинно-наслідкових зв'язків. На цьому етапі досліджуються всі активності та можливості, що впливають на результат оцінки кіберситуаційної обізнаності. Це породжує критичні питання співробітників як цілі для поліпшення рівня кіберситуаційної обізнаності в майбутньому.

Відмінним інструментом інформаційної безпеки державної адміністрації є моніторинг. Цей вимір обізнаності співробітників про безпеку використано для спостереження за реагуванням на конкретні питання і ситуації, пов'язані з безпекою. Результати тестування доречно використати для визначення сфер навчання захисту. Згенерований бал і рівень обізнаності можна відстежувати з плином часу як метрику для вимірювання програмних цілей та ініціатив, розробки рекомендацій, для порівняння з колегами по відділу (рисунок1), тощо.

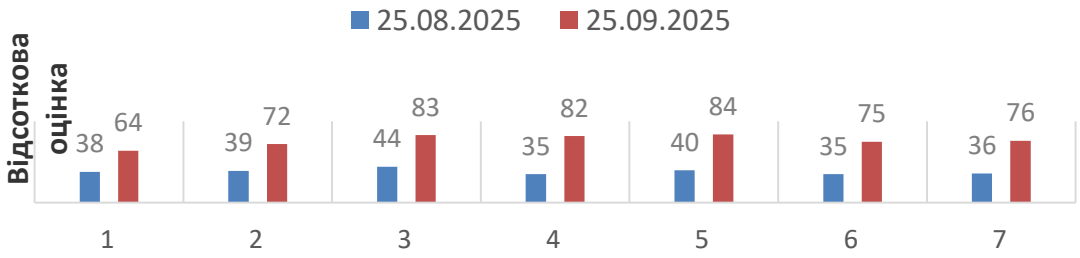


Рисунок 1 - Результати тестування

Висновок. У результаті аналізу існуючого стану захищеності інформаційних ресурсів об'єкту державного управління, зокрема, Одеської державної (військової) адміністрації, вивчили рівень ризику та можливі загрози інформаційній безпеці з боку співробітників. Проведені дослідження методів оцінки ситуаційної обізнаності дозволили обрати комп'ютерне тестування та метод аналізу ієрархій.

Для програмної реалізації моніторингу кіберситуаційної обізнаності обгрунтувано використано середовище Python та Michrosoft Excel. Виконано експериментальне дослідження та проаналізавано одержані результати. Розроблено заходи з підвищення рівня кіберситуаційної обізнаності та рекомендації по їх впровадженню для підвищення захищеності інформаційних ресурсів органу державного управління. За рахунок впровадження заходів з інформаційної безпеки рівень кіберситуаційної обізнаності підвищився на 38%.

Перелік використаних джерел.

1. Микіч Х.І., Буров Є.В. Формальна модель опрацювання знань у системах із ситуаційною обізнаністю. Вісник Національного університету «Львівська політехніка». Інформаційні системи та мережі. 2017. № 872. С. 25-35.

2. Чепурний К., Тимошенко Л. Захист об'єкту критичної інфраструктури в умовах воєнного стану. Інформаційна безпека та інформаційні технології.Зб. матеріалів доп. учасн. V Міжнар. наук.-практ. конф. : Львів, 2024. С. 102-105.

3. Saaty Thomas L. Fundamentals of Decision Making and Priority Theory with the Analytic Hierarchy Process (1994). Pittsburgh: RWS. ISBN 0-9620317-6-3.

УДК 004.056

Чабаненко К.С., Бобок І.І., Кушніренко Н.І.

Національний університет «Одеська політехніка»

МОДЕЛЬ CYBERCRIME-AS-A-SERVICE В СУЧАСНОМУ ЛАНДШАФТІ КІБЕРЗАГРОЗ

Вступ. Кіберзлочинність як послуга (Cybercrime-as-a-Service, CaaS) є однією з найбільш актуальних загроз, що докорінно трансформувала ландшафт кібербезпеки. Ця модель комерціалізувала злочинну діяльність, знизивши технічний поріг входу для зловмисників і зробивши атаки масштабованими та менш витратними. Розвиток тіньових маркетплейсів і спеціалізація акторів (наприклад, брокерів початкового доступу, розробників MaaS) призвели до формування розподіленої, професійної та стійкої злочинної екосистеми, в якій окремі компоненти кібератаки можна придбати або орендувати як послугу. У контексті гібридних конфліктів CaaS стає інструментом для досягнення не лише фінансових, але й геополітичних та руйнівних цілей.

Мета: Аналіз концепції CaaS як сучасної моделі організації кіберзлочинності, виявлення її структурних елементів, економічних та технологічних чинників розвитку, а також оцінка ризиків, які вона створює для системи кібербезпеки держави та бізнесу.

Основна частина

Ринок CaaS дуже динамічно зростає, пропонуючи клієнтам все більше різноманітних послуг. Його особливістю є запровадження в традиційний вектор атаки від розробника до жертви третю сторону: клієнта, який за плату отримує готові експлойт-набори та інструменти для виконання атак, не пишучи коду й не шукаючи вразливостей самостійно, а також орієнтація на «сервісність»: постачальники часто забезпечують цілодобову технічну підтримку, форуми для спілкування й детальні покрокові інструкції, що робить виконання атак доступнішим. Послуги можуть надаватися за різними моделями: фіксована щомісячна підписка, одноразова ліцензійна плата, відсоток від прибутку клієнтів або комбінування цих підходів. Основними моделями екосистеми CaaS є Malware-as-a-service (MaaS), Phishing-as-a-service (PaaS/PhaaS) та DDoS-as-a-service (DaaS).

Оператори MaaS пропонують широкий спектр різних типів шкідливого програмного забезпечення, найпоширенішими прикладами є: інформаційні викрадачі, завантажувачі, бекдори, шпигуньське програмне забезпечення, кейлогери, трояни і тд. Окремою, спеціалізованою реалізацією моделі MaaS є Ransomware-as-a-Service (RaaS): окрім самого шифрувальника вона зазвичай пропонує повну інфраструктуру для атаки - панель керування, канали переговорів і механізми прийому викупу - і заробляє в основному за рахунок частки від сплачених жертвами викупів.

Прикладом відомого постачальника послуг RaaS є угруповання LockBit, яке має прихований вебпортал у мережі Tor. Після реєстрації афілійовані користувачі отримували доступ до панелі керування, де могли переглядати список жертв,

статус шифрування систем, а також публікувати викрадені дані у разі відмови від сплати викупу - така тактика відома як «подвійне вимагання» (double extortion).

Сервіси PaaS надають клієнтам можливість придбання готових фішингових комплектів (phishing kits). Наприклад, популярна PaaS платформа LabHost надавала такі комплекти, що могли містити готові рішення для перехоплення 2FA-кодів через проксування трафіку (Adversary-in-the-Middle), велику бібліотеку фішингових шаблонів, можливість замовити індивідуальні фішингові сторінки під бренд-ціль, автоматичну розгортку на VPS та аналітику.

Модель, що дозволяє клієнтам оплачувати проведення DDoS-атак проти визначених цілей має назву DDoS-as-a-Service (DaaS), або також можна зустріти під назвою «DDoS-for-hire-services» та «Botnets-for-hire-services». Сервіси для проведення атак на відмову в обслуговуванні поділяються на легітимні «stressers» (для тестування власної IT-інфраструктури) та нелегітимні «booters», але, попри декларовану законність, деякі «stressers» не перевіряють право власності на цільовий сервер, що дозволяє їх використання у злочинних цілях. Типовий набір для побудови ботнету включає шкідливий модуль (payload) та інструменти для розгортання та адміністрування C2 (Command-and-Control) інфраструктури.

Для всього світу поширення ринку CaaS призводить до зростання кіберзагроз. Наразі найнебезпечнішими є DDoS, програми-вимагачі та фішинг, зокрема як вектор початкового проникнення для інших типів атак (рисунок 1).

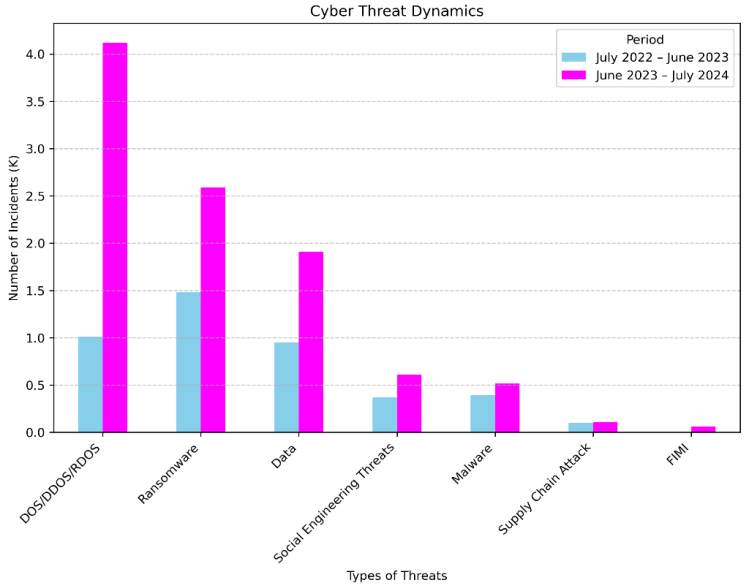


Рисунок 1 - Динаміка зростання кіберзагроз за даними звітів ENISA [1, 2]

Паралельно із розвитком CaaS зросла «площа ураження»: сучасні мобільні пристрої, гаджети та численні IoT/OT-пристрої часто мають слабкий або відсутній базовий захист, що робить їх привабливою мішенню для зловмисників і сприяє швидкому розширенню ботнетів, які використовуються у DDoS-атаках. Ботнет-сервіси часто використовуються хактивістами. Прикладом є платформа DDoSia, пов'язана з групою NoName057, яка, за даними ENISA за 2025 рік, відповідальна за понад 60 % зареєстрованих DDoS-інцидентів у Європі [3]. Також цей приклад пов'язаний із тенденцією переходу від мотивів грошової вигоди до більш складних, зокрема політичних, адже активність групи зростала в періоди, коли ЄС демонстрував підтримку окремих геополітичних ініціатив або під час

національних виборів. Зокрема, однією з причин є підтримка України.

Паралельно програми-вимагачі залишаються одним із найбільш небезпечних інструментів кіберзлочинності, попри певне коливання активності в окремі періоди. Посередники початкового доступу (Initial Access Brokers) продовжують торгувати дешевими VPN- та RDP-доступами, що спрощує проникнення в корпоративні мережі. Загалом фішинг залишається провідним вектором вторгнення - на нього припадає близько 60 % інцидентів, тоді як ще близько 21 % атак починаються з експлуатації відомих вразливостей і завершуються інфікуванням шкідливим ПЗ.

Для України розвиток CaaS має подвійний ефект. З одного боку, держава залишається ціллю численних кібератак у межах російсько-українського протистояння і не тільки, з іншого - досягає помітних результатів у протидії кіберзлочинності. Наприклад, у 2024 році українські правоохоронці спільно з міжнародними партнерами взяли участь у ліквідації угруповання LockBit та проведенні операції Endgame, що знищила інфраструктуру кількох транснаціональних кібергруп [4].

Висновки. Модель Cybercrime-as-a-Service стала однією з найвагоміших трансформацій у розвитку сучасного кіберпростору, докорінно змінивши структуру та механізми функціонування глобальної кіберзлочинності. Її поширення призвело до суттєвого зростання частоти та масштабності кіберінцидентів, серед яких домінують DDoS-атаки, кампанії програм-вимагачів і фішингові операції. Найбільш уразливими до таких загроз залишаються державний сектор і критична інфраструктура, зокрема об'єкти енергетики, телекомунікацій, промислового виробництва та фінансової сфери. Ефективна протидія CaaS вимагає комплексного, багаторівневого підходу, що поєднує розвиток систем моніторингу даркнет-ресурсів, підготовку кваліфікованих фахівців, підвищення рівня кіберобізнаності користувачів і впровадження сучасних архітектур безпеки, орієнтованих на принципи Zero Trust та Secure-by-Design.

Перелік використаних джерел

1. European Union Agency for Cybersecurity (ENISA). ENISA Threat Landscape 2023: ETL. ENISA, 2023. 160 p. [Електроний ресурс] - Режим доступу: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2023

2. European Union Agency for Cybersecurity (ENISA). ENISA Threat Landscape 2024: ETL. ENISA, 2024. 130 p. [Електроний ресурс] - Режим доступу: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2024

3. European Union Agency for Cybersecurity (ENISA). ENISA Threat Landscape 2025: ETL. ENISA, 2025. 86 p. [Електроний ресурс] - Режим доступу: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2025

4. Рада національної безпеки і оборони України. Річний аналітичний огляд (жовтень 2023 – вересень 2024 рр.). – Київ: Аппарат РНБО України, 2024. 32 с. [Електроний ресурс] - Режим доступу: https://www.rnbo.gov.ua

УДК 004.056

Шамарін В.В., Вінковська І.С.

Національний університет «Одеська політехніка»

БЕЗПЕЧНИЙ ОБМІН ДАНИМИ В ДЕЦЕНТРАЛІЗОВАНИХ P2P-СИСТЕМАХ

Вступ. Сучасні інтернет-комунікації активно використовуються для обміну конфіденційною інформацією, тому питання захисту даних стає критично важливим. Централізовані моделі обміну повідомленнями, які базуються на клієнт-серверній архітектурі, мають суттєві недоліки – залежність від одного сервера, ризик перехоплення даних, цензурування та можливість компрометації інформації.

У зв'язку з цим актуальною є створення децентралізованих систем обміну повідомленнями, які базуються на архітектурі peer-to-peer (P2P) та забезпечують наскрізне шифрування.

Мета: Проаналізувати проблеми захисту інформації в сучасних месенджерах і обґрунтувати архітектурні підходи для побудови безпечного P2P-месенджера з інтегрованими криптографічними засобами.

1. Проблематика захисту інформації у сучасних месенджерах

Ключові проблеми більшості популярних месенджерів полягають у централізованій архітектурі, де вся комунікація та метадані користувачів проходять через сервери компаній – власників. Це призводить до таких ризиків:

– несанкціонований доступ до серверів – навіть за наявності наскрізного шифрування, метадані часто залишаються незахищеними;

– витік ключів шифрування – у деяких системах резервні копії чатів або ключі можуть зберігатися на хмарних сервісах, створюючи «єдину точку відмови»;

– недостатній контроль автентичності – існує ризик підміни відправника або повідомлення, якщо не застосовуються надійні механізми цифрового підпису.

Вирішенням цих проблем є перехід до децентралізованої (P2P) моделі, де дані передаються безпосередньо між користувачами, а всі криптографічні операції виконуються локально [1].

2. Застосування криптографічних методів у P2P-архітектурі

Криптографічні методи є основою безпечного P2P-месенджера, що гарантує конфіденційність, цілісність та автентичність переданих даних.

Конфіденційність забезпечується гібридним підходом, який поєднує асиметричне шифрування (ECC, RSA) для безпечного обміну ключами та симетричне шифрування (AES-256, ChaCha20) для швидкої та ефективної передачі великих обсягів даних.

Цілісність та автентичність гарантуються за допомогою цифрових підписів (Ed25519, RSA) та хеш-функцій (SHA-256), що підтверджують справжність відправника та незмінені повідомлення [2].

Для підвищення стійкості до атак типу «людина посередині» (MitM) доцільно впровадити протоколи обміну ключами з властивістю forward secrecy,

наприклад X3DH або Double Ratchet, як це реалізовано в Signal-протоколах [3].

Планується створення програмного засобу, який реалізує комбінований підхід до захисту, інтегруючи зазначені механізми безпосередньо в P2P-протокол.

3. Аналіз проблем та ризиків при побудові P2P-месенджера

Під час створення безпечного P2P-месенджера виникають ключові проблеми, які необхідно враховувати:

– складність управління ключами – безпечний обмін, зберігання та оновлення ключів без центрального сервера є технічно складним завданням;

– проблема ідентифікації – у P2P-мережі складніше забезпечити автентифікацію користувачів і гарантувати унікальність ідентифікаторів;

– людський фактор – помилки користувачів при керуванні ключами або налаштуванні параметрів безпеки можуть послабити ефективність технічних засобів захисту;

– продуктивність – криптографічні операції, особливо асиметричне шифрування, можуть уповільнювати передачу даних, тому необхідна оптимізація алгоритмів.

Також важливим аспектом є захист метаданих (часу, адреси комунікації), адже навіть при повному шифруванні вони можуть розкривати структуру взаємодії користувачів [4].

Висновок. Аналіз сучасних комунікаційних систем підтверджує, що створення безпечного P2P-месенджера з інтегрованими криптографічними механізмами є актуальним завданням кібербезпеки. Основними викликами залишаються управління ключами, автентифікація користувачів та мінімізація впливу людського фактора.

Подальша робота буде зосереджена на формуванні архітектури месенджера, що поєднує гібридне шифрування та цифрові підписи, а також на розробленні прототипу системи з автоматизованими процесами шифрування й перевірки автентичності. Очікуванні результати включають підвищення рівня конфіденційності комунікацій і демонстрацію практичної реалізації безпечних децентралізованих обмінів повідомленнями.

Перелік використаних джерел

1. CyberLab.ua «Загрози месенджера Telegram: що потрібно знати про ризики користування улюбленим месенджером?». 2023. – URL: https://cyberlab.ua/archives/5413

2. Яремчук Ю.Є., Салієва О.В., Бондаренко І.О. Основи криптографічного захисту інформації. Вінниця. 2024. – URL: https://pdf.lib.vntu.edu.ua/books/2024/Yaremchuk_2024_139.pdf

3. MoxieMarlinspike, TrevorPerrin. Signal. The X3DH Key Agreement Protocol. 2016. URL: https://signal.org/docs/specifications/x3dh/

4. Матвій О.В., Мельник В.С., Черевко І.М. Основи комп'ютерних мереж. Чернівці. Навчальний посібник. 2024. – URL: https://archer.chnu.edu.ua/bitstream/handle/123456789/10326/Основи%20комп%27ютерних%20мереж_%20навчальний%20посібник.pdf?sequence=1

УДК 004.056.53:004.912

Власова А.С., Кушніренко Н.І., Назарова І.В.

Національний університет «Одеська політехніка»

АЛГОРИТМ ТЕКСТОВОГО АНАЛІЗУ ДЛЯ ПРОФІЛЮВАННЯ КОРИСТУВАЧІВ В OSINT ДОСЛІДЖЕННЯХ

Вступ. В умовах стрімкого зростання обсягів текстової інформації в соціальних мережах виникає потреба в автоматизованих методах аналізу цифрового сліду користувачів. Текстові дані містять найбільший обсяг інформації про особистість, погляди та поведінкові характеристики користувачів, що робить їх аналіз ключовим компонентом систем OSINT (Open Source Intelligence) [1].

Традиційні методи текстового аналізу часто не враховують морфологічні особливості української мови та специфіку інтернет-комунікації, що знижує точність профілювання. Існуючі міжнародні рішення демонструють точність лише 48-55% для україномовного контенту, що недостатньо для практичного застосування в системах розвідки з відкритих джерел [2]. Додаткову складність створює необхідність обробки змішаного контенту, де поєднуються українська та англійська мови, інтернет-сленг, емоджі та неологізми.

Сучасні дослідження показують, що навіть короткі текстові фрагменти містять достатньо лінгвістичних маркерів для побудови психологічного профілю автора [3], однак для їх ефективного виявлення необхідні спеціалізовані алгоритми, адаптовані до особливостей цільової мови та культурного контексту.

Мета. Підвищити точність профілювання користувачів соціальних медіа в OSINT-дослідженнях шляхом розробки алгоритму комплексного аналізу текстового контенту, адаптованого до української мови та особливостей інтернет-комунікації.

Основна частина

Розроблений алгоритм комплексного аналізу текстового контенту складається з п'яти послідовних етапів обробки, кожен з яких вирішує специфічні задачі аналізу з урахуванням особливостей української мови [4].

Перший етап включає нормалізацію та очищення тексту від технічних артефактів. Процес нормалізації передбачає приведення тексту до стандартного формату UTF-8 з корекцією можливих помилок кодування, видалення HTML тегів та спеціальних символів, які не несуть семантичного навантаження, обробку емоджі зі збереженням їх емоційного значення, оскільки вони є важливим маркером стилю комунікації.

Другий етап реалізує токенізацію та лематизацію з урахуванням морфологічних особливостей української мови. Використовується спеціалізований токенізатор, який враховує українські конструкції з прийменниками з апострофом, складні числівники та назви власні, специфічні інтернет-скорочення та абревіатури. Лематизація виконується з використанням морфологічних словників для української мови, що дозволяє привести всі словоформи до їх основної форми.

Третій етап включає автоматичне визначення мови тексту з використанням бібліотеки langdetect, що важливо для обробки змішаного контенту. Алгоритм

використовує статистичний підхід на основі частотного аналізу символьних n-грам для української та англійської мов. Система здатна визначати мову з точністю 91% для українських текстів, 92% для англійських та 87% для змішаного контенту.

Четвертий етап реалізує сентимент-аналіз текстового контенту для визначення емоційного забарвлення повідомлень користувача [2]. Використовується гібридний підхід, який поєднує словникові методи з елементами частотного аналізу. Система класифікує тексти за трьома основними категоріями емоційного забарвлення: позитивне, негативне та нейтральне. Для української мови створено спеціалізований словник емоційно забарвлених слів та виразів з урахуванням контекстуальних особливостей їх використання. Алгоритм досягає точності 74% для українських текстів, що на 17-19% вище за універсальні міжнародні рішення.

П'ятий етап включає частотний аналіз ключових слів та виділення сутностей з тексту користувача. Алгоритм використовує метод TF-IDF для автоматичного виявлення найбільш важливих термінів у корпусі текстів користувача:

$$TF - IDF(t,d) = \left(\frac{f(t,d)}{max_{freq(d)}}\right) \times \log\left(\frac{N}{|\{d \in D: t \in d\}|}\right), \quad (1)$$

де $f(t,d)$ – частота терміну t у тексті d,

$max\_freq(d)$ – максимальна частота будь-якого терміну в тексті d,

N – загальна кількість текстових фрагментів користувача,

$|\{d \in D: t \in d\}|$ – кількість текстових фрагментів, що містять термін t.

Метод TF-IDF дозволяє виявляти терміни, які є специфічними для конкретного користувача та відрізняють його тексти від загального корпусу. Система аналізує частотні характеристики лексики для визначення тематичних преференцій користувача та побудови профілю його інтересів з точністю 77%.

Експериментальне тестування алгоритму проводилось на вибірці з 100 користувачів україномовних соціальних медіа різних вікових категорій та рівнів активності. Результати точності алгоритмів текстового аналізу для різних типів мовного контенту представлені в таблиці 1.

Таблиця 1 – Результати точності алгоритмів текстового аналізу

Метрика	Українська мова	Англійська мова	Змішаний контент
Визначення мови	91%	92%	87%
Сентимент-аналіз	74%	71%	65%
Виділення ключових слів	77%	74%	69%

Інтеграція всіх п'яти етапів створює комплексну систему текстового аналізу, яка забезпечує багатоаспектну характеристику мовної поведінки користувача. Результати обробки включають лексичний профіль з показником багатства словника, емоційний профіль з розподілом позитивних, негативних та нейтральних висловлювань, тематичний профіль з виявленими ключовими інтересами та сферами діяльності. Ці характеристики використовуються для побудови інтегрованого цифрового профілю користувача та оцінки його

особистісних якостей на основі цифрового сліду.

Порівняльний аналіз розробленого алгоритму з існуючими міжнародними рішеннями показує його переваги саме для української аудиторії. Система враховує понад 800 популярних українських інтернет-скорочень та неологізмів, які активно використовуються в соціальних медіа. Створено спеціалізований словник емоційно забарвлених слів, адаптований до особливостей українського менталітету та культурного контексту, що суттєво підвищує точність сентимент-аналізу порівняно з універсальними багатомовними рішеннями.

Особливу увагу приділено обробці змішаного україно-англійського контенту, який є типовим для сучасної інтернет-комунікації українських користувачів. Алгоритм здатен коректно обробляти тексти з частковим перемиканням мов, транслітерацією українських слів латиницею та використанням англомовних термінів у україномовному контексті.

Висновок. У процесі дослідження було підвищено точність профілювання користувачів соціальних медіа шляхом розробки алгоритму комплексного аналізу текстового контенту, адаптованого до української мови та особливостей інтернет-комунікації. Алгоритм включає п'ять етапів: нормалізація тексту, токенізація та лематизація з морфологічними словниками української мови, визначення мови через langdetect, сентимент-аналіз та частотний аналіз ключових слів методом TF-IDF.

Експериментальне тестування на вибірці з 100 користувачів підтвердило підвищення точності: 74% для сентимент-аналізу українських текстів та 77% для виділення ключових слів, що на 17-19% вище за універсальні міжнародні рішення. Система забезпечує ефективну обробку змішаного контенту з точністю визначення мови 87% та враховує понад 800 популярних українських інтернет-скорочень.

Розроблений алгоритм може бути успішно застосований в OSINT дослідженнях для автоматизованого профілювання користувачів україномовних соціальних медіа та створює основу для побудови інтегрованих цифрових профілів користувачів.

Перелік використаних джерел.

1. Deeva I. Computational Personality Prediction Based on Digital Footprint of A Social Media User. Procedia Computer Science. 2019. Vol. 156. P. 185–193.

2. Birjali M., Kasri M., Beni-Hssane A. A comprehensive survey on sentiment analysis: Approaches, challenges and trends. Knowledge-Based Systems. 2021. Vol. 226. P. 107134. URL: https://doi.org/10.1016/j.knosys.2021.107134

3. Azucar D., Marengo D., Settanni M. Predicting the Big 5 personality traits from digital footprints on social media: A meta-analysis. Personality and Individual Differences. 2018. Vol. 124. P. 150–159.

4. Nandwani P., Verma R. A review on sentiment analysis and emotion detection from text. Social Network Analysis and Mining. 2021. Vol. 11, no. 1. URL: https://doi.org/10.1007/s13278-021-00776-6

УДК 004.056.5

Вікторія ПЯНКОВСЬКА, Інна ЯРОВА

Національний університет «Одеська політехніка»

СУЧАСНІ МЕТОДИ ТЕЛЕФОННОГО ТА ОНЛАЙН-ШАХРАЙСТВА В УКРАЇНІ: МЕТОДІ ПРОТИДІЇ ТА РОЗКРИТТЯ ЗЛОЧИНІВ

Вступ. В епоху цифрових технологій методи соціальної інженерії стають все більш витонченими, а в умовах повномасштабного вторгнення в Україні ця проблема набула особливої гостроти. Зловмисники майстерно адаптують свої схеми під актуальні потреби громадян, спекулюючи на темах евакуації, благодійних зборів та державних виплат. Масштаби загрози підтверджує статистика НБУ: хоча кількість шахрайських операцій у 2024 році дещо зменшилась, загальна сума збитків зросла на 37 % і сягнула 1,1 млрд грн, а середня сума однієї незаконної операції досягла значення 4247 грн [1]. Причому 83 % шахрайських операцій відбулися в мережі Інтернет, що робить кіберпростір основним полем діяльності для злочинців. Ключовою причиною злочинів залишається соціальна інженерія – 84 % збитків сталися через те, що люди самі розголошували свої дані .

Мета: Проведення аналізу поширених та новітніх схем шахрайства в кіберпросторі, розгляд методів протидії з боку правоохоронних органів та надання практичних порад для захисту громадян.

1. Найпоширеніші схеми шахрайства

Спираючись на Конвенцію Ради Європи про кіберзлочини, використання методів соціальної інженерії в кіберпросторі можна кваліфікувати як шахрайство, що пов'язане з використанням цифрових технологій, спрямоване на порушення конфіденційності персональних даних з їх подальшим незаконним використанням, вчинене зовнішнім порушником. Поширеними методами онлайн-шахрайства є фішинг і вішинг, причому вішінг може реалізовуватися як в кіберпросторі з використанням соцмереж та поштових сервісів, так і у вигляді телефонного шахрайства.

Використовуючи фішинг, злочинці створюють фейкові сайти, що копіюють сайти банків, поштових служб, або державні портали (наприклад, «Дія») з метою збору логінів, паролів та даних карток. Новим способом фішингу є створення та поширення фейкового контенту, в якому під виглядом благодійних фондів або державних організацій пропонується надання допомоги малозабезпеченим верствам населення, в тому числі внутрішнім переселенцям [2]. На підконтрольному зловмисникам сайті жертва має ввести свої персональні дані та дані банківської картки.

Також поширеним є шахрайство в інтернет-торгівлі: продаж неіснуючих товарів за передоплатою або надсилання фішингових посилань на «безпечну оплату».

В умовах обмеженого офлайн-спілкування актуальною проблемою є фішинг в соцмережах: злам акаунтів для розсилки повідомлень з проханням позичити гроші та створення фейкових сторінок для псевдоблагодійних зборів.

Вішинг – маніпуляція з використанням онлайн-листування або телефонного дзвінка з метою отримання конфіденційних даних. Основні сценарії вішингу:

– «лист або дзвінок з банку»: під приводом підозрілої активності на рахунку злочинець отримує від жертви CVV-код, паролі з SMS та іншу банківську інформацію;

– «Ви виграли приз»: повідомляючи про виграш, злочинець просить сплатити неіснуючий «податок» або «комісію» для отримання призу;

– «дзвінок від мобільного оператора»: вішинг з використанням телефону, коли під виглядом «покращення зв'язку» жертву просять набрати комбінацію, яка встановлює переадресацію SMS, надаючи доступ до онлайн-банкінгу;

– «родич у біді»: створюючи емоційний шок телефонним повідомленням про ДТП чи затримання родича, зловмисник вимагає перерахувати гроші для «вирішення питання» онлайн-банкінгом.

2. Новітні загрози: Deepfake та атаки на eSIM

Прогрес створює нові вектори атак, залучаючи новітні цифрові технології, які складно розпізнати звичайному користувачу.

Аудіо-дипфейки (Deepfake): злочинець використовує штучний інтелект для клонування голосу. На основі оригінальних зразків голосу злочинець може згенерувати аудіоповідомлення або зателефонувати від лиця знайомої людини: родича, друга чи керівника. Метою цього є отримання термінового переказу грошей або одноразових кодів доступу [3]. Атаки на віртуальні SIM-карти (eSIM): зловмисники отримують контроль над eSIM жертви через підроблені запити до оператора, таким чином перехоплюючи SMS-повідомлення з кодами підтвердження для входу в додатки онлайн-банкінгу [3].

3. Протидія шахрайству в кіберпросторі з боку правоохоронних органів

Ключову роль у боротьбі з кіберзлочинністю відіграє Департамент кіберполіції, який застосовує комплексний підхід. Основні напрямки його діяльності:

– відстеження фінансових транзакцій: аналіз руху коштів через ланцюжки «транзитних» карток для виявлення організаторів злочину;

– блокування шахрайських ресурсів: у співпраці з банками, операторами мобільного зв'язку та інтернет-провайдерами відбувається оперативне блокування фішингових сайтів, номерів телефонів та рахунків;

– ліквідація колл-центрів: правоохоронці регулярно викривають організовані «офіси», облаштовані комп'ютерною технікою та телекомунікаційним обладнанням;

– активна протидія злочинним групам в соцмережах, які ошукують громадян під приводом надання грошової допомоги переселенцям;

– міжнародне співробітництво: взаємодія з Європолом та Інтерполом є важливою, оскільки кіберзлочинність часто має транснаціональний характер.

4. Як захистити себе: практичні поради для користувачів

Ефективним засобом профілактики кіберзлочинності є підвищення обізнаності громадян щодо можливих дій злочинців і способів протидії. Головною

рекомендацією є порада зберігати спокій і критичне мислення при отриманні неочікуваних листів, повідомлень або дзвінків, адже емоції заважають раціональним діям.

Для пересічного користувача мережі можна запропонувати наступні «золоті правила» інформаційної безпеки:

– нікому не повідомляйте конфіденційні дані: CVV-код, термін дії картки, паролі з SMS та PIN-код: справжні співробітники банків їх ніколи не питають;

– перевіряйте інформацію: отримавши тривожний дзвінок, покладіть слухавку та самостійно зателефонуйте до установи (банку, поліції) за офіційним номером;

– будьте уважні до посилань: не переходьте за підозрілими посиланнями з SMS, месенджерів чи електронної пошти, завжди перевіряйте URL-адресу сайту на наявність помилок та чи захищений він (протокол https);

– використовуйте надійний захист: створюйте складні, унікальні паролі для різних акаунтів та обов'язково вмикайте двофакторну автентифікацію;

– окремий фінансовий номер телефону: використовуйте для онлайн-банкінгу номер телефону, який ви не використовуєте для соцмереж та інших сайтів.

Якщо Ви стали жертвою кібершахрайства:

– негайно заблокуйте картку через онлайн-додаток банку або дзвінком на гарячу лінію банку;

– повідомте банк про несанкціоноване списання коштів;

– подайте заяву до Департаменту кіберполіції (онлайн через сайт) та до найближчого відділення Національної поліції України.

Висновки. Шахрайство з використанням цифрових технологій в Україні постійно еволюціонує, використовуючи для маніпуляцій соціальну інженерію та новітні технології, як-от штучний інтелект. Злочинці активно експлуатують актуальні для суспільства теми, що робить їхні атаки більш переконливими. В цих умовах головним інструментом протидії є високий рівень цифрової грамотності громадян. Розуміння механізмів обману, критичне мислення та дотримання базових правил безпеки дозволяють вчасно розпізнати загрозу. Тому підвищення власної пильності та обізнаності є ключовим для ефективного захисту від шахраїв в кіберпросторі.

Перелік використаних джерел.

1. Національний банк України: офіційний сайт. Кількість випадків шахрайства з картками знизилася, збитки за ними – зросли. [Електронний ресурс]. - Режим доступу: https://bank.gov.ua/ua/news/all/kilkist-vipadkiv-shahraystva-z-kartkami-znizilasya-zbitki-za-nimi--zrosli

2. Національна поліція України: офіційний вебпортал. [Електронний ресурс]. - Режим доступу: https://npu.gov.ua/news/dopomoha-z-sizo-kiberpolitsiia-prypynyla-diialnist-shakhraiskoho-uhrupovannia

3. Офіційний сайт Кіберполіції України. Афери з діпфейками: кіберполіція застерігає від шахраїв. [Електронний ресурс]. - Режим доступу: https://cyberpolice.gov.ua/article/afery-z-dipfejkamy-kiberpolicziya-zasterigaye-vid-shaxrayiv-7638/

УДК 004.056

Завадський Д.О., Кушніренко Н.І.

Національний університет «Одеська політехніка»

РОЗРОБКА НАВЧАЛЬНОГО ЗАСТОСУНКУ ДЛЯ ПРОТИДІЇ АТАКАМ СОЦІАЛЬНОЇ ІНЖЕНЕРІЇ

Вступ. Соціальна інженерія є потужним і дедалі більш актуальним чинником кіберзагроз: вона обминає технічні бар'єри, експлуатуючи людський фактор.Згідно з нещодавнім звітом, близько 74% інцидентів інформаційної безпеки пов'язані з людськими помилками, зокрема з використанням психологічних тактик соціальної інженерії [1].

Така статистика підкреслює вразливість організацій і користувачів перед цими атаками. Натомість більшість традиційних навчальних підходів – суто теоретичні курси чи періодичні перевірки за допомогою фішингових листів – не забезпечують глибокого засвоєння знань і практичних навичок. Тож зростає потреба в нових інтерактивних методах навчання, які долають пасивність і формалізм існуючих програм.

Мета: створення веб-застосунку, який у доступній та інтерактивній формі навчає користувачів розпізнавати атаки методами соціальної інженерії.

Основна частина

Ключовою ідеєю запропонованого підходу є додавання елементів гейміфікації в процес навчання - використання механізмів, властивих комп'ютерним іграм (бали, рівні, досягнення, рейтинги), для підвищення мотивації користувачів. Завдяки цьому навчальний процес набуває інтерактивного характеру: замість послідовного виконання інструкцій він перетворюється на динамічну гру, у якій кожне рішення має наслідки..

Застосунок побудовано за модульним принципом і включає кілька рівнів складності:

1. Базовий рівень - користувач знайомиться з основними типами атак: фішинг, смішинг, вішинг, бейтинґ, пре-текстинг. На цьому етапі система демонструє типові приклади обману й пояснює, як розпізнати ознаки маніпуляції.

2. Практичний рівень - користувач отримує завдання у вигляді реалістичних сценаріїв. Наприклад, на екрані з'являється фальшивий лист «від банку» або запит на оновлення паролю, і користувач повинен обрати, як діяти. Залежно від рішення система показує наслідок - успішне уникнення атаки чи умовну «втрату даних» користувача або організації де працює індивід.

3. Поглиблений рівень - тренування з підвищеною складністю, де потрібно аналізувати контекст, поведінку співрозмовника або структуру вебсторінки. Такі завдання розвивають критичне мислення та вчать оцінювати інформацію більш комплексно, та позитивно сприяють на критичне мислення.

Окрім навчальних сценаріїв, у програмі реалізовано систему миттєвого зворотного зв'язку. Після кожної відповіді користувач бачить коротке пояснення - чому обране рішення було правильним або помилковим, які маніпулятивні прийоми застосував зловмисник і як їх розпізнати у майбутньому.

Важливим компонентом стала адаптивність навчання. Застосунок відстежує успішність користувача і автоматично змінює рівень складності наступних завдань. Якщо учасник впевнено проходить базові тести, система пропонує складніші ситуації, що вимагають детальнішого аналізу; якщо ж він робить багато помилок - надає підказки та додаткові приклади.

Такий підхід дозволяє навчатися у власному темпі, що особливо корисно для користувачів з різним рівнем цифрової компетентності. Ще однією перевагою є простота доступу. Застосунок є веборієнтованим - для роботи не потрібно встановлювати додаткове програмне забезпечення. Він адаптований для мобільних пристроїв і персональних комп'ютерів, що робить його універсальним інструментом для освітніх установ, компаній або індивідуального користування [2].

Технічна реалізація проєкту базується на використанні мови програмування JavaScript, що забезпечує високу продуктивність, гнучкість та масштабованість системи. За візуальне відображення використано HTML.

Психологічна складова - одна з головних переваг розробки. Кожен сценарій не лише навчає, а й пояснює, які емоційні тригери використовуються у конкретній атаці: довіра до авторитету, страх, терміновість, співчуття чи цікавість. Таким чином, користувач не просто запам'ятовує набір правил, а розуміє логіку дій зловмисника, що дозволить у подальшому бути більш уважним.

Інтерфейс застосунку створено з урахуванням принципів когнітивної ергономіки: мінімалістичний дизайн, інтуїтивна навігація, короткі інструкції, чітка візуалізація ризиків. Завдяки цьому навчання не перевантажує користувача інформацією, а натомість сприяє концентрації на суті завдання.

Висновок. Запропонований веб-застосунок спрямований на посилення цифрової безпеки користувачів шляхом інтерактивного навчання. Він допомагає практично закріпити навички розпізнавання соціально-інженерних атак і виробити алгоритми реагування на них. Поєднання теоретичних пояснень, ігрових тренінгів та регулярного тестування знань робить навчання ефективнішим. [3]. Завдяки такому комплексному підходу - інтерактивна практика плюс акцент на психологічних механізмах атак - користувачі краще підготовлені і менш схильні до помилок. Таким чином інтерактивне навчання з практичними симуляціями і психологічною обізнаністю сприяє суттєвому підвищенню загального рівня кібербезпеки.

Перелік використаних джерел.

1. Verizon Data Breach Investigations Report 2024. Verizon Enterprise, 2024. 98 p. URL: https://www.verizon.com/business/resources/reports/dbir/ (дата звернення: 11.10.2025).

2. ISO/IEC 27032:2023 Cybersecurity Guidelines. International Organization for Standardization, Geneva, 2023. 65 p..

3. Gartner Research. The Impact of Gamified Cybersecurity Training on Employee Awareness, 2024. 12 p. DOI: https://doi.org/10.1016/gartner.cybersec.2024.0415.

УДК 004.056.5

Валентин БЕВЗ

Західноукраїнський національний університет

АНАЛІЗ АКТУАЛЬНИХ ВРАЗЛИВОСТЕЙ MS OFFICE

Вступ. Аналіз актуальних вразливостей MS Office є надзвичайно важливим через широку поширеність цього програмного забезпечення в урядових, корпоративних та освітніх установах. Зловмисники часто використовують уразливості в MS Office як вектор для фішингу, доставки шкідливого коду та ескалації привілеїв. Регулярне дослідження таких вразливостей дозволяє своєчасно виявляти загрози, зменшувати ризики компрометації систем і підвищувати загальний рівень кіберзахисту. Це робить тему особливо актуальною в умовах зростання кількості кібератак на документообіг та офісні середовища.

Мета: виявлення, класифікація та аналіз актуальних вразливостей у програмному середовищі MS Office, а також оцінка їхнього впливу на інформаційну безпеку користувачів. Особлива увага приділяється методам виявлення та усунення цих вразливостей, а також рекомендаціям щодо зменшення ризиків їх експлуатації.

1. Аналіз сучасних загроз офісним застосункам MS Office

Загальну кількість виявлених уразливостей будемо аналізувати за даними бази CVE. Усього за другий квартал 2024 року там було опубліковано інформацію про 8559 уразливостей. Це не остаточна цифра, оскільки часто дані в цій базі оновлюються «заднім числом». Це трохи більше за показники другого кварталу 2023 року: кількість вразливостей, інформація про які стає публічною, продовжує зростати. Із загальної кількості уразливостей 332 є критичними.

За неповною статистикою за перше півріччя 2024 року можна зробити висновок про зниження частки багів, для яких доступний публічний експлойт або Proof of Concept. Зате зросла кількість інцидентів, у яких використовуються вразливі легітимні драйвери для програмного забезпечення.

Найбільш серйозними вразливостями, що найчастіше використовуються зловмисниками для Windows будуть наступні вразливсоті:

– CVE-2018-0802 – вразливість у компоненті Equation Editor пакету Microsoft Office;

– CVE-2017-11882 – ще одна вразливість у Equation Editor, схема зараження якого приведена на рисунку 1.

– CVE-2017-0199 - вразливість у Microsoft Office та WordPad;

– CVE-2021-40444 - вразливість віддаленого виконання коду в компоненті MSHTML.

У другому кварталі 2024 року було відзначено значне зростання атак на користувачів систем на базі Linux з використанням експлойтів для поширених уразливостей. Серед найчастіше експлуатованих багів два (CVE-2022-0847, CVE-2023-2640) відносяться до ядра системи. Ще одна вразливість (CVE-2021-4034) відноситься до утиліти pkexec, що дозволяє виконувати команди від імені іншого користувача.

Рисунок 1 - Схема вразливості CVE-2017-11882

Якщо «користувальницьке» шкідливе ПЗ експлуатує одні і ті ж уразливості роками, то в атаках на бізнес частіше застосовуються експлойти до нещодавно виявлених проблем в корпоративному ПЗ. виявленим у 2024 році: CVE-2024-3400 для програмного забезпечення Palo Alto Networks, CVE-2024-20353 для рішень Cisco, CVE-2024-1709 у ПЗ для IT-менеджменту ConnectWise, а також відома вразливість CVE-2024e2 Зловмисники, що атакують компанії, шукають насамперед уразливі точки входу в корпоративну мережу та регулярно оновлюють набір інструментів, що використовуються.

2. Аналіз вразливості CVE-2022-30190

30 травня Microsoft розкрила деталі вразливості нульового дня у всіх версіях локального та хмарного офісного пакету MS Office, також надала рекомендації IT-фахівцям із захисту від експлойту, який доступний у мережі деякий час.

Microsoft зареєструвала цю вразливість під номером CVE-2022-30190. Компанія поки що не випустила проти неї патчі, розробники займаються цим інцидентом.

Ця вразливість зазнає всіх версій Microsoft Office з 2016 по 2021 і Office 365. З її допомогою зловмисник може віддалено запустити довільний код. У мережі вже є кілька підтверджень, що ця вразливість використовувалася під час атак. Експерти навели приклад експлойту для цієї вразливості, коли проаналізували шкідливий документ Word 05-2022-0438.doc, нещодавно завантажений на VirusTotal.

12 квітня дослідник Shadowchasing1 повідомив Microsoft про проблему і надіслав до Microsoft Security Response Center (MSRC) приклад експлойту.

21 квітня MSRC закрила тикет, заявивши, що проблема не пов'язана з безпекою, проігнорувавши, що в експлойті відбувається виконання msdt з відключеними макросами.

У травні Microsoft, ймовірно, намагалася виправити цю вразливість у новій тестовій версії Office 365. Компанія не задокументувала CVE щодо цього

інциденту.

27 травня експерти виявили факти застосування зловмисниками цієї вразливості та знову повідомили у MSRC. Заражений документ використовує функцію віддаленого шаблону Word для вилучення HTML-файлу з віддаленого сервера, який використовує URI схему ms-msdt MSProtocol для завантаження коду та виконання скриптів PowerShell. Microsoft Word виконує код через інструмент підтримки ms-msdt навіть за відключених макросів. Захищений перегляд запускається, але якщо змінити документ на формат RTF, захищений перегляд включається навіть без відкриття документа, наприклад, через вкладку попереднього перегляду у Провіднику. На рисунку 2 приведено фрагмент коду з зараженого документа.

```
$cmd = "c:\windows\system32\cmd.exe";Start-Process $cmd -windowstyle hidden -ArgumentList "/c
taskkill /f /im msdt.exe";Start-Process $cmd -windowstyle hidden -ArgumentList "/c cd C:\users
\public\&&for /r %temp% %i in (05-2022-0438.rar) do copy %i 1.rar /y&&findstr TVNDRgAAAA
1.rar>1.t&&certutil -decode 1.t 1.c &&expand 1.c -F:* .&&rgb.exe";
```

Рисунок 2 - Приклад коду, що виконується при запуску спеціально зараженого документа

У результаті Microsoft погодилася, що вразливість дійсно критична і опублікувала додаткові рекомендації з безпеки клієнтів офісного пакету.

Microsoft рекомендує системним адміністраторам вимкнути протокол MSDT URL за допомогою команди "reg delete HKEY_CLASSES_ROOT\ms-msdt /f", попередньо зробивши резервну копію цього ключа реєстру ("reg export HKEY_CLASSES_ROOT\ms-msdt filename").

Також для блокування використання вразливості можна включити в налаштуваннях Microsoft Defender правило для відображення напрямків атаки BlockOfficeCreateProcessRule, яке забороняє програмам Office створювати дочірні процеси.

Microsoft радить в офісних пакетах не відключати в налаштуваннях захисту параметри за замовчуванням Protected View і Application Guard, які також запобігають можливості використання вразливості нульового дня CVE-2022-30190, але не для всіх версій MS Office.

Microsoft пообіцяла випустити незабаром необхідні оновлення для всіх версія MS Office проти нової вразливості.

Висновок. У результаті проведеного аналізу встановлено, що MS Office залишається одним із найбільш привабливих об'єктів для атак через підтримку макросів, складні формати документів і глибоку інтеграцію з операційною системою. Більшість вразливостей пов'язані з соціальною інженерією та використанням шкідливих вкладень у документах. Регулярне оновлення програмного забезпечення, обмеження прав доступу та використання сучасних засобів захисту дозволяють істотно знизити ризики. Отримані результати можуть бути використані для підвищення кіберстійкості як окремих користувачів, так і організацій.

Перелік використаних джерел.

1. Security Update Guide [Електронний ресурс]. – Режим доступу: https://msrc.microsoft.com/update-guide/vulnerability

УДК 004.056.5

Лаковський Б.А., Сиропятов О.А., Тимошенко Л.М.

Національний університет «Одеська політехніка»

ПОТОЧНИЙ СТАН ТА ПРОБЛЕМАТИКА ВПРОВАДЖЕНЯ ЗАХИСТУ ІНФОРМАЦІЇ У ДЕРЖАВНИХ ПРОМИСЛОВИХ СИСТЕМАХ

Вступ.У сучасних умовах цифровізації промисловості питання захисту інформації на державних об'єктах набуває стратегічного значення для національної безпеки. Особливу роль у цьому аспекті відіграють автоматизовані системи управління технологічними процесами (АСУ ТП), які забезпечують безперервне функціонування виробничих і енергетичних об'єктів.

На відміну від звичайних корпоративних мереж, більшість систем АСУ ТП на державних промислових підприємствах мають власну ізольовану мережеву інфраструктуру, спеціалізовані операційні системи, контролери та промислові протоколи зв'язку. Зазвичай такі системи не під'єднані безпосередньо до мережі Інтернет або корпоративних IT-мереж, що створює ілюзію високого рівня безпеки.

Проте ізольованість не гарантує абсолютного захисту. Практика міжнародних і вітчизняних кіберінцидентів свідчить, що навіть ізольовані технологічні системи можуть мати приховані канали зв'язку або бути скомпрометовані - через помилки конфігурації, неконтрольоване використання переносних носіїв, сервісні підключення, бездротові модулі або втручання внутрішніх користувачів. Відповідно до Закону України «Про основні засади забезпечення кібербезпеки України» №2163-VIII від 5 жовтня 2017 року, державні промислові об'єкти, що забезпечують функціонування енергетичних, транспортних, оборонних і виробничих систем, відносяться до об'єктів критичної інформаційної інфраструктури (КІІ). Порушення роботи таких об'єктів може призвести до масштабних наслідків - від зупинки виробництва до виникнення техногенних аварій.

Дані аспекти роблять проблему захисту інформації АСУ ТП особливо складною та актуальною.

Мета: Оцінка актуального стану впровадження захисту інформації на державних промислових об'єктах, аналіз основних проблем та ризиків кібербезпеки в ізольованих автоматизованих системах управління технологічними процесами.

1. Поточний стан впровадження захисту інформації

Більшість державних промислових підприємств сьогодні реалізують заходи із впровадження комплексних систем захисту інформації (КСЗІ) відповідно до вимог Державної служби спеціального зв'язку та захисту інформації (ДССЗЗІ) України [1].

Основні кроки - створення локальних політик інформаційної безпеки, аудит технологічних сегментів, моніторинг доступу до контролерів та операторських станцій, а також фізичну ізоляцію критичних вузлів.

Однак значна частина обладнання функціонує на базі застарілих промислових операційних систем (наприклад, Windows XP Embedded, VxWorks,

QNX), для яких не існують актуальні оновлення безпеки. Додаткову загрозу становить можливість прихованого або несанкціонованого підключення таких систем до зовнішніх мереж - наприклад, через неналежне адміністрування доступу персоналу, використання сервісних ноутбуків або несанкціоноване втручання підрядних організацій.

Практика міжнародних інцидентів, таких як атака Stuxnet [2], доводить, що навіть повністю ізольовані мережі можуть стати об'єктом цілеспрямованого кібервпливу.

Тож навіть у закритих промислових середовищах існує ризик появи «мостів зв'язку», які потенційно можуть бути використані зловмисником для несанкціонованого доступу або проникнення шкідливого програмного забезпечення.

Сьогодні в Україні поступово впроваджується комплексна політика забезпечення кіберзахисту об'єктів критичної інфраструктури відповідно до вимог постанови КМУ № 518 [3].

Проте практична реалізація цих вимог у державних промислових системах часто наражається на нестачу фінансування, відсутність сертифікованих рішень, сумісних із застарілим обладнанням, та недостатній рівень підготовки персоналу у сфері кіберзахисту.

Таким чином, актуальність захисту інформації на державних промислових об'єктах зумовлена поєднанням високої технологічної залежності виробництва, інертності модернізації технічних систем та необхідності виконання сучасних нормативних вимог.

Подальший розвиток системи захисту інформації потребує поєднання заходів нормативного, технічного та організаційного характеру, зокрема розроблення адаптивних механізмів кіберзахисту для ізольованих мереж АСУ ТП без порушення їхньої функціональної стабільності.

2. Проблематика впровадження кіберзахисту в промислових системах

Інертність оновлення виробничих систем. Промислове обладнання має довгий життєвий цикл - 15-30 років, тому його модернізація або заміна є складним і витратним процесом. Це створює розрив між рівнем актуальних кіберзагроз і можливостями реагування.

Обмежена сумісність старих систем із сучасними засобами захисту. Часто контроллери, ПЛК та SCADA-сервери не підтримують нові стандарти безпеки або протоколи шифрування, що унеможливлює пряме впровадження типових IT-рішень.

Людський фактор та внутрішні загрози. Через відсутність централізованих політик безпеки персонал може несвідомо підключати до промислових систем зовнішні пристрої (USB-накопичувачі, модеми тощо), що створює потенційні канали зараження або віддаленого доступу.

Відсутність механізмів моніторингу в реальному часі. У багатьох АСУ ТП відсутня система відстеження аномалій або подій безпеки, що дозволяє виявляти підключення неавторизованих пристроїв.

Нормативна та організаційна фрагментарність. Попри наявність загальних законів і стандартів (наприклад, НП 306.2.237-2022 [4]), практичні методики

побудови кіберзахисту для ізольованих технологічних мереж у державному секторі поки що не мають уніфікованої регламентації, не встановлюють часові вимоги до модернізації обладнання, що не відповідає вимогам з точки зору захисту інформації [5].

Також варто зазначити, що документальна база розглядає захист інформації мережевих систем, обладнання «де-факто» в контексті кіберпростору, що зміщує важливість захисту ізольованих промислових систем на другий план.

Висновок. Промислові системи управління не є гарантовано захищеними лише через свою відокремленість від мереж загального користування. Навпаки, відсутність постійного контролю, обмежені механізми оновлення та людський фактор створюють приховані вектори загроз, які часто складно виявити.

З огляду на інертність модернізації обладнання, державним промисловим об'єктам необхідно реалізовувати поетапну стратегію підвищення кіберстійкості, що містить:

– аудит ізольованих мереж та виявлення потенційних каналів доступу;

– впровадження систем моніторингу (SIEM/IDS) у межах закритого сегменту;

– контроль дій персоналу та політику використання зовнішніх носіїв;

– розробку планів реагування на інциденти згідно з вимогами ISO/IEC 27035:2023.

Комплексний підхід до захисту інформації в ізольованих промислових системах повинен бути складовою державної стратегії з кібербезпеки, відповідно до Стратегії кібербезпеки України (Указ Президента №447/2021). Лише поєднання технічних, організаційних і кадрових заходів дозволить забезпечити належний рівень інформаційної безпеки в умовах сучасних загроз сьогодення.

Перелік використаних джерел.

1. ДССЗЗІ. Нормативні документи ТЗІ. URL: https://cip.gov.ua/ua/news/normativni-dokumenti-sistemi-tzi2024

2. Wikipedia. Stuxnet – malicious computer worm. URL:https://wikipedia.org/wiki/Stuxnet

3. Постанова Кабінету Міністрів України № 518 від 19.06.2019 року «Про затвердження Порядку забезпечення кіберзахисту об'єктів критичної інфраструктури».

4. НП 306.2.237-2022 "Вимоги до кіберзахисту інформаційних та керуючих систем атомних станцій" (2022 р.)

5. О.А. Сиропятов, Л.М. Тимошенко, І. В. Назарова, Н. Г. Козаченко. Експрес-аудит як інструмент оцінки вразливостей в системах обробки даних: підходи, методики та рекомендації. Інформатика та математичні методи в моделюванні. 2024.Том 14, № 4. С.391-404. DOI 10.15276/imms.v14.no4.391

УДК 004.421

Євген СЕГЕДА, Аліна ДАВЛЕТОВА

Західноукраїнський національний університет

КОМБІНОВАНА СИСТЕМА МОНІТОРИНГУ ТА ВИЯВЛЕННЯ MALWARE-ЗАГРОЗ

Вступ. На інформаційну інфраструктуру сучасних організацій постійний вплив мають складні та варіативні кіберзагрози. Традиційні методи захисту, що базуються на сигнатурному аналізі, не завжди забезпечують ефективне виявлення шкідливого програмного забезпечення (ШПЗ), яке може порушувати конфіденційність, цілісність та доступність корпоративних мереж і кінцевих вузлів. Для підвищення точності виявлення загроз необхідно застосовувати комплексні підходи, що інтегрують локальний контроль цілісності файлів, аналіз поведінки процесів та використання зовнішніх аналітичних джерел для верифікації та класифікації підозрілих артефактів.

Управління безпекою інформаційних систем передбачає не лише моніторинг подій, але й автоматизоване реагування на інциденти та мінімізацію ризиків компрометації активів. Для підвищення ефективності управління безпекою необхідна інтеграція локальних і віддалених джерел інформації, застосування алгоритмічних методів класифікації подій та механізмів автоматичного оновлення індикаторів компрометації, що дозволяє швидко і точно реагувати на загрози та підтримувати високий рівень захисту кінцевих вузлів.

Мета дослідження полягає у дослідженні та розробці комбінованої архітектури системи виявлення загроз що інтегрує локальні механізми моніторингу та аналізу кінцевих вузлів з віддаленим аналітичним сервісом для автоматизованого формування та оновлення індикаторів компрометації.

1. Дослідження інструментів виявлення загроз

Системи виявлення ШПЗ потребують використання сучасних інструментів, які забезпечують комплексний аналіз подій безпеки та контроль цілісності систем. Одним із таких інструментів є Wazuh [1]. Це платформа для моніторингу безпеки кінцевих вузлів та аналізу логів, що забезпечує виявлення загроз, контроль цілісності файлів, аудит конфігурацій та реагування на інциденти безпеки.

Платформа поєднує локальні засоби контролю з централізованим аналізом подій, що дозволяє ефективно управляти безпекою великих мережевих інфраструктур. Основні можливості Wazuh:

– File Integrity Monitoring (FIM) – контроль цілісності файлів та системних об'єктів;

– механізми створення правил;

– поведінковий аналіз процесів та активностей користувачів;

– YARA-сканування для виявлення відомих сигнатур Ш ПЗ;

– Active Response – автоматичне реагування на інциденти, напркилад, блокування, карантин, видалення загроз;

– централізований збір і аналіз логів з можливістю інтеграції з SIEM-системами.

Wazuh підтримує інтеграцію з різними операційними системами, Linux, Windows та macOS, та дозволяє налаштовувати правила і політики безпеки відповідно до потреб конкретної організації. Завдяки цьому Wazuh може виконувати багаторівневий аналіз подій безпеки, виявляючи як відомі загрози за сигнатурами, так і аномальні поведінкові патерни. Водночас, її ефективність обмежується здатністю самостійно виявляти нові, ще невідомі загрози, що потребує підключення зовнішніх аналітичних ресурсів.

Такою зовнішньою аналітичною системою є сервіс VirusTotal [2], який надає глобальну репутаційну інформацію про файли, URL-адреси, домени та IP-адреси. Основні можливості:

– аналіз файлів і URL-адрес понад 70 антивірусними движками;
– надання репутаційних даних про файли, домени та IP-Адреси;
– пошук індикаторів компрометації (IoC) для підозрілих об'єктів;
– API для інтеграції з локальними системами безпеки та автоматизації перевірок.

Сервіс має відкритий API, що дозволяє інтегрувати його з локальними системами моніторингу, автоматизуючи процес перевірки підозрілих файлів та об'єктів мережі. Основними обмеженнями VirusTotal є залежність від доступу до Інтернету та обмеження безкоштовної версії щодо кількості запитів, а також обмежена здатність виявляти абсолютно нові загрози, відсутні в базах репутаційних даних.

Інтеграція Wazuh та VirusTotal [3, 4] забезпечує багаторівневу модель виявлення ШПЗ, у якій локальне виявлення та реагування доповнюються глобальним репутаційним аналізом. Такий підхід підвищує точність і швидкість виявлення загроз, зменшує кількість хибнопозитивних спрацювань та створює передумови для автоматизованого доповнення індикаторів компрометації у процесі управління безпекою інформаційної інфраструктури.

2. Архітектура системи виявлення ШПЗ

Поєднання досліджених інструментів дозволить підвищити точність виявлення загроз, скоротити час реагування на інциденти та автоматизувати доповнення індикаторів компрометації, що, у свою чергу, забезпечує більш ефективний захист кінцевих вузлів мережі від відомих і потенційних загроз. Запропонована система складається з таких основних компонентів:

– Wazuh Agent - встановлюється на кінцеві вузли, здійснює моніторинг цілісності файлів, збір логів і виконання реакційних дій;
– Wazuh Manager - централізований сервер кореляції подій, аналізу правил та генерації сповіщень;
– Wazuh Integrator - модуль, що забезпечує обмін даними з зовнішніми API, зокрема з сервісом VirusTotal;
– VirusTotal API - онлайн-сервіс, який агрегує результати перевірки файлів антивірусними рушіями, для класифікації загроз та репутаційного аналізу.

Взаємодія компонентів може бути реалізована за схемою, що наведена на рисунку 1.

Active Response – механізм Wazuh, що автоматично реагує на загрози: карантин, ізоляція або видалення заражених файлів/процесів.

SIEM / SOC – система централізованого збору логів і управління інцидентами безпеки де відбувається логування подій та створення інцидентів для аналітики та реагування.

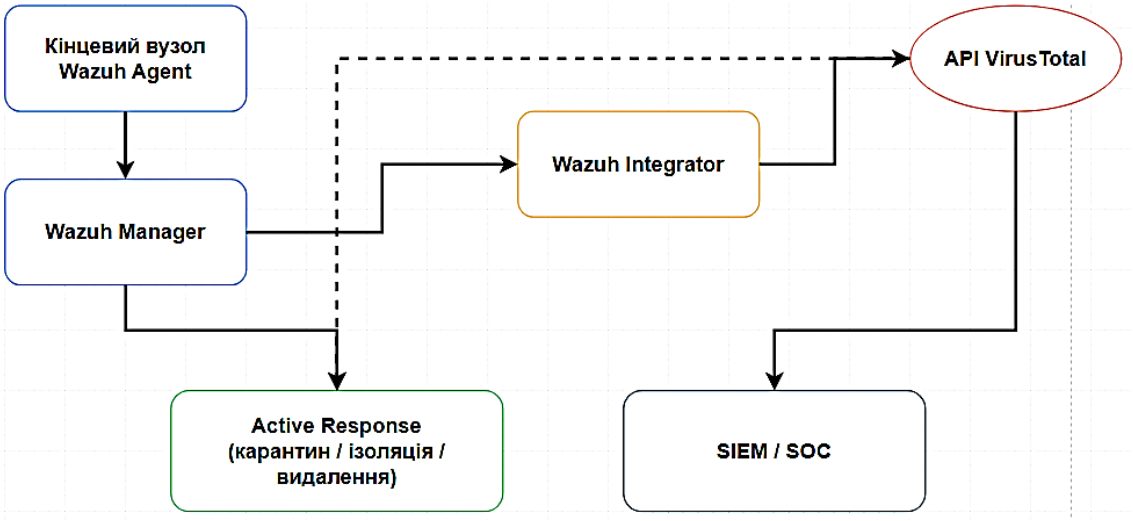


Рисунок 1 - Основні етапи обміну даними між компонентами системи виявлення ШПЗ на основі Wazuh та VirusTotal

На рисунку 1 пунктир позначає логічний зв'язок між результатом аналітики VirusTotal та активацією дій модуля Active Response, який реалізується через Wazuh Manager. Це означає, що інформація з VirusTotal може опосередковано впливати на активні дії, але не є прямою командою на реагування.

3. Алгоритм функціонування системи виявлення шкідливого ПЗ

1. Моніторинг змін. Wazuh Agent здійснює безперервне відстеження змін у системних і користувацьких каталогах. При появі нового або модифікованого файлу обчислюються його контрольні суми (MD5, SHA1, SHA256), які разом із метаданими (шлях, користувач, час, процес-ініціатор) передаються на Wazuh Manager.

2. Локальний аналіз. Змінені файли проходять попередню перевірку за допомогою Wazuh Manager, що містить сигнатури відомих загроз. Паралельно активуються поведінкові правила Wazuh, які виявляють аномальні дії, наприклад, спроби самозапуску, виконання коду з тимчасових каталогів чи звернення до підозрілих мережевих адрес.

3. Інтеграція з VirusTotal. Якщо локальний аналіз визначає файл як потенційно шкідливий, його хеш передається через Wazuh Integrator до VirusTotal API. Отримана відповідь містить оцінку кількості антивірусних рушіїв, що класифікували файл як шкідливий, тип виявленого сімейства та метадані попередніх перевірок.

4. Валідація та кореляція результатів.Wazuh Manager порівнює локальні результати із даними VirusTotal. Якщо кількість підтверджень шкідливості перевищує встановлений поріг (наприклад, 5 або 10 рушіїв), файл класифікується як загроза.

5. Автоматичне реагування (Active Response). Для підтверджених загроз реалізується шляхом виконання попередньо визначених дій, що можуть включати ізоляцію ураженого вузла, усунення шкідливого об'єкта та сповіщення

адміністратора або системи моніторингу.

Для реалізації інтеграції Wazuh із сервісом VirusTotal необхідно налаштувати відповідну секцію в конфігураційному файлі ossec.conf, зокрема вказати назву інтеграції, API-ключ доступу до VirusTotal, ідентифікатор правила Wazuh, при спрацюванні якого дані передаються на перевірку, а також формат повідомлень, наприклад, JSON:

```
<integration>
    <name>virustotal</name>
    <api_key>YOUR_API_KEY</api_key>
    <rule_id>554</rule_id>
    <alert_format>json</alert_format>
</integration>
```

Завдяки такій конфігурації підозрілі файли або їх хеші автоматично відправляються до VirusTotal для отримання репутаційної інформації.

Для мінімізації навантаження на мережу рекомендовано передавати тільки хеші файлів (SHA256 або MD5), а не повні їх копії. Важливим є регулярне оновлювлення правил Wazuh Manager для врахування нових загроз. При обробці результатів перевірки VirusTotal порогове значення підтвердження шкідливості визначається експериментально, щоб балансувати між чутливістю та кількістю хибних спрацювань. При використання публічного API VirusTotal важливо враховувати обмеження швидкості запитів (4 запити/хвилину). Для великих потоків подій рекомендується надсилати тільки критично важливі хеші або використовувати платну версію API з підвищеним лімітом запитів.

Висновок. Впровадження запропонованого підходу дозволить забезпечити підвищення точності виявлення відомого та модифікованого ШПЗ, зниження часу виявлення та реагування, інтеграцію процесів аналізу, підтвердження та нейтралізації в єдиній платформі. Поєднання механізмів Wazuh і VirusTotal формує гнучку, масштабовану та аналітично обґрунтовану систему моніторингу та виявлення ШПЗ, яка здатна підвищити рівень кіберзахисту.

Перелік використаних джерел.

1. Wazuh. Malware detection. [Електронний ресурс].- Режим доступу: https://documentation.wazuh.com/current/getting-started/use-cases/malware-detection.html

2. VirusTotal integration with Wazuh. [Електронний ресурс].- Режим доступу: https://medium.com/%40aravindraja150/virustotal-integration-with-wazuh-c79328d7543f

3. Wazuh. VirusTotal integration. [Електронний ресурс].- Режим доступу: https://documentation.wazuh.com/current/user-manual/capabilities/malware-detection/virus-total-integration.html

4. Detecting and removing malware using VirusTotal integration. [Електронний ресурс].- Режим доступу: https://documentation.wazuh.com/current/proof-of-concept-guide/detect-remove-malware-virustotal.html

УДК 004.056.55

Назаров В.О.

Національний університет «Одеська політехніка»

АВТОМАТИЗОВАНИЙ МЕТОД РИЗИК-ОРІЄНТОВАНОГО ВИЯВЛЕННЯ ПРОБЛЕМНИХ ПРОФІЛІВ У СОЦМЕРЕЖАХ

Вступ. У соціальних мережах системно діють проблемні профілі, що використовуються для розповсюдження спаму, фішингових повідомлень [1] і скоординованих інформаційних операцій з боку країни-агресорки. Їх відстеження ускладнюють короткий життєвий цикл і ротація обліковок, мімікрування під легітимні спільноти, варіативність мовних патернів та обфускація посилань, а також масштаби потоку контенту, які не покриваються ручною модерацією.

Правила й «чорні списки» швидко деградують, знижуючи відтворюваність результатів у часі та між мовами. Доцільним є компактний підхід, що оперує небагатьма, але інформативними ознаками на рівні повідомлення (можлива винагорода, часовий тиск, емоційне забарвлення, наявність зовнішнього посилання) з імовірнісним рішенням на базі сигмоїдної активації [2].

Водночас для каналу електронної пошти існують продуктивні гібридні нейромережеві методи [3]; однак у контексті соціальних мереж із публічними API перевага запропонованого підходу полягає в мілісекундній латентності, інтерпретованості чинників і можливості прямого калібрування та налаштування порогів під канал/мову, що спрощує прозоре агрегування ризику до рівня профілю за часовими вікнами активності в реальному часі.

Мета: Сконструювати та обґрунтувати технічне рішення для автоматизованого пошуку й відсікання проблемних профілів у соціальних мережах на базі чотирифакторної моделі ризику з імовірнісним класифікаційним правилом та агрегуванням ознак на рівні профілю.

Основна частина

Методика ґрунтується на припущенні, що ознаки зловживань у соціальних мережах проявляються спершу на рівні окремих повідомлень (пости, коментарі, приватні звернення), а вже потім - на рівні профілю як сукупності таких дій. Для кожного повідомлення обчислюються чотири інформативні показники: R_m- ступінь обіцянки винагороди, T_m- виразність часових обмежень/терміновості, E_z- інтенсивність емоційного тиску (імперативи, залякування, нав'язливі обіцянки), R_z - наявність зовнішнього посилання. Перші три оцінюються у шкалі від 0 до 10, останній є бінарним (0 - відсутність, 1 - наявність). Таке компактне подання дозволяє застосувати швидку, інтерпретовану модель ризику та підтримувати низьку латентність у великих потоках даних.

Імовірність фішингової/маніпулятивної природи повідомлення визначається логістичною моделлю :

$$p = \sigma(\beta_0 + \beta_1 R_m + \beta_2 T_m + \beta_3 E_z + \beta_4 R_z) \quad (1)$$

де σ(·) - логістична функція, а $\beta_0 \dots \beta_4$ - параметри, отримані під час навчання на розміченому корпусі. Рішення на рівні повідомлення формулюється словами: якщо оцінка ризику p не нижча за обраний поріг, повідомлення вважається

підозрілим; інакше - звичайним. Поріг добирають з урахуванням допустимої частоти хибних спрацьовувань і специфіки каналу/мови; для підвищення стабільності модельні ймовірності калібруються окремо для різних каналів і локалей. Для профілю користувача ризик агрегується в межах рухомого часового вікна W за правилом:

$$S_u(W) = 1 - \prod_{t \in W}(1 - p_t) \quad (2)$$

де p_t - імовірності для окремих повідомлень профілю; величина $S_u(W)$ інтерпретується як інтегральний ризик профілю за період, що зростає як з частотою підозрілих повідомлень, так і з їх індивідуальними оцінками.

Технічна реалізація методу передбачає наскрізний контур «дані → фактори → інференс → агрегування → політика». Наскрізний контур у цій роботі розглядається як окремий програмний продукт, спроєктований для експлуатації в реальному часі у середовищі соціальних мереж. Конкретні технічні засоби для кожного етапу контуру наведено у структурованому вигляді (таблиця 1).

На етапі отримання даних застосовується офіційний інтерфейс платформи (наприклад, публічні ендпоінти Facebook для сторінок і груп) з дотриманням політик доступу, обмежень швидкості та вимог конфіденційності; приватні повідомлення не обробляються.

Текст нормалізується за мовою (токенізація, лематизація, усунення стоп-слів), після чого визначаються R_m, T_m, E_z за контрольованими лексиконами та правилами, а R_z - шляхом аналізу URL (пунікод-нормалізація, виявлення піддоменів, редиректів, параметрів відстеження).

Обчислення p виконується у легкому інференс-сервісі, здатному працювати з мілісекундними затримками; далі формується $S_u(W)$ і приймається рішення на рівні профілю з урахуванням обраного порога для агрегованого ризику. Для підвищення надійності передбачено калібрування ймовірностей, регулярний підбір порогів під цільові метрики (наприклад, утримання FPR у заданих межах), реєстр версій моделі та журналювання рішень для періодичного донавчання.

Адаптивність забезпечується керованим оновленням лексиконів: прикордонні випадки передаються на експертну перевірку (людина-в-контурі), підтверджені нові тригери надходять до кандидатного словника, який після частотних і контекстних фільтрів промотується в робочий. Моніторинг дрифту ознак і якості моделі (розподіли R_m, T_m, E_z, R_z, стабільність калібрування, динаміка FPR/TPR) дозволяє виявляти зміни атакувальних патернів і своєчасно коригувати як словники, так і параметри моделі без втрати відтворюваності.

Таблиця 1 – Технічні засоби та методи реалізації

Збір і підготовка даних		Модель і експлуатація	
Етап	Засіб / метод	Етап	Засіб / метод
1	2	3	4
Доступ до платформи	Публічні API (напр., Graph API), OAuth, квоти	Класифікація повідомлень	Логістична модель (ONNX Runtime), ймовірність у [0;1]
Потік подій	Kafka / RabbitMQ; retries з exponential backoff + jitter	Калібрування ймовірностей	Platt або Isotonic (per-channel / per-locale)

1	2	3	4
Визначення мови	fastText LID, langdetect	Підбір порогів повідомлення/профілю	ROC/PR-аналіз; line/grid search під цільовий FPR
Нормалізація тексту	spaCy / Stanza; токени, леми, стоп-слова	Агрегування ризику профілю	Інтегральний ризик у рухомому вікні; згладжування
Оцінка R_m, T_m, E_z	Керовані лексикони та правила (шкала 0–10)	Метрики якості та затримок	AUC, FPR/FNR, Recall@FPR≤x, latency p95
Виявлення R_z (посилання)	idna/punycode, tldextract, редиректи, URL-патерни	Моніторинг дрифту	PSI/JS для факторів ризику; алерти
Безпека даних	TLS/mTLS, hashing PII, шифрування на диску	Деплой та масштабування	Docker, Kubernetes; канарейкові релізи, rollback
Human-in-the-Loop	Адмін-UI, Label Studio; модерація «прикордонних» кейсів	Інтеграції / реагування	Webhooks; реєстр профілів із ризиком ≥ порога; маршрути модерації

Висновок. Подано автоматизований ризик-орієнтований підхід до виявлення проблемних профілів у соціальних мережах, що поєднує чотирифакторну оцінку на рівні повідомлень з агрегуванням ризику у часовому вікні профілю та реалізується як завершений програмний продукт за контуром «дані → фактори → інференс → агрегування → політика». Підхід вирізняється компактністю, інтерпретованістю й низькою латентністю, забезпечує калібрування під канали та мови і керованість хибних спрацьовувань через порогові налаштування. Практичну придатність і засоби розгортання систематизовано у структурі технічних засобів та процедур (таблиця 1), що гарантує інтеграцію з офіційними API, журналювання та контроль якості в експлуатації.

Перелік використаних джерел.

1. Штонда Р., Черниш Ю., Терещенко Т., Терещенко К., Цикало Ю., Поліщук С. Класифікація та методи виявлення фішингових атак. Кібербезпека: освіта, наука, техніка. 2024. Т. 4, № 24. С. 69–80. DOI: 10.28925/2663-4023.2024.24.6980.

2. Назаров В. О., Садченко А. В., Кушніренко О. А. Алгоритм виявлення фішингу в листах месенджерів та електронної пошти із використанням нейронних мереж з фіксованою кількістю ранжованих чинників ризику. Інформатика та математичні методи в моделюванні. 2025. Т. 15, № 2. С. 247–259.

3. Фещенко Є. О., Заболотня Т. М. Метод автоматизованого виявлення фішингу в електронних листах на основі гібридної нейромережевої архітектури. Наукові праці Вінницького національного технічного університету. 2025. № 2. С. 145–154. DOI: 10.31649/2307-5376-2025-2-145-154.

УДК 004.056.55

Драгін Д., Садченко А.

Національний університет «Одеська політехніка»

РОЗРОБКА ЛОКАЛЬНОЇ МОДЕЛІ МАШИННОГО НАВЧАННЯ ЩОДО ЗАХИСТУ КОНФІДЕНЦІЙНОЇ ІНФОРМАЦІЇ У ВІДКРИТОМУ ПРОГРАМНОМУ КОДІ

Вступ. В сучасному цифровізованому світі, ми кожного дня використовуємо велику кількість різних застосунків, які використовують, зберігають або передають конфіденційну інформацію. Безпека цих даних з кожним роком стає все більш важливим питанням і розглядається не тільки, як проблема особистої безпеки або компанії, а інколи може досягати загальносвітового рівня.

Загалом, добре побудова інформаційна система в технічному плані немає слабких місць, які зловмисник міг би використати для незаконного доступу до конфіденційної інформації. Проте під час створення застосунку, розробники можуть залишити у відкритому доступі конфіденційну інформацію (API ключі, токени доступу, паролі до баз даних), що робить їх вразливими до автоматизованих сканерів, які постійно переглядають платформи розробки, такі як GitHub, GitLab, npm або PyPi, у пошуках витоків, для несанкціонованого доступу до персональної інформації користувача або злому системи.

Мета: розробка локальної моделі машинного навчання для захисту конфіденційної інформації у відкритому програмному коді для забезпечення превентивного захисту від витоку секретної інформації.

Основна частина

Згідно зі звітом GitGuardian за 2023 рік, у публічних репозиторіях було виявлено 12 778 599 нових секретів, з яких 3 698 686 є унікальними, і ця кількість з кожним роком лише зростає. Для порівняння у 2021 році - 6 мільйонів, а у 2022 році - 10 мільйонів, що на 28% менше, ніж у 2023 році [1].

Згідно зі звітом Агентства з кібербезпеки та інфраструктури безпеки США (CISA) за 2022 рік, у більш ніж 54% випадків незаконного доступу до систем було отримано через компрометацію конфіденційної інформації, в той час як експлуатація вразливостей становила лише 1% [2].

Для вирішення цієї проблеми було створено багато рішень, але більшість ініціатив в цій проблемі фокусуються вже на аналізі репозиторіїв після завантаження коду, що в деяких випадках може бути запізно. Про що свідчіть дослідження Cybenari, компанії що спеціалізується на кібербезпеці та тестуванні ПО, для виявлення та використання чутливої інформації на GitHub треба лише 127 секунд. Набагато гірші результати має менеджер пакетів і репозиторій для JavaScript і Node.js – npm, для якого зловмисникам треба лише 60 секунд.

Також важливо пам'ятати, що npm і PyPi не надають функцію автоматичного сканування коду та оповіщення та знайдену чутливу інформацію в коді, що робить захист від витоку інформації на цих платформах майже неможливим [3].

Для побудови цієї моделі використовується логістична регресія, яка була спеціально розроблена для задач класифікації, що дозволяє їй ефективно розподіляти об'єкти між класами на основі ймовірнісного підходу.

Для визначення моделі логістичної регресії, вводиться певна випадкова величина Y, що набуває значення від 0 до 1. Найчастіше 0 відповідає за те, що певний об'єкт не відповідає певному класу, а 1 – навпаки. Об'єкт класифікується шляхом порівняння отриманої ймовірності з пороговим значенням. Логістична регресія дозволяє оцінити вплив змінних на ймовірність належності до певного класу. Результатом є ймовірності для кожного класу, що допомагають приймати рішення про класифікацію [3].

Ця величина залежить від певної множини змінних, які впливають на те, яке значення буде приймати змінна Y.

$$x = (1, x_1, \dots, x_{n)^T}, \tag{1}$$

Для того, щоб отримати залежність змінної Y від вектору пояснювальних змінних, вводиться додаткова прихована змінна y*, яка відповідає за лінійний результат.

$$y^* = \theta^T x = \theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n + \varepsilon, \tag{2}$$

Ця зміна є лінійною комбінацією параметрів θ, що визначають вплив кожної ознаки x, до яких додається випадкова похибка ε, що зазвичай є підпорядкованою логістичному розподілу та є випадковою величиною з певним логістичним розподілом ймовірностей.

Тому залежність Y від y^* має вигляд:

$$Y = \begin{cases} 0, & y^* \le 0 \\ 1, & y^* > 0 \end{cases}, \tag{3}$$

Для перетворення лінійного результату у вірогідність використовується сигмоїдна функція. Сигмоїдна функція є математичною функцією, яка має S-подібну (sigmoid) форму. Її основна мета - перетворення будь-якого дійсного числа в значення в інтервалі від 0 до 1 [4].

$$\sigma(y^*) = \frac{1}{1+e^{-y^*}} \tag{4}$$

Додатково ефективність системи було підвищено за рахунок гібридних підходів, що поєднують методи обробки тексту (TF-IDF, регулярні вирази) з алгоритмами машинного навчання.

Під час тестування системи було виявлено, що для сканування 38000 файлів(розмір проєкту сягає 263 МБ та містить 30 конфіденційних токенів) займає приблизно 64 секунди, в залежності від розмірів файлів, а також кількості секретної інформації в них, що є кращим результатом ніж той, що надає GitHub. Повідомлення про наявність секретної інформації в коді надійшла приблизно через 43 секунди після завантаження.

Ще одним недоліком GitHub є те, що він не проводить сканування закритих репозиторіїв, а лише відкритих, що дає можливість зловмисникам отримати секретну інформацію до того, як буде отримано повідомлення про неї.

Також GitHub не проводить евристичний аналіз, через що не було знайдено всі паролі в програмному коді, які в свою чергу знайшла локальна модель.

Після сканування директорії було визначено, що моделі потрібно 1.5 мс для сканування 1 файлу. Модель відмітила 42 токени, які потенційно можуть містити конфіденційну інформацію, з яких 25 токенів дійсно є такою інформацією, проте 17 токенів були відмічені хибно, шо свідчить про те, що локальна модель демонструю певні ознаки надмірності в свої рішеннях, коли рядки які не є конфіденційною інформацію вона помічає такими, що потенційно можуть бути такими.

Вирішенням цієї проблеми на перших етапах є збільшення навчальної вибірки кількість даних для навчання моделі, а також додавання нових функцій обробки тексту та модернізація вже наявних методів. Проте вже отримані результати свідчать про певний успіх в розробці моделі.

Висновок. Таким чином, можна зробити висновок про те, що в сучасних системах увагу треба приділяти не тільки класичним методам захисту інформації, а також шляхам вирішення проблеми людського фактору. Дана модель забезпечує захист та аналіз відкритого програмного коду в умовах обмежених ресурсів, а також з забезпечення потрібної швидкості обробки та надійності, про що свідчать результати тестування моделі, за допомогою поєднання різних методів обробки тексту з алгоритмами машинного навчання.

Перелік використаних джерел.

1. The State of Secrets Sprawl Report. GitGuardian. 2024.

2. Active Adversary for Tech Leaders. Sophos News. 2022.

3. You Have One Minute to Save Your Leaked AWS Credentials ThreatDown Blog. 2023.

4. Hastie T., Tibshirani R., Friedman J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. – Springer, 2001. – 533 p.

УДК 004.056.5

Дмитро ПІДЛИСЬКИЙ

Західноукраїнський національний університет

ДОСЛІДЖЕННЯ МОЖЛИВОСТЕЙ ВИКОРИСТАННЯ ПЛАГІНУ KIBANA ДЛЯ РОЗВІДКИ КІБЕРЗАГРОЗ

Вступ. У сучасному інформаційному середовищі об'єктів корпоративної та державної безпеки важливість систем раннього виявлення й реагування на кіберзагрози зростає. Підходи типу розвідки загроз (Threat Intelligence) дедалі більше інтегруються у рішення з моніторингу, аналізу та візуалізації даних. У цьому контексті платформа Kibana, як компонент ELK-стеку набуває особливого значення завдяки своїм можливостям інтеграції даних, візуалізації та побудови дашбордів [1-4].

Метою є: аналіз плагіну Kibana, що дозволяє розширити її функціонал для безпекового моніторингу та розвідки загроз та оцінка практичних можливостей побудови платформи розвідки загроз на основі цього плагіну, включно з інтеграцією індикаторів, створенням правил виявлення та побудовою аналітики загроз.

1. Аналіз плагіну Kibana

Плагін Kibana створює доповнення до основного інструментарію візуалізації даних, дозволяючи розширити можливості у сфері безпеки та розвідки загроз. Він формує додатковий функціонал, наприклад: нові аплікації, модулі візуалізації, інтеграцію з даними індикаторів загроз, спеціалізовані дашборди та правила.

Згідно з офіційною документацією, Kibana [1] підтримує встановлення плагінів, що додають власну функціональність. Наприклад, існує пакет «Threat Intelligence Utilities», який включає дашборд для огляду даних зі всіх підключених TI-джерел (рисунок 1) [2].

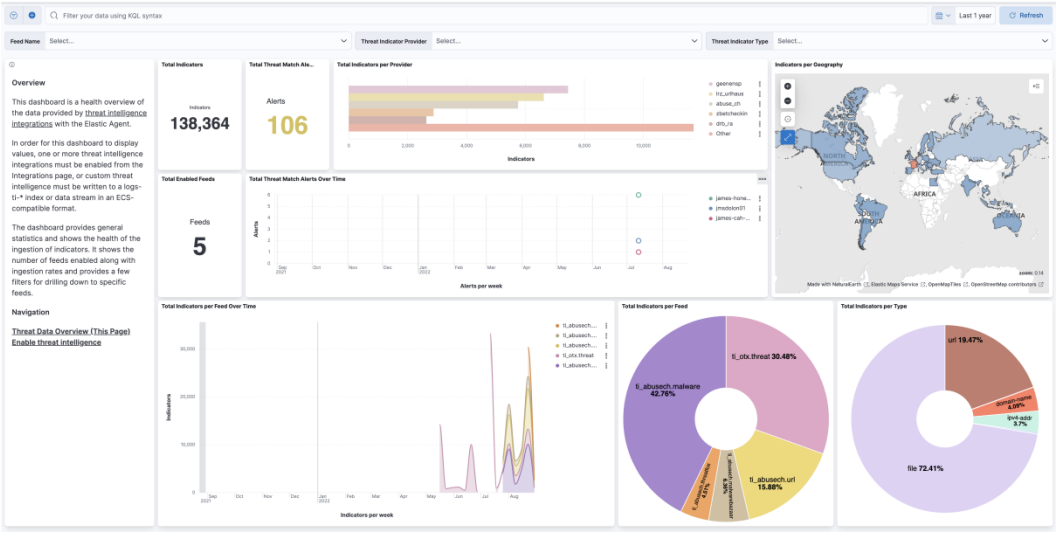


Рисунок 1 – Приклад відображення даних

Ключові можливості включають:

– підключення модулю TI (Threat Intelligence) через агент Elastic Agent або

Filebeat для збору індикаторів (IP, домени, хеші) та їх індексації.

– створення дашбордів у Kibana для візуалізації, фільтрації та кореляції даних індикаторів та подій.

– використання правил «Indicator Match», коли індикатор TI збігається з логом в середовищі, що дозволяє генерувати попередження. leveleffect.

Проведений аналіз дозволяє визначити ряд переваг застосування плагіну. Основними з них є можливість інтеграції з екосистемою Elastic, зокрема використовуючи Elasticsearch, Logstash і Kibana, організація може об'єднати збір логів, індексацію, пошук і візуалізацію в єдиний процес. Також варто зазначити гнучкість візуалізації, оскільки Kibana дозволяє створювати власні дашборди, фільтри, використовувати Lens, ES|QL запити для глибшого аналізу.

Плагін забезпечує реальне виявлення індикаторів, зокрема модуль TI дозволяє зв'язати дані індикаторів із подіями в логах, що підвищує ефективність виявлення загроз. Важливим аспектом є його масштабованість, оскільки рішення на базі Elasticsearch сімейства дозволяє обробляти великі обсяги даних з високою швидкістю індексації та пошуку.

Проте існують певні обмеження та виклики, зокрема офіційна документація вказує, що інтерфейси змінюються, і немає гарантії сумісності з новими версіями Elastic. Необхідність спеціалізованого налаштування, зокрема підключення модулю TI, налаштування індексів, правил, API-ключів тощо потребує технічної експертизи. Недоліки стосуються також продуктивності та ресурсів, оскільки активація великої кількості правил виявлення індикаторів може значно підвищити навантаження на стек. На застосування також може впливати оОбмеженість готових дашбордів, оскільки у деяких випадках потрібно створювати власні візуалізації, оскільки стандартні рішення можуть бути надто загальними.

Плагін Kibana є потужним інструментом для розширення функціональності платформи до задач безпеки і розвідки загроз. Завдяки інтеграції з Elastic Stack він дає змогу об'єднати збір, обробку, індексацію та візуалізацію даних індикаторів загроз. Проте впровадження вимагає ретельного планування, налаштування ресурсів та уваги до сумісності. Як основа платформи розвідки загроз – він має перспективу, але потребує доповнення відповідними процесами та даними.

2. Розвідка загроз на основі плагіну Kibana

Реалізація платформи розвідки загроз з використанням Kibana передбачає використаня типової архітектури (рисунок 2), що може включати наступні компоненти[1-3]:

– джерела індикаторів: відкриті TI-фіди (наприклад, AlienVault OTX, MalwareBazaar), внутрішні дані (логи, DNS-запити, мережевий трафік);

– агент або модуль збору (Filebeat, Elastic Agent) з увімкненим модулем Threat Intel;

– індексація даних у Elasticsearch: створення індексів для індикаторів (наприклад, logs-ti*) та для логів подій (logs-*);

– інтерфейс Kibana з дашбордами, фільтраціями, інструментами аналітики;

– правила виявлення (Indicator Match) - налаштування автоматичного порівняння логів і індикаторів, що дозволяє генерувати попередження.

– процес реагування – при генерації попередження , аналітик переглядає дашборд, проводить аналіз, ініціює розслідування чи заходи.

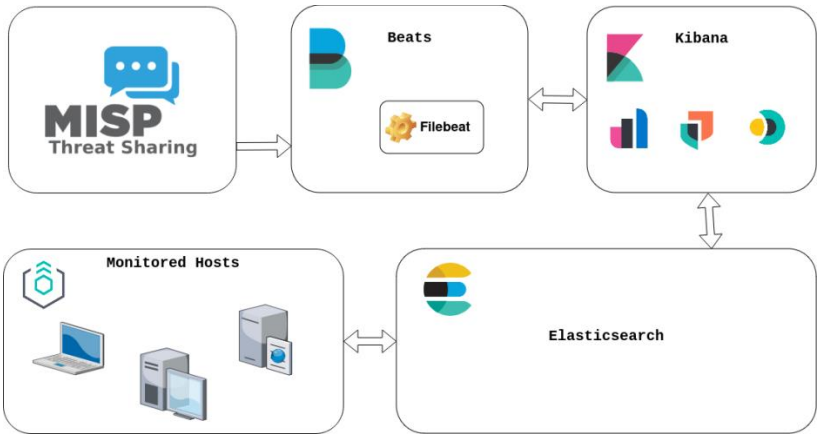


Рисунок 2 - Архітектура платформи розвідки загроз

Практичні можливості застосування використання можуть включати наступні сценарії.

– Моніторинг індикаторів у реальному часі. Плагін дозволяє візуалізувати кількість нових індикаторів, їх розподіл за типами (IP, домен, хеш), а також співставити з подіями в мережі чи кінцевих точках. Наприклад, можна побудувати дашборд із часовою серією підвищеної активності за індикаторами.

– Кореляція індикаторів з логами. Використовуючи індекси, можна створити правило, що спрацьовує коли destination.ip у логу співпадає з threatintel.indicator.ip. Це дозволяє швидко визначати події, в яких ваші системи контактували з відомим шкідливим ресурсом.

– Аналітика тенденцій загроз. Дашборд може показати, як змінюється обсяг нових індикаторів, як часто відбуваються спрацьовування, які типи індикаторів переважають - це допомагає команді безпеки прогнозувати активність атак.

– Звітність та візуалізація для керівництва. Системи, створені на базі Kibana, можуть перетворювати технічні дані на зрозумілий дашборд для менеджменту, наприклад показуючи кількість інцидентів, ступінь ризику, тенденції.

3. Рекомендації щодо практичної реалізації платформи виявлення загроз

Для реалізації платформи розвідки загроз на основі плагіну Kibana необхідно встановити модуль Threat Intel та перевірити, що дані індикаторів коректно індексуються у відповідний індекс.

Налаштувати правила та фільтри, які мінімізують кількість хибнопозитивних спрацьовувань, наприклад, відфільтрувати внутрішні IP-адреси чи низькоризикові індикатори.

Важливим аспектом є планування ресурсів, оскільки великі обсяги даних індикаторів та логів могуть потребувати масштабування Elasticsearch і оптимізації запитів. Необхідно підготувати шаблони дашбордів, зокрема одразу створити набір базових візуалізацій, наприклад кількість індикаторів за типами, співставлення з подіями, карта геолокацій тощо.

Наступним кроком є вровадження процес реагування: алерт -> аналіз -> реакція (рисунок 3).

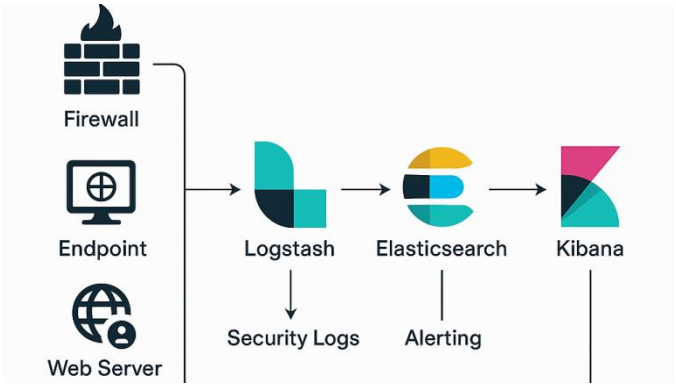


Рисунок 3 – Реагування на загрози

Дашборд Kibana повинен бути інтегрований в процес реагування на загрози. Тому потрібно періодично оцінювати ефективність, зокрема визначити як часто алерти на основі індикаторів перетворюються на реальні інциденти Це допоможе скоригувати правила та пріоритизацію.

При реалізації платформи виявленя загроз важливо враховувати певні ризики та функціональні обмеження, зокрема індикатори TI не гарантують, що подія є атакою, тому потрібно розглядати її у контексті. Також великі обсяги індикаторів можуть створювати «шум» й приводити до перевантаження аналітиків. У випадках, коли система не налаштована належним чином, можливі затримки індексації або пропущені події. Надмірне фокусування лише на індикаторах загроз може ігнорувати поведінкові або аналітичні детекції, тому рішення має бути мультиаспектним.

Висновок. Побудова платформи розвідки загроз на базі плагіну Kibana є актуальною задачею, яка дозволяє об'єднати індикатори загроз, логи подій і візуалізацію в одному середовищі. При правильній архітектурі, налаштуванні та процесах така платформа може значно підвищити видимість загроз і скоротити час реагування. Водночас вона потребує фокусування не лише на технологіях, але й на процесах і людському факторі.

Перелік використаних джерел.

1. Kibana plugins. [Електронний ресурс].- Режим доступу: https://www.elastic.co/docs/reference/kibana/kibana-plugins

2. Threat Intelligence Utilities. [Електронний ресурс].- Режим доступу: https://www.elastic.co/docs/reference/integrations/ti_util

3. Building Effective Dashboards for Threat Intelligence with Kibana and Grafana. [Електронний ресурс].- Режим доступу: https://thinkcloudly.com/blog/building-effective-dashboards-for-threat-intelligence-with-kibana-and-grafana

4. Home Lab: Enabling and Configuring Threat Intelligence and Detections. [Електронний ресурс].- Режим доступу: https://www.leveleffect.com/blog/home-lab-enabling-and-configuring-threat-intelligence-and-detections

УДК 004.056

Котляров А.В., Кушніренко Н.І.

Національний університет «Одеська політехніка»

АЛГОРИТМ ПОШУКУ ПРОФІЛІВ КОРИСТУВАЧІВ ЗА НІКНЕЙМОМ У СОЦІАЛЬНИХ МЕДІА ЯК ЕЛЕМЕНТ ПРОТИДІЇ КІБЕРЗЛОЧИННОСТІ

Вступ. У сучасному цифровому світі соціальні медіа стали важливою частиною повсякденного життя мільярдів людей, забезпечуючи швидку комунікацію та обмін інформацією. Водночас їх широке використання поєднується з високим рівнем анонімності, що створює умови для здійснення кіберзлочинів. В боротьбі з такими загрозами ключову роль почав відігравати OSINT - збір та аналіз даних із відкритих джерел для використання як розвідданих або доказів. Це зумовлює потребу у розробці спеціалізованих алгоритмів і застосунків для пошуку користувачів за цифровими слідами.

Мета: Розробка алгоритму автоматизованого пошуку профілів користувачів за нікнеймом у соціальних медіа, що сприяє протидії кіберзлочинності.

Основна частина

Для ефективного пошуку користувачів у соціальних медіа необхідно мати вихідні дані, що слугують відправною точкою. Такі дані називають ідентифікаторами - унікальними або частково унікальними відомостями, які допомагають звузити пошук або однозначно ідентифікувати особу. У OSINT-розвідці по соцмедіа зазвичай виділяють п'ять основних ідентифікаторів: номер телефону, електронну адресу, прізвище та ім'я, нікнейм [1].

Кожен із цих ідентифікаторів має свої обмеження. Телефони та електронні адреси унікальні, але зазвичай приховані й рідко доступні для відкритого пошуку. Прізвище та ім'я часто поширені та мають багато варіантів написання, що ускладнює точну ідентифікацію. Натомість найбільш універсальним і зручним для пошуку є нікнейм, оскільки він поєднує відкритість та певний рівень унікальності

Нікнейм (від англ. nickname - прізвисько) - це вигадане ім'я, яке користувач обирає для використання у соціальних медіа. Воно може базуватися на власному імені чи прізвищі, надихатися вигаданими персонажами або відомими особистостями, а також мати інше значення, пов'язане з власником. У соцмедіа нікнейм виконує роль унікального ідентифікатора, замінюючи числовий ID.

Здійснити ручний пошук можна двома способами: перший - ввести запит у пошукове поле будь-якої соціальної медіа; другий (лише для веб-версій) - змінити в адресному рядку значення нікнейму на потрібне. Наприклад, у URL https://twitter.com/user1 частина «user1» є нікнеймом, і її можна замінити на будь-який інший для пошуку потрібного. Зрозуміло, що ручний пошук займає багато часу і є незручним, адже доводиться перевіряти кожну соцмедіа окремо, а їхня велика кількість значно ускладнює процес. Тому для спрощення було розроблено алгоритм, який автоматизує процес пошуку.

Алгоритм працює за такою послідовністю кроків.

Крок 1. Введення даних. Користувач задає нікнейм (наприклад, user2).

Крок 2. Формування посилання. З бази шаблонів URL вибирається перше

(наприклад, https://instagram.com/{}) і замість {} підставляється введений нікнейм - утворюється повна адреса, наприклад https://instagram.com/user2.

Крок 3. Перевірка доступності профілю. Виконується запит за сформованим URL, після чого аналізується отриманий HTTP-статус.

Крок 4. Обробка відповіді сервера. Якщо код належить до груп 3xx або 4xx - профілю із таким нікнеймом не існує; якщо код 2xx - додатково аналізується HTML-код сторінки.

Крок 5. Аналіз вмісту сторінки. Якщо у HTML немає фраз, що вказують на помилку (наприклад, "User not found"), URL вважається успішним; у протилежному випадку - невдалим.

Крок 6. Перехід до наступного шаблону. Якщо всі шаблони ще не перевірені, алгоритм повторюється для наступного; якщо перевірено останній - формуються та виводяться результати перевірки.

Для програмної реалізації цього алгоритму використовуються бібліотеки Requests для отримання HTTP-запитів та BeautifulSoup для аналізу HTML-коду сторінки. Замість Requests також можна використовувати Selenium, і цей вибір залежить від можливостей користувача. Головні проблеми Selenium - висока ресурсозатратність та повільність, проте він має суттєві переваги: імітує дії людини, тому CAPTCHA не розпізнають його як бота, і дозволяє отримувати доступ до соцмедіа з обов'язковою авторизацією [2].

Слід зазначити, що подібний, хоча й спрощений, алгоритм уже реалізовано в деяких застосунках. Найчастіше вони не виконують перевірку фраз-помилок. Це, звісно, прискорює роботу алгоритму, проте ігнорує так звані soft 404 (м'яка помилка 404) [3] - випадки, коли не повертаються коди 4xx навіть при відсутності користувача. Замість цього може повернутися код 200 з повідомленням про помилку або відбутися перенаправлення на головну сторінку. У результаті алгоритм може помилково визначити, що профіль існує, хоча насправді його немає. Через це такі застосунки зазвичай уникають перевірки тих соціальних медіа, де спостерігається подібна поведінка.

Висновок. Розроблений алгоритм забезпечує автоматизоване визначення наявності профілів користувачів за нікнеймом у соціальних медіа, що значно спрощує процес пошуку у порівнянні з ручними методами. У ході було продемонстровано його структуру, описано логіку функціонування та запропоновано способи програмної реалізації. Його застосування може стати базою для створення програмних засобів OSINT, підвищуючи ефективність аналізу цифрових слідів та підтримуючи заходи протидії кіберзлочинності.

Перелік використаних джерел.

1. Walkow M., Pohn D. Systematically Searching for Identity-Related Information in the Internet with OSINT Tools, 2024. 8 p. DOI: https://doi.org/10.48550/arXiv.2407.16251.

2. Python Selenium vs Python Requests. [Електронний ресурс]. - Режим доступу: https://scrapeops.io/python-web-scraping-playbook/python-selenium-vs-python-requests/.

3. Prieto, V.M., Alvarez, M., Cacheda, F. Soft-404 Pages, A Crawling Problem. 2014 J. Digit. Inf. Manag., 12, 73-92.

УДК 004.056.5:614

Мельник М.О., Величканич Ю.Ю., Назарова І.М.

[1]Національний університет «Одеська політехніка»

МЕТОДИКИ ОЦІНКИ РИЗИКІВ СОЦІАЛЬНОЇ ІНЖЕНЕРІЇ У МЕДИЦИНІ

Вступ. У сучасному цифровому середовищі система охорони здоров'я є однією з найуразливіших до кібератак галузей. Це зумовлено значним обсягом оброблюваних персональних і медичних даних, використанням розгалужених інформаційних систем, а також наявністю численного персоналу з різним рівнем цифрової компетентності. В останні роки спостерігається зростання кількості інцидентів, пов'язаних не з технічними вразливостями, а з людським фактором.

Соціальна інженерія (СІ) у сфері медицини - це сукупність методів психологічного впливу, які використовуються зловмисниками для отримання несанкціонованого доступу до даних або інформаційних ресурсів шляхом маніпуляції персоналом. До типових прикладів належать фішингові електронні листи, телефонні дзвінки від «постачальників» або «державних інспекторів», запити від імені адміністрації, підроблені вебпортали систем охорони здоров'я тощо. У результаті таких атак можуть бути порушені конфіденційність, цілісність і доступність медичної інформації, що створює не лише технічні, але й етичні та юридичні наслідки. Саме тому оцінка ризиків СІ є ключовим компонентом системи кіберзахисту медичних закладів

Мета: теоретичний аналіз існуючих методик оцінки ризиків СІ у сфері охорони здоров'я, виявлення їхніх переваг і недоліків, а також формування рекомендацій щодо впровадження комплексного підходу до управління цими ризиками у медичних організаціях.

1. Огляд існуючих рішень для оцінки та проведення тестів соціальної інженерії у медичній сфері

За даними звітів відкритих джерел IBM Security (2024) та Verizon Data Breach Investigations Report (2025), понад 80 % кібератак у сфері охорони здоров'я мають елемент СІ. Основними цілями зловмисників є отримання доступу до електронних медичних карток, облікових записів систем eHealth, лабораторних систем або фінансових даних. Найчастіше атаки здійснюються через фішингові повідомлення, телефонні дзвінки з проханням підтвердити логін або пароль, маніпуляції у месенджерах під виглядом екстрених запитів. Особливістю СІ є те, що основним вектором атаки виступає людський фактор, тому стандартні математичні або технічні моделі оцінки ризиків не є достатньо ефективними без урахування поведінкових характеристик персоналу.

За підходом до аналізу можна виокремити три групи методів:

1. Якісні методи – базуються на експертному аналізі, анкетуванні, самооцінюванні рівня обізнаності співробітників.

2. Кількісні методи – передбачають використання математичних моделей для оцінки імовірності атаки та потенційних втрат.

3. Комбіновані методи – поєднують поведінковий аналіз із кількісною оцінкою.

Далі представимо існуючі методичні підходи до оцінки ризиків у медичних закладах які складаються з наступних етапів:

– ідентифікації ризиків
– аналізу ризиків
– оцінювання рівня ризику
– розробки заходів по зниженню ризиків

Графіки загальної динаміка зростання фішингових/соціоінженерних атак у секторі медицини 2020-2024 рр на рисунку 1

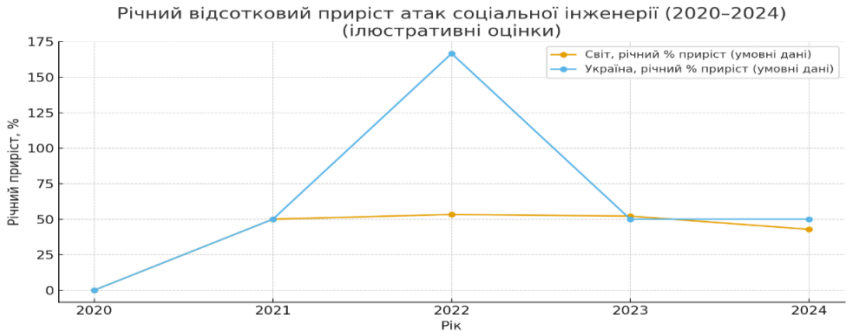


Рисунок 1 - Загальна динаміка зростання фішингових/соціоінженерних

2. Рекомендації щодо вдосконалення оцінки ризиків CI у медичній сфері

З урахуваннням статистики та ризиків від CI в медицині рекомендації щодо вдосконалення оцінки ризиків, а саме :

– Інтеграція психологічних індикаторів.
– Регулярне проведення симульованих атак (Phishing Simulation Program) для перевірки готовності персоналу.
– Створення єдиної бази інцидентів CI, яка дозволяє аналізувати типові сценарії атак.
– Використання систем штучного інтелекту для аналізу листування та виявлення ознак маніпуляцій у реальному часі.
– Розробка національних рекомендацій з урахуванням специфіки медичної сфери України, аналогічних до NIST або NHS Digital Security Guidelines.

Висновок. Методики оцінки ризиків CI у медицині, повинні враховувати не тільки технічні аспекти, та і психологічну складову, яка є визначальною у таких атаках. Вважаємо, що ефективна система управління ризиками CI у медичних закладах повинна поєднувати наступне: стандартизовані підходи оцінки, поведінковий аналіз персоналу, постійне навчання і контроль через симуляції, моніторинг індикаторів ризику у динаміці. Застосування інтегрованої методики дозволить не лише зменшити кількість інцидентів, але й підвищити рівень кібергігієни медичного персоналу, сформувати культуру безпечної поведінки та забезпечити стійкість інформаційної інфраструктури медичних установ.

Перелік використаних джерел.

1.Венгерський П.С., Вишневська Н.С., Хохлачова Ю.Є., Хорошко В.О., Чобаль О.І., Кількісна оцінка кіберзахищеності інформації. Захист інформації. – 2023. – Т. 25, №2. – С. 53-61.

2. S.Yevseev, S.Pogasiv O.Shmatko, M. Melnyk Cybersecurity:security of linux operating system / Laboratory workshop, Kharkov, 2021

УДК 004.056

Вадим ХМЕЛИК, Ренат ДАВЛЕТОВ

[1]Західноукраїнський національний університет

ДОСЛІДЖЕННЯ ПОБУДОВИ ОПЕРАЦІЙНОГО ЦЕНТРУ БЕЗПЕКИ

Вступ. Сучасні тенденції цифрової трансформації супроводжуються зростанням кількості та складності кіберзагроз, що вимагає від організацій переходу до проактивного моніторингу подій безпеки. Операційний центр безпеки (Security Operations Center, SOC) є ключовим елементом такої стратегії, забезпечуючи безперервне виявлення, аналіз і реагування на інциденти в реальному часі.

Використання інструментів із відкритим програмним кодом, зокрема Wazuh, Suricata, Zeek та MISP, дає змогу створювати ефективні та гнучкі SOC-рішення з мінімальними витратами. Дослідження побудови SOC на основі open-source технологій спрямоване на підвищення рівня кіберстійкості організацій та оптимізацію процесів управління інформаційною безпекою.

Мета дослідження полягає у аналізі принципів побудови SOC, визначенні його архітектури, основних компонентів та функціональних можливостей, а також у розробці підходів до впровадження SOC з використанням інструментів з відкритим програмним кодом.

1. Призначення операційного центру безпеки

SOC - це організаційна та технологічна структура, призначена для централізованого моніторингу, виявлення, аналізу та реагування на інциденти інформаційної безпеки в реальному часі [1, 2]. Основною метою SOC є забезпечення безперервного контролю за станом інформаційної інфраструктури, зменшення часу між виявленням загрози та реагуванням (показники Mean Time to Detect - MTTD та Mean Time to Respond - MTTR), а також підвищення рівня захищеності організації.

SOC функціонує як ядро системи управління інформаційною безпекою (ISMS), забезпечуючи взаємодію між технічними засобами захисту, аналітичними інструментами, базами знань про загрози (Threat Intelligence), а також персоналом, який здійснює моніторинг і реагування. Завдяки інтеграції різнорідних джерел даних - систем виявлення вторгнень (IDS/IPS), антивірусів, файрволів, систем контролю доступу, серверів логів тощо - SOC дозволяє створити єдину картину безпеки інформаційного середовища [3].

Основними завданнями SOC є:

– Моніторинг подій безпеки - збір, кореляція та аналіз журналів подій з усіх компонентів IT-інфраструктури за допомогою систем управління подіями безпеки (SIEM).

– Виявлення та класифікація інцидентів - ідентифікація аномалій, відхилень від базової поведінки або збігів із відомими сигнатурами атак.

– Оцінювання ризиків і пріоритезація - визначення критичності інцидентів відповідно до впливу на активи організації.

– Реагування на інциденти - ініціювання відповідних дій (ізоляція вузлів, блокування облікових записів, активація плейбуків реагування).

– Аналіз інцидентів - дослідження цифрових слідів для встановлення джерела атаки, методів проникнення та наслідків.

– Звітність і аудит - формування аналітичних звітів для оцінювання ефективності заходів безпеки та дотримання політик.

Залежно від масштабу та потреб організації SOC може бути реалізований у декількох формах [4]:

– Внутрішній SOC (Internal SOC) - функціонує в межах організації, забезпечуючи повний контроль над процесами безпеки.

– Керований SOC (Managed SOC) - частково або повністю делегований зовнішньому провайдеру, що забезпечує моніторинг та реагування як послугу (Security-as-a-Service).

– Гібридний SOC - поєднує внутрішні ресурси організації з можливостями зовнішніх аналітичних платформ.

У сучасних умовах SOC стає ключовим елементом концепції Zero Trust та Cyber Resilience, забезпечуючи безперервний моніторинг і адаптивну реакцію на нові вектори атак. Розвиток SOC спрямований у бік автоматизації процесів за допомогою технологій SOAR (Security Orchestration, Automation and Response), штучного інтелекту та машинного навчання, що дозволяє підвищити ефективність реагування та зменшити навантаження на аналітиків.

Використання інструментів з відкритим кодом (open source), таких як Wazuh, Suricata, Zeek, MISP, OpenSearch, Shuffle або TheHive, робить можливим створення повноцінного SOC навіть для організацій із обмеженим бюджетом. Такі рішення забезпечують високу гнучкість, прозорість і можливість кастомізації під специфічні вимоги безпеки.

Отже, SOC є стратегічним компонентом інформаційної безпеки, який забезпечує проактивний підхід до виявлення, аналізу та усунення загроз, а також підтримує процес безперервного вдосконалення системи захисту організації.

2. Архітектура та основні компоненти операційного центру безпеки

Архітектура SOC визначає організаційно-технічну структуру, взаємозв'язок компонентів та інформаційні потоки, необхідні для забезпечення повного циклу моніторингу, виявлення, аналізу та реагування на інциденти інформаційної безпеки. Ефективна архітектура SOC повинна забезпечувати централізовану обробку даних з різних джерел, інтеграцію з інфраструктурою організації та можливість масштабування відповідно до зростання обсягів інформації.

Концептуальна архітектура SOC (рисунок 1) демонструє взаємозв'язок між основними компонентами системи моніторингу, виявлення та реагування на інциденти безпеки [5]. Така архітектура визначає модель функціонування SOC у розрізі джерел даних, технологій, процесів та результатів їх взаємодії.

На схемі представлено ключові елементи SOC:

– вхідні джерела даних – системні журнали, мережевий трафік, події безпеки, системи автентифікації.

– інструменти збору та кореляції інформації – рішення SIEM, що агрегують і аналізують події з різних джерел.

– системи аналітики – модулі для виявлення аномалій, поведінкового аналізу та розслідування інцидентів.

– засоби реагування – автоматизовані механізми (SOAR) для ізоляції, блокування чи усунення загроз.

– моніторинг і звітність – панелі візуалізації, метрики продуктивності та оцінка ефективності заходів безпеки.

– вихідні результати – у вигляді сповіщень, дій з реагування та показників ефективності.

Це дозволяє забезпечити цілісне уявлення про структуру SOC і принципи взаємодії його компонентів.

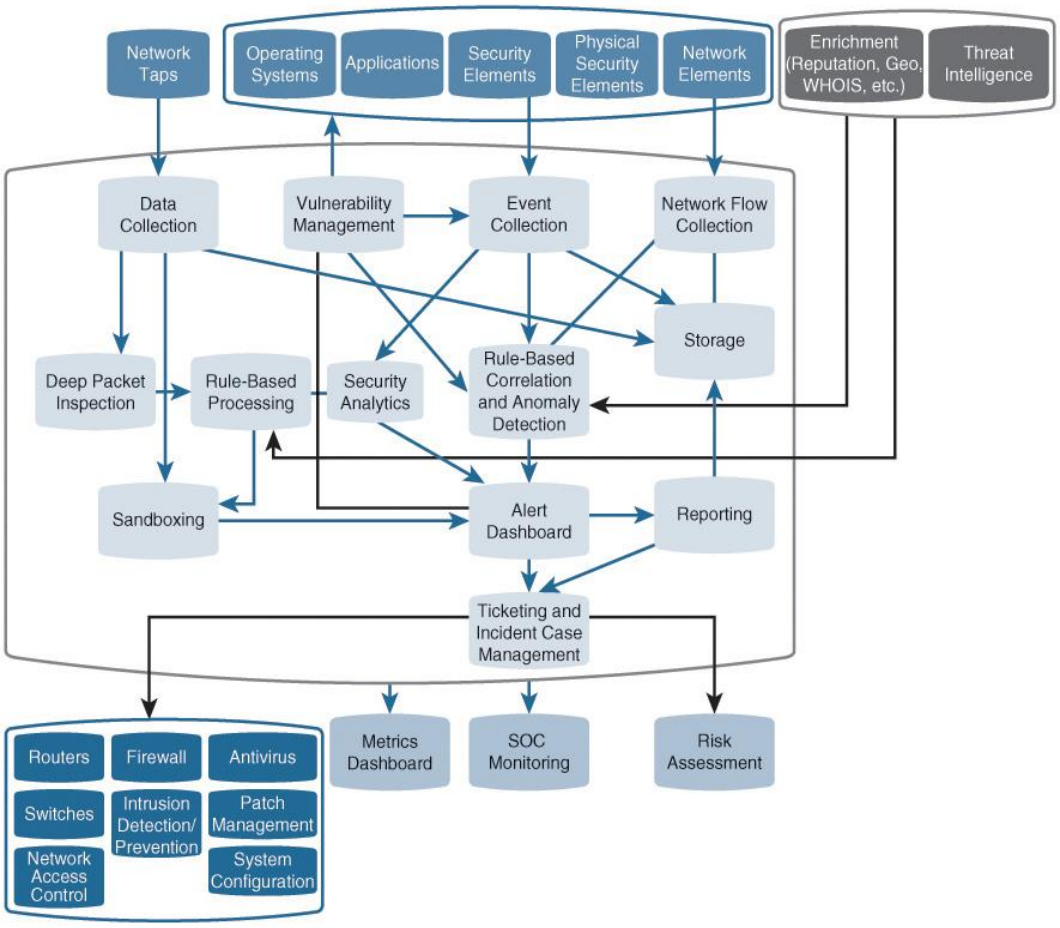


Рисунок 1 - Концептуальна архітектура SOC

Серед типових архітектурних моделей SOC можна виділити:

– монолітну (All-in-One) - усі функції (SIEM, IDS, FIM, SOAR) реалізовані в одному комплексному рішенні, наприклад, Security Onion або Wazuh All-in-One. Такий підхід зручний для малих організацій і навчальних середовищ.

– модульну (Distributed/Scalable) - окремі компоненти SOC (Wazuh, Suricata, Zeek, MISP, Shuffle) розміщені на різних вузлах і взаємодіють через API та черги повідомлень. Цей підхід забезпечує гнучкість, масштабованість та можливість розподіленої обробки даних.

Архітектура SOC являє собою багаторівневу інтегровану систему, яка поєднує апаратні, програмні та організаційні засоби для забезпечення комплексного захисту інформаційного середовища. Від ефективності взаємодії її компонентів залежить швидкість виявлення, точність аналізу та своєчасність реагування на кіберзагрози.

3. Сучасні підходи щодо розвитку SOC

Сучасний розвиток SOC передбачає використання комплексних методів і технологій для підвищення ефективності виявлення, аналізу та реагування на кіберзагрози. Впровадження SOC на базі open-source технологій передбачає поєднання кількох аспектів: вибір компонентів, інтеграцію між ними та налаштування процесів збору, аналізу і реагування на інциденти. Основні підходи включають:

– Вибір стеку технологій – визначаються інструменти для SIEM/XDR (Wazuh), мережевого моніторингу та виявлення вторгнень (Suricata, Zeek), управління інтелектом про загрози (MISP), автоматизації реагування (SOAR – Shuffle або StackStorm) та збору повного трафіку (Arkime).

– Модульний та інтегрований підхід – SOC може будуватися як єдина платформа (Security Onion) або як модульний стек із розподіленими сервісами, що дозволяє масштабувати інфраструктуру та додавати нові функції без повного переналаштування системи.

– Налаштування процесів збору та обробки даних включає підключення агентів на кінцевих точках (Windows, Linux), налаштування логування з мережевих сенсорів, інтеграцію фідів IOC у SIEM, створення правил детекції (Sigma, YARA, Suricata rules) та формування процедур реагування.

– Організація реагування – впровадження SOAR дозволяє автоматично обробляти сповіщення, виконувати сценарії ізоляції інфікованих систем, блокування загроз і створення кейсів інцидентів, що зменшує час реагування (MTTR) і підвищує ефективність SOC.

– Тестування та оцінка ефективності включає експерименти та сценарії атак, перевірку працездатності правил і кореляційних алгоритмів, а також збір метрик ефективності SOC для подальшого удосконалення.

Висновок. Побудова SOC базується на чіткій архітектурі, інтеграції основних компонентів та застосуванні стандартизованих процесів моніторингу й реагування. Реалізація SOC на базі open-source дозволить створювати гнучкі, масштабовані та адаптивні рішення, що сприятиме підвищенню кіберстійкості, оптимізації процесів управління інформаційною безпекою.

Перелік використаних джерел.

1. Security Operations Center (SOC). [Електронний ресурс].- Режим доступу: https://www.wallarm.com/what/security-operations-center-soc

2. What Is A Security Operations Center? [Електронний ресурс].- Режим доступу: https://purplesec.us/learn/security-operations-center-soc/

3. Security Operation Center (SOC). [Електронний ресурс].- Режим доступу: https://blogs.halodoc.io/security-operation-center-soc/

4. Building an Intelligent Security Operations Center. [Електронний ресурс].- Режим доступу: https://www.balbix.com/insights/introduction-to-security-operations-center/

5. Overview of Security Operations Center Technologies. [Електронний ресурс].- Режим доступу: https://www.ciscopress.com/articles/article.asp?p=2455014&seqNum=7

УДК 004.056.53

Єрмак А.Р., Алексєєва С.А.

Національний університет «Одеська політехніка»

КІБЕРБЕЗПЕКА МОЛОДІ: РОЛЬ ОСВІТИ У ФОРМУВАННІ БЕЗПЕЧНОЇ ПОВЕДІНКИ В ЦИФРОВОМУ ПРОСТОРІ

Вступ. Сучасне суспільство переживає епоху масштабної цифровізації, що охоплює всі сфери життя. Молодь є найактивнішою категорією користувачів цифрових технологій: понад 75% підлітків мають мобільні телефони, які використовуються для спілкування, навчання та розваг, а більше половини відвідують соціальні мережі кілька разів на день. Водночас, інтенсивне використання цифрових технологій супроводжується зростанням кількості та складності кіберзагроз, спрямованих саме на молоде покоління.

Статистика свідчить про серйозність проблеми: за даними DQ Institute, майже 70% дітей та підлітків у світі зазнали впливу кіберризиків у 2023 році [1]. Дослідження показують, що 91% підлітків ділилися своїми зображеннями онлайн, а більшість з них також розголошували персональні дані, такі як місцезнаходження, електронна адреса.При цьому 95% витоків даних у 2024 році були пов'язані з людським фактором, що підкреслює критичну важливість освітніх ініціатив у сфері кібербезпеки [2].

Глобальний контекст також тривожний: за дослідженням CheckPoint, глобальні кібератаки зросли на 30% у другому кварталі 2024 року, досягнувши 1636 атак на організацію щотижня. Особливу тривогу викликає той факт, що молодь, попри високий рівень технічної грамотності, демонструє низьку культуру кібербезпеки: використовує слабкі паролі, ігнорує оновлення безпеки та неусвідомлено розголошує персональні дані в соціальних мережах.

Ця ситуація загострюється відсутністю системного підходу до навчання основ кібербезпеки в освітніх закладах України, що створює значний розрив між рівнем цифрової активності молоді та їхньою здатністю захищати себе від кіберзагроз у цифровому просторі.

Мета: Обґрунтувати необхідність системного впровадження освіти з кібербезпеки для молоді та визначити ключові напрями формування культури безпечної поведінки в цифровому просторі.

1. Аналіз кіберзагроз для молоді

Цифрове середовище, в якому сьогодні зростає молодь, характеризується безпрецедентним рівнем інтерактивності та відкритості. За даними Pew Research Center, 73% підлітків щодня відвідують YouTube, включаючи 15%, які описують своє використання як "майже постійне", а близько 60% щодня відвідують TikTok[3]. У 2023 році 77% учнів старших класів використовували соціальні мережі кілька разів на день. Така інтенсивна цифрова активність створює широкий спектр кіберзагроз.

Фішинг та соціальна інженерія. Молоді користувачі є особливо вразливими до фішингових атак через недостатній досвід розпізнавання шахрайських повідомлень. Зловмисники активно використовують популярні серед молоді

платформи для поширення шкідливих посилань, створення фейкових профілів та виманювання персональних даних.

Кібербулінг. У 2024-2025 роках з'явилися нові форми кібербулінгу: підлітки використовують штучний інтелект для створення підроблених, неприйнятних зображень своїх однокласників і потім розповсюджують їх. Це призводить до серйозних психологічних наслідків для жертв, які відчувають інтенсивну втрату контролю над своїм цифровим іміджем.

Витік персональних даних у соціальних мережах. Молоді користувачі часто не усвідомлюють довгострокові наслідки публікації особистої інформації онлайн. Вони розміщують дані про своє місцезнаходження, навчальні заклади, розклад дня, що може бути використано зловмисниками для різних цілей - від таргетованих атак до фізичного переслідування.

Шкідливе програмне забезпечення. Завантаження неліцензійного контенту, ігор, додатків з неперевірених джерел часто призводить до зараження пристроїв молоді шкідливими програмами, включаючи програми-вимагачі, шпигунське ПЗ та криптомайнери.

Аналіз поведінкових патернів молодих користувачів виявляє декілька ключових проблем, що підвищують їхню вразливість до кіберзагроз.

Ризикована онлайн-активність. Молодь демонструє схильність до експериментування в цифровому просторі без належної оцінки ризиків. Це включає спілкування з незнайомцями, участь у сумнівних онлайн-активностях, обмін особистими фотографіями та відео без розуміння можливих наслідків.

Відсутність навичок кібергігієни. Базові практики кібербезпеки, такі як використання надійних паролів, двофакторна аутентифікація, регулярне оновлення програмного забезпечення, часто ігноруються молодими користувачами. Вони використовують однакові прості паролі для різних сервісів, не перевіряють налаштування приватності в соціальних мережах, не роблять резервні копії важливих даних.

Довіра до неперевірених джерел. Молодь часто не володіє навичками критичної оцінки інформації в цифровому середовищі. Вони можуть довіряти фейковим новинам, переходити за підозрілими посиланнями, завантажувати файли з незнайомих джерел, не перевіряючи їхню автентичність та безпечність.

Ілюзія анонімності та безкарності. Багато молодих людей вважають, що їхні дії в інтернеті є анонімними і не матимуть реальних наслідків. Це призводить до необережної поведінки, публікації компрометуючого контенту, участі в незаконних онлайн-активностях.

Ці особливості поведінки, поєднані з відсутністю систематичної освіти з кібербезпеки, створюють критичну вразливість молодого покоління до широкого спектру кіберзагроз у цифровому просторі.

2. Роль освіти у формуванні безпечної поведінки

Освіта відіграє ключову роль у формуванні цифрової культури молодого покоління. Вона забезпечує не лише технічну підготовку користувачів, а й розвиток усвідомленого ставлення до власної діяльності в інформаційному просторі. Формування культури безпечної поведінки вимагає системного підходу, який охоплює всі рівні освіти - від школи до університету.

Система освіти України перебуває на етапі поступового впровадження тематики кібербезпеки в навчальні програми. У шкільному курсі інформатики передбачено базові знання про безпечну роботу в мережі, захист персональних даних і правила поведінки в соціальних медіа[4]. Однак навчання має фрагментарний характер, бракує системності та практичної спрямованості. У закладах вищої освіти основна увага зосереджена на професійній підготовці фахівців з кібербезпеки, тоді як формування базової цифрової культури серед широкої молодіжної аудиторії залишається недостатньо розвиненим. Позитивну роль відіграють освітні ініціативи МОН України та громадських організацій, однак вони мають локальний характер.

У провідних країнах світу (США, Велика Британія, Сінгапур, Канада) освіта з кібербезпеки є частиною національних стратегій цифрової трансформації. Навчання здійснюється безперервно: від початкової школи, де діти опановують основи цифрової етики, до університетів, які пропонують прикладні курси та симуляційні тренінги. Поширеною практикою є проведення CTF-змагань, хакатонів, використання ігрових платформ і симуляторів кіберзагроз. Ефективним також визнано підвищення кваліфікації педагогів, що забезпечує сталість і якість навчального процесу.

Сучасні підходи передбачають використання гейміфікації, інтерактивних платформ, симуляцій та навчальних ігор, що сприяють підвищенню мотивації учнів. CTF-змагання, освітні хакатони та квести розвивають логіку, командну взаємодію та інтерес до практичного застосування знань. Інтеграція основ кібербезпеки у шкільні предмети, онлайн-курси та позакласну діяльність створює умови для формування культури безпечної поведінки з раннього віку.

3. Виклики та перспективи

Розвиток кібербезпеки в українській освіті супроводжується низкою суттєвих викликів, що стримують формування ефективної системи підготовки молоді до безпечної діяльності у цифровому середовищі. Одним із ключових бар'єрів є брак кваліфікованих викладачів, здатних поєднувати педагогічні навички з актуальними знаннями у сфері інформаційної безпеки. Більшість педагогів не мають спеціальної підготовки з питань кіберзахисту, тому передавання учням практичних навичок відбувається обмежено або поверхово.

Ще однією проблемою є відсутність стандартизованих навчальних матеріалів. Освітні програми, які стосуються цифрової безпеки, часто створюються окремими ентузіастами, громадськими організаціями чи IT-компаніями без єдиних методичних підходів і державного регулювання. Це призводить до нерівномірного рівня знань серед учнів різних навчальних закладів. Водночас, стрімкий розвиток інформаційних технологій і швидкість зміни кіберзагроз роблять існуючі навчальні матеріали швидко застарілими, що вимагає постійного оновлення змісту освіти.

Перспективи вдосконалення освіти з кібербезпеки пов'язані передусім із формуванням національної стратегії цифрової безпеки в освіті. Така стратегія має визначити єдині стандарти, компетентнісну модель та механізми реалізації на всіх рівнях освіти. Важливим напрямом є створення адаптивних навчальних програм,

здатних швидко реагувати на нові виклики у сфері інформаційних технологій. Одночасно необхідно інвестувати у підготовку викладацьких кадрів, організовуючи системні курси підвищення кваліфікації, стажування та спільні освітні ініціативи з представниками ІТ-сектору.

Велике значення має розвиток партнерства між закладами освіти, ІТ-компаніями та кіберспільнотою. Таке співробітництво дозволяє інтегрувати у навчальний процес реальні кейси, симуляції кіберінцидентів, практичні тренінги та хакатони. Спільні проєкти сприяють формуванню у молоді прикладних навичок і створюють міст між академічними знаннями та реальними потребами ринку праці.

Висновок. Результати аналізу свідчать, що формування культури безпечної поведінки неможливе без системного освітнього підходу. Освіта має стати центральним елементом національної політики кіберзахисту, спрямованої на підготовку свідомих та відповідальних користувачів цифрового простору. Практична значущість полягає у можливості впровадження розроблених підходів у навчальні програми різних рівнів освіти. Перспективними напрямами подальших досліджень є розробка цифрових симуляторів, методів гейміфікації та оцінювання рівня кіберкомпетентності учнів.

Перелік використаних джерел.

1. DQ Institute. Cyber risk exposure among children and adolescents. Security Magazine, 2023. URL: https://www.securitymagazine.com/articles/100099-almost-70-of-children-and-adolescents-have-been-exposed-to-cyber-risks

2. Mimecast. State of Human Risk Report 2024 SC Media, 2025. URL: https://www.scworld.com/news/95-of-data-breaches-involve-human-error-report-reveals

3. Chang V. Cybersecurity for children: an investigation into the application of social media. Taylor & Francis, 2023.URL: https://www.tandfonline.com/doi/full/10.1080/17517575.2023.2188122#abstract

4. Міністерство освіти і науки України. Модельні навчальні програми для 5-9 класів Нової української школи. URL: https://mon.gov.ua/osvita-2/zagalna-serednya-osvita/osvitni-programi/modelni-navchalni-programi-dlya-5-9-klasiv-novoi-ukrainskoi-shkoli-zaprovadzhuyutsya-poetapno-z-2022-roku

УДК 004.056.5

Владислав ОСІДАК

Західноукраїнський національний університет

ПОВЕДІНКОВИЙ АНАЛІЗ У ЗАДАЧІ ВИЯВЛЕННЯ ШКІДЛИВИХ ПРОГРАМ

Вступ. Актуальність теми "Поведінковий аналіз у задачі виявлення шкідливих програм" обумовлена зростаючими загрозами кібербезпеки та складністю виявлення нових типів шкідливого ПЗ. Традиційні методи, які базуються на сигнатурах, часто не здатні ефективно розпізнати нові або модифіковані віруси.

Поведінковий аналіз дозволяє відстежувати дії програм у реальному часі, що дає змогу виявляти шкідливі програми навіть до їх поширення. Це робить методи поведінкового аналізу важливими для підвищення ефективності захисту від кіберзагроз.

Метою дослідження є аналіз ефективності застосування методів поведінкового аналізу для виявлення шкідливих програм. Дослідження також має на меті виявити основні переваги та обмеження цього підходу порівняно з традиційними методами. Окрім того, буде розглянуто можливості інтеграції поведінкового аналізу в сучасні системи кіберзахисту для покращення виявлення та нейтралізації нових загроз.

1. Підходи до аналізу шкідливого програмного забезпечення

Шкідливе програмне забезпечення вже довгий час є однією з основних загроз в галузі інформаційної безпеки. Підходи до аналізу та захисту від таких атак бувають різні. Загалом поділяють два підходи: статичний та динамічний аналіз.

Завдання статичного аналізу - пошук шаблонів шкідливого вмісту у файлі чи пам'яті процесу. Це можуть бути рядки, фрагменти закодованих або стислих даних, послідовності компільованого коду. Може здійснюватися пошук як окремих шаблонів, а й їх комбінацій з додатковими умовами (наприклад, з прив'язкою до місця знаходження сигнатури, перевіркою відносної відстані в розміщені один від одного).

Динамічний аналіз – це аналіз поведінки програми. Варто зазначити, що програма може бути запущена в так званому емульованому режимі. Передбачається безпечне інтерпретування дій без завдання пошкоджень операційній системі. Інший спосіб - запуск програми у віртуальному середовищі (пісочниці). У такому разі буде чесне виконання дій на системі з подальшою фіксацією дзвінків. Ступінь подробиці логування - це свого роду баланс між глибиною спостереження та продуктивністю аналізуючої системи. На виході виходить журнал дій програми операційній системі (траса поведінки), який піддається подальшому аналізу.

Динамічний чи поведінковий аналіз дає ключову перевагу - незалежно від спроб заплутування програмного коду та прагнень приховати наміри зловмисника від вірусного аналітика шкідливий вплив буде зафіксовано. Зведення завдання

виявлення ВПО до аналізу дій дозволяє висунути гіпотезу про стійкість просунутого алгоритму виявлення шкідливих даних. А відтворюваність поведінки, завдяки тому самому початковому стану середовища для аналізу (зліпка стану віртуального сервера), спрощує вирішення завдання класифікації легітимного і шкідливого поведінки.

Часто підходи у поведінковому аналізі ґрунтуються на наборах правил. Експертний аналіз переноситься в сигнатури, на основі яких інструмент детекту шкідливого ПЗ та файлів робить висновки. Однак у такому разі може виникнути проблема: можуть враховуватися лише ті атаки, які суворо відповідають написаним правилам, а атаки, які не виконують ці умови, але все ще шкідливі, можна пропустити. Та ж проблема виникає у разі змін одного й того ж шкідливого ПЗ. Вирішити це можна за допомогою більш м'яких критеріїв спрацьовування, тобто можна написати більш загальне правило, або за допомогою великої кількості правил під кожен шкідливість. У першому сценарії ми ризикуємо отримати багато помилкових спрацьовувань, а другий вимагає серйозних витрат за часом, що може призвести до запізнення необхідних оновлень.

З'являється потреба у поширенні вже наявних знань інші схожі випадки. Тобто ті, які раніше ми не зустрічали і не обробляли правилами, але на основі схожості деяких ознак можемо зробити висновок, що активність може бути шкідливою. Тут і допомагають алгоритми машинного навчання.

ML-моделі під час коректного навчання мають узагальнюючу здатність. Це означає, що навчена модель не просто вивчила всі приклади, на яких навчалася, а здатна приймати рішення для нових прикладів на основі закономірностей із навчальної вибірки.

Однак для того, щоб узагальнююча здатність працювала, необхідно враховувати два основні фактори на етапі навчання:

Набір ознак повинен бути якомога повнішим (щоб модель могла бачити якнайбільше закономірностей, відповідно, краще поширювала свої знання на нові приклади), але не надлишковим (щоб не зберігати і не обробляти ознаки, які не несуть у собі корисну інформацію для моделі).

Набір даних має бути репрезентативним, збалансованим та регулярно оновлюваним.

2. Процес переносу експертного знання в моделі машинного навчання

У контексті аналізу шкідливого програмного забезпечення вихідні дані - це самі файли, а проміжні дані - це створені ними допоміжні процеси. Процеси, у свою чергу, здійснюють системні виклики. Послідовності таких викликів є дані, які нам необхідно перетворити на набір ознак.

Складання датасету розпочалося на експертній стороні. Було обрано ознаки, які, на думку експертів, мають бути значущими з погляду виявлення ШПЗ. Усі ознаки можна було звести до виду n-грам за системними викликами.

Далі за допомогою моделі проведена оцінка, тих ознак які роблять найбільший внесок у виявлення, відкинули зайве і отримали підсумкову версію датасета.

Вихідні дані:

{"count":1,"PID":"764","Method":"NtQuerySystemInformation","unixtime":"1639557419.628073","TID":"788","plugin":"syscall","PPID":"416","Others":"REST:
,Module=\"nt\",vCPU=1,CR3=0x174DB000,Syscall=51,NArgs=4,SystemInformationClass=0x53,SystemInformation=0x23BAD0,SystemInformationLength=0x10,ReturnLength=0x0","ProcessName":"windows\\system32\\svchost.exe"}

{"Key":"\\registry\\machine","GraphKey":"\\REGISTRY\\MACHINE","count":1,"plugin":"regmon","Method":"NtQueryKey","unixtime":"1639557419.752278","TID":"3420","ProcessName":"users\\john\\desktop\\e95b20e76110cb9e3ecf0410441e40fd.exe","PPID":"1324","PID":"616"}

{"count":1,"PID":"616","Method":"NtQueryKey","unixtime":"1639557419.752278","TID":"3420","plugin":"syscall","PPID":"1324","Others":"REST:
,Module=\"nt\",vCPU=0,CR3=0x4B7BF000,Syscall=19,NArgs=5,KeyHandle=0x1F8,KeyInformationClass=0x7,KeyInformation=0x20CD88,Length=0x4,ResultLength=0x20CD98","ProcessName":"users\\john\\desktop\\e95b20e76110cb9e3ecf0410441e40fd.exe"}

3. Покращення якості моделі з кожним оновленням

Вважаємо вибірку найбільш коректною, тому що приклади цієї вибірки перевіряються і розмічуються експертами вручну, і з кожним оновленням перевіряється в першу чергу те, що гарантується 100% точності на цій вибірці. Тестування in the wild підтверджує, що точність покращується.

Досягається це за рахунок очищення навчальної вибірки від еталонних даних, що суперечать. Під даними, що суперечать, ми розуміємо приклади, накопичені з потоку, які досить близькі по векторній відстані до трас з еталонної вибірки, але при цьому мають протилежну мітку.

Експерименти показали, що такі приклади є викидами навіть з погляду даних із потоку, оскільки після видалення їх із навчальної вибірки з метою підвищення точності на еталонній вибірці, зростала і точність на потоці.

4. Взаємодоповнення ML-підходу та поведінкових детектів у вигляді кореляцій

ML-модель дуже добре проявила себе у поєднанні з поведінковими детектами у вигляді кореляцій. Важливо зауважити, що саме в поєднанні, так як узагальнююча здатність моделі хороша у випадках, коли необхідно розширити рішення виявленням схожих та близьких інцидентів, але не у випадках, коли потрібен детект у рамках чіткого розуміння правил та критеріїв того, що є шкідливим ПЗ.

Прикладами, де ML-підхід зміг дійсно розширити рішення, стали:

– Аномальні ланцюжки підпроцесів. Саме собою велика кількість гіллястих ланцюжків - явище легітимне. Але аномальність у кількості вузлів, ступеня вкладеності, повторюваності чи повторюваності якихось конкретних імен процесів модель зауважує, а людина заздалегідь таке не нафантазує знайти шкідливим.

– Нестандартні параметри дзвінків за промовчанням. Найчастіше

аналітика цікавлять значні параметри функцій, у яких шукають ШПЗ. Інші параметри, грубо кажучи, значення за замовчуванням, вони не особливо цікавлять. Але в якийсь момент так виходить, що замість припустимо п'яти значень за умовчанням зустрічається шосте. Аналітик міг припустити, що таке можливо, а модель помітила.

– Нетипові послідовності викликів функцій. Той випадок, коли кожна функція окремо робить нічого шкідливого. Та й разом - теж. Але так сталося, що їхня послідовність не зустрічається в легітимному ПЗ. Аналітику буде потрібний гігантський досвід, щоб самостійно помітити таку закономірність. А модель помічає (і не одну), вирішуючи нестандартно завдання класифікації за ознакою, яка взагалі не закладалася як показник шкідливості.

Використання конкретного компонента одним викликом для шкідливої дії. Система використовує сотні об'єктів у різній варіативності, різною мірою. Вловити використання одного на тлі мільйона інших навряд чи вдасться - гранулярність аномалії все ж таки занизька. Проактивний детект за моделлю загроз. Вирішили, що певний вплив на певний об'єкт у системі хоча б один раз, неприпустимо. Модель може з першого разу не зрозуміти, що це значуще явище і буде шанс помилки чи невпевненого рішення на етапі класифікації чогось схожого. Обфускація послідовності процесів. Наприклад, може бути відомо, що потрібно зробити 3-4 дії у визначеному порядку. Не має значення, що буде між ними. Якщо накидати випадкові дії між 3-4 ключовими - це модель, рішення буде прийнято неправильно. При цьому розмірність числа ознак не дозволяє враховувати такі заплутування зберігання всіх комбінацій послідовностей викликів, а не тільки загальної кількості.

Висновок. Поведінковий аналіз є ефективним інструментом для виявлення шкідливих програм, оскільки дозволяє виявляти нові та модифіковані загрози, яких не можна розпізнати за допомогою традиційних методів. Він забезпечує гнучкість та адаптивність у боротьбі з кіберзагрозами, враховуючи динамічний характер сучасних вірусів. Однак для досягнення максимальної ефективності необхідно інтегрувати поведінковий аналіз із іншими методами захисту. У результаті, використання цього підходу може значно підвищити рівень безпеки в цифрових середовищах.

Перелік використаних джерел.

1. A. A. Selçuk, F. Orhan and B. Batur, "Undecidable problems in malware analysis," 2017 12th International Conference for Internet Technology and Secured Transactions (ICITST), Cambridge, UK, 2017, pp. 494-497, doi: 10.23919/ICITST.2017.8356458.

2. A. Afreen, M. Aslam and S. Ahmed, "Analysis of Fileless Malware and its Evasive Behavior," 2020 International Conference on Cyber Warfare and Security (ICCWS), Islamabad, Pakistan, 2020, pp. 1-8, doi: 10.1109/ICCWS48432.2020.9292376.

3. O. Or-Meir, A. Cohen, Y. Elovici, L. Rokach and N. Nissim, "Pay Attention: Improving Classification of PE Malware Using Attention Mechanisms Based on System Call Analysis," 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 2021, pp. 1-8, doi: 10.1109/IJCNN52387.2021.9533481.

СЕКЦІЯ 2

БЕЗПЕКА ІНТЕРНЕТ РЕЧЕЙ

УДК 004.056.53:004.89

Анастасія КАРА

Національний університет «Одеська політехніка»

ІНТЕЛЕКТУАЛЬНИЙ АНАЛІЗ ФІШИНГОВИХ АТАК З ВИКОРИСТАННЯМ EXPLAINABLE AI І ГЕНЕРАТИВНИХ МОДЕЛЕЙ

Вступ. Фішинг залишається одним із найпоширеніших і найнебезпечніших видів соціотехнічних атак. Зловмисники постійно вдосконалюють підходи, використовуючи персоналізовані повідомлення, підроблені вебресурси та генеративні моделі для створення переконливих контентів. У результаті традиційні засоби захисту – сигнатури, чорні списки чи прості евристики – втрачають ефективність у динамічному середовищі загроз.

Моделі машинного навчання підвищують точність виявлення фішингових атак через аналіз численних ознак (структури URL, контенту, поведінки користувачів). Проте вони часто працюють як «чорні скриньки», не пояснюючи причини рішень, що знижує довіру й ускладнює аудит безпекових систем.

У цьому контексті важливу роль відіграють методи Explainable AI (XAI), які дають змогу інтерпретувати поведінку моделей і пояснювати результати класифікації [1,2]. Поєднання XAI з генеративними моделями, здатними створювати приклади фішингових сценаріїв, відкриває нові можливості для побудови гібридних систем аналізу [3] – таких, що не лише виявляють атаки, а й навчаються на контрприкладах, підвищуючи власну інтерпретованість.

Таким чином, інтеграція Explainable AI та генеративних моделей у процес виявлення фішингових атак спрямована на підвищення точності, прозорості та адаптивності інтелектуальних систем кіберзахисту нового покоління.

Мета. Метою дослідження є розроблення підходу до інтелектуального аналізу фішингових атак на основі поєднання Explainable AI (XAI) та генеративних моделей, що забезпечує високу точність виявлення загроз і зрозуміле пояснення рішень системи для підвищення довіри користувачів.

Запропонований підхід має усунути обмеження традиційних систем, які працюють як «чорні скриньки» без можливості інтерпретації результатів. Інтеграція XAI-методів (SHAP, LIME, Grad-CAM) із генеративними моделями (VAE, Diffusion, GPT-подібними трансформерами) дозволяє не лише детектувати фішингові об'єкти, а й пояснювати, які ознаки вплинули на рішення [1–3]. Для досягнення мети передбачено:

– Проаналізувати сучасні підходи до виявлення фішингових атак і визначити їхні обмеження;

– Розробити архітектуру комбінованої моделі, що поєднує XAI-механізми з генеративними нейромережами;

– Реалізувати експериментальну систему на основі Python-бібліотек (scikit-learn, PyTorch, Captum);

– Провести порівняльний аналіз ефективності гібридної системи за

критеріями точності, інтерпретованості та стійкості до нових атак.

Результатом стане система кіберзахисту нового типу, що не лише виявляє фішингові загрози, а й формує зрозумілі пояснення своїх рішень, забезпечуючи прозорість і довіру до автоматизованого аналізу.

1. Аналіз сучасних підходів до виявлення фішингових атак

Методологічна основа дослідження ґрунтується на поєднанні двох сучасних напрямів штучного інтелекту – пояснюваного машинного навчання (Explainable AI, XAI) та генеративного моделювання (Generative AI). Такий підхід дозволяє не лише підвищити точність виявлення фішингових атак, але й отримати зрозуміле пояснення процесу прийняття рішень моделлю, що є критично важливим у сфері кібербезпеки.

У межах аналітичного етапу дослідження здійснено збір і підготовку набору даних, що включав реальні фішингові та легітимні вебсторінки з відкритих джерел (наприклад, PhishTank, OpenPhish, Kaggle) [4]. Для кожного запису було сформовано вектор ознак, який охоплював:

– синтаксичні характеристики URL (довжина, кількість піддоменів, спеціальних символів, наявність IP-адреси тощо);

– контентні особливості HTML-структури (форми, JavaScript-скрипти, метатеги);

– поведінкові показники (редиректи, час завантаження, активність скриптів).

Для побудови базової моделі класифікації використано ансамблеві методи машинного навчання (Random Forest, XGBoost) та нейронну мережу типу Multi-Layer Perceptron (MLP) [3]. Ці моделі забезпечили порівняльний базис для подальшого впровадження пояснюваних механізмів. Отримані результати аналітичного етапу стали основою для подальшого проєктування гібридної системи виявлення фішингових атак, що поєднує можливості XAI та Generative AI. Результати порівняльного аналізу підтвердили доцільність поєднання пояснюваних і генеративних підходів у рамках єдиної гібридної системи.

2. Розроблення гібридної системи на основі Explainable AI та Generative AI

Подальша частина дослідження присвячена реалізації Explainable AI-модулів, що дозволяють дослідити вплив кожної ознаки на кінцевий результат класифікації. Зокрема:

– метод LIME (Local Interpretable Model-agnostic Explanations) використано для побудови локальних пояснень окремих рішень моделі – визначення, які атрибути URL або HTML вплинули на прогноз [1];

– метод SHAP (SHapley Additive exPlanations) застосовано для оцінювання глобальної важливості ознак у всій вибірці [2];

– для нейронної мережі додатково використано Grad-CAM, який дозволяє візуалізувати увагу моделі при аналізі HTML-структури або текстового контенту [4].

Завершальним елементом розробки стала інтеграція генеративної підсистеми, яка базується на Variational Autoencoder (VAE) та GPT-подібних

трансформерах. Генеративна модель навчалася на легітимних і фішингових зразках, після чого використовувалася для:

– синтезу нових прикладів фішингових URL або текстів електронних листів, близьких до реальних;

– створення контрприкладів для тестування стійкості класифікатора до "адаптивних" атак;

– формування пояснюваних сценаріїв – тобто моделі могли не лише виявити фішинг, але й показати користувачу згенеровані приклади подібних атак для навчальних або аналітичних цілей.

Для реалізації експериментальної системи використано стек технологій: Python (scikit-learn, PyTorch, Captum, SHAP, LIME), а також бібліотеки BeautifulSoup для парсингу HTML-контенту та tldextract для аналізу структури доменів. Обчислення проводилися в середовищі Google Colab із використанням GPU-прискорення.

Комбінування XAI та Generative AI дало змогу створити адаптивну систему виявлення фішингових атак, яка не лише класифікує об'єкти, а й надає прозоре пояснення рішень і здатна розширювати власний навчальний набір за рахунок синтетичних даних. На рисунку 1 наведено структуру розробленої системи.

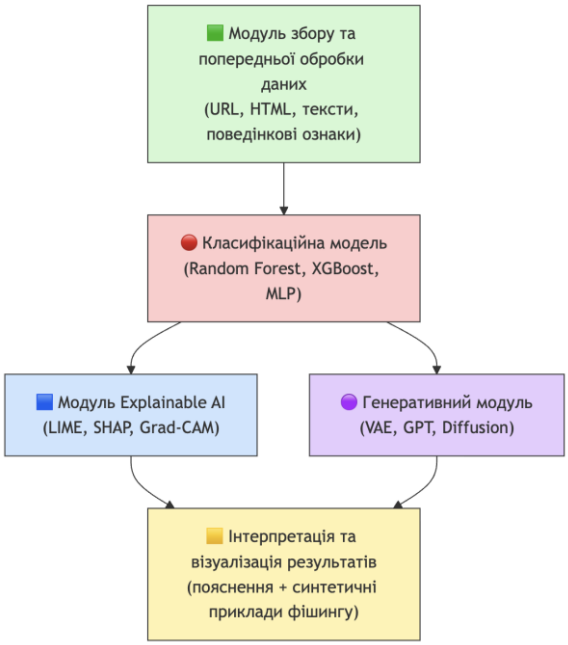


Рисунок 1 – Архітектура гібридної системи виявлення фішингових атак на основі XAI та Generative AI

За результатами реалізації запропонованого підходу створено експериментальну гібридну систему виявлення фішингових атак, що поєднує Explainable AI та генеративні моделі. Тестування виконано на наборі даних із понад 12 000 вебсторінок і 3 000 листів.

Базові моделі (Random Forest, XGBoost, MLP) досягли точності 93–96 %, проте залишалися непрозорими. Інтеграція LIME та SHAP дала змогу пояснити вплив окремих ознак (наявність IP-адреси, довжина домену, зовнішні посилання) та візуалізувати результати у зрозумілій формі, що підвищило довіру до системи.

Генеративні моделі (VAE, GPT-подібні трансформери) збагатили навчальні дані на 25 %, підвищивши точність до 97,8 % і зменшивши хибнонегативні результати на 18 %. Згенеровані приклади – реалістичні шаблони фішингових атак – використано для тренування користувачів.

Опитування студентів і фахівців з кібербезпеки підтвердило, що пояснювані моделі підвищують довіру до автоматизованих систем, а гібридний підхід забезпечує кращу стійкість до нових атак і адаптацію без ручного оновлення сигнатур. Отже, отримані результати підтверджують ефективність запропонованого підходу: поєднання XAI та Generative AI не лише підвищує точність виявлення фішингових атак, але й забезпечує прозорість, навчальний ефект і гнучкість системи при зміні середовища загроз.

Висновок. У межах дослідження розроблено гібридний підхід до виявлення фішингових атак, що поєднує Explainable AI (XAI) та генеративні моделі. Така інтеграція забезпечує високу точність класифікації й підвищує довіру користувачів завдяки прозорому поясненню роботи системи.

Система аналізує структурні, контентні та поведінкові ознаки вебресурсів і повідомлень, визначає їхній внесок у рішення та генерує синтетичні приклади атак для розширення навчальних даних. Порівняльні тести з традиційними ML-підходами показали зростання точності до 97,8 % і скорочення хибнонегативних результатів на 18 %.

Використання XAI-методів (LIME, SHAP) створило зрозумілий інтерфейс для демонстрацій і навчання, а генеративні моделі (VAE, GPT-подібні) підвищили адаптивність системи до нових сценаріїв атак [4].

У подальшому доцільно розширити дослідження через інтеграцію багатомодальних моделей, що аналізують текстові, візуальні та поведінкові ознаки, а також адаптацію системи до реального моніторингу трафіку. Це формує основу для створення прозорих і самооновлюваних систем кіберзахисту нового покоління [5].

Перелік використаних джерел.

1. Рібейро М. Т., Сінгх С., Гестрін К. "Чому я маю довіряти цій моделі?" Пояснення прогнозів будь-якого класифікатора. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16). – 2016. – С. 1135–1144.

2. Лундберг С. М., Лі С.-І. Єдиний підхід до інтерпретації прогнозів моделей. Advances in Neural Information Processing Systems (NeurIPS). – 2017.

3. Гудфеллоу І., Бенжіо Й., Курвіль А. Глибинне навчання. – Київ: Наукова думка, 2022. – 775 с.

4. Лі Ю., Ван С., Чжан С. Виявлення фішингових вебсайтів за допомогою генеративних змагальних мереж і пояснюваного ШІ. IEEE Access. – 2023. – Т. 11. – С. 67215–67229.

5. Коляда А. С., Павлишко А. В., Лопаков О. С. Криптографія після квантової ери: нові виклики та рішення для інформаційної безпеки. Інформатика та математичні методи в моделюванні. – 2024. – Т. 14, № 3. – С. 183–191.

УДК 004.056

Пашнєв Г.Р., Волошин В.Ю., Кушніренко Н.І.

Національний університет «Одеська політехніка»

РОЗРОБКА АЛГОРИТМУ ПРОТИДІЇ ПОШИРЕНИМ ВРАЗЛИВОСТЯМ БЕЗПЕКИ ВЕБ-ЗАСТОСУНКІВ

Вступ.Сучасні веб-застосунки є ключовим елементом цифрової інфраструктури, через який користувачі отримують доступ до послуг, фінансових операцій, баз даних та персональних кабінетів. Через це вони часто стають об'єктом кібератак. Найпоширенішими з них є SQL-ін'єкції, XSS, CSRF, brute-force та атаки через повторне використання токенів [1].

Наявність вразливостей у веб-застосунках може призвести до витоку конфіденційної інформації, компрометації облікових записів та порушення роботи інформаційних систем.

Мета: Аналіз основних типів атак на веб-застосунки та розробка універсального алгоритму для підвищення їх захищеності.

Основна частина

Для аналізу безпеки веб-застосунків було розглянуто модель взаємодії

«користувач – веб – сервер – база даних».

Основна мета атак на застосунки, побудовані за такою моделлю полягає в порушенні конфіденційності, цілісності або доступності даних. Одним з основних видів атак на веб-застосунки є SQL-ін'єкції, які використовують некоректну обробку введених даних для виконання довільних запитів до бази, що вимагає захисту через підготовлені вирази, валідацію та використання ORM-систем.

Загрози на клієнтському боці включають:

– Cross-Site Scripting (XSS) - вбудовування шкідливого JavaScript-коду з метою викрадення сесійних даних

– Cross-Site Request Forgery (CSRF) - підміну запитів від імені користувача.

Запобігання XSS здійснюється шляхом HTML-ескейпінгу та впровадження політики Content Security Policy, тоді як захист від CSRF забезпечують токени автентичності та заголовок SameSite.

Для захисту облікових записів від brute-force атак (автоматизований підбір паролів) критично важливе обмеження спроб входу, використання CAPTCHA та обов'язкова двофакторна автентифікація (2FA).

Проти повторного використання токенів (Replay attacks) - перехоплення дійсного токена - застосовують TLS-шифрування, встановлення короткого часу життя токена та механізми оновлення токенів після входу.

Результати дослідження щодо популярності атак, отримані на основі аналізу сучасних звітів OWASP та практичного тестування, наведені на рисунку 1.

Рисунок 1 - Кількість атак

Детальні описи кожного типу атаки, їхні наслідки та можливі методи протидії наведені в таблиці 1. На основі отриманих даних сформовано рекомендації щодо підвищення рівня безпеки веб-застосунків та мінімізації ризиків несанкціонованого доступу [3].

Таблиця 1 – Поширені атаки та методи захисту

Атака / проблема	Орієнтовна поширеність (%)	Вплив	Складність експлуатації	Рекомендовані заходи захисту
SQL-ін'єкції	11.5	Високий - витік/зміна БД	Середня	Prepared statements, ORM, WAF, SAST/DAST
Ненадійне зберігання паролів	17.5	Високий - компрометація облікових записів	Низька	Argon2/Bcrypt, соль, MFA, захищені бекапи
Недостатній контроль доступу	22.5	Дуже високий - ескалація привілеїв	Низька–середня	RBAC/ABAC, централізована авторизація, тестування прав
Витік конфіденційних даних	27.5	Дуже критичний - фінансові та репутаційні збитки	Варіюється	Шифрування at-rest/in-transit, DLP, моніторинг
Відсутність обмеження кількості спроб входу	14.0	Середній-високий - компрометація облікових записів	Низька	Rate limiting, lockout, CAPTCHA, MFA
Повторне використання токенів	8.5	Середній - відтворення транзакцій, сесійне захоплення	Середня	TLS, nonce, короткий TTL, refresh token flows

Згідно з таблицею, пріоритетом має бути захист від витоку даних через шифрування та усунення слабкого контролю доступу шляхом централізованої авторизації. Для запобігання компрометації облікових записів слід використовувати сильні алгоритми хешування (Argon2/Bcrypt) із сіллю, MFA і Rate limiting/lockout для обмеження спроб входу. SQL-ін'єкції мають блокуватись через Prepared statements, а повторне використання токенів - через TLS і короткий TTL.

Розробка алгоритму протидії поширеним вразливостям безпеки передбачає системний підхід, який охоплює всі етапи життєвого циклу програмного забезпечення - від розробки до експлуатації. Основною метою такого алгоритму є запобігання виникненню вразливостей ще на етапі створення коду, а також забезпечення швидкого виявлення і усунення потенційних загроз у вже функціонуючих системах.

Серед основних кроків алгоритму передбачено перевірку та фільтрацію всіх вхідних даних для запобігання ін'єкційним атакам, зокрема SQL Injection та XSS. Важливо забезпечити надійне шифрування інформації під час її зберігання та передавання, що мінімізує ризик витоку конфіденційних даних. Для запобігання несанкціонованому доступу алгоритм передбачає впровадження багатофакторної автентифікації, обмеження кількості спроб входу та проведення регулярного аудиту облікових записів користувачів.

Висновок. У межах дослідження проаналізовано найпоширеніші типи атак на веб-застосунки (SQL-ін'єкції, XSS, brute-force) та існуючі методи захисту від них. Результатом роботи є запропонований алгоритм, що може бути використаний під час розробки веб-застосунків з підвищеним рівнем захищеності.

Перелік використаних джерел.

1. OWASP Foundation. OWASP Top 10: 2021.

2. Сидоренко П.В. Безпека веб-застосунків: методи захисту та тестування. - Київ: КПІ, 2023.

3. Ristic I. ModSecurity Handbook: The Complete Guide to Web Application Firewalls. Feisty Duck, 2021.

УДК 004.056.5

Владислав РУЩАК, Степан ІВАСЬЄВ

Західноукраїнський національний університет

ДОСЛІДЖЕННЯ ВРАЗЛИВОСТІ БІБЛІОТЕКИ CLICKBAR/DOT-DIVER

Вступ. Бібліотека @clickbar/dot-diver, реалізована на TypeScript (відкритий вихідний код), надає зручний API для зчитування значення поля об'єкта (функція getByPath) та запису значення у поле об'єкта (функція setByPath), має широке застосування. В бібліотеці було виявлено вразливість типу prototype pollution, що критично для безпеки вебзастосунків на TypeScript, що її використовують.

Мета: дослідити шляхи реалізації вразливості prototype pollution бібліотеки dot-diver.

1. Аналіз вразливості функції `setByPath`

Уразливість виявлено у функції setByPath, яка приймає три аргументи:

– object - об'єкт, у властивість якого встановлюється значення;
– path - шаблон шляху до властивості, що підлягає зміні;
– value - значення, яке має бути записане у вказане поле.

При виклику setByPath відбувається рекурсивний обхід елементів у шаблоні шляху. Після знаходження цільового поля йому присвоюється нове значення. Нижче зазначено, що у вихідному повідомленні наводилося загальне пояснення механіки роботи setByPath.

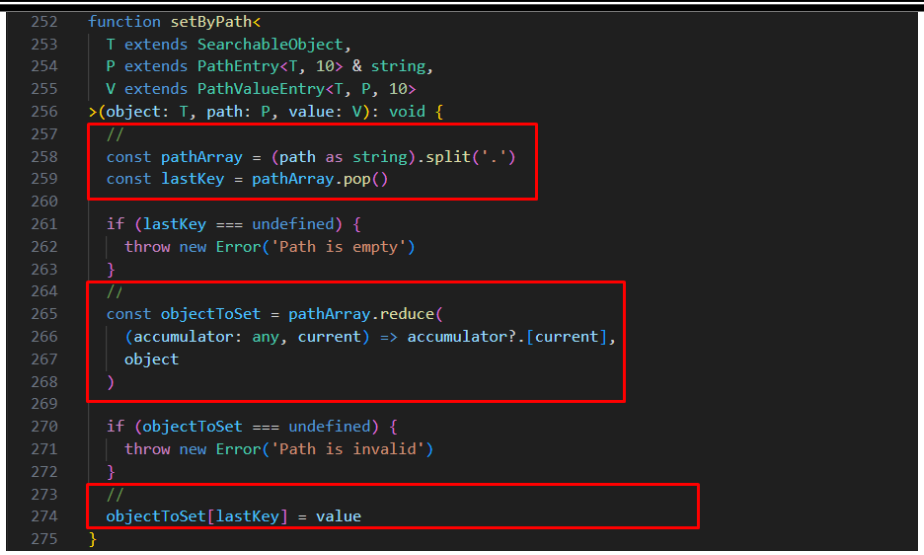
```
1  import { getByPath, setByPath } from '@clickbar/dot-diver'
2
3  // Define a sample object with nested properties
4  const object = {
5    a: 'hello',
6    b: {
7      c: 42,
8      d: {
9        e: 'world',
10     },
11   },
12   f: [{ g: 'array-item-1' }, { g: 'array-item-2' }],
13 }
14 // Example 2: Set a value by path
15 setByPath(object, 'a', 'new hello')
16 console.log(object.a) // Output: 'new hello'
17
18 setByPath(object, 'f.1.g', 'new array-item-2')
19 console.log(object.f[1].g) // Output: 'new array-item-2'
```

Рисунок 1 - Механізм роботи setByPath

Основна проблема застосування цієї функції полягає в тому, що якщо значення шляху містить посилання на прототип, з'являється можливість встановити властивості прототипу, що може призвести до забруднення глобального прототипу.

Цей же код після виправлення (версія 1.0.2) приведено на рисунку 3.

У версії 1.0.1 функція setByPath дозволяє встановлювати властивості за шляхом, що подається у вигляді рядка (наприклад, "a.b.c").



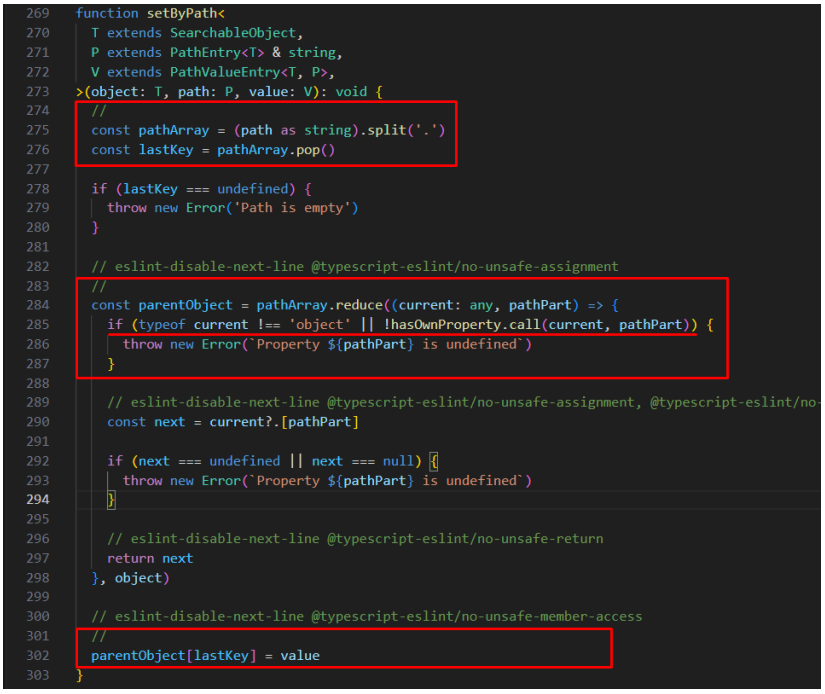
```
function setByPath<
  T extends SearchableObject,
  P extends PathEntry<T, 10> & string,
  V extends PathValueEntry<T, P, 10>
>(object: T, path: P, value: V): void {
  //
  const pathArray = (path as string).split('.')
  const lastKey = pathArray.pop()

  if (lastKey === undefined) {
    throw new Error('Path is empty')
  }
  //
  const objectToSet = pathArray.reduce(
    (accumulator: any, current) => accumulator?.[current],
    object
  )

  if (objectToSet === undefined) {
    throw new Error('Path is invalid')
  }
  //
  objectToSet[lastKey] = value
}
```

Рисунок 2 - Код функції до виправлення (версія 1.0.1)

Через відсутність перевірок спеціальних ключів (наприклад, __proto__, constructor.prototype) зловмисник може змінювати властивості прототипу - тобто відбувається prototype pollution. Це відкриває шлях до обходу механізмів контролю доступу й іншої небажаної модифікації поведінки застосунку. setByPath виконує рекурсивний або ітеративний обхід шляхового шаблона (split('.') → reduce/loop) і без додаткових перевірок записує значення у знайдене поле.

2. Аналіз виправлень вразливості

Проблемою що викликала вразливість є відсутність захисту проти модифікації властивостей прототипу: не відкидаються ключі на зразок __proto__, prototype, constructor та відсутній код, щоб перевірити, чи властивість є власною (own property) об'єкта, або чи операція створює нове поле на самому об'єкті, а не змінює глобальний прототип.



```
function setByPath<
  T extends SearchableObject,
  P extends PathEntry<T> & string,
  V extends PathValueEntry<T, P>,
>(object: T, path: P, value: V): void {
  //
  const pathArray = (path as string).split('.')
  const lastKey = pathArray.pop()

  if (lastKey === undefined) {
    throw new Error('Path is empty')
  }

  // eslint-disable-next-line @typescript-eslint/no-unsafe-assignment
  //
  const parentObject = pathArray.reduce((current: any, pathPart) => {
    if (typeof current !== 'object' || !hasOwnProperty.call(current, pathPart)) {
      throw new Error(`Property ${pathPart} is undefined`)
    }

    // eslint-disable-next-line @typescript-eslint/no-unsafe-assignment, @typescript-eslint/no-
    const next = current?.[pathPart]

    if (next === undefined || next === null) {
      throw new Error(`Property ${pathPart} is undefined`)
    }

    // eslint-disable-next-line @typescript-eslint/no-unsafe-return
    return next
  }, object)

  // eslint-disable-next-line @typescript-eslint/no-unsafe-member-access
  //
  parentObject[lastKey] = value
}
```

Рисунок 3 - Код функції після виправлення(версія 1.0.2)

У виправленій версії було додано перевірку наявності власної властивості в об'єкті (за допомогою методу Object.prototype.hasOwnProperty) перед внесенням змін. Якщо потрібне поле відсутнє, викликається виняток із відповідним повідомленням.

3. Вектор атаки забруднення прототипу

Вектор атаки можна описати наступними кроками. Зловмисник подає шлях, який містить __proto__ або інший спеціальний сегмент (через користувацькі поля/JSON). setByPath проходить шлях і в кінці записує задане значення не в локальний об'єкт, а в прототип (Object.prototype або інший спільний прототип).

Після цього будь-який інший код, що покладається на наявність певної властивості або перевіряє її присутність через спадкування, може побачити змінену поведінку (наприклад, user.isAdmin === true), що призводить до ескалації привілеїв або обходу авторизації. Алгоритм реалізації вразливості приведено на рисунку 4.

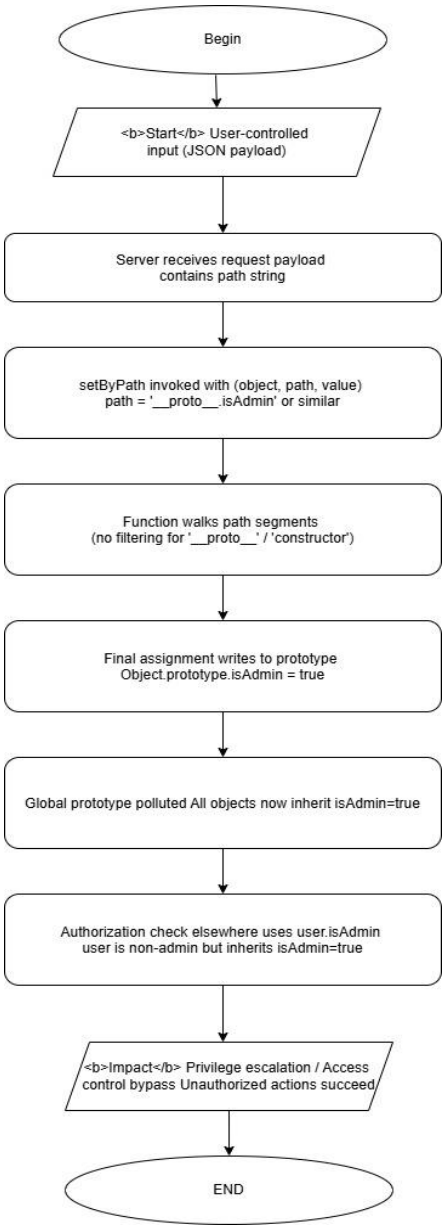


Рисунок 4 – Схема алгоритму використання вразливості setByPath

У прикладі розглядається застосунок для обліку прочитаних книг, у якому реалізовано механізм розмежування користувачів із різними ролями: user - має право додавати дані про прочитані книги та переглядати їх, admin - має право видаляти книги зі списку.

Розмежування доступу реалізовано за допомогою додаткової властивості `isAdmin`, яка присутня лише в об'єкта користувача з успішною автентифікацією для ролі admin. В інших користувачів це поле відсутнє, як показано на рисунку 5.

Рисунок 5 - Механізм роботи setByPath

Фрагмент коду, який виконує обробку запиту на видалення книги за ключем title та реалізує контроль прав доступу, може виглядати так, як показано на рисунку 6.

```
app.delete('/books/:title', (req, res) => {
  const { title } = req.params;
  const user = req.user; // об'єкт користувача, отриманий після автентифікації

  // Перевірка прав доступу
  if (!user || !user.isAdmin) {
    return res.status(403).json({
      error: 'Access denied: insufficient privileges.'
    });
  }

  // Пошук книги за назвою
  const index = books.findIndex((book) => book.title === title);

  if (index === -1) {
    return res.status(404).json({
      error: `Book with title '${title}' not found.`
    });
  }

  // Видалення книги
  books.splice(index, 1);
  return res.status(200).json({
    message: `Book '${title}' was successfully deleted.`
  });
});
```

Рисунок 6 - Фрагмент коду, який виконує обробку запиту на видалення книги

У застосунку використовуються такі API-команди, як отримати список книг:

$ curl -v -X GET -H http://192.169.27.1:6000/

Оновити список книг:

$ curl -v -X PUT -H "Content-Type:application/json" --data '{"auth":{"name":"reader", "pwd":"books"},"title":"Tom Sawyer"}' http://192.169.27.1:6000/

Видалити книгу з певною назвою:

$ curl -v -X DELETE -H "Content-Type:application/json" --data '{"auth":{"name":"reader", "pwd":"books"},"title":"Tom Sawyer"}' http://192.169.27.1:6000/

Після спроби видалення книги від імені користувача reader повертається відповідь із кодом 403 та повідомленням "Access denied", що свідчить про відсутність прав на виконання цієї операції.

До запиту додається корисне навантаження для атаки:

$ curl -v -X PUT -H "Content-Type:application/json" --data '{"auth":{"name":"reader", "pwd":"books"},"title":"Tom Sawyer", "note":"__proto__.isAdmin", "text":true}' http://192.169.27.1:6000/

У відповіді на цей запит відображається результат успішного оновлення списку книг. Окрім того, у глобальному об'єкті Object з'явилося поле isAdmin, як показано на рисунку 7.

Рисунок 7 - Object з полем isAdmin

Після повторного запиту на видалення книги від імені користувача reader у відповіді повертається повідомлення про успішне видалення книги, що свідчить про реалізацію атаки та обхід механізму розмежування доступу, реалізованого в застосунку.

Висновок. Необхідно забезпечити регулярне оновлення всіх бібліотек та залежностей до останніх стабільних версій. Більшість випадків «забруднення прототипу» виникає через застарілі пакети, що містять уразливі функції для глибокого копіювання чи об'єднання об'єктів (наприклад, у lodash, jQuery, dot-prop, deepmerge). Важливу роль відіграє також ретельна перевірка та фільтрація вхідних даних, які надходять від користувачів або зовнішніх API. Усі параметри, що використовуються для побудови об'єктів або динамічних шляхів властивостей, мають проходити валідацію.

Перелік використаних джерел.

1. GitHub Security Advisory. Prototype Pollution (PP) vulnerability in setByPath (GHSA-9w5f-mw3p-pj47). [Електронний ресурс]. – Режим доступу: // GitHub – clickbar/dot-diver. – Опубл.: 02.11.2023.

2. National Vulnerability Database (NVD). CVE-2023-45827: dot-diver – Prototype Pollution у setByPath; виправлення у релізі 1.0.2 Опубл.: 06.11.2023. [Електронний ресурс]. –Режим доступу: https://nvd.nist.gov/vuln/detail/CVE-2023-45827.

УДК 004.031

Сергій ТЕЛЕНЬКО, Сергій КУЛИНА

Західноукраїнський національний університет

СИСТЕМИ ЗАХИСТУ ПРИВАТНИХ КЛЮЧІВ НА ОСНОВІ АПАРАТНИХ МОДУЛІВ БЕЗПЕКИ

Вступ. В основі сучасної цифрової безпеки лежить асиметрична криптографія, де приватні ключі є найбільш критичним активом будь-якої організації. Вони є "ключами до королівства", що засвідчують цифрові особистості, шифрують конфіденційні дані та авторизують транзакції високої цінності. Однак традиційне зберігання цих ключів у програмному забезпеченні - у файлах конфігурації, базах даних або навіть в оперативній пам'яті сервера - робить їх надзвичайно вразливими. У разі компрометації сервера через зловмисне програмне забезпечення, мережеву атаку або інсайдерську загрозу, ці ключі можуть бути викрадені, що призводить до катастрофічних наслідків: повної втрати довіри, фінансових збитків та незворотної шкоди репутації.

Саме для вирішення цієї фундаментальної проблеми були створені апаратні модулі безпеки (Hardware Security Modules, HSM). Це спеціалізовані, захищені від фізичного втручання (tamper-resistant) пристрої, єдиною метою яких є безпечна генерація, зберігання та управління криптографічними ключами протягом усього їхнього життєвого циклу. Ключова перевага HSM полягає в тому, що приватні ключі ніколи не залишають захищеного апаратного периметра: всі криптографічні операції (наприклад, підписання або розшифрування) відбуваються безпосередньо всередині модуля. Таким чином, навіть якщо основний сервер, до якого підключений HSM, буде повністю скомпрометований, зловмисник не зможе викрасти сам приватний ключ, а отримає лише доступ до обмеженого набору функцій, що значно підвищує загальний рівень захищеності системи..

Метою дослідження є підвищення рівня безпеки та ефективності систем захисту приватних ключів шляхом розробки або вдосконалення методів їх інтеграції та управління на базі апаратних модулів безпеки (HSM).

1. Дослідження існуючих систем захисту приватних ключів на основі апаратних модулів безпеки

Системи захисту приватних ключів на основі апаратних модулів безпеки (HSM) є золотим стандартом для захисту найбільш критичних криптографічних активів в інфраструктурі будь-якої сучасної організації.

В основі цієї технології лежить принцип що приватні ключі ніколи не повинні залишати захищене апаратне середовище. У той час як програмні методи зберігають ключі у файлах, базах даних або пам'яті, що робить їх вразливими до викрадення через віруси, мережеві атаки або помилки конфігурації, HSM створює фізично та логічно ізольовану "чорну скриньку".

HSM - це спеціалізований обчислювальний пристрій, розроблений із захистом від несанкціонованого доступу. Його основні завдання - це безпечна генерація ключів, із використанням апаратного генератора випадкових чисел, захищене зберігання та суворий контроль над їхнім використанням. Коли

програмі або серверу потрібно виконати операцію з приватним ключем (наприклад, підписати документ, розшифрувати дані або випустити сертифікат), він надсилає запит до HSM. Модуль виконує операцію всередині свого захищеного криптографічного периметра і повертає лише результат, але не сам ключ. Навіть якщо зловмисник отримає повний контроль над сервером, він не зможе витягти ключ з апаратного модуля, оскільки він завжди біля власника (рисунок 1).

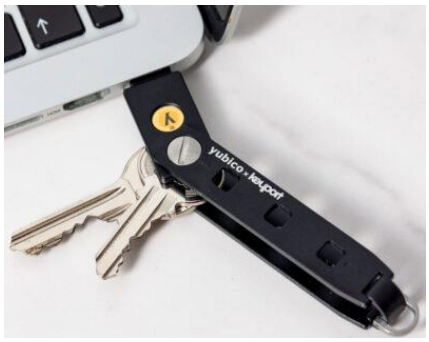


Рисунок 1 – Апаратний ключ доступу у вигляді брелка до ключів

Такі системи є життєво необхідними для інфраструктур Центрів сертифікації, де HSM захищає кореневий ключ, компрометація якого зруйнує всю систему довіри. Вони також є невід'ємною частиною банківських платіжних систем для захисту PIN-кодів та ключів транзакцій, підписання коду (для гарантії автентичності програмного забезпечення), шифрування баз даних (захист головного ключа шифрування) та блокчейн-додатків, захист ключів "гарячих" гаманців. Використання HSM кардинально підвищує стійкість системи до атак, забезпечуючи найвищий рівень гарантії, що приватні ключі залишаються справді приватними.

2. Типова структура системи захисту приватних ключів на основі HSM

Структура системи захисту приватних ключів на основі HSM є багаторівневою і поєднує програмні, апаратні та процедурні компоненти. Її типова архітектура, від "споживача" до "ядра" містить 5 рівнів:

1. Рівень застосунків (Споживачі). Бізнес-додатки, які нічого не знають про те, де знаходиться ключ, але знають, що їм потрібна криптографічна операція. Це можуть бути:

– Веб-сервери наприклад, Nginx, Apache для SSL/TLS-операцій.

– Сервери застосунків наприклад, Java, .NET для підписання токенів (JWT).

– Інфраструктура PKI - Центри сертифікації, для випуску сертифікатів.

– Бази данихтакі як Oracle, SQL Server для прозорого шифрування даних.

– Платіжні шлюзи для шифрування номерів карток.

2. Рівень абстракції (API та Клієнт HSM). Це "посередник" або "драйвер", який встановлюється на сервері застосунків. Він надає стандартизований інтерфейс для програм і "перекладає" їхні запити на мову, зрозумілу конкретному HSM.

3. Рівень підключення (Мережа або Шина). Це фізичний або логічний спосіб, у який сервери застосунків "спілкуються" з HSM. Найпоширенішим

варіантом є Мережевий HSM, якиц є окремим пристроєм у мережі (має свою IP-адресу). До нього можуть одночасно звертатися багато серверів. Зв'язок завжди захищений зашифрованим TLS-тунелем. Іншим поширеним вибором є вбудований HSM. Це карта, що вставляється безпосередньо в сервер, або USB-пристрій, який використовується коли потрібна максимальна продуктивність для одного конкретного сервера.

4. Апаратний Модуль Безпеки (HSM) - Ядро Системи. Це сам захищений пристрій, який має власну захищену операційну систему та апаратні компоненти.

Його внутрішня структура складається з:

– Захищений корпус, який має захист від несанкціонованого доступу. При спробі фізичного злому (наприклад, свердління) або зміни умов (температура, напруга) модуль автоматично знищує всі ключі.

– Криптографічний процесор – це спеціалізований чип, оптимізований для швидкого виконання асиметричних та симетричних операцій.

– Справжній генератор випадкових чисел (TRNG). Апаратний компонент для генерації криптографічно стійких ключів.

– Безпечне сховище ключів. Зашифрована енергонезалежна пам'ять, де ключі зберігаються у зашифрованому вигляді, а головний ключ, що шифрує всі інші, часто зберігається так, що його неможливо витягти.

– Захищена ОС та прошивка.

5. Рівень управління та аудиту - це процедури та інтерфейси, які використовуються для адміністрування самого HSM. Він містить безпечний канал управління - окремий, захищений інтерфейс (часто через SSH або захищений GUI) для налаштування політик, створення та видалення ключів.

Використання такої 5 рівневої структури дозволяє створити повнофункціональну систему, що відповідатиме поставленим задачам та забезпечить необхідний рівень надійності.

Висновок. З проведеного дослідження зрозуміло, що системи на основі апаратних модулів безпеки є не просто технічним доповненням, а фундаментальним рішенням однієї з найгостріших проблем кібербезпеки - захисту приватних ключів. Їхня ключова важливість полягає у створенні фізично ізольованого та захищеного від несанкціонованого доступу середовища, де криптографічні ключі генеруються, зберігаються та використовуються, але ніколи не залишають апаратного периметра. Як ми обговорили, така архітектура робить систему стійкою до програмних зломів, мережевих атак та навіть фізичного втручання. Таким чином, HSM є критичною інфраструктурною ланкою, що забезпечує найвищий рівень довіри та гарантій безпеки для важливих процесів, від функціонування центрів сертифікації до захисту фінансових транзакцій.

Перелік використаних джерел.

1. Sebestyen H., Popescu D. E., Zmaranda R. D. A Literature Review on Security in the Internet of Things: Identifying and Analysing Critical Categories. Computers. 2025. Vol. 14. № 2. Art. 61. DOI: 10.3390/computers14020061.

2. Khan, M., Ilyas, M., & Bayat, O. (2024, September). Enhancing IoT Security Through Hardware Security Modules (HSMs). In 2024 International Conference on Intelligent Computing, Communication, Networking and Services (ICCNS) pp.278-282.

УДК 004.8:004.056.5

Андрій ПРИЛОЖЕНКО, Олексій СТОПАКЕВИЧ

Національний університет «Одеська політехніка»

ШТУЧНИЙ ІНТЕЛЕКТ У СИСТЕМАХ КІБЕРБЕЗПЕКИ

Вступ. Сучасний ландшафт кіберзагроз характеризується безпрецедентним зростанням складності, швидкості та обсягів атак. Традиційні методи захисту, такі як антивіруси на основі сигнатур, міжмережеві екрани з фіксованими правилами та прості системи виявлення вторгнень (Intrusion Detection System), дедалі частіше виявляються неефективними проти новітніх загроз, зокрема атак нульового дня, поліморфного шкідливого програмного забезпечення та складних цільових атак (Advanced Persistent Threat). Ці атаки часто маскуються під легітимний трафік, що робить їх виявлення майже неможливим для статичних систем захисту.

Мета: Метою дослідження є аналіз можливостей та методів застосування технологій штучного інтелекту (ШІ), зокрема машинного (Machine Learning) та глибокого (Deep Learning) навчання, для побудови адаптивних та проактивних систем кібербезпеки, здатних ідентифікувати та реагувати на складні кіберзагрози в режимі реального часу.

1.Недоліки традиційних систем кіберзахисту

Традиційні системи, такі як Signature-based IDS/IPS, покладаються на заздалегідь відому базу даних сигнатур (зразків) відомих загроз. Розглянемо основні недоліки цього підходу.

1. Реактивність. Система може виявити лише ті загрози, які вже були ідентифіковані, проаналізовані, і для яких була створена сигнатура. Це залишає організації вразливими до атак нульового дня.

2. Обсяг сигнатур. Бази даних сигнатур розростаються до величезних розмірів, що вимагає значних обчислювальних ресурсів для сканування трафіку та файлів.

3. Маскування. Зловмисники активно використовують техніки обфускації та поліморфізму, щоб змінити «вигляд» шкідливого коду, роблячи його нерозпізнаваним для сигнатурних сканерів. Статичні правила міжмережевих екранів також не здатні аналізувати поведінковий контекст дій користувача або мережевого потоку, що дозволяє атакам розвиватися всередині периметра мережі після початкового проникнення. Виникає гостра потреба в переході від реактивного до проактивного, предиктивного захисту, який можуть забезпечити інструменти ШІ [1], [2].

2.Два підходи машинного навчання для виявлення аномалій.

Ключовою перевагою штучного інтелекту в кібербезпеці є його здатність навчатися та адаптуватися. Замість пошуку відомих «поганих» зразків, системи на основі ШІ вивчають «нормальну» поведінку мережі, користувачів, програм та пристроїв. Будь-яке відхилення від цієї встановленої базової лінії (baseline) розглядається як потенційна аномалія або загроза. Для цього використовуються переважно два такі підходи машинного навчання.

1. Навчання з учителем (Supervised Learning). Моделі (наприклад, Support Vector Machines, Random Forests, нейронні мережі) навчаються на величезних, заздалегідь розмічених наборах даних, що містять приклади як легітимного, так і шкідливого трафіку чи файлів. Цей підхід ефективний для класифікації спаму, фішингових листів та відомих типів шкідливого ПЗ.

2. Навчання без учителя (Unsupervised Learning). Цей підхід є критично важливим для виявлення атак нульового дня. Моделі (наприклад, кластеризація, K-Means, Density-Based Spatial Clustering of Applications with Noise) аналізують нерозмічені дані, самостійно знаходячи в них приховані патерни та структури. Система будує профіль нормальної активності і сигналізує про будь-які викиди (outliers), які не відповідають цьому профілю. Наприклад, якщо обліковий запис користувача, який зазвичай працює з 9:00 до 18:00 з однієї IP-адреси, раптово починає масове завантаження даних о 3:00 ночі з іншої країни – це явна аномалія, яку ШІ негайно зафіксує [1], [2].

3. Глибоке навчання та автоматизоване реагування

Розглянемо більш просунуті методи, зокрема глибоке навчання (Deep Learning). Моделі DL, такі як рекурентні нейронні мережі (РНМ) та їх різновиди (Long Short-Term Memory, Gated Recurrent Unit), особливо ефективні для аналізу послідовних даних, якими є мережеві пакети або лог-файли. Вони здатні розуміти контекст і виявляти складні, розтягнуті в часі атаки, які складаються з багатьох, на перший погляд, не пов'язаних між собою подій. Однак, виявлення загрози – це лише частина завдання. Сучасні системи кібербезпеки інтегрують ШІ в платформи класу SOAR (Security Orchestration, Automation and Response)[3].

SOAR використовує ШІ для автоматизації рутинних завдань аналітиків безпеки. Коли ML-модель виявляє аномалію, SOAR-платформа матиме такі можливості.

1. Збагатити дані. Автоматично зібрати додаткову інформацію про підозрілу IP-адресу, хеш файлу чи домен з відкритих джерел (Threat Intelligence).

2. Оцінити ризик. Присвоїти інциденту пріоритет на основі контексту (наприклад, чи стосується це критичного сервера).

3. Виконати реагування. Автоматично виконати заздалегідь визначений сценарій (playbook), наприклад, заблокувати IP-адресу на міжмережевому екрані, ізолювати інфікований пристрій від мережі або призупинити дію скомпрометованого облікового запису. Це дозволяє реагувати на загрози за мілісекунди, замість годин чи днів, які потрібні людині-аналітику. Порівняльний аналіз основних алгоритмів ML, що застосовуються в кібербезпеці, наведено в таблиці 1 [4].

Разом з тим, впровадження ШІ створює і нові виклики. По-перше, це так звані «змагальні атаки» (adversarial attacks), коли зловмисники цілеспрямовано "обманюють" ML-модель, подаючи їй на вхід згенеровані дані, що призводять до хибної класифікації[5]. Наприклад, незначна зміна кількох пікселів у зображенні може змусити нейронну мережу "не побачити" загрозу. По-друге, ШІ-моделі вимагають величезних обсягів якісних даних для навчання, що може бути проблемою для багатьох організацій.

Таблиця 1 – Порівняння алгоритмів ШІ для завдань кібербезпеки

Алгоритм	Тип навчання	Основне завдання	Переваги	Недоліки
Random Forest	З учителем	Класифікація (спам, шкідливе ПЗ)	Висока точність, стійкість до перенавчання	"Чорна скринька", важко інтерпретувати
K-Means	Без учителя	Виявлення аномалій (поведінка)	Простота, швидкість на великих даних	Потрібно знати кількість кластерів (k)
DBSCAN	Без учителя	Виявлення аномалій (викиди)	Не вимагає знання 'k', знаходить кластери довільної форми	Чутливий до параметрів, повільний
LSTM (RNN)	З учителем/без учителя	Аналіз послідовностей (мережеві пакети, лог-файли)	Розуміння контексту, виявлення APT	Висока обчислювальна складність

Висновок. Отже, штучний інтелект фундаментально змінює парадигму кібербезпеки, переміщуючи фокус з реактивного захисту на основі сигнатур до проактивного виявлення загроз на основі аналізу поведінки та аномалій. Методи машинного та глибокого навчання дозволяють ідентифікувати складні та раніше невідомі атаки. Інтеграція ШІ з платформами SOAR забезпечує автоматизоване реагування, що критично важливо в умовах зростаючої швидкості кібератак. Попри виклики, пов'язані зі змагальними атаками та потребою у великих даних, ШІ є сьогодні найперспективнішим інструментом для побудови стійких та адаптивних систем захисту інформації в майбутньому.

Перелік використаних джерел.

1. Buczak, A. L., & Guven, E. A Survey of Data Mining and Machine Learning Methods for Cyber Security. IEEE Communications Surveys & Tutorials. 2021. Vol. 18, No. 2. P. 1153–1176.

2. Xin, Y., et al. Machine Learning and Deep Learning Methods for Cybersecurity. IEEE Access. 2022. Vol. 6. P. 35365–35381.

3. Gartner Magic Quadrant for Security Orchestration, Automation and Response (SOAR) Platforms. (2024). [Електронний ресурс]. Режим доступу: https://www.gartner.com/en/research

4. Goodfellow, I., Bengio, Y., Courville, A.Deep Learning. MIT Press, 2022. 800 p.

5. Корченко О.Г., Іванченко Є.В. Аналіз ризиків в системах кіберзахисту на основі нечіткої логіки та нейронних мереж. Київ: Видавництво "Політехніка", 2023. 240 с.

УДК 004.056.5

Максим ЧУХНІЙ, Надія Гавришків[2], Віктор ДЗЯДИК[2]

[1]Західноукраїнський національний університет
[2]Галицький фаховий коледж ім. В'ячеслава Чорновола

СУЧАСНІ МЕТОДИ ДОСЛІДЖЕННЯ БЕЗПЕКИ ВЕБ-ДОДАТКІВ

Вступ. Веб-додатки є ключовими інструментами для взаємодії між користувачами та інформаційними системами, що робить їх привабливою ціллю для кіберзлочинців. Забезпечення безпеки таких додатків стало критично важливим завданням для розробників та фахівців з кіберзахисту. Сучасні методи дослідження безпеки включають автоматизоване сканування вразливостей, пентестинг, аналіз коду та поведінкові моделі. Використання цих підходів дозволяє виявити та усунути потенційні загрози ще на ранніх етапах розробки.

Мета роботи: проаналізувати сучасні методи дослідження безпеки веб-додатків, оцінити їх ефективність та практичну застосовність для виявлення й усунення вразливостей з метою підвищення рівня кіберзахисту веб-систем.

1. Методи тестування

Для успішного тестування веб-застосунків необхідно застосовувати систематизований підхід або методологію. Найбільш відомі це OWASP та WASC. Вони є найбільш повними та формалізованими методологіями на сьогоднішній день.

Далі необхідно визначиться з веб-додатком - для дослідження можна взяти останню версію однієї з безкоштовних CMS, і встановити в неї вразливий плагін (вразливі версії можна завантажити з сайту exploit-db.com).

Є кілька принципів тестування, які ми можемо застосувати:

DAST - динамічний (тобто вимагає виконання) аналіз програми без доступу до вихідного коду та серверної частини, по суті BlackBox.

SAST – статичний (тобто не вимагає виконання) аналіз програми з доступом до вихідного коду веб-додатка та до веб-сервера, по суті це аналіз вихідного коду за формальними ознаками наявності вразливостей та аудит безпеки сервера.

IAST – динамічний аналіз безпеки веб-додатку, з повним доступом до вихідного коду, веб-серверу – по суті є WhiteBox тестуванням.

Аналіз вихідного коду – статичний чи динамічний аналіз із доступом до вихідного коду без доступу до серверного оточення.

Ці методи повністю підійдуть для тренування навичок виявлення вразливостей веб-програми за наявності доступу до веб-додатку, або частинок, якщо ви досліджуєте веб-додаток, наприклад, за участю в програмі BugBounty.

2. Основні етапи тестування

Для повноти тестування необхідно намагатися дотримуватися наведених нижче рекомендацій кастомізую ті чи інші етапи в залежності від веб-додатку.

Розвідка включає наступні етапи: сканування портів та піддоменів; дослідження видимого контенту; пошук прихованого контенту (директорій, файлів); визначення платформи та веб-оточення та визначення форм введення.

Контроль доступу передбачає перевірку засобів автентифікації та авторизації; визначення вимог парольної політики; проведення наступних видів тестування: підбору облікових даних, відновлення облікового запису, функцій збереження сесії, функцій ідентифікації облікового запису, перевірку повноважень та прав доступу, перевірка CSRF, а також дослідження сесії (час життя, сесійні токени, ознаки, спроби одночасної роботи і т.д.);

Фазинг параметрів включає тестування додатків до різного виду ін'єкцій (SQL, SOAP, LDAP, XPATH тощо) та тестування додатків до XSS-уразливостей. На даному етапі відбувається перевірки заголовків HTTP; редиректів та переадресацій; виконання команд ОС; локального та віддаленого включення; впровадження XML-сутностей; темплейт-ін'єкцій та взаємодії веб-сокетів.

Перевірки логіки роботи веб-програми передбачають перевірку можливості дублювання чи поділу даних а також тестування логіки роботи програми за клієнта на так званий "Стан гонки" - race condition, каналу передачі та доступності інформації, виходячи з прав доступу або його відсутності.

Перевірка серверного оточення включає:

– перевірку архітектури сервера та серверних облікових записів (служби та послуги), а також прав доступу;

– пошук та виявлення публічних уразливостей;.

– визначення параметрів сервера або компонентів (SSL тощо).

Маючи план тестування програми, ми можемо крок за кроком дослідити всі його компоненти на наявність тих чи інших вразливостей. Виходячи з веб-програми, ті чи інші пункти можуть бути доповнені специфічними для цієї програми перевірками.

Висновок. У результаті проведеного дослідження було проаналізовано основні сучасні методи дослідження безпеки веб-додатків, такі як автоматизоване сканування вразливостей, тестування на проникнення, аналіз вихідного коду та моніторинг поведінки системи. На основі отриманих даних було складено узагальнений план тестування безпеки, який може бути використаний для систематичної перевірки веб-додатків на наявність критичних вразливостей. Застосування такого плану дозволяє підвищити ефективність виявлення загроз і забезпечити більш високий рівень захисту інформаційних систем. Отже, комплексний підхід до тестування безпеки є важливою складовою сучасної практики розробки безпечного програмного забезпечення.

Перелік використаних джерел.

1. R. A. Muzaki, O. C. Briliyant, M. A. Hasditama and H. Ritchi, "Improving Security of Web-Based Application Using ModSecurity and Reverse Proxy in Web Application Firewall," 2020 International Workshop on Big Data and Information Security (IWBIS), Depok, Indonesia, 2020, pp. 85-90, doi: 10.1109/IWBIS50925.2020.9255601.

2. M. Agreindra Helmiawan, E. Firmansyah, I. Fadil, Y. Sofivan, F. Mahardika and A. Guntara, "Analysis of Web Security Using Open Web Application Security Project 10," 2020 8th International Conference on Cyber and IT Service Management (CITSM), Pangkal, Indonesia, 2020, pp. 1-5, doi: 10.1109/CITSM50537.2020.9268856.

УДК 004.056.5

Владислав БАГМЕТ[1], Віктор ДЗЯДИК[2]

[1]Західноукраїнський національний університет
[2]Галицький фаховий коледж ім. В'ячеслава Чорновола

GAME VULNERABILITIES ЯК ЗАГРОЗА КІБЕРБЕЗПЕКИ

Вступ. Баги та вразливості у комп'ютерних іграх є поширеним явищем, особливо в продуктах, що були випущені значний час тому. Така тенденція пояснюється тим, що розробники зосереджують основні ресурси на створенні нових проєктів, тоді як підтримка старих версій поступово скорочується. D наслідок цього популярні ігри з часом перетворюються на зручне середовище для дослідження та експлуатації вразливостей. До кола потенційних цілей кіберзловмисників належать як розробники, так і користувачі ігор, а в окремих випадках навіть організації, у межах яких ці користувачі працюють.

Мета: дослідити поширені вразливості у популярних продуктах гейміндустрії.

1. Загрози використання службових ПК з ігровим ПЗ

Комп'ютерна гра є програмним продуктом, тому може містити помилки в реалізації чи пропуски під час тестування. Наявність таких дефектів підтверджується публічними реєстрами (CVE), які фіксують відомі вразливості та їхню класифікацію за рівнем загрози. З метою дослідження цієї проблеми було проведено аналіз даних агрегатора вразливостей і перевірено інформацію щодо окремих проєктів на платформі Steam. Для прикладу було відібрано CVE, пов'язані з клієнтом гри Dota 2, середній рейтинг тяжкості яких за шкалою CVSS становив 7,8 із 10 [1].

Серед виявлених інцидентів, найсерйозніша була зафіксована у 2023 році внаслідок дослідження, проведеного фахівцями Avast: клієнт Dota 2 використовував застарілу версію рушія JavaScript (V8), яка містила вразливість, що дозволяла виконувати небажаний JavaScript-код на машині користувача. Наслідком дефектів може бути несанкціоноване виконання коду на комп'ютері жертви та повна компрометація її середовища.

Подібні ситуації відзначалися й у інших популярних проєктах. Так, для серії Counter-Strike виявлено кілька CVE із середнім значенням CVSS близько 7,76, а у грудні 2023 року було продемонстровано реалізацію XSS-вектору в контексті нової функції додавання зображень у чаті, що створювало можливості цілеспрямованих атак на користувачів. У випадку GTA Online дефект, позначений як CVE-2023-24059, було виявлено та виправлено на початку 2023 року. Цей збій дозволяв не лише отримати дані облікових записів, а й розміщувати шкідливе програмне забезпечення (ПЗ)на пристроях жертв.

Слід підкреслити, що запис у реєстрі CVE відображає лише вразливості, які вже стали загальновідомими і, як правило, були усунуті та опубліковані. Отже, загальна кількість дефектів у продуктах ігрової індустрії, ймовірно, значно перевищує кількість задокументованих CVE. Додатково практикою є те, що постачальники ПЗ іноді мають відомості про певні вразливості, але затримують їх

усунення через пріоритети розробки або інші операційні причини. Прикладом є ситуація з проєктом Call of Duty: Black Ops III (реліз 2015 року), де було задокументовано повідомлення про RCE-вразливості, які залишалися невиправленими тривалий час. В наслідок відсутності офіційних виправлень частина спільноти розробників-ентузіастів змушена була створювати власні модифікації та виправлення, доступні у відкритих репозиторіях.

Отже, із встановленим ігровим ПЗ з'являється низка ризиків: від локальних компрометацій окремих робочих станцій до потенційного розповсюдження загроз у внутрішній корпоративній мережі. Це обґрунтовує необхідність оцінки ризиків використання ігрового ПЗ на службі та запровадження політик контролю встановлення й оновлення програмного забезпечення, а також засобів виявлення й реагування на інциденти кібербезпеки.

2. Основні загрози ігровим акантам

Переважна більшість інцидентів із компрометацією облікових записів геймерів мають соціо-інженерний характер, причому найпоширенішим інструментом виступає фішинг. Атаки такого типу організовуються через чати, тематичні форуми та інші майданчики спільнот; їхня популярність пояснюється простотою реалізації та низькими витратами для зловмисника. Типовим прикладом є шахрайські схеми «я продав тобі, але ти не заплатив», які спрямовані на отримання облікових даних або доступу до платіжних інструментів жертви.

Технічно складніші вектори, що використовують програмні вразливості ігрових клієнтів або плагінів, зустрічаються рідше, оскільки вимагають відповідних навичок і часу на розробку експлойтів. Водночас такі атакувальні сценарії не є поодинокими і використання вразливостей може призводити до віддаленого виконання коду, підміни сесійних токенів або похищення облікових даних без прямого залучення користувача. Наслідком експлуатації вразливостей найчастіше стає крадіжка акаунтів, що, з огляду на розвиток внутрішньоігрових ринків, може мати значну матеріальну шкоду. Прикладом є інцидент 2022 року з відомим колекціонером предметів у грі серії Counter-Strike, в результаті якого було втрачене право розпоряджатися скінами загальною вартістю близько мільйонів доларів.

Значну загрозу також становить використання модифікованих або нелегально отриманих інсталяційних образів. Піратський софт, поширюваний через торренти й подібні ресурси, часто постачається разом із бекдорами та іншими типами шкідливого ПЗ, що робить його джерелом компрометації кінцевих пристроїв. У багатьох випадках саме завантаження та запуск модифікованих інсталяторів стають початковою точкою проникнення.

Окрему категорію ризиків формують користувацькі практики, які навмисно або мимоволі знижують рівень захисту кінцевої системи. Частина гравців вимикає антивірусні рішення або брандмауери, посилаючись на їх вплив на продуктивність або сумісність із грою. Інші користувачі взагалі відмовляються від активних засобів захисту. Ефективність таких налаштувань у сенсі підвищення продуктивності є предметом дискусій, натомість їхній внесок у підвищення вразливості системи виявлена й документована, зокрема зниження рівня захисту істотно збільшує ймовірність успішної компрометації облікових

записів і пристроїв.

Загрози ігровим акаунтам мають багатовимірний характер, від простих соціо-інженерних схем до технічно складних експлуатацій вразливостей і постачання шкідливого ПЗ разом із піратським контентом.Ускладнює ситуацію також поведінка користувачів, що іноді свідомо знижують заходи захисту. Це підкреслює необхідність комплексного підходу - поєднання технічних засобів, освітніх ініціатив для спільнот і проактивного моніторингу інцидентів.

3. Загрози додаткового ПЗ

Багато загроз також несуть у собі програми, які геймери активно використовують крім ігор. Вони теж можуть містити критичні дефекти безпеки, як це показано в таблиці 1.

Таблиця 1 – Вразливості ігрових майданчиків та спеціального ПЗ

Steam та інші майданчики для розміщення ігор	Особливо небезпечна вразливість у Steam була виявлена у 2020 році. Використовуючи її, зловмисник міг захопити сотні тисяч комп'ютерів, не вимагаючи від геймерів натискати на шкідливий лист або посилання. На відміну від інших уразливостей, жертви несвідомо траплялися під вплив хакера. Для цього їм потрібно було просто увійти до гри. У 2023 році зловмисники зламали облікові записи сотні розробників на платформі Steam і додали до їхніх ігор шкідливе ПЗ. Але вендор швидко виявив проблему і повідомив про це користувачам.
Discord та інші утиліти для спілкування з командою	Повідомлень про проблеми у продукті чимало. Наприклад, минулого року розробник визнав витік даних 760 тис. користувачів, яка сталася з вини співробітника.
GeForce Experience, OBS Studio та інші програми для запису відео, оцінки FPS тощо.	У 2020 році розробник GeForce Experience залатав відразу дві серйозні дірки. Одна з уразливостей (CVE-2020-5977) отримала CVSS 8,2 і могла призвести до безлічі шкідливих атак на порушені системи, включаючи виконання коду, відмову в обслуговуванні, підвищення привілеїв та розкриття інформації.
AutoHotKey та аналоги для налаштування кнопок клавіатури та миші	З його допомогою злочинці поширювали трояни для віддаленого доступу до пристроїв жертв, у тому числі Revenge RAT, LimeRAT, AsyncRAT, Houdini та Vjw0rm.
Spotify та інші сервіси для прослуховування музики під час гри	У 2020 році через виток даних Spotify скинув 350 тис. паролів користувачів. Хоча в офіційній заяві власник продукту повідомив, що проблема торкнулася лише невеликої частини акаунтів.

CVE-2020-5977 - уразливість типу «небезпечний/неконтрольований пошук шляху завантаження модулів» (untrusted search path) в компоненті, що

використовує середовище виконання Node.js. Через цю слабину процес, який завантажує модулі Node.js (через require() / import), може підхопити шкідливий модуль із непередбачуваного або керованого зловмисником каталогу. Унаслідок цього можливе виконання довільного коду в контексті вразливого процесу, що може призвести до локальної компрометації системи.

Схема вразливості CVE-2020-5977 приведена на рисунку 1.

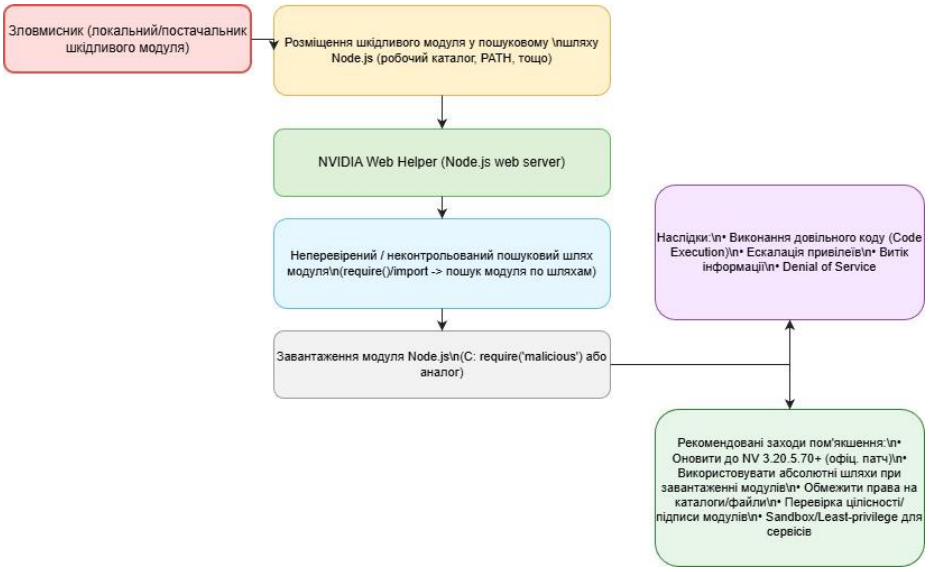


Рисунок 1 - Схема вразливості CVE-2020-5977

CVE-2020-5977 належить до класу вразливостей, які використовують недоліки в управлінні шляхами завантаження модулів у середовищах виконання, таких як Node.js. Механізм експлуатації є концептуально простим, проте практичні наслідки можуть бути критичними. Найефективнішими заходами захисту є своєчасне оновлення ПЗ, управління правами доступу, використання перевірок цілісності модулів та впровадження контролю завантажених компонентів у рантаймі.

Висновок. Безпека відеоігор є багатовимірною проблемою, що поєднує технічні вразливості програмного коду, ризики соціальної інженерії та операційні загрози, пов'язані з практиками розповсюдження та використання ПЗ. Наслідки експлуатації вразливостей охоплюють як індивідуальні втрати користувачів, так і бізнес-ризики для розробників та їхньої інфраструктури. Ефективна протидія потребує комбінованого підходу, що включає безпечну розробку, регулярне патчування, контроль цілісності файлів і жорстке управління правами доступу. Додатково необхідні механізми та інструменти моніторингу (EDR, HIDS, SIEM), а також просвітницькі заходи для користувацької спільноти щодо фішингу та безпечних практик. Лише системне поєднання технічних, організаційних і правових заходів здатне істотно знизити експлуатаційний ризик і підтримати довгострокову стійкість ігрової екосистеми.

Перелік використаних джерел.

1. CVE-2015-7985. [Електронний ресурс]. - Режим доступу: https://nvd.nist.gov/vuln/detail/CVE-2015-7985.

2. CVE-2020-5977. [Електронний ресурс]. - Режим доступу: https://nvd.nist.gov/vuln/detail/CVE-2020-5977.

УДК 004.421

Василь ПОМАЗИБІДА, Сергій КУЛИНА

Західноукраїнський національний університет

АЛГОРИТМИ ГОМОМОРФНОГО ШИФРУВАННЯ ДЛЯ БЕЗПЕЧНИХ ХМАРНИХ ОБЧИСЛЕНЬ

Вступ. У сучасному цифровому світі хмарні обчислення стали фундаментальною технологією для зберігання та обробки величезних масивів даних. Однак передача чутливої інформації - як-от медичні записи, фінансові дані чи комерційна таємниця - стороннім провайдерам створює значні ризики для безпеки та конфіденційності. Традиційне шифрування захищає дані під час зберігання та передачі, але вимагає їх розшифрування для будь-якої обробки, роблячи їх вразливими у хмарному середовищі.

Саме тому дослідження алгоритмів гомоморфного шифрування є надзвичайно важливим, оскільки вони дозволяють виконувати обчислення (наприклад, аналіз чи машинне навчання) безпосередньо над зашифрованими даними. Це дає змогу використовувати потужні ресурси хмари для обробки інформації, не розкриваючи при цьому самі дані, та є ключем до побудови справді безпечних хмарних сервісів, що гарантують повну конфіденційність.

Метою дослідження є підвищення рівня безпеки та ефективності обробки конфіденційних даних шляхом розробки або вдосконалення алгоритмів гомоморфного шифрування, що дозволяють збалансувати високий рівень захисту інформації з прийнятними обчислювальними витратами.

1. Дослідження існуючих алгоритмів гомоморфного шифрування

Дослідження існуючих алгоритмів гомоморфного шифрування зазвичай починається з їх класифікації за підтримуваними операціями та рівнем складності. Історично першими з'явилися частково гомоморфні системи (PHE), такі як Paillier (дозволяє необмежену кількість операцій додавання) або "чистий" RSA (дозволяє необмежену кількість операцій множення). Вони є обчислювально ефективними, але їхня функціональність обмежена вузькоспеціалізованими завданнями, як-от безпечне голосування чи агрегація статистичних даних. Наступним кроком стали дещо гомоморфні схеми (SHE), які підтримують обмежену кількість і додавань, і множень. Їхня головна проблема — неконтрольоване "зашумлення" даних: кожна операція збільшує "шум" у шифротексті, і після досягнення певного, заздалегідь визначеного порогу, дані стають неможливими для коректного розшифрування.

Сучасні дослідження зосереджені переважно на повністю гомоморфних (FHE) схемах, які вирішують проблему "шуму" за допомогою ресурсоємної операції "перешифрування" (bootstrapping). На практиці домінують кілька ключових "родин" алгоритмів, що базуються на проблемі складності (Ring) LWE. Схеми, як-от BGV та BFV, чудово підходять для точних обчислень над цілими числами (арифметичні схеми), що корисно для запитів до баз даних. З іншого боку, схема CKKS стала де-факто стандартом для прикладного машинного навчання та аналізу даних, оскільки вона працює з приблизними (дійсними) числами. Окремо стоять схеми TFHE/FHEW, оптимізовані для надшвидкого

перешифрування, що робить їх ідеальними для оцінки булевих схем. Таким чином, фундаментальна проблема, яку аналізує дослідження, — це компроміс: не існує єдиного "найкращого" алгоритму, і вибір (BGV, BFV, CKKS чи TFHE) залежить від конкретного хмарного завдання, при цьому всі вони все ще мають значні накладні витрати на продуктивність та розмір даних.

2. Розробка алгоритму гомоморфного шифрування

Варто зазначити, що гомоморфне шифрування — це не один конкретний алгоритм, а радше криптографічний протокол, що описує взаємодію між власником даних та обчислювальним середовищем. Розглянемо загальні кроки цього процесу на прикладі однієї з сучасних FHE-схем (як-от BFV або CKKS). Цей процес завжди включає щонайменше двох учасників: Клієнта (який володіє даними та секретним ключем) і Сервера (який виконує обчислення, маючи лише публічний ключ). Етапи роботи гомоморфного протоколу:

Крок 1. Ініціалізація та генерація ключів. На цьому етапі клієнт готує криптографічну систему та обирає набір криптографічних параметрів (наприклад, ступінь поліноміального кільця N, розмір модулів q і t). Від цих параметрів залежить рівень безпеки та "глибина" обчислень (скільки операцій можна виконати до того, як "шум" стане занадто великим).

Клієнт генерує три типи ключів:

– Секретний ключ (SK) - це приватний ключ, який клієнт нікому ніколи не передає. Він єдиний, що здатен розшифрувати дані.

– Публічний ключ (PK), який використовується для шифрування даних.

– Ключі оцінки (Evaluation Keys) - це спеціальний набір публічних даних (наприклад, "ключі перелінеаризації"), які необхідні Серверу для виконання складних операцій, найчастіше — множення шифротекстів.

Після чого клієнт надсилає Публічний ключ та Ключі оцінки на хмарний Сервер. Секретний ключ залишається у Клієнта.

Крок 2. Шифрування та передача даних.

Клієнт бере свої конфіденційні дані. Залежно від схеми (наприклад, BFV чи CKKS), дані кодуються у спеціальний формат (наприклад, у поліноми). Використовуючи свій Публічний ключ, Клієнт шифрує підготовлені дані та отримує шифротекст.

Клієнт надсилає зашифровані дані на сервер для зберігання та обробки.

Крок 3. Гомоморфна оцінка.

Сервер не має Секретного ключа і не може бачити дані. Сервер отримує від Клієнта (або авторизованого користувача) програму, яку потрібно виконати над даними. Ця програма має бути представлена у вигляді арифметичної схеми (послідовності операцій додавання та множення).

Сервер використовує властивості схеми для виконання обчислень. Кожна операція (особливо множення) додає до шифротексту "шум". Якщо "шум" перевищить певний поріг, дані буде неможливо розшифрувати. Сервер використовує Ключі оцінки для керування цим шумом (наприклад, перелінеаризація після множення).

Якщо програма дуже складна і "шум" стає критичним (у FHE-схемах), Сервер може виконати процедуру bootstrapping. Це операція, яка гомоморфно

"розшифровує" і "зашифровує" дані заново, використовуючи надані Клієнтом ключі. Вона "очищує" шифротекст від шуму, дозволяючи виконувати необмежену кількість операцій.

Виконавши всю програму, Сервер отримує кінцевий шифротекст.

Крок 4. Розшифрування результату.

Сервер надсилає зашифрований результат назад Клієнту.

Клієнт використовує свій Секретний ключ (SK) для розшифрування.

В результаті Клієнт отримує фінальний відкритий текст, який дорівнює результату обчислень, так, ніби вони виконувались на його власних незашифрованих даних.

Ефективність алгоритмів гомоморфного шифрування є головним компромісом та ключовим викликом для їхнього практичного впровадження. Хоча вони пропонують теоретично ідеальний рівень безпеки, дозволяючи обчислення на зашифрованих даних, їхня практична ефективність наразі залишається низькою порівняно з традиційною обробкою у відкритому вигляді. Це виражається у трьох основних аспектах: обчислювальна складність, роздуття даних та складність реалізації.

Висновок. На сучасному етапі розвитку, повністю гомоморфне шифрування не є ефективним для загальноцільових хмарних обчислень. Воно залишається нішевою технологією, ефективність якої є прийнятною лише для вузькоспеціалізованих завдань, де вимоги до конфіденційності є абсолютними і переважають будь-які витрати на продуктивність. Це можуть бути, наприклад, прості статистичні запити до медичних баз даних, приватне об'єднання фінансових наборів даних або виконання простих моделей машинного навчання.

Таким чином, головний напрямок досліджень сьогодні — це не стільки розробка нових криптографічних схем, скільки оптимізація існуючих: створення спеціалізованого апаратного забезпечення (FPGA, ASIC) для прискорення гомоморфних операцій та розробка кращих компіляторів і бібліотек, які б автоматизували складний процес оптимізації програм під FHE.

Перелік використаних джерел.

1. Dunuwila R. M. T. R., Senevirathne T., Weerasinghe P., Kankanamge C. A Comprehensive Survey on Fully Homomorphic Encryption for Cloud Computing: A Practical Perspective. IEEE Access. 2022. Vol. 10. P. 111494–111521.

2. Al-Khafaji S. A. G. G. N., Al-Naji A., Mohammed A. H. A survey of privacy-preserving SQL queries using homomorphic encryption in cloud databases. Journal of Cloud Computing. 2023. Vol. 12. Art. 48. DOI: 10.1186/s13677-023-00478-4.

3. Chillotti I., Gama N., Gkoulalas-Divanis A., Poly F. Faster fully homomorphic encryption: Bootstrapping in less than 0.1 seconds. Journal of Cryptology. 2021. Vol. 34. № 4. Art. 28. DOI: 10.1007/s00145-021-09404-8.

4. Ghani M. K. A., Yusoff Z., Yunus M. A. M., Ariffin A. Homomorphic encryption in cloud computing: challenges and future directions. Multimedia Tools and Applications. 2023. Vol. 82. P. 3673–3697. DOI: 10.1007/s11042-022-13237-y.

5. Chen H., Ma M., Cui T., Han M., Wang Y. Efficient privacy-preserving machine learning framework in cloud computing using homomorphic encryption. Information Sciences. 2022. Vol. 609. P. 1007–1020. DOI: 10.1016/j.ins.2022.07.126.

УДК 004.056.5

Соколов А.В., Кілко В.В.

Національний університет «Одеська політехніка»

ОЦІНКА СТІЙКОСТІ СТЕГАНОГРАФІЧНОГО МЕТОДУ З КОДОВИМ УПРАВЛІННЯМ ДЛЯ РІЗНИХ КЛАСІВ КОНТЕЙНЕРІВ

Вступ. У статті розглянуто вплив вибору кодового слова на надійність сприйняття та стійкість стеганографічного вбудовування для зображень різних типів контейнерів. Актуальність визначається необхідністю забезпечення збереження прихованої інформації за умов типових впливів, насамперед JPEG-стиску, та прагненням зменшити обчислювальні витрати шляхом керування процесом приховування не переходом до інших областей представлення даних, а за рахунок параметрів кодових структур (кодовим управлінням).

Мета: Проаналізувати, як вибір частотного профілю кодового слова впливає на точність відновлення прихованих даних після JPEG-стиску в контейнерах різної текстурованості, та сформулювати рекомендації для адаптивного добору кодового слова залежно від умов стиску і властивостей контейнера.

Основна частина

Експерименти проведено на вибірці з 500 PNG-зображень, класифікованих на чотири групи за рівнем текстурованості (гладкі, середньотекстуровані, високодеталізовані та змішані). Розглядалися шість типів кодових слів: Const (постійне), LF (низькочастотне), LF-C (комбіноване низькочастотне), MF (середньочастотне), HF (високочастотне) та Bent (на основі бент-функції). Для кожного контейнера виконувалося вбудовування однакового обсягу даних, після чого здійснювали JPEG-стиск з рівнями якості QF=10…100 і відновлення повідомлення. Ефективність оцінювали за відсотком бітових помилок відновлення.

Кодові слова формувалися на базі функцій Уолша з використанням перетворення Уолша–Адамара. Бент-кодове слово утворене з максимально невзаємнокорельованих булевих функцій, що забезпечує рівномірний розподіл енергії по трансформантах і практичну ізотропність впливу на частотну область. Такий підхід дозволяє керувати компромісом між надійністю сприйняття та стійкістю без переходу до складних перетворень (DCT, DWT тощо) і знижує обчислювальну складність. Результати та обговорення:

– Вибір кодового слова має вирішальний вплив на стійкість прихованої інформації; тип контейнера впливає відчутно, але меншою мірою.

– Для QF > 20 найменший відсоток бітових помилок у більшості груп забезпечує низькочастотне кодове слово (LF).

– Для жорсткого стиску (QF ≤ 20) найкращі результати забезпечує постійне кодове слово (Const).

Таблиця 1 – Кодові слова, що брали участь в експерименті і їх перетворення Уолша-Адамара

Кодове слово, що брало участь в експерименті	Перетворення Уолша-Адамара
$Const = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$	$W_{Const} = \begin{bmatrix} 64 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
$LF = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \end{bmatrix}$	$W_{LF} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 64 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
$LFC = \begin{bmatrix} 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \end{bmatrix}$	$W_{LFC} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 64 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
$MF = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$	$W_{MF} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 64 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
$HF = \begin{bmatrix} 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 & -1 & 1 & -1 & 1 \end{bmatrix}$	$W_{HF} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 64 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$
$Bent = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 & 1 & -1 & -1 & 1 \\ -1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 \\ -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix}$	$W_{Bent} = \begin{bmatrix} 8 & 8 & 8 & 8 & 8 & 8 & -8 & -8 \\ 8 & -8 & 8 & -8 & 8 & -8 & -8 & 8 \\ 8 & 8 & 8 & 8 & 8 & 8 & -8 & -8 \\ 8 & -8 & 8 & -8 & 8 & -8 & -8 & 8 \\ 8 & 8 & 8 & 8 & 8 & 8 & -8 & -8 \\ 8 & -8 & 8 & -8 & 8 & -8 & -8 & 8 \\ -8 & -8 & -8 & -8 & -8 & -8 & 8 & 8 \\ -8 & 8 & -8 & 8 & -8 & 8 & 8 & -8 \end{bmatrix}$

– Високочастотне кодове слово (HF) забезпечує мінімальні візуальні спотворення, але практично не зберігає дані після стиснення.

– Bent-кодове слово демонструє найменший розкид помилок між різними класами зображень і стабільне зменшення помилок зі зростанням QF, що підтверджує його універсальність.

– Гладкі зображення краще зберігають вбудовані дані для LF/MF; у високодеталізованих контейнерах LF і MF також переважають HF за стійкістю; для змішаних зображень Bent показує найбільш передбачувану поведінку.

Практичні рекомендації для кодового управління:

– Очікуваний стиск слабкий/помірний (QF > 20): застосовувати LF або LF-C.

– Очікуваний жорсткий стиск (QF ≤ 20): застосовувати Const.

– Нейтральність до типу контейнера (різнорідні набори): застосовувати Bent як універсальний варіант або для калібрування системи.

– Пріоритет максимальної візуальної якості за відсутності атак: допустиме використання HF.

Висновок. Експериментально показано, що правильний вибір частотного профілю кодового слова забезпечує суттєве підвищення стійкості стеганоповідомлення до JPEG-стиску в різних класах контейнерів. Отримані закономірності покладено в основу адаптивних правил добору кодових слів для систем із кодовим управлінням, що автоматично враховують властивості контейнера та очікуваний рівень стиску.

Перелік використаних джерел.

1. Chinnusami M. et al. Scientific Reports. 2025. 15(1):31610.

2. Kumar N. N., Viswanathan R., Kumar P. S. ICSCSA, IEEE. 2024. 479–486.

3. Mandal P. C., Mukherjee I., Chatterji B. N. Multimedia Tools and Applications. 2024. 83(23):62651–62675.

4. Nagini R. V. S. S. S. et al. AIP Conf. Proc. 2025. 3263(1):150001.

5. Apau R. et al. PLoS One. 2024. 19(9):e0308807.

6. Angulakshmi M., Deepa M. IGI Global. 2025. 53–74.

7. Kobozeva A.A., Sokolov A.V. Problemele energeticii regionale. 2021. (4)52:115–130.

8. Кобозєва А.А., Соколов А.В. Вісті ВНЗ. Радіоелектроніка. 2023. 66(4):205–222.

9. Кілко В.В., Соколов А.В., Баландіна Н.М. Кібербезпека та комп'ютерно-інтегровані технології. 2024. 110–114.

10. Sokolov A.V., Ihnatenko O.O., Balandina N.M. Problems of regional energetics. 2024. 62(2):121–137.

11. Rothaus O. S. Journal of Combinatorial Theory, Ser. A. 1976. 20(3):300–305.

12. Sokolov A.V., Tsevukh I.V. JTEC. 2018. 10(2):51–54.

УДК 004.056.5

Борисенко І.І., Дідик Є.Ю.

Національний університет «Одеська політехніка»

СТЕГАНОГРАФІЧНА СИСТЕМА КОНТРОЛЮ РОЗМІЩЕННЯ ПОВІДОМЛЕННЯ В КОНТЕЙНЕРІ

Вступ. У сучасному світі теорія графів є однією з актуальних та ефективних, серед математичних технологій, бо сфера її застосування охоплює різні області діяльності людства, зокрема у інформаційній та кібернетичній безпеці. Аналіз наукової літератури [1] свідчить про наявність глибокої зацікавленості вчених до проблеми використання графових технологій у кібербезпеці. Сформувалися наступні напрями застосування теорії графів в інформаційній та кібернетичній безпеці: в інформаційній системі та у програмуванні; моделювання, аналіз та застосування графів атак; криптографічні перетворення за допомогою теорії графів; побудова дерева рішень у задачах прийняття рішень в умовах ризику і невизначеності. Теорія графів знайшла своє застосування і в стеганографії, оскільки завдяки графам можна представити структуру контейнера, повідомлення, яке вбудовується, та вирішити безліч задач.

Мета: Модифікація стеганографічного алгоритму засобами теорії графів.

Основна частина.

Розглянемо як можна застосувати розвинуту в стеганографії теорію графів до алгоритмів, які вже мають практичне застосування, тобто як і надалі можна розвивати практичну теорію графів.

В роботі [1] пропонується новий стеганографічний алгоритм просторової області вбудовування в цифрове зображення. Основним принципом розробки є мінімізація впливів вбудованого повідомлення на контейнер. В основу алгоритму покладено порівняння бітових послідовностей контейнера та повідомлення, модифікація елементів контейнера виконується тільки у випадку, коли виявлено неспівпадіння відповідних бітів. Алгоритм дозволяє зменшити викривлення контейнера, зберегти статистики першого порядку та забезпечити стійкість до найбільш відомих статистичних атак.

Повідомлення і пікселі контейнера розбиваються на підпослідовності. Початок підпослідовностей, в які вбудовується повідомлення фіксуються в ключі К. Але саме цю задачу можна вирішити за допомогою графа. Що ефективніше використовувати ключ чи граф, це окрема задача. Зараз розглянемо, саме яким чином можна фіксувати, за допомогою графа, підпослідовності контейнера, в які вбудована інформація, яку треба передати адресату. Окрім цього, пропонується дублювати інформацію, яку треба переслати. За рахунок клонування відліків інформації, що вбудовується, алгоритм підвищить свою стійкість не тільки до статистичних але і до інших видів атак таких як, наприклад, зашумлення стегоконтейнера, а в окремих випадках до геометричних атак, таких як поворот та обрізання.

Вузли графа – це початки підпослідовностей контейнера, в які вбудоване повідомлення, ребра – показують, з якої вершини графа потрібно переміститись в

іншу, а саме ту вершину, щоб одержати зв'язне повідомлення.

Оскільки є клони кожного відліку повідомлення, то граф буде представляти собою не ланцюг, а дерево (рисунок 1).

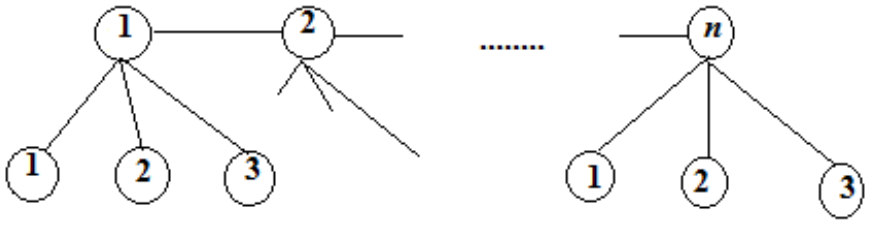


Рисунок 1. – Граф-дерево розміщення повідомлення

При декодуванні повідомлення, якщо ланцюг графа 1-2…n не дає зв'язного тексту (шум в каналі зв'язку або навмисно накладений шум як атака на стегоконтейнер, інші види атак), то є можливість використати листи графа 11, 12 або 13 і так далі n1, n2, n3. Дублювання окремих блоків інформації дещо перевантажує контейнер, але дає можливість протистояти атакам, які вносять невеликі збурення в контейнер,наприклад, накладання шуму, який непомітний або ледь помітний оку. Характеристики алгоритмів, що оперують із графами, зазвичай дуже чутливі до способу їх представлення.

Відомі схеми представлення графів [1]. Однією з найбільш простих схем зберігання графа є таблиця зв'язків - двовимірний масив, який має n рядків і m стовпців, де m - максимальна степінь вершин в G=(X,E). Список суміжності i-го вузла зберігається в i-ому рядку.

Дана схема зберігання надзвичайно проста при реалізації, доступ до списку суміжності чергового вузла - доступ до відповідного рядка матриці, модифікація графа приводить до зміни елементів відповідних рядків матриці без порушення загальної структури (якщо при модифікації не змінюється m). Однак ця схема може бути надзвичайно неефективна, якщо велика кількість вузлів графа має степінь, меншу (значно), ніж максимальна, оскільки її вимоги до пам'яті визначаються як mn «збережених» елементів.Найбільш зручною з погляду можливостей проведення модифікацій графа є схема, що використовує поле зв'язків.

Дана схема містить три одновимірні масиви A, A_s, A_{ind}, перші два з яких мають довжини 2|E|, останній - |X|. Значенням покажчика $A_{ind}(i)$ є початок списку суміжності i-го вузла в масиві A. Якщо A(k)- це черговий сусід i-го вузла, то $A_s(k)$ - покажчик розташування наступного його сусіда в масиві A^A. Від'ємне значення $A_s(k)$ говорить про закінчення списку суміжності вузла, що розглядається.

Загальна довжина масивів при такому способі представлення графа – 4|E|+|X|, що значно більше, ніж у першій схемі. Однак модифікація графа вимагає лише незначних змін у вже сформованій частині масивів.

Перелік використаних джерел

1. Борисенко І.І. Застосування методів порівняння послідовностей в стеганографічних перетвореннях цифрових зображень. Сучасна спеціальна техніка. Київ, 2014. №2. С. 110-115.

2. Нікольський Ю.В, Пасічник В.В, Щербина Ю.М. Дискретна математика. – К.: Видавнича група BHV, 2007. – 368 с.

УДК 004.056

Вадим ЛОГОШ, Дмитро СМІРНОВ, Роман ХОМЯК

Західноукраїнський національний університет

ПОПУЛЯРНІ БІБЛІОТЕКИ ТА ФРЕЙМВОРКИ ГОМОМОРФНОГО ШИФРУВАННЯ

Вступ. Гомоморфне шифрування дозволяє проводити обчислення безпосередньо над зашифрованими даними. Це дає змогу стороннім сервісам, наприклад, хмарним платформам, обробляти конфіденційну інформацію, не розшифровуючи її. Результат обчислень також залишається зашифрованим і доступним лише власнику секретного ключа, що усуває ризик компрометації даних під час обробки.

Гомоморфні криптосистеми поділяються на частково гомоморфні (PHE), обмежено гомоморфні (SHE/LHE) та повністю гомоморфні (FHE) схеми. Частково гомоморфні забезпечують виконання лише одного типу математичної операції над шифротекстами (лише додавання *або* лише множення). Обмежено гомоморфні дозволяють обмежену кількість як додавань, так і множень (наприклад, довільна кількість додавань і лише одне множення), тобто підтримують обидва типи операцій, але з обмеженою глибиною складання операцій.

Повністю гомоморфні шифрування не мають таких обмежень – вони дозволяють виконувати будь-які обчислення (довільні схеми з додаваннями і множеннями) над зашифрованими даними необмежену кількість разів. FHE-системи, по суті, забезпечують можливість реалізувати на шифротекстах довільну функцію (вони є тюринг-повними), тоді як PHE та обмежені SHE – ні.

Мета. Аналіз сучасних бібліотек та фреймворків гомоморфного шифрування та порівняння їх основних характеристик.

1. Аналіз бібліотек та фреймворків гомоморфного шифрування

Сьогодні існує ряд відкритих бібліотек, які реалізують схеми і надають розробникам зручні інструменти для роботи з гомоморфним шифруванням. Нижче проаналізовано найпоширеніші з них та дано коротку характеристику.

1. Microsoft SEAL. Популярна відкрита бібліотека від Microsoft, що підтримує схеми BFV (Brakerski/Fan–Vercaut.) та CKKS(Cheon -Kim et al.). Орієнтована на простоту використання: надає високорівневий API для виконання гомоморфних обчислень, дозволяючи будувати повністю зашифровані сховища даних і сервіси обробки без розкриття ключів [1].

Написана на C++ (доступні обгортки для .NET, Python), оптимізована для швидкодії, має детальну документацію і приклади.

2. PALISADE. Бібліотека з відкритим кодом, розроблена консорціумом за підтримки DARPA. Підтримує кілька гомоморфних схем – BGV, BFV, CKKS, а також бульові TFHE/FHEW, у тому числі в багатокористувацькому (Multiparty) режимі. Відрізняється модульною архітектурою і гнучкістю налаштувань. На основі PALISADE у 2022 р. створено нову об'єднану платформу OpenFHE.

3. Helib. Одна з перших FHE-бібліотек, розроблена IBM (перший випуск – 2013/2014, публічний реліз – 2016 р.). Реалізує схеми BGV і CKKS (додані

пізніше) та підтримує bootstrapping для BGV. Написана на C++ з відкритим кодом, HElib була націлена передусім на дослідників, надаючи гнучкий, хоч і відносно низькорівневий інтерфейс. IBM продовжує вдосконалювати HElib, зокрема оптимізуючи швидкодію.

4. Бібліотека TFHE. Спеціалізована бібліотека для схеми TFHE (Torus FHE). Розроблена командою дослідників (Chillotti, Gama, Georgieva, Izabachène) і вперше опублікована у 2016–2017 рр. Орієнтована на швидкі булеві операції з частим bootstrapping. Забезпечує виконання логічних схем (AND, OR, XOR тощо) над шифрованими бітами за кілька мілісекунд. Реалізована на C++, використовує швидку операцію БПФ над тором і інші оптимізації. Підтримує також багатокористувацький режим. Недоліком є обмеженість до побітових операцій – для роботи з великими числами чи векторами рекомендується комбінувати її з іншими бібліотеками (або використовувати гібридні підходи).

5. HEAAN (HeaAn). Відкрита бібліотека для схеми CKKS, розроблена дослідниками Сеульського національного університету (авторами CKKS). Назва розшифровується як *"*Homomorphic Encryption for Arithmetic of Approximate Numbers*"*. Перший випуск – 2016 р., згодом підтримку проекту продовжила корейська компанія CryptoLab [3].

HEAAN реалізує всі основні можливості CKKS: гомоморфні додавання, множення, масштабування, пакування/розпакування векторів. В ній однією з перших з'явилася реалізація bootstrapping для CKKS, що дозволяє виконувати необмежену кількість операцій на зашифрованих речових числах. HEAAN оптимізована для високої точності і швидкості обчислень, підтримує GPU-акселерацію для основних операцій над поліномами. Її часто використовують у дослідженнях, пов'язаних з приватними обчисленнями в AI, оскільки вона швидко впроваджує найновіші алгоритмічні вдосконалення CKKS.

6. OpenFHE. новітній фреймворк (перший реліз – липень 2022) для гомоморфного шифрування, який об'єднує напрацювання кількох попередніх бібліотек [4].

OpenFHE розроблено командою експертів з різних установ під егідою організації Duality Technologies, за участі спільноти (проект під патронатом NumFocus). Бібліотека є наступником PALISADE і включає в себе підтримку всіх основних схем: BGV, BFV, CKKS для арифметичних обчислень, а також схем TFHE і FHEW для булевих операцій. OpenFHE від початку спроектована з урахуванням можливості *bootstrapping* для всіх схем (тобто підтримує перезавантаження і для BGV/BFV/CKKS, чого раніше не було «з коробки») [4].

Великі технологічні компанії вкладаються в розвиток FHE, розробляючи інструменти для спрощення його використання. Наприклад, Microsoft випустила бібліотеку SEAL, яка допомагає інтегрувати гомоморфне шифрування у прикладні рішення (вже реалізовані пілотні проекти з повністю зашифрованого аналізу даних у партнерстві з фінтех-компаніями). Google розробила інструмент Private Join and Compute для захищеного спільного аналізу даних, а також FHE Transpiler – компілятор, що перетворює звичайний код на еквівалентні гомоморфні обчислення [5].

IBM активно працює над прискоренням HElib і пропонує хмарні сервіси з підтримкою FHE. Отже, є підстави вважати, що повністю гомоморфне

шифрування поступово виходитиме за межі дослідницьких лабораторій і інтегруватиметься у реальні системи, забезпечуючи новий рівень безпеки даних.

В таблиці 1 наведено основні гомоморфні схеми шифрування – як часткові (PHE), так і повні (FHE). Порівнюються їх тип, підтримувані гомоморфні операції, криптографічна основа, стійкість та рік появи.

Таблиця 1.1 – Основні гомоморфні схеми шифрування

Схема	Тип шифрування	Підтримувані операції	Безпека (основа)	Рік
RSA	PHE (мультиплікативна)	Множення шифротекстів (без обмежень)	Факторизація цілого *n* (не стійка проти квантових атак)	1978
Paillier	PHE (адитивна)	Додавання шифротекстів; множення на константу	Композитний модуль (n^2); (не постквантова)	1999
Gentry FHE	FHE (повна схема)	Додавання, множення (необмежено завдяки bootstrapping)	Ідеальні ґратки (LWE) + підмножина суми; постквантова	2009
BGV	FHE (на ґратках)	Додавання, множення (рівнева або з bootstrap)	RLWE (кільцеві ґратки); постквантова	2011
BFV	FHE (на ґратках)	Додавання, множення (рівнева або з bootstrap)	RLWE (scale-invariant варіант); постквантова	2012
TFHE	FHE (булева)	Булеві операції (бітові над шифротекстами)	Гратки GSW на торі (реал. в кільці); постквантова	2016
CKKS	FHE (наближені обчислення)	Додавання, множення над зашифрованими дійсними числами	RLWE (дод. округлення при операціях); постквантова	2017

Висновок. Досліджено стан розвитку алгоритмів гомоморфного шифрування. Проведено порівняльний аналіз сучасних бібліотек та фреймворків гомоморфного шифрування. Розкрито можливості та обмеження, зокрема, підтримувані гомоморфні операції, криптографічна основа, та стійкість.

Перелік використаних джерел.

1. Microsoft SEAL. Режим доступу: https://www.microsoft.com/en-us/research/project/microsoft-seal/#:~:text=Microsoft%20SEAL-powered%20by%20open,their%20key%20with%20the%20service

2. FHE Libraries: Established Cryptographic Building Blocks. https://el3ctrum.com/uncategorized/fhe-libraries-established-cryptographic-building-blocks/#:~:text=Background%20%26%20Developers%3A%20HEAAN%20stands,the%20library's%20performance%20and%20features

3. OpenFHE. Режим доступу: https://openfhe.org

4. Private Join and Compute. Режим доступу: https://github.com/google/private-join-and-compute

УДК 004.056.5

Олександр ДРОЖАК

Західноукраїнський національний університет

АНАЛІЗ ТЕСТІВ ПРОСТОТИ ФЕРМА ТА МІЛЛЕРА-РАБІНА

Вступ. Аналіз тестів простоти числа є важливим аспектом у теорії чисел та криптографії, оскільки ефективне визначення простоти числа має ключове значення для безпеки сучасних криптографічних алгоритмів. Розвиток швидких і точних методів тестування простоти чисел дозволяє знижувати обчислювальні витрати при роботі з великими числами, що є критичним для практичних застосувань, таких як генерація ключів у криптографії.

Актуальність таких тестів також зростає з урахуванням потреб у безпечному зберіганні даних та захисті інформації в цифрову епоху.

Метою аналізу тестів простоти Ферма та Тесту Міллера-Рабіна є оцінка їх ефективності, точності та надійності при визначенні простоти великих чисел.

1. Тест Ферма

Тест Ферма дозволяє швидко виявити складні числа, але може давати хибнопозитивні результати, що робить його менш надійним для великих чисел.

За теоремою Ферма, якщо n – просте число, тоді будь-якого a справедливо така рівність

$$a^{n-1}=1 \pmod{n}.$$

Звідси ми можемо вивести правило тесту Ферма на перевірку простоти числа: візьмемо випадкове

$$a \in \{1, ..., n-1\}$$

і перевіримо чи дотримуватиметься рівність

$$a^{n-1}=1 \pmod{n}.$$

Якщо рівність недотримується, отже швидше за все n – складове.

Проте умова рівності може бути дотримано, навіть якщо n – не просте. Наприклад, візьмемо $n = 561 = 3 \times 11 \times 17$. Відповідно до Китайської теореми про залишки:

$$Q_{561} = Q_3 \times Q_{11} \times Q_{17},$$

де кожне $a \in Q*_{561}$ відповідає наступному:

$$(x,y,z) \in Q*_3 \times Q*_{11}11 \times Q*_{17}.$$

По теоремі Ферма

$$x^2=1,\ y^{10}=1 \text{ і } z^{16}=1.$$

Оскільки 2, 10 і 16 всі є дільниками 560, це означає, що $(x, y, z)^{560} = (1, 1, 1)$, тобто $a^{560} = 1$ для будь-якого $a \in Q*_{561}$.

Не має значення яке ми виберемо, 561 завжди буде проходити тест Ферма незважаючи на те, що воно складене, доки а є взаємно простим з n. Такі числа називаються числами Кармайкла і встановлено, що їх існує безліч.

Якщо a не взаємно просте з n, воно тест Ферма не проходить, але в цьому випадку ми можемо відмовитися від тестів і продовжити шукати дільники n,

обчислюючи НСД(a,n).

2. Тест Міллера-Рабіна

Тест Міллера-Рабіна, у свою чергу, є більш точним і менш схильний до помилок, що робить його одним з найбільш використовуваних методів для перевірки простоти в криптографії. Аналіз цих тестів сприяє вибору оптимального алгоритму для застосувань, де важлива швидкість та точність перевірки простоти чисел.

Можна вдосконалити тест, сказавши, що n - просте тоді і тільки тоді, коли рішеннями

$$x^2 = 1 \pmod{n} \text{ є } x = \pm 1.$$

Таким чином, якщо n проходить тест Ферма, тобто

$$a^n - 1 = 1,$$

тоді ми ще перевіряємо щоб

$$a^{(n-1)/2} = \pm 1,$$

оскільки

$$a^{(n-1)/2}$$

це квадратний корінь 1.

На жаль, такі числа, як, наприклад 1729 - третє число Кармайкла, досі можуть обдурити цей покращений тест. Можливим вдосконаленням буде проведення ітерацій. Тобто поки це буде можливо, зменшуватимемо експоненту вдвічі, доки не дійдемо до якогось числа, крім 1. Якщо ми отримаємо в результаті щось, крім -1, тоді n буде складним. Якщо говорити формальніше, то нехай 2^S буде найбільшим ступенем 2, що ділиться на n-1, тобто

$$n - 1 = 2^S q$$

для якогось непарного числа q.

Кожне число із послідовності

$$a^{n-1} = a^{(2\text{^}S)q}, a^{(2\text{^}S\text{-}1)q}, \ldots, aq.$$

Це квадратний корінь попереднього члена послідовності.

Тоді якщо n – просте число, то послідовність повинна починатися з 1 і кожне наступне число теж має бути 1, або перший член послідовність може бути не дорівнює 1, але тоді він дорівнює -1.

Тест Міллера-Рабін бере випадкове $a \in Z_n$. Якщо вищезазначена послідовність не починається з 1, або перший член послідовності не дорівнює 1 або -1, тоді n - не просте.

Виявляється, що для будь-якого складеного n, включаючи числа Кармайкла, можливість пройти тест Міллера-Рабіна дорівнює приблизно 1/4. (У середньому значно менше.) Таким чином, ймовірність того, що n пройде декілька прогонів тесту, зменшується експонентно.

Якщо n не проходить тест Міллера-Рабіна з послідовністю, що починається з 1, тоді у нас з'являється нетривіальний квадратний корінь з 1 по модулю n, і ми можемо ефективно знаходити дільники n. Тому числа Кармайкла завжди зручно розкладати на множники.

Коли тест застосовується до чисел виду pq, де p і q - великі прості числа,

вони не проходять тест Міллера-Рабіна практично у всіх випадках, оскільки послідовність не починається з 1.

На практиці тест Міллера-Рабіна реалізується так:

Дано n, потрібно знайти s, що

$$n - 1 = 2^S q$$

для деякого непарного q.

Візьмемо випадкове

$$a \in \{1,...,n-1\}$$

Якщо $a^q = 1$, n проходить тест і припиняємо виконання. Для $i = 0, \ldots, s-1$ перевірити рівність

$$a^{(2\wedge i)q} = -1.$$

Якщо рівність виконується, то n проходить тест (припиняємо виконання).Якщо жодна з вищенаведених умов не виконана, то n – складене.

Перед виконанням тесту Міллера-Рабін варто провести ще кілька тривіальних поділів на маленькі прості числа. Строго кажучи ці тести є тестами на те чи вважається число складеним, оскільки вони не доводять по суті, що число просте, що перевіряється, але точно доводять, що воно може виявитися складовим.

Існують ще детерміновані алгоритми, які працюють за поліноміальний час для визначення простоти (Agrawal, Kayal і Saxena), проте на сьогоднішній день вони вважаються непрактичними.

Висновок. Аналіз тестів простоти Ферма та Міллера-Рабіна показує, що кожен з них має свої переваги та обмеження. Тест Ферма є швидким, але може давати хибнопозитивні результати для псевдопростих чисел, що обмежує його надійність при перевірці великих чисел. Тест Міллера-Рабіна, в свою чергу, є більш точним і менш схильним до помилок, тому його часто використовують як основний метод у криптографії. Враховуючи це, оптимальним є використання тесту Міллера-Рабіна в поєднанні з іншими методами для досягнення високої точності при визначенні простоти чисел.

Перелік використаних джерел.

1. J.P. Buhler Algorithmic Number Theory: Proc. ANTS-III – Portland, OR, v.1423, Lect.Not.Comp.Sci. Springer–Verlag, 1998, 640 p.

2. D. Venturi Lecture Notes on Algorithmic Number Theory. – Springer-Verlag, New-York, Berlin, 2009, 217 p.

3. Sh.T. Ishmukhametov Methods of factorization of natural numbers: a tutorial.– Kazan, Kazan University, 2011, 190 p.

4. Ya.M.Nikolaichuk, Kasianchuk M.M., Yakymenko I.Z., Ivasiev S.V Vector and modular method of multiplication of multidigit numbers in RademacherKrestenson basis. Herald of the National University "Lviv Polytechnic" "Computer systems and networks", no. 694, 2014, pp. 118–125.

5. M.Kasyanchuk, I. Yakymenko, Y.Nykolajchuk. Matrix Algorithm of Processing of the Information Flow in Computer Systems Based on Theoretical and Numerical Krestenson's Based. Proceedings of the Integrational Conference TCSET'2010, February 23-27, 2010, p. – C: 241

УДК 004.056.5

Борисенко І.І., Кас'яненко М.М.

Національний університет «Одеська політехніка»

МАТЕМАТИЧНІ МЕТОДИ КОМБІНАТОРИКИ, ЯК ЗАСІБ СТВОРЕННЯ КРИПТОГРАФІЧНИХ ШИФРІВ

Вступ. Техніка транспозиції - це криптографічний прийом, який використовується для перетворення простого тексту в текст шифру. Це досягається шляхом перестановки положення символів у простому тексті. Є різні методи транспозиції такі як техніка залізничного паркану, прості методи стовпчастої транспозиції прості методи стовпчастої транспозиції - кілька раундів. На жаль, оскільки транспозиційний шифр не змінює частоту окремих літер, він все ще сприйнятливий до частотного аналізу, хоча транспозиція дійсно усуває інформацію з пар літер. Тому розробка подальшого розвитку цього виду шифру є актуальною. В роботі пропонується шифр, який базується на математичних методах обробки перестановок та дій з ними.

Мета: Розробка криптографічного шифру комбінаторними засобами та подальше його застосування для цілей сеганографії.

Основна частина.

Оскільки основою нового стеганографічного алгоритму є такий комбінаторний об'єкт як перестановки (позначимо літерою P), то введемо саме поняття перестановки та операції над перестановками, які будемо використовувати під час вбудовування повідомлення у контейнер. Перестановкою будемо називати бієкцію φ скінченої множини на себе. Отже, φ є перестановка на M тоді і тільки тоді, коли для довільних елементів $a,b\in M$, $a\neq b$, маємо $\varphi(a)\neq\varphi(b)$. А це означає, що перестановка визначається таблицею виду

$$\begin{pmatrix}1 & 2 & 3 & \dots & n-1 & n\\ a_1 & a_2 & a_3 & \dots & a_{n-1} & a_n\end{pmatrix},$$

де a_1, a_2, ..., a_n - різні елементи з множини M .

Для будь-яких перестановок φ та ψ визначена операція добутку $\varphi\circ\psi$, який знаходиться за правилом: спочатку переставляють стовпці в таблиці ψ так, щоб її верхній рядок співпав з нижнім рядком таблиці φ, а потім будують нову таблицю, першим рядком якої є перший рядок таблиці φ, а другим – другий рядок таблиці ψ. Щоб побудувати перестановку обернену даній φ^{-1} треба поміняти рядки перестановки φ місцями, а потім стовпці переставити так, щоб числа першого рядка були розміщені у зростаючому порядку. Для реалізації алгоритму, що пропонується в роботі, потрібно вміти розв'язувати рівняння, елементами якого є перестановки. Розглянемо рівняння:

$$\varphi\circ x=\psi \qquad (1)$$

В роботі досліджується питання чи існує така перестановка x, для якої виконується рівність (1). Якщо така перестановка існує, то чи вона єдина? Оскільки φ - перестановка, то розв'язок рівняння (1) існує і він єдиний. Оскільки по означенню φ - бієкція, то для неї існує обернена перестановка φ^{-1}, тому можна

розглянути перетворення $\varphi^{-1}\circ\psi$. Щоб показати, що $\varphi^{-1}\circ\psi$ є розв'язком рівняння (1), треба обчислити добуток $\varphi\circ(\varphi^{-1}\circ\psi)$. Використовуючи властивість асоціативності добутку [1] та визначення оберненої перестановки, яке подано вище, одержимо $\varphi\circ(\varphi^{-1}\circ\psi)=(\varphi\circ\varphi^{-1})\circ\psi=\varepsilon\circ\psi=\psi$, де ε - тотожна перестановка, яка має вигляд $\begin{pmatrix}1 & 2 & 3 & .... & n\\ 1 & 2 & 3 & .... & n\end{pmatrix}$. А це означає, що $\varphi^{-1}\circ\psi$ - є розв'язком рівняння (1). Щоб показати, що розв'язок $\varphi^{-1}\circ\psi$ - єдиний, позначимо його літерою α, тоді рівняння (1) прийме вигляд $\varphi\circ\alpha=\psi$ помноживши одержане рівняння зліва на φ^{-1}, одержимо $\varphi^{-1}\circ(\varphi\circ\alpha)=\varphi^{-1}\circ\psi$, або $(\varphi^{-1}\circ\varphi)\circ\alpha=\varphi^{-1}\circ\psi$ і, остаточно $\varepsilon\circ\alpha=\varphi^{-1}\circ\psi$, $\alpha=\varphi^{-1}\circ\psi$. Ця рівність означає, що ніяких інших розв'язків рівність (1) не має. Ще одне питання, яке треба розглянути – це обчислення кількості перестановок, які можна побудувати з елементів множини M, яка має n елементів. Вирішення цього питання залежить від того, чи різні елементи становлять множину M, або ж вона має однакові елементи. Методи такого розділу математики, як комбінаторика, дають формули для підрахунку перестановок для різного складу множини M. Якщо всі елементи множини M різні, то кількість перестановок дорівнює:

$$P_n=n! \tag{2}$$

Якщо ж в множині M є k_1 елементів першого типу, k_2 - елементів другого типу і так далі, k_s елементів s-го типу , то кількість перестановок обчислюється за формулою:

$$P_{(n;k_1,\ldots,k_s)}=\frac{n!}{k_1!\cdot k_2!\cdot\ldots\cdot k_s!} \tag{3}$$

До безпосереднього процесу вбудовування повідомлення потрібно виконати препроцесінг і самого повідомлення і контейнера, в якості якого виступає цифрове зображення в градаціях сірого, або ж синя складова кольорового зображення [2], оскільки зорова система людини менш чутлива до синього кольору. Елементи повідомлення кодуються цифрами, які належать деякій множині $M=\{1,2,...,k\}$. Елементи контейнера послідовно групуються у блоки розміром $1\times n$. При вбудовуванні повідомлення використовується деякий допоміжний масив $masiv$, в якому знаходиться k перестановок довжини n. Всі перестановки занумеровані. При вбудовуванні елемента повідомлення з кодом i в масиві $masiv$ знаходимо перестановку з номером i, перемножуємо її на ключ φ, одержуємо перестановку ψ. Масив $masiv$ не містить перестановки, добуток якої з ключем φ дає тотожну перестановку. Блоки контейнера, які складаються з однакових елементів випускаються. Елементи інших блоків переставляються згідно перестановці ψ тільки в тому випадку, якщо для будь-якої пари елементів, які обмінюються місцями, різниця їх значень не перевищує деяке число d .Стеганографічний алгоритм, назвемо його Permut, в загальному вигляді представимо наступними кроками.

Крок 1. Розбиваємо матрицю контейнера – зображення на блоки заданого розміру $1\times n$.

Крок 2. Для елемента повідомлення з кодом i в масиві $masiv$ знаходимо

перестановку з номером i, множимо її на ключ φ, одержуємо перестановку ψ.

Крок 3. Якщо блок контейнера складається з однакових елементів переходимо до наступного блоку.

Якщо блок має хоча б два різні елементи виконуємо перестановку пікселів згідно перестановці ψ тільки в тому випадку, якщо для будь-якої пари елементів, які обмінюються місцями, різниця їх значень не перевищує деяке число d, інакше переходимо до наступного блоку. Алгоритм Permut не є «сліпим» тому для декодування повідомлення потрібна наявність контейнера. Щоб декодувати повідомлення, вбудоване повідомлення алгоритмом Permut треба розбити матрицю контейнера та стеганоконтейнера на блоки того самого розміру, що і при вбудовуванні. Порівняти відповідні блоки контейнера та стего, якщо вони співпали, то це означає, що в блок повідомлення не вбудовувалося. У іншому разі треба обчислити перестановку φ^{-1} обернену до ключа φ та обчислити добуток $\alpha = \varphi^{-1} \circ \psi$. Одержану перестановку α знай ти в масиві $masiv$, порядковий номер, який відповідає α є кодом елемента вбудованого повідомлення.

Із збільшенням довжини блоку складність алгоритму збільшується, оскільки збільшується кількість випадків стосовно складу блоків, які треба аналізувати. Продемонструємо сказане на конкретному прикладі. Нехай $n = 4$, розглянемо які випадки можуть бути стосовно складу блоків контейнера та визначимо, яку мінімальну кількість перестановок повинен містити масив $masiv$. Якщо всі елементи блоку різні, то кількість різних перестановок дорівнює 4!=24. В блоці 3 однакових елементи, тоді маємо $P_{(4;3)} = \frac{4!}{3!} = \frac{24}{6} = 4$ перестановки, в блоці 2 однакових елементи – маємо $P_{(4;2)} = \frac{4!}{2!} = \frac{24}{2} = 12$ перестановок, в блоці два елементи зі значенням $c1$ і два елементи зі значенням $c2$, тоді $P_{(4;2,2)} = \frac{4!}{2!2!} = \frac{24}{4} = 6$ перестановок. Таким чином мінімальна кількість перестановок, яка нам потрібна дорівнює чотирьом, тому повідомлення можна кодувати вже чотирма цифрами. Зрозуміло, що якщо довжину блоку контейнера ще збільшити на одиницю, то кількість випадків значно зросте і складність алгоритму збільшиться. Але, якщо будуть поставлені вимоги щодо зменшення викривлень контейнера, то розширити Permut завжди можливо, включивши додаткові перевірки складу блоків.

Висновок. Розроблено стеганографічний алгоритм просторової області вбудовування Permut, заснований на використанні такої комбінаторної конфігурації як перестановки та дій з ними. Подальший розвиток роботи – це дослідження Permut для різної довжини блоків, а саме $n = 3$ та $n = 4$ з метою вивчення їх властивостей.

Перелік використаних джерел

1. Нікольський Ю.В, Пасічник В.В, Щербина Ю.М. Дискретна математика. – К.: Видавнича група BHV, 2007. – 368 с.

2. Гонсалес Р. Цифровая обработка изображений / Р.Гонсалес, Р.Вудс; пер. с англ. под ред. П.А.Чочиа. - М.: Техносфера, 2005. - 1072 с.

УДК 004.056.5:004.946

Марія ХАНЕНКО

Національний університет «Одеська політехніка»

ВИКОРИСТАННЯ ТЕХНОЛОГІЙ ДОПОВНЕНОЇ РЕАЛЬНОСТІ ДЛЯ ВІЗУАЛІЗАЦІЇ КРИПТОГРАФІЧНИХ АЛГОРИТМІВ

Вступ. Стрімкий розвиток технологій доповненої (AR) та віртуальної реальності (VR) відкриває нові можливості для наочного представлення складних наукових концепцій [1]. Особливо це актуально для криптографії, яка традиційно сприймається як абстрактна й математично складна дисципліна. Візуалізація криптографічних алгоритмів у середовищі AR дає змогу відтворювати процеси шифрування, дешифрування, генерації ключів та хешування в інтерактивному тривимірному просторі, що спрощує розуміння їхньої роботи.

Попри зростання інтересу до застосування AR у технічній освіті [2], більшість проєктів орієнтовані на механіку, біологію чи архітектуру, тоді як інформаційна безпека залишається майже не охопленою. Розроблення AR-моделей, що демонструють роботу алгоритмів AES, RSA чи ECC, може істотно підвищити засвоєння матеріалу та зацікавленість учасників освітнього процесу. Такі рішення мають цінність не лише в освіті, а й у наукових лабораторіях, де можна моделювати роботу криптосистем, візуалізувати етапи перетворення даних і досліджувати вплив параметрів на стійкість алгоритмів. Отже, поєднання AR з криптографічними моделями створює перспективний напрям для інтерактивного навчання та досліджень у галузі кібербезпеки [3].

Мета. Метою дослідження є створення моделі та прототипу системи візуалізації криптографічних алгоритмів у середовищі доповненої реальності, що дозволяє інтерактивно демонструвати процеси шифрування, дешифрування та генерації ключів.

Запропоновано інструмент для наочного відображення внутрішніх етапів роботи алгоритмів (підстановки, перестановки, раундів, розширення ключа) у графічній формі з освітньою та демонстраційною функціями. Для реалізації мети визначено завдання:

– проаналізувати сучасні підходи застосування AR у технічній освіті та можливості їх адаптації до криптографії;

– розробити архітектуру взаємодії віртуальних криптографічних об'єктів із користувачем;

– створити покрокові візуальні моделі алгоритмів AES, RSA або ECC;

– оцінити ефективність розробленого рішення у навчальних сценаріях.

Дослідження спрямоване не лише на створення прототипу, а й на перевірку результативності AR-підходу для пояснення абстрактних процесів криптографії та підвищення розуміння інформаційної безпеки.

1. Аналіз предметної області та технологічних рішень

У першому розділі проведено огляд предметної області та сучасних AR-платформ для навчальних і демонстраційних застосунків. Дослідження базується на міждисциплінарному підході, що поєднує принципи інформаційної безпеки,

програмної інженерії та технологій доповненої реальності (AR). Для реалізації поставленої мети використано комплекс методів – аналітичних, проєктних і експериментальних.

На аналітичному етапі проведено огляд сучасних AR-платформ (Unity, Unreal Engine, ARCore, ARKit, WebAR) та інструментів для тривимірної візуалізації алгоритмів. Ключові критерії вибору стеку: кросплатформенність, динамічні об'єкти, інтеграція зі скриптами та взаємодія в реальному часі. За результатами аналізу обрано платформу Unity з AR Foundation, що підтримує Android та iOS й дає змогу створювати інтерактивні демонстрації. Окрему увагу приділено педагогічним ефектам AR у технічній освіті. За рахунок поєднання просторової візуалізації та інтерактивності зменшується когнітивне навантаження та активується подвійне кодування інформації, що забезпечує швидший зворотний зв'язок [2]. Для дисциплін, де значущими є просторові перетворення й абстрактні структури (наприклад, раунди шифрування, побітові операції), AR надає наочні «якорі» для розуміння причинно-наслідкових зв'язків між кроками алгоритму.

З технологічного погляду порівняно маркерні, маркерлес і WebAR-підходи. Маркерні сценарії спрощують точне позиціювання об'єктів та відтворюваність демонстрацій у аудиторії; маркерлес-режими краще підходять для відкритих просторів і змішаних сцен; WebAR зручний для швидкого доступу з мобільних пристроїв без інсталяцій, але потребує ретельнішої оптимізації моделей. Вибір залежить від цілей заняття, особливостей приміщення та парку пристроїв [1, 2].

У контексті предметної області інформаційної безпеки AR практично не використовується у порівнянні з механікою чи архітектурою, що формує чітку нішу новизни. Для криптографії доречно застосовувати шкалу реальність–віртуальність (за Milgram–Kishino) для добору ступеня занурення: у навчальних демонстраціях достатньо AR-режиму з прозорими 3D-шарами над реальним середовищем; для дослідницьких експериментів може знадобитися зміщення до MR/VR для повного контролю сцени [3]. Така стратифікація дозволяє методично обґрунтувати, коли і який режим доцільний.

З позицій інженерії інтерфейсів визначено критерії якості AR-сцени: стабільність трекінгу, час до появи контенту (<1–2 с), цільова частота кадрів (≥30 fps на пристроях середнього класу), ергономіка взаємодії (жести/дотик/кнопки), доступність (кольорова палітра, контраст, альтернативні підказки). Для уникнення перевантаження рекомендовано прогресивне розкриття (tooltips за запитом), мінімальну кількість одночасних анімацій та обмеження полігональності моделей для WebAR [2]. Дотримання цих принципів підвищує відтворюваність занять і переносимість контенту між аудиторіями.

2. Розроблення концептуальної моделі та прототипу AR-візуалізації

Архітектура розроблюваної системи передбачає три основні модулі:

1. Модуль візуалізації, який відповідає за тривимірне відображення елементів криптографічного алгоритму – блоків даних, ключів, операцій підстановки, перестановки, побітових операцій тощо.

2. Модуль логіки алгоритму, реалізований на мові C# або Python, який обчислює внутрішні перетворення (наприклад, раунди AES або RSA-

експоненцію) та передає проміжні результати у візуальний шар.

3. Модуль взаємодії з користувачем, що дозволяє змінювати параметри алгоритму (розмір блоку, довжину ключа, кількість раундів) і спостерігати за змінами процесу у просторі доповненої реальності.

Методичною основою побудови візуалізації обрано поетапне відображення криптографічного процесу, де кожен етап – окремий об'єкт AR-сцени з анімацією та коротким описом дії. Наприклад, при моделюванні AES користувач послідовно бачить кроки SubBytes, ShiftRows, MixColumns, AddRoundKey, а у випадку RSA – створення ключів, шифрування та дешифрування числових даних. Кожен етап супроводжується підсвічуванням активних елементів, що полегшує сприйняття логічних залежностей між операціями [4]. На рисунку 1 зображено приклад AR-візуалізації алгоритму AES у тривимірному просторі, де кольорові кільця символізують етапи шифрування (SubBytes – синій, ShiftRows – жовтий, MixColumns – червоний), а напівпрозорий куб – 128-бітовий блок даних.

Рисунок 1 – Приклад AR-візуалізації алгоритму AES у тривимірному просторі

Запропоновано використання адаптивних пояснювальних елементів (tooltips), які з'являються поруч із об'єктами та пояснюють їхню роль у процесі. Реалізовано також порівняння двох алгоритмів – наприклад, AES і ChaCha20 – через одночасне відображення їхніх структур в одній AR-сцені, що дає змогу візуально порівняти принципи потокового й блочного шифрування.

Для тестування освітнього потенціалу створено сценарій, у якому користувач, навівши камеру смартфона на маркер або QR-код, активує 3D-модель криптосхеми. Він може змінювати ключ, довжину повідомлення чи тип алгоритму, а система в реальному часі відображає, як змінюються етапи обчислення. Це створює ефект «занурення у процес шифрування» та підвищує розуміння принципів криптографії.

Методологія базується на інтеграції аналітичного моделювання з AR-візуалізацією, що перетворює абстрактні математичні операції на наочні тривимірні об'єкти, доступні для дослідження. Розроблено концептуальну модель і демонстраційний прототип, які відтворюють основні етапи симетричного та асиметричного шифрування.

Під час демонстрації AES користувач бачить, як повідомлення розбивається на блоки, що послідовно проходять етапи SubBytes, ShiftRows, MixColumns,

AddRoundKey; для RSA – генерацію ключів і процес шифрування-дешифрування. AR-підсвічування та інтерактивні пояснення дозволяють простежити логіку дій і взаємозв'язки між ними.

Експериментальне тестування показало підвищення рівня розуміння принципів шифрування на 25–30 % порівняно з традиційними лекційними матеріалами. Учасники відзначили зручність подання інформації та підвищення інтересу до тематики криптографії.

Система продемонструвала стабільність і кросплатформенність: використання Unity з AR Foundation забезпечує роботу на Android, iOS та можливість WebAR-версії. Оптимізовані FBX-моделі забезпечують плавну анімацію навіть на пристроях середнього рівня.

Результати підтверджують, що поєднання криптографії з AR є перспективним для освіти й наукових демонстрацій. Підхід може стати основою віртуальних криптографічних лабораторій, де користувачі експериментують із алгоритмами та миттєво бачать вплив змін на результат шифрування.

Висновок. Дослідження підтвердило доцільність використання технологій доповненої реальності для візуалізації криптографічних алгоритмів та створення інтерактивних навчальних середовищ у галузі інформаційної безпеки [5]. Розроблений прототип підтвердив ефективність AR-візуалізації для спрощення сприйняття криптографічних процесів і підвищення мотивації до навчання.

Реалізація системи у середовищі Unity з AR Foundation продемонструвала можливість тривимірного відтворення етапів AES і RSA з динамічним керуванням параметрами та поясненнями в реальному часі. Такий підхід відкриває нові способи представлення криптографічних процесів у навчальному й демонстраційному форматах. У перспективі передбачено розширення функціоналу через інтеграцію VR/MR-технологій, розроблення навчальних модулів для різних типів алгоритмів та впровадження елементів гейміфікації. Це створює основу для віртуальних лабораторій криптографії нового покоління, що поєднують навчання, дослідження та візуальну аналітику.

Перелік використаних джерел.

1. Азума Р. Огляд технологій доповненої реальності. Presence: Teleoperators and Virtual Environments. – 1997. – Т. 6, № 4. – С. 355–385.

2. Крейг А. Б. Розуміння доповненої реальності: концепції та застосування. – Київ: Видавництво Університету, 2021. – 284 с.

3. Мілграм П., Кісіно Ф. Таксономія змішаних візуальних дисплеїв реальності. IEICE Transactions on Information and Systems. – 1994. – Т. E77-D, № 12. – С. 1321–1329.

4. Столлінгс В. Криптографія та безпека комп'ютерних мереж: принципи та практика. – 8-ме вид. – Харків: Ранок, 2023. – 890 с.

5. Коляда А.С., Павлишко А.В., Лопаков О.С., Тігарєв В.М., Космачевський В.В. Використання машинного навчання для виявлення вразливостей криптографічних алгоритмів на основі шифротексту. Інформатика та математичні методи в моделюванні. – 2025. – Т. 15, № 1. – С. 83–94.

УДК 004.056

Гнєдова В.О., Вінковська І.С.

Національний університет «Одеська політехніка»

КРИПТОГРАФІЧНИЙ ЗАХИСТ DICOM–ЗОБРАЖЕНЬ: ПРОБЛЕМИ, РИЗИКИ ТА НАПРЯМИ РОЗРОБКИ ПРОГРАМНИХ ЗАСОБІВ

Вступ. Сучасна медицина активно використовує цифрові технології для діагностики, зокрема медичні зображення, які зберігаються та передаються у форматі DICOM (Digital Imaging and Communications in Medicine). Цей стандарт забезпечує обмін результатами різних обстежень – комп'ютерної томографії, магнітно-резонансної томографії, ультразвукових досліджень та інших. Зростання обсягів медичних даних підвищує ефективність діагностики та лікування, але водночас створює серйозні ризики для конфіденційності та цілісності інформації. Несанкціонований доступ до медичних зображень може призвести до порушення прав пацієнтів, маніпуляцій із результатами досліджень і юридичних наслідків для медичних установ.

Мета: Проаналізувати проблеми криптографічного захисту медичних зображень у форматі DICOM. У роботі досліджуються вразливості медичних зображень під час зберігання та передачі, сучасні методи забезпечення конфіденційності, цілісності та автентичності DICOM–файлів, а також виявляються ключові проблеми й ризики для медичних закладів і пацієнтів.

1. Проблематика зберігання та передачі DICOM–зображень

Файли DICOM містять не лише медичні зображення, але й метадані пацієнта, які включають особисту інформацію та діагностичні відомості. Це робить їх особливо цінними й водночас уразливими до різного роду загроз. Однією з ключових проблем є несанкціонований доступ: у деяких медичних системах відсутні ефективні засоби шифрування, що дозволяє стороннім особам переглядати конфіденційні дані. Витік інформації може відбутися під час передачі DICOM-файлів через мережі з низьким рівнем захисту, створюючи ризик перехоплення або підміни даних. Крім того, можливі спроби модифікації файлів, коли зміни в зображеннях або метаданих спотворюють діагностичну інформацію, що критично для точності медичних висновків. Тому забезпечення надійного захисту DICOM–зображень є надзвичайно актуальним завданням у сучасній електронній медицині [1].

2. Методи криптографічного захисту

Криптографічні методи відіграють ключову роль у забезпеченні безпеки медичних даних, гарантують їх конфіденційність, цілісність та автентичність. Конфіденційність досягається шляхом шифрування даних, цілісність – контролем змін після створення чи передачі, а автентичність – підтвердженням походження даних від надійного джерела [2].

Серед сучасних підходів виділяють симетричне шифрування (AES, DES), що є швидким і ефективним, але вимагає безпечного обміну ключами, і асиметричне шифрування (RSA, ECC), яке забезпечує безпечну передачу ключів, але має меншу швидкодію при великих обсягах даних. Цифрові підписи та хеш–

функції (SHA–256) дозволяють перевіряти цілісність файлів і підтверджувати їх достовірність [2]. Водночас багато медичних систем досі не інтегрують повноцінні криптографічні засоби, що створює реальні ризики для безпеки даних пацієнтів.

У подальших дослідженнях планується розробити програмний застосунок, який реалізовуватиме комбінований підхід до шифрування DICOM–зображень, забезпечуючи безпеку та швидкість одночасно.

3. Аналіз проблем та ризиків

Основними проблемами безпеки DICOM–зображень є:

– недостатня стандартизація криптографічного захисту у DICOM–системах, що призводить до різного рівня безпеки серед виробників;

– складність інтеграції криптографічних засобів у медичні процеси, адже лікарі потребують швидкого доступу до зображень;

– людський фактор – помилки персоналу при керуванні ключами або налаштуванні систем;

– відсутність контролю автентичності при передаванні файлів через відкриті мережі, що може призвести до непомітної підміни даних [3].

Отже, існує потреба у комплексному підході до захисту DICOM–зображень, який поєднує технічні та організаційні заходи безпеки. Особливо важливо забезпечити створення програмних інструментів, здатних автоматизувати процеси шифрування, перевірки цілісності та автентифікації файлів, мінімізуючи вплив людського фактора.

Висновок. Аналіз проблем криптографічного захисту медичних зображень показує, що забезпечення конфіденційності, цілісності та автентичності DICOM–файлів є ключовим завданням сучасної електронній медицині. Виявлені проблеми – недостатня стандартизація шифрування, складність інтеграції криптографії у медичні процеси, ризики, пов'язані з людським фактором, та відсутність контролю автентичності при передачі даних – свідчать про високий рівень потенційних загроз. Це підкреслює необхідність комплексного підходу, який включає впровадження криптографічних технологій, політик управління доступом і навчання персоналу. Такий підхід підвищить рівень захисту медичних даних, зменшить ризики витоку інформації та сприятиме достовірності діагностичних процесів.

Перелік використаних джерел.

1. Medcrypt. "Чому захищений DICOM погано впроваджується: аналіз проблем." 2021. [Електронний ресурс]. – Режим доступу: https://www.medcrypt.com/blog/why-secure-dicom-is-poorly-accepted-understanding-the-challenges

2. Рахман, А. А. Т., Іслам, М. М., & Хоссайн, М. А. "Захист медичних зображень у телемедицині: систематичний огляд." 2022. [Електронний ресурс]. – Режим доступу: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8938747/

3. Бернс, Е. С., & Брейнінг, Р. Дж. "Безпека даних пацієнта та досліджень у DICOM-зображеннях." 2010. [Електронний ресурс]. – Режим доступу: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3043670/

УДК 004.056

Дмитро ПЕРЕРВА

Західноукраїнський національний університет

АЛГОРИТМИ ШИФРУВАННЯ ДЛЯ ПІДВИЩЕННЯ БЕЗПЕКИ ОБМІНУ ПОВІДОМЛЕННЯМИ

Вступ. У сучасному світі безпечний обмін повідомленнями є ключовою складовою цифрової комунікації. Зростання кількості кіберзагроз, атак на мережеві сервіси та спроб перехоплення даних вимагає використання надійних методів криптографічного захисту. Алгоритми шифрування є основою безпечних комунікацій у більшості месенджерів і протоколів – від TLS до Signal. Від ефективності цих алгоритмів залежить не лише конфіденційність даних, але й довіра користувачів до цифрових сервісів.

Мета. Дослідження сучасних алгоритмів шифрування, їхньої архітектури, взаємодії компонентів і практичної реалізації для підвищення рівня безпеки обміну повідомленнями.

1. Аналіз криптографічних підходів у системах обміну повідомленнями

На сьогодні існує два базових підходи до шифрування – симетричний та асиметричний. Симетричні алгоритми (наприклад, AES, ChaCha20) використовують один спільний ключ для шифрування і розшифрування повідомлень. Вони мають високу швидкодію і застосовуються для обробки великих обсягів даних. Натомість асиметричні методи (RSA, ECC, Curve25519) використовують пару ключів – відкритий і приватний, що дозволяє безпечно передавати ключі та забезпечує автентичність учасників обміну.

Більшість сучасних протоколів, таких як Signal Protocol чи OMEMO, поєднують обидва підходи. Асиметричні алгоритми відповідають за створення сеансових ключів, тоді як симетричні – за безпосереднє шифрування даних. Така гібридна модель гарантує і конфіденційність, і високу продуктивність системи. Для побудови власної системи шифрування в рамках цього дослідження обрано поєднання алгоритмів AES-GCM і Curve25519, що відповідає сучасним стандартам безпеки, зокрема рекомендаціям NIST і Signal Foundation. На рисунку 1. наведена структурна схема алгоритму Curve25519 [1].

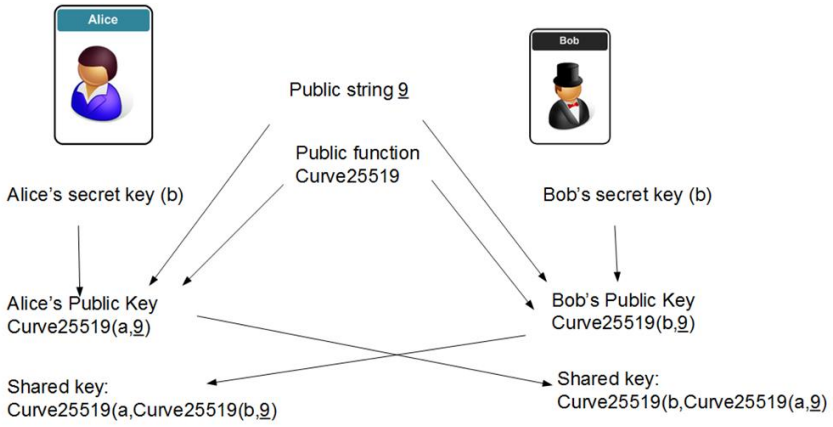


Рисунок 1 - Структурна схема алгоритму Curve25519

З огляду на сучасні тенденції розвитку цифрової комунікації, головною вимогою до будь-якої системи обміну повідомленнями стає забезпечення конфіденційності, цілісності та автентичності даних. Використання криптографічних алгоритмів дозволяє створити стійкі до атак канали зв'язку, однак ефективність таких систем залежить не лише від вибору алгоритму, але й від його коректного впровадження. Останніми роками у світі відбувається активний перехід до асиметричних схем на еліптичних кривих, зокрема на базі Curve25519, яка демонструє високу швидкодію при мінімальному споживанні обчислювальних ресурсів. Вона широко застосовується у популярних месенджерах – Signal, WhatsApp, Session, Element, – де реалізована в рамках протоколів безпечного обміну ключами. Її перевагою є відсутність необхідності у складних сертифікаційних механізмах, що суттєво спрощує інтеграцію в мобільні та десктопні застосунки.

Таким чином, поєднання AES-GCM та Curve25519 утворює гібридну криптосистему, у якій швидкість симетричного шифрування поєднана з безпечним розподілом ключів через асиметричну схему. Особливої актуальності така комбінація набуває в умовах постійного зростання кількості кібератак на месенджери, хмарні сервіси та соціальні платформи. Зловмисники дедалі частіше використовують методи аналізу трафіку, підміни відкритих ключів або атаки типу «людина посередині» (Man-in-the-Middle). Тому реалізація ефективних механізмів шифрування на основі перевірених алгоритмів є ключовим елементом сучасних систем безпечного обміну повідомленнями[2].

2. Реалізація та взаємодія компонентів алгоритму AES-GCM з Curve25519

Для забезпечення конфіденційності повідомлень у сучасних комунікаційних системах важливо застосовувати криптографічні алгоритми, які поєднують високу швидкодію, надійність і можливість реалізації у програмних продуктах з обмеженими ресурсами. У даній роботі розглядається використання симетричного алгоритму AES у режимі Galois/Counter Mode (GCM) у поєднанні з асиметричним механізмом обміну ключами Curve25519. Такий підхід реалізовано у багатьох сучасних протоколах безпечного обміну повідомленнями, зокрема у Signal Protocol та Session, завдяки його здатності забезпечувати Forward Secrecy і високу стійкість до атак. У програмному коді (рисунок 2) реалізовано модуль AESGCM, який виконує основні операції шифрування та дешифрування даних, а також генерує симетричний ключ на основі обміну відкритими ключами за алгоритмом X25519. Код написано мовою Kotlin і використовує бібліотеки javax.crypto для базових криптографічних операцій та Curve25519 для еліптичної криптографії.

```
internal fun encrypt(plaintext: ByteArray, hexEncodedX25519PublicKey: String): EncryptionResult
    val x25519PublicKey = Hex.fromStringCondensed(hexEncodedX25519PublicKey)
    val ephemeralKeyPair = Curve25519.generateKeyPair()
    val symmetricKey = generateSymmetricKey(x25519PublicKey, ephemeralKeyPair.secretKey.data)
    val ciphertext = encrypt(plaintext, symmetricKey)
    return EncryptionResult(ciphertext, symmetricKey, ephemeralKeyPair.pubKey.data)
}
```

Рисунок 2 - Фрагмент коду реалізації алгоритму AES-GCM з Curve25519 (Kotlin)

Основною метою модуля є забезпечення безпечного шифрування повідомлень з автентифікацією даних. Режим AES-GCM дозволяє одночасно виконувати шифрування та контроль цілісності, що усуває потребу у додаткових механізмах перевірки автентичності.

У процесі шифрування генерується вектор ініціалізації (IV) розміром 12 байтів, який додається до зашифрованих даних, що забезпечує унікальність кожної операції. Тег автентичності (GCM Tag) довжиною 128 біт додається до результату шифрування, дозволяючи виявляти будь-які зміни у переданих даних. Функція generateSymmetricKey() реалізує генерацію симетричного ключа через обмін публічними та приватними ключами з використанням алгоритму Curve25519. Для формування остаточного симетричного ключа застосовується функція HMAC-SHA256, яка забезпечує криптографічно стійке перетворення проміжного секрету (shared secret). Цей підхід дозволяє уникнути зберігання ключів у відкритому вигляді та підвищує стійкість системи до атак типу перехоплення ключа. Під час шифрування даних функція encrypt() створює випадковий IV, ініціалізує об'єкт Cipher у режимі GCM, виконує операцію шифрування та поєднує IV з результатом шифрування.

Процес дешифрування реалізований у функції decrypt(), яка виділяє IV з початку блоку даних, ініціалізує Cipher у режимі дешифрування та відновлює оригінальне повідомлення. Комбінація AES-GCM і Curve25519 утворює надійну гібридну криптосистему, у якій симетричне шифрування відповідає за швидкість, а асиметричний обмін ключами – за безпеку. Це дозволяє будувати протоколи з властивістю проспективної секретності (Forward Secrecy), тобто навіть у разі компрометації ключів у майбутньому зловмисник не зможе розшифрувати вже передані повідомлення. Розглянутий модуль може бути використаний як основа для реалізації безпечного обміну повідомленнями у мобільних або десктопних застосунках [3].

Висновок. Проведене дослідження показало, що поєднання симетричних і асиметричних алгоритмів у межах одного криптографічного рішення забезпечує високий рівень безпеки при збереженні швидкодії. Реалізований механізм шифрування на основі AES-GCM та Curve25519 гарантує конфіденційність, цілісність і автентичність даних у системах обміну повідомленнями.

Підхід може бути використаний як база для подальшої інтеграції у месенджери або корпоративні платформи з підтримкою E2EE.

Перелік використаних джерел.

1. Структурна схема алгоритму Curve25519. [Електронний ресурс]. – Режим доступу: https://asecuritysite.com/encryption/go_25519ecdh2

2. Signal Protocol Documentation. [Електронний ресурс]. – Режим доступу: https://signal.org/docs/

3. RFC 5116 – An Interface and Algorithms for Authenticated Encryption. [Електронний ресурс]. – Режим доступу: https://www.rfc-editor.org/rfc/rfc5116

УДК 004.056.53

Сарапук О.І., Рибінський В.О., Сапіташ В.І.

Західноукраїнський національний університет

АРХІТЕКТУРА СИСТЕМИ КВАНТОВОГО РОЗПОДІЛУ КЛЮЧІВ

Вступ. Традиційні криптографічні методи, засновані на складності математичних задач, поступово втрачають надійність у зв'язку з появою потужних обчислювальних засобів і перспективних квантових обчислювачів, здатних ефективно зламувати класичні шифри. У цьому контексті особливої актуальності набувають квантові методи захисту інформації [1], зокрема квантовий розподіл ключів (КРК) (Quantum Key Distribution, QKD), який забезпечує безумовну стійкість криптографічного обміну завдяки використанню фундаментальних принципів квантової механіки [2].

Розробка ефективної архітектури системи квантового розподілу ключів є актуальним науково-технічним завданням і важливим кроком до впровадження квантової безпеки в практичні телекомунікаційні мережі, який визначає перспективи формування безпечних інформаційних середовищ у майбутніх квантових комунікаційних мережах.

Мета: розробити архітектуру системи квантового розподілу ключів.

1. Розробка архітектури системи квантового розподілу ключів

Квантова апаратура, що реалізує протокол КРК, являє собою комплекс із двох пристроїв, з'єднаних квантовим каналом. Спрощена архітектура комплексу наведена на рисунку 1.

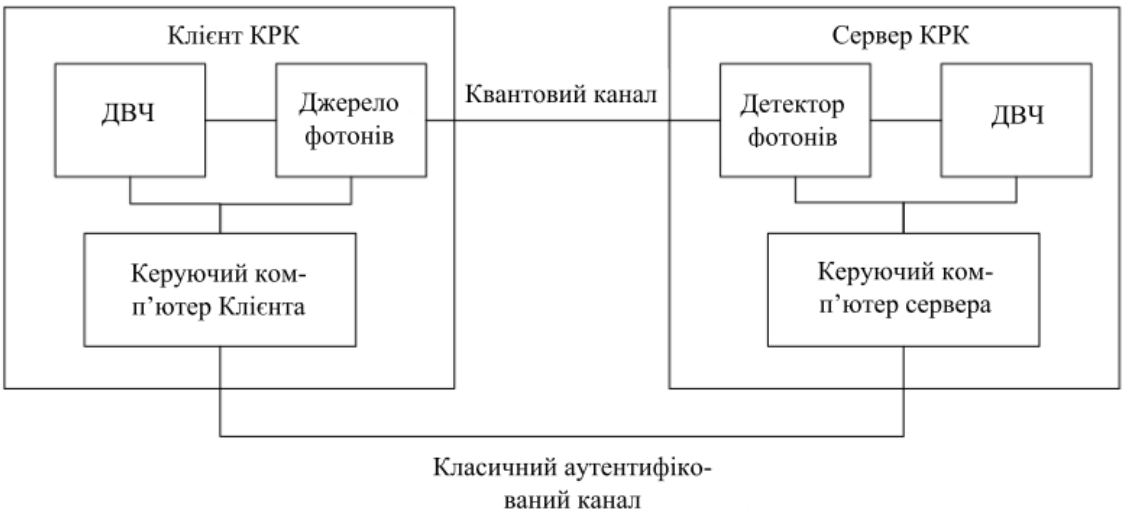


Рисунок 1 – Схема комплексу квантової апаратури

Один з пристроїв комплексу, що містить генератор (джерело) поодиноких фотонів, прийнято називати Клієнтом КРК. Суміжний пристрій, що містить детектор (приймач) поодиноких фотонів, називають Сервером КРК. Кожен з пристроїв має датчик випадкових чисел (ДВЧ). При цьому рекомендується використовувати датчики, в основі випадкових процесів яких лежать квантові ефекти, що дозволяє отримати істинно випадкову послідовність, з якої надалі формується квантовий ключ.

Сервер КРК і Клієнт КРК з'єднані двома логічними каналами: квантовим і класичним. Квантовий канал призначений для передачі квантових інформаційних станів, тобто фотонів і, як правило, реалізується за допомогою звичайного оптоволокна. Існують системи КРК, у яких як квантовий канал застосовується повітряне середовище, але вони поки що перебувають на стадії лабораторних установок.

Важливою особливістю технології КРК є повна доступність квантового каналу для зловмисника, тобто цей канал не контролюється і не захищається від втручання. Крім квантового каналу, Сервер і Клієнт КРК повинні бути з'єднані класичною лінією зв'язку, де реалізується класичний автентифікований канал. До цього каналу висуваються вимоги щодо забезпечення цілісності переданих даних та автентифікації відправника.

Реальна система КРК додатково має ще логічний службовий канал для передачі даних, який передає команди управління й моніторингу апаратури, не пов'язані безпосередньо з протоколом КРК. У деяких реалізаціях може знадобитися забезпечення не лише цілісності, а й конфіденційності цих даних. Для роботи системи КРК в апаратуру необхідно завантажити попередньо розподілені ключі, які потрібні щонайменше для побудови класичного автентифікованого каналу до першого успішного отримання достатньої кількості квантових ключів. Одну ітерацію реалізації протоколу КРК називають сеансом КРК.

Зазвичай кожен сеанс КРК складається з таких етапів:

– підготовка квантового каналу;
– передача поодиноких фотонів квантовим каналом;
– постобробка переданої послідовності.

У результаті передачі квантовим каналом обидва пристрої отримують так званий сирий ключ. Далі постобробка відбувається через класичний автентифікований канал і включає три підетапи:

– узгодження базисів вимірювання на стороні приймача з базисами кодування на стороні джерела. Неспівпадіння відкидаються, а сирий ключ перетворюється на просіяний ключ;

– виправлення помилок у просіяних ключах для отримання ідентичних послідовностей у Сервері та Клієнті КРК. Результат – очищений ключ;

– посилення секретності – стиснення очищеного ключа для зменшення інформації, доступної зловмиснику. Результат – секретний квантовий ключ.

На рисунку 2 представлена узагальнена послідовність виконання протоколу КРК.

Потрібно відзначити, що результат роботи квантового протоколу не зовсім коректно називати квантовим ключем. Правильніше говорити, що результатом сеансу КРК є випадкова квантова ґамма, ідентична у двох абонентів, оскільки цей результат має змінну довжину, яка не завжди збігається з довжиною ключів, що застосовуються в алгоритмах кодування. Більше того, результат виконання одного й того ж протоколу КРК суттєво відрізняється для квантових каналів із низькими та високими втратами, що безпосередньо впливають на величину помилок під час передачі в квантовому каналі (QBER). Це, своєю чергою, впливає на обсяг

інформації про квантову ґамму, доступної порушнику, і яка зменшується на етапі посилення секретності.

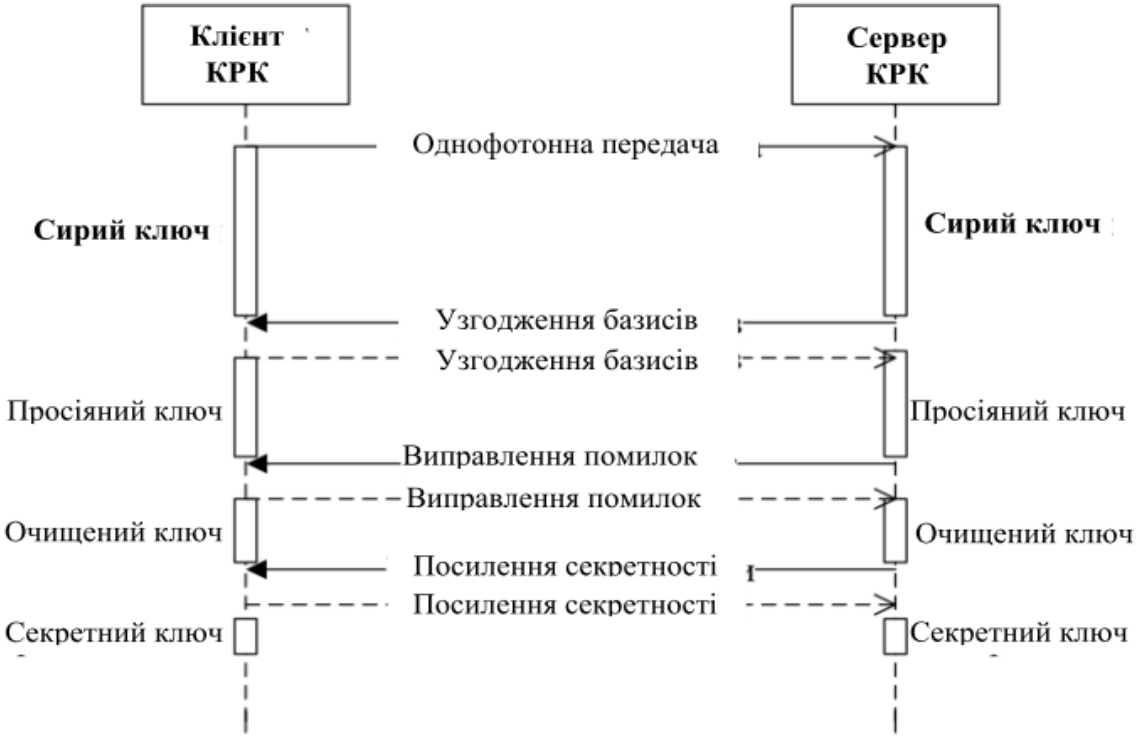


Рисунок 2 – Послідовність виконання протоколу КРК

Згідно з експериментальними даними, наведеними у [1], при довжині лінії у 50 км (що відповідає втратам 10 дБ при типовому затуханні у ВОЛЗ 0,2 дБ/км [2]) ефективність вироблення квантових ключів, тобто відношення числа зареєстрованих імпульсів на сервері КРК до загальної кількості імпульсів, відправлених клієнтом КРК, становить 2×10^{-5}. Таким чином, щоб отримати 256-бітний квантовий ключ при довжині лінії 50 км за один сеанс КРК, потрібна послідовність у 2×10^{7} імпульсів. Втрати при довжині квантового каналу у 100 км складають 20 дБ, тобто у 10 разів більше, ніж при 50 км. Тому для вироблення 256-бітного квантового ключа за один сеанс КРК послідовність імпульсів, що передається квантовим каналом, має бути в 10 разів більшою, тобто не менше, ніж 2×10^{8} імпульсів.

Висновок. Розроблено архітектуру системи квантового розподілу ключів, що дало можливість регулярної генерації спільних квантових ключів у користувацькі пристрої.

Перелік використаних джерел.

1. Quantum Safe Cryptography and Security [Електронний ресурс]. ETSI. Режим доступу: http://www.etsi.org/images/files/ETSIWhitePapers/QuantumSafeWhitepaper.pdf.

2. Quantum Key Distribution Products [Електронний ресурс]. TOSHIBA CORPORATION. Режим доступу: https://www.toshiba.co.jp/qkd/en/products.htm.

УДК 004.056.53:004.932.2

Микола ГУЛА, Олена АГАДЖАНЯН

Національний університет «Одеська політехніка»

РОЗРОБКА СТЕГАНОАНАЛІТИЧНОГО АЛГОРИТМУ ДЛЯ ЦИФРОВИХ ЗОБРАЖЕНЬ

Вступ. Невпинний розвиток інформаційних технологій та інтенсифікація кіберзагроз зумовлюють необхідність постійного удосконалення засобів захисту конфіденційної інформації. Стеганографічні методи є одним з дієвих інструментів приховування важливих даних шляхом вбудовування секретних повідомлень у цифрові зображення.

Однак стрімкий розвиток стеганографії також створює ризики її незаконного використання для приховування злочинної діяльності чи розповсюдження шкідливої інформації. Тому виникає необхідність у розробці ефективних методів стеганоаналізу для виявлення факту наявності прихованої інформації в цифрових носіях.

Мета: детектування вбудованого повідомлення в цифрових зображеннях шляхом розробки стеганоаналітичного алгоритму на основі аналізу частотної області без наявності оригінального контейнера.

1. Стеганоаналітичний алгоритм

Об`єктом аналізу є стеганографічний метод [1], що вбудовує додаткову інформацію (ДІ) у частотній області зображення. Ключова особливість методу полягає в тому, як саме відбувається підготовка контейнера та безпосередньо вбудовування ДІ у кожен блок його частотних коефіцієнтів. Спочатку виконується модифікація блоків просторової області таким чином, щоб коефіцієнти у частотній області дискретного перетворення Фур`є (ДПФ) стали цілими числами. Після переходу у частотну область ДПФ змінює парність всіх елементів блоку одночасно.

Така жорстка маніпуляція є ефективною для вбудовування ДІ, але водночас створює статистичну вразливість. Суть її полягає в тому, що для природних зображень характеристики частотних коефіцієнтів, отриманих після ДПФ, мають певні закономірності. Зокрема, парність цих коефіцієнтів має рівноймовірний характер. Це означає, що очікувана кількість парних і непарних коефіцієнтів є приблизно однаковою.

Саме ця статистична особливість і лежить в основі стеганоаналітичного методу. Його принцип полягає у підрахунку аномально великої кількості блоків, де парність усіх коефіцієнтів була штучно уніфікована.

На рисунку 1 наведено гістограму розподілу частки блоків розміром 2×2, в яких всі коефіцієнти частотної області після переходу до частотної області є або парними, або непарними.

Кількість зображень досить різко спадає зі збільшенням частки «парних»/«непарних» блоків. Зокрема, лише близько 40 зображень мають частку таких блоків 30%, а менше 30 зображень – 50% і вище.

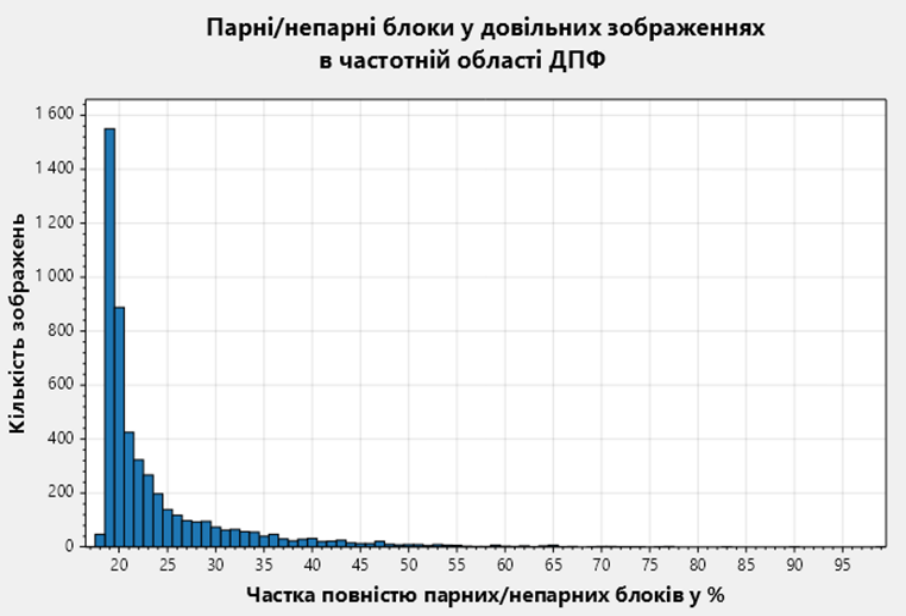


Рисунок 1 - Гістограма розподілу частки блоків розміром 2×2

Запропонований стеганоаналітичний алгоритм, що працює у частотній області, виконується в кілька послідовних етапів.

На першому етапі вихідна матриця розбивається на блоки 2×2.

Після цього, на другому етапі, виконується перехід у частотну область за допомогою ДПФ.

Третій етап полягає у підрахунку загальної кількості блоків, які складаються з усіх парних, або непарних коефіцієнтів:

$$C_{\text{парні/непарні}} = \sum_n \text{if } (\text{k} = 0 \,||\, \text{k} = 4), \quad (1)$$

де $C_{\text{парні/непарні}}$ – загальна кількість блоків, що складаються з усіх парних, або непарних коефіцієнтів;

n – загальна кількість блоків;

k – кількість парних елементів у блоці.

Далі, на четвертому етапі, розраховується частка таких блоків:

$$Q_{\text{парні/непарні}} = \frac{C_{\text{парні/непарні}}}{n} \times 100, \quad (2)$$

де $Q_{\text{парні/непарні}}$ – частка блоків, що складаються з усіх парних, або непарних коефіцієнтів;

$C_{\text{парні/непарні}}$ – загальна кількість блоків, що складаються з усіх парних, або непарних коефіцієнтів;

n – загальна кількість блоків.

Завершальним, п`ятим етапом є порівняння частки блоків, що складаються з усіх парних або непарних коефіцієнтів, з пороговим значенням:

$$\begin{cases} 0, & if\ Q_{\text{парні/непарні}} \leq Q_{\text{макс}} \\ 1, & if\ Q_{\text{парні/непарні}} \geq Q_{\text{макс}} \end{cases}, \quad (3)$$

де $Q_{\text{парні/непарні}}$ – частка блоків, що є або повністю парними або непарними;

$Q_{\text{макс}}$ – максимально допустима частка парних блоків.

Значення $Q_{\text{макс}}$ пропонується брати 55%. Наведений вище на рисунку 1 статистичний аналіз показує, що частка блоків, які мають усі парні або непарні коефіцієнти в частотній області ДПФ у ~99% природних зображень не перевищує 55%. Тому встановлення порогового значення в діапазоні 55–100% дозволяє ефективно виявляти аномалії, спричинені стеганографічними алгоритмами, що маніпулюють парністю коефіцієнтів у частотній області ДПФ.

2. Ефективність розробленого алгоритму

Для оцінки ефективності алгоритму використовується ймовірність виявлення ДІ, враховуючи помилки першого та другого роду.

Проведено дослідження ефективності на вибірці із 5000 зображень. Ця вибірка складалася з двох рівних частин: перша містила 2500 «чистих» зображень без будь-якої вбудованої ДІ, а друга – 2500 зображень, в які було приховано ДІ за допомогою стеганографічного методу, заснованого на ДПФ.

Дослідження проводилося також із різним рівнем вкладення ДІ, а саме 100%, 75%, 50% та 25%.

На рисунку 2 зображено гістограму, що демонструє помилки на різних рівнях вкладення ДІ.

Рисунок 2 - Гістограма, що демонструє помилки на різних рівнях вкладення ДІ

Для всіх рівнів вкладення частки помилок першого роду залишаються низькими – близько 0,42%. Проте частки помилок другого роду різко зростають від 0% для високих рівнів вкладення 100%, 75% та 50% до 47,62% для найнижчого рівня вкладення 25%.

Важливим параметром оцінки стеганоаналітичних алгоритмів є також точність виявлення (Detection Accuracy) (4) [2]:

$$Accuracy = \frac{True\ Positives + True\ Negatives}{True\ Positives + False\ Positives + True\ Negatives + False\ Negatives}. \quad (4)$$

На рисунку 3 наведено гістограму, що демонструє точність детектування алгоритму відносно різних рівнів вкладення ДІ.

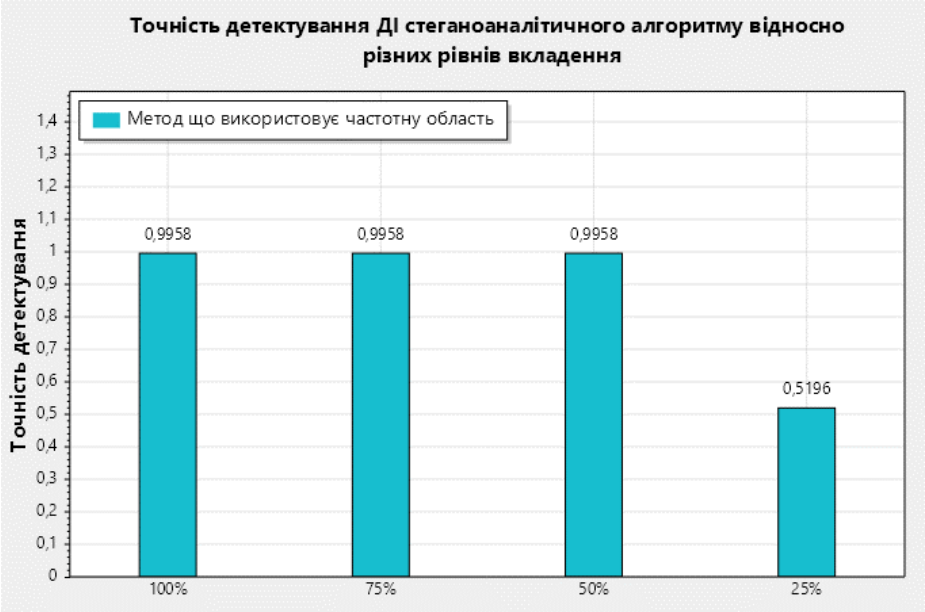


Рисунок 3 - Гістограма, що демонструє точність детектування алгоритму відносно різних рівнів вкладення ДІ

Алгоритм демонструє високу точність, близько 99,58%. Однак при зменшенні рівня вкладення до 25% точність стеганоаналітичного алгоритму суттєво знижується та складає близько 51,96%.

Висновок. Розроблено стеганоаналітичний алгоритм, що дозволяє виявляти приховані повідомлення в цифрових зображеннях без наявності оригінального контейнера, вбудовані методом, що маніпулює парністю коефіцієнтів блоку зображення у частотній області дискретного перетворення Фур`є. Експериментально продемонстровано, що алгоритм є високоефективним при значних рівнях вбудовування даних (50-100%), однак його точність суттєво знижується при низькому рівні заповнення контейнера (25%).

Таким чином, розроблений алгоритм є дієвим інструментом для виявлення певного класу стеганографічних методів без наявності оригінального контейнера, особливо при великому обсязі вбудованої інформації.

Перелік використаних джерел.

1. Kozina M.O. Discrete Fourier transform as a basis for steganographic method. Праці Одеського політехнічного університету. 2014. Вип. 2(44). С. 147–152. URL: http://dspace.op.edu.ua/jspui/bitstream/123456789/2807/1/24.pdf

2. Rasool Z. The Detection of Data Hiding in RGB Images Using Statistical Steganalysis. URL: https://meu.edu.jo/libraryTheses/5a154213e5882_1.pdf

УДК 004

Катерина БАТЬКІВСЬКА, Сергій КУЛИНА

Західноукраїнський національний університет

МЕТОДИ ВИЯВЛЕННЯ ПІДРОБЛЕНИХ АБО ЗМІНЕНИХ ЗОБРАЖЕНЬ ІЗ ЗАСТОСУВАННЯМ КРИПТОГРАФІЧНИХ ХЕШ-ФУНКЦІЙ

Вступ. У цифрову епоху зображення стали одним із ключових носіїв інформації, що використовуються в медіа, освіті, електронній комерції, державних установах та правовій системі. З розвитком технологій редагування та створення візуального контенту (зокрема DeepFake, GAN) значно зросла кількість підроблених зображень, які можуть бути використані для дезінформації, фальсифікації доказів або маніпуляції громадською думкою [1]. Це зумовлює необхідність створення надійних методів перевірки автентичності цифрових файлів. Одним із найефективніших і водночас простих інструментів є криптографічні хеш-функції, які забезпечують контроль цілісності та виявлення будь-яких змін у зображенні [2].

Мета: дослідження методів виявлення підроблених або змінених зображень із застосуванням криптографічних хеш-функцій.

1. Принципи роботи криптографічні хеш-функції

Криптографічна хеш-функція перетворює будь-яку кількість даних на короткий, унікальний ідентифікатор - хеш-код. Якщо змінити навіть один піксель, колір або метадані (EXIF) у файлі, нове хеш-значення буде повністю відрізнятися від оригіналу. Ця властивість, відома як ефект лавини, дозволяє надійно виявляти навіть незначні підробки.

Основні вимоги до безпечної хеш-функції:

- односторонність - неможливо відновити вихідні дані з хешу;
- стійкість до колізій - важко знайти два різні файли з однаковим хешем;
- висока продуктивність - швидке обчислення хешів для великої кількості зображень.

Серед сучасних алгоритмів найпоширенішими є MD5, SHA-1, SHA-256 та SHA-3 [3]. Однак, MD5 та SHA-1 більше не вважаються безпечними через доведені колізії. На противагу їм алгоритми сімейства SHA-2 (зокрема SHA-256) забезпечують високу стійкість до атак, а SHA-3 забезпечує ще вищий рівень безпеки завдяки принципу губчастої конструкції.

2. Практичні аспекти виявлення змінених зображень

Для перевірки автентичності зображення використовується наступний алгоритм дій:

- обчислення хешу оригінального файлу (SHA-256 або SHA-3);
- збереження цього хешу в безпечному сховищі або базі даних;
- під час повторної перевірки, обчислення нового хешу та порівняння його з оригіналом;
- якщо хеші не збігаються, зображення було змінено або пошкоджено.

Графічно алгоритм зображено на рисунку 1.

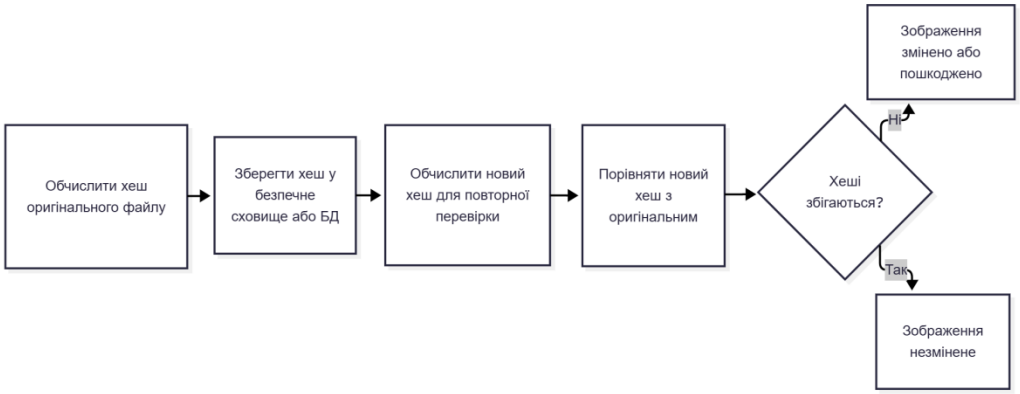


Рисунок 1 - Алгоритм перевірки автентичності зображення

Алгоритм, представлений на рисунку 1, дозволяє легко виявити будь-які спроби несанкціонованого редагування, навіть якщо зміни невидимі для людського ока. Для підвищення рівня безпеки система може бути доповнена цифровими підписами (DSA або RSA), які гарантують автентичність автора, а також використанням стеганографічних водяних знаків, які приховують хеш безпосередньо в піксельних даних зображення [4].

3. Порівняльний аналіз алгоритмів хешування

MD5 та SHA-1 – це ранні криптографічні хеш-функції, але зараз вважаються застарілими через відомі вразливості до колізій. MD5 забезпечує високу швидкість, але низький рівень безпеки, і більше підходить для неформального контролю даних, ніж для захисту. SHA-1 дещо надійніший, але також більше не рекомендується для критично важливих застосувань.

На противагу цьому, SHA-256, який є частиною сімейства SHA-2, забезпечує значно вищу безпеку, зберігаючи при цьому хорошу продуктивність. Він широко використовується для перевірки цілісності файлів, цифрових підписів та в блокчейн-системах. Найновіша – SHA-3, яка базується на принципово іншій конструкції (Keccak). Вона демонструє дуже високу стійкість до атак і використовується в критично важливих інформаційних системах, зокрема для захисту цифрових доказів та в урядових криптографічних стандартах. На відміну від SHA-2, який використовує класичну схему Меркла-Дамгарда, SHA-3 реалізує губкоподібну (sponge) конструкцію, що дозволяє гнучко налаштовувати параметри безпеки та ефективно працювати в умовах обмежених ресурсів. Обидва алгоритми підтримуються сучасними криптографічними протоколами (TLS, PGP, S/MIME) і рекомендовані до використання такими організаціями, як NIST та ISO. В умовах зростаючих загроз цифровій безпеці вони забезпечують надійний фундамент для побудови систем автентифікації, верифікації цифрового контенту та зберігання доказів у юридичній практиці.

Алгоритми SHA-256 та SHA-3 вважаються сучасним стандартом криптографічного хешування завдяки їхній високій стійкості до криптоаналітичних атак, включаючи колізії та атаки на основі прообразів, що робить їх надійним інструментом для забезпечення цілісності файлів, автентифікації цифрових повідомлень, захисту електронних підписів, а також для використання в технологіях блокчейн та цифровій криміналістиці, де цілісність даних є критично важливою.

Таблиця 1 - Порівняльний аналіз алгоритмів хешування

Алгоритм	Довжина хешу (біт)	Рівень безпеки	Стійкість до колізій	Швидкодія	Застосування
MD5	128	Низький	Уразливий	Висока	Лише базова перевірка
SHA-1	160	Середній	Часткова	Висока	Обмежене використання
SHA-256	256	Високий	Висока	Середня	Перевірка цілісності файлів
SHA-3	256/512	Дуже високий	Дуже висока	Середня	Критичні системи, захист доказів

Як видно з таблиці 1, незважаючи на високу продуктивність алгоритм MD5 є повністю скомпрометованим алгоритмом, оскільки для нього вже давно виявлені колізії, що робить його непридатним для використання в системах безпеки. SHA-1, хоча й демонструє кращу стабільність, також не рекомендується для використання в сучасних рішеннях через часткові колізії, виявлені у 2017 році. Алгоритм SHA-256 забезпечує оптимальний баланс між продуктивністю та безпекою: він ефективно працює з великими зображеннями у форматах PNG та JPG, має високу стійкість до колізій та є фактичним стандартом для перевірки цілісності файлів у більшості операційних систем та мережевих протоколів [5].

SHA-3 характеризується підвищеною стійкістю до криптоаналітичних атак завдяки іншій структурі (модель губки). Доцільно використовувати його в системах, де довгострокова надійність є критично важливою, наприклад, у судових реєстрах, блокчейнах або при зберіганні цифрових доказів. Таким чином, SHA-256 можна вважати оптимальним рішенням для практичного контролю цілісності зображень, а SHA-3 – стратегічним вибором для майбутніх систем, орієнтованих на підвищення довіри та юридичної значущості цифрових даних[6].

Висновок. Аналіз підтверджує ефективність криптографічних хеш-функцій як універсального механізму виявлення підроблених або змінених зображень. Алгоритми SHA-256 та SHA-3 забезпечують оптимальне поєднання стійкості до колізій, продуктивності та захисту від сучасних атак. Їх інтеграція з цифровими підписами та технологіями блокчейн створює новий рівень безпеки для підтримки довіри до цифрових матеріалів у медіа, правовій та технічній сферах.

Перелік використаних джерел.

1. Verdoliva L. Media Forensics and DeepFakes: An Overview. IEEE Journal of Selected Topics in Signal Processing, 2020, Vol. 14, No. 5, pp. 910–932. DOI: 10.1109/JSTSP.2020.3002103

2. Столлінгс В. Криптографія та мережева безпека: принципи та практика. – Pearson, 2022.

3. Кучер В.І., Бондаренко І.В. Основи криптографії: підручник. – Київ: НАУ, 2020.

4 Menezes A., Van Oorschot P., Vanstone S. Handbook of Applied Cryptography. – CRC Press, 2021.

5. Daemen J., Rijmen V. The Design of Rijndael: AES - The Advanced Encryption Standard. – Springer, 2020.

6. NIST. Secure Hash Standard (SHS): FIPS PUB 180-4. – National Institute of Standards and Technology, 2015.

УДК 004.056.5

Якименко Є.В., Борисенко І.І.

Національний університет «Одеська політехніка»

МЕТОД МІНІМІЗАЦІЇ ЗБУРЕНЬ КОНТЕЙНЕРА НА ОСНОВІ ПОДВІЙНОГО АНАЛІЗУ

Вступ. На сьогодні, з метою забезпечення конфіденційності передавання даних, для приховування інформації, використовують системи цифрової стеганографії. Однією з основних вимог до цих систем – непомітність змін, які під час вбудовування повідомлення вносяться до контейнера.

Так звані збурення – це зміни структури контейнера. Вони призводять до спотворення його статистичних, візуальних характеристик, і чим більші спотворення, тим легше виявити наявність прихованої інформації методами стегоаналізу.

Мінімізація збурень контейнера – один з головних напрямів сучасних досліджень, оскільки стійкість і ефективність алгоритмів стеганографії визначає саме цей фактор.

Проблема полягає в тому, що більша частина відомих підходів аналізує тільки контейнер. Структура самих даних, що вбудовуються, не враховується. У результаті навіть незначні зміни можуть накопичуватися, створюючи помітні спотворення.

Мета: Розробка методу мінімізації збурень контейнера під час стеганографічного перетворення шляхом одночасного аналізу характеристик контейнера і структури даних, які вбудовуються, для підвищення рівня непомітності та збереження якості цифрового зображення.

Основна частина.

За основу роботи взято метод [1], в якому оцінюється збурення контейнера-зображення шляхом аналізу сингулярних чисел його матриці під час стеганографічного перетворення. Запропонований метод мінімізації збурень є його розвитком, що передбачає подвійний аналіз як самого контейнера, так і даних, які вбудовуються. Такий метод дозволяє виконувати адаптивне вписування інформації з мінімальним впливом на структуру контейнера.

В роботі [2] визначаються елементи контейнера, зміна яких впливає найменше на візуальну якість зображення. Для цього формується карта вартості зміни:

$$W = [w_{ij}], \tag{1}$$

де кожен елемент оцінює «ціну» зміни відповідного пікселя. Обчислюється вартість за модифікованою формулою:

$$w_{ij} = \alpha \cdot |m_{ij}| + \beta \cdot \Phi_{psy}(i,j) + \gamma \cdot \Phi_{tex}(i,j), \tag{2}$$

де m_{ij} – сингулярні коефіцієнти з SVD-контейнера,

$\Phi_{psy}(i,j)$ – психовізуальний коефіцієнт,

$\Phi_{tex}(i,j)$ – коефіцієнт текстурності,

α, β, γ – вагові коефіцієнти.

Мінімальні значення w_{ij} відповідають ділянкам, куди бажано виконувати

вписування повідомлення.

Паралельно для даних (повідомлення), що підлягають вбудовуванню, виконується попередній структурний аналіз.

Так для текстового повідомлення застосовується кодовий або частотний аналіз, який допомагає визначити найбільш повторювальні символи та оптимізувати їх розміщення у контейнері.

Частотний аналіз підраховує кількість появи однакових символів і типові бітові комбінації повідомлення; елементи, що з'являються найчастіше першими зіставляються з позиціями контейнера, де очікується мінімальна зміна, що зменшує кількість відкорегованих бітів.

Кодовий аналіз [3] враховує обраний спосіб кодування: оцінюються довжини та шаблони бітів, після чого порядок вписування підлаштовується під розподіл молодших бітів у контейнері.

Якщо повідомлення є зображенням або бінарним файлом, то використовується ентропійний або сингулярно-спектральний аналіз, принцип якого узгоджується з методом [1], але в даному випадку застосовується не лише для оцінки контейнера, а й для дослідження структури самого повідомлення.

Ентропійний аналіз [4] оцінює локальну складність (ентропію/дисперсію) блоків, оскільки у високотекстурних ділянках зображення зміни яскравостей пікселей менш помітні, тому саме у цих пікселей більший пріоритет для вписування. Цей підхід допомагає оцінити спектральну схожість об'єктів, мінімізувати різницю їхніх енергетичних характеристик.

На базі отриманих параметрів формується матриця схожості між бітами повідомлення й потенційними позиціями контейнера. Це забезпечує добір тих ділянок, де максимально узгоджуються локальні властивості обох об'єктів: даних та контейнера.

Після подвійного аналізу виконується процедура адаптивного вписування, яка проходить у три етапи.

На першому етапі біти повідомлення, що вже збігаються з бітами контейнера, вважаються вписаними, тобто без будь яких модифікацій матриці контейнера.

На другому – для часткових збігів, де різниця становить 1, змінюються лише елементи з мінімальними значеннями w_{ij} . Тобто корекція торкається тільки найменш «чутливих» елементів контейнера, що суттєво знижує середнє збурення.

На третьому – залишкові біти розміщують за класичними правилами, але враховується вагова карта, тобто відбувається компенсуюче вписування. Після кожної зміни елемента оцінюється величина локального спотворення, і корекція припиняється, якщо досягнуто допустимий поріг D_{max} .

На кожному етапі процес вписування адаптується до локальних характеристик контейнера, що забезпечує плавне розподілення змін та мінімальний рівень помітності.

Загальне збурення контейнера оцінюється функцією [3]:

$$D = \sum_{ij} w_{ij} \cdot |c'_{ij} - c_{ij}|, \quad (3)$$

де c_{ij} – початкові значення елементів контейнера,

а c'_{ij} – модифіковані після вписування.

Метою є мінімізація D при збереженні повноти і цілісності переданого повідомлення.

Для оцінки ефективності запропонованого методу проведено модельне порівняння з відомими алгоритмами LSB, LSB Matching та GRAPH_Matching. Порівняння проводилося за показниками середнього збурення D, а також за метриками якості PSNR і SSIM (Таблиця 1).

Таблиця 1 – Отримані експериментальні дані

Метод	Середнє збурення D	PSNR, дБ	SSIM
LSB	2.1358	40.12	0.926
LSB Matching	1.8325	41.80	0.951
GRAPH_Matching [5]	1.6010	42.37	0.961
Адаптивний (запропонований метод)	1.0765	43.21	0.972

Експериментальні результати показали, що використання подвійного аналізу допомагає зменшити середнє збурення в 1,5–2 рази порівняно з класичним LSB-методом. При цьому показник PSNR підвищується в середньому на 2–4 дБ, а SSIM наближається до 1, що підтверджує високу непомітність і якість отриманого стеганоконтейнера.

Висновок. Розроблено метод мінімізації збурень контейнера під час стеганографічного перетворення, що ґрунтується на подвійному аналізі: даних і контейнера. Запропонований підхід поєднує сингулярно-спектральну оцінку чутливості контейнера з аналізом структури повідомлення, дозволяючи адаптивно підбирати області вписування та мінімізувати сумарне спотворення. Отримані результати можуть бути використані для удосконалення існуючих стеганографічних систем і створення нових методів приховування даних з підвищеною непомітністю та стійкістю.

Перелік використаних джерел.

1. Борисенко И. И. Оценка возмущения контейнера при его стеганопреобразовании // Високі технології в машинобудуванні, 2015, випуск 1 (25). С.127-132.

2. Pitas I. Digital Image Processing Algorithms. – New York: John Wiley & Sons, 1993. – 432 p.

3. Fridrich J. Steganography in Digital Media: Principles, Algorithms, and Applications. – Cambridge: Cambridge University Press, 2009. – 336 p.

4. Sparavigna A.C. Entropy in Image Analysis II. – Basel : MDPI, 2023. – 356 p.

5. Борисенко І.І. Застосування теорії графів в задачах створення стеганографічних повідомлень / І.І. Борисенко // Сучасна спеціальна техніка. – 2015. – №2. С. 26-33.

UDK 004.056.5

Lidiia TYMOSHENKO, Anna YAKYMOVA, Irina NAZAROVA

Odesa Polytechnic National University

DEVELOPMENT OF AN APPLICATION FOR THE CRYPTOGRAPHIC PROTECTION OF AUDIO STREAMING SERVICES CONSIDERING COMPRESSION CODECS

Introduction. The modern world actively utilizes audio streaming services, which creates new challenges in the field of transmitted information protection. Audio streams often contain confidential or copyrighted data that require reliable protection against interception, tampering, or unauthorized access. The use of compression codecs adds particular complexity, as they alter the structure of audio data and affect compatibility with cryptographic methods [1].

Objective. The objective of this thesis is to develop an application for the cryptographic protection of audio streaming services, considering the specifics of compression codecs. The relevance of this study is driven by the need to protect transmitted audio data from unauthorized access and ensure their integrity while using compression methods.

One of the most effective methods for ensuring confidentiality and integrity is the AES algorithm in GCM mode [2]. Figure 1 shows a general scheme of encryption using the AES-GCM algorithm.

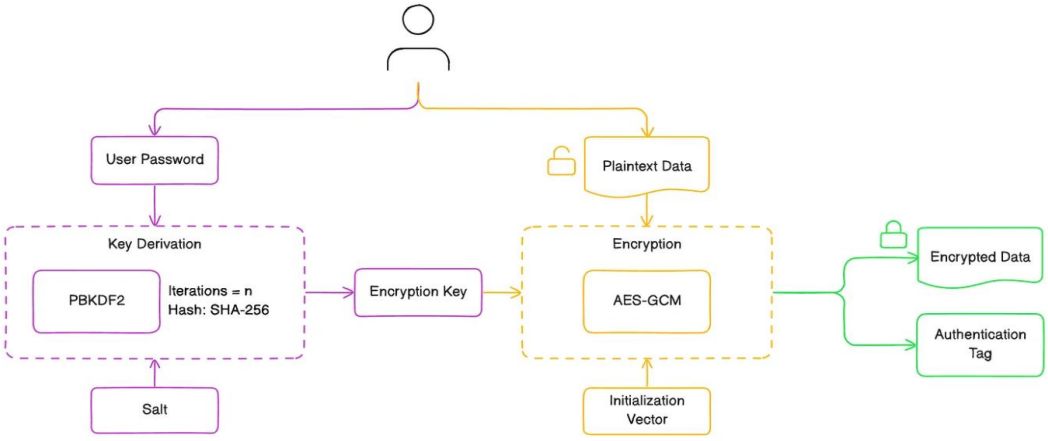


Figure 1 – AES-GCM encryption

This mode allows simultaneous encryption of data and verification of its authenticity through a tag. It is suitable for real-time data stream processing due to its high speed and support for parallel block processing. The reliability of the algorithm relies on the use of a unique initialization vector and authentication tag, which prevents replay or tampering attacks [3].

Figure 2 schematically depicts the generalized process of data decryption and authentication.

The combination of cryptography with the specific features of compression codecs plays a significant role. Encrypting audio before or after compression requires a precise understanding of stream structure changes in order to avoid quality loss and ensure compatibility. It is important to ensure such processing does not affect

transmission time, reduce performance, or compromise playback quality [4].

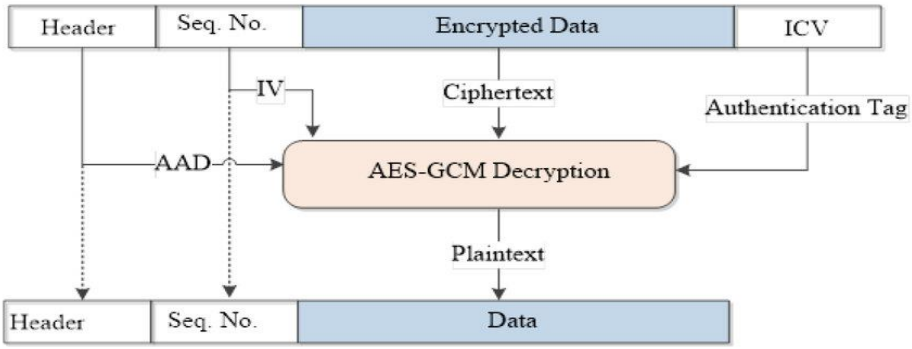


Figure 2 – AES-GCM Decryption

Special attention should be given to synchronization between the sender and receiver. Each encrypted packet must contain an initialization vector, a sequence number, and an authentication tag. This structure ensures correct decryption, enables detection of packet loss, and provides protection against replay attacks. Ensuring the uniqueness of vectors and reliable encryption key management are essential security conditions.

It is worth noting that AES-GCM has found widespread adoption among leading technology companies. Google uses it to protect data in its cloud platform and Chrome browser traffic, Amazon uses it in its AWS services to encrypt information, Microsoft uses this approach in Azure, and Apple protects messages in iMessage and data in iCloud. All of these companies have chosen AES-GCM because of its efficiency, speed, and comprehensive approach to data security. To achieve this goal, an analysis of modern cryptographic protection methods for audio streams was conducted, with a particular focus on the AES-GCM algorithm, which provides efficient encryption in streaming mode. The study includes the implementation of encryption and decryption algorithms for audio streams.

The advantages of AES-GCM include compatibility with streaming transmission, minimal latency, high speed, and resistance to common attacks. The effective implementation of this method in audio streaming services helps maintain a balance between security, performance, and user experience quality.

Conclusions. The main result of the research is the development of a software application that ensures the cryptographic protection of audio streaming without significantly affecting system performance or audio playback quality.

List of sources.

1. Stallings W. Cryptography and Network Security: Principles and Practice. 8th ed. Boston: Pearson, 2020. 752 p.

2. National Institute of Standards and Technology. FIPS PUB 197: Advanced Encryption Standard (AES) / NIST. 2001. 51 p.

3. Dworkin M.J. Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC. NIST Special Publication 800-38D. Gaithersburg, MD: National Institute of Standards and Technology, 2007. 56 p.

4. Baccour L., Atri M. Security Challenges in Multimedia Streaming Services: A Survey. Multimedia Tools and Applications. 2022. 27973 p.

УДК 681.32

Прищепа О.І.

Національний університет »Одеська політехніка»

ДОСЛІДЖЕННЯ МЕТОДІВ ВИЯВЛЕННЯ ФАЛЬСИФІКАЦІЇ ЗОБРАЖЕНЬ

Вступ. Розвиток сучасних технологій характеризується постійним зростанням значення інформації та її широким використанням у різних сферах діяльності людини. Природно, що задачі інформаційної безпеки набувають особливої актуальності. В сучасних умовах створення, зберігання та передачі інформації в електронному вигляді виникає необхідність перевірки цілісності цифрових сигналів, зокрема, цифрових зображень (ЦЗ), що обумовлено бурхливим розвитком програмних засобів для створення та редагування цифрових зображень, які дають можливість для фальсифікації. Під фальсифікацією розумітимемо навмисне порушення цілісності цифрового зображення.

Мета: Експериментальні дослідження методів виявлення фальсифікації зображень для розробки практичного методу локалізації фальсифікації, задля підвищення безпеки передачі інформації.

1. Аналіз поширених методів виявлення фальсифікації в цифрових зображеннях

Різноманітність методів виявлення фальсифікації ЦЗ обумовлена широким вибором засобів та умов проведення фальсифікації. Для виявлення заміни деякої частині ЦЗ на іншу частину цього ж зображення запропоновано метод лексикографічного впорядковування коефіцієнтів дискретного косинусного перетворення (ДКП) блоків ЦЗ. Однакові групи відповідають однаковим просторовим областям і можуть свідчити про наявність фальсифікації [1].

Проте, цей метод не може використовуватися, якщо деяка область одного ЦЗ замінюється на частину іншого зображення. Для детектування фальсифікації в цьому випадку проводиться аналіз статистичної характеристики третього порядку в частотної області. Недоліком методу є неможливість визначення вставки, якщо вона проведена на границі двох текстур. Існують також методи виявлення фальсифікації, що використовують аналіз кореляції між пікселями ЦЗ [2].

Такі геометричні перетворення ЦЗ, як масштабування і поворот, викликають появу періодичної кореляції, яка не властива оригінальним зображенням і вказує на фальсифікацію. Однак ці методи не дають результатів, якщо зображення зберігається з використанням стиснення. Більшість сучасних цифрових фотоапаратів використовують формат JPEG [1] (із втратою інформації). Тому сучасні методи виявлення фальсифікації пов'язані з аналізом коефіцієнтів ДКП матриці ЦЗ. У стандартній схемі стиснення JPEG кольорове зображення (RGB) спочатку переводиться у простір YCbCr. Кожний канал ділиться на блоки 8x8 пікселів та, використовуючи ДКП, отримуємо частотний спектр матриці ЦЗ. Кожний коефіцієнт ДКП, f_i, квантується з якістю q:

$$f_i^q = \left[\frac{f_i}{q}\right] \cdot q. \quad (1)$$

2. Дослідження функції квадрата середньоквадратичного відхилення

Метод базується на дослідженні функції квадрата середньоквадратичного відхилення значень коефіцієнтів ДКП від значень повторно відквантованих коефіцієнтів ДКП матриці ЦЗ з різними коефіцієнтами квантування. Для визначення значення кроку квантування треба побудувати графік функції:

$$F(q) = \sum_{i=1}^{n} \left(f_i - f_i^q\right)^2, \quad (2)$$

де n – кількість коефіцієнтів ДКП, що відповідають заданій частоті; f_i – коефіцієнт ДКП; f_i^q визначається за формулою (1); $q \in (1; 30]$.

У результаті проведення обчислювального експерименту, виявилось, що наявність локальних мінімумів функції (2) залежить не лише від наявності або відсутності фальсифікації ЦЗ, але від умов її проведення. Наявність фальсифікації І-го типу передбачає появу другого локального мінімуму. При фальсифікації ІІ-го типу поява другого локального мінімуму функції (2) пояснюється стисненням деякої частини ЦЗ з використанням коефіцієнту квантування β. Крім того, значення функції (2) в локальному мінімумі, що відповідає коефіцієнту квантування α, стане більше, ніж відповідне значення в оригінальному ЦЗ. Якщо проведена фальсифікація ІІІ-го типу, разом з локальним мінімумом γ, відповідним останньому стисненню сукупного ЦЗ, передбачається наявність третього локального мінімуму, відповідного коефіцієнту квантування β і зменшення локального мінімуму, відповідного коефіцієнту квантування α.

Один із результатів обчислювального експерименту наведено на рисунку 1.

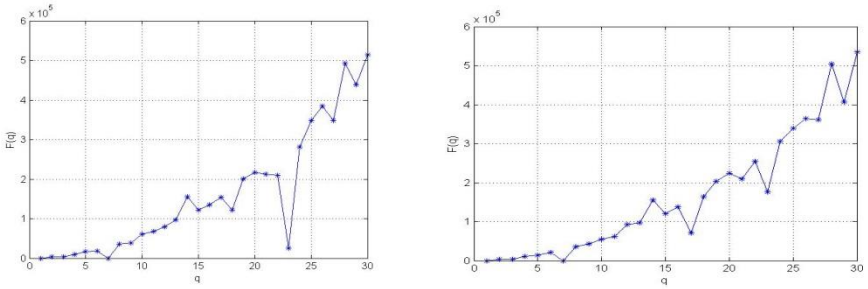


Рисунок 1 – Графік функції F(q) (а) – для оригінального ЦЗ із коефіцієнтами квантування α=23, γ=7; (б) – для фальсифікації ЦЗ І-го типу з коефіцієнтами квантування α=23, β=17, γ=7

Висновок. Проведені дослідження дозволяють зробити висновок, що: якщо 2 коефіцієнта квантування виявляться кратними, то на графіку функції (2) виникають додаткові локальні мінімуми в значеннях q, кратних коефіцієнтам; за наявності фальсифікації графік функції показує - третій коефіцієнт квантування, відповідний фальсифікованій області, однозначно не визначається; коефіцієнт квантування, відповідний останньому стисненню всього ЦЗ, визначається як локальний мінімум функції зі значенням F(q) = 0.

Перелік використаних джерел.

1. Конахович Г., Прогонов Д., Пузиренко О. Комп'ютерна стеганографічна обробка й аналіз мультимедійних даних.К.:Центр навчальної літератури,2018. 558с.

2. Кобилін О.А., Творошенко І.С. Методи цифрової обробки зображень: навч. посібник. Харків: ХНУРЕ, 2021.124 с..

УДК 004.056

Петровська М.Г., Кушніренко Н.І.

Національний університет «Одеська політехніка»

СИСТЕМА АВТОМАТИЗОВАНОГО МОНІТОРИНГУ КІБЕРЗАХИЩЕНОСТІ ВЕБ-ЗАСТОСУНКІВ

Вступ. У сучасному цифровому світі кількість веб-застосунків у бізнесі, державному управлінні, освіті та повсякденному житті стрімко зростає. Водночас збільшується й кількість кіберзагроз, що створює потребу у впровадженні ефективних методів контролю та моніторингу кіберзахищеності.Веб-застосунки є основною ціллю зловмисників, тому їхній захист - один із ключових напрямів кібербезпеки. Ручні методи перевірки вже не забезпечують потрібної оперативності та масштабованості, тож автоматизовані рішення стають основним інструментом для оцінювання рівня захищеності [1].

Мета: Підвищення рівня безпеки веб-застосунків шляхом створення системи автоматизованого моніторингу та оцінювання їх кіберзахищеності.

Основна частина

У ході дослідження проаналізовано основні існуючі рішення для перевірки безпеки веб-застосунків. Найпоширенішими інструментами є OWASP ZAP, Nikto, Burp Suite, Nmap, які дозволяють здійснювати як базове, так і розширене тестування на наявність вразливостей. Кожен із них має власну сферу застосування та особливості.

Результати порівняльного аналізу наведено у таблиці 1:

Таблиця 1 – Порівняння інструментів для аналізу безпеки веб-застосунків

Назва	OWASP ZAP	Nikto	Burp Suite	Nmap
1	2	3	4	5
Основне призначення	Розробка стандартів, рекомендацій, інструментів для веббезпеки	Перевірка вебсерверів на відомі вразливості	Перехоплення, аналіз та автоматизація тестування вебзапитів	Виявлення хостів, портів, служб у мережі
Основні функції	OWASP Top 10, ASVS, ZAP	Виявлення старих версій серверів, небезпечних скриптів	Перехоплення трафіку, автоматичне сканування, тестування XSS, SQLi	Порт-сканування, виявлення ОС, служб, топології мережі
Інтерфейс	Вебсайт, документи, іноді GUI в проєктах (наприклад, ZAP)	CLI (Командний рядок)	GUI	CLI, графічна версія Zenmap

1	2	3	4	5
Ліцензія / доступність	Відкрита (Open Source)	Відкрита (Open Source)	Відкрита (Open Source)	Безкоштовна і платні версії
Тип звітності	Стандарти, рекомендації, шаблони	Текстовий звіт (TXT, HTML)	Текстовий звіт (TXT, HTML)	Детальні інтерактивні звіти

Виходячи з аналізу, саме OWASP ZAP є оптимальним вибором для розробки системи автоматизованого моніторингу та оцінювання кіберзахищеності веб-застосунків, оскільки він дозволяє інтегрувати власні модулі, змінювати вихідний код, отримувати результати й обробляти їх у власній системі, зокрема із подальшою обробкою модулем AI, а також розширювати функціональність під конкретні потреби наукового дослідження.

Розроблений підхід передбачає використання інтегрованої системи, що поєднує автоматичне сканування вразливостей із контекстним аналізом ризиків. Система дозволяє виконувати періодичний моніторинг стану безпеки веб-застосунків, класифікувати виявлені загрози за рівнем критичності та формувати звіти у зручному форматі (дашборди, таблиці, графіки) [3].

Методологічною основою оцінювання рівня безпеки є стандарти OWASP Top 10 та CVSS [2], що дають змогу визначати критичність знайдених вразливостей і пріоритетність їх усунення. Автоматизований підхід забезпечує оперативність, об'єктивність та зменшує вплив людського фактора.

Система може бути інтегрована у процеси DevSecOps, що забезпечує безперервний моніторинг безпеки під час розробки та оновлення веб-застосунків.

Очікуваним результатом роботи є веб-застосунок, який забезпечує:

- автоматичне виявлення вразливостей веб-застосунків;
- оцінювання рівня кіберзахищеності за обраними критеріями;
- візуалізацію результатів у вигляді інтерактивних звітів;
- формування рекомендацій щодо усунення знайдених проблем.

Висновок. Розробка методів та засобів автоматизованого моніторингу кіберзахищеності веб-застосунків є актуальним напрямом у сфері інформаційної безпеки. Запропонований підхід дозволяє поєднати технічний аналіз із системним оцінюванням ризиків, що підвищує ефективність виявлення вразливостей і знижує ймовірність успішної реалізації кібератак на веб-застосунки. Отримані результати можуть бути використані для подальшої розробки інтегрованих рішень у межах корпоративних систем кіберзахисту та як основа для подальших наукових досліджень у галузі кібербезпеки.

Перелік використаних джерел.

1. OWASP Foundation. OWASP Top 10 – 2021: The Ten Most Critical Web Application Security Risks. – Режим доступу: https://owasp.org/www-project-top-ten/

2. FIRST Organization. Common Vulnerability Scoring System (CVSS) v3.1 Specification Document. – Режим доступу: https://www.first.org/cvss/

3. NIST Special Publication 800-115. Technical Guide to Information Security Testing and Assessment. – National Institute of Standards and Technology, 2008.

УДК 004.056.53

Капелюшний В.Р., Кушніренко Н.І., Троянський О.В.

Національний університет «Одеська політехніка»

РОЗРОБКА ЗАХИЩЕНОЇ СИСТЕМИ ДЛЯ СТВОРЕННЯ ТА ПРОВЕДЕННЯ ОПИТУВАНЬ

Вступ. У сучасних умовах активного розвитку дистанційного навчання та цифровізації освітнього процесу зростає потреба у створенні безпечних платформ для проведення онлайн-опитувань та тестування. Більшість наявних сервісів (Google Forms, Quizlet, SurveyMonkey, Typeform) зручні у використанні, але не гарантують повного захисту даних користувачів, що може призвести до витоку інформації, маніпуляцій результатами або підміни облікових записів.

Постає завдання розробити спеціалізовану систему, що не лише забезпечує комфортне проходження тестів, а й реалізує комплексну кіберзахисну інфраструктуру – від шифрування даних до контролю академічної доброчесності.

Мета: Розробити захищену веб-систему для створення та проведення онлайн опитувань, що забезпечує конфіденційність, цілісність і доступність даних за рахунок криптографічних та організаційних механізмів безпеки.

Основна частина

Було проведено порівняльний аналіз популярних платформ для опитувань, виявлено їх переваги та недоліки. Google Forms має простий інтерфейс і надійне з'єднання через HTTPS, проте не дозволяє контролювати поведінку користувача чи забезпечити захист контенту після розповсюдження [1]. Quizlet орієнтований на інтерактивне навчання, але не забезпечує шифрування даних користувачів у стані спокою [2]. SurveyMonkey має добру аналітику, однак більшість функцій безпеки доступна лише у платній версії [3]. Typeform приваблює дизайном, але має обмеження на обсяг даних і мінімальний захист [4].

Отже, актуальним є створення власної системи, у якій рівень безпеки не поступається комерційним рішенням, а функціонал враховує потребу освітнього середовища. Проєкт побудовано за трирівневою клієнт-серверною моделлю, що включає: інтерфейс користувача, який реалізований з використанням HTML, CSS і Bootstrap; серверну частину на базі Python [5] та Django [6], що відповідає за обробку запитів, авторизацію та логіку безпеки; базу даних – реляційну структуру для зберігання курсів, тестів, результатів і логів безпеки.

Архітектура передбачає розмежування доступу до ресурсів системи через механізм RBAC [7], що дозволяє обмежувати дії користувачів відповідно до їхніх ролей (студент, викладач, адміністратор).

В системі передбачено механізм захисту інформації який реалізується комплексом технічних засобів безпеки: шифрування даних за допомогою алгоритму AES-256 у режимі EAX, що забезпечує захист у стані спокою та під час передавання даних; двофакторну аутентифікацію через додаток Google Authenticator; авторизацію через Google для спрощеного й безпечного входу; підтвердження критичних змін через email-OTP-коди; перевірку геолокації користувача при вході; автоматичне завершення сесії після 12 години активності;

захист від XSS-атак через санітизацію HTML та фільтрацію даних.

Окрім цього, передбачено введення журналу входів і дій користувачів, що дозволяє проводити аудит безпеки.

Забезпечення академічної доброчесності також приділено велику увагу. Реалізовано функціонал вбудовування цифрових водяних знаків у тестові завдання, сформованих шляхом хешування персональних даних користувача (ID, прізвище, ім'я та дату народження користувача) за допомогою алгоритму BLAKE2b. Отриманий хеш-код вбудовується у фон сторінки та в текст запитань. Це дозволяє ідентифікувати джерело витоку завдань у разі публікації, створити психологічний бар'єр для студентів щодо порушення правил, забезпечити прозорість та доказовість у розслідуванні порушення академічної доброчесності.

Додатково реалізовано захист від спроб копіювання, відкриття консолі браузера, зняття скріншотів і використання комбінації клавіш. Такі функції реалізуються через події JavaScript.

Інтерфейс та функціональні можливості системи підтримують різні ролі користувачів: викладач, може створювати курси, формувати опитування, переглядати статистику проходження, обмежувати кількість спроб і контролювати академічну доброчесність; студент може приєднатися до курсів, проходити опитування та переглядати результати. Інтерфейс адаптований для будь-яких пристроїв і має просту навігацію.

Висновок. Розроблена система поєднує сучасні підходи до веброзробки та кіберзахисту, забезпечує безпечне та зручне проведення опитувань. Реалізація водяних знаків, двофакторної ідентифікації та шифрування даних підвищує довіру до онлайн опитувань.

Система може бути впроваджена у закладах освіти та організаціях, де критично важливим є захист персональних даних і прозорість результатів тестування. У перспективі – додавання системи поведінкового аналізу, розширення типів завдань і підтримка мультимовності.

Перелік використаних джерел.

1. Google Форми: онлайн-редактор форм. [Електронний ресурс]. - Режим доступу: https://workspace.google.com/products/forms/

2. Навчальні інструменти, картки та рішення з підручників. [Електронний ресурс]. - Режим доступу: https://quizlet.com/

3. SurveyMonkey: The World's Most Popular Survey Platform. [Електронний ресурс]. - Режим доступу: https://www.surveymonkey.com/?msockid=35a01cc759c3678138bb09d358d166a0

3. Typeform: People-Friendly Forms and Surveys. [Електронний ресурс]. - Режим доступу: https://www.typeform.com/

5. Matthes E. Python Crash Course: A Hands-On, Project-Based Introduction to Programming. San Francisco: No Starch Press, 2019. ISBN 978-1-59327-928-8.

6. Django documentation. [Електронний ресурс]. - Режим доступу:: https://docs.djangoproject.com/en/5.2/

7. Cruz J. P., Kaji Y., Yanai N. RBAC-SC: Role-based access control using smart contract // IEEE Access. 2018. P. 12240–12251.

УДК 004.415.25

Львов І.Д.

Національний університет «Одеська політехніка»

ПРОЄКТУВАННЯ ТА РЕАЛІЗАЦІЯ UX/UI ВЕБСАЙТУ SUITEBOT ЯК ЦИФРОВОГО ОНЛАЙН-АСИСТЕНТА

Вступ. У сучасному цифровому середовищі успіх вебпродукту значною мірою визначається якістю користувацького досвіду (UX) і візуальної реалізації інтерфейсу (UI). Теоретичні основи взаємозв'язку між поведінкою користувача та дизайном інтерфейсу описані у класичних працях, що підкреслюють важливість ментальних моделей і спрощення взаємодії [1]. Концептуальна модель багаторівневого проєктування дозволяє розділяти завдання на логічні щаблі - від контентної стратегії до детального візуального оформлення [2]. Практичні принципи юзабіліті акцентують увагу на простоті й очевидності елементів управління, що критично для лендингових сторінок та промо-сайтів [5]. З огляду на зазначене, виникає потреба у формалізації UX/UI-процесу та демонстрації його ефективного застосування у реальному проєкті.

Мета: Проаналізувати та задокументувати повний UX/UI-процес розробки вебсайту Suitebot.ai, обґрунтувати вибір методів і інструментів, а також показати практичну реалізацію дизайнерських рішень від прототипу до фронтенд-впровадження.

Основна частина

Методологія поєднувала огляд фахової літератури, опитування, побудову персонажів і сценаріїв, поетапне прототипування у Figma і побудову дизайн-системи. Для валідації зроблених рішень застосовано принципи швидких ітерацій, що відповідають сучасним підходам. У процесі дослідження особливу увагу приділено принципам, визначеним Дональдом Норманом [1] та Якобом Нільсеном [4], зокрема законам зворотного зв'язку, консистентності й візуальної ієрархії. Фронтенд-реалізацію виконано на основі адаптивної сітки Bootstrap, анімаційні елементи експортувалися як .lottie через сервіс Jitter і інтегрувалися з використанням бібліотеки Lottie.js. Це дозволило створити плавні інтерактивні ефекти без збільшення ваги сторінки. Принципи доступності оцінювалися відповідно до рекомендацій WCAG 2.1 [7], що дало змогу забезпечити комфортне сприйняття контенту для користувачів із різними можливостями. Окрему увагу приділено ролі штучного інтелекту у формуванні інтерфейсних рішень і контент-логіки онлайн-асистента, який є центральним продуктом сайту.

Проєктування починалося з дослідження цільової аудиторії і формування користувацьких сценаріїв, що зумовили структуру інформаційної архітектури сайту. На початковому етапі створено дві варіації Lo-Fi wireframes для тестування різних підходів до воронки уваги. Далі підготовлено Mid-Fi макети з уточненням розташування текстів і функціональних елементів. Hi-Fi прототипи відтворювали остаточну візуальну концепцію із вибором шрифту Outfit, опрацюванням кольорової палітри, до якої входять нейтральні, основні акценти, статусні кольори, а також набором компонентів. Дизайн-система містила специфікації для кнопок, форм, карток і соціальних іконок із деталізацією станів default, hover, active, disabled. Завдяки цьому забезпечено повторюваність і масштабованість

візуальних рішень [3].

Для підтвердження контрастності між текстом і фоном застосовано онлайн-інструмент Colour Contrast Checker, результати якого відповідали стандарту WCAG AA. Front-end реалізація передбачала застосування Bootstrap для адаптивної сітки і стандартних компонентів. Секція FAQ реалізована як акордеон із використанням нативних компонентів фреймворку. Для збору заявок інтегровано Typeform, що надало гнучкість у зміні полів та зборі аналітики. Інтерактивна анімація смартфона в Hero-блоці створена у Figma, експортована через Jitter у формат .lottie та підключена за допомогою бібліотеки Lottie.js. Це рішення дало змогу досягти сучасної візуальної динаміки без навантаження на сервер і зберегти плавність роботи сайту. У поєднанні з анімацією заголовку Hero-блок формує перше враження й утримує увагу користувача.

У результаті реалізації отримано функціональний лендинг Suitebot.ai, який поєднує естетичність, ефективність та технічну легкість. Його структура включає Hero-блок з інтерактивною анімацією, секцію довіри з логотипами клієнтів, блок переваг у вигляді карток, покроковий розділ How it Works, секцію FAQ, інтегровану форму заявки та футер із навігацією й контактами. Розроблена дизайн-система гарантує консистентність усіх елементів та полегшує масштабування продукту в майбутньому. Базова перевірка на відповідність принципам доступності показала коректні показники контрастності для основних кольорів відповідно до WCAG 2.1 [7].

Практичним результатом стало збереження логіки воронки уваги навіть при адаптації для мобільних пристроїв, що доводить ефективність прийнятого UX-рішення. Отримані результати можуть використовуватися як еталонна модель для створення сайтів AI-сервісів, де швидкість реалізації й узгодженість дизайну відіграють ключову роль.

Висновок. Системний UX/UI-процес підтвердив свою ефективність у контексті проєктування промо-сайту для онлайн-асистента. Поєднання класичних принципів дизайну з сучасними інструментами та підходами Lean UX дозволяє створювати інтерфейси, які одночасно відповідають вимогам користувачів і бізнес-цілям.

Перелік використаних джерел.

1. Norman D. A. The Design of Everyday Things. - Revised and expanded edition. New York : Basic Books, 2013. - 368 с.

2. Garrett J. J. The Elements of User Experience: User-Centered Design for the Web and Beyond. - 2nd ed. Berkeley : New Riders, 2010. - 190 с.

3. Nielsen Norman Group. Wireframes and Prototypes. - Nielsen Norman Group, 2015.

4. тNielsen J. 10 Usability Heuristics for User Interface Design. - Nielsen Norman Group, 1994 (updated ed.).

5. Krug S. Don't Make Me Think, Revisited: A Common Sense Approach to Web Usability. - 3rd ed. Berkeley : New Riders, 2014. - 216 с.

6. Gothelf J., Seiden J. Lean UX: Applying Lean Principles to Improve User Experience. - Sebastopol : O'Reilly Media, 2013. - 192 с.

7. W3C. Web Content Accessibility Guidelines (WCAG) 2.1. - W3C Recommendation, 2018.

8. Nielsen Norman Group. AI in UX Design. - Nielsen Norman Group, 2023.

УДК 004.056.55

Садченко А.В., Кушніренко О.А.

Національний університет «Одеська політехніка»

ПЕРЕВІРКА НАДІЙНОСТІ КРИТЕРІЇВ ПОРІВНЯННЯ БІОМЕТРИЧНИХ ЗОБРАЖЕНЬ

Вступ. У сучасних інтелектуальних системах безпеки, біометричної автентифікації [1] та моніторингу дедалі ширше використовуються технології розпізнавання обличь на базі моделей штучного інтелекту. Проте ефективність і точність таких систем значною мірою залежать від стабільності умов зйомки. Одним із ключових факторів, що впливають на якість ідентифікації, є зміна експозиції зображення – коливання рівня освітленості, контрасту та яскравості, які можуть змінювати візуальні особливості обличчя.

Нестабільна експозиція спричиняє появу шумів, втрату деталей у темних або пересвічених ділянках, що ускладнює процес виділення характерних ознак і порівняння біометричних шаблонів. Унаслідок цього класичні алгоритми розпізнавання, які базуються на інтенсивності пікселів, демонструють зниження точності. Отже, виникає необхідність у розробці алгоритмів, стійких до змін освітлення, здатних адаптивно обробляти зображення та забезпечувати надійну ідентифікацію навіть за несприятливих умов. Додатково на точність розпізнавання впливають зашумлення зображення та його афінні перетворення, зокрема обертання відносно центру мас.

Мета: Дослідження та експериментальна перевірка надійності критеріїв порівняння біометричних зображень, що використовуються в системах розпізнавання особи, з урахуванням впливу зовнішніх факторів – зміни освітлення, зашумлення та афінних перетворень. Дослідження спрямоване на визначення найбільш стійких показників подібності, здатних забезпечити високу точність ідентифікації за різних умов зйомки [2]**.**

Основна частина

Основні критерії, що застосовуються у системах аналізу зображень наступні: коефіцієнт кореляції [3]; середньоквадратична похибка (Mean Squared Error, *MSE*) [3]; пікове відношення сигнал / шум [3]; структурна подібність (Structural Similarity Index, *SSIM*) [4]; евклідова відстань (Euclidean Distance) [3]; косинусна подібність (Cosine Similarity) [4]. Алгоритм тестування якості критеріїв оцінювання відмінності між зображеннями, що реалізований в середовищі MATLAB має наступний вигляд.

Крок 1. Зчитування двох зображень *A* та *B* із файлів.

Крок 2. Якщо розміри різняться то виконуємо масштабування до зображення з найменшим розміром.

Крок 3. Якщо зображення кольорові то робимо переведення їх у відтінки сірого.

Крок 4. Перетворення двовимірних масивів у формат *double* для обчислень, де *double* – формат представлення числа з плаваючою комою, що займає в пам'яті 64 біти, або 8 байт.

Крок 5. Обчислюємо коефіцієнт кореляції як: $r = \mathrm{corr2}\ (A, B)$

Крок 6. Обчислюємо середньоквадратичну похибку:

$$MSE = \text{mean}((\text{A}(:) - \text{B}(:)).\text{^}2$$

Крок 7. Обчислюємо пікове відношення сигнал / шум:

$$PSNR = 10 * log10(L\text{^}2 \,/\, MSE), L = 255$$

Крок 8. Обчислюємо структурну подібність:

$$SSIM_{\text{val}} = \text{ssim}\left(\text{uint8}(A), \text{uint8}(B)\right)$$

Крок 9. Обчислюємо евклідову відстань:

$$Euc = \text{sqrt}(\text{sum}((A(:) - B(:)).\text{^}2))$$

Крок 10. Обчислюємо косинусну подібність:

$$CosSim = \text{dot}(A(:), B(:)) \,/\, (\text{norm}(A(:)) * \text{norm}(B(:)))$$

Крок 11. Вивід результатів на екран ПК.

Перевіримо стійкість критеріїв порівняння зображень в умовах зміни яскравості (рисунок 1 а)), під впливом імпульсного шуму із дисперсією σ = 0.2 (рисунок 1 б)), поворотом в процесі зйомки (рисунок 1 в)), а також одночасного зашумлення, поворот в процесі зйомки та зміні яскравості (рисунок 1 г)).

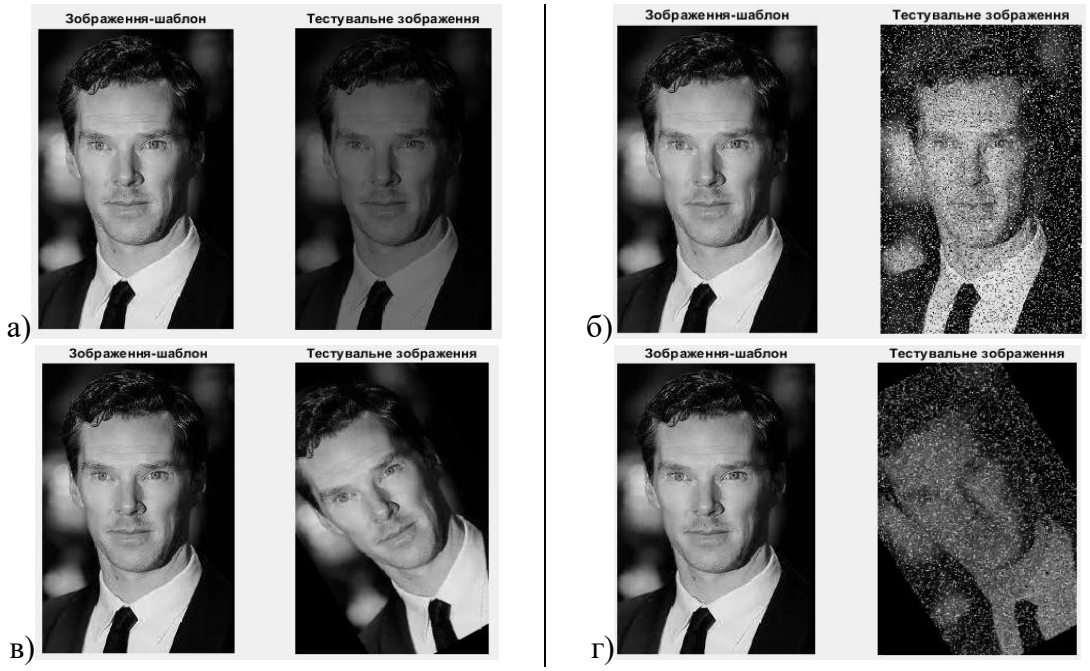


Рисунок 1 Тестове зображення: а) при недотримці експозиції (50% яскравості); б) при впливі імпульсного шуму із дисперсією $\sigma = 0.2$; в) при умові повороту на кут 30°; г) при умові одночасного повороту на кут 30°, впливу імпульсного шуму із дисперсією $\sigma = 0.2$ та зміні яскравості на 50%

Результати розрахунків коефіцієнтів подібності щодо всіх тестових випадків для зображення розміром 512 × 512 пікселів приведені в таблиці 1

Таблиця 1 – Отримані експериментальні дані

Тестовий випадок	Коефіцієнт кореляції, *r*	Середньо-квадратична похибка, *MSE*	Пікове відношення сигнал/шум *PSNR, дб*	Індекс структурної подібності, *SSIM*	Евклідова відстань, *D*	Косинусна подібність, *CosSim*
Зміна яскравості	1	2999	13,36	0,74	12293	1

Вплив імпульсного шуму	0,8	2339	14,4	0,51	10856	0,91
Поворот зображення (30^0)	0,39	7186	9,5	0,2	19026	0,7
Зміна яскравості, вплив імпульсного шуму, поворот зображення	0,32	6433	10	0,05	18002	0,7

Висновок. У ході проведеного дослідження було здійснено аналіз та перевірку надійності різних критеріїв порівняння біометричних зображень, які застосовуються в системах розпізнавання обличчя. Реалізований у середовищі MATLAB алгоритм дозволив оцінити поведінку таких критеріїв, як коефіцієнт кореляції, середньоквадратична похибка (*MSE*), пікове відношення сигнал/шум (PSNR), індекс структурної подібності (*SSIM*), евклідова відстань та косинусна подібність, за різних умов зміни зображення.

Отримані результати показали, що:

– при зміні експозиції (зниженні яскравості) найбільш чутливими виявились *MSE*, *SSIM* та евклідова відстань, тоді як коефіцієнт кореляції та косинусна подібність демонстрували вищу стійкість;

– при впливі імпульсного шуму найбільш стабільними критеріями були коефіцієнт кореляції та косинусна подібність;

– при повороті зображення на заданий кут найменше спотворення результатів спостерігалося для коефіцієнта косинусної подібності;

– при поєднанні кількох впливів (зміна яскравості, шум, поворот) коефіцієнт косинусної подібності зберіг найвищу стійкість, тобто його значення залишалось найбільш наближеним до одиниці.

Таким чином, проведені експерименти підтвердили, що косинусна подібність є найнадійнішим критерієм для порівняння біометричних зображень за наявності комплексних зовнішніх впливів. Використання цього критерію доцільно на початкових етапах ідентифікації, тоді як для уточнення результатів можуть застосовуватись інші показники подібності – зокрема кореляційний або структурний. Це дозволяє підвищити точність і стійкість систем розпізнавання обличь до змін умов зйомки.

Перелік використаних джерел.

1. Biometric facial recognition – Enhancing user verification and authentication. [Електронний ресурс].- Режим доступу: https://www.fraud.com/post/biometric-facial-recognition.

2. Hrytsyk V. V. Modeling and synthethis of complex symmetrical images / V. V. Hrytsyk, K. M. Berezska, O. M. Berezsky // International journal of pattern recognition and artificial intelligence. – 2004. – V. 18, № 2. – P. 175–195.

3. Messer K. et al. Performance characterization of face recognition algorithms and their sensitivity to severe illumination changes – ICB. – 2006

4. EL Fadel N. Facial Recognition Algorithms: A Systematic Literature Review. Journal of Imaging. 2025; 11(2):58. https://doi.org/10.3390/jimaging11020058

УДК 004.4

Шендрик Є.В., Головачова О.В.

Національний університет «Одеська політехніка»

ДОСЛІДЖЕННЯ ДИНАМИЧНИХ ЯВИЩ ПРИ ВИМІРЮВАННІ МАСИ РУХОМИХ ОБ'ЄКТІВ

Вступ: У роботі досліджуються проблеми виявлення інформативних параметрів сигналів, що отримуються при отриманні сигналів при зважуванні вантажу при русі об'єкту, що зважується. Основна проблематика в тому, що при русі об'єкта, що зважується виникає дуже багато перешкод, які впливають на результати: швидкість, якість покриття до зважувального комплексу, завади при передачі сигналу. Тому відокремлення інформаційної складової від завад є актуальна проблема виявлення корисної інформації із вхідного сигналу у будь який сфері діяльності: у комерційний сфері застосування, або при розробці нових винаходів у подальшому розвитку виявлення інформації при використанні інших засобів вимірювання

Мета: Виявлення корисної інформації на фоні завад отриманих з пристроїв прийому інформації при замалій кількості отриманих даних.

Основна частина

У якості виявлення методу виявлення інформативних параметрів досліджуваних даних використовується платформа для виявлення маси вантажу у русі. Особливістю досліджень є підвищена швидкість зважування, що не дозволяє виявити інформативні параметри (масу вантажу) при швидкості більше 5 км/год.

Дослідження подібних процесів показує, що модель сигналу повинна відповідати (1). Приведена модель сигналу адекватна узагальненій моделі процесу зважування та представлена сукупністю трьох складових

$$f(t)=D+A\sin(\Omega t+\psi)+\xi\rightarrow(t), \qquad (1)$$

де $f(t)$ - досліджуваний тензометричний сигнал; D - постійна складова сигналу (інформативний параметр, відповідний масі об'єкта, що зважується); $A\sin(\Omega t+\psi)$ - низькочастотна періодична складова сигналу; $\vec{\xi}(t)$ - випадкова величина, що виникає під час зважування.

Періодична завада представлена амплітудою - A, частотою - Ω і початковою фазою - ψ. А сигнал представлений сукупністю рівномірно розподілених у часі відліків $t_{i+i}-t_i=t_i-t_{i-1}=\Delta t$, де $i=\overline{0,n}$, $n+1=N$ - кількість значень сигналу.

У якості методів підвищення завадостійкості методу заданого діапазону частот використовуються: метод подвійного інтегрування, метод вагової функції, метод воріт. Метод подвійного інтегрування заснований на дослідженні реальних сигналів і припускає аналіз випадкового шуму $\vec{\xi}(t)$ у виді тригонометричного ряду (2)

$$\xi(t)=A_1\sin(\Omega_1 t+\Psi_1)+A_2\sin(\Omega_2 t+\Psi_2)+...A_m\sin(\Omega_m t+\Psi_m) \qquad (2)$$

у якому величини амплітуд A_1, A_2, ..., A_m як мінімум на порядок менше, а величини частот Ω_1, Ω_2, ... ,Ω_m як мінімум, на порядок більше відповідних величин сигналу (1). Двічі проінтегрувавши сигнал (1), підставивши замість $\vec{\xi}$

вираз (2). Інтегрування проводиться двічі по двох причинах. Однократне інтегрування недостатньо забезпечує зниження випадкового шуму, і, як наслідок, не дає бажаного збільшення точності вимірів. При багаторазовому інтегруванні виникає похибка алгоритму чисельного інтегрування, що приводить до зниження точності одержуваного результату. Таким чином, двічі проінтегрований сигнал має вигляд $f(t) = dtdt = \frac{Dt^2}{2} - \frac{A}{\Omega^2}\sin(\Omega t + \psi) - \frac{A_1}{\Omega_1^2}\sin(\Omega_1 t + \psi_1) - ... \frac{A_m}{\Omega_m^2}\sin(\Omega_m t + \psi_m)$

Після дворазового інтегрування величина амплітуди кожної із складових сигналу ділиться на квадрат її частоти. З урахуванням того, що величини амплітуд шуму на порядок менше, а величини частот на порядок більше відповідних величин сигналу (1), отримується істотне зниження високочастотних складових сигналу. Подальший хід обчислень зводиться до використання методу заданого діапазону частот. Запропонована нова система базисних функцій (3)

$$\begin{cases} \phi_0(t) = t^2, \\ \phi_1(t) = \cos(\Omega t), \\ \phi_2(t) = \sin(\Omega t), \\ \phi_3(t) = t, \\ \phi_1(t) = 1. \end{cases} \quad (3)$$

Пропонується змінити спосіб обчислення оцінок шуканих параметрів сигналу:

$$D = 2a_{0j}, \Omega = \Omega_{\min}, A = \Omega^2\sqrt{a_{1j}^2 + a_{2j}^2}, \Psi = arct\left(\frac{a_{1j}}{a_{2j}}\right).$$

Запропонований підхід дозволяє досягти десятикратного збільшення точності вимірів при малій тривалості досліджуваного сигналу. Вплив випадкового шуму особливо гостро виявляється на граничних ділянках досліджуваного сигналу. Таким чином, для підвищення точності оцінок сигналу вводиться система вагових коефіцієнтів, відповідно до якої граничним значенням задається найменша вага, що збільшується в міру наближення до центральних значень досліджуваного сигналу. Функція, що визначає значення кожного вагового коефіцієнта w_i, визначена як (4), де N - кількість значень сигналу.

$$W(t) = \sin\left(\frac{\pi}{N}t + \frac{\pi}{2N}\right), \quad (4)$$

Нова система базисних функцій (5) приймає вигляд

$$\begin{cases} \phi_0(t) = W(t), \\ \phi_1(t) = \cos(\Omega t)W(t), \\ \phi_2(t) = \sin(\Omega t)W(t), \end{cases} \quad (5)$$

На величину (4) також збільшуються значення досліджуваного сигналу (1) - $.f(t)W(t)$. При застосуванні методу вагової функції обчислення середньоквадратичного відхилення в методі заданого діапазону частот повинне визначатися відповідно до формули

$$S = \sum_{i=0}^{N-1}(P_m(t_i) - f(t_i))^2 W(t_i). \quad (6)$$

Використання (6) дозволяє отримати мінімум середньоквадратичного відхилення відповідний найбільш точному значенню частоти із заданого

діапазону $[\Omega_{min}\ldots\Omega_{max}]$, і, як наслідок, найбільш точне значення постійної складової сигналу *D*. Запропонований підхід дозволяє досягти десятикратного збільшення точності вимірів при малій тривалості досліджуваного сигналу, і його використання дозволяє одержати більшу точність оцінок параметрів сигналу (1) у порівнянні з методом подвійного інтегрування. В реальних умовах похибка вимірів при використанні кожного з розглянутих методів може перевищити похибку, отриману шляхом простого усереднення сигналу, що є неприпустимим і може розцінюватися як збійна ситуація. У зв'язку з цим пропонується доповнити метод вагової функції, як кращий із представлених, таким чином, щоб гарантувати величину похибки не перевищуючу похибку усереднення значень сигналу при будь-яких умовах виміру.

Запропоновано використовувати обмеження на вибір оцінок параметрів моделі (1) - ворота, які визначають припустимий діапазон отриманого значення постійної складової сигналу *D*. З проведених досліджень виявлено, що величина амплітуди низькочастотної складової сигналу (1) складає не більш 10...15 % величини постійної складової сигналу *D*. Таким чином, у результаті застосування методу заданого діапазону частот, обчислене значення величини $D = a_0$ не повинне перевищити середньоарифметичне значення досліджуваного сигналу $\overline{F}$, на величину (7), де *G* - величина воріт.

$$G = \pm\overline{F}/10, \tag{7}$$

Таким чином, з урахуванням уведених доповнень найбільш точним вважається той результат, що знаходиться в інтервалі

$$a_0 \in [\overline{F} \pm G], \tag{8}$$

де a_0 - поточний результат апроксимації, що відповідає *D*;

та має найменшу середньоквадратичну похибку, розраховану по формулі (6). Якщо жодне з отриманих значень не задовольняє (8), то як шуканий результат виступає середньоарифметичне значення досліджуваного сигналу $\overline{F}$.

Програмна реалізація приведеного алгоритму в складі автоматизованого ваговимірювального комплексу при обмеженому часі зважування дозволяє підвищити точність вимірів у 2...7 разів стосовно усереднення значень сигналу.

Висновки. Проаналізовано методи та засоби підвищення точності та швидкодії автоматизованих ваговимірювальних систем. Визначено, що класичними методами та засобами, заснованими на усередненні вибірки сигналу не можливо забезпечити необхідну точність вимірювань, тому що їхнє використання не враховує особливості вхідного сигналу при автоматизованому зважуванні об'єктів на швидкості руху.

Перелік використаних джерел.

1. Демидович Б.П., Марон И.А., Шувалова Э.З. Численные методы анализа. Приближение функций, дифференциальные и интегральные уравнения. - М.: Наука, 1967. - 368 с.

2. Фиакко А., Мак-Кормик Г. Нелинейное программирование. Методы последовательной безусловной минимизации: Пер с англ. - М.: Мир, 1972. - 240 с.

3. Демидович Б.П., Марон И.А. Основы вычислительной математики. - М.: Наука, 1970. - 664 с.

УДК 004.056

Іван ТИХОНОВ, Олександр СИРОПЯТОВ

[1]Національний університет «Одеська політехніка»

МАШИННЕ НАВЧАННЯ ДЛЯ ПРОТИДІЇ ФІШИНГОВИМ АТАКАМ

Вступ. Фішингові атаки залишаються однією з найпоширеніших та найнебезпечніших загроз у кіберпросторі. Їхня мета – викрадення конфіденційних даних користувачів, таких як логіни, паролі та фінансова інформація, шляхом маскування під легітимні веб-ресурси.

Традиційні методи захисту, що базуються на чорних списках, часто виявляються неефективними проти нових атак через високу швидкість створення та реєстрації фішингових доменів. У зв'язку з цим, актуальною задачею є розробка інтелектуальних систем, здатних проактивно ідентифікувати загрози.

Мета: Розробка та дослідження моделі машинного навчання (ML) для автоматичної класифікації веб-ресурсів на «легітимні» та «фішингові» на основі аналізу їхніх ключових характеристик для підвищення ефективності систем протидії кіберзагрозам.

1. Класифікація ознак фішингу

В якості основи для моделі ML було виділено декілька груп ознак (features), що характеризують веб-сторінку. На відміну від класичних підходів, запропонована модель аналізує три ключові вектори даних:

– ознаки на основі URL: Довжина URL, кількість піддоменів, наявність IP, використання не-стандартних портів, наявність символів «@», «-»;

– ознаки на основі HTML: Кількість тегів <form>, наявність <iframe>, перенаправлення, кількість посилань на зовнішні домени;

– ознаки на основі домену: Вік домену (Domain Age), наявність SSL-сертифікату, термін дії сертифікату, рейтинг у пошукових системах [1].

Загальний вектор ознак X для кожного сайту можна представити як:

$$X = (x_1, x_2, \ldots, x_n), \quad (1)$$

де n – загальна кількість виділених ознак.

2. Навчання моделі

Задача зводиться до бінарної класифікації, де кожному вектору X необхідно присвоїти мітку $y \in \{0, 1\}$ – фішинг, а 0 – легітимний сайт. Для навчання моделі було використано алгоритми Support Vector Machine (SVM) та Random Forest (RF) [2].

Результати та аналіз. Для того, щоб зрозуміти, наскільки добре наша модель справляється із завданням, ми використали стандартний підхід для оцінки класифікаторів. Основою для цього є матриця плутанини (або Confusion Matrix). Вона наочно показує, де саме модель помиляється, а де приймає правильні рішення.

Матриця для нашої задачі, бінарної класифікації «Фішинг» / «Легітимний» представлена в Таблиці 1.

Таблиця 1 – Отримані експериментальні дані

Факт (Рядки) / Прогноз (Стовпці)	Прогноз: Фішинг (Позитивний)	Прогноз: Легітимний (Негативний)
Факт: Фішинг (Позитивний)	True Positive (TP) Вірно визначена загроза	False Negative (FN) Пропуск загрози
Факт: Легітимний (Негативний)	False Positive (FP) Хибна тривога	True Negative (TN) Вірно визначений легітимний сайт

де:

– TP (True Positive) – кількість фішингових сайтів, коректно визначених як фішинг;

– TN (True Negative) – кількість легітимних сайтів, коректно визначених як легітимні;

– FP (False Positive) – «хибна тривога», кількість легітимних сайтів; помилково визначених як фішинг;

– FN (False Negative) – «пропуск загрози», кількість фішингових сайтів, помилково визначених як легітимні [3].

На основі цих показників розраховуються ключові метрики.

– Точність (Accuracy) – загальна частка правильних прогнозів;

– Влучність (Precision) – частка об'єктів, вірно названих «фішингом»;

– Повнота (Recall) – частка фішингових сайтів, які модель зуміла знайти.

– F1-Score - гармонійне середнє між влучністю та повнотою, що дає збалансовану оцінку.

Для оцінки якості моделі використовувалися метрики Accuracy, Precision, Recall та F1-Score.

Попередні результати (або вставте сюди ваші результати, таблицю чи посилання на рис.) демонструють, що модель Random Forest показує найвищу точність ідентифікації (напр., >95%) та здатна коректно класифікувати значну частину раніше невідомих загроз.

Висновок. Розроблено та протестовано модель машинного навчання для детектування фішингових веб-ресурсів. Експериментальні дослідження підтвердили високу ефективність обраного підходу, зокрема використання ансамблевих методів, для виявлення загроз на основі комплексного аналізу характеристик сайту. Результати можуть бути використані при проєктуванні інтелектуальних модулів для веб-браузерів, поштових клієнтів та корпоративних систем безпеки з метою проактивного захисту користувачів від фішингових атак.

Перелік використаних джерел.

1. Коваленко А.В., Петренко С.М. Використання алгоритмів машинного навчання для детектування фішингових сайтів. *Кібербезпека: освіта, наука, техніка*. Львів, 2023. Т. 4, №1. С. 58-67.

2. Сидоренко В.В., Іванов Д.Ю. Порівняльний аналіз моделей Random Forest та SVM у задачах виявлення фішингу. *Матеріали X Міжнародної науково-практичної конференції «Інформаційна безпека та комп'ютерні технології»*. Київ. 2024. С. 112-114.

3. PhishTank. Open Database of Phishing Sites. URL: https://phishtank.org/

УДК 004.056.5:004.8

Ігор РУДЬКО, Юрій ДОРОФЕЄВ

Національний університет «Одеська політехніка»

РОЗРОБЛЕННЯ ЗАСТОСУНКУ ДЛЯ НАВЧАННЯ РОЗПІЗНАВАННЮ ВІРУСНИХ ЕЛЕКТРОННИХ ЛИСТІВ

Вступ. В умовах інтенсивного розвитку цифрових технологій електронна пошта є одним із ключових засобів професійної та особистої комунікації. Однак саме через електронну пошту щорічно здійснюється понад половина усіх кібератак, спрямованих на викрадення даних, встановлення шкідливого програмного забезпечення або компрометацію корпоративних мереж[1].

Особливої актуальності набуває питання навчання користувачів методам виявлення підозрілих повідомлень та формування стійких навичок кібергігієни.

Мета. Метою роботи є розроблення інтерактивного навчального застосунку, який моделює процеси отримання, аналізу та оцінювання електронних листів із можливими ознаками фішингу або вірусної активності. Основна ідея полягає у поєднанні генеративних мовних моделей і методів машинного навчання для створення навчального середовища, що імітує реальні сценарії атак.

1. Архітектура та функціональні модулі системи

Запропонована система складається з трьох основних модулів: генерації контенту, аналізу та оцінювання безпеки. Така модульна структура забезпечує гнучкість і можливість масштабування розробки, що дозволяє адаптувати застосунок для різних категорій користувачів – від студентів до працівників IT-компаній.

Модуль генерації відповідає за створення фішингових і вірусних листів, максимально наближених до реальних. Для цього використовується локальна велика мовна модель Ollama, здатна формувати тексти різного стилю - від офіційних бізнес-листів до коротких особистих повідомлень. Завдяки цьому користувач стикається з правдоподібними прикладами атак, що підвищує ефективність тренування.

Особливу увагу приділено реалістичності сценаріїв, приклади яких наведено у статті[2]. Модель імітує не лише структуру повідомлень, але й характерні помилки зловмисників: підроблені адреси відправників, незначні стилістичні відхилення, емоційно маніпулятивні фрази. Таким чином, користувач вчиться помічати деталі, які часто ігноруються у реальному житті.

Модуль аналізу реалізований за допомогою комбінації класичних і сучасних алгоритмів обробки тексту: Word2Vec, Doc2Vec, GloVe, а також байєсівського класифікатора SpamBayes. Вони дозволяють автоматично визначати потенційно небезпечні елементи, такі як аномалії лексики, підозрілі посилання чи приховані елементи HTML, що можуть бути використані для експлуатації вразливостей.

Аналіз здійснюється локально, без підключення до зовнішніх серверів. Це забезпечує конфіденційність персональних даних і виключає можливість витоку інформації, що робить систему придатною для використання в освітніх та корпоративних середовищах.

На рисунку 1 подано загальну схему роботи системи, що демонструє послідовність основних етапів – від створення листа до аналізу та оцінки безпеки.

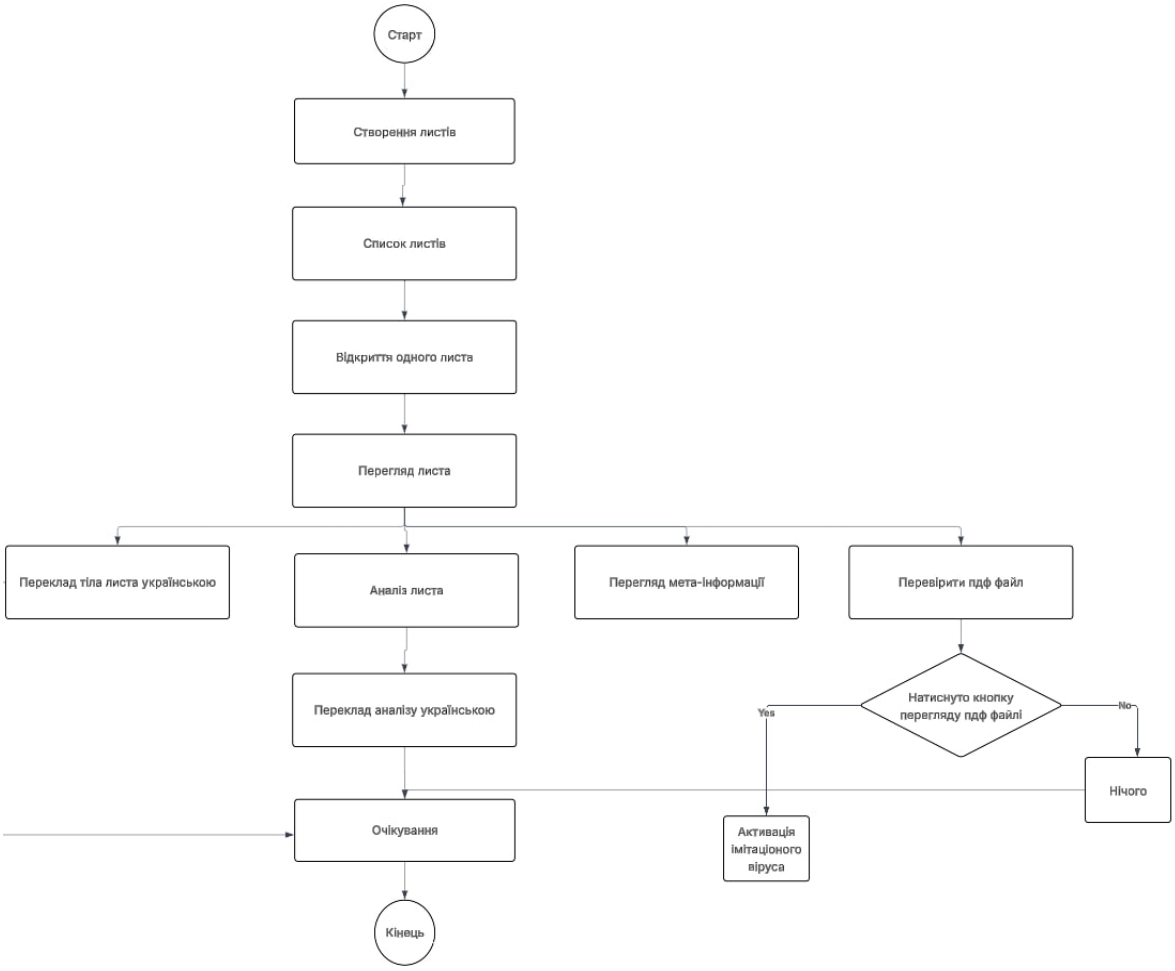


Рисунок 1 – Блок-схема інтерфейсу демонстрації:

Блок-схема інтерфейсу демонстрації: послідовність створення листів, перегляду списку та одного листа, дій користувача (переклад, аналіз, перегляд метаданих, перевірка PDF і активація імітаційного вірусу)

Модуль оцінювання безпеки відображає результати аналізу та надає зворотний зв'язок користувачу. Після кожного тренування програма формує статистику: кількість правильно ідентифікованих загроз, помилки та рекомендації щодо покращення уваги до деталей. Для реалізації інтерфейсу використано Python і бібліотеку Tkinter, що забезпечує кросплатформну сумісність і простоту використання. Інтерфейс інтуїтивний, з поділом на вкладки: «Пошта», «Аналіз», «Результати». Кожен лист супроводжується кнопкою для позначення як безпечного або небезпечного, після чого система миттєво відображає оцінку рішення з коротким поясненням.

2. Тестування та результати

Для оцінювання ефективності застосунку проведено експеримент серед студентів спеціальності «Кібербезпека» Національного університету «Одеська політехніка». Учасники проходили серію тренувальних сеансів, під час яких отримували згенеровані приклади листів із різним рівнем складності. Уже після трьох сесій середній рівень точності розпізнавання небезпечних повідомлень підвищився на 63 %, що підтвердило навчальну ефективність розробленої системи.

Отримані результати засвідчили, що користувачі почали приділяти більшу увагу дрібним деталям у тексті листів, таким як доменна адреса, стиль написання, логотипи або формулювання термінових закликів. Це вказує на підвищення рівня обізнаності й критичного мислення при взаємодії з електронними повідомленнями, що є ключовою метою навчання.

Крім того, учасники відзначили зручність інтерфейсу та високу реалістичність змодельованих сценаріїв, що сприяло зануренню в процес і підвищенню мотивації до навчання.

Подальший розвиток системи передбачає розширення функціональності додаванням мультимовної підтримки, інтеграцією з корпоративними поштовими клієнтами та впровадження адаптивного рівня складності завдань залежно від результатів користувача. Також планується створення адміністративної панелі для викладачів або керівників, які зможуть відстежувати прогрес групи в режимі реального часу.

Таким чином, розроблений застосунок може бути ефективним інструментом не лише для освітніх закладів, а й для корпоративних тренінгів із підвищення кіберграмотності працівників

Висновок. Розроблений застосунок є ефективним засобом інтерактивного навчання безпечній роботі з електронною поштою. Поєднання методів NLP, машинного навчання та генеративних мовних моделей дозволяє створити сучасну систему, що моделює реальні сценарії атак і формує стійкі навички протидії фішинговим і вірусним загрозам.

Перелік використаних джерел.

1. Newman M. E. J., Forrest S., Balthrop J. Email Networks and the Spread of Computer Viruses. Physical Review E. 2002. Vol. 66, No 3. P. 035101. DOI: 10.1103/PhysRevE.66.035101.

2. Singh P., Maravi Y. P. S., Sharma S. Phishing Websites Detection Through Supervised Learning Networks. International Conference on Computing and Communications Technologies (ICCCT). 2015. P. 61–65. DOI: 10.1109/ICCCT2.2015.7292720.

УДК 004.056.5

Горбатій Б.М.,Садченко А.В.

Національний університет «Одеська політехніка»

ІМПЛЕМЕНТАЦІЯ АДАПТИВНОГО АЛГОРИТМУ ЗАХИСТУ АКУСТИЧНОГО КАНАЛУ ВИТОКУ ІНФОРМАЦІЇ

Вступ. У сучасних інформаційно-комунікаційних системах дедалі більшої актуальності набуває проблема захисту інформації від несанкціонованого доступу через технічні канали витоку. Одним із найбільш небезпечних та складних для виявлення є акустичний канал, який виникає внаслідок перетворення вібраційних або мовних коливань у відновлювану інформацію. Використання високочутливих мікрофонів, лазерних віброметрів або побічних акустичних випромінювань дозволяє зловмисникам здійснювати перехоплення конфіденційних даних навіть без прямого доступу до об'єкта.

Традиційні методи акустичного захисту, такі як пасивне екранування, акустичне маскування або генерація шумових сигналів мають обмежену ефективність, оскільки не враховують динамічні зміни умов середовища, частотний спектр мовних сигналів та рівень завад. У зв'язку з цим виникає потреба у створенні адаптивних алгоритмів, здатних в реальному часі змінювати свої параметри відповідно до характеристик акустичного простору та типу виявленої інформаційної загрози.

Запропонований у даній роботі підхід ґрунтується на адаптивній обробці акустичних сигналів з використанням фільтрації та генерації компенсаційних шумів, спрямованих на мінімізацію можливості реконструкції мовної інформації. Розроблений алгоритм дозволяє автоматично підлаштовувати рівень і спектральну структуру шуму під поточні умови, забезпечуючи баланс між ефективністю захисту та акустичним комфортом у приміщенні.

Мета: імплементація та дослідження адаптивного алгоритму захисту акустичного каналу витоку інформації, який забезпечує підвищення стійкості системи до відновлення мовного сигналу сторонніми засобами. Для досягнення цієї мети проаналізовано існуючі методи акустичного захисту, розроблено модель адаптивної фільтрації, а також проведено експериментальну перевірку ефективності запропонованого рішення.

Основна частина

Розглянемо основні риси мовного сигналу, що відрізняє його від від випадкових шумів [1]. Мовний сигнал має низку характерних ознак, які дозволяють відрізнити його від випадкових шумів як у часовій, так і у частотній областях. На відміну від шуму, що зазвичай має стохастичний або рівномірно розподілений спектральний характер, мовлення є складним, але структурованим процесом, зумовленим фізіологічними особливостями артикуляційного апарату людини.

По перше мовний сигнал не є повністю випадковим - він формується послідовністю фонем, які мають визначену тривалість і акустичні параметри. На коротких інтервалах (10–30 мс) мовлення можна вважати квазістаціонарним, що

дає можливість застосовувати спектральні методи аналізу.

Мовний сигнал має властивості періодичності та гармонічності. Так у голосних звуках спостерігається виражена періодичність, пов'язана з коливанням голосових зв'язок. Це проявляється у вигляді чіткої гармонічної структури спектра, де присутня основна частота та її кратні гармоніки. Шумові компоненти, навпаки, мають неперіодичний і нерегулярний характер [1].

Мовний сигнал має формантну структуру, тобто для мовного сигналу характерна наявність спектральних максимумів, які визначають артикуляційні особливості звуків. Розташування формант (зазвичай 3–4 основні) залишається стабільним у межах певної фонеми, що дає змогу ефективно відокремлювати мовлення від шуму у частотній області.

Інтенсивність, частота та енергетичний розподіл мовного сигналу змінюються у часі відповідно до мовленнєвого потоку. У той час як шум, особливо білий або техногенний, часто має постійний або повільно змінний рівень енергії.

Мовлення містить паузи між словами або фразами, що чергуються з активними ділянками сигналу. Ця ритмічна структура суттєво відрізняє мовлення від безперервного шумового процесу.

Для мовного сигналу характерний нерівномірний спектральний розподіл енергії при якому більшість енергії мовного сигналу зосереджена у діапазоні 300–3400 Гц, тоді як шум може мати ширший або рівномірний спектр. Ця особливість дозволяє застосовувати фільтрацію та спектральне маскування для виділення мовлення.

Мовний сигнал характеризується високим рівнем автокореляції на коротких інтервалах часу, що відображає його внутрішню упорядкованість. Для шуму автокореляційна функція близька до дельта-функції, тобто спостерігається відсутність зв'язку між сусідніми відліками.

В запропонованому алгоритмі зашумлення мовного сигналу шум генерується як випадковий сигнал, який підпорядковується нормальному закону розподілу. Для кожної вибірки , шум обчислюється наступним чином:

$$n(t) = A \cdot N(0,1), \quad (1)$$

де А-амлітуда шумового впливу, що дорівнює максимальної амплітуді формант мовного сигналу,

$N(0,1)$ −білий Гаусовський шум із нульовим математичним очікуванням;

Шум додається до завантаженого аудіо сигналу. Перед додаванням, амплітуда шуму масштабується, щоб відповідати заданому співвідношенню сигнал/шум (SNR) з урахуванням максимальної амплітуди формант мовного сигналу.

Для коректної роботи алгоритму, спочатку, обчислюється потужність сигналу та потужність шуму:

$$P_{\text{сигналу}} = \frac{1}{N}\sum_{t=1}^{N} x^2(t) \quad (2)$$

де $x(t)$ – вихідний аудіосигнал;

$n(t)$ – шум, N – кількість відліків;

Далі, обчислюється необхідна потужність шуму на основі заданого SNR за наступними формулами:

$$SNR_{dB} = 10lg\left(\frac{P_{\text{сигналу}}}{P_{\text{н.ш}}}\right), \quad (3)$$

$$P_{\text{н.ш}} = \frac{P_{\text{сигналу}}}{10^{\frac{SNR_{dB}}{10}}}, \quad (4)$$

де $P_{\text{сигналу}}$ – потужність сигналу,

$P_{\text{н.ш}}$ – потужність необхідного шуму,

SNR_{dB} – відношення сигнал/шум, що представлене в децибелах.

Для попередньої фільтрації аудіофайлу використовується фільтр низької частоти із максимально пласкою характеристикою, що апроксимується поліномом Баттерворта [3].

Для розгляду роботи програми було використано аудіофайл у форматі wav довжиною 73 секунди. Ця довжина аудіофайлу не є стандартизованою і може обиратись будь-якою. Але, для коректної роботи програми, довжина аудіофайлу повинна бути достатньою, щоб здійснити всі необхідні перетворення файлу [2].

Накладання шуму відбувається за рахунок варіювання показника SNR_{dB}. Для дослідів, було обрано динамічний діапазон корисного сигналу від -40 до 40 dB. Результати оцінки якості звуку в залежності від показника SNR (таблиця 1):

Таблиця 1 – Оцінка якості звуку в залежності від показника SNR

SNR(dB)	Рівень шуму (dB)	Розбірливість мови (%)	Суб'єктивне сприйняття мови
-40 – -30	>50	<60	Аудіофайл повністю перекривається шумом, не можна ідентифікувати оригінальний звук
-30 – -20	25-50	60-70	Оригінальний звук майже повністю спотворений шумом
-20 – -10	10-25	70-80	Рівень шуму дозволяє ідентифікувати оригінальний звук, але з досить невеликим значенням точності
-10 – 0	5-10	80-90	Рівень шуму знижений, оригінальний звук ідентифікується більш чітко
0-10	1-5	90-100	Рівень шуму залишається помітним, але недостатнім для перешкоджання ідентифікації оригінального звуку

Динамічний діапазон показує, яким чином зростає якість звуку при збільшенні SNR. Для демонстрації роботи програми варто навести кілька прикладів при різних значеннях параметрів. Розглянемо результат обробки аудіофайлу при наступних параметрах: SNR=0 дБ, нижня частота 195 Гц, верхня частота 8192 Гц, Амплітуда 0,5 В. Результати моделювання зображені на рисунку 1. Після оброблення, вихідний сигнал втрачає вигляд чітко сегментованої структури і не може бути розпізнаний за допомогою алгоритмів формантного аналізу мови.

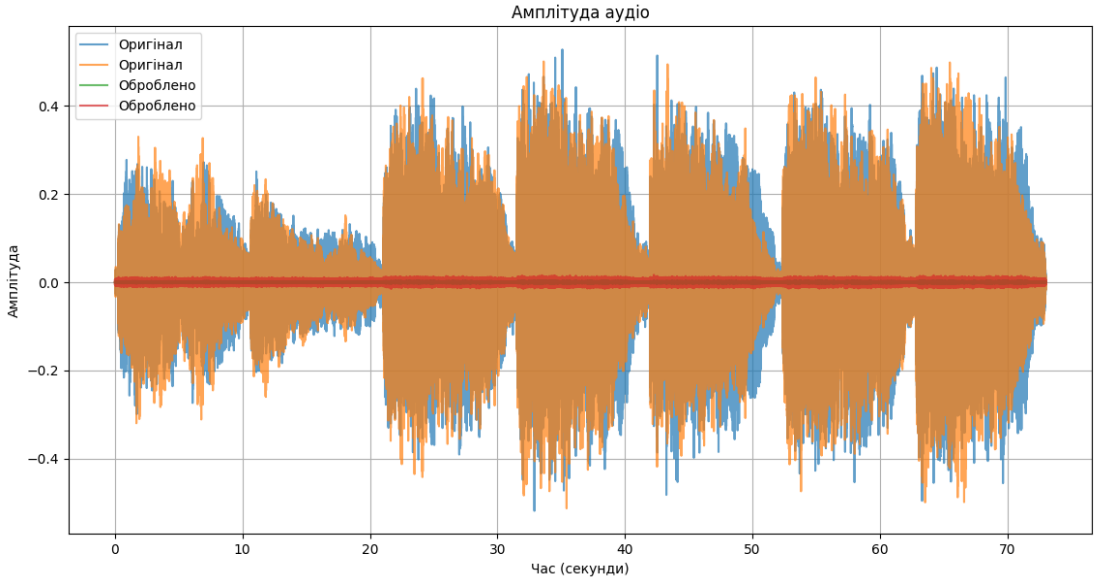


Рисунок 1 - Результати моделювання

Висновок. У роботі розглянуто способи захисту акустичного каналу витоку інформації шляхом генерації адаптивного шуму. Проаналізовано етапи обробки сигналу - параметризацію, сегментацію, фільтрацію та маскування шумом.

Реалізований підхід базується на використанні математичних моделей що дозволяють розраховувати потужність сигналу, шуму та формувати ефективний рівень маскування. Застосування фільтра Баттерворта забезпечує очищення сигналу від небажаних складових.

Розроблений алгоритм дає змогу підвищити рівень захисту інформації, зменшуючи ризик її перехоплення через акустичний канал. Поставлена мета роботи досягнута.

Перелік використаних джерел.

1. Методи та засоби технічного захисту інформації. [Електронний ресурс] : навч. посіб. для здобувачів ступеня бакалавра за освітньою програмою «Системи технічного захисту інформації» спеціальності 125 «Кібербезпека» / КПІ ім. Ігоря Сікорського; уклад.: В. М. Луценко, Д. О. Прогонов., 2021 – URL: https://ela.kpi.ua/handle/123456789/42397

2. Lukmanova O., Horev A.A., Vorobeyko E., Volkova E.A. Research of the analog and digital noise generators characteristics for protection device // Proc. of the IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering, (EIConRus), 2020. P. 2093–2096. https://doi.org/10.1109/EIConRus49466.2020.9039193

3. Полотай О.В., Мороз Ю.О., Великий В.П. Методи технічного захисту інформації у сфері інформаційної безпеки. Інформаційна безпека інформаційні технології: Збірник тез доповідей IV Всеукраїнської науково-практичної конференції молодих учених, студентів і курсантів. Львів, 2020. С. 40-41.

УДК 004.738.5

Космачевський М.В., Садченко А.В.

Національний університет «Одеська політехніка»

РОЗРОБКА ЗАХИЩЕНОЇ ВЕБ-ПЛАТФОРМИ ДЛЯ ЗАМОВЛЕННЯ ТА ПРОДАЖУ АВТОМОБІЛІВ

Вступ.Сучасний ринок автомобілів переживає активну цифрову трансформацію. За останні роки попит на онлайн-платформи для купівлі та замовлення транспортних засобів зріс у декілька разів. Традиційні платформи, де будь-який користувач може публікувати оголошення, часто стикаються з проблемами недостовірної інформації, шахрайських схем та низької якості сервісу.

Брокерська модель продажу автомобілів набуває популярності, оскільки професійний брокер виступає гарантом якості послуг, перевіряє автомобілі, забезпечує юридичну чистоту угод та надає комплексний супровід клієнтам. Відсутність спеціалізованих захищених платформ для брокерської діяльності створює потребу у розробці рішення, яке поєднує безпеку, зручність та ефективність.

Мета: Розробити захищену веб-платформу для замовлення та продажу автомобілів з обмеженим доступом до публікації оголошень, яка забезпечить високий рівень безпеки даних та зручність використання для всіх категорій користувачів.

Основна частина

Платформа реалізує трирівневу модель доступу на основі ролей (RBAC -- Role-Based Access Control) [1]:

- адміністратор,
- брокер
- клієнт.

Адміністратор має повний доступ до управління користувачами, створення облікових записів брокерів, налаштування параметрів системи та перегляду аналітики.

Брокер може створювати, редагувати та видаляти власні оголошення, завантажувати фотографії та документи автомобілів, управляти статусами оголошень, обробляти замовлення від клієнтів та комунікувати з ними через вбудований чат.

Клієнт має можливість реєстрації, перегляду каталогу автомобілів, використання фільтрів та пошуку, створення замовлень та відстеження їх статусу [3].

Проведено порівняльний аналіз існуючих платформ (AUTO.RIA, OLX, AutoScout24, Carvana), який виявив ключові недоліки: низький рівень достовірності інформації (40-70%), проблеми безпеки даних, відсутність кваліфікованої підтримки та неефективну систему пошуку[5].

Закрита модель публікації через верифікованих брокерів забезпечує достовірність даних понад 95%, що на 25-55% вище порівняно з відкритими

платформами. Обмеження права публікації виключно професійними брокерами гарантує контроль якості інформації, захист від шахрайства, професійний супровід кожної угоди та стандартизацію процесів.

Технічна реалізація безпеки включає багатофакторну автентифікацію (MFA), шифрування даних за стандартами SSL/TLS та AES-256, хешування паролів алгоритмом bcrypt, систему аудиту дій, захист від SQL-ін'єкцій та XSS-атак, обмеження швидкості запитів (Rate Limiting) для захисту від DDoS-атак [2].

Політика паролів передбачає мінімальну довжину 12 символів з використанням різних типів символів. Платформа забезпечує GDPR-сумісність для захисту персональних даних користувачів.

Функціональні можливості платформи включають інтелектуальну систему фільтрації з багатокритеріальним пошуком, покращену візуалізацію з великими якісними зображеннями та відео-презентаціями, поетапне оформлення замовлення з вибором способу комунікації та бажаних опцій, автоматичні email-сповіщення та відстеження статусу замовлення на всіх етапах обробки.

Загальний вигляд платформи зображений на рисунку 1.

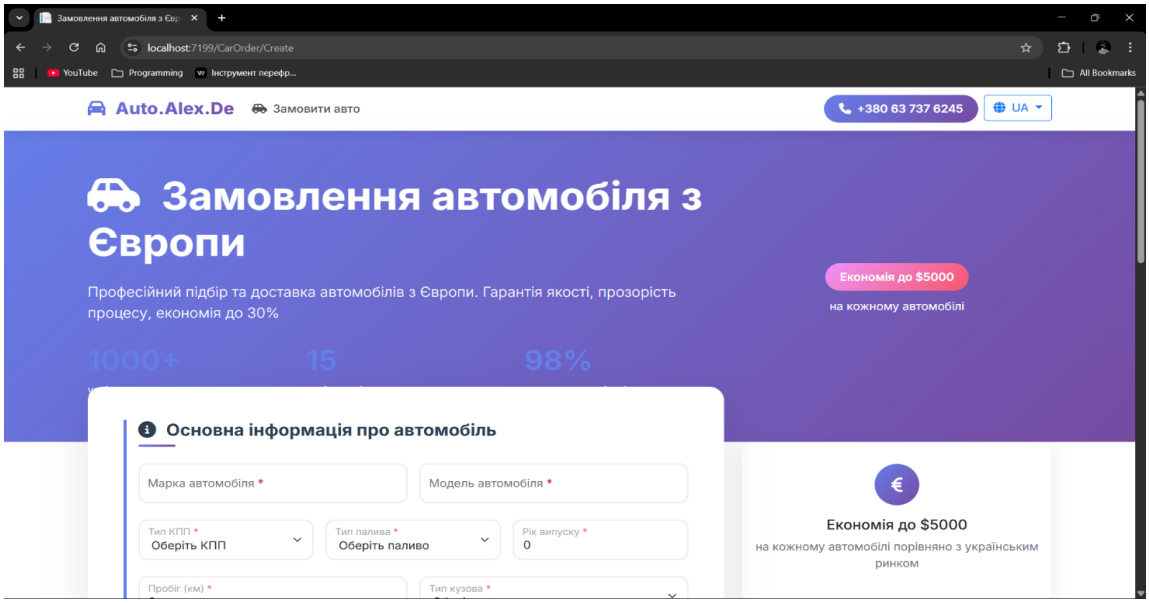


Рисунок 1 - . Загальний вигляд платформи

Адаптивний дизайн забезпечує комфортне користування на будь-яких пристроях. Оптимізація продуктивності досягається через ледаче завантаження зображень, кешування статичного контенту та мінімізацію CSS/JavaScript файлів [2].

Порівняльний аналіз показує конкурентні переваги розроблюваної платформи (таблиця 1) [5]:

– спеціалізацію на брокерській моделі,

– максимальну довіру клієнтів через гарантію перевірки кожного автомобіля,

– ефективність для брокерів завдяки зручним інструментам управління,

– швидкість угод (скорочення часу у 2-3 рази)

– прозорість процесів.

Таблиця 1 – Порівняльний аналіз ключових показників платформ

Критерій	AUTO.RIA	OLX	AutoScout24	Розроблювана платформа
Модель доступу	Відкрита	Відкрита	Напівзакрита	Закрита (брокери)
Достовірність даних	60-70%	40-50%	80-85%	95%+
Безпека угод	Низька	Дуже низька	Середня	Висока
Професійний супровід	Відсутній	Відсутній	Опціонально	Обов'язковий
Час обробки замовлення	Не застосовне	Не застосовне	24-48 год	4-12 год
Захист даних	Базовий	Мінімальний	Стандартний	Розширений (MFA, шифрування)

Висновок. Розроблена концепція захищеної веб-платформи представляє інноваційний підхід до організації брокерського бізнесу в автомобільній сфері. Закрита модель публікації через верифікованих брокерів усуває до 90% проблем, пов'язаних з недостовірною інформацією та шахрайством. Впровадження багаторівневої системи захисту даних забезпечує конфіденційність персональної інформації користувачів.

Автоматизація бізнес-процесів підвищує ефективність роботи брокерів та скорочує час обробки запитів клієнтів. Інтуїтивний інтерфейс, розумні фільтри та прозорість процесів роблять платформу зручною для всіх категорій користувачів.

Перспективи розвитку включають впровадження штучного інтелекту для прогнозування попиту та персоналізації рекомендацій, інтеграцію з blockchain-технологіями для абсолютної прозорості історії автомобіля, розширення функціоналу до екосистеми автомобільних послуг (сервіс, страхування, фінансування).

Перелік використаних джерел.

1. Sommerville I. Software Engineering. 10th Edition. Pearson, 2015. 816 p.

2. OWASP Top Ten Web Application Security Risks [Електронний ресурс]. URL: https://owasp.org/www-project-top-ten/ (дата звернення: 25.10.2025).

3. Regulation (EU) 2016/679 of the European Parliament and of the Council on the protection of natural persons with regard to the processing of personal data (General Data Protection Regulation). Official Journal of the European Union, 2016.

4. Kleppmann M. Designing Data-Intensive Applications: The Big Ideas Behind Reliable, Scalable, and Maintainable Systems. O'Reilly Media, 2017. 616 p.

5. Статистика ринку автомобілів України 2023-2024 [Електронний ресурс] / Укравтопром. URL: https://ukrautoprom.com.ua (дата звернення: 20.10.2025).

УДК 004.93

Пасько В.В.

[1]Західноукраїнський національний університет

АДАПТИВНІ ТЕХНОЛОГІЇ ТРИВИМІРНОГО РЕНДЕРИНГУ В ДОДАТКАХ ВІРТУАЛЬНОЇ РЕАЛЬНОСТІ

Вступ. У сучасному світі технологій віртуальної реальності (VR) потреба у високоякісному та продуктивному відтворенні тривимірних моделей стає дедалі критичною. Для створення переконливих VR-сцен необхідно застосовувати алгоритми рендерингу, які забезпечують не лише реалістичну візуалізацію, але й достатню продуктивність, щоб уникнути затримок і зменшити втому користувача. Метою цієї роботи є дослідження існуючих підходів до рендерингу тривимірних моделей у VR-системах, обґрунтування вибору певних алгоритмів та подальший розвиток оптимізованих рішень.

Мета: провести аналіз сучасних технологій рендерингу у VR, виділити ключові алгоритми з огляду на якість і продуктивність, а також навести перспективи їх вдосконалення.

1. Аналіз сучасних адаптивних технологій рендерингу тривимірних моделей для систем віртуальної реальності

Процес рендерингу тривимірних моделей у VR вимагає високої обчислювальної потужності, оскільки користувач взаємодіє з повноцінним тривимірним середовищем у режимі реального часу. При цьому обробляються тисячі об'єктів, текстур, освітлювальних ефектів та динамічних тіней у кожному кадрі. Особливістю VR є необхідність рендерити зображення окремо для кожного ока (стереоскопічне бачення), що подвоює навантаження на графічну підсистему.

Одним із базових методів візуалізації залишається растеризація. Вона широко використовується у таких рушіях, як Unity та Unreal Engine, і дозволяє швидко трансформувати 3D-геометрію у 2D-проекцію. Основна перевага – висока швидкодія, однак метод менш придатний для фізично точного моделювання світла та відбиттів. З огляду на потребу у фотореалістичності, особливо в архітектурній візуалізації або VR-тренажерах, зростає роль трасування променів (ray tracing).

Трасування променів, реалізоване через RTX-технології NVIDIA [1], імітує фізику розповсюдження світла. В умовах VR його комбінують з растеризацією у гібридному підході, що дозволяє отримати складні оптичні ефекти при збереженні продуктивності [2].

Такий підхід забезпечує компроміс між якістю зображення та швидкодією. Архітектура RTX Hybrid Rendering підтримується сучасними VR-шоломами, зокрема HTC Vive Pro та Meta Quest 3 через Link.

Для досягнення стабільної частоти кадрів застосовується адаптивна деталізація сцени (Level of Detail, LOD). Принцип роботи LOD полягає у заміні складних 3D-моделей на спрощені варіанти залежно від відстані до глядача. Це дозволяє рендерити віддалені об'єкти з меншою кількістю полігонів, не втрачаючи візуальної якості. У VR, де користувач може швидко змінювати

напрямок огляду, ефективна LOD-система є обов'язковою умовою.

Крім того, VR-рушії реалізують алгоритми відсічення об'єктів (culling):

- Back-Face Culling пропускає полігони, що обернені від камери;
- Frustum Culling ігнорує об'єкти, що повністю за межами поля зору;
- Occlusion Culling не рендерить об'єкти, які повністю закриті іншими.

Ці технології дозволяють уникати обробки непотрібних елементів сцени, що суттєво знижує навантаження на GPU і покращує реактивність VR-середовища.

В останні роки особливої популярності набуває Foveated Rendering – інноваційна технологія, що адаптує якість рендерингу до положення погляду користувача. За допомогою трекінгу очей (eye-tracking) визначається фовеальна зона (зона максимальної гостроти зору), у якій виконується рендеринг з максимальною роздільною здатністю. У периферійних зонах застосовується понижена якість. Такий підхід знижує обчислювальні витрати до 40% без помітного погіршення якості зображення [3].

Ще одним ключовим елементом є шейдери – спеціальні програми, що керують обчисленням кольору пікселів, освітленням, ефектами матеріалів. Спрощення шейдерів, а також передрендеринг фонових ефектів, дозволяє скоротити затримки при рендерингу сцени. Наприклад, попередньо обчислене глобальне освітлення (baked GI) широко використовується в стаціонарних VR-сценах.

Таким чином, ефективний рендеринг у VR є результатом інтеграції численних методів і оптимізацій: від класичної растеризації до трасування променів, від LOD до foveated rendering, а також використання сучасних графічних API (DirectX 12, Vulkan, OpenXR).

Висновок. Проведений огляд показав, що для систем віртуальної реальності найбільш перспективними є технології, які адаптують рендеринг до особливостей сприйняття людини (наприклад, фовеальний рендеринг), а також технології, що динамічно оптимізують сцену (LOD, culling). Комбінування таких підходів з апаратними та алгоритмічними оптимізаціями створює можливість реалізації реалістичного рендерингу у режимі реального часу навіть на обмежених ресурсах. У подальшому доцільно розробляти власні впровадження цих алгоритмів, тестувати їх в контексті VR-систем, особливо з урахуванням адаптації до різних платформ (мобільних, стаціонарних). Вдосконалення цих напрямів відкриє шлях до більш глибокої інтеграції 3D-рендерингу у VR-додатки, архітектурні візуалізації, симуляції та інші прогресивні області.

Перелік використаних джерел.

1. NVIDIA Developer. NVIDIA RTX Real-Time Ray Tracing. https://developer.nvidia.com/rtx

2. Guo H. The Application of Virtual Reality Technology and Real-Time Rendering on Modern Platforms. CAD Journal, 2024, Vol. 21(S28), pp. 238–251.

3. Wang L., Shi X., Liu Y. Foveated Rendering: A State-of-the-Art Survey. arXiv preprint arXiv:2201.02777, 2022.