МАТЕРІАЛИ ІХ ВСЕУКРАЇНСЬКОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ

ІНФОРМАЦІЙНЕ СУСПІЛЬСТВО: «ПРОБЛЕМИ ТА ПЕРСПЕКТИВИ»

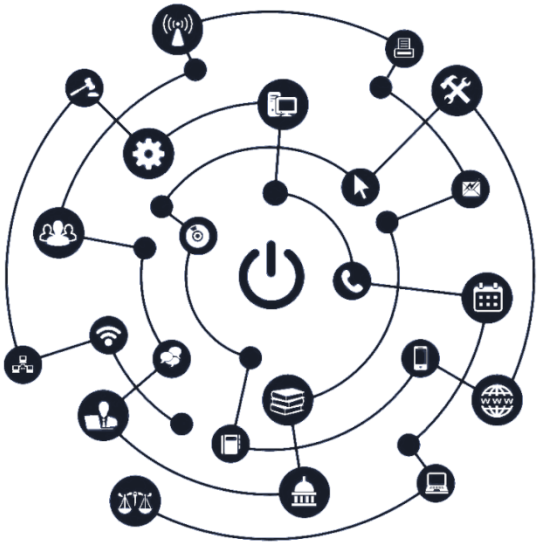
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»

ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ОДЕСА
24 травня 2024 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Одеська юридична академія»
Факультет кібербезпеки та інформаційних технологій
Кафедра інформаційних технологій



ІНФОРМАЦІЙНЕ СУСПІЛЬСТВО: ПРОБЛЕМИ ТА ПЕРСПЕКТИВИ

МАТЕРІАЛИ IX ВСЕУКРАЇНСЬКОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ

24 травня 2024 року, м. Одеса

Одеса, 2024

УДК 340.12:316.3

ББК 67.0

Відповідальній редактор:

Н. І. Логінова, завідувачка кафедри інформаційних технологій, к.пед.н., доцент

Друкується за рішенням кафедри інформаційних технологій
Національного університету «Одеська юридична академія»
(протокол засідання кафедри № 11 від 23.05.2024 р.)

Матеріали IX Всеукраїнської науково-практичної конференції «Інформаційне суспільство: проблеми та перспективи» (м. Одеса, 24 травня 2024 р.). Одеса, 2024. 203 с.

Всеукраїнська науково-практична конференція «Інформаційне суспільство: проблеми та перспективи» присвячена розгляду актуальних питань розвитку інформаційного суспільства в Україні.

До участі у конференції запрошені студенти, аспіранти, молоді вчені, викладачі та науковці. Конференція проводиться на базі Національного університету «Одеська юридична академія».

ПЕРЕДМОВА

Сучасні інформаційно-комунікаційні технології пронизують практично усі сфери діяльності людини: політику, економіку, бізнес, освіту, охорону здоров'я, державне управління, соціальну сферу, дозвілля тощо. Використання інформаційно-комунікаційних технологій дозволяє підвищити ефективність прийняття багатьох рішень за рахунок своєчасного отримання необхідної інформації.

Наразі спостерігаються бурхливі процеси розвитку інформаційних відносин, які є наслідком еволюції інформаційно-комунікаційних технологій. Ці процеси несуть як позитивний ефект, що відображається в забезпеченні прав громадян на інформацію та реалізації їх інформаційних потреб, так і потенційну небезпеку зловживання певною інформацією та заподіяння шкоди інформаційним правам та інтересам особи. За умов воєнного часу актуальними стають питання вибору достовірної інформації, попередження несанкціонованого та безконтрольного поширення інформації.

Технологічні та правові аспекти комп'ютерних злочинів, інформаційна та кібернетична безпека, особливості обробки даних за допомогою інформаційно-комунікаційних технологій в різних галузях знань, електронне врядування, захист інформації, інформаційні війни, право на інформацію, надійність та ефективність сучасних інформаційно-комунікаційних технологій, принципи формування та функціонування інформаційного середовища, управлінські та економічні аспекти розвитку сучасної IT індустрії – ось далеко неповний перелік питань, що цікавлять сьогодні молодіжне наукове середовище, і які склали основну тематику доповідей учасників конференції.

Ми сподіваємося на подальшу плідну роботу наукової молоді та цікавий обмін досвідом щодо розвитку сучасного інформаційного суспільства в Україні.

Секція 1. СУЧАСНІ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНІ ТЕХНОЛОГІЇ В ЖИТТІ ЛЮДИНИ ТА СУСПІЛЬСТВА

АНАЛІЗ РОЛІ QA ІНЖЕНЕРА У ЗАБЕЗПЕЧЕННІ ЯКОСТІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Атанасевич Анастасія.

студентка 3-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Трофименко Олена

кандидат технічних наук, доцент, доцент кафедри інформаційних технологій Національного університету «Одеська юридична академія»

Забезпечення якості (Quality Assurance, QA) створюваного програмного забезпечення є життєво важливою складовою діяльності IT-компаній, які прагнуть надавати клієнтам високоякісні програмні продукти та послуги. Забезпечення якості поєднує планування, впровадження, моніторинг і вдосконалення процесів та стандартів для забезпечення якісних результатів [1]. Є різні фахівці, які працюють у галузі забезпечення якості: інженер із забезпечення якості (QA Engineer), спеціаліст із забезпечення якості (QA Specialist), тестувальник (Tester), аналітик із забезпечення якості (QA Analyst), менеджер із забезпечення якості (QA Manager) тощо [2]. Кожна з цих професій має свої нюанси та вимоги до навичок у роботі.

QA Engineer (QE) є фахівцем, який є відповідальним за дизайн, реалізацію та впровадження стратегій тестування для того, щоб програмні продукти відповідали необхідним стандартам якості. У QE поділяють на Manual та Automation [3]. Manual-тестувальники проводять експерименти, щоб перевірити відповідність вимогам, проаналізувати ризики, дослідити програму для відповіді на поведінку, яка зазвичай не документується, і спробувати виявити невідоме в програмі. Тобто вони в ролі звичайного користувача самостійно проходять через усі етапи взаємодії з програмою. Проте деякі консультанти з тестування сьогодні активно борються з використанням «ручних QA» (Manual QA Engineer), стверджуючи, що насправді вони не гарантують якість [4]. Сьогодні існує сильна тенденція до автоматизованого тестування, але інколи таке тестування недоцільне, адже вимагає багато часу та витрат. Ба більше, реальну взаємодію користувача просто неможливо автоматизувати на 100%. Automation-тестувальники

перевіряють функціонал продукту за допомогою спеціальних програмних інструментів тестування API (Postman, SoapUI, Katalon) для автоматизованих перевірок, які пропонують попередньо визначені фрагменти для автоматизації тестування API без потреби написання коду, або власноруч пишуть код автотестів засобами тих чи інших IDE (VSCode, Visual Studio, IntelliJ IDEA).

Систематизація та аналіз вимог на вакантні посади QE (Manual та Automation) на сайті DOU з пошуку роботи виявив доволі широкий набір вимог до претендентів, а саме [5]: знання англійської мови (зазвичай рівень B1+); розуміння процесів забезпечення якості та SDLC, включаючи гнучкі методології; досвід роботи в процесі розробки Agile/Scrum; досвід комерційного тестування (як ручного, так і автоматизованого); здатність визначати стратегію автоматизації тестування; вміння писати тест-плани, тест-сценарії, тест-кейси; знання архітектури клієнт-сервер, DNS, HTTP/HTTPS; володіння Dev Tools, Postman, Swagge, REST API, Cypress, Playwright, Google Analytics, Amplitude; знання баз даних (SQL Server, MongoDB) та мов програмування (JavaScript, Python тощо), вміння писати автотести; досвід створення фреймворка автоматизації тестування з нуля; досвід роботи з відстеження помилок, інструментами документування (JIRA, Confluence); інтеграція автоматизованих тестів у конвеєр збірки та розгортання; досвід роботи з інструментами та технологіями автоматизації Selenium WebDriver, Selenide, JUnit тощо [4]. Крім того, у вимогах завжди зазначають важливі для роботи QE софт-скіли: комунікативні навички для ефективної співпраці з командою розробників і зацікавленими сторонами; хороші технічні навики, самонавчання, здатність глибоко занурюватися в проблеми, гнучкий режим роботи; досвід лідерства, здатність керувати та наставляти команду інженерів з автоматизації контролю якості тощо.

Гарні комунікативні навички важливі для всіх членів команди розробників, та особливо для QE. Саме завдяки їм та якісній співпраці між відділами підвищується якість продукту та/або послуг. Способи комунікації вибираються, залежно від потреб та можливостей компанії, тому тут дуже важко сформувати чіткий список.

Залежно від розміру команди, структури та конкретних потреб організації, роль QE розглядається та використовується по-різному. Зазвичай QE (інженер із забезпечення якості) – це ключовий спеціаліст із забезпечення якості. Саме він проєктує, розробляє та виконує тести для перевірки якості продуктів або послуг. QE може працювати над різними аспектами якості, як-от: функціональність, продуктивність, зручність використання, безпека та сумісність. Він використовує

різні інструменти та методи для виконання ручного або автоматизованого тестування, наприклад: тест-кейси, сценарії, фреймворки та відповідне спеціалізоване програмне забезпечення. Все це спрямовано на те, щоб виявити дефекти в системі та переконатися, що вимоги до тестування відповідають цілям тестування. Під час виконання тесту QE порівнюють фактичні результати з результатами, зазначеними в тестовому випадку, реєструючи дефект, якщо вони не збігаються. Виявлені дефекти повідомляються іншим відділам або іншим членам команди у *звітах про стан виконання тесту* або через будь-який інструмент відстеження та керування помилками, наприклад, Jira [2]. Також QE щоденно документує та звітує про результати та недоліки тестування, а тому мати технічні, аналітичні та творчі навички.

Список використаних джерел:

1. О.Трофименко, Пастернак Ю.Ю., Манаков С.Ю., Лобода Ю.Г. Автоматизація тестування вебсайтів електронної комерції. Сучасна спеціальна техніка. 2021. № 2(65). С. 46-59. DOI: https://doi.org/10.36486/mst2411–3816.2021.2(65).5.
2. What are the most important roles in quality assurance? URL: https://www.linkedin.com/advice/0/what-most-important-roles-quality-assurance-poiqe.
3. Які бувають тестувальники і що вони роблять. URL: https://goit.global/ua/blog/qa-manual-vs-qa-automation/
4. QA Engineering Roles: Skills, Tools, and Responsibilities in a Testing Team. URL: https://www.altexsoft.com/blog/qa-engineering-roles-skills-tools-and-responsibilities-within-a-testing-team/
5. Вакансії в категорії QA. URL: https://jobs.dou.ua/vacancies/?category=QA

Ключові слова: тестувальник, QA, якість, забезпечення якості
Keywords: testing, QA, quality, quality assurance

МЕТОДИ АНАЛІЗУ ПРИ ФОРМУВАННЯ ПРОФІЛЮ КЛІЄНТА БАНКУ ТА ЇХ НЕДОЛІКИ

Бондаренко Сергій

студент 1 курсу -го курсу магістратури центру заочного та дистанційного навчання Національного університету «Одеська юридична академія»

Створення профілю клієнта є однією з головних передумов для досягнення банками своєї головної мети – отримання прибутку. Крім того, організація клієнтської бази за рахунок упорядкування інформації також підвищує стабільність і конкурентоспроможність банку.

Профіль клієнта – це набір характеристик, що відображають ставлення клієнта та рівень прихильності до відповідних банківських продуктів та пропозицій, це допомагає банку адаптувати свої пропозиції для підвищення ефективності роботи з клієнтами [1].

Профілі клієнтів складаються відповідно до правил визначення сегментів банківського ринку та включають чотири основні характеристики [2]: географічне розташування, демографічний статус, особисті характеристики (соціальний статус, стиль життя, індивідуальні риси), моделі поведінки. Ступінь деталізації кожної характеристики залежить від специфіки клієнтського продуктового портфеля банківської установи.

При формуванні профілю аналізуються такі показники [3-4]:

- характеристика району проживання клієнта;
- індивідуальний або корпоративний клієнт;
- рід занять клієнта;
- важливість клієнта для банку;
- відповідність конкретним банківським послугам;
- кількість банків, що працюють з клієнтом;
- обставини, за яких банк припиняє відносини з клієнтом;
- тривалість користування певними банківськими продуктами та послугами;
- умови розпорядження поточними рахунками та кредитом;
- використання сучасних банківських засобів зв'язку;
- розуміння рекламних пропозицій банку;
- канали отримання інформації про банківські послуги.

Залежно від мети створення профілю, він може бути доповнений даними про кредитні операції та виконання договірних зобов'язань клієнта. Для цього розраховуються такі коефіцієнти [5]:

- виконання договірних зобов'язань;
- своєчасність розрахунків;
- відмінності в поведінці залежно від конкретних банківських послуг;
- преференції щодо різних складових продуктового портфеля банку.

Враховуючи наведені вище коефіцієнти, банки можуть виділяти інші структурні елементи своєї клієнтської бази та орієнтуватися на потенційні групи клієнтів, тим самим підвищуючи ефективність розробки цільових маркетингових стратегій.

При роботі з клієнтами найпоширенішими методами збору даних є: інтерв'ю, анкетування, телефонне опитування. До методів вивчення мотивації поведінки споживачів використовуються: глибинні інтерв'ю, групові інтерв'ю, асоціації та групові дослідження [6].

Основним завданням аналізу профілів клієнтів є мінімізація невизначеності у прийнятті рішень щодо формування та використання банківських ресурсів. Для цього використовуються такі методи кластеризації та регресії.

Деякі методи машинного навчання, такі як нейронні мережі, випадковий ліс та градієнтне підвищення, також використовуються для підвищення якості моделі. Методи регресії дозволяють прогнозувати характеристики об'єктів, наприклад, доходи клієнтів, що допомагає банкам надавати адекватні кредитні продукти та інші фінансові послуги.

Для ранжування клієнтів використовуються різні рейтингові методики [5]:

– комбінація показників з присвоєнням рейтингових позицій;

– віднесення клієнтів до категорій на основі системи оцінки;

– розрахунок індексу кожного показника та визначення комплексного індексу з врахуванням вагових коефіцієнтів.

Недоліками існуючих методів є:

– складність визначення допустимих комбінацій значень показників;

– труднощі в застосуванні складних аналітичних методів;

– необхідність систематичного оновлення даних, що може бути дорогим для банків;

– стандартні моделі не враховують індивідуальні особливості окремих споживачів.

Формування профілю клієнта є критично важливим для банківської діяльності. Використання сучасних методів аналізу даних дозволяє банкам приймати більш обґрунтовані рішення, але існуючі недоліки вимагають подальшого вдосконалення методів.

Список використаних джерел:

1. Chen H. Business intelligence and analytics: From big data to big impact. MIS quarterly. 2012. № 36(4).
2. Варцаба В. І. Сучасне банківництво: теорія і практика : навч. посібник. / В. І. Варцаба, О. І Заславська. Ужгород: УжНУ «Говерла», 2018. 364 с.
3. Копилюк О. І. Аналіз банківської діяльності : навч. посібник / О. І. Копилюк, О. М. Музичка. Львів : Львівський торгівельно-економічний університет, 2016. 416 с.
4. Бондаренко Л. А. Ринок банківських продуктів і послуи: поняття та сегментація. Ефективна економіка. 2013. № 6. URL: http://surl.li/uaono

5. Дубчак Л. В. Інформаційні системи і технології в банківській діяльності: навч. посіб. / Л. В. Дубчак, Л. А. Клюско, В. Ю. Свириденко. Ірпінь: Видавництво Національного університету державної податкової служби України, 2016. 248 с.
6. Шелудько В. М. Фінансовий ринок. К. : Знання, 2015. 535 с.

Ключові слова: профіль клієнта, аналіз даних, банк, методи, методики

Keywords: client profile, data analysis, bank, methods, techniques

Науковий керівник: доцент Ю. Лобода

ДЕЯКІ АСПЕКТИ РОЗВИТКУ ТЕХНОЛОГІЙ ДОПОВНЕНОЇ РЕАЛЬНОСТІ В ОСВІТНЬОМУ ПРОЦЕСІ

Грезіна Олена

асистент кафедри інформаційних технологій, доктор філософії у галузі права
Національний університет «Одеська юридична академія»

Еволюція технологій доповненої реальності в освітньому процесі є результатом динамічного розвитку цифрових інструментів та їхнього впливу на сучасну освітню парадигму. Доповнена реальність (AR) забезпечує унікальні можливості для інтеграції віртуальних елементів з реальним світом, створюючи нові підходи до викладання і навчання що дозволяє зробити освітній процес більш інтерактивним, візуально привабливим та ефективним, сприяючи кращому засвоєнню матеріалу здобувачами різного рівня підготовки. AR-технології суттєво підвищують мотивацію і залученість здобувачів освіти, надаючи можливість вивчати складні концепції через візуалізацію та взаємодію з тривимірними моделями. Крім того, впровадження AR в освітній процес дозволяє адаптувати освітні програми до індивідуальних траєкторій навчання кожного здобувача, що особливо важливо в контексті, наприклад, інклюзивної освіти.

Аналіз сучасних досліджень свідчить про значний потенціал AR у покращенні освітніх результатів та розвитку критичного мислення, креативності та навичок вирішення проблем. Враховуючи глобальні тенденції цифрової трансформації освіти, дослідження та впровадження доповненої реальності є надзвичайно важливими для забезпечення конкурентоспроможності та інноваційного розвитку освітніх систем.

У роботі авторів [1] розглядається опис доповненої реальності (AR) як технології, яка в реальному часі точно накладає комп'ютерні віртуальні зображення на фізичні об'єкти, доповнюючи реальний світ цифровою інформацією. AR

використовує навколишнє середовище людини та до нього конкретні елементи віртуальної інформації, такі як графіка, звуки та дотики. Така візуалізація може бути представлена за допомогою спеціальних пристроїв, таких як екрани смартфонів чи кишенькових пристроїв, окуляри, монітори. Доповнена реальність також містить технологію одночасної локалізації та відображення (S.L.A.M).

У науковому дослідженні авторів [2] зазначається, що однією з основних переваг використання технологій доповненої реальності в освітньому середовищі є можливість зробити навчання цікавішим та доступнішим для засвоєння. Шляхом застосування технологій доповненої реальності можна здійснити ефективний захоплюючий підхід до викладання, який приверне увагу здобувачів освіти та залучить їх до навчального процесу. Використання яскравих та цікавих можливостей AR дозволить науково-педагогічним працівникам привернути увагу здобувачів і відповідно до методики викладання. Замість сірих, нудних підручників, перед здобувачами можуть з'явитися яскраві 3D моделі з анімованими об'єктами та зрозумілим текстом, який легко запам'ятовується підсвідомо. Використання AR дозволяє зробити заняття ще більш інтерактивним за допомогою 3D симуляторів, які надають можливість виконувати фізичні дії у віртуальному середовищі. Це особливо корисно у випадку дистанційного навчання, де здобувачам важко виконувати практичні завдання в реальному житті. Таким чином, здобувачі освіти можуть навчатися на основі експериментів, не турбуючись про можливі пошкодження себе чи реальних об'єктів, а затрати ресурсів практично дорівнюють нулю (за винятком електроенергії).

За допомогою доповненої реальності здобувачі освіти активно взаємодіють з навчальним матеріалом, що забезпечує більш ефективне засвоєння знань. AR дозволяє візуалізувати абстрактні або складні концепції, що полегшує їх розуміння та запам'ятовування.

Завдяки AR здобувачі освіти можуть отримати реалістичний навчальний досвід, наприклад, вивчаючи анатомію через віртуальні моделі тіла або відпрацьовуючи навички в симуляціях реальних ситуацій.

Отже, у висновку слід зазначити, що розвиток технологій доповненої реальності в освітньому процесі є важливим етапом у цифровій трансформації сучасної освіти. Використання AR в освітніх програмах відкриває нові перспективи для створення більш ефективного, інтерактивного та інклюзивного навчального середовища. Підвищення мотивації здобувачів освіти, покращення засвоєння матеріалу та розвиток ключових навичок стають результатом технологічної

еволюції в освіті щодо покращення якості освіти й підготовки молодого покоління до викликів сучасного цифрового світу.

Список використаних джерел:

1. *Браславська О., Думанський О., & Лемешева Н. (2024) Використання доповненої реальності для покращення процесу вивчення складних наукових концепцій. Перспективи та інновації науки. № 2(36). https://doi.org/10.52058/2786-4952-2024-2(36)-104-115*
2. Алексов С., & Дідик А. (2023). Залучення технологій доповненої реальності в освітній процес. Трансформаційна економіка. № 1 (01), 5-9. https://doi.org/10.32782/2786-8141/2023-1-1

Ключові слова: інформаційні технології, технології доповненої реальності, освітній процес, цифрові інструменти, візуалізація абстрактних або складних концепцій

Keywords: information technologies, augmented reality technologies, educational process, digital tools, visualization of abstract or complex concepts

АНАЛІЗ ВИКОРИСТАННЯ СУЧАСНИХ ФРЕЙМВОРКІВ JAVASCRIPT ДЛЯ ВЕБРОЗРОБКИ

Гурін Євген

студент 1-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Сучасний ринок frontend-розробки складно уявити у вигляді HTML, CSS та JavaScript (надалі JS) без додаткових засобів і модифікацій. Наразі є багато інструментів для конструювання стилів (CSS-фреймворки та препроцесори) та логіки (JS-фреймворки та бібліотеки), що значно спрощують процес розробки frontend-частини проєкту. Так, фреймворки Angular, Vue та бібліотека React займають близько 95% усього ринку frontend-розробки, що робить їх найпопулярнішими інструментами у веброзробці. Решта 5% – непопулярні чи то вузькопрофільовані фреймворки і бібліотеки: EmberJS для кросплатформи, SvelteJS та BackBoneJS для веброзробки тощо [1].

Наразі найпопулярнішим інструментом веброзробки є React, що, по суті, є opensource JS-бібліотекою, призначеною для розробки інтерфейсів користувачів (UI) та вебзастосунків у вигляді SPA (Single Page Application, укр. односторінкові застосунки) [2]. Засобами React написані такі продукти: Facebook (автори бібліотеки), Instagram, Netflix, Reddit, Uber та ін. Ця бібліотека мультифункціональна і може використовуватись у проєктах будь-якого призначення, проте

інколи React може бути менш ефективним варіантом, ніж інші фреймворки та бібліотеки. До слабких позицій React можна віднести [3]:

- зайвий акцент на користувацькому інтерфейсі;
- слабка вбудована SEO-оптимізація;
- повільний запуск;
- брак документації та вбудованих бібліотек;
- синтаксис може здатися складним для новачків.

Загалом React є гарним вибором для створення середніх та великих вебзастосунків. Попри повільний запуск проєкту, бібліотека надає багато інструментів для керування складним користувацьким інтерфейсом та поточним станом роботи застосунку. Але вона не є універсальним рішенням. Для проєктів, що потребують модифіковану SEO-оптимізацію або швидкого запуску, React не є найефективнішим інструментом.

Angular – TypeScript-фреймворк, розроблений та підтримуваний компанією Google, спеціалізується на створенні масштабованих SPA вебзастосунків. Він має компоненту архітектуру, що дозволяє розробляти користувацький інтерфейс незалежними частинами. Фреймворк використовує статичну типізацію (через TypeScript), що дозволяє реалізовувати більш логічну модифікацію коду та спрощує рефакторинг [4]. Angular широко використовується в enterprise-розробці та в складних системах. До його недоліків відносять:

- великий обсяг коду, порівняно з іншими фреймворками/бібліотеками;
- складність оновлення застосунку для нової версії фреймворка;
- робота з legacy (застарілим) кодом;
- складний вхід для початківців, розробник повинен мати доволі значний багаж знань, навіть для Trainee/Junior позиції.

Попри те, що для невеликих за обсягом проєктів Angular буде занадто потужним, він є універсальним варіантом для проєктів будь-якого призначення. При цьому найкраще фреймворк розкривається саме у складних проєктах, для яких він і був розроблений.

Vue – JS-фреймворк з простим синтаксисом, порівняно з React та Angular. Має реактивну архітектуру, яка надає можливість автоматично оновлювати подання даних, без потреби ручного оновлення DOM. Як і Angular, Vue має компоненту архітектуру. Він гнучкий та масштабований, має віртуальний DOM, а за функціоналом дещо схожий з React. Найбільшою його перевагою є гнучкість, швидкість та малорозмірність. Приклади відомих застосунків, написаних на Vue: GitLab, Grammarly, Trello та ін. До недоліків Vue відносять [5]:

– недостатня підтримка розробників;
– невелика кількість допоміжних та сторонніх бібліотек;
– складна масштабованість;
– обмежені можливості серверного рендерінгу;
– складність кар'єрного зростання.

Кожний з розглянутих інструментів веброзробки має свої особливості та зумовлені ними переваги і недоліки, але всі вони є гнучкими, ефективними й масштабованими, що важливо для сучасного світу веброзробки. Вибір засобу розробки вебзастосунку чи то сайту має базуватися на вимогах до проєкту, знаннях та досвіді веброзробників, а також можливих часових обмеженнях.

За рівнем складності вебпроєкти можна умовно поділити на:

– прості проєкти не потребують тривалого часу розробки та не вимагають обов'язкового використання фреймворків/бібліотек у своєму коді. Вони акцентовані саме на інтерфейс користувача (UI). Здебільшого це «статичні» сайти (сайти-портфоліо, лендінги, блоги), які не потребують взаємодії з користувачем;

– середні проєкти мають, крім функціоналу простих проєктів, покращену взаємодію з користувачем: авторизацію, реєстрацію, форму зворотного зв'язку, базу даних тощо. Прикладами таких проєктів є невеликі інтернет-магазини, вебпортали вишів та шкіл, сайти для роботи з віджетами тощо;

– складні проєкти переважно зосереджені на обробленні великих обсягів даних користувачів та їх безпеки, мають складну бізнес-логіку та можливість масштабування. Прикладами таких проєктів є соціальні мережі, системи онлайн-банкінгу, корпоративні портали, великі онлайн-магазини, освітні платформи, медичні системи тощо.

Для простих проєктів краще використовувати чистий («ванільний») JavaScript [6]. Аргументується це просто – навіщо фреймворки/бібліотеки, якщо обсяг коду невеликий? Проте, якщо і треба використати якийсь фреймворк, то найкращим варіантом тожі буде Vue. Він простий, зручний у використанні, а його недоліки у проєктах цієї складності не відчутні.

Для середніх проєктів немає одного найкращого варіанта, всі три інструменти є ефективними. І React, і Angular, і Vue не мають суттєвих відмінностей у функціоналі для цих проєктів. Справа лише у виборі веброзробника та стеку технологій, який він використовує.

Для складних проєктів найефективнішим фреймворком є Angular. Фреймворк надає потужні інструменти для створення складних SPA застосунків з великою кількістю компонентів та складною бізнес-логікою. Його функціонал

дозволяє розробляти й контролювати enterprise-застосунки та інші складні проєкти. Також слід зазначити вбудовану систему керування станом проєкту в Angular, інструменти для тестування та рефакторінгу, що робить його найкращим вибором для складних проєктів. Так само ефективно можна використовувати й React у складних проєктах, проте у цій бібліотеці немає вбудованих засобів та інструментів, що спрощують розробку, які є в Angular. Однак їх можна додавати у проєкт разом з React (наприклад – фреймворк для React Next.Js) [7].

Вибір інструментарію веброзробки залежить від персональному смаку та досвіду. Можна писати сайти-портфоліо на Angular або робити складні проєкти на чистому JavaScript, але чи є в тому сенс? Найкращим вибором для розробника буде технологія, яку він знає на достатньому рівні, та може ефективно її використовувати.

Список використаних джерел:

1. Сучасні JS-фреймворки. URL: https://spacelab.ua/articles/top-10-frejmvorkiv-javascript-u-2023-roci/
2. React. URL: https://legacy.reactjs.org/
3. Що таке ReactJS. URL: https://dan-it.com.ua/uk/blog/chto-takoe-react-js-i-dlja-chego-on-nuzhen/
4. Angular. URL: https://angular.io/docs
5. What is Vue? URL: https://vuejs.org/guide/introduction
6. Vanilla JS. URL: https://foxminded.ua/vanilla-js/
7. Angular vs React. URL: https://foxminded.ua/angular-vs-react/

Ключові слова: веброзробка, JS-фреймворки, Angular, React, Vue
Keywords: web development, JS frameworks, Angular, React, Vue

Науковий керівник: доцент О. Трофименко

РОЛЬ ВЕКТОРНОЇ АЛГЕБРИ В РОЗРОБЦІ ІГОР

Гусельніков Ілля

студент 2-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Наразі векторна алгебра, як розділ математики, є визначальним інструментом у розробці цікавих та захоплюючих ігор. Вектори відіграють важливу роль у визначенні руху ігрових об'єктів, взаємодії між ними та створенні реалістичного світу гри.

Вектори є фундаментальною математичною концепцією, яка дозволяє

описати напрямок і величину. В іграх і програмах вектори часто використовуються для опису таких фундаментальних властивостей: положення персонажа, швидкість руху, відстань між двома об’єктами тощо. Кожен об'єкт у грі має свої координати, які визначають його положення у віртуальному світі. Застосування векторної арифметики дозволяє програмістам точно визначити шлях, яким має рухатися персонаж чи то інший об'єкт, забезпечуючи при цьому плавність та реалістичність руху [1].

Приклад коду на C# для ігрового середовища Unity 3D демонструє метод руху гравця у двовимірному просторі, де метод Update() виконується циклічно з кожним оновленням кадру:

```
public float moveSpeed = 5f; // Швидкість руху гравця
void Update()
{
    float horizontalInput = Input.GetAxis("Horizontal");
    float verticalInput = Input.GetAxis("Vertical");
    Vector3 movement = new Vector3(horizontalInput, verticalInput, 0f) * moveSpeed * Time.deltaTime;
    transform.position += movement;
}
```

Тут moveSpeed – це швидкість руху гравця; Time.deltaTime – час між поточним та попереднім кадрами, його значення відповідає за те, щоб рух гравця був плавним, без ривків. Це значення підлаштовується автоматично під фактичну частоту кадрів на конкретному комп’ютері.

Виклик конструктора Vector3 з параметрами horizontalInput, verticalInput та 0f створює вектор movement, який задає напрям руху гравця у двовимірному просторі, з компонентами X та Y відповідно до команд користувача (натискання клавіш керування рухом гравця), а компонента Z залишається нульовою, оскільки не враховується рух у третьому (глибинному) вимірі.

Вектор movement є нормалізованим, тому треба помножити його на швидкість гравця moveSpeed і на час оновлення одного кадру. По суті, такий вектор можна розглядати як кінцеву точку, в якій має опинитися гравець. Набір такого роду векторів на координатній площині формує відстані від початку координат (локації на ігровому полі) до гравця і до всіх наявних об’єктів на ігровому полі. Додавання вектора руху гравця movement до вектора попередньої позиції гравця (поточних координат об’єкта) transform.position формує рух об’єкта.

Векторна алгебра використовується для моделювання фізичних явищ і забезпечення реалістичного сприйняття гравцями ігрового світу. По-перше, завдяки застосуванню векторної алгебри можна реалістично відтворити

гравітацію. Так, за допомогою векторів можна визначити силу тяжіння, яка діє на об'єкт, та відповідно визначити його шлях руху. По-друге, векторна арифметика дозволяє моделювати силу тертя між поверхнями у грі, що дозволяє реалістично моделювати різні варіанти взаємодії гравця з наявними на полі об’єктами. Тобто можна в різний спосіб відтворювати рух об'єкта при взаємодії з різними типами поверхонь: відштовхуватись від стіни чи то іншої поверхні, змінювати рух від удару іншого гравця або вибуху з урахуванням маси та відстані до об’єкта впливу тощо.

Список використаних джерел:

1. Mathf. Unity documentation. URL: https://docs.unity3d.com/Manual/class-Mathf.html

Ключові слова: вектор, векторна алгебра, розробка ігор
Keywords: vector, vector algebra, game development

Науковий керівник: доцент О. Трофименко

ОПТИМІЗАЦІЇ РЕКУРСИВНИХ АЛГОРИТМІВ ЗАСОБАМИ МЕМОІЗАЦІЇ

Дирда Олександра

студентка 2-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Трофименко Олена

кандидат технічних наук, доцент, доцент кафедри інформаційних технологій Національного університету «Одеська юридична академія»

У комп'ютерних мережевих системах доволі часто виникають різні аномалії, які можуть призводити до критичних втрат даних або несанкціонованого доступу в системах [1]. Аномалія переповнення буфера є важливою проблемою, що становить серйозну загрозу безпеці мережі та даних.

Для запобігання виникнення подібних аномалій застосовують різні техніки оптимізації. Метод оптимізації «мемоізація» використовується при створенні налаштованого набору правил. Він підвищує швидкість й ефективність процесу аналізу коду. Як наслідок аналіз коду не повторюється для тих фрагментів коду, які були проаналізовані раніше, тим самим значно скорочується час оброблення та використання ресурсів [1].

Мемоізація є особливою формою кешування, яка використовується в

динамічному програмуванні. Метою кешування є підвищення продуктивності програм і підтримка доступності даних, які можуть бути використані згодом [2]. При такому підході опрацьовані дані динамічно кешуються в локальному буфері мемоізації. Коли результат обчислення є таким самим, як результат, вже наявний в кеші, то він не обчислюється, а одразу береться з буфера мемоізації, тим самим скорочуються витрати пам'яті та часу на зайві обчислення [3]. Тобто відбувається зберігання обчисленого результату підзадачі і його використання для іншої підзадачі. Це скорочує додаткові зусилля та витрати на повторні перерахунки підзадач [2]. Така техніка запам'ятовування застосовується тоді, коли використання попередньо обчислених результатів стає очевидним. Цей вид задачі здебільшого використовується в контексті рекурсії.

Реалізувати концепцію мемоізації на рівні програмування можна будь-якою мовою та у будь-якому середовищі розробки, використовуючи відповідні структури даних та методи. Так, у мовах Python, JavaScript і Ruby її можна реалізувати за допомогою словників або об'єктів [4].

У дослідженні методу «мемоізації» для оптимізації коду авторами проаналізовано алгоритм множинної рекурсії підрахунку числа Фібоначчі. Цей алгоритм має експоненціальну складність O(2^*n*).

Рекурсивна функція обчислення чисел Фібоначчі без мемоізації в C++ може мати вигляд:

```
// Обчислення чисел Фібоначчі
// Множинна рекурсія зі складністю O(2^n)
int Fibonacci(int n) {
     if (n <= 1)
          return n;
     else
          return Fibonacci(n - 1) + Fibonacci(n - 2);
}
```

Цей код рекурсивно будує дерево викликів, що і пояснює його експоненціальну складність. Тут у кожного значення (вузла) обов'язково є два нащадки. Рекурсивні виклики для них зберігаються у стеку до їх опрацювання. Головною небезпекою таких об'ємних рекурсій є швидке переповнення стека, за якого відбувається аварійне завершення роботи програми.

Проведене дослідження виявило максимально можливе число, яке здатна опрацювати ця рекурсивна функція до переповнення стека викликів. Код був запущений 10 разів для зростального на 1 набору натуральних чисел; кожного разу максимально оброблене число становило 57, після якого програма

завершувалася переповненням стека. Середньоарифметичні результати роботи рекурсивної функції без мемоізації для зростальної послідовності натуральних чисел наведено на рис. 1.

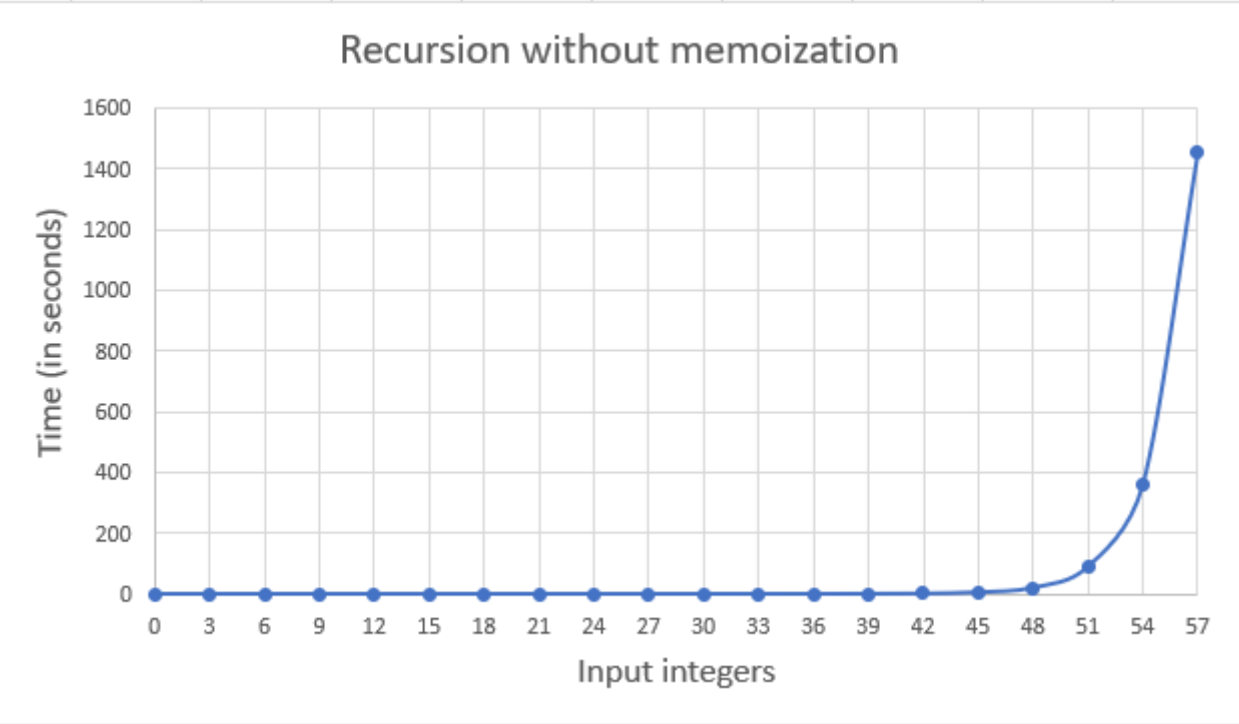


Рисунок 1 – Графік часової складності виконання алгоритму без мемоізації

Оптимізація коду за допомогою мемоізації полягає в кешуванні даних, яку можна реалізувати в різний спосіб. Для її реалізації мовою C++ в роботі використовували структуру даних «map», яка по суті є асоціативним масивом даних «ключ» – «значення». У ключах зберігається вхідне число, а результат його обробки є значенням цього ключа. Кожного разу перед обробленням вхідного числа перевірятиметься наявність результату в асоціативному масиві. Якщо таке число є, то воно повертається як результат, інакше – воно обробляється та дописується у масив як нове значення. Рекурсивна функція обчислення чисел Фібоначчі з мемоізацією в C++ має вигляд:

```
// Оптимізована множинна рекурсія зі складністю O(n)
// за допомогою мемоізації
unordered_map<int, int> memo;
int FibonacciMemo(int n) {
if (n <= 1)
     return n;
if (memo.find(n) != memo.end())
     return memo[n];
else {
```

```
        memo[n] = FibonacciMemo(n - 1) + FibonacciMemo(n - 2);
        return memo[n];
    }
}
```

Результати для зростального на 1 набору натуральних чисел у діапазоні до 1000 не виявили жодних проблем обчислень, переповнення стека не виникало. Тому з метою виявлення максимально можливого числа, для якого відбуватиметься переповнення стека (адже функція теж є рекурсивною), крок між вхідними цілими числами був збільшений до 500. Для чистоти експерименту кожного разу асоціативний масив очищався від раніше збережених результатів:

```
for (int num = 0; num < 100000; num += 500) {
    start = clock();
    fibonacciMemo(num);
    memo.clear();
    end = clock();
    time = ((double)(end - start))/ CLOCKS_PER_SEC;
     cout << "\nTime of processing of number " << num <<
 " with memoization: " << time;
}
```

На рис. 2 наведено залежність часу роботи алгоритму з мемоізацією від значення вхідних даних.

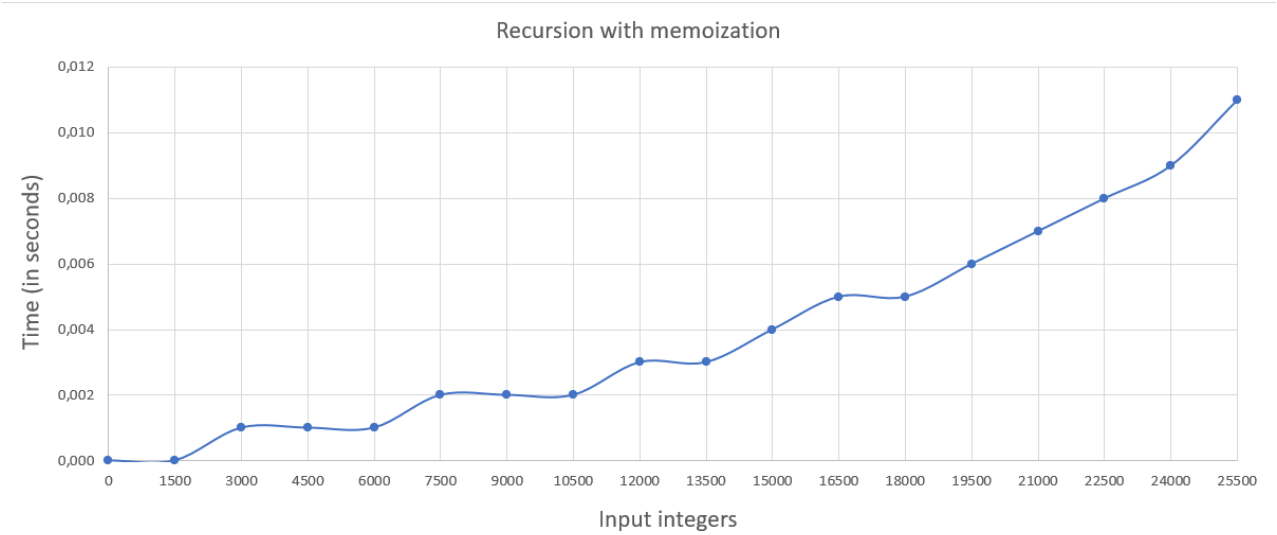


Рисунок 2 – Графік часової складності виконання алгоритму з мемоізацією

Аналіз результатів роботи алгоритму Фібоначчі з мемоізацією виявив переповнення стека, коли число досягло значення 25500. Тобто діапазон можливих вхідних чисел розширився майже у 447 разів без значних часових витрат. Складність алгоритму покращилася до лінійної O(*n*). Отже, оптимізація

рекурсивного алгоритму за допомогою мемоізації допомагає обробляти набагато більші діапазони чисел за відносно швидкий час.

Переваги оптимізації рекурсивного алгоритму засобами мемоізації:

– *ефективність*: мемоізація дозволяє уникати повторного виконання рутинних обчислень, зберігаючи результати в пам'яті;

– *швидкість*: значно скорочується час роботи рекурсивної програми;

– *простота використання*: мемоізація легко імплементується, особливо в мовах програмування високого рівня.

Недоліки оптимізації рекурсивного алгоритму засобами мемоізації:

– *використання пам'яті*: мемоізація вимагає зберігання результатів у пам'яті, що може бути проблематичним для великих обсягів даних;

– *вартість зберігання*: час, потрібний для зберігання та пошуку результатів в пам'яті, може бути значним, особливо для великих обсягів даних;

– *непостійність*: мемоізація не ефективна, якщо функція є непостійною або залежить від зовнішніх змінних, які можуть змінюватися в процесі виконання програми.

Мемоізація є лише одним з численних методів оптимізації коду. Концепція цього методу полягає в кешуванні результатів оброблення даних у пам'яті, що оптимізує їх подальше опрацювання даних. Вона пришвидшує виконання коду, робить його більш ефективним та зручним, але для цього потрібна наявна пам'ять, як і для будь-якого рекурсивного алгоритму. Таку оптимізацію можна реалізувати засобами різних мов програмування, використовуючи притаманні їм методи та структури даних. Проведене дослідження оптимізації рекурсивного алгоритму на прикладі обчислення чисел Фібоначчі підтвердило ефективність методу мемоізації.

Мемоізацію варто використовувати в контексті рекурсій та у випадках, де основна задача розбивається на підзадачі. Перед використанням варто добре проаналізувати задачу, оцінити переваги та недоліки того чи іншого методу оптимізації, у тому числі й мемоізації, щоб зрозуміти можливі переваги та ризики його застосування.

Список використаних джерел:

1. Özger O., Öztekİn H. Detection of Buffer Overflow Attacks with Memoization-based Rule Set. Journal of Computer Science Research. 2023. Vol. 5. P. 13-26. DOI: 10.30564/jcsr.v5i4.6044. https://doi.org/10.30564/jcsr.v5i4.6044
2. What is memoization? A Complete tutorial. URL: https://www.geeksforgeeks.org/what-is-memoization-a-complete-tutorial/

3. Franyell S., Gem D., Jose-Maria A., Antonio G. Neuron-level fuzzy memoization in rnns. Annual IEEE/ACM International Symposium on Microarchitecture. 2019. P. 782–793. DOI: 10.1145/3352460.3358309. https://dl.acm.org/doi/10.1145/3352460.3358309.
4. What is Memoization? How and When to Memoize in JavaScript and React. URL: https://www.freecodecamp.org/news/memoization-in-javascript-and-react/#what-is-memoization

Ключові слова: мемоізація, рекурсія, кешування, оптимізація алгоритму, O(n)
Keywords: memoization, recursion, caching, algorithm optimization, O(n)

СОРТУВАННЯ В JAVASCRIPT

Карагуц Олександр
студент 2-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Мова JavaScript, як більшість інших мов програмування свій має вбудований бібліотечний метод сортування sort(), який ще поширено називають стандартним, оскільки він усталено заданий в налаштуваннях. Задачею методу sort() впорядкування елементів того масиву, для якого його застосовують. Усталено порядок сортування відбувається за абеткою, тобто в порядку зростання Unicode. Програмістів-початківців це може спантеличити, бо числа сортуватимуться не за зростанням числових знань, а по значеннях кодів символьного подання елементів масиву (рис. 1).

```
> [6, 4, 1, 2, 33, 52].sort()
< ▶ (6) [1, 2, 33, 4, 52, 6]
> [6, 4, 1, 2, 33, 52].sort((a,b)=>a-b)
< ▶ (6) [1, 2, 4, 6, 33, 52]
> [6, 4, 1, 2, 33, 52].sort((a,b)=>b-a)
< ▶ (6) [52, 33, 6, 4, 2, 1]
```

Рисунок 1 – Приклад сортування стандартним методом JavaScript

Позаяк JavaScript є скриптовою мовою програмування, то її реалізація декілька відрізняється в різних браузерах. Це стосується і зовсім різних алгоритмів сортування, задіяних для стандартного методу sort() в JavaScript, при виконанні у різних браузерах. Дослідження виявило:

– десктопний Chrome використовує швидке сортування, однак на iOS він використовує сортування злиттям;

– Firefox використовує сортування злиттям, але на малих масивах переключається на сортування вставками;

– Safari на MacOS та iOS використовує сортування злиттям;

– Opera використовує також сортування злиттям;

– Konqueror та rekonq використовують сортування бінарним та червоно-чорним деревом.

Оскільки NodeJS використовує V8 (двигун від Chrome), то і він використовує швидке сортування.

Запустивши в NodeJS стандартний метод sort() декілька раз для різної кількості елементів масиву випадкових чисел, отримано результат, показаний на рис. 2.

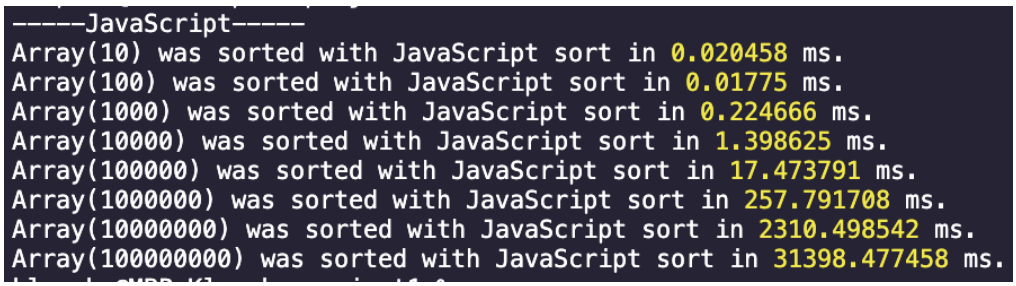


Рисунок 2 – Результат роботи стандартного методу sort() в NodeJS

Також автором протестовано роботу алгоритму швидкого сортування в NodeJS за таких самих вхідних даних (рис. 3).

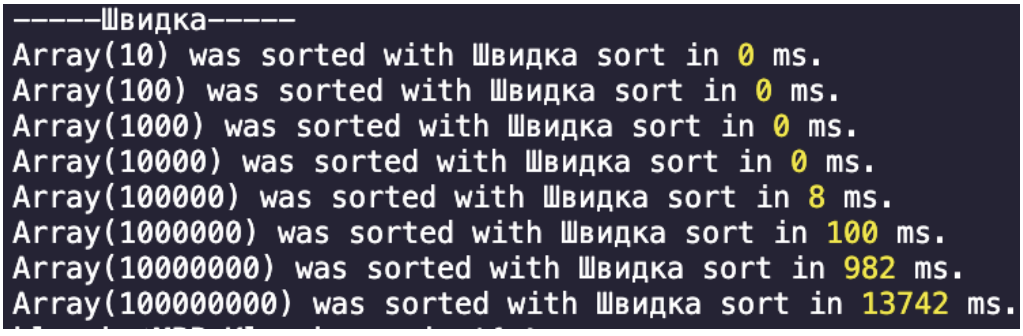


Рисунок 3 – Результат роботи алгоритму швидкого сортування в NodeJS

Порівняння результатів роботи цих сортувань дозволив з'ясувати, що швидке сортування для масиву з 100 000 000 випадкових елементів виконано за 13.7 секунд. Це приблизно в 2.3 рази краще за стандартне сортування. Однак, варто враховувати те, що в метод ще передається функція, яка вказує, як

потрібно сортувати масив. Ймовірно, що як раз це і робить його повільнішим за швидке сортування.

Список використаних джерел:

1. Mathf. Unity documentation. URL: https://docs.unity3d.com/Manual/class-Mathf.html

Ключові слова: вектор, векторна алгебра, розробка ігор
Keywords: vector, vector algebra, game development

Науковий керівник: доцент О. Трофименко

ОСОБЛИВОСТІ РОЗРОБКИ ТЕЛЕГРАМ-БОТІВ

Ковальський Олександр
студент 2-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Наразі попит на Телеграм-боти продовжує зростати, адже їх використання дозволяє покращити якість і швидкість обслуговування клієнтів, завдяки гнучкості та зручності комунікації з ними в режимі 24/7. За допомогою Телеграм-ботів бізнес автоматизує процес надання клієнтської підтримки, адже чат-боти можуть працювати цілодобово і без участі людини [1].

Важливою особливістю розробки Телеграм-бота є вибір методу комунікації із сервером, на якому буде розгорнуто бота. Для цього є два основні методи: webhooks та long polling. Головна відмінність між ними полягає у тому, що при використанні webhooks сервер сам повідомляє клієнтській частині програми нову інформацію, а при long polling клієнтська частина програми надсилає до сервера регулярні запити про нову інформацію. Приклади програмного коду мовою Python для використання long polling та webhooks наведено на рис. 1 і 2 відповідно. При цьому залучено бібліотеку TeleBot та фреймворк Flask.

```python
import telebot

API_TOKEN = 'BOT_API_TOKEN'
bot = telebot.TeleBot(API_TOKEN)

bot.polling()
```

Рисунок 1 – Фрагмент коду для використання long polling

```python
import telebot
from flask import Flask, request

API_TOKEN = 'YOUR_BOT_API_TOKEN'
WEBHOOK_URL = 'https://yourdomain.com/YOUR_WEBHOOK_PATH'

bot = telebot.TeleBot(API_TOKEN)

app = Flask(__name__)


@app.route('/YOUR_WEBHOOK_PATH', methods=['POST'])
def webhook():
    json_str = request.get_data().decode('UTF-8')
    update = telebot.types.Update.de_json(json_str)
    bot.process_new_updates([update])
    return '!', 200

bot.remove_webhook()
bot.set_webhook(url=WEBHOOK_URL)

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8443, ssl_context=('cert.pem', 'key.pem'))
```

Рисунок 2 – Фрагмент коду для використання webhooks

При розробленні Телеграм-бота є можливість використовувати два типи кнопок клавіатури для взаємодії користувачів з ботом: ReplyKeyboardButton (кастомні) та InlinekeyboardButton (вбудовані або інлайнові) [2]. Перший з них (custom buttons) використовується для формування набору кнопок з попередньо визначеними відповідями. Коли користувач натискає кнопку, текст з написом кнопки надсилається телеграм-боту як повідомлення. Він забезпечує простий і легкий спосіб взаємодії між користувачем і ботом Telegram. На рис. 3 показано вигляд таких кнопок, а на рис. 4 наведено фрагмент коду для додавання їх у Телеграм-боті.

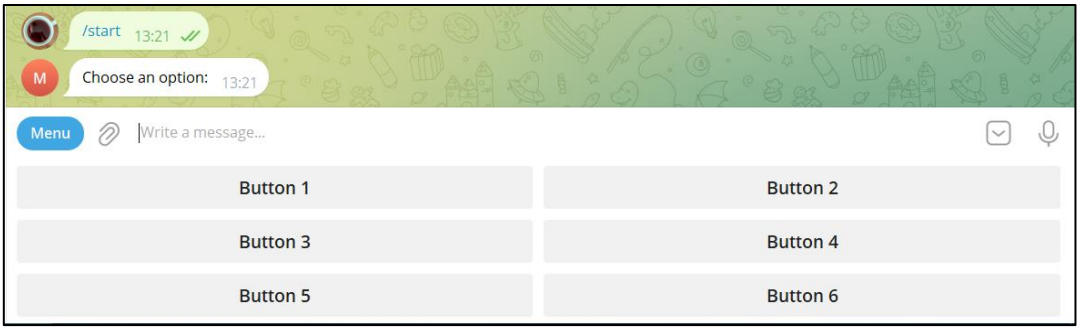


Рисунок 3 – Вигляд Reply кнопок

```python
@bot.message_handler(commands=['start'])
def send_welcome(message):
    # Creating a custom keyboard with two buttons in one row
    keyboard = types.ReplyKeyboardMarkup(row_width=2, resize_keyboard=True)
    button1 = types.KeyboardButton('Button 1')
    button2 = types.KeyboardButton('Button 2')
    button3 = types.KeyboardButton('Button 3')
    button4 = types.KeyboardButton('Button 4')
    button5 = types.KeyboardButton('Button 5')
    button6 = types.KeyboardButton('Button 6')
    keyboard.add(button1, button2, button3, button4, button5, button6)

    # Sending a message with the custom keyboard
    bot.send_message(message.chat.id, "Choose an option:", reply_markup=keyboard)
```

Рисунок 4 – Код для створення Reply кнопок

InlinekeyboardButton – тип кнопок вбудованої клавіатури (inline button). Такі кнопки прикріплені до певного повідомлення в чаті і при натисканні на них вони надсилають це повідомлення на сервер і викликають певну функцію зворотного виклику (callback).

Вбудовані кнопки використовуються частіше за кастомні, бо надають більше можливостей для взаємодії з ботом. На рис. 5 і 6 показано вигляд і програмний код для створення вбудованих кнопок.

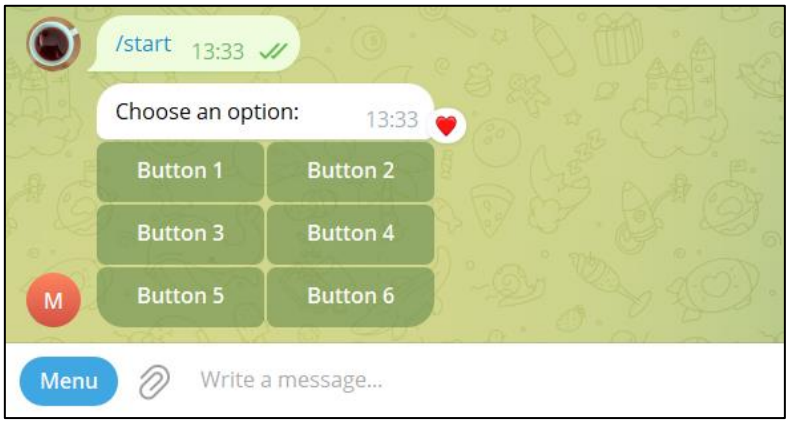


Рисунок 5 – Вигляд Inline кнопок

Ще однією особливістю розробки сучасних Телеграм-ботів є вбудований режим (inline mode) роботи. Раніше для спілкування з користувачами потрібно було відправляти їм повідомлення в окремих чатах або додавати їх у свої групи. Завдяки вбудованому режиму боти стають усюдисущими, і їх можна використовувати як інструмент у будь-якому з чатів, груп чи каналів – не має значення, чи є бот учасником чи ні [3]. Для цього потрібно ввести назву бота і бажаний запит.

```python
@bot.message_handler(commands=['start'])
def send_welcome(message):
    # Creating an inline keyboard with six buttons
    keyboard = types.InlineKeyboardMarkup(row_width=2)
    button1 = types.InlineKeyboardButton("Button 1", callback_data='button1')
    button2 = types.InlineKeyboardButton("Button 2", callback_data='button2')
    button3 = types.InlineKeyboardButton("Button 3", callback_data='button3')
    button4 = types.InlineKeyboardButton("Button 4", callback_data='button4')
    button5 = types.InlineKeyboardButton("Button 5", callback_data='button5')
    button6 = types.InlineKeyboardButton("Button 6", callback_data='button6')
    keyboard.add(button1, button2, button3, button4, button5, button6)

    # Sending a message with the inline keyboard
    bot.send_message(message.chat.id, "Choose an option:", reply_markup=keyboard)
```

Рисунок 6 – Код для створення Inline кнопок

Вбудовані боти корисні для швидкого надсилання відповідних GIF-файлів, зображень з інтернету, відео YouTube, статей у Вікіпедії тощо. Тим самим взаємодія з чат-ботами у Телеграмі суттєво спрощується. Фрагмент програмного коду для використання такого режиму і приклад результатів роботи вбудованого режиму бота наведено на рис. 7 і 8 відповідно.

```python
@bot.inline_handler(lambda query: query.query)
def query_text(query):
    results = []

    article = types.InlineQueryResultArticle(
        id='1',
        title='Example',
        input_message_content=types.InputTextMessageContent(
            message_text='This is an inline mode message!'
        ),
        description='Click to send this message'
    )
    results.append(article)
    bot.answer_inline_query(query.id, results)
```

Рисунок 7 – Код для використання вбудованого режиму роботи

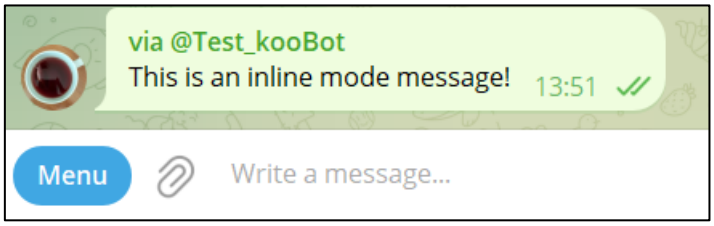


Рисунок 8 – Результати роботи вбудованого режиму

При розробленні чат-ботів слід зважати на те, що засоби та підходи до розробки переглядаються й оновлюються. Розвивається API, з'являються нові методи для забезпечення безперебійної роботи бота, забезпечуючи його сумісність як з груповими, так і з приватними чатами.

Список використаних джерел:

1. О.Трофименко, Прокоп Ю.В., О.Задерейко, Логінова Н.І. Класифікація чат-ботів. Системні технології. 2022. № 2 (139). С. 147-159. DOI: 10.34185/1562-9945-2-139-2022-14. URL: https://journals.nmetau.edu.ua/index.php/st/issue/view/120/87
2. Keyboard buttons in Telegram bot Using Python. URL: https://www.geeksforgeeks.org/keyboard-buttons-in-telegram-bot-using-python/
3. Introducing Inline Bots. URL: https://telegram.org/blog/inline-bots?setln=it.

Ключові слова: чат-бот, Телеграм-бот, Python

Keywords: chat bot, Telegram, Python

Науковий керівник: доцент О. Трофименко

UNREAL VS UNITY: ВИБІР РУШІЯ ДЛЯ СТВОРЕННЯ ВЛАСНОГО ІГРОВОГО ПРОЄКТА

Задерейко Олександр

кандидат технічних наук, доцент, доцент кафедри інформаційних технологій Національного університету "Одеська юридична академія"

Толокнов Анатолій

асистент кафедри інформаційних технологій Національного університету "Одеська юридична академія"

Курчук Роман

студент 4-го курсу факультету кібербезпеки та інформаційних технологій Національного університету "Одеська юридична академія"

З кожним роком ігрова індустрія все більше розвивається та інтегрується в наше суспільство. Станом на 2020-2021 роки в світі налічувалось близько 3 млрд людей, які в тій чи іншій мірі грають в ігри. І з кожним днем ця цифра збільшується. Очевидно, що зі збільшенням кількості гравців необхідно збільшувати кількість та, що головне, розвивати якість створюваних ігрових додатків – покращувати графічну частину, якість звуку, рівень штучного інтелекту та фізики середовища тощо, все більше наближаючи гру до реальності. Звісно ж, для реалізації всіх фантазій та амбіцій як великих компаній-розробників, так і іноді-розробників, їм неохідний набір спеціалізованого програмного забезпечення. Деякі компанії створюють власні рушії для створення проєктів, але більшість віддають перевагу використанню вже наявних, іноді модифікуючи їх під свої потреби.

Існує багато різноманітних ігрових рушіїв, багато з яких обмежені у використанні – Cocos 2D, GDevelop, CryEngine, Godot та інші. Але абсолютними лідерами у використанні є Unreal Engine та Unity. Unreal Engine – рушій, створений компанією Epic Games та реалізований мовою C++. Він дає змогу створювати ігри для більшості існуючих ОС та платформ [1]. На даному рушії створено багато популярних ігор, таких як Fortnite, серії Bioshock та Borderlands, PUBG, серія Gears of War тощо. Unity – рушій, створений Unity Technologies, в якому для написання ігрової логіки використовується мова C# [2]. Unity також є кросплатформовим та підтримує 25 платформ. Даний рушій використовувався для створення ігор Cuphead, Genshin Impact, Plague Inc, Subway Surfers та інш.

Тож в розробників може виникнути питання: який рушій обрати для створення власного продукту? В першу чергу, вибір повинен виходити з особливостей гри: обраний жанр, використання штучного інтелекту в ігровому процесі, стиль графіки, тип гри (2D/3D), цільові платформи тощо. Також треба враховувати власний рівень володіння потрібною мовою програмування, так як в обох рушіях різний поріг входу.

Рушій Unreal використовується в більш масштабних проєктах, в той час як Unity використовується в основному для створенні інді- та мобільних ігор (хоча є винятки, такі як Genshin Impact). Вибір найкраще зробити, розглянувши порівняння обох рушіїв за різними критеріями та визначивши їх переваги/недоліки.

Проаналізувавши створені засобами цих рушіїв ігри та, можна однозначно сказати – в цьому плані Unreal набагато кращий. Epic Games має великий досвід та з роками все більше розвивала власний рушій, що дозволило досягти неперевершеного рівня графіки, при цьому додавши потужні інструменти для зміни графіки під власні потреби. Unity, в свою чергу, виглядає дещо застарілим, так як він орієнтований, в основному, на розробку мобільних або 2D ігор. Безумовно, використовуючи Unity цілком реально досягти якості графіки, не гіршої за Unreal (наприклад, гра Forest), але для цього потрібен неабиякий досвід та велика кількість часу. Також Unreal містить новітні технології, які не тільки покращують, але й полегшують роботу з певною системою. Прикладами таких технологій є:

Nanite – система віртуалізованої геометрії оточення в реальному часі в залежності від відстані до об’єкта. Вона використовує новий формат сітки та технологію візуалізації для відтворення деталей об’єкта у піксельному масштабі [3];

Lumen – динамічна система глобального освітлення та відображень, яка створює дифузне взаємовідбиття з нескінченними відскоками та непрямими

дзеркальними відбиттями у великих деталізованих середовищах різного масштабу (від міліметрів до кілометрів) [4].

Рушій Unity використовує для роботи мову програмування C#, що в купі з незліченною кількістю різноманітних курсів, гайдів та документації забезпечує низький поріг входу для новачків. В Unreal використовується мова C++, яка є складнішою, ніж C#, тому поріг входу в нього вищий. Але для Unreal, як і для Unity, існує безліч курсів (в тому числі безкоштовних), а також зручна та зрозуміла документація на офіційному ресурсі [5], тому з базовими знаннями C++ цілком реально вивчити функціонал рушія та навчитись застосовувати його. І найголовніше – в Unreal є досить вагома перевага, якою не може похизуватись Unity – система візуального програмування Blueprint (Blueprint Visual Scripting). Blueprint заснована на концепції використання вузлів (nodes) для створення елементів та функціоналу ігрового процесу. Дана система є дуже гнучкою та дозволяє використовувати повний спектр можливостей без написання коду. Це дозволяє застосовувати блупринти для створення потрібних функцій та, при необхідності, модифікувати їх. За допомогою системи Blueprint можливо створити повноцінну гру, не використовуючи C++, хоча варто відмітити, що подібні проєкти займатимуть більше місця та працюватимуть повільніше, ніж проєкти, реалізовані на C++, тому найкращим варіантом буде комбіноване використання Blueprint та C++, що дозволить створити проєкт швидше та з меншими затратами, при цьому зберігши швидкодію. В підсумку – в Unity на розробку витрачається набагато більше часу, так як використовується виключно код, тоді як в Unreal завдяки Blueprint можна реалізувати проєкт набагато швидше.

Варто відмітити сумісність різних версій рушіїв. В Unreal Engine можливо перейти на наступну підверсію, повністю перенісши проєкт без виникнення проблем або перейти на новішу версію рушія, зробивши деякі зміни, щоб відповідати особливостям конкретної версії. В обох випадках, перехід на нову версію (підверсію) можливий без повторної повної реалізації проєкта. В свою чергу перехід на нову версію Unity може стати кропітким та понести за собою безліч технічних проблем, що призведе до необхідності створювати проєкт заново, використовуючи нову версію рушія.

Обидва рушія є безкоштовними, але Unreal має відкритий вихідний код, що дозволяє розробникам змінювати та покращувати рушій в залежності від вимог. Нажаль, Unity, протилежно Unreal, має закритий вихідний код, що унеможливлює модифікації рушія. Обидва рушії підтримують інтеграцію сторонніх рішень (наприклад, імпорт моделей, створених в Blender), що значно розширює

можливості їх використання. Обидва рушія використовуються великою кількістю розробників, тому необхідно подбати про ефективну підтримку для користувачів. Epic Games надає повну підтримку, допомогаючи з вирішенням навіть найменшої проблеми, також пропонуючи розгорнуту та зрозумілу документацію. Щодо Unity – є певні проблеми. Отримання фідбеку від розробників може зайняти чимало часу, тому часто знайти рішення самостійно чи з допомогою спільноти може виявитись набагато швидшим. До того ж, різні посібники та документація для Unity може бути складною чи навіть нечитабельною, тоді як для Unreal створюються зручні та зрозумілі посібники – Epic Games навіть надає фінансування компаніям, які створюють якісну документацію.

Звісно ж в обох рушіїв є також власні проблеми. В Unreal є багато технологій, які знаходяться на етапі допрацювання і для яких створено недостатньо документації, що призводить до неправильного використання технології та, як наслідок, поганої оптимізації гри. Також оновлення рушія часто можуть містити баги, хоча більшість з них швидко виправляють. Unreal погано підходить для створення 2D ігор – наявні інструменти просто незручні у використанні. Також для роботи з Unreal треба мати потужний персональний комп'ютер.

Дуже часто нові функції виходять напівготовими і довго допрацьовуються. Unity, як і Unreal, містить баги, але в порівнянні з ним багів набагато більше і вони дійсно заважають роботі. Також Unity не містить готових рішень для реалізації багатокористувальницьких проєктів, внаслідок чого поріг створення подібних ігор вище, ніж в Unreal. Також варто згадати введення нової політики монетизації, яка передбачала стягнення комісії, яка залежатиме від кількості встановлень гри. Хоча монетизація рушія досі вважається вигідною для інді-розробників, багато користувачів втратили довіру до компанії. Внаслідок даних недоліків виникло чимало питань щодо подальшого розвитку рушія, тому багато розробників віддають перевагу іншим рушіям.

Незважаючи на всі наявні проблеми та недоліки, обидва рушія залишаються найпопулярнішими для розробки. Однозначно сказати, який рушій краще неможливо, хоча тут також грає роль суб'єктивна думка кожної людини. Якщо користувач на меті створювати 2D- або мобільні інді-ігри чи підбирає рушій для комфортного входження та розвитку в індустрії геймдеву – Unity стане хорошим вибором. Якщо він бажає створити більш розвинутий проєкт, з розвинутим геймплеєм та хорошою графікою – треба віддавати перевагу рушію Unreal Engine.

Список використаних джерел

1. 1. Del Gallego, Neil Patrick. "Constructing a game engine: A proposed game engine architecture course for undergraduate students." Entertainment Computing (2024): 100657. URL: https://www.sciencedirect.com/science/article/abs/pii/S1875952124000259.
2. 2. Pérez, Carlos, et al. "A Comparative Analysis of Energy Consumption Between The Widespread Unreal and Unity Video Game Engines." arXiv preprint arXiv:2402.06346 (2024). URL: https://arxiv.org/pdf/2402.06346.pdf.
3. 3. Lumen Global Illumination and Reflections | Epic Developer Community. URL: https://dev.epicgames.com/documentation/en-us/unreal-engine/lumen-global-illumination-and-reflections-in-unreal-engine.
4. 4. Abramowicz, Kamil, and Przemysław Borczuk. "Comparative analysis of the performance of Unity and Unreal Engine game engines in 3D games." Journal of Computer Sciences Institute 30 (2024): 53-60. URL: https://ph.pollub.pl/index.php/jcsi/article/download/5473/4438

Ключові слова: рушій, Unity, Unreal, документація, розробка, графіка, доступність та підтримка
Keywords: engine, Unity, Unreal, documentation, development, graphics, availability and support.

ПРОГРАМНА РЕАЛІЗАЦІЯ ПРОЦЕСУ ВІДСТЕЖЕННЯ ПОШИРЕННЯ ПО ПРОГРАМІ НЕПЕРЕВІРЕНИХ ЗОВНІШНІХ ДАНИХ

Колісніченко Руслан

студент 4-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Аналіз текстів програм для пошуку потенційних загроз – це важливий етап у виявленні потенційних проблем в програмному забезпеченні. Цей аналіз зазвичай виконується з метою виявлення можливих вразливостей, вбудованих помилок, порушень безпеки та інших проблем, які можуть стати джерелом загроз для безпеки системи чи користувачів. Програми для пошуку загроз можуть аналізувати вихідний код програмного забезпечення з метою виявлення певних конструкцій або використання патернів, які можуть вказувати на потенційні проблеми безпеки.

Існують кілька методів та технік пошуку потенційних загроз в текстах програм, зокрема так званий taint-аналіз – це метод аналізу безпеки програмного забезпечення, який виявляє потенційно небезпечні дії або вразливості, враховуючи потоки даних через програму. Основна ідея полягає в тому, щоб відстежувати, як дані, які можуть бути контрольовані зловмисником, проникають у систему та як вони обробляються програмою.

Наразі відомі дослідження [1-2], у яких проблема пошуку неперевірених зовнішніх даних вирішується за допомогою застосування техніки символьного виконання коду, яка дозволяє знайти всі набори вхідних даних, що сприяють виконанню кожного з його можливих шляхів.

Символьне виконання (symbolic execution) – це техніка аналізу програмного забезпечення, яка полягає в автоматизованому створенні символьних виразів, які представляють всі можливі шляхи виконання програми. Замість фактичного виконання програми з конкретними вхідними даними, як в традиційному методі тестування, символьне виконання використовує символьні значення для вхідних даних та відстежує, як ці значення впливають на виконання програми через всі можливі шляхи виконання.

Основна ідея полягає у тому, щоб створити символьні вирази для вхідних даних програми та аналізувати вирази, щоб знайти всі можливі шляхи виконання програми, включаючи шляхи, які можуть призвести до помилок або небажаних станів програми. Символьне виконання може бути корисним для виявлення вразливостей програм, таких як витоки пам'яті, ділення на нуль, недоступний код та інші проблеми безпеки.

Актуальність дослідження обумовлена високою затребуваністю програмних систем виявлення дефектів, помилок та вразливостей у програмах на мовах високого рівня, а також необхідністю модернізації процесу відстеження потенційно небезпечних або ненадійних даних, що вводяться у програму зловмисниками і можуть бути використані для атаки.

Метою виконання дослідження є програмна реалізація статичного аналізатору Java-коду для відстеження поширення неперевірених зовнішніх даних за програмою.

Для досягнення мети було сформульовано такі завдання:

– провести огляд предметної галузі та підходів існуючих статичних аналізаторів з метою виявлення їх переваг та недоліків;

– реалізувати модуль, який на основі тестів, згенерованих інструментом автоматичної генерації тестів UnitTestBot [4], буде надаватиме користувачеві звіт із знайденими помилками;

– розробити інтерфейс користувача для перегляду звіту статичного аналізу і вбудувати його в плагін UnitTestBot для IntelliJ IDEA;

– модифікувати ядро символьної віртуальної машини, вбудувавши в нього техніку taint-аналізу, яка дозволить виявляти порушення безпеки в коді;

– протестувати розроблений статичний аналізатор для визначення ефективності знаходження помилок та вразливостей у різних програмах.

Більшість програмістів використовують різноманітні інтегровані середовища розробки, які відрізняються можливістю підключення плагінів – сторонніх модулів для вирішення конкретних завдань. Це дозволяє кожному користувачеві налаштувати необхідний набір плагінів безпосередньо в IDE для зручності програмування, що часто є вигідніше, ніж використання окремих сервісів.

Багато статичних аналізаторів мають свої плагіни для різних середовищ розробки. У цьому випадку було вирішено створити плагін для IntelliJ IDEA [3], що дозволить запускати статичний аналіз і переглядати його результати у форматі звіту SARIF. Плагін здатен виявляти використання зовнішніх даних, які можуть бути потенційними джерелами вразливостей у вашому коді. Плагін має надавати рекомендації щодо того, як зменшити ризик використання таких даних і підвищити безпеку вашого проекту. Розробка базується на плагіні для автоматичної генерації модульних тестів UnitTestBot.

Платформа IntelliJ надає широкі можливості для створення нових модулів розширення. Наприклад, існує стандартний механізм Inspections, що дозволяє відображати список виявлених проблем у вигляді панелі Problems View. Для використання цієї можливості було виконано наступні кроки.

1. Зареєстровано новий InspectionTool у конфігурації плагіна.

2. Перевизначено метод GlobalSimpleInspectionTool.checkFile, який додає виявлені помилки для конкретного переданого файлу psiFile до загального списку проблем для psiFile.

3. Додано автоматичний запуск механізму Inspections відразу після створення тестів і створення звіту SARIF, щоб відобразити знайдені дефекти в IDE.

Нижче представлений сценарій використання модифікованого плагіна.

1. Користувач запускає генерацію тестів у спеціальному режимі, який передбачає подальше відображення результатів статичного аналізу.

2. Після закінчення роботи символьної машини та створення звіту SARIF на екрані відкривається вкладка Problems View, де перераховані знайдені за допомогою плагіну помилки вразливості у вигляді списку повідомлень (поле message у SARIF).

3. При натисканні на будь-який з елементів цього списку, IDE показує відповідний рядок коду (поле location).

4. У вікні праворуч з'являється кнопка для переходу до потрібного згенерованого тесту (поле relatedLocation).

5. Також, є можливість подивитися трасування стека знайденого виключення (поле codeFlows). Причому ця функціональність підтримує навігацію за кодом для кожного представленого кадру стека.

Виконано тестування розробленого плагіну для відстеження поширення по програмі неперевірених зовнішніх даних на різних тестових завданнях. Вся взаємодія з налаштуванням, запуском та результатами аналізатора відбувалася виключно засобами модифікованого плагіна для IntelliJ IDEA.

У ході експериментів порівнювалася ефективність знаходження вразливостей між інструментом SpotBugs та технікою taint-аналізу, реалізованою в роботі, за такими показниками:

– загальна кількість знайдених проблем;

– частка виявлених помилок від кількості дефектів у програмі;

– частка хибно-позитивних спрацьовувань від загальної кількості виявлених проблем.

У якості тестового проєкту обрано Juliet Java [5] від Центру гарантованого програмного забезпечення NSA – це набір синтетичних програм, у яких міститься понад 100 типів уразливостей. Для тестування алгоритмів taint-аналізу обрано розділ із SQL-ін'єкціями. Модифікований плагін зміг знайти 1002 SQL-ін'єкції, що становить 41% з усіх допущених у наборі помилок. При цьому не було видано жодного хибно-позитивного спрацьовування. Інші помилки не були знайдені, в основному, з тієї причини, що інструменту не вдалося згенерувати жодного тесту за наданий йому час.

Інструмент SpotBugs виявив усі вразливості, проте разом з цим знайшов ще приблизно 3000 неіснуючих проблем, що значно більше, ніж загальна кількість реальних помилок у програмах. Таким чином, частота помилкових спрацьовувань становила 57%.

Отримані результати показують, що розроблений модуль taint-аналізу дозволив плагіну UnitTestBot знаходити вразливості виду SQL-ін'єкцій практично у половині випадків на відповідному тестовому наборі.

Список використаних джерел:

1. Чикунов П.О. Відстеження поширення по програмі неперевірених зовнішніх даних / Європейські орієнтири розвитку України: науково-практичний вимір в умовах воєнних викликів : Матеріали Міжнародної науково-практичної конференції (Одеса, 26 квітня 2024 р.). Одеса, вид-во «Гельветика», 2024. С. 100-105.
2. Wang P., Chao W.J., Chao K.M., Lo C.C. Using taint-analysis for threat risk of cloud applications. In IEEE 11th International Conference. IEEE, 2014. Pp. 185-190.

3. Sapozhnikov A., Olsthoorn M., Panichella A., Kovalenko V., Derakhshanfar, P. TestSpark: IntelliJ IDEA's Ultimate Test Generation Companion. arXiv preprint: 2401.06580, Cornell University, 2024.
4. UnitTestBot. URL: https://plugins.jetbrains.com/plugin/19445-unittestbot.
5. Charest T., Rodgers N., Wu Y. Comparison of static analysis tools for Java using the Juliet test suite. In 11th International Conference on Cyber Warfare and Security, 2016. Pp. 431-438.

Ключові слова: taint-аналіз, неперевірені зовнішні данні, символьна машина, IntelliJ IDEA, генератор тестів, плагін, SQL-ін'єкція

Keywords: taint-analysis, unverified external data, symbol engine, IntelliJ IDEA, test generator, plugin, SQL injection

Науковий керівник: доцент П. Чикунов

НАБОРИ ДАНИХ В МАШИННОМУ НАВЧАННІ: ПРОБЛЕМИ, РІШЕННЯ, ПІДХОДИ

Лобода Юлія

доцент кафедри інформаційних технологій, кандидат педагогічних наук, доцент Національного університету «Одеська юридична академія»

При роботі з наборами даних можуть виникати різні проблеми, які впливають на ефективність навчання та точність моделей машинного навчання. Деякі з цих проблем можна вирішити за допомогою спеціальних методів та технік машинного навчання.

Набори даних – це колекції даних, що використовуються для тренування, валідації та тестування моделей машинного навчання. Вони можуть містити різноманітну інформацію, організовану у вигляді таблиць, зображень, текстів або інших форматів. Кожен запис у наборі даних, як правило, складається з ознак (характеристик) та міток (значень, які потрібно передбачити).

Існує кілька основних видів наборів даних, які класифікуються за різними критеріями. Розглянемо види наборів даних детальніше:

За типом даних:

– табличні дані – це структуровані дані у вигляді таблиць, де кожен рядок представляє окремий запис, а кожен стовпчик – ознаку. Наприклад, дані про продажі, медичні записи;

– текстові дані – дані, представлені у вигляді тексту. Наприклад, документи, новинні статті, відгуки користувачів, твіти;

– зображення – дані у вигляді зображень. Наприклад, фотографії, медичні знімки, рукописні цифри (MNIST) [1];

– аудіо дані – дані у вигляді звукових файлів. Наприклад, записи голосу, музика, звукові сигнали;

– відео дані – це дані у вигляді відеофайлів. Наприклад, відеоролики, записи з камер спостереження, фільми;

– часові ряди – дані, що представляють значення, записані у хронологічному порядку. Наприклад, фінансові дані, дані про погодні умови;

– просторові дані – дані, що містять інформацію про географічне положення. Наприклад, карти, GPS координати.

За призначенням:

– навчальні набори даних застосовуються для навчання моделі, містять приклади з відповідними мітками;

– валідаційні набори даних – для налаштування параметрів моделі та оцінки її продуктивності під час навчання;

– тестові набори даних застосовуються для остаточної оцінки продуктивності моделі після завершення навчання.

За структурою:

– розмічені набори даних містять дані з відповідними мітками;

– нерозмічені набори даних містять дані без міток;

– частково розмічені набори даних містять як розмічені, так і нерозмічені дані. Використовуються у задачах частково наглядового навчання;

– набори даних для навчання з підкріпленням – це специфічний вид даних, що містить інформацію про стани, дії та винагороди у середовищі, де агент навчається через взаємодію.

Спеціальні види наборів даних:

– генеративні набори даних – дані, що створені штучно для імітації реальних наборів даних. Використовуються для збільшення розміру набору даних або для специфічних експериментів, наприклад, в автономних автомобілях для симуляцій;

– набори даних для змагального навчання – дані, що містять змагальні приклади, використовуються для тренування моделей, стійких до атак.

Правильний розподіл даних на тренувальний, валідаційний та тестовий набори є ключовим для успішного навчання моделі. Зазвичай використовуються такі співвідношення: тренувальний набір (Training Set): 70%, валідаційний набір (Validation Set): 20%, тестовий набір (Test Set): 10%.

Неперервний моніторинг та оцінка якості наборів даних є важливими для підтримки високої ефективності моделей машинного навчання. Це включає в себе: перевірку наявності пропущених даних, виявлення шуму та аномалій, оцінку розмірності та кореляції ознак, перевірку балансу класів [2]. Додаткові методи оцінки включають візуалізацію даних, аналіз розподілу даних та перевірку наявності структурних змін у даних з часом.

Моделі машинного навчання потребують великої кількості даних для ефективного навчання. Недостатня кількість даних може призвести до недонавчання моделі. Це можна вирішити за допомогою технік генерації синтетичних даних, використання попередньо навчених моделей тощо.

Якщо одна категорія у наборі даних представлена значно більше, ніж інші, модель може погано працювати для менш представлених класів. Методи, як-от підвибірка та надвибірка, а також алгоритми, що враховують ваги класів, такі як SMOTE (Synthetic Minority Over-sampling Technique), можуть допомогти розв'язати цю проблему [3].

Відсутність значень у деяких полях може ускладнити навчання моделей. Методи заповнення пропущених значень включають середнє значення, медіану або використання більш складних методів, як-от моделювання відсутніх значень.

Наявність помилкових або нерелевантних даних може знизити якість моделі. Техніки видалення викидів та фільтрації шуму можуть поліпшити якість даних.

Велика кількість ознак може призвести до надлишкового навчання і високих обчислювальних витрат. Методики зниження розмірності, такі як PCA та t-SNE, можуть допомогти впоратися з цією проблемою [4].

Висока кореляція між ознаками може створити проблеми для деяких моделей. Методи відбору ознак та регуляризація можуть допомогти розв'язати цю проблему.

Ефективність моделей машинного навчання значною мірою залежить від якості використовуваних наборів даних. Проблеми, такі як недостатність даних, дисбаланс класів, пропущені дані, шум у даних, висока розмірність даних та кореляція ознак, можуть негативно вплинути на процес навчання та точність прогнозування. Спеціальні методи та техніки машинного навчання можуть допомогти вирішити деякі з цих проблем. Наприклад, методи збільшення даних можуть допомогти вирішити проблему недостатності даних, техніки балансування класів можуть вирішити проблему дисбалансу класів, методи обробки пропущених даних та видалення шуму можуть поліпшити якість даних.

Список використаних джерел:

1. The MNIST database of handwritten digits. URL: http://yann.lecun.com/exdb/mnist/
2. Лобода Ю.Г. Особливості підготовки даних з метою подальшого аналізу. Інформаційне суспільство: проблеми та перспективи. VII Всеукр. наук.-практ. конф. (м. Одеса, 20 травня 2022 р.). Одеса, 2022. С. 32-34. URL: https://doi.org/10.32837/11300.17736
3. SMOTE. URL: http://surl.li/tyzkv
4. How t-SNE Outperforms PCA in Dimensionality Reduction. URL: http://surl.li/tyznf

Ключові слова: набори даних, машинне навчання, моделі, види, класифікація

Keywords: data sets, machine learning, models, types, classification

ОСОБЛИВОСТІ ВИВЧЕННЯ БАЗ ДАНИХ ДЛЯ ЗДОБУВАЧІВ ВИЩОЇ ОСВІТИ З ГАЛУЗІ ЗНАНЬ «ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ» В НУ «ОЮА»

Логінова Наталія

завідувач кафедри інформаційних технологій,
кандидат педагогічних наук, доцент
Національного університету «Одеська юридична академія»

Бази даних (БД) є основою практично будь-якого програмного продукту, оскільки вони зберігають та організують дані, необхідні для його роботи. Знання принципів проєктування та роботи з БД допомагає не тільки зрозуміти принципи побудови застосунків, але й оптимізувати їх продуктивність. Крім того, БД є невід'ємною частиною сучасних інформаційних систем та вебзастосунків. Поняття того, як працювати з даними, забезпечувати їх доступність та цілісність є основним навиком для майбутніх фахівці у ІТ-галузі.

Метою вивчення освітнього компоненту «Бази даних» є формування у майбутніх фахівців теоретичних знань та практичних навичок з ефективного використання сучасних систем керування базами даних (СКБД) у майбутній професійній діяльності. Знання в області програмних засобів для роботи з БД забезпечать майбутнім фахівцям високу конкурентоздатність на ринку праці й ефективне виконання своїх професійних обов'язків [1].

На факультеті кібербезпеки та інформаційних технологій НУ «ОЮА» за освітнім рівням «бакалавр» здійснюється навчання за трьома спеціальностями з 12 галузі знань «Інформаційні технології» – це 121 «Інженерія програмного забезпечення», 122 «Комп'ютерні науки» та 125 «Кібербезпека та захист інформації». Для цих спеціальностей освітній компонент «Бази даних» є обов'язковим, але відрізняється кількістю кредитів та змістом.

Для спеціальностей 121 і 122 [2, 3] зміст освітнього компоненту є однаковим та складається з наступних тем:

1. Теоретичні основи БД та системами керування БД.
2. Моделі даних.
3. Реляційні БД.
4. Теорія нормалізації реляційної моделі даних.
5. Проєктування БД.
6. Мова структурованих запитів SQL.
7. Технології адміністрування СКБД.

При вивченні визначених тем, здобувачі розглядають теоретичні основи БД та СКБД, вивчають основні поняття «база даних», «система керування базами даних», «інформаційна система». Знайомляться з класифікацію СКБД (за моделями, за типами збереженої інформації, за способом організації доступу до даних). Вивчають реляційні БД, які є найпоширеною формою зберігання даних у сучасному світі та забезпечують структуроване зберігання та управління інформацією, що дозволяє ефективно організовувати дані та забезпечує їх цілісність. Розглядають процеси розбиття БД на набори таблиць, що дозволяє уникнути дублювання даних та забезпечити їх консистентність. Нормалізація сприяє ефективності інформаційних систем та зменшує ймовірність помилок під час роботи з даними. Вивчають методи та принципи створення структур баз даних, розробку схем даних, оптимізацію взаємодії між даними, а також інші аспекти пов'язані з управлінням і обробкою інформації в інформаційних системах. Набувають практичних навиків із застосування мови структурованих запитів SQL, яка використовується для створення, модифікації та керування даними в реляційних БД. Вивчають різні методи, підходи та інструменти, які використовуються для ефективного управління СКБД – встановлення, конфігурування, моніторинг, оптимізація та забезпечення безпечної роботи БД.

Для спеціальності 125 «Кібербезпека та захист інформації» освітній компонент складається з наступних тем [4]:

1. Теоретичні основи БД та системами керування БД.
2. Моделювання даних.
3. Проєктування та управління БД.
4. Мова структурованих запитів SQL.

Зміст освітнього компонента «Бази даних» для майбутніх фахівців з кібербезпеки відрізняється від змісту для інших спеціальностей, оскільки вони повинні володіти термінологією баз даних, знаннями та навиками проектування логічної

структури баз даних засобами нормалізації відношень, розробки фізичної структури таблиць баз даних і зв'язків між ними засобами відповідного програмного забезпечення. Ці знання є підґрунтям для засвоєння знань при вивченні освітніх компонентів професійної та практичної підготовки, а саме «Безпека даних», «Безпека додатків та захист ПЗ» тощо.

Вивчення освітнього компоненту «Бази даних» є важливим для фахівців IT-галузі оскільки, знання БД дозволяють ефективно зберігати, організовувати та керувати великими обсягами інформації, що особливо важливо у сучасному світі, де дані стають все більш цінним ресурсом. Розуміння БД допомагають розробляти ефективні програмні продукти та застосунки, які використовуються для обробки та аналізу даних, що допомагає підвищити продуктивність та швидкість роботи розробників програмного забезпечення.

Для вдосконалення знань з БД та СКБД для кожної спеціальності в освітніх програмах пропонується вибірковий блок «Database Developer», якій складається з наступних вибіркових компонентів:

1. Хмарні технології.
2. Програмування PHP.
3. Фреймворки для Backend розробки.
4. Фреймворки для Frontend розробки.
5. Поглиблене вивчення БД.
6. Бази даних сімейства NoSQL.
7. Програмування баз даних.
8. DevOps.
9. Документо-орієнтовані бази даних.
10. Сучасні методології та технології розроблення ПЗ.
11. DevOpsSec.
12. Адміністрування та захист баз та сховищ даних.

Вивчення цих освітніх компонентів дозволить здобувачам вищої освіти поглибити та систематизувати свої знання щодо БД та СКБД. Фахівці, які вміють проєктувати, налаштовувати та підтримувати БД, дуже затребувані на ринку IT-праці.

Отже, вивчення обов'язкового компоненту «Бази даних» та низки вибіркових компонентів, які складають блок «Database Developer», важливо для здобувачів освіти зі спеціальностей, які входять до 12 галузі знань «Інформаційні технології», оскільки допомагає їм розуміти основні принципи та технології роботи з даними та БД. Наддасть можливість значно підвищить кваліфікацію

фахівців у ІТ-галузі та дозволить їм успішно працювати з інформацією в сучасному цифровому світі.

Список використаних джерел:

1. Організація баз даних : навч. посібник / О. Г. Трофименко, Ю. В. Прокоп, Н. І. Логінова, І. М. Копитчук. 2-ге вид. виправ. і доповн. Одеса : Фенікс, 2019. 246 с.
2. Силабус навчальної дисципліни «Бази даних» (спеціальність 121 «Інженерія програмного забезпечення»). URL: http://surl.li/ueiet
3. Силабус навчальної дисципліни «Бази даних» (спеціальність 122 «Комп'ютерні науки»). URL: http://surl.li/ueiep
4. Силабус навчальної дисципліни «Бази даних» (спеціальність 125 «Кібербезпека та захист інформації»). URL: http://surl.li/ueiha

Ключові слова: бази даних, системи керування базами даних, інформаційні технології, освітній компонент

Key words: databases, database management systems, information technologies, educational component

HTTP ТА WEBSCOKET ПРОТОКОЛИ ПРИ РОЗРОБЦІ ВЕБЗАСТОСУНКІВ ЗАСОБАМИ SPRING

Люлік Анастасія

студентка 3-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Комунікація між сервером і клієнтом є важливою складовою кожного вебзастосунку. Завдяки їй користувач може надсилати та отримувати дані, необхідні для роботи із сайтом. Так, у разі переходу за інтернет-посиланням на якийсь сайт на сервер надсилається запит з адресою відповідної вебсторінки, відбувається оброблення даного запиту та надсилається відповідь. Відповідь містить HTML-код сторінки з CSS та JavaScript, або ж дані JSON для REST-застосунків. Тим самим в результаті виконання запиту користувач побачить вміст вебсторінки сайту у його браузері.

Описана процедура формування та оброблення HTTP-запитів до сервера є дієвою для більшості сайтів у мережі Інтернет. Однак, її виконання потребує певного часу, до того ж запити та відповіді не зберігаються. Це призводить до затримок у виконанні й неефективності роботи застосунків (чатів, ігор тощо) у реальному часі. Проблема стає ще більш очевидною, якщо критично важливим для вебзастосунку є швидке оброблення запитів клієнта та надання швидких

відповідей, принаймні, без перезавантаження сторінки після кожного оновлення даних. Вирішити цю проблему може Websocket-протокол, оскільки він здатен забезпечити постійне з'єднання без потреби повторних з'єднань для кожного наступного запиту.

Доречність HTTP та Websocket протоколів залежить від специфічних вимог до вебзастосунку. Так, HTTP (Hypertext Transfer Protocol) є протоколом прикладного рівня для передачі гіпермедійних документів [1]. Цей протокол працює за чітким принципом запит-відповідь: запит містить метод, URL, заголовок (header) та тіло (body), а відповідь – стан, заголовок, тіло [2].

Метод у запиті визначає різновид виконуваної дії. Розрізняють такі методи: GET – запит на отримання даних, при цьому тіло такого запиту зазвичай порожнє/відсутнє; POST – передача даних у тілі повідомлення; PUT – оновлення даних з тіла повідомлення; DELETE – видалення даних.

Передача даних запиту через body використовується переважно для великих обсягів даних, наприклад, для об'єктів JSON. В свою чергу, невеликі за обсягом дані можна передавати одразу у параметрах посилання URL. До того ж, параметризованим посиланням можна поділитись, тоді як при передачі даних у тілі це неможливо зробити.

Для позначення результатів виконання запитів використовують відповідні статуси, тобто індикатори успішності виконання запиту. Залежно від номера статусу діагностується успішність формування відповіді та причини можливих відмов: 1xx – запит отриманий і обробляється, 2xx – успішне виконання, 3xx – перенаправлення / необхідні додаткові дії, 4xx – неможливо виконати через клієнтську помилку, 5xx – помилка сервера [3].

HTTP використовується для створення застосунків засобами Spring-MVC та Spring-REST. При розробці за архітектурним шаблоном Model-View-Controller застосовується такий механізм: Dispatcher Servlet (головний контролер) отримує HTTP-запит та делегує роботу методу контролера з потрібним мапуванням – відповідним URL, контролер обробляє запит та надсилає дані до View, користувач бачить сторінку з потрібними даними [4].

Контролери у Spring є об'єктами-компонентами з відповідною анотаціями. Їх створення, управління та зв'язки повністю керуються Spring-контейнером. View зазвичай є сторінкою html, jsp, thymeleaf чи mustache. Дані між поданням та контролером передаються завдяки моделі, що у Spring реалізована як клас, що містить дані ключ-значення. Модель заповнюється у контролері, а відкривається у View для демонстрації користувачу. Також модель може навпаки

заповнюватись на стороні клієнта, наприклад, через форми введення, та передаватись до контролера в тілі запиту. Крім того, передача можлива через параметри в URL.

REST має схожу реалізацію та добре працює разом з MVC. Основна різниця – методи контролера позначаються відповідно до методів HTTP: GET, POST, PUT, DELETE. Тобто, якщо декілька методів контролера мають однакове мапування «/example», то робота буде делегуватись методу контролера з аналогічним HTTP-методом. Обмін даними відбувається завдяки JSON-об'єктам, які автоматично серіалізуються та десеріалізуються.

WebSocket-протокол – двосторонній протокол, який використовується переважно в каналі зв'язку клієнт-сервер [5]. Для встановлення з'єднання надсилається HTTP-запит та отримується відповідь з успішним статусом. Даний процес називається рукостисканням, результат якого – залишене відкритим TCP-з'єднання. Подальша робота відбувається, завдяки обміну через протокол Websocket. З'єднання триває, доки одна зі сторін його не завершить.

Отже, на відміну від використання багатьох URL, які обробляються різними методами контролерів для HTTP, отримуємо єдиний URL. Ще одна важлива відмінність – структура та вміст повідомлень. Для Websocket необхідно при рукостисканні обумовлювати підпротокол, який буде визначати формат та семантику повідомлень. В якості такого підпротоколу у Spring використовується STOMP – текстовий протокол обміну повідомлень [6]. Основний механізм – підписка-публікація. Користувач підписується на певний топік, за яким надсилає та отримує повідомлення від інших підписаних клієнтів та сервера.

Для оброблення повідомлень у Spring-застосунку все ще використовуються контролери, однак їх методи мають інші анотації, де зазначається мапування та топік, підписникам якого буде надсилатись відповідь. Завдяки JavaScript встановлюється з'єднання з сервером через STOMP, оформлюються підписки, надсилаються дані до контролерів.

Послідовність роботи вебсайту з Websocket є такою: декілька користувачів заходять на сайт; надсилаються запити на з'єднання з підпротоколом STOMP; сервер приймає запит та створює з'єднання; користувачі підписується на тему; користувачі надсилають дані на сервер; сервер прослуховує канал з темою, отримує на нього повідомлення, надсилає оброблені дані; всі підписані на канал користувачі отримують оброблені дані; користувачі закривають сторінку; з'єднання завершено.

Попри потужність Websocket, його постійне використання не є оптимальним варіантом для більшості сайтів. Адже підтримка постійного з'єднання означає додаткове навантаження на сервер, що може призвести до великих втрат. А деякі середовища можуть блокувати або й взагалі не підтримувати Websocket.

Наразі обидва протоколи, і HTTP, і Websocket, використовуються для розробки різних типів вебзастосунків, але через свою специфіку не можуть замінювати один одного. Тим не менш, можливе використання одразу обох технологій, що є логічним виходом для створення застосунків, які мають як статичні дії (відправлення, отримання, редагування та видалення даних), так і дії реального часу (спілкування між користувачами, ігри, push-повідомлення тощо). Наприклад, у разі створення певного аналоги WhatsApp у вигляді сайту частина, що відповідає за вхід, налаштування профілю, пошук користувачів має використовувати протокол HTTP, а безпосередньо спілкування між клієнтами через чати – Websocket. Тоді, завдяки використанню обох технологій, з'являється можливість створити цікаві користувачу проєкти, які приваблюють своєю надійністю та інтерактивною складовою.

Список використаних джерел:

1. Mdn web docs. HTTP. URL: https://developer.mozilla.org/en-US/docs/Web/HTTP
2. Mdn web docs. HTTP Messages. URL: https://developer.mozilla.org/en-US/docs/Web/HTTP/Messages
3. HTTP Status Codes. URL: https://http-status-code.de/
4. Web MVC framework. URL: https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/mvc.html.
5. Wallarm. Websocket protocol. URL: https://www.wallarm.com/what/a-simple-explanation-of-what-a-websocket-is
6. Geeks for Geeks. STOMP protocol. URL: https://www.geeksforgeeks.org/stomp-protocol/

Ключові слова: протокол передачі гіпертексту, протокол Websocket, комунікація клієнт-сервер, протокол STOMP, розробка вебзастосунків

Keywords: Hypertext Transfer Protocol, WebSocket protocol, client-server communication, STOMP protocol, web application development

Науковий керівник: доцент О. Трофименко

АКТУАЛЬНІ ПАТЕРНИ ПРОЄКТУВАННЯ ДЛЯ РОЗРОБКИ ANDROID-ЗАСТОСУНКІВ

Манаков Сергій

кандидат технічних наук, доцент, доцент кафедри інформаційних технологій Національного університету «Одеська юридична академія»

Засоби мобільної розробки продовжують свій активний розвиток. Коли майбутній розробник починає вивчати одну з сучасних платформ та мов програмування для створення мобільних застосунків, він поступово усвідомлює, що всі інші аналоги якнайшвидше втілюють актуальні новації, які були визнані ефективними у професійному ком'юніті. Це приводить до уніфікації процесу розробки.

Якщо розглядати нативну розробку для платформи Android, то основними засобами являються IDE Android Studio та мова Kotlin [1], які офіційно підтримуються Google.

У сучасному програмуванні широко використовується об'єктно-орієнтована парадигма, яка призвела до появи цілої низки патернів (шаблонів) проєктування [2], які відображають найкращі практики організації коду.

Розглянемо основні шаблони проектування, які на думку автора має знати кожний розробник застосунків для Android.

1. Singleton (одинак). Цей шаблон обмежує створення класу одним об'єктом. Сам клас містить спосіб доступу до свого єдиного екземпляра. У Kotlin це можна зробити за допомогою декларації об'єкта, лише у один рядок коду. У Java немає вбудованого способу зробити це, але цього можна досягти за допомогою статичних посилань.

2. Factory (фабрика). Використовується, коли потрібно створювати об'єкти одного з декількох типів. Фабрика вирішує, який тип об'єкта створювати з урахуванням деякої логіки.

3. Builder (будівельник). Шаблон використовується для створення складних об'єктів поетапно. Шаблон допомагає зробити код більш читаним та простим в обслуговуванні. У Kotlin можна досягти аналогічного результату, використовуючи іменовані параметри та параметри за промовчанням у конструкторі.

4. Facade (фасад). Дозволяє приховати складність коду шляхом надання простого інтерфейсу до складної системи. Хорошим прикладом у розробці Android є інтерфейс API бібліотеки Retrofit, яка використовується для мережевих запитів. Retrofit приховує базові мережеві запити та дає розробнику простий спосіб отримувати дані із сервера.

5. Dependency Injection (ін’єкція залежностей): цей шаблон є способом надати об'єкту необхідні йому залежності. Це можна зробити за допомогою впровадження у конструктор, але у великих застосунках часто використовуються такі бібліотеки, як Dagger. Впровадження залежностей робить код більш модульним та простим для тестування.

6. Adapter (адаптер) використовується для адаптації одного інтерфейсу до іншого. У класичній Android-розробці найпоширенішим прикладом є адаптер RecyclerView. Він приймає список даних і адаптує його до елементів інтерфейсу користувача, які можна відображати на екрані пристрою.

У міру зростання програмного забезпечення важливо добре його структурувати, щоб його було легше розуміти, підтримувати та розширювати. Один із способів досягти цього – розділити обов'язки. У розробці програмного забезпечення поділ обов'язків означає поділ різних частин програмного забезпечення, що мають різні завдання. Для вирішення цих питань, доцільним виявляється застосування деяких архітектурних шаблонів проєктування.

У контексті Android, архітектурний шаблон MVVM [3] поділяє застосунок на три частини: Model, View та ViewModel:

Model є джерелом даних. Це можуть бути дані з бази даних, віддаленого API або того й іншого.

ViewModel містить бізнес-логіку. Бізнес-логіка – це код, що реалізує основні функції програми, такі як фільтрація даних або перевірка введення користувача. ViewModel отримує дані з Model та готує їх для представлення.

View – це видима частина програми. Він містить елементи інтерфейсу користувача і оновлює їх на основі даних, отриманих від моделі подання. View не повинно містити жодної бізнес-логіки.

MVVM – це потужний шаблон, який робить Android-застосунки більш масштабованими, зрозумілими та тестованими.

В свою чергу, архітектурний шаблон MVI [4] схожий на MVVM, оскільки обидва вони стосуються того, як View взаємодіє з ViewModel. Однак, деякі ключові відмінності полягають у тому, як відбувається ця взаємодія та які дані передаються.

MVI слідує принципу єдиного джерела, що означає, що він має одну точку входу та одну точку виходу. Точка входу – це метод Send, а точка виходу – це об'єкт стану. На відміну від MVVM, де ViewModel може мати кілька публічних методів, MVI має лише один метод Send. Цей метод може приймати різні події як аргументи. Залежно від типу події виконуються різні дії.

Об'єкт стану містить всю інформацію, необхідну для роботи поточного екрану застосунку. Це включає як поточну, так і минулу інформацію. View підписується на об'єкт стану, щоб отримувати оновлення.

Рекомендується реалізувати додаткові функції, такі як обробка натискання кнопки повернення для переходу до попереднього стану та використання списку станів.

Як висновок, можна заначити, що зростання складності мобільних застосунків підкреслює важливість використання ефективних інструментів та практик розробки, таких як розглянуті патерни проєктування та архітектурні шаблони.

Список використаних джерел:

1. Манаков С. Ю., Дика А. І. Вибір сучасних засобів розробки мобільних застосунків. Інформаційне суспільство: проблеми та перспективи : матеріали VIII Всеукр. наук.-практич. конф. (м-. Одеса, 2 черв. 2023 р.) / відп. ред. Н. І. Логінова. - Одеса, 2023. - С.86-89. - Режим доступу: https://doi.org/10.32837/11300.25263
2. Martin Fowler. Patterns of Enterprise Application Architecture. Addison-Wesley Professional, 2002. 560 p. ISBN: 978-0321127426.
3. ViewModel overview. Android Developers Guides.
4. URL: https://developer.android.com/topic/libraries/architecture/viewmodel
5. Rim Gazzah. MVI Architecture with Android.
6. URL: https://medium.com/swlh/mvi-architecture-with-android-fcde123e3c4a

Ключові слова: Android, патерни проєктування, архітектурні шаблони, MVVM, MVI, ін'єкція залежностей

Keywords: Android, design patterns, architectural patterns, MVVM, MVI, dependency injection

СПЕЦИФІКА ВЕБРОЗРОБКИ ОНЛАЙН-МАРКЕТПЛЕЙСУ

Мустафаєв Тимур

студент 4-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

На сьогодні комерційні компанії масово розвивають надання своїх товарів та послуг через вебпростір. Сучасний ринок повсюдно переходить в онлайн, адже інтернет-магазини дозволяють підприємцям знизити витрати на оренду приміщень, сплату комунальних послуг, оплату зарплатні продавців та іншого персоналу, що своєю чергою впливає на зниження ціни на товар. Нижчі ціни, порівняно з «магазинними», приваблюють більше людей на онлайн-шопінг. Крім того, швидкість та зручність обслуговування є суттєвою перевагою для

вибору клієнтом інтернет-обслуговування. Розвиток інтернет-комерції призводить до залучення нових ринків у вебпростір.

При цьому не кожен малий підприємець може собі дозволити розробку власного сайту для інтернет-магазину. Малий бізнес потребує економного способу торгівлі в мережі інтернет. Онлайн-маркетплейси є програмними платформами, які об'єднують продавців і покупців. Вони надають можливість користувачам шукати, переглядати й купувати товари різних брендів і категорій [1].

Попри те, що запуск нового ринку вимагає зусиль й інвестицій, це дуже прибуткова ніша. Інтернет-ринки настільки подобаються продавцям і покупцям, що майбутній постачальник програмного забезпечення гарантовано матимете багато зацікавлених користувачів. Адже цифрова трансформація триває, і ринок онлайн-комерції продовжуватиме зростати. Тому, навіть у висококонкурентному середовищі нові гравці завжди можуть знайти способи увійти та заповнити нові ніші, щоб створити базу лояльних клієнтів. За даними досліджень [1] 47% цифрових покупок у всьому світі відбуваються через платформи онлайн-ринків і лише 6% покупців B2B зараз не користуються онлайн-ринками. За прогнозами, протягом наступних 5 років 75% витрат на закупівлі B2B відбуватимуться через онлайн-ринки. Наразі 53% роздрібних торговців використовують інструменти, які допомагають їм продавати через різні канали (Shopify).

Бізнес-модель ринку – це тип бізнес-моделі, де платформа з’єднує покупців і продавців, полегшуючи операції між ними. Платформа зазвичай надає інструменти та послуги для безперебійних і безпечних транзакцій і отримує від цього дохід [2]. Створення торгової платформи надає переваги для всіх сторін: власника платформи, продавців і покупців.

Можливі переваги для власника онлайн-маркетплейсу [3, 4]:

– зручні моделі монетизації: власники комерційних майданчиків можуть вибирати різні гнучкі моделі доходів: комісії, реєстраційний збір, платна підписка, внесок за реєстрацію лістингу товарів, фіксована плата, що стягується з продавця за кожну транзакцію, чи то поєднання кількох схем;

– масштабований бізнес, доступ до глобальної ринкової ніші з мільярдами споживачів: оскільки ринки розширюються, додавання нових продавців відповідає зростанню попиту, зберігаючи баланс між попитом і пропозицією для сталого росту;

– додатковий дохід від реклами;

– інтерес користувачів та інвесторів: широкий спектр пропозицій привертає інтерес користувачів, а ринкові інвестиції привертають значну увагу інвесторів.

Переваги використання комерційної платформи для продавців [3]:

– постійний потік покупців;

– не потрібно перейматися питаннями самореклами: продавці не потребують незалежного просування, заощаджуючи на розробці, підтримці та витратах на SEO;

– покращений пошуковий рейтинг: списки товарів маркетплейсів незмінно мають високі позиції в пошукових результатах, забезпечуючи кращу видимість, ніж окремі онлайн-магазини;

– певна економія коштів: не потрібно інвестувати в розробку інтернет-магазину на замовлення, немає потреби мати технічну команду для обслуговування сайту;

– розширені можливості просування власного торгового бренду: продавці отримують вигоду від збільшення пізнаванності бренду на торгових платформах, що призводить до більшого потоку клієнтів;

– доступ до аналітики: продавці отримують уявлення про перегляди продуктів, покупки та іншу статистику через свої облікові записи на торговому майданчику.

Перевагами для покупців онлайн-маркетплейсів є [1, 2]:

– широкий асортимент продукції: інтернет-ринки надають покупцям доступ до широкого спектра продуктів в одному місці.

– зручні варіанти оплати та доставки;

– централізований інтерфейс: система перегляду широкого вибору брендів на одній платформі;

– надійні платформи: рейтинги продавців та товарів, наявні відгуки підвищують довіру покупців до ринків, порівняно з окремими фірмами. Крім того, передбачена надійна політика відшкодування;

– акції та зручність: ринкові майданчики часто пропонують знижки, залучаючи й утримуючи користувачів, а процес бронювання в один клік підвищує зручність.

Інтернет-ринки протягом останніх років стрімко зростають у доходах і кількості користувачів. Так, до 2024 року продажі B2B можуть сягнути 3,6 трильйонів доларів , а сегмент B2C – 3,5 трильйонів доларів на рік [4].

Щоб створити успішну платформу, зручну для користувачів, важливо врахувати багато різних аспектів роботи онлайн-маркетплейсу, а саме [2-4]:

– велика кількість продавців: завдяки онлайн-майданчикам безліч продавців мають унікальну можливість просувати свої товари широкій аудиторії. Це можуть бути як великі бренди, так і незалежні дизайнери та дрібні підприємці. У кожного продавця є власний профіль і магазин на торговому майданчику. Це дає їм змогу виставляти товари на продаж, встановлювати ціни та безпосередньо спілкуватися з покупцями;

– керування товарами: можливість зручного додавання і редагування товарів, управління запасами і рекламними акціями;

– керування замовленнями: на торговому майданчику етапи процесу купівлі охоплюють процеси вибору товару, оформлення замовлення, відстеження доставки та оформлення можливого повернення. Продавці повинні мати можливість керувати замовленнями, оновлювати статуси і спілкуватися з покупцями в зручний для них спосіб, а покупці – бачити оновлення щодо замовлень у режимі реального часу;

– відгуки та рейтинги на торгових майданчиках забезпечують прозорість і довіру. Покупці можуть залишати відгуки про продавців і товари, ґрунтуючись на різних критеріях. Це дає змогу продавцям покращувати свої товари та послуги, а новим покупцям – робити більш усвідомлений вибір.

– асортимент продукції: широкий вибір товарів – ключова перевага платформи маркетплейсу. Покупці можуть знайти різноманітні види одягу, взуття та аксесуарів – від класики до розкоші. Це робить онлайн-майданчик для продажу товарів зручним місцем для покупок для найрізноманітніших покупців.

Окремо слід зазначити важливий аспект розвитку, який стосується не тільки онлайн-маркетів, а й розробки будь-яких масово-орієнтованих застосунків. До цих аспектів належать [5]:

– зручний інтерфейс користувача (UI): привабливий і зручний дизайн, що дає змогу покупцям легко знаходити товари та інформацію, які їх цікавлять;

– користувацький досвід (UX): навігація сайтом має бути інтуїтивно зрозумілою, а процес вибору і купівлі товару – максимально простим і швидким;

– масштабованість: платформа має бути здатна впоратися зі збільшенням кількості користувачів і продуктів без шкоди для продуктивності;

– безпека: захист даних користувачів і транзакцій, використання шифрування і дотримання нормативних вимог;

– пошукова оптимізація та маркетинг: оптимізація для пошукових систем і використання маркетингових інструментів для залучення нових користувачів;

– аналітика: збір і аналіз даних про поведінку користувачів для поліпшення послуг і продуктів;

– підтримка клієнтів: системи зворотного зв'язку, чати підтримки та інші інструменти для спілкування з клієнтами.

Якщо ці аспекти враховуються при розробці платформи для онлайн-маркетплейсу, шанси на успішний запуск проєкту, який знайде відгук в аудиторії, набагато вищі. Це пов'язано з тим, що торговельний майданчик не тільки поєднує великі бренди, малі та середні підприємства, а й надає зручний інтерфейс як для покупців, так і для продавців.

Зручність використання – основний фактор, що впливає на інтерес користувачів до програмних продуктів. Це один із найважливіших аспектів, на які варто звернути увагу під час розробки. Призначений для користувача інтерфейс має бути зручним та інтуїтивно зрозумілим, щоб покупці не відволікалися на пошук кнопок або конкретних розділів онлайн-майданчика. Вони мають бути зосереджені лише на виборі товарів та їхніх характеристиках.

Список використаних джерел:

1. How to Build a Marketplace Website: The Definitive Guide. URL: https://smarttek.solutions/blog/how-to-build-a-marketplace/
2. Bashir A. Marketplace Business Model. URL: https://www.linkedin.com/pulse/marketplace-business-model-bashir-ahmed
3. How to Build a Marketplace Website in 15 Steps: Ultimate Guide. URL: https://www.codica.com/blog/how-to-build-online-marketplace/
4. Paratsii T. How to build an online marketplace: steps, features, and cost. URL: https://leobit.com/blog/how-to-build-an-online-marketplace-steps-features-and-cost/
5. О.Трофименко, Пастернак Ю.Ю., Манаков С.Ю., Лобода Ю.Г. Автоматизація тестування вебсайтів електронної комерції. Сучасна спеціальна техніка. 2021. № 2(65). С. 46-59. DOI: https://doi.org/10.36486/mst2411–3816.2021.2(65).5.

Ключові слова: онлайн-маркетплейс, інтернет-магазин, веброзробка
Keywords: online marketplace, Internet-shop, web development

Науковий керівник: доцент О. Трофименко

ІНСТРУМЕНТИ ДЛЯ ВЕБ-АНАЛІТИКИ ТА АНАЛІЗУ ДАНИХ

Орел Анастасія

студентка 1-го курсу магістратури факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Розуміння клієнта та забезпечення надання якісних послуг є ключем до успішного управління бізнесом в сучасному світі, де конкуренція постійно зростає, а умови на ринку постійно змінюються. Щоб бути спроможнім конкурувати в таких умовах, підприємствам потрібно мати високий рівень експертизи в своїй галузі та швидко реагувати на зміни в потребах та уподобаннях клієнтів. Це змушує замислитись про шляхи збору необхідних для просування бізнесу даних, і в цьому контексті веб-аналітика виступає одним з найважливіших інструментів.

В наш час значна кількість компаній, а також малих підприємств, володіють веб-сайтами, які виконують подвійну функцію. З одного боку, вони слугують візитною карточкою підприємства, де клієнти можуть отримати інформацію про продукти та послуги. З іншого боку, ці веб-ресурси виступають онлайн-платформами для просування товарів та послуг, а також здійснення продажів. Ці веб-ресурси стають важливим каналом комунікації з клієнтами, що, окрім цього, є цінним джерелом даних про їх потреби та уподобання, оскільки вони можуть збирати інформацію про їх інтереси та попередні дії.

Веб-аналітика надає підприємствам цінні дані про взаємодію користувачів з їхніми сайтами. Зокрема, вона надає змогу вимірювати трафік, аналізувати поведінку відвідувачів, відстежувати конверсії та інші метрики ефективності. Ці дані не лише допомагають розуміти, як клієнти знаходять продукти або послуги підприємства, але й визначають, чи успішно вони проходять усі етапи, та чи доходять до фінального етапу придбання. Якщо користувач не виконав ключову дію, такої як оформлення замовлення чи заповнення контактної форми, за допомогою веб-аналітики є можливість встановити причини, що саме могло зупинити його та змусити залишити сайт. В подальшому, це відкриває підприємствам можливості вдосконалення свого веб-сайту, та за необхідності, змінити маркетингову стратегію, щоб залучити та утримати більшу кількість клієнтів [1, 2].

Зокрема, веб-аналітика є важливим інструментом, що допомагає підприємствам адаптуватися до постійно мінливих умов ринку. Наприклад, шляхом аналізу даних про джерела трафіку та поведінку користувачів, компанії можуть зрозуміти які маркетингові канали є ефективними і які слід оптимізувати, а які варто припинити використовувати. Крім того веб-аналітика дозволяє виявити та вирішити проблеми на сайті, такі як помилки на сторінках, неоптимальні

елементи інтерфейсу або повільне завантаження. Виправлення таких дрібних проблем може значно покращити користувальницький досвід та сприяти збільшенню утримання користувачів на сайті, що, в свою чергу, може позитивно вплинути на конверсію та загальну ефективність веб-ресурсу [3].

Для проведення аналізу веб-сайтів існують різноманітні інструменти, кожен з яких має свої переваги та недоліки. Деякі з них підходять для аналізу невеликих ресурсів, а інші надають великий спектр можливостей, що дозволяє проводити більш глибокий та комплексний аналіз великих веб-сайтів.

Вибір конкретного інструменту залежить від потреб підприємства, обсягу даних, бюджету та рівня експертизи користувача. Одним з найпопулярніших інструментів для веб-аналітики є Google Analytics 4, який надає широкий спектр функцій для аналізу трафіку та поведінки користувачів. Він безкоштовний, та має простий інтерфейс, що робить доступним його для користувачів на будь-якому рівні експертизи. Однак, у порівнянні з деякими іншими інструментами аналітики, він може бути обмежений у функціональності для складних аналітичних завдань.

Як і усі інші інструменти аналітики, він дозволяє отримати детальну інформацію про трафік та поведінку користувачів. Також, цей інструмент використовує штучний інтелект та машинне навчання, що дозволяє проводити точні аналізи продуктивності сайту та прогнозувати поведінку користувача у майбутньому. Окрім того, Google Analytics 4 може збирати дані не лише з веб-сайтів компанії, але й із застосунків на iOS та Android, що значно розширює діапазон охоплень даних. Це дозволяє отримати більш повний та об'єктивний аналіз взаємодії користувачів з продуктами або послугами компанії на різних платформах [4].

Однією з ключових особливосте Google Analytics 4 є його підхід до конфіденційності даних користувачів. Наприклад, він не збирає особисту інформацію, таку як IP-адреси чи файли cookie. Замість цього інструмент використовує анонімізовані ідентифікатори, що дозволяють відстежувати трафік та взаємодію користувачів з сайтом без зберігання особистих даних. Дані що він маже збирати, лишаються в його системі до 14 місяців, після чого вони автоматично видаляються. Це дозволяє забезпечити більшу приватність та захист персональних даних користувачів [5].

Ще одним не менш відомим інструментом веб-аналітики є Adobe Analytics. Його головною особливістю є можливість отримувати статистичні дані в реальному часі та створювати цілісне уявлення про клієнта. Подібно до Google

Analytics 4, він має власний штучний інтелект та машинне навчання, які допомагають аналізувати потік інформації, знаходити нові можливості та методи збільшення конверсії. За допомогою штучного інтелекту та машинного навчання Adobe Analytics здатен проводити прогностичну аналітику, що дає можливість передбачити можливі загрози чи можливості на ринку. За допомогою цих інструментів, він аналізує великі обсяги даних, щоб виявити тенденції та зробити прогнози, спираючись на поведінку користувачів та інші дані. Такий підхід дозволяє компаніям оперативно реагувати на зміни на ринку та вдосконалювати свої стратегії маркетингу та продажів [6].

Серед інструментів для аналітики сайтів також є платформа Crazy Egg, що надає широкий спектр інструментів для аналізу сторінок, таких як теплові карти. До них входять карта кліків, карта скролінгу та карта активності мишки.

Карта кліків візуалізує дані про те, як користувач взаємодіяв з веб-сторінкою шляхом натискання на різні елементи, такі як посилання, кнопки, зображення і т.п. За допомогою цієї карти з'являється можливість переглянути, що найбільше привертало увагу користувача, а які фрагменти інтерфейсу були проігноровані, що може свідчити про потребу у їх вдосконаленні. Це важливий інструмент для оптимізації розміщення елементів на сторінці та вдосконалення їх ефективності. Крім того, карта кліків допомагає виявити можливі проблеми з навігацією або недоліки розташування деяких елементів, що допоможе підвищити користувальницький досвід.

Карта скролінгу збирає інформацію про те, на скільки користувачі, що переглядали сторінку, опустилися вниз. Зібрана інформація за допомогою такої карти дуже корисна для сайтів, вміст який є суто інформаційний. Вона дозволяє виявити, на якому розділі сторінки більшість користувачів зупиняється або прогортає далі, що допомагає зрозуміти, який контент є найбільш привабливим для користувача. За її допомоги можна визначити ефективність розташування інформаційних блоків на сторінці [7]. Такий аналіз допомагає вдосконалити структуру сторінки та покращити взаємодію з нею користувачів.

Карта активності миші чимось схожа на карту кликів, проте на відміну від другої, вона аналізує не самі кліки, а місця, куди найчастіше наводив користувач мишку. Цей вид аналізу особливо корисний для сайтів, де графічні компоненти не потребують повної взаємодії користувача, але виконують певні функції при наведенні. Це можуть бути такі елементи як інформативні блоки, що відкриваються після наведення на них мишки, анімації, приближення фото, тощо. Аналіз карти активності миші допомагає розуміти, на які елементи або ділянки сторінки

користувачі найчастіше звертають увагу, навіть якщо вони не виконують жодних конкретних дій. Це дозволяє виявити зацікавленість користувачів та визначити, чи потрібен такий функціонал, чи його варто модифікувати [8].

Вибір інструменту для веб-аналітики є важливим етапом для будь-якого веб-ресурсу, оскільки від нього залежить ефективність аналізу та прийняття стратегічних рішень. При виборі варто ретельно враховувати тип сайту, обсяг трафіку, доступний бюджет, та функціональні потреби.

Наприклад, для невеликих компаній з обмеженим бюджетом та основною потребою у веб-аналітиці гарним рішенням може стати Google Analytics 4, оскільки він безкоштовний та має простий інтерфейс. Для великих підприємств з високим обсягом трафіку та потребами у складній аналітиці може підходити Adobe Analytics, що надає більш розширені можливості.

Також важливо враховувати специфіку веб-ресурсу та потреби в аналітиці користувачів. Наприклад для сайтів з великою кількістю графічних елементів і контенту, може стати корисним використання платформи Crazy Egg, що спеціалізується на аналізі сторінок за допомогою теплових карт та інших інструментів.

У будь-якому випадку, вибір інструменту повинен враховувати специфіки бізнесу та мету аналізу, забезпечуючи найбільш оптимальний інструментарій для збору та аналізу даних, необхідних для прийняття вірних стратегічних рішень.

Список використаних джерел:

1. Аналіз веб трафіку: як зрозуміти відвідувачів вашого сайту. URL: https://www.04597.com.ua/list/444651
2. Complete Guide to Web Analytics. URL: https://netpeak.net/uk/blog/povniy-gayd-po-vebanalititsi/
3. Web analytics. URL: http://surl.li/tteiw
4. Що таке веб аналітика – повний гайд. URL: https://www.whitepress.com/ua/baza-znan/103/google-analytics-dlia-pochatkivtsiv
5. Google Analytics 4: можливості, переваги та особливості. URL: https://horoshop.ua/ua/blog/google-analytics-4/
6. Adobe Web Analytics & Adobe for Business. URL: https://business.adobe.com/ua/products/analytics/web-analytics.html
7. Web-аналітика: що це, як її використовувати, основні інструменти веб-аналітик? URL: http://surl.li/tquve
8. Що може розказати теплова карта сайту та як її створити. URL: http://surl.li/tquwb

Ключові слова: веб-аналітика, аналіз даних, веб-сайти, Google Analytics 4, Adobe Analytics, Crazy Egg
Keywords: web analytics, data analysis, websites, Google Analytics 4, Adobe Analytics, Crazy Egg

Науковий керівник: доцент Ю. Лобода

СТВОРЕННЯ КРОСПЛАТФОРМЕНИХ GUI - ЗАСТОСУНКІВ З ВИКОРИСТАННЯМ WXPYTHON ТА MYSQL

Сейрік Юлія

студентка 4-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

У сучасному світі з досить високим рівнем розвитку інформаційних технологій широким попитом користуються застосунки, які можуть працювати на багатьох різних платформах та операційних системах. Їхня актуальність пояснюється тим, що такі додатки охоплюють більшу аудиторію клієнтів порівняно з тими що є зосередженими лише на одній платформі. Це надає можливість більшій кількості людей почати взаємодіяти з ними.

Крім того, є й інші причини популярності кросплатформних програмних продуктів. Однією з них є те, що розробка та підтримка таких застосунків відбувається значно простіше, оскільки відсутня необхідність створювати окремий код для кожної платформи. Це також дозволяє знизити витрати часу та ресурсів що є без сумнівів великою перевагою.

Кросплатформені GUI - застосунки – це програмні продукти, які мають графічний інтерфейс користувача, і можуть працювати на різних платформах без потреби у створенні окремого коду для кожної з них. Одним з інструментів, який можна використовувати для їх розробки – бібліотека wxPython, яка належить мові програмування Python.

wxPython - це кросплатформний інструментарій графічного інтерфейсу для мови програмування Python. Він дозволяє програмістам Python легко і просто створювати програми з надійним, функціональним графічним інтерфейсом користувача. Він реалізований у вигляді набору модулів розширення Python, які обгортають компоненти графічного інтерфейсу популярної кросплатформної бібліотеки wxWidgets, написаної на C++ [1].

Наразі підтримуваними платформами є Microsoft Windows, Mac OS X та macOS, а також Linux або інші unix - подібні системи з бібліотеками GTK2 або GTK3 [1].

Ця бібліотека має перевагу у виборі не лише завдяки можливості використання програм, що розроблені з її допомогою, на різних операційних системах, а й через широкий вибір різних інструментів, для створення будь-якого дизайну інтерфейсу. Крім того, вона не складна у використанні і має досить зрозумілу і докладну документацію.

Більшість GUI - застосунків використовують клієнт-серверну модель архітектури, яка складається з клієнтської частини (інтерфейсу користувача) та серверної частини, що забезпечує взаємодію між клієнтом та базою даних через запити до неї.

Однією з баз даних, яка може бути вигідним вибором для використання у розробці кросплатформених GUI – застосунків, є реляційна. В першу чергу реляційна база даних має підтримку СУБД на різних платформах. Окрім того вона може працювати за об'єктно-орієнтованою моделлю, що особливо актуально при застосуванні мови програмування Python. Також вона використовує запити SQL, для роботи з даними, що робить її незалежною від платформи, а також дозволяє використовувати однин і той самий запит багато разів у коді.

Мова структурованих запитів (скор. "SQL" від англ. "Structured Query Language") - це стандартна мова запитів, яка використовується для роботи з реляційними базами даних. SQL використовується для: створення баз даних, створення таблиць в базі даних, читання даних з таблиць, вставки даних в таблиці, оновлення даних в таблиці, видалення даних з таблиці, видалення таблиць бази даних, видалення баз даних, та інших операцій [2].

MySQL – це система управління реляційними базами даних. Окрім багатоплатформеності, серед її переваг також можна визначити те що вона є простою у використанні, має високу продуктивність, а також є стабільною та надійною.

Для створення серверної частини, яка займається з'єднанням інтерфейсу користувача та бази даних можна використати драйвер MySQL Connector/Python.

MySQL Connector/Python дозволяє програмам на Python отримувати доступ до баз даних MySQL за допомогою API, що відповідає специфікації Python Database API Specification v2.0 [3].

MySQL Connector/Python включає підтримку:

– Майже всіх функцій, що надаються сервером MySQL версії 8.0 і вище.

– X DevAPI.

– Перетворення значень параметрів між типами даних Python і MySQL, наприклад, Python datetime і MySQL DATETIME.

– Всіх розширень MySQL до стандартного синтаксису SQL.

– Стиснення протоколу, що дозволяє стискати потік даних між клієнтом і сервером.

– Підключення за допомогою TCP/IP-сокетів, а в Unix - за допомогою Unix-сокетів.

– Безпечних TCP/IP-з'єднань за допомогою SSL.

– Самостійний драйвер. Connector/Python не потребує клієнтської бібліотеки MySQL або будь-яких модулів Python за межами стандартної бібліотеки [3].

Інструменти для створення багатоплатформених додатків, такі як база даних MySQL та драйвер MySQL Connector/Python мають досить велику та докладну документацію, яка роз'яснює всі аспекти цих засобів у доступній формі.

Кросплатформені GUI – застосунки розроблені з використанням бібліотеки wxPython, бази даних MySQL та драйверу MySQL Connector/Python мають численні вагомі переваги завдяки яким користувачі можуть легко застосовувати подібні програмні продукти.

Список використаних джерел

1. Overview of wxPython. wxPython. URL: http://surl.li/ttdwj
2. Що таке SQL та Бази даних? Acode. URL: https://acode.com.ua/sql-intro/
3. Chapter 1 Introduction to MySQL Connector/Python. MySQL Connectors Documentation. URL: http://surl.li/ttdvb

Ключові слова: платформа, застосунок, GUI, wxPython, MySQL, драйвер
Keywords: platform, application, GUI, wxPython, MySQL, driver

Науковий керівник: доцент Ю. Лобода

РОЗРОБКА ОПТИМІЗУВАЛЬНОГО КОМПІЛЯТОРА ДЛЯ МОВИ C++

Сидоренко Олег

студент 4-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Основним завданням програмування є створення правильних, а не ефективних програм. Правильну, але не ефективну програму можна оптимізувати та зробити ефективною. У світі існує безліч програмного забезпечення, написаного на мові C++. Вважається, що ця мова є дуже швидкою, але існують сфери, що вимагають від програми високої продуктивності, яка недосяжна лише вдалим вибором мови. Для великих програм ще на стадії проєктування визначаються параметри, що включають середній час їх виконання, потрібний обсяг пам'яті та розмір файлу.

Актуальність дослідження обумовлена тим фактом, що при розробці високопродуктивного коду, існує потреба у застосуванні додаткової оптимізації вихідного коду. З цим завданням досить добре справляється компілятор, використовуючи оптимізаційні прапори компіляції [1], при чому часто досягається результат, що задовольняє користувача, проте далеко не завжди цього достатньо.

Метою дослідження є програмна реалізацію процесу оптимізації часу виконання програм на мові C++ засобами регульованого агресивного вбудовування функцій.

Для досягнення мети було сформульовано такі завдання:

– провести огляд предметної галузі та підходів;

– протестувати розроблений статичний аналізатор для визначення ефективності знаходження помилок та вразливостей у різних програмах;

– отримати покращення затримки прикладного проєкту у порівнянні з компіляцією без використання розробленої системи в однакових умовах.

Відомо, що компілятор розглядає вбудовані параметри розширення та ключові слова як пропозиції. Однією з найважливіших оптимізації компілятора є вбудовування функцій – заміна виклику функції безпосередньо тілом функції [5].

Процес вбудовування функції – це оптимізація компілятора, що спрямована на заміну виклику функції безпосередньо тілом функції. Ця дія може покращити швидкість роботи програми, оскільки після вбудовування відсутні накладні витрати на виклик функції. Вбудовування функцій у мові C++ дозволяє оптимізувати роботу компілятора, зменшуючи витрати на виклики функцій та покращуючи швидкодію програми. У середовищі Visual Studio це може бути досягнуто за допомогою ключового слова _forceinline, яке підказує компілятору вбудувати вміст функції безпосередньо в місце виклику, не створюючи окремого виклику функції.

LLVM – це програмна система, що є набором інструментів для розробки компіляторів, асемблерів та інших систем, пов'язаних з обробкою програмного коду [2]. LLVM надає універсальний, низькорівневий та модульний інтерфейс для компіляції програм, який може бути використаний для розробки компіляторів для різних мов програмування.

Процес компіляції в LLVM включає кілька етапів. Під час першого етапу, вихідний код програми перетворюється на проміжне уявлення (intermediate representation) LLVM IR за допомогою компілятора Clang [3] за допомогою виклику програми clang++ або іншого сумісного компілятора. Потім IR

передається оптимізатор LLVM, який застосовує різні оптимізації, такі як вбудовування функцій, видалення недосяжного коду та інші, для поліпшення продуктивності та якості згенерованого коду.

У компіляторі на основі LLVM інтерфейс відповідає за розбір, перевірку та діагностику помилок у вхідному коді, а потім переклад проаналізованого коду в LLVM IR. Це дуже проста реалізація трифазної конструкції (рис. 1), але цей простий опис приховує частину потужності та гнучкості, які архітектура LLVM отримує від LLVM IR.

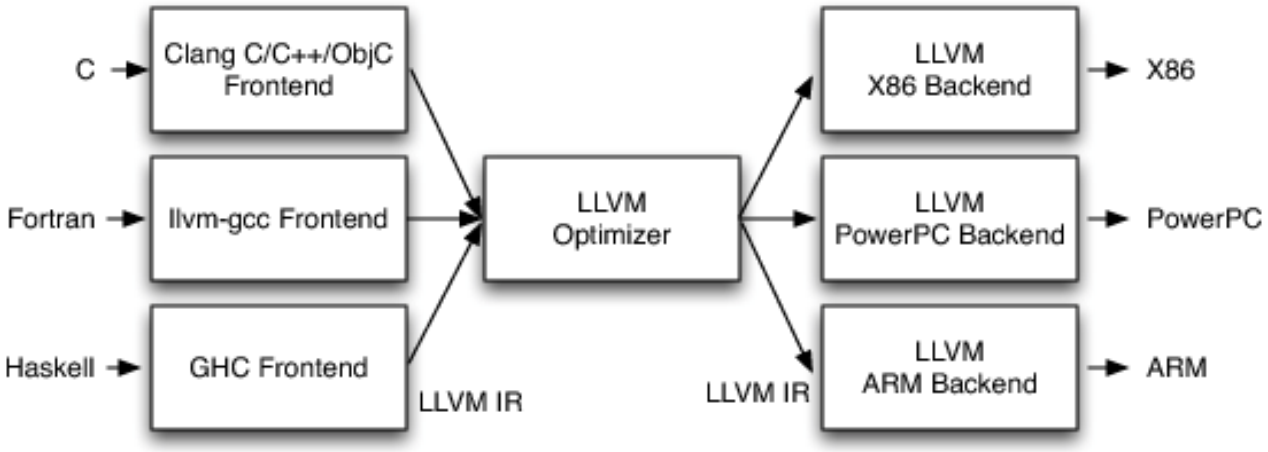


Рисунок 1 – Реалізація трифазної конструкції LLVM

Починаючи з версії 2019, MS Visual Studio включає підтримку редагування, створення та налагодження за допомогою Clang/LLVM у проєктах CMake [4], націлених на Windows. Після налаштування конфігурації Clang, розробник може створювати та налагодити проєкт. Середовище Visual Studio самостійно виявляє компілятор Clang, і надає можливості IntelliSense для підсвічування, навігації та інших функції редагування.

Далі оптимізоване проміжне уявлення IR перетворюється на машинний код цільової архітектури, використовуючи генератор коду LLVM. При генерації машинного коду LLVM використовує безліч оптимізації та техніки генерації коду для створення швидкого та ефективного виконуваного файлу.

Крос-платформовий відкритий генератор сценаріїв складання CMake є стандартом де-факто для створення коду на мові C++. Це потужне комплексне рішення для керування процесом створення програмного забезпечення. CMake дозволяє розробникам описувати, як створювати прості та дуже складні програмні системи за допомогою єдиного набору вхідних файлів. Систему можна використовувати для створення застосунків на багатьох платформах, від Android до iOS і високопродуктивних систем.

Програмна реалізація оптимізувального компілятору буде базуватися на мовах C++ і Python, наборі компіляторів LLVM, генераторі сценаріїв складання CMake.

Виконавцем команд оптимізувального компілятора є додаток run_compile_commands, який запускає команди компіляції в порядку, заданому у файлі з командами компіляції, паралельно виводячи на екран поточну команду, що виконується. Виконавець приймає єдиний аргумент – шлях до файлу з командами компіляції. Реалізація цієї компоненти зводиться до аналізу JSON-файлу та послідовного застосування команд, розміщених у файлі.

Під час виконання дослідження реалізовано програмну систему, користуватися якою можна, якщо виконати наступні кроки:

1. зібрати модифіковану версію llvm (це необхідно зробити один раз);
2. форсувати генерацію compile_commands.json;
3. викликати команду cmake;
4. модифікувати файл compile_commands.json;
5. запустити команди з модифікованого compile_commands.json;
6. викликати команду make.

Для вимірювання середнього часу роботи програми були написані додаткові програми, що симулюють навантаження на проєкти, що тестуються. Також у випадку декартового дерева необхідно було зафіксувати ініціалізацію генератора випадкових чисел, тому що в іншому випадку вимірювання проводилися б на деревах різної структури.

Віртуальний сервер дозволяє створити ізольоване середовище, яке може бути налаштоване для забезпечення однакових умов тестування, максимальної продуктивності та стійкості до зовнішніх впливів. Для створення віртуального серверу обрано хостинг Google Compute Engine.

Для тестування та вимірювання затримок було вибрано три проєкти, що розташовані у вільному доступі: CodeGuida для парсингу XML-файлів, RedBlackTrees з реалізацією червоно-чорного дерева, Cartesian Tree з реалізацією декартового дерева. Виконано порівняльний аналіз середнього часу роботи тестових проєктів для встановлення відмінностей між оптимізувальним компілятором та базовим компілятором, у ході якого виявлено, що навіть за використання оптимізаційних прапорів компіляції вдається досягти прискорення часу виконання до 15%, що підтверджує значущість дослідження.

Практичним результатом дослідження є реалізація прикладного інструменту, який дозволяє C++ розробникам застосовувати регульовану оптимізацію та впливати на обмеження розміру виконуваного файлу.

Список використаних джерел:

1. Налаштування компілятора: Попередження та помилки. URL: https://acode.com.ua/urok-9-nalashtuvannya-kompilyatora-poperedzhennya-ta-pomylky.
2. Writing an LLVM Pass (legacy PM version). URL: https://llvm.org/docs/WritingAnLLVMPass.html.
3. Clang/LLVM support in Visual Studio projects. URL: https://learn.microsoft.com/en-us/cpp/build/clang-support-cmake?view=msvc-170&source=recommendations.
4. CMake: The Standard Build System. URL: https://cmake.org/features.
5. Чикунов П.О. Відстеження поширення по програмі неперевірених зовнішніх даних / Європейські орієнтири розвитку України: науково-практичний вимір в умовах воєнних викликів : Матеріали Міжнародної науково-практичної конференції (Одеса, 26 квітня 2024 р.). Одеса, вид-во «Гельветика», 2024. С. 100-105.

Ключові слова: оптимізація часу виконання, компіляція, C++, вбудовування функцій, Low Level Virtual Machine

Keywords: runtime optimization, compilation, C++, function embedding, Low Level Virtual Machine

Науковий керівник: доцент П. Чикунов

ПРОГРАМУВАННЯ ЧАТ-БОТІВ ЗАСОБАМИ NODE.JS

Сідельников Денис

студент 1-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

У сучасному інформаційному середовищі чат-боти стають все більш важливими інструментами, що використовуються в самих різних сферах клієнтської підтримки: електронна комерція, освіта, call-центри, ігрова індустрія, культурно-просвітницькі складові нашого життя [1]. За місцем розташування чат-боти можуть бути у месенджерах, у спеціалізованих програмних застосунках, вбудовані в сайти [2]. Так чи інакше використання чат-ботів полегшує і прискорює процес замовлення товарів або послуг, допомагає клієнту відповісти на його запитання або вирішити інші комунікаційні запити або проблеми щодо унікальних завдань чи то налаштувань. Вони дозволяють автоматизувати взаємодію з користувачами, забезпечуючи швидкі й ефективні відповіді в режимі 24/7.

Використання Node.js для програмування чат-ботів має кілька переваг. По-перше, Node.js є легковаговою та високоефективною платформою, яка дозволяє швидко реалізовувати функціональність чат-ботів і підтримує асинхронне програмування. Крім того, Node.js має численні зовнішні модулі та пакети, що полегшують та прискорюють розробку.

Node.js (https://nodejs.org/) є кросплатформним середовищем для виконання JavaScript на рушії V8 від Google Chrome. Завдяки інтеграції з V8, Node js дає змогу обробляти численні паралельні запити і працювати з неблокувальними операціями введення/виведення, що критично важливо для чат-ботів.

Серед відомих фреймворків і бібліотек для розробки чат-ботів на Node.js можна відзначити Botpress, Microsoft Bot Framework, Dialogflow та Botkit. Ці інструменти надають розробникам широкий функціонал для створення різноманітних чат-ботів, включаючи інтеграцію зі сторонніми сервісами, оброблення природної мови та взаємодію з користувачами через різні канали зв'язку. Фреймворки Botkit (https://www.botkit.com/) та Microsoft Bot Framework (https://dev.botframework.com/) значно полегшують розробку чат-ботів на Node.js, забезпечуючи вбудовану підтримку популярних месенджерів та бібліотек. Botkit надає зручний програмний інтерфейс для створення конверсаційних потоків та оброблення різних типів вхідних даних від користувачів. Microsoft Bot Framework пропонує сервіси для розгортання, тестування та безпосереднього підключення ботів до власних каналів.

З позиції архітектури є різні підходи до реалізації чат-ботів. Один з найпоширеніших підходів – це використання архітектури клієнт-сервер, де клієнти (користувачі) спілкуються з сервером (чат-ботом) через мережу. Цей підхід дозволяє легко масштабувати та керувати чат-ботом.

Для інтеграції з месенджерами (Telegram, Viber, WhatsApp тощо) використовують спеціалізовані бібліотеки та API, а саме: messaging-apis, node-telegram-bot-api, node-viber-bot-api тощо. Ці бібліотеки абстрагують специфічні деталі різних платформ месенджерів та уніфікують взаємодію з ними [3]. У сучасному середовищі використання чат-ботів стає все більш популярним, особливо в контексті взаємодії з користувачами через платформу Telegram. Для створення таких ботів використовуються різноманітні технології, серед яких бібліотеки telegraf та node-telegram-bot-api для взаємодії з Telegram Bot API, а також бібліотека axios для виконання HTTP-запитів і sqlite3 для роботи з базою даних SQLite.

Однією з ключових переваг Node.js для створення чат-ботів є його здатність до масштабованості та обробки великої кількості одночасних з'єднань з

високою продуктивністю. Це досягається завдяки неблокуючій моделі введення/виведення та подієвому циклу, що дозволяє ефективно розподіляти ресурси між численними клієнтами, що є критично важливим для чат-ботів. Особливу увагу слід звернути на можливість низькорівневого контролю над TCP-сокетами, яку надає Node.js. Це дозволяє оптимізувати трафік даних та знизити накладні витрати, що є важливими аспектами для швидкої відповіді на запити користувачів та загальної продуктивності чат-бота.

Попри переваги під час розробки чат-ботів на Node.js можуть виникати певні проблеми. Наприклад, асинхронна модель програмування може бути складною для розуміння та налагодження, особливо для початківців. Крім того, недостатній рівень типобезпеки в JavaScript та відсутність офіційної підтримки типізації в Node.js можуть призводити до проблем у розробці та підтримці коду. Для подолання цих обмежень поширено використовують TypeScript – надбудову над JavaScript, яка додає статичну типізацію. TypeScript дозволяє визначати типи даних змінних, функцій та класів на етапі написання коду, що полегшує виявлення помилок на етапі компіляції та підвищує рівень інтелектуальної підтримки коду [4]. Це зробило TypeScript популярним інструментом у розробці Node.js застосунків, особливо для великих проєктів, де важлива чітка структура та підтримка коду.

Node.js є чудовим засобом для розробки чат-ботів, завдяки його масштабованості, універсальності та ефективності. Незалежно від різновиду та сфери застосування чат-бота, Node.js може допомогти створити надійного, швидкого та легкого в обслуговуванні чат-бота. Але при цьому, як і в будь-якому іншому проекті розробки програмного забезпечення, важливо ретельно спланувати процес розробки чат-бота та дотримуватися найкращих практик, щоб забезпечити успішний результат. Використовуючи потужність Node.js і широкий спектр доступних фреймворків та інструментів, можна створити чат-боти для відповідних потреб користувача.

Список використаних джерел:

1. Трофименко О. Г. Сфери застосування чат-ботів. Інформаційне суспільство: проблеми та перспективи : матер. VII всеукр. наук.-практ. конф. (20 травня 2022). Одеса, 2022. С. 68-71. DOI: 10.32837/11300.17736.
2. О.Трофименко, Прокоп Ю.В., О.Задерейко, Логінова Н.І. Класифікація чат-ботів. Системні технології. 2022. № 2 (139). С. 147-159. DOI: 10.34185/1562-9945-2-139-2022-14. URL: https://journals.nmetau.edu.ua/index.php/st/issue/view/120/87
3. Bot API Library Examples. URL: https://core.telegram.org/bots/samples

4. How to Build a Telegram Bot Using Typescript & Node.js. URL: https://www.cyclic.sh/posts/how-to-build-a-telegram-bot/

Ключові слова: чат-бот, Node.js, Telegram
Keywords: chat bot, Node.js, Telegram

Науковий керівник: доцент О. Трофименко

СТВОРЕННЯ ВЕБ-ВОДАТКІВ З ВИКОРИСТАННЯМ ТЕХНОЛОГІЙ MERN

Стоянов Артем

студент 4-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

У сучасному світі, веб-технології з кожним роком розвиваються дуже стрімкого, тому з'являються нові методи розробки веб-застосунків, що дають змогу створювати потужні продукти швидко та більш легко. Одним з таких методів є стек MERN.

Стек Mern – потужний інструмент у руках досвідчених розробників для створення та розгортання унікальних SPA застосунків. Він поєднує кілька технологій – MongoDb, Express.js, React.js, Node.js. З перших букв назв технологій складається абревіатура MERN [1].

Цей стек дозволяє створювати якісні, високофункціональні застосунки, враховуючи потреби замовників та досягнути максимальну зручність для взаємодії користувача із застосунком.

Давайте більш детально розглянемо технології, що використовуються у стеку MERN:

MongoDB – це база даних документів, розроблена для полегшення розробки та масштабування програм. Вона була обрана для керування базами даних [2].

Основні можливості MongoDB включають:

- Збереження даних у форматі JSON-подібних документів, що дозволяє гнучко організувати структуру даних;
- Використання гнучкої мови для формування запитів;
- Підтримка індексів для швидкого доступу до інформації;
- Функціонал профілювання запитів, для підвищення продуктивності;
- Ефективне зберігання різноманітних даних, таких як фотографії та відео;
- Ведення журналу операцій для моделювання баз даних.

Використання MongoDB дозволить легко вносити зміни у дані та їх структуру, не шкодячи вашому проекту, завдяки гнучкісті роботи з даними.

Express.js – це мінімалістичний та гнучкий фреймворк для веб-застосунків, побудованих на Node.js, що надає широкий набір функціональності. Маючи в свому розпорядженні безліч допоміжних HTTP-методів та проміжних обробників, створювати надійні API можна легко і швидко [3].

Крім того, Express.js пропонує широкий набір додаткових модулів, які допомагають у реалізації інших функцій, таких як маршрутизація, обробка запитів форми та автентифікація.

React – це бібліотека JavaScript для відтворення інтерфейсів користувача. Інтерфейс користувача складається з невеликих елементів, таких як кнопки, текст і зображення. React дозволяє вам об'єднувати їх у багаторазово використовувані вкладені компоненти. Від веб-сайтів до програм для телефону, все на екрані можна розбити на компоненти [3]. Також він має такі переваги [4]:

– Швидкість програми збільшується завдяки внесенню мінімальних змін до реальної DOM завдяки використанню віртуальної DOM і ефективного методу оновлення;

– Дизайн компонентів, який лежить в основі JS React, дозволяє розділити інтерфейс на окремі компоненти. Це полегшує написання, тестування та підтримку коду, останній з яких можна легко змінювати та повторно використовувати;

– React заохочує односторонній потік даних, що робить керування станом програми легшим і передбачуваним, а також полегшує тестування та налагодження програм;

– Існує велика та активна спільнота розробників, які використовують React. Багато інструментів розробки, бібліотек і ресурсів. Крім того, велика кількість сторонніх бібліотек допомагає React і розширює його функції.

Node.js – це безкоштовне міжплатформне середовище виконання JavaScript із відкритим кодом, яке дозволяє розробникам створювати сервери, веб-програми, інструменти командного рядка та сценарії. Node.js має унікальну перевагу, оскільки мільйони інтерфейсних розробників, які пишуть JavaScript для браузера, тепер можуть писати код на стороні сервера на додаток до коду на стороні клієнта без необхідності вивчати абсолютно іншу мову [5].

Завдяки широкому спектру модулів і пакетів, доступних через npm, Node.js дозволяє розробникам швидко створювати широкий спектр програм, включаючи програми баз даних, веб-сервери до інструментів командного

рядка. Node.js використовується багатьма великими підприємствами, через свою, через свою ефективність, швидкодію та простоту використання.

MERN займає провідне місце в галузі розробки додатків завдяки таким факторам:

– Кожен рядок коду пишеться на JavaScript, що позбавляє необхідності перемикати контекст.;

– Стек MERN сприяє зростанню ідеальних умов для створення високопродуктивних web-додатків;

– MERN пропонує широкий спектр можливостей тестування, які охоплюють весь життєвий цикл розробки програмного забезпечення.

Технологічний стек MERN є дуже корисним інструментом для розробки сучасних веб-додатків. Він поєднує різноманітні інструменти та технології, що дозволяє розробникам створювати динамічні та ефективні онлайн-додатки з єдиною мовою програмування для використання як на стороні клієнта, так і на стороні сервера. Будь то великий бізнес-проект чи маленька персональна веб-сторінка, у MERN є все необхідне для належної реалізації вашої ідеї.

Список використаних джерел:

1. Створення сайтів та застосунків на MERN: http://surl.li/tteih
2. MongoDB – Official website: https://www.mongodb.com/docs/manual/
3. Express.js – Official website: https://expressjs.com/uk/
4. React.js – Official Website: https://react.dev/learn/describing-the-ui
5. Node.js – Official website: https://nodejs.org/en/

Ключові слова: MERN, стек, сервер, веб
Keywords: MERN, stack, server, web

Науковий керівник: доцент Ю. Лобода

ПРАВОВЕ РЕГУЛЮВАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В ЄС

Пашко Андрій

студент 1 курсу факультету адвокатури та антикорупційної діяльності Національного університету «Одеська юридична академія»

13 березня 2024 року Європейський парламент ухвалив довгоочікуваний Регламент про регулювання штучного інтелекту в Європейському Союзі. Регламент передбачає створення єдиного правового підходу до регулювання використання штучного інтелекту, незалежно від індустрії або технології [1].

Регламент поширюється на операторів (постачальників, імпортерів, дистриб'юторів і виробників) систем штучного інтелекту (ШІ) в ЄС або за його межами, якщо ШІ використовується на території ЄС.

Невідповідність вимогам Регламенту може мати наслідком застосування високих штрафів, які можуть сягати до 35 млн євро, або 7% від річного обороту. Регламент запроваджує ризик-орієнтований підхід та розділяє системи ШІ на чотири категорії залежно від групи ризику: Неприпустимий ризик означає, наприклад, що системи ШІ, які маніпулюють поведінкою людей, не можна вводити в експлуатацію, розміщувати на ринку або використовувати. Високий ризик виникає, коли системи ШІ можуть впливати на безпеку або фундаментальні права. Ці системи переважно використовуються у таких галузях, як критична інфраструктура, освіта, працевлаштування, охорона здоров'я, банківські послуги, правоохоронна діяльність, управління міграцією та кордонами, правосуддя, виборчий процес. Обмежений ризик застосовується до систем ШІ з обмеженим впливом на маніпуляції. Мінімальний ризик охоплює системи ШІ, які не належать до зазначених вище категорій.

Генеративні системи штучного інтелекту повинні чітко позначати свої результати як штучно створені. Це означає, що будь-який аудіо- або відеоконтент, а також зображення, створені з використанням штучного інтелекту, мають містити відповідні маркери їхнього походження. Усі системи дистанційної біометричної ідентифікації вважаються потенційно небезпечними та підлягають суворим вимогам.

Використання дистанційної біометричної ідентифікації в загальнодоступних місцях для правоохоронних цілей заборонено, за винятком випадків, передбачених законом. Використання систем штучного інтелекту з низьким або мінімальним ризиком буде допускатися лише за умови відповідності вимогам, встановленим Регламентом.

Системи штучного інтелекту з високим ризиком підлягають найсуворішому правовому контролю. Перед впровадженням системи штучного інтелекту, яка вважається високоризиковою, оператор повинен провести оцінку впливу на основні права, щоб визначити, як використання такої системи може вплинути на права осіб, що потенційно можуть постраждати.

Згідно з Регламентом, надійним ШІ має бути: (1) законний – з дотриманням усіх чинних законів і правил (2) етичний – повага до етичних принципів і цінностей (3) надійний – як з технічної точки зору, так і з урахуванням соціального середовища [2].

Оператори систем ШІ з високим рівнем ризику зобов'язані: запровадити системи управління ризиками системи ШІ протягом усього життєвого циклу; забезпечувати релевантність, повноту та репрезентативність наборів даних для навчання, перевірки та тестування системи ШІ, щоб мінімізувати потенційні ризики; розробити технічну документацію для демонстрації відповідності системи ШІ вимогам Регламенту та провести оцінку відповідності; впровадити в систему ШІ можливість автоматичної фіксації подій, важливих для визначення ризиків національного рівня та суттєвих модифікацій протягом життєвого циклу системи; надати інструкції щодо використання системи ШІ в разі інтеграції системи іншими провайдерами, з метою забезпечення відповідності при подальшому розгортанні; забезпечити належні заходи людського нагляду для мінімізації ризику в разі інтеграції системи ШІ іншими провайдерами; розробити систему ШІ з високим рівнем ризику, щоб досягти належного рівня точності, надійності та кібербезпеки; встановити систему управління якістю для забезпечення відповідності.

Системи загального призначення штучного інтелекту (GPAI) можуть використовуватися як системи із високим ризиком штучного інтелекту або інтегруватися в такі системи. Оператори GPAI мають співпрацювати з операторами високоризикових систем штучного інтелекту, щоб забезпечити відповідність вимогам Регламенту [3].

Для регулювання GPAI встановлено спеціальні вимоги. Серед них - вимоги щодо прозорості, які включають відповідність законодавству ЄС щодо авторського права та публікацію опису вмісту, що використовується для навчання, для систем GPAI та їхніх моделей.

Загалом можна відзначити що правове регулювання штучного інтелекту в Європейському Союзі (ЄС) на сьогоднішній день визначається складними взаємозв'язками між інноваціями та захистом прав людини, а ЄС розвиває стратегічний підхід до штучного інтелекту, зосереджуючись на принципах етичного використання, прозорості та відповідальності. При цьому регулювання штучного інтелекту в ЄС базується на принципі «людина у центрі», спрямованому на забезпечення безпеки, приватності та захисту особистих даних.

Список використаних джерел:

1. В ЄС прийнято регламент про регулювання штучного інтелекту. URL: http://surl.li/toeqm.
2. Ethics guidelines for trustworthy AI. URL: https://digital-strategy.ec.europa.eu/en/library/ethics-guidelines-trustworthy-ai.
3. 2023 GPAI Ministerial Declaration 14 December 2023, New Delhi URL: https://gpai.ai/.

Ключові слова: штучний інтелект , ЄС, Регламент про тучний інтелект
Keywords: Artificial Intelligence, EU, AI Act about Artificial Intelligence

Науковий керівник: доцент Р. Чанишев

ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ В ОБОРОННІЙ СФЕРІ

Задерейко Олександр
кандидат технічних наук, доцен, доцент кафедри інформаційних технологій Національного університету "Одеська юридична академія"

Гура Володимир
кандидат технічних наук, доцент кафедри інформаційних технологій Національного університету "Одеська юридична академія"

Процик Максим
студент 1 курсу факультету кібербезпеки та інформаційних технологій Національного університету "Одеська юридична академія"

У сучасному світі все здебільшого значення набувають технології, завдання яких автоматизація процесів, які раніше виконувалися людьми. Штучний інтелект – одна з новітніх областей наука та технологія створення інтелектуальних машин, в особливості інтелектуальних комп'ютерних програм [1]. Його основним завданням є використання комп'ютерів для розуміння людського інтелекту та виконання завдань, які раніше були в змозі виконувати лише людина, що і надало стимулу його розвитку та удосконаленню.

Ця тема стає все більш актуальною, оскільки сучасні технології мають можливість впровадження інноваційних рішень у військовій галузі. Інтеграція штучного інтелекту в оборонну сферу та аспекти його використання вивчалися в наукових роботах таких вчених, як: Д. Маккарті, М. Мінскі, А. Тьюринг, Г. Саймон, Д. Вейценбаум [2].

Штучний інтелект - новітній напрям прикладних наукових розробок, що виник на основі таких традиційних дисциплін, як системний аналіз, теорія прийняття рішень і дослідження операцій [3]. Цілком природно, що фундаментальною особливістю концепції штучного інтелекту є використання його як основного інструменту для впровадження різних типів збройної боротьби [4].

Важливість використання штучного інтелекту для національної безпеки підтверджується висновками науково-технічних підрозділів організації НАТО [5].

Відповідно, найважливішими напрямками для використання штучного інтелекту є: великі дані, автономні транспортні засоби, космос, гіперзвукові об’єкти, квантові і біотехнології та виготовлення нових матеріалів [6].

Основними функціями інтеграції штучного інтелекту в оборонну сферу є підвищення ефективності військових операцій та зменшення ризику для військових сил України та громадського населення:

– автоматизація специфічних військових завдань, таких як розвідка, спостереження та атака.

– аналіз зображень, отриманих за допомогою безпілотних літальних апаратів (БПЛА), для виявлення потенційних загроз;

– виявлення цілей, управління зброєю для отримання найбільш ефективних даних і системи нападу, які можуть бути виконані без втручання людини (наприклад, БПЛА);

– побудова оборонних систем для захисту цивільного населення, військовослужбовців та певних стратегічних об'єктів інфраструктури;

– підвищення ефективності управління військами, штучний інтелект може бути використаний для підвищення ефективності управління військами, надаючи командирам кращу обізнаність про ситуацію і забезпечуючи більш ефективне прийняття рішень;

Це підтверджує актуальність використання штучного інтелекту для впровадження інноваційних рішень у військовому секторі. Приклади цього ми все частіше бачимо на передовій сучасної війни. Різні країни вже налагодили впровадження штучного інтелекту для покращення своєї обороноздатності. Цей процес складний і вимагає великих інвестицій.

З огляду на сучасні війни, можна побачити, що все більш великого значення у бойових діях набувають БПЛА. Вони використовуються для виконання широкого спектру завдань, включаючи розвідку, удари по супротивнику, а також постачання для підтримки військ. Можливість самостійно орієнтуватися у просторі, автоматично виявляти і класифікувати цілі, самостійно вражати цілі без участі людини робить БПЛА одним із найважливіших компонентів сучасних війн. Вони розрізняються за такими типами, як:

– розвідувальні - вони використовуються для збору інформації про умови на полі бою. Вони оснащені різними давачами, включаючи камери, радари та інфрачервоні сенсори;

– ударні - використовуються для знищення цілей. Вони можуть бути оснащені ракетами, бомбами та іншим озброєнням;

– підтримки військ - використовуються для надання підтримки військам на полі бою. Вони використовуються для транспортування вантажів, евакуації поранених і надання вогневої підтримки;

– забезпечення доставки вантажів військам - для транспортування продовольства, боєприпасів та інших матеріалів.

Автономні БПЛА програмуються з великою кількістю альтернативних реакцій на різноманітні виклики, які можуть виникнути під час виконання своїх місій. БПЛА, що використовуються як вдосконалені транспортні засоби, запрограмовані визначеним людиною набором поведінки для вирішення нештатних завдань, вже випробовуються в багатьох цивільних університетах і військових науково-дослідних установах. Це підтверджується випробуваннями "роїв БПЛА", які отримують і виконують завдання від інших БПЛА, робота яких заснована на повністю автономній обробці інформації [7].

Найбільш вдосконаленими БПЛА, здатними корегуватися та виконувати складні завдання за допомогою штучного інтелекту є:

– MQ-9 Reaper - ударний безпілотник, розроблений Сполученими Штатами. MQ-9 Reaper використовує штучний інтелект для автономної навігації, розпізнавання цілей і управління зброєю;

– J-20 - багатоцільовий винищувач, розроблений Китаєм. J-20 використовує штучний інтелект для аналізу даних, розпізнавання цілей і управління зброєю;

– Sky Guardian- це розвідувальний безпілотник, розроблений у Великій Британії. Sky Guardian використовує штучний інтелект для автономної навігації та розпізнавання цілей.

Можливості штучного інтелекту, що відкриваються завдяки його інтеграції, змушують уряди і оборонні компанії по всьому світу інвестувати більше ресурсів в його інтеграцію у оборонний сектор. Лідерами серед інвестицій є передові технологічні країни та корпорації:

У США Міністерство оборони інвестує значні кошти в розробку та впровадження систем штучного інтелекту. У 2023 році Міністерство оборони США виділило 10 мільярдів доларів на програму DefenceAI. Ця програма спрямована на розробку систем штучного інтелекту для використання в таких сферах, як розвідка, напад, кібервійна та кібербезпека [8].

Китай також є одним із лідерів у розвитку оборонного напрямку штучного інтелекту. Уряд Китаю виділить 150 мільярдів юанів (близько 22 мільярди доларів США) на національну програму розвитку оборонного штучного інтелекту на

наступні 5 років. Програма спрямована на розробку систем штучного інтелекту для використання в таких сферах, як розвідка, наступальні дії, протиповітряна оборона та кібервійна [9].

Ізраїль також є одним з лідерів у розробці оборонного штучного інтелекту: У 2023 році уряд Ізраїлю виділив 1 мільярд доларів США на програму оборонного штучного інтелекту. Програма спрямована на розробку систем штучного інтелекту для використання в таких сферах, як розвідка, наступальні дії, протиповітряна оборона та кібервійна [10].

Однак використання штучного інтелекту також вимагає вирішення етичних, правових питань та питань безпеки. Тому, перш ніж впроваджувати штучний інтелект в оборонний сектор, необхідно ретельно оцінити всі можливі наслідки і забезпечити належні заходи контролю і нагляду.

Список використаних джерел:

1. Коцовський, В. М. (2017). Методи та системи штучного інтелекту. URL: https://dspace.uzhnu.edu.ua/jspui/handle/lib/16306. (Дата звернення 29.04.2024).
2. "A Proposal for the Dartmouth Summer Research Project on Artificial Intelligence" by John McCarthy. URL: http://jmc.stanford.edu/articles/dartmouth/dartmouth.pdf. (Дата звернення 29.04.2024).
3. І. Ю. Черепанська, А. Ю. Сазонов, Сучасні інформаційні технології та системний аналіз у наукових дослідженнях. URL: ttps://ela.kpi.ua/bitstream/123456789/47996/1/Suchasni.doc.pdf. (Дата звернення 29.04.2024).
4. Демідов Б.О., Борисенко М.В., Кучеренко Ю.Ф., Задорожна А.Ю. Перспективні напрямки розвитку і використання методів і технологій штучного інтелекту для Збройних Сил України в рамках впровадження сучасних інновацій у військовій сфері. URL: https://cpc.com.ua/articles/shtuchniy-intelekt-v-ukraini-dosvid-vikoristannya-perspektivi-trendi-v-media. (Дата звернення 29.04.2024).
5. Штучний інтелект та національна безпека: Дослідження НАТО. URL: https://www.nato.int/docu/review/articles/2021/10/25/an-artificial-intelligence-strategy-for-nato/index.html. (Дата звернення 29.04.2024).
6. Ніно Пацурія. Упровадження технологій штучного інтелекту в забезпечення національної безпеки та обороноздатності україни: правові проблеми і перспективи повоєнного періоду. URL: http://uran.inprojournal.org/article/view/282185. (Дата звернення 29.04.2024).
7. Col G. Lage Dyndal, LtCol T. Arne Berntsen, Pr S. Redse-Johansen Автономні військові дрони – це вже не фантастика. URL: https://www.nato.int/docu/review/uk/articles/2017/07/28/avtonomn-vjs-kov-droni-tse-vyoe-ne-fantastika/index.html. (Дата звернення 29.04.2024).
8. The Pentagon is requesting more than $3 billion for AI and JADC2. URL: https://defensescoop.com/2023/03/13/pentagon-requesting-more-than-3b-for-ai-jadc2/. (Дата звернення 29.04.2024).

9. Чан М. США можуть нейтралізувати військову перевагу. URL: https://www.scmp.com/news/china/military/article/3247868/us-advances-ai-blunt-china-military-depend-more-streamlined-pentagon-senior-official. (Дата звернення 29.04.2024).
10. Звіт Стокгольмського міжнародного інституту досліджень миру (SIPRI) URL: https://www.sipri.org/research/armament-and-disarmament/arms-and-military-expenditure/military-expenditure. (Дата звернення 29.04.2024).

Ключові слова: штучний інтелект, оборонна сфера, безпека
Keywords: artificial intelligence, defense, security

ВИКОРИСТАННЯ ШТУЧНОГО ІНТЕЛЕКТУ У ВІЙСЬКОВИЙ ТЕХНОЛОГІЯХ

Трофименко Олена

кандидат технічних наук, доцент, доцент кафедри інформаційних технологій Національного університету «Одеська юридична академія»

Кіх Яна

студентка 2-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

У сучасному світі штучний інтелект (ШІ) перетворюється на ключовий елемент багатьох сфер людської діяльності. Важливу роль штучний інтелект відіграє й у військовій сфері, де його використання відкриває нові горизонти ефективності, точності та безпеки.

Міжнародний досвід локальних конфліктів останніх років показує, що сучасні бойові дії відрізняються динамічністю і проводяться за умов високої інтенсивності. Ця обставина обумовлює формування величезних потоків інформації, оброблення яких й оперативне прийняття по них ефективних рішень наразі доволі проблематичні без застосування технологій ШІ. Успіх на полі бою за нинішніх умов безпосередньо залежить від технологічних інновацій, а ШІ стає дедалі частіше вирішальним фактором у визначенні результатів бойових операцій.

ШІ здатен імітувати різні аспекти людського пізнання, сприйняття, міркування, планування та здатність автономно виконувати різні завдання: розуміння мови, розпізнавання об’єктів і звуків, навчання, прогнозування [1].

Загалом використання штучного інтелекту важливе для забезпечення захисту, що є одним із стратегічних пріоритетів у сфері формування та розвитку військового потенціалу та озброєння.

Ні для кого не секрет, що зараз, під час проведення військових дій, наші військові активно використовують дрони, а штучний інтелект підвищує їхню точність та автономність. Дрони зі ШІ можуть стежити за великими територіями, виявляти підозрілі дії та відстежувати людей або транспортні засоби в режимі реального часу. Алгоритми розпізнавання обличчя та виявлення об'єктів покращують їхню здатність ідентифікувати потенційні загрози. Але на цьому можливості не вичерпується, ще один напрям – скоординована атака "роєм" [2]. Дрони можуть одночасно обстрілювати цілі, здійснюючи скоординовану ройову атаку. Також вони можуть ефективно використовуватись для придушення чи знищення ППО противника, спотворення зображення засобів на радарах противника, виявлення місць розташування радарів противника та передачі інформації операторам. Це відбувається, коли дрони автоматично взаємодіють, обмінюються інформацією про цілі та розподіляють її між собою автоматично або за мінімальної участі оператора [3].

ТГП і «Волля» – це турелі або кулеметні установки, які створила українська компанія DevDroid. Турель дозволяє керувати кулеметом дистанційно через комп'ютер і використовує комп'ютерний зір для пошуку та відстеження цілей – ворожих солдатів. Нейромережа вміє знаходити їх на відстані до 800 метрів. Система працює з використанням ШІ в двох режимах: ручному й автоматичному. Бойові модулі «Волля» і ТГП автоматично знаходять цілі на полі бою, наводяться на них, налаштовують балістику, а від оператора потрібно лише натиснути кнопку «вогонь» [4].

Крім того, ШІ використовується для контррозвідувальних заходів, зокрема для оцінки того, які військові цілі можуть стати предметом уваги ворога, а також для захисту комунікацій в кіберпросторі. Тобто ШІ може виступати не тільки як технологія для захисту, а й цілком може бути й наступальною технологією, яка дозволяє прогнозувати дії ворога, прорахувати й пропонувати оптимальні стратегії поведінки, збільшуючи тим самим втрати ворога [5]. Штучний інтелект, закладений в операційну систему Lattice, автономного підводного човна Ghost Shark від австралійської компанії Anduril використовує дані з різних сенсорів, комп'ютерне і машинне навчання. Зброя, яка має власну операційну систему, виводить військову техніку на якісно новий рівень. Прототипом цього проєкту є менша субмарина Dive-LD від цих же розробників, яка може опускатися на

глибину до 6000 метрів, ховаючись від ворожих радарів, і працювати автономно протягом 10 днів. Для порівняння, військові підводні човни з людьми на борту занурюються не більше, ніж на кілька сотень метрів [6].

Ще одним прикладом інтеграції ШІ у військову сферу є безпілотний бойовий літак Loyal Wingman, розроблений компанією Boeing у співпраці з Королівськими австралійськими Військово-повітряними силами (ВПС). Літак може здійснювати автономні польоти, виконуючи завдання за заздалегідь запрограмованими алгоритмами або реагуючи на зміни в реальному часі. Loyal Wingman координує свої дії з пілотованими літаками, обмінюючись інформацією в реальному часі та спільно плануючи операції. ШІ аналізує великі обсяги даних, зібраних сенсорами і розвідувальними системами літака, що дозволяє швидко і точно ідентифікувати цілі, оцінювати бойову обстановку і надавати командиру необхідну інформацію для прийняття рішень. Крім того, ШІ у Loyal Wingman є здатність до самонавчання та адаптації, що підвищує його ефективність та надійність у майбутніх операціях. Літак також може бути оснащений обладнанням для електронної війни, включаючи системи радіоелектронної боротьби, які використовують ШІ для виявлення і придушення ворожих комунікацій та радарів. Використання автономних систем і можливості співпраці з пілотованими літаками відкривають нові горизонти для військової авіації, забезпечуючи підвищення ефективності і безпеки виконання місій [7].

З іншого боку, технології ШІ не є досконалими і подекуди демонструють неточності. Системи ШІ є вразливими до технічних збоїв і помилок, які можуть стати критичними під час бойових операцій. Недостатня надійність технологій може поставити під загрозу виконання важливих місій. Неправильні або неточні алгоритми можуть призвести до хибних рішень на полі бою, що може спричинити небажані наслідки, помилкові удари включаючи та невинні жертви. Тому важливо враховувати ризики, пов'язані з розвитком і застосуванням технологій ШІ.

Список використаних джерел:

1. О.Трофименко, Кіх Я.Т. Застосування алгоритмів штучного інтелекту для військових задач. Сучасні технології в енергетиці, електромеханіці, системах управління та машинобудуванні: матер. VI всеукр. наук.-практ. інтернет-конференції. Харків, 06-07 грудня 2023 р. Харків: ННППІ УІПА. С. 28-29. URL: https://www.nnppi.in.ua/index.php/abit/2-uncategorised/270-naukovi-konferentsiyi
2. Smoke and mirrors: FoI shows there appears little substance to the RAF's drone swarm squadron. URL: https://dronewars.net/2022/07/04/smoke-and-mirrors-foi-shows-there-appears-little-substance-to-the-rafs-drone-swarm-squadron/

3. Залата О. Україна озброюється дронами з ШІ: чим кращі за звичайні та коли масово з'являться на фронті. URL: https://focus.ua/uk/digital/628357-ukrajina-ozbroyuyetsya-dronami-z-shi-chim-krashchi-za-zvichayni-ta-koli-masovo-z-yavlyatsya-na-fronti
4. DevDROID. URL: https://devdroid.tech/catalog
5. Бойко Д., Городиський І. Штучний інтелект у сфері оборони: виклики регулювання. URL: https://dc.org.ua/news/shtuchnyy-intelekt-u-sferi-oborony-vyklyky-regulyuvannya
6. ANDURIL. URL: https://www.anduril.com/
7. MQ-28. URL: https://www.boeing.com/defense/mq28#overview

Ключові слова: штучний інтелект (ШІ), військовий ШІ, оборонний сектор
Keywords: Artificial Intelligence (AI), Military AI, defense sector

ШТУЧНІ НЕЙРОННІ МЕРЕЖІ В ОПТИМІЗАЦІЇ КОМП'ЮТЕРНИХ МЕРЕЖ

Задерейко Олександр
кандидат технічних наук, доцент,
доцент кафедри інформаційних технологій
Національного університету "Одеська юридична академія"

Гура Володимир
кандидат технічних наук,
доцент кафедри інформаційних технологій
Національного університету "Одеська юридична академія"

Соломанін Іван
студент 2-го курсу факультету кібербезпеки та інформаційних технологій
Національного університету «Одеська юридична академія»

Штучні нейронні мережі (ШНМ) є моделі, натхненні структурою і функцією людського мозку. Вони складаються з безлічі штучних нейронів, об'єднаних у шари, які здатні навчатися на основі даних. ШНМ знайшли застосування в різних галузях, включаючи обробку зображень, розпізнавання мови, прогнозування часових рядів та багато інших [1].

Основні компоненти ШНМ включають вхідний шар, приховані шари і вихідний шар. Вхідний шар приймає вхідні дані, які потім передаються через приховані шари, де відбувається обробка та перетворення інформації, і нарешті вихідний шар видає результати.

Оптимізація комп'ютерних мереж є ключовим елементом для забезпечення високої продуктивності, надійності та ефективності роботи мережевих систем. В умовах зростаючих обсягів даних та вимог до швидкості передачі інформації оптимізація мережевих ресурсів стає невід'ємною частиною сучасних технологічних рішень.

Основні завдання оптимізації включають маршрутизацію, управління трафіком, балансування навантаження, виявлення і запобігання збоїв, а також моніторинг і управління мережевими ресурсами.

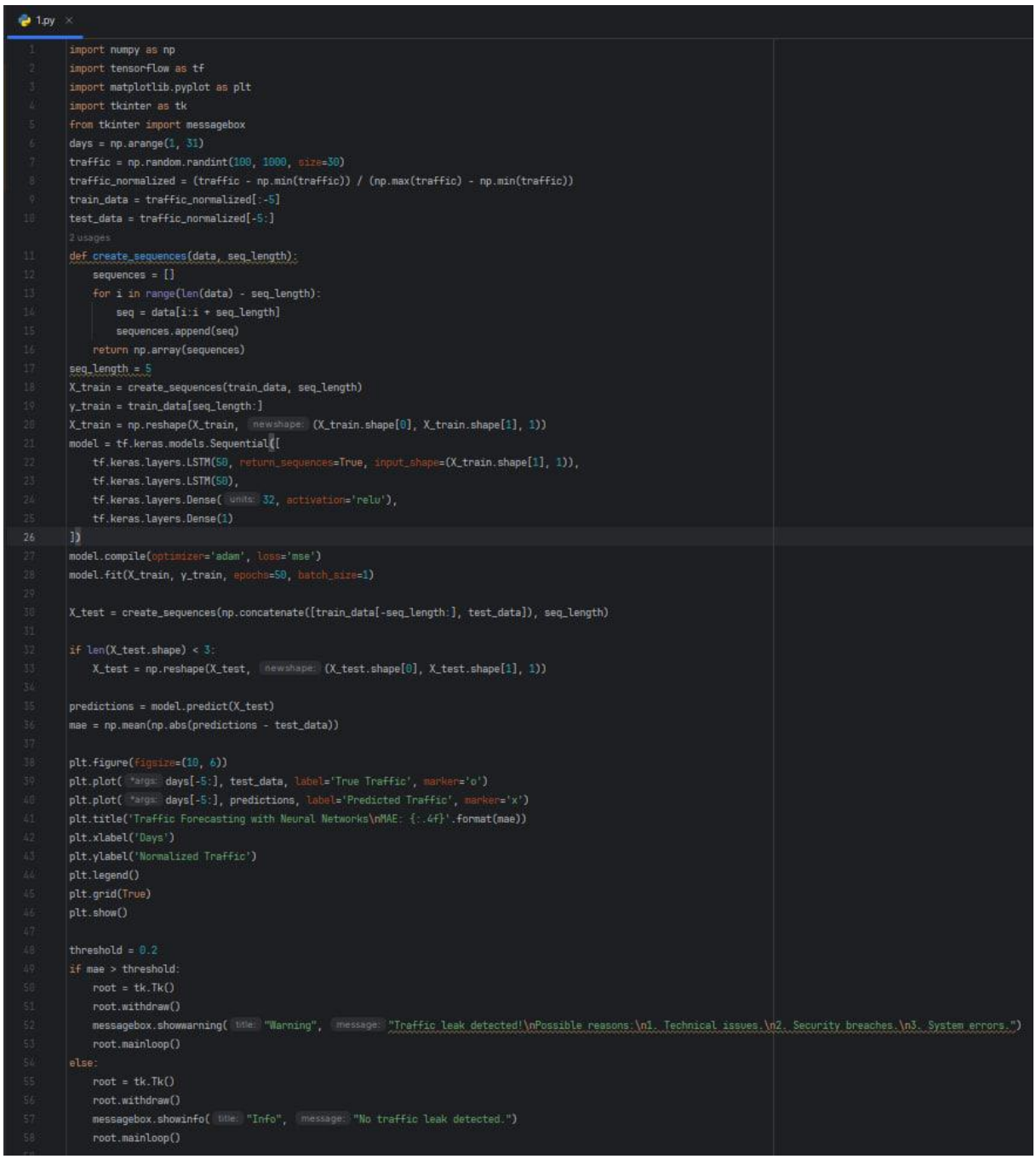
ШНМ пропонують нові можливості для оптимізації комп'ютерних мереж завдяки своїй здатності до адаптивного навчання та аналізу складних даних. Вони можуть бути використані для прогнозування трафіку, оптимізації маршрутизації, адаптивного управління ресурсами та виявлення аномалій у мережевому трафіку.

Алгоритми машинного навчання, засновані на ШНМ, можуть автоматично адаптуватися до умов, що змінюються, і оптимізувати роботу мережі, забезпечуючи високу продуктивність і надійність [2].

Переваги застосування ШНМ в оптимізації комп'ютерних мереж включають автоматизацію процесів, підвищення ефективності та надійності роботи мережі, адаптивність до умов, що змінюються, і здатність до виявлення складних закономірностей у даних [3].

Однак існують обмеження, такі як необхідність великого обсягу даних для навчання, складність налаштування та оптимізації моделей, а також потенційні ризики безпеки, пов'язані з використанням алгоритмів машинного навчання.

Код для прогнозування трафіку у комп'ютерних мережах за допомогою штучних нейронних мереж і рекурентних нейронних мереж (LSTM) [4], (рис. 1).



```python
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
import tkinter as tk
from tkinter import messagebox
days = np.arange(1, 31)
traffic = np.random.randint(100, 1000, size=30)
traffic_normalized = (traffic - np.min(traffic)) / (np.max(traffic) - np.min(traffic))
train_data = traffic_normalized[:-5]
test_data = traffic_normalized[-5:]
2 usages
def create_sequences(data, seq_length):
    sequences = []
    for i in range(len(data) - seq_length):
        seq = data[i:i + seq_length]
        sequences.append(seq)
    return np.array(sequences)
seq_length = 5
X_train = create_sequences(train_data, seq_length)
y_train = train_data[seq_length:]
X_train = np.reshape(X_train, newshape: (X_train.shape[0], X_train.shape[1], 1))
model = tf.keras.models.Sequential([
    tf.keras.layers.LSTM(50, return_sequences=True, input_shape=(X_train.shape[1], 1)),
    tf.keras.layers.LSTM(50),
    tf.keras.layers.Dense( units: 32, activation='relu'),
    tf.keras.layers.Dense(1)
])
model.compile(optimizer='adam', loss='mse')
model.fit(X_train, y_train, epochs=50, batch_size=1)

X_test = create_sequences(np.concatenate([train_data[-seq_length:], test_data]), seq_length)

if len(X_test.shape) < 3:
    X_test = np.reshape(X_test, newshape: (X_test.shape[0], X_test.shape[1], 1))

predictions = model.predict(X_test)
mae = np.mean(np.abs(predictions - test_data))

plt.figure(figsize=(10, 6))
plt.plot( *args: days[-5:], test_data, label='True Traffic', marker='o')
plt.plot( *args: days[-5:], predictions, label='Predicted Traffic', marker='x')
plt.title('Traffic Forecasting with Neural Networks\nMAE: {:.4f}'.format(mae))
plt.xlabel('Days')
plt.ylabel('Normalized Traffic')
plt.legend()
plt.grid(True)
plt.show()

threshold = 0.2
if mae > threshold:
    root = tk.Tk()
    root.withdraw()
    messagebox.showwarning( title: "Warning", message: "Traffic leak detected!\nPossible reasons:\n1. Technical issues.\n2. Security breaches.\n3. System errors.")
    root.mainloop()
else:
    root = tk.Tk()
    root.withdraw()
    messagebox.showinfo( title: "Info", message: "No traffic leak detected.")
    root.mainloop()
```

Рисунок 1 – Програмний код

Цей код представляє приклад прогнозування трафіку у комп'ютерній мережі за допомогою ШНМ і LSTM. В ньому створюються синтетичні дані про трафік, які нормалізуються та перетворюються на послідовності для навчання моделі. Модель навчається на навчальних даних, а потім використовується для прогнозування трафіку на тестовому наборі даних.

Результати прогнозу візуалізуються на графіку для порівняння з реальними значеннями трафіку (рис. 2) з обов’язковим повідомленням про наявність витоків трафіку або про відсутність його витоку.

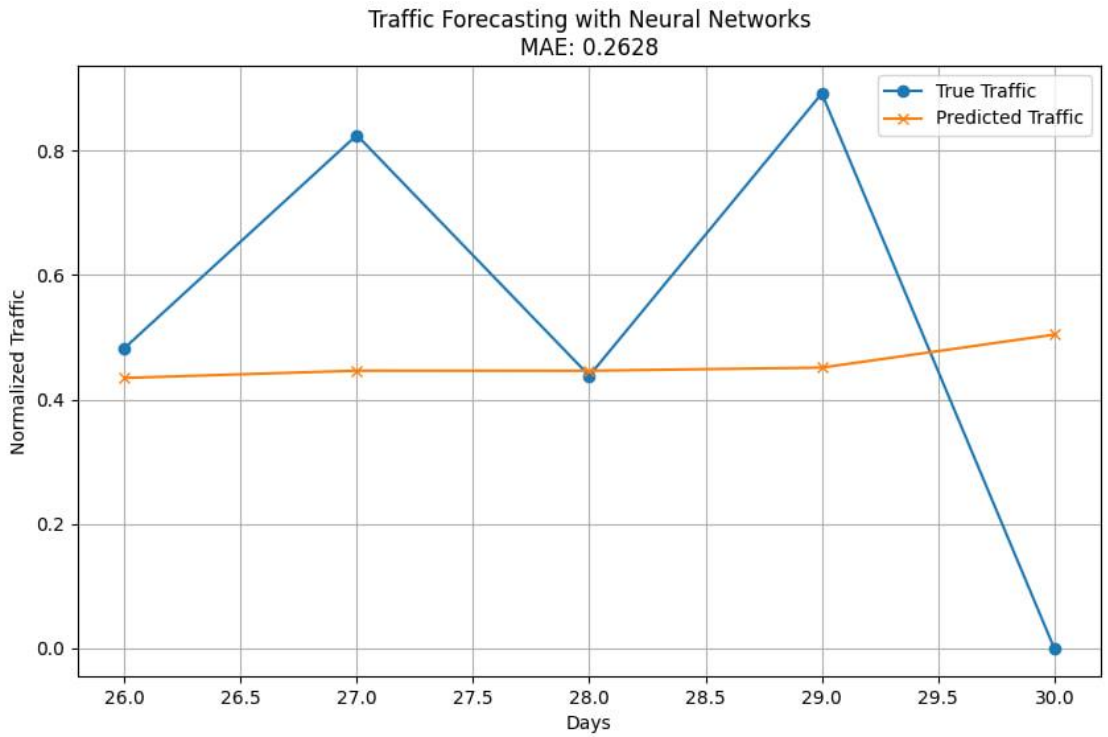


Рисунок 2 – Результат виконання програмного коду

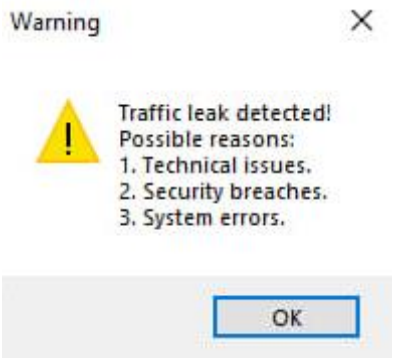


Рисунок 3 – Вивід повідомлення про виток трафіку.

Якщо середня абсолютна похибка перевищує заданий поріг (threshold), користувач отримає сповіщення про виявлення "витоку трафіку". Це може бути зумовлено різними причинами, такими як технічні проблеми, порушення безпеки або системні помилки. У протилежному випадку, якщо середня абсолютна похибка не перевищує поріг, користувач отримає інформаційне повідомлення про відсутність витоку трафіку. Код демонструє можливість використання машинного навчання для ефективного прогнозування та оптимізації управління трафіком у комп'ютерних мережах.

Штучні нейронні мережі є перспективним напрямом для оптимізації комп'ютерних мереж, забезпечуючи високу продуктивність, надійність і ефективність роботи мережевих систем. Подальші дослідження та розробки в цій галузі можуть призвести до створення більш досконалих та адаптивних мережевих рішень.

Список використаних джерел:

1. Штучні нейронові мережі .URL: https://futurum.today/shtuchni-neironni-merezhi/.
2. Машинне навчання. URL: https://www.it.ua/machine-learning/.
3. Оптимізації комп'ютерних мереж. URL: https://zp.edu.ua/metody-optymizaciyi-kompyuternyh-merezh/.
4. Рекурентні нейронні мережі. URL: https://itwiki.dev/recurrent-neural-network.

Ключові слова: мережа, інтернет, програмування, штучні нейронні мережі
Keywords: network, internet, programming, artificial neural networks.

ЩОДО ПРОБЛЕМИ ШТУЧНОГО ІНТЕЛЕКТУ ТА АВТОРСЬКОГО ПРАВА

Гордієнко Валерія

студентка 1 курсу факультету адвокатури та антикорупційної діяльності Національного університету «Одеська юридична академія»

Згідно з позовом, поданим 27 грудня 2023 р. відомою газетою The New York Times, фірмами OpenAI та Microsoft для навчання чат-ботів (наприклад, таких, як ChatGPT від OpenAI), незаконно використовувалися мільйони статей, захищених авторським правом [1]. У позові газети стверджується, що розробники намагалися «вільно використати значні інвестиції The New York Times у свою журналістику» для створення альтернативних способів доставки інформації читачам, оскільки на відміну від багатьох інших видань подібного роду, газета New York Times не надає безкоштовний доступ до своїх публікацій через Інтернет.

Газета не вимагала конкретної суми збитків, але оцінила збитки в 1 мільярд доларів. При цьому газета попросила обидві компанії знищити навчальні набори з матеріалами газети. Раніше газета намагалася домовитися про це безпосередньо з компанією, але безуспішно.

Цей крок став прецедентом, адже NYT перетворилося на перше велике американське медіа, яке публічно звинуватило розробників ШІ в подібному порушенні.

Кількість судових розглядів, пов'язаних з порушенням авторських прав компаніями, що розробляють генеративні моделі штучного інтелекту, стрімко зростає.

З іншого боку, технологічні компанії вдаються до аргументів про своє право на інновації, стверджуючи, що ШІ відкриває нові горизонти для творчості.

Так, OpenAI заявила, що The New York Times найняла фахівців для злому ChatGPT (продукту OpenAI), з метою «підлаштувати судовий позов» проти цього

провідного виробника систем ШІ. У своєму позові OpenAI стверджує, що газеті The New York Times знадобилися «десятки тисяч спроб, щоб згенерувати» ці нібито «вкрай аномальні результати», використовуючи помилку в програмі ChatGPT, яку буде тепер виправлено.

При цьому, жодні користувачі не могли використовувати ChatGPT для отримання статей з The New York Times. [2].

Аналогічні суперечки вирують у музичній та кіноіндустріях, а також у сфері стокових фото. За кожним подібним позовом стоять мільярдні фінансові претензії. Ця ситуація чітко демонструє, що проблема авторського права та ШІ виходить далеко за межі звичайного плагіату в текстовому контенті, охоплюючи й інші сфери творчості [3].

На тлі зростаючого тиску з боку медіа та регуляторів, деякі компанії, як-от OpenAI, шукають компромісні рішення. Вони укладають угоди з медіакомпаніями, щоб отримати доступ до їхнього контенту для навчання ШІ-моделей.

Хоча такі угоди можуть дати тимчасове заспокоєння, вони не вирішують корінних проблем, пов'язаних з авторським правом та ШІ. Відсутність чітких правил та норм у цій сфері призводить до розбіжностей у трактуванні ситуації судами різних країн. Це, своєю чергою, створює невизначеність для учасників ринку, стримуючи інновації та розвиток технологій.

Найактивнішими учасниками судових позовів проти компаній, що розробляють генеративні моделі ШІ, стають медіа та журналісти. Їхня головна претензія полягає в тому, що при навчанні ШІ-моделей використовувався їхній контент без належного дозволу.

Однак ці претензії є лише частиною ширшого набору вимог до технологічних компаній. Американське видання стверджує, що OpenAI і Microsoft видаляють дані про журналістів із відповідей, створених чат-ботами. Крім того, газети звинуватили компанії в репутаційній шкоді, спричиненій «ілюзіями» ШІ. Позов звинувачує ChatGPT у фабрикації статті в The Denver Post про дослідження, які нібито доводять, що куріння може вилікувати астму.

Судовий процес навколо авторського права та штучного інтелекту виходить за рамки звинувачень у використанні ЗМІ їх вмісту. У всіх сферах, де ШІ може створювати контент, виникають звинувачення в плагіаті та незаконному використанні продуктів. Це включає зображення, музику, поезію, прозу та інший вміст.

Отже, наразі судова практика будь-якої країни не може дати однозначної відповіді, чи можуть продукти, створені штучним інтелектом, захищатися

авторським правом. Значною мірою спроби різних країн інтерпретувати такі випадки зводяться до того, що контент, створений виключно штучним інтелектом, не може бути об'єктом авторського права.

Список використаних джерел:

1. The New York Times подала в суд на OpenAI та Microsoft за порушення авторських прав. URL: https://www.ukrinform.ua/rubric-technology/3805618-the-new-york-times-podala-v-sud-na-openai-ta-microsoft-za-porusenna-avtorskih-prav.html.
2. Орос Юрій. OpenAI звинуватила The New York Times у зломі ChatGPT із метою подати позов про порушення авторських прав. URL: https://itc.ua/ua/novini/openai-zvynuvatyla-the-new-york-times-u-zlomi-chatgpt-iz-metoyu-podaty-pozov-pro-porushennya-avtorskyh-prav/
3. Матяш Андрій. Війна за мільярди: проблема авторського права та ШІ вийшла за рамки звичайного плагіату. URL: https://www.epravda.com.ua/publications/2024/05/7/713178/

Ключові слова: авторське право, штучний інтелект, судовий розгляд, The New York Times ,OpenAI

Keywords: copyright, artificial intelligence, litigation, The New York Times, OpenAI

Науковий керівник: доцент Р. Чанишев

ПРИМУСОВЕ ВІДДІЛЕННЯ TIKTOK ВІД КИТАЙСЬКОЇ BYTEDANCE

Очеретнюк Каріна

студентка 1-го курсу факультету прокуратури та слідства
Національного університету «Одеська юридична академія»

У світі сучасних технологій і соціальних медіа зміни корпоративних стратегій можуть викликати значні резонанси. Одним з останніх важливих рухів на цій арені стало відділення популярної платформи TikTok від свого батьківського конгломерату ByteDance. Цей крок відбувся в умовах гострого політичного тиску з боку Сполучених Штатів Америки, що породило ряд питань щодо майбутнього TikTok, його користувачів і впливу на глобальну кіберкультуру [1].

Як наслідок TikTok та її китайська материнська компанія ByteDance подали позов до федерального суду США з метою заблокувати Закон, підписаний президентом Байденом. Цей Закон вимагає від ByteDance продати свою частку в TikTok протягом року або заборонити використання цієї соціальної мережі в США, якою користується близько 170 мільйонів американців. Захисників TikTok обурює той факт, що Закон обмежує доступ користувачів до популярної платформи, порушуючи їх права за Першим поправкою Конституції США [2].

Одним із ключових аспектів цього відділення є питання прозорості та безпеки даних. Обурення з боку США стосовно можливого використання TikTok для збору даних про своїх користувачів та їхніх побутових звичок стало провідним мотивом вимагання відділення платформи від ByteDance. Це відображає загальний тренд у світі технологій, де влада та споживачі стають все більше обізнаними щодо приватності даних та контролю над ними. Отже, відділення TikTok може слугувати як крок у напрямку більшої прозорості та відповідальності у цій сфері.

З іншого боку, це рішення також породжує певні виклики для самої компанії. ByteDance втрачає одне з найбільш прибуткових і популярних підрозділів, що може вплинути на її фінансові показники та репутацію. Крім того, існує ризик, що відділення TikTok може спровокувати інші країни або ринки до аналогічних дій, що загрожує бізнес-моделі ByteDance у цілому.

З точки зору користувачів, відділення TikTok може мати як позитивні, так і негативні наслідки. З одного боку, це може збільшити довіру користувачів до платформи, оскільки вони можуть бути менш обурені щодо потенційного втручання у їх приватність [3]. З іншого боку, це може призвести до нестабільності у вмісті та функціях, які користувачі звикли бачити на платформі, оскільки відділення може вплинути на стратегію розвитку та ринкову конкурентоспроможність TikTok.

Стратегічна важливість TikTok, яка здобула популярність у всьому світі, особливо серед молоді, робить його ключовим інструментом впливу на глобальну культуру та мислення. Таким чином, зміна власності та управління може мати вплив на те, як змінюється взаємодія країн у сфері цифрових технологій та відповідних регуляторних питань.

Одним з можливих наслідків цього відділення може бути зміна стратегії розвитку TikTok як глобальної платформи. ByteDance могла б бути більш зацікавлена в розвитку платформи в контексті глобального ринку, зосереджуючись на інтеграції з іншими своїми продуктами та послугами. Однак з відділенням TikTok може виникнути можливість для платформи розвиватися самостійно, зосереджуючись на конкретних ринках та різноманітних вимогах користувачів у кожному регіоні.

Крім того, це відділення може стати прецедентом для подібних дій у майбутньому. Інші країни можуть використовувати цей приклад для регулювання своїх власних технологічних гігантів або для захисту своїх національних інтересів у кіберпросторі. Це може призвести до зростання протекціонізму та

розбіжностей у глобальній технологічній сфері, що потенційно може загрожувати інноваціям та міжнародному співробітництву.

У кінцевому підсумку, відділення TikTok від ByteDance має далекосяжні наслідки як для корпоративного світу, так і для глобальної політики та культурного ландшафту. Воно відображає складність взаємозв'язків між технологічними компаніями та державними регуляторами, а також наголошує на необхідності узгодженого підходу до регулювання цифрових платформ та захисту приватності користувачів. Таким чином, подальший розвиток цього сценарію буде важливим для майбутньої динаміки глобального цифрового простору.

Відділення TikTok від ByteDance представляє собою складний рух, який має потенційно значний вплив на індустрію соціальних медіа, корпоративну політику та споживчу довіру. Хоча це може бути першим кроком у вирішенні проблем приватності даних та регуляторної дії, воно також відкриває двері для нових викликів та можливих змін у ландшафті цифрових комунікацій. Тим не менш, незалежно від його наслідків, відділення TikTok є ключовим моментом у розвитку і регулюванні глобальної цифрової екосистеми.

Відділення TikTok від ByteDance стало ключовим етапом у розвитку і регулюванні глобальної цифрової екосистеми. Цей рух має потенційно значний вплив на індустрію соціальних медіа, корпоративну політику та споживчу довіру. На фоні світових технологічних та соціальних змін, таке рішення відображає загальні тенденції у сфері приватності даних, контролю за ними та взаємодії між корпораціями та державними регуляторами [4].

Цей крок може слугувати як першим кроком у вирішенні проблем приватності даних та регуляторної дії в індустрії соціальних медіа. Проте він також відкриває двері для нових викликів та можливих змін у ландшафті цифрових комунікацій.

Крім того, відділення TikTok може мати важливі геополітичні наслідки, викликаючи реакції з боку інших країн та ринків. Це може спровокувати зростання протекціонізму та розбіжностей у глобальній технологічній сфері, що загрожує інноваціям та міжнародному співробітництву.

Отже, відділення TikTok від ByteDance є ключовим моментом у розвитку цифрової екосистеми. Його наслідки відчуватимуться як на корпоративному, так і на політичному рівні, підкреслюючи необхідність узгодженого підходу до регулювання цифрових платформ та захисту приватності користувачів.

Список використаних джерел:

1. В Сенаті США підтримали заборону TikTok. URL: https://minfin.com.ua/ua/2024/04/24/125616965/.
2. Павлиш Олексій. TikTok оскаржує в суді Закон США, який вимагає продажу або заборони платформи. URL: https://www.epravda.com.ua/news/2024/05/8/713414.
3. Русанов Андрій. Канадські шкільні ради подають до суду на TikTok, Meta та SnapChat — платформи заважають дітям жити та навчатися. URL: https://itc.ua/ua/novini/kanadski-shkilni-rady-podayut-do-sudu-na-tiktok-meta-ta-snapchat.
4. Карпусь Вадим. Генеральний директор TikTok закликає користувачів у США «захищати свої конституційні права» (і платформу заразом) URL: https://itc.ua/ua/novini/generalnyj-dyrektor-tiktok-zaklykaye-korystuvachiv-u-ssha-zahyshhaty-svoyi-konstytutsijni-prava.

Ключові слова: захист даних користувачів, глобальні тенденції у сфері цифрових технологій та медіа, правові та регуляторні вимоги, геополітичні аспекти

Keywords: user data protection, global trends in digital technologies and media, legal and regulatory requirements, geopolitical aspects

Науковий керівник: доцент Р. Чанишев

ЗАКОН ЄВРОСОЮЗУ «ПРО ЦИФРОВІ ПОСЛУГИ (DSA)» ТА ЗАКОН УКРАЇНИ «ПРО ЦИФРОВИЙ КОНТЕНТ ТА ЦИФРОВІ ПОСЛУГИ»

Перерва Єлизавета

студентка 1-го курсу факультету судового та міжнародного права Національного університету «Одеська юридична академія»

Закон «Про цифрові послуги (DSA)» був повністю введений в дію в Європейському Союзі 17 лютого 2024 року. Цей Закон є новим зведенням правил, спрямованих на створення безпечного та відповідального онлайн-середовища для європейців [1]. Він регулює діяльність онлайн-платформ, зокрема соціальних мереж, торговельних майданчиків та пошукових систем. Закон встановлює обов'язки для платформ щодо боротьби з незаконним контентом, захисту неповнолітніх та інших важливих аспектів.

Перший раз його презентували ще у грудні 2020 року. У зв'язку із стрімким ростом цифрової сфери, поширенням недостовірної інформації та появою нових платформ виникла потреба в контролі й уведенні жорстких правил. Основний акцент Закону спрямований на зміну умов для великих технологічних гігантів на ринку, які мають понад 45 мільйонів унікальних користувачів щомісяця.

Закон «Про цифрові послуги» суттєво вплине на кожну соціальну мережу, пошукову систему та онлайн-ринок, які здійснюють свою діяльність на території ЄС. Інший основний акцент DSA полягає в захисті користувачів Інтернету через встановлення «нового стандарту» для онлайн-платформ, щоб компанії, такі, як Google, Meta (Facebook) несли відповідальність за незаконний та шкідливий контент. Крім того, DSA зобов'язує онлайн-платформи розкривати інформацію про функціонування їхніх алгоритмів, впроваджувати процеси швидкого вилучення незаконних товарів та контенту, а також боротися з користувачами, які поширюють дезінформацію [2].

У сучасному світі цифрові технології перетворилися на неодмінну складову сфери комунікацій, бізнесу, освіти та розваг. Швидкий розвиток інтернету, мобільних пристроїв та програмного забезпечення змінив наше розуміння про можливості спілкування, доступу до інформації та здійснення різноманітних операцій. Однак із зростанням впливу цифрового середовища виникають нові виклики, пов'язані з безпекою, приватністю та доступністю цифрових послуг. У зв'язку з цим уряди багатьох країн активно працюють над розробкою та впровадженням законодавства, що регулюватиме цифровий простір. Закон "Про цифровий контент та цифрові послуги" є однією з ключових ініціатив у цьому напряму. Він має на меті встановити стандарти та вимоги до провайдерів цифрових послуг, захистити права користувачів та забезпечити безпеку персональних даних. Правильне впровадження цього Закону визначатиме не лише майбутнє цифрової економіки, а й забезпечить захист інтересів суспільства в цифровому просторі.

В інтернеті ми постійно залишаємо свої особисті дані, від електронної пошти до історії перегляду веб-сторінок. Регулювання цифрового простору включає в себе заходи для захисту цієї особистої інформації від несанкціонованого доступу та зловживання. Цифрове регулювання також забезпечує безпеку мережі та інфраструктури. Це включає заходи для запобігання кібератак, захисту від вірусів та зловмисного програмного забезпечення, а також забезпечення цілісності мережі та надійності зв'язку. Це також передбачає регулювання реклами, захист від шахрайства та обману, а також забезпечення доступності та якості цифрових послуг. Правильне регулювання цифрового простору може стимулювати інновації та розвиток нових технологій. Ідеться про сприяння конкуренції на ринку, заохочення інвестицій у цифрові технології, а також розвиток підприємництва та створення нових робочих місць.

Щодо основних положень Закону, що діють в Україні, можна сказати, що цей Закон так само встановлює правові відносини між постачальником та споживачем щодо надання цифрового вмісту і послуг. Він передбачає, що надання цифрового вмісту або послуг відбувається через укладання договору, який може бути платним, безкоштовним або на обмін персональними даними. Однією із новел цього Закону є введення об'єктивних та суб'єктивних критеріїв якості цифрового вмісту і послуг. Якщо ці критерії не виконані, споживач має право вимагати іншого вмісту, знижки у протягом 14 днів або розірвати договір. Закон також посилює захист прав споживачів цифрового вмісту і послуг, покладаючи цю відповідальність на Держспоживслужбу [3] .

Закон України «Про цифровий контент та цифрові послуги» має чіткий перелік об'єктів, які є предметом його регулювання. Закон «Про віртуальні активи» стосується операцій, пов'язаних з цими активами, і не включає конкретних видів віртуальних активів, за винятком окремих характеристик, таких, як забезпеченість чи незабезпеченість. У той час як Закон «Про цифровий контент та цифрові послуги» визначає, що регульовані операції стосуються комп'ютерних програм, застосунків, відео-, аудіо- та музичних файлів, цифрових ігор та електронних книг. Таким чином, системне тлумачення співвідношення цих двох законів вказує на те, що Закон «Про віртуальні активи» є особливим або спеціальним відносно Закону України «Про цифровий контент та цифрові послуги»[4].

Слід зауважити, що дія цього Закону не охоплює регулювання відносин щодо: надання цифрового контенту та/або цифрових послуг, які входять до складу товарів з цифровими елементами або пов'язані з такими товарами і надаються разом з ними за договором купівлі-продажу, незалежно від того, чи здійснюється надання цифрового контенту та/або цифрової послуги продавцем таких товарів чи третьою особою. Цифровий контент та/або цифрова послуга вважаються такими, що надаються у складі товару з цифровими елементами або є пов'язаними з таким товаром, якщо інше не прямо передбачено в договорі купівлі-продажу такого товару, а також якщо товар може використовуватися за своїм призначенням і без відповідного цифрового контенту та/або цифрової послуги, за умови, що споживач укладає окремий договір про надання такого цифрового контенту та/або цифрової послуги, який не є частиною договору купівлі-продажу товару з цифровими елементами.

Вплив Закону на цифрову економіку та інновації може бути значним і має багатоаспектний характер. Закон «Про цифрові послуги» може стимулювати інновації, сприяючи розвитку нових технологій та цифрових послуг. При

належному регулюванні цифрового простору створюються умови для розвитку стартапів, дослідницьких проектів та технологічних інновацій. Законодавче регулювання може сприяти розвитку цифрових платформ та екосистем, що об'єднують різні галузі та гравців на ринку. Це сприяє ефективнішому використанню ресурсів, забезпеченню конкурентоспроможності та стимулюванню інновацій. Надійне законодавство про цифрові послуги може стати ключовим фактором для інвесторів, які шукають стабільність та прозорість у цифровому ринковому середовищі. Це може сприяти залученню інвестицій у цифрові технології та інноваційні проекти. Ефективне регулювання цифрового простору може забезпечити захист прав інтелектуальної власності, що сприяє розвитку інновацій та стимулює творчість в цифровій сфері. Це важливо для залучення інвестицій та розвитку нових технологій.

Насамкінець можна зазначити, що Закон «Про цифровий контент та цифрові послуги» відіграє ключову роль у формуванні сучасного цифрового простору, впливаючи на цифрову економіку та стимулюючи інновації. Правильне регулювання цього сегмента суспільства не лише забезпечує захист прав та інтересів користувачів, але й створює сприятливі умови для розвитку бізнесу, наукових досліджень та технологічних інновацій. Через прийняття і впровадження такого закону спостерігається збільшення інвестицій у цифрові технології, створення нових робочих місць та підвищення конкурентоспроможності компаній на міжнародному ринку. Такий підхід допомагає забезпечити стабільний розвиток цифрової економіки та сприяє досягненню інноваційного прориву у різних сферах життя. Отже, Закон визначає не лише правила гри у цифровому просторі, але й сприяє створенню умов для просування суспільства в майбутнє, де цифрові технології відіграють все більш важливу роль. Тільки шляхом впровадження ефективного та прозорого законодавства можна забезпечити розвиток інноваційного, динамічного та безпечного цифрового середовища для всіх.

Список використаних джерел:

1. Свистуха Дар'я. Для онлайн-платформ Євросоюзу почав діяти Закон про цифрові послуги. Що він передбачає? URL: http://surl.li/trrjm.
2. Сомова Ольга. Закон про цифрові послуги: що потрібно знати? Ключові пункти та тези. URL: https://web-promo.ua/ua/blog/zakon-o-czifrovyh-uslugah-chto-nuzhno-znat-klyuchevye-punkty-i-tezisy/
3. Про цифровий контент та цифрові послуги: Закон України від 10.08.2023 №3321-IX. Відомості Верховної Ради України. 2023. №90. Ст. 345.

4. Загнітко Олег. Огляд Закону України «Про цифровий контент та цифрові послуги» URL: https://jurliga.ligazakon.net/analitycs/223718_oglyad-zakonu-ukrani-pro-tsifroviy-kontent-ta-tsifrov-poslugi.

Ключові слова: міжнародне співробітництво та взаємодія в сфері регулювання цифрових послуг, регулювання онлайн-платформ та цифрових сервісів, законодавчий контекст цифрових послуг

Keywords: international cooperation and interaction in the field of digital services regulation, regulation of online platforms and digital services, legislative context of digital services

Науковий керівник: доцент Р. Чанишев

ЩОДО РОЗВИТКУ ВІДКРИТИХ ДАНИХ В УКРАЇНІ

Печенюк Маргарита

студентка 1-го курсу факультету прокуратури та слідства Національного університету «Одеська юридична академія»

Розвиток відкритих даних в Україні в останні роки був помітним. Державні органи поступово почали активно публікувати дані у відкритому форматі, що сприяє більшій прозорості, доступності та інноваціям. Відкриті дані використовуються для розробки різноманітних додатків та сервісів, а також для аналізу та прийняття рішень у сферах економіки, охорони здоров'я, освіти, транспорту та інших галузях.

Урядові програми та ініціативи сприяють розвитку відкритих даних в країні. Наприклад, «Прозорість та доступність даних уряду» є однією з таких програм, яка спрямована на створення умов для вільного доступу до даних, їх аналізу та використання [1].

Також важливою складовою розвитку відкритих даних є залучення громадськості. Активна участь громадськості в процесі використання відкритих даних сприяє створенню нових інноваційних рішень та підвищенню ефективності використання даних у суспільстві.

Важливо відзначити, що розвиток відкритих даних в Україні сприяє також залученню громадськості до процесів прийняття рішень та формуванню громадського контролю за діяльністю влади. Це відкриває нові можливості для розвитку громадського діалогу, взаємодії між владою та громадянами, а також підвищення рівня довіри до державних інституцій.

Крім того, розвиток відкритих даних є важливим інструментом для стимулювання інноваційного підприємництва та розвитку IT-сектору. Доступ до

відкритих даних сприяє появі нових продуктів та послуг, розвитку нових технологій та підвищенню конкурентоспроможності українських компаній на світовому ринку.

Ініціативи з розвитку відкритих даних також сприяють підвищенню ефективності державного управління та забезпеченню більшої відповідальності органів влади перед громадянами. Широкий доступ до даних про діяльність владних структур дозволяє громадськості вивчати та аналізувати інформацію, що сприяє виявленню корупційних схем, неефективного використання бюджетних коштів та інших проблем.

Під час заходу Open Data Day 2024 міністр цифрової трансформації України Михайло Федоров представив своє бачення розвитку відкритих даних у країні. За заявою Мінцифри, відкриті дані є основою для ефективної та прозорої роботи держави. Україна, навіть перебуваючи у стані війни, продовжує розвивати цю сферу і посідає третє місце в Європі за рівнем зрілості відкритих даних. Стратегія розвитку відкритих даних на найближчі роки включає в себе затвердження національної стратегії, зміни в законодавстві, а також оновлення національного порталу відкритих даних [2].

Бачення Мінцифри щодо розвитку відкритих даних в Україні відображає важливість цього напряму для суспільного прогресу та транспарентності влади. Відкриті дані – це не просто технологічна ініціатива, це новий спосіб мислення, який змінює взаємовідносини між громадянами, державою та бізнесом.

Основна ідея відкритих даних полягає в тому, щоб зробити доступними та зрозумілими для всіх дані, які належать громаді. Це може бути інформація про бюджет, статистика, географічні дані, результати досліджень та багато іншого. Мета полягає в тому, щоб забезпечити прозорість управління, сприяти інноваціям у різних галузях та стимулювати розвиток громадянського суспільства [3].

Важливим аспектом бачення Мінцифри є розвиток інфраструктури для збирання, зберігання та обробки даних. Це включає в себе створення централізованих платформ, стандартизацію даних та захист персональної інформації. Подальша робота над вдосконаленням цих процесів дозволить забезпечити ефективний та безпечний доступ до даних для всіх зацікавлених сторін.

Другим важливим напрямом є популяризація відкритих даних серед громадян та бізнесу. Це може включати проведення освітніх кампаній, організацію хакатонів та створення інструментів для аналізу даних. Чим більше людей буде використовувати відкриті дані, тим більше вони будуть сприяти інноваціям та розвитку суспільства [4].

Не можна не зазначити важливість співпраці між різними секторами суспільства – урядом, бізнесом та громадянським суспільством. Тільки шляхом спільних зусиль можна досягти максимального ефекту від впровадження відкритих даних.

Бачення Мінцифри щодо розвитку відкритих даних в Україні є кроком у майбутнє. Відкриті дані можуть стати потужним інструментом для забезпечення ефективного управління, стимулювання інновацій та підвищення рівня довіри громадян до влади. Тому важливо продовжувати працювати у цьому напряму, розвивати інфраструктуру, популяризувати відкриті дані серед громадян та забезпечувати їх доступність для всіх зацікавлених сторін.

Також важливо зазначити, що розвиток відкритих даних може відігравати ключову роль у боротьбі з корупцією та підвищенні ефективності управління. Через доступність і прозорість інформації про діяльність урядових структур, громадяни та журналісти можуть більш ефективно контролювати дії влади та виявляти випадки недопустимих дій.

Крім того, розвиток відкритих даних створює сприятливі умови для розвитку інновацій та створення нових продуктів та послуг. Бізнес може використовувати ці дані для аналізу ринку, прогнозування тенденцій та розробки нових продуктів, що в свою чергу сприяє економічному зростанню та підвищенню конкурентоспроможності країни в цілому.

Однак для досягнення повного потенціалу відкритих даних необхідно вирішувати ряд проблем, таких як захист персональних даних, забезпечення якості та достовірності інформації, а також забезпечення доступності для всіх шарів суспільства, включаючи людей з обмеженими можливостями та малозабезпечені верстви населення.

Бачення Мінцифри щодо розвитку відкритих даних в Україні відображає важливість цього напрямку для створення прозорого, ефективного та інноваційного суспільства. Продовження роботи у цьому напрямку буде сприяти розвитку країни, забезпеченню довіри громадян до влади та підвищенню їхнього життєвого рівня.

Зважаючи на це, важливо продовжувати розвивати та вдосконалювати інфраструктуру відкритих даних, забезпечуючи їхню безпеку, доступність та якість. Також потрібно активно працювати над усуненням перешкод у шляху впровадження відкритих даних, зокрема, розробляти ефективні механізми захисту особистої інформації та забезпечення відповідності стандартам безпеки даних.

Необхідно проводити постійну освітню роботу серед громадян, бізнесу та урядових структур щодо важливості та переваг відкритих даних. Це може включати проведення тренінгів, семінарів, конференцій та інших подій, спрямованих на підвищення освіченості щодо використання та користування відкритими даними.

Розвиток відкритих даних є стратегічним напрямком для подальшого розвитку України як сучасної, транспарентної та інноваційної країни. Шляхом спільних зусиль уряду, бізнесу та громадянського суспільства можна досягти значних успіхів у використанні відкритих даних для покращення управління, стимулювання економічного розвитку та підвищення якості життя населення.

Розвиток відкритих даних в Україні в останні роки виявився помітним і перспективним напрямом. Активна публікація даних у відкритому форматі державними органами сприяє більшій прозорості та доступності інформації, стимулює інновації та сприяє ефективному управлінню. Урядові програми, такі як «Прозорість та доступність даних уряду», ініціюють створення умов для вільного доступу до даних та їх аналізу. Громадська участь використанні відкритих даних сприяє новаторським рішенням та підвищує довіру до влади.

Розвиток відкритих даних також стимулює інноваційне підприємництво та розвиток IT-сектору, що сприяє економічному зростанню та конкурентоспроможності. Заходи щодо розвитку інфраструктури та популяризації відкритих даних серед громадян та бізнесу є важливими для досягнення максимального ефекту. Співпраця між урядом, бізнесом та громадськістю є ключовою для успішного розвитку відкритих даних.

Бачення Мінцифри про розвиток відкритих даних в Україні свідчить про важливість цього напряму для створення прозорого, ефективного та інноваційного суспільства. Продовження роботи у цьому напряму буде сприяти розвитку країни та підвищенню її життєвого рівня. Однак важливо вирішувати проблеми з захистом персональних даних, якістю і достовірністю інформації, а також забезпечувати доступність для всіх верств суспільства.

Список використаних джерел:

1. Бочарова Ю.Г., Чернега О.Б., Кожухова Т.В. Діджиталізація та цифрові трансформації в ЄС. Економіка і організація управління. 2021. №2 (42). С. 6–19. URL: https://jeou.donnu.edu.ua/article/view/11026/10925.
2. Карпусь Вадим Мінцифри представило бачення розвитку відкритих даних в Україні. URL: https://itc.ua/ua/novini/mintsifry-predstavylo-bachennya-rozvytku-vidkrytyh-danyh-v-ukrayini/.

3. Цифровізація економіки України: трансформаційний потенціал: монографія /В.П. Вишневський, О.М. Гаркушенко, С.І. Князєв, Д.В. Липницький, В.Д. Чекіна; за ред. В.П. Вишневського та С.І. Князєва; НАН України, Інститут економіки промисловості. Київ: Академперіодика, 2020. 188 с.
4. Мельник Л.Г., Карінцева О.І., Кубатко О.В., Сотник І.М., Завдов'єва Ю.М. Цифровізація економічних систем та людський капітал: підприємство, регіон, народне господарство. Механізм регулювання економіки. 2020. №2. С. 9–28.

Ключові слова: відкриті дані, майбутні перспективи та стратегічні цілі, прозорість, відкритість та відповідальність, ініціативи Мінцифри України

Keywords: open data, future prospects and strategic goals, transparency, openness and responsibility, initiatives of the Ministry of Digital Transformation of Ukraine

Науковий керівник: доцент Р. Чанишев

ЗАКОН ФРАНЦІЇ ПРО ЗРОСТАННЯ ТА ТРАНСФОРМАЦІЮ ПІДПРИЄМСТВ (ЗАКОН «РАСТЕ»): ЗМІСТ І ЗНАЧЕННЯ

Почтаренко Катерина

студентка 1-го курсу факультету судового та міжнародного права Національного університету «Одеська юридична академія»

Закон «Pacte» є важливим кроком для стимулювання економічного зростання та підтримки підприємництва у Франції. Цей Закон містить у собі низку положень, спрямованих на зменшення адміністративних обмежень для бізнесу, заохочення інвестицій у стартапи та молоді технологічні компанії, а також зміцнення корпоративного управління. Враховуючи глобальні виклики та потреби сучасного ринку, Закон «Pacte» відображає спробу уряду Франції створити сприятливе середовище для бізнесу та сприяти його конкурентоспроможності.

Якщо брати коротко з історії, то цей Закон, офіційно названий «Loi pour la croissance et la transformation des entreprises» (Закон про зростання та трансформацію підприємств), був прийнятий у Франції 22 травня 2019 року [1].

Цей Закон став одним із наймасштабніших реформ французької економіки за останні десятиліття, маючи на меті стимулювати інвестиції, підтримку підприємництва та інновації, а також зробити країну більш конкурентоспроможною на світовому ринку. Він був розроблений урядом Франції під керівництвом прем'єр-міністра Едуара Філіппа, а сам проєкт Закону був представлений на розгляд парламенту в лютому 2019 року і після декількох місяців бурхливих дебатів був прийнятий 22 травня 2019 року [2].

Потреба у прийнятті Закону «Pacte» виникла внаслідок низки факторів, які протягом багатьох років негативно впливали на французьку економіку:

Низький рівень інвестицій: рівень інвестицій у Франції був нижчим, ніж у багатьох інших більш розвинених країнах, що обмежувало зростання та створення нових робочих місць.

Складні адміністративні процедури: підприємці у Франції стикалися з численними бюрократичними бар'єрами, що ускладнювало ведення бізнесу та стримувало інновації.

Нерозвинений ринок цінних паперів: французький ринок цінних паперів був менш розвиненим, ніж у інших країнах, що ускладнювало доступ до капіталу для підприємств, які прагнуть розширюватися.

Слабка підтримка стартапів: екосистема підтримки стартапів у Франції була недостатньо розвиненою, що обмежувало можливості для нових та інноваційних підприємств. Закон «Pacte» містить низку положень, спрямованих на стимулювання інвестицій, підтримку підприємництва та інновацій, а також на покращення конкурентоспроможності французької економіки.

До ключових положень Закону належать:

– зниження податків: Закон «Pacte» передбачає ряд податкових пільг для підприємств та акціонерів, спрямованих на стимулювання інвестицій та створення нових робочих місць;

– спрощення адміністративних процедур: Закон «Pacte» значно спрощує адміністративні процедури для підприємств, роблячи ведення бізнесу більш гнучким та динамічним;

– розвиток ринку цінних паперів: Закон «Pacte» містить ряд заходів, спрямованих на розвиток французького ринку цінних паперів, роблячи його більш привабливим для інвесторів;

– підтримка стартапів: Закон «Pacte» створює сприятливе середовище для розвитку стартапів, надаючи їм доступ до фінансування, менторської підтримки та інших ресурсів;

– цифровізація державних послуг: Закон «Pacte» передбачає перехід до онлайн-надання державних послуг, роблячи їх більш зручними та доступними для громадян.

Очікується, що Закон «Pacte» матиме значний позитивний вплив на французьку економіку, де одними з очікуваних результатів будуть зростання інвестицій, створення нових робочих місць, підвищення конкурентоспроможності та зростання ВВП.

Якщо брати саме інформатику та нові можливості щодо цього, то зростання попиту на цифрові послуги, підтримка розвитку штучного інтелекту та збільшення доступу до інвестицій мають важливу роль, адже цифровізація державних послуг, розвиток електронної комерції та підтримка стартапів призведуть до значного зростання попиту на цифрові послуги. Це відкриє нові можливості для IT-компаній, які зможуть запропонувати інноваційні рішення та продукти [3].

Інвестиції ж у дослідження та розробки штучного інтелекту, а також підготовка кадрів у цій сфері створитимуть сприятливе середовище для розвитку цієї галузі. IT-компанії, які зможуть успішно впровадити штучний інтелект у свою діяльність, отримають значну конкурентну перевагу. А розвиток ринку цінних паперів та підтримка стартапів полегшить IT-компаніям доступ до капіталу, що дозволить їм реалізовувати амбітні проекти та розширювати свою діяльність.

Окрім нових можливостей, потрібно враховувати виклики. Зростання конкуренції, необхідність адаптації до нових умов, етичні аспекти використання штучного інтелекту, ці вади можуть призвести до посилення конкуренції на IT-ринку. IT-компаніям буде необхідно постійно вдосконалювати свої продукти та послуги, щоб залишатися конкурентоспроможними. IT-компаніям та фахівцям буде необхідно адаптуватися до нових умов, які виникають внаслідок впровадження закону «Pacte» [4].

Це може потребувати освоєння нових навичок, зміни методів роботи та трансформації бізнес-моделей. Розвиток штучного інтелекту викликає низку етичних питань, які необхідно вирішити. IT-компаніям буде необхідно чітко визначити принципи використання штучного інтелекту та забезпечити його етичне і відповідальне застосування.

Важливо зазначити, що Закон «Pacte» створює не лише нові можливості, але й виклики для IT-компаній та фахівців. Ті, хто зможуть успішно адаптуватися до нових умов та скористатися новими можливостями, зможуть досягти значного успіху. Як видається, Закон «Pacte» в цілому утворить зростання кількості робочих місць в IT-секторі, збільшенняя IT-сектора може стимулювати розвиток інших галузей економіки та може зробити Францію більш привабливою для IT-фахівців з усього світу.

Список використаних джерел:

1. LOI n° 2019-486 du 22 mai 2019 relative à la croissance et la transformation des entreprises NOR: ECOT1810669L JORF n°0119 du 23 mai 2019 URL: https://www.legifrance.gouv.fr/jorf/id/JORFTEXT000038496102.

2. La loi PACTE: pour la croissance et la transformation des entreprises URL: https://www.economie.gouv.fr/loi-pacte-croissance-transformation-entreprises.
3. Comprendre la loi pacte, censée «transformer les entreprises» URL: https://www.lemonde.fr/economie-francaise/video/2018/10/05/comprendre-la-loi-pacte-censee-transformer-les-entreprises_5365297_1656968.html.
4. Thibault de Ravel d'Esclapon Loi PACTE: simplifier la vie de l'entreprise URL: https://www.actu-juridique.fr/affaires/loi-pacte-simplifier-la-vie-de-lentreprise/.

Ключові слова: законодавство Франції, Закон «Pacte», стимулювання підприємництва, інтеграція цифрових технологій у стратегії розвитку підприємств, трансформація та модернізація галузей промисловості

Keywords: French legislation, Laws «Pacte», stimulating entrepreneurship, integrating digital technologies into enterprise development strategies, transforming and modernising industries

Науковий керівник: доцент Р. Чанишев

ПРАВОВИЙ СТАТУС EULA В УКРАЇНІ

Сорочан Анна

студентка 1-го курсу факультету судового та міжнародного права Національного університету «Одеська юридична академія»

В умовах швидкого розвитку інформаційних технологій та зростання популярності програмного забезпечення виникає потреба в регулюванні відносин між користувачами і розробниками програмного забезпечення. Одним з основних документів, що регулюють ці відносини, є EULA (End-User License Agreement) або Ліцензійна угода з кінцевим користувачем.

EULA є юридичним документом, який встановлює правила використання програмного забезпечення, обмеження щодо його розповсюдження та відповідальність сторін у разі порушення умов угоди. Українське законодавство не має окремого закону, який би регулював EULA безпосередньо. Проте, воно забезпечує деякий рівень захисту прав інтересів користувачів програмного забезпечення через загальні норми цивільного та авторського права.

Закон України «Про авторське право та суміжні права» встановлює права та обов'язки сторін у сфері авторського права. Відповідно до цього Закону, програмне забезпечення, як об'єкт авторського права, захищене від несанкціонованого використання, копіювання та розповсюдження. Оскільки EULA визначає умови використання програмного забезпечення, вона може бути використана для установлення обмежень на такі дії [1].

Крім того, Господарський кодекс України регулює умови укладення договорів і зобов'язання сторін. EULA може бути розглянута як один із видів договору, укладеного між розробником та кінцевим користувачем, де встановлюються умови використання програмного забезпечення та відповідальність за їх порушення.

Незважаючи на відсутність прямого регулювання EULA в українському законодавстві, судова практика свідчить про те, що угоди, у тому числі і EULA, мають силу договору між сторонами, якщо вони відповідають загальним принципам дії договорів і не суперечать чинному законодавству.

Додатково до вищезазначеного, варто звернутися до міжнародних стандартів та практик у сфері EULA. Багато з розробників програмного забезпечення, особливо тих, які діють на міжнародному ринку, використовують загальноприйняті зразки EULA, які часто базуються на законодавстві країни, де вони зареєстровані. Наприклад, багато EULA містять пункти, що обмежують відповідальність розробника за можливі збитки, включаючи втрату даних чи переривання роботи. Ці пункти можуть базуватися на міжнародних стандартах і практиці, яка визначає обсяг відповідальності розробника програмного забезпечення за його використання [2].

У деяких випадках EULA можуть містити пункти, які обмежують права користувачів, наприклад, щодо перерозподілу чи зміни програмного коду. Ці пункти можуть бути суперечливими у контексті окремих юрисдикцій, включаючи Україну, де існує принцип вільного доступу до інформації та програмного забезпечення.

Також важливо врахувати, що умови EULA можуть бути піддані змінам з часом. Розробники можуть випускати оновлення програмного забезпечення та змінювати умови використання через оновлення EULA. У таких випадках користувачі можуть бути зобов'язані приймати нові умови перед використанням програмного забезпечення.

Крім того, важливо зазначити, що укладення EULA може бути складним процесом для звичайних користувачів, оскільки ці угоди часто містять складні юридичні терміни та умови, які можуть бути непересічними для неюридичної аудиторії. Це може створювати ситуацію, коли користувачі приймають умови, не повністю розуміючи їх зміст та наслідки.

У цьому контексті важливою є роль державних органів у захисті прав інтересів користувачів. Уряд може розглядати можливості регулювання EULA для забезпечення більшої прозорості та захисту прав споживачів. Наприклад, це може

включати вимоги щодо чіткого формулювання умов угоди та обов'язкового надання користувачам можливості ознайомитися з умовами до укладання угоди.

Зрештою, в контексті швидкого розвитку цифрової економіки та інформаційного суспільства важливо постійно переглядати та адаптувати законодавство для відповідності новим викликам та технологічним тенденціям. Це може включати не лише розгляд правового статусу EULA, але й створення нових правових інструментів для регулювання відносин у сфері цифрової торгівлі та забезпечення захисту прав інтернет-користувачів.

Насамперед, важливо врахувати вплив Європейського Союзу на правовий статус EULA в Україні. Україна прагне гармонізувати своє законодавство з європейськими стандартами, зокрема у сфері цифрового права та захисту споживачів. Європейські директиви та регламенти можуть впливати на вимоги до умов EULA та їх захисту. Також слід звернути увагу на роль міжнародних організацій та угод у регулюванні EULA. Наприклад, Всесвітня організація інтелектуальної власності (ВОІВ) може розробляти рекомендації та стандарти у сфері ліцензійних угод для програмного забезпечення, які впливають на практику укладання EULA у всьому світі [3].

У контексті застосування EULA до хмарних послуг та онлайн-сервісів важливо розглянути такі аспекти, як захист персональних даних користувачів та відповідальність за їх обробку. Законодавство України щодо захисту персональних даних (зокрема, Закон "Про захист персональних даних") має важливе значення у контексті укладення та виконання EULA для онлайн-сервісів.

Крім того, варто розглянути практику інших країн у регулюванні EULA та вирішенні спорів, пов'язаних з їх виконанням. Аналіз судової практики у Сполучених Штатах, Європейському Союзі та інших країнах може допомогти у визначенні кращих практик та прецедентів, на яких можна буде спиратися при вирішенні суперечок українським судам. Також важливо розглянути аспекти взаємодії EULA з іншими видами договорів та законодавством, зокрема у сфері електронної комерції. Укладання EULA може супроводжуватися іншими договорами, такими як угоди про конфіденційність, угоди про обробку персональних даних тощо. Важливо враховувати взаємозв'язок між цими документами та їх вплив на правовий статус та обов'язки сторін.

Потрібно також звернути увагу на аспекти захисту споживачів у контексті укладання EULA. У багатьох країнах, включаючи Україну, існують законодавчі акти та органи, що забезпечують захист прав споживачів у випадку недотримання умов угоди. Правовий статус EULA може бути підданий контролю та

регулюванню органами захисту прав споживачів з метою запобігання недобросовісним практикам та забезпечення справедливих умов для користувачів.

Значною мірою важливим є аспект врегулювання спорів, які виникають між сторонами в процесі виконання EULA. Можливість вирішення конфліктів шляхом альтернативних методів врегулювання спорів, таких, як медіація чи арбітраж, може бути визначальною для швидкого та ефективного вирішення суперечок між розробниками та користувачами програмного забезпечення.

У контексті швидкого розвитку технологій та цифрової економіки важливо також враховувати можливість виникнення нових форм програмного забезпечення та відповідно нових видів угод, які можуть впливати на правовий статус EULA. Наприклад, угоди про використання штучного інтелекту чи блокчейн-технологій потребують нових підходів до регулювання та захисту прав сторін.

Правовий статус EULA в Україні є невизначеним, що породжує ризики та проблеми для користувачів і розробників програмного забезпечення. Відсутність чіткого регулювання цих угод українським законодавством може призвести до різних тлумачень та спірних ситуацій між сторонами, а також порушень прав споживачів. Для вирішення цих проблем необхідно провести подальше законодавче регулювання та розробити спеціальне законодавство, яке б чітко визначало правовий статус та умови EULA в українському правовому просторі. Такий крок дозволить створити ефективну та збалансовану систему регулювання цього виду угод, що сприятиме розвитку цифрової економіки країни та забезпечить захист прав всіх зацікавлених сторін.

Список використаних джерел:

1. Про авторське право і суміжні права: Закон України від 01.12.2022 № 2811-IX Відомості Верховної Ради України. 2023. №57. Ст. 166. URL: https://zakon.rada.gov.ua/laws/show/2811-20/print (дата звернення: 09.05.2024).
2. End User License Agreement або Угода з кінцевим користувачем. Legal Support. URL: https://legal-support.top/end-user-license-agreement/ (дата звернення: 09.05.2024).
3. Навіщо потрібен EULA (End-user license agreement)? - Legarithm. Legarithm. URL: https://legarithm.io/news/navishho-potriben-eula-end-user-license-agreement/ (дата звернення: 09.05.2024).

Ключові слова: ліцензійна угода, правовий статус, українське законодавство, авторське право, ліцензування програмного забезпечення, права та обов'язки користувачів, регулювання цифрової сфери

Keywords: license agreement, legal status, Ukrainian legislation, copyright, software licensing, rights and responsibilities of users, regulation of the digital sphere

Науковий керівник: доцент Р. Чанишев

ВИМОГИ СИСТЕМИ ПРОТИДІЇ ВІДМИВАННЮ ГРОШЕЙ (AML) СТАНОМ НА 2024 РІК

Томащук Артем

студент 1-го курсу факультету прокуратури та слідства
Національного університету «Одеська юридична академія»

Дотримання вимог систем протидії відмиванню грошей (Anti-Money Laundering, AML) та фінансуванню тероризму (Combating the Financing of Terrorism, CFT) є ключовим завданням для всіх країн у світі.

У 2024 році ця проблема набула ще більшої актуальності через швидкі зміни в фінансовій технологічній сфері, зростання транскордонних фінансових операцій та розвиток криптовалют. Глобальний погляд на дотримання вимог AML у 2024 році показує, що в цьому напряму вже були зроблені значні кроки, але залишається багато викликів, які потребують уваги.

По-перше, важливо відзначити зростання ролі технологій у виявленні та запобіганні відмиванню грошей. У 2024 році були впроваджені нові інноваційні інструменти, такі, як штучний інтелект, блокчейн та аналіз великих даних, щоб ефективно виявляти підозрілі фінансові транзакції та ідентифікувати особи, які їх здійснюють. Це допомагає зменшити ризик фінансових злочинів і забезпечити більшу безпеку фінансової системи.

По-друге, у 2024 році спостерігається збільшення співпраці між країнами у сфері боротьби з відмиванням грошей. Оскільки багато фінансових операцій мають транскордонний характер, ефективне виявлення та припинення таких операцій вимагає спільних зусиль різних країн. Міжнародні організації, такі, як Financial Action Task Force (FATF), активно сприяють цій співпраці, розробляючи стандарти та рекомендації для держав у сфері AML та CFT.

По-третє, у зв'язку з розвитком криптовалют, у 2024 році було звернено особливу увагу на регулювання цього сектору з метою запобігання використанню криптовалют для нелегальних цілей. Країни по всьому світу розробляють та впроваджують нові законодавчі акти та регулятивні механізми для контролю за криптовалютними платежами та обміном криптовалют.

Незважаючи на ці позитивні тенденції, викликів у сфері AML у 2024 році також достатньо. Одним з головних викликів є необхідність постійного оновлення технологій та методів виявлення фінансових злочинів, оскільки злочинці постійно шукають нові способи обходу заходів безпеки. Крім того, необхідно зміцнювати співпрацю між країнами, особливо в тих регіонах, де існують значні ризики AML та CFT [1].

Досягнення успіху в боротьбі з відмиванням грошей та фінансуванням тероризму також потребує широкої підтримки від громадськості та приватного сектору. У 2024 році спостерігається зростання усвідомленості серед громадськості щодо важливості AML та CFT. Багато неприбуткових організацій, активісти та освітні програми працюють над підвищенням свідомості та навчанням громадян про ризики фінансових злочинів.

Приватний сектор, зокрема фінансові установи, також відіграє ключову роль у боротьбі з відмиванням грошей. Банки, фінансові компанії та інші установи повинні дотримуватися вимог AML та CFT, проводити обов'язкову перевірку клієнтів та здійснювати моніторинг фінансових транзакцій. У 2024 році багато компаній вдосконалюють свої системи та процедури, використовуючи нові технології та кращі практики, щоб забезпечити ефективний контроль над фінансовими ризиками [2].

Важливо зазначити, що успішна боротьба з відмиванням грошей та фінансуванням тероризму потребує стійкого політичного зобов'язання з боку урядів по всьому світу. Країни повинні приділяти достатню увагу розробці та виконанню ефективного законодавства, сприяти співпраці між правоохоронними органами та забезпечувати належне фінансування для проведення розслідувань та заходів з протидії AML та CFT.

Потрібно враховувати також роль міжнародних організацій у зміцненні боротьби з відмиванням грошей та фінансуванням тероризму. Організації, такі як ООН, Європейський Союз, Міжнародний валютний фонд (МВФ) та інші, активно сприяють співпраці між країнами та розвитку міжнародних стандартів у сфері AML та CFT.

У 2024 році виникають нові виклики, такі, як збільшення використання криптовалют та анонімних платіжних систем для фінансування тероризму та злочинності. Це вимагає від урядів та регуляторів по всьому світу швидкого реагування та розвитку нових стратегій для виявлення та запобігання таким видам фінансових злочинів.

Наприклад, у 2024 році багато країн почали розглядати можливість впровадження централізованих державних криптовалют або цифрових валют, які можуть бути контрольовані та моніторовані урядами з метою запобігання використанню їх для нелегальних цілей. Крім того, розвиток нових технологій для виявлення та аналізу фінансових транзакцій у реальному часі є важливим напрямком розвитку в цій сфері.

Важливо також враховувати вплив геополітичних подій та конфліктів на ефективність боротьби з AML та CFT. Наприклад, санкції проти деяких країн можуть викликати перегрупування фінансових потоків та спроби обходу санкційних обмежень через різні канали [3].

У 2024 роц, зростаюча увага до проблем AML та CFT свідчить про розуміння їх серйозності та необхідності спільних зусиль для їх вирішення. Шляхом поєднання новітніх технологій, міжнародної співпраці, розвитку регуляторного середовища та підтримки громадськості можна досягти подальшого прогресу у боротьбі з цими викликами та забезпечити стабільність світової фінансової системи [4].

Отже, глобальний погляд на дотримання вимог AML у 2024 році показує важливість спільних зусиль урядів, громадськості та приватного сектору для забезпечення безпеки та стабільності фінансової системи. Попри досягнуті успіхи, виклики лишаються значними, і тільки шляхом посилення співпраці та інновацій можна досягти подальшого прогресу у боротьбі з цими загрозами.

Список використаних джерел:

1. Anti-Money Laundering and Combating the Financing of Terrorism (AML/CFT) URL: https://www.imf.org/en/Topics/Financial-Integrity/amlcft.
2. Anti-Money Laundering: International Law and Practice" by Wouter H. Muller, Edward Elgar Publishing, 2019. P.12 URL: http://surl.li/trnlm.
3. Gaviyau William, Global Anti-Money Laundering and Combating Terrorism Financing Regulatory Framework: A Critique URL: https://www.mdpi.com/1911-8074/16/7/313.
4. Нікітіна Олександра Нові рецепти AML в ЄС. URL: https://rates.fm/ua-uk/expert-opinion/novi-recepti-aml-v-yes.

Ключові слова: технологічні інновації, регулювання та нагляд за системами AML, глобалізація, вимоги міжнародного співробітництва в боротьбі з відмиванням грошей
Keywords: technological innovations, regulation and supervision of AML systems, globalization, requirements of international cooperation in the fight against money laundering

Науковий керівник: доцент Р. Чанишев

ЩОДО ВІДПОВІДАЛЬНОСТІ АВТОВИРОБНИКІВ НА ПРИКЛАДІ СУДОВОГО ПОЗОВУ ПРОТИ TESLA ЧЕРЕЗ СМЕРТЕЛЬНУ АВАРІЮ З AUTOPILOT

Шаповал Катерина

студентка 1-го курсу факультету прокуратури та слідства
Національного університету «Одеська юридична академія»

Глибинна інтеграція штучного інтелекту до структури більшості суспільних відносин та процесів спричиняє ускладнення у категоріальному визначенні його змісту, а, отже, і у розробці оптимальних алгоритмів правового впливу на вказані процеси. Штучний інтелект (далі – ШІ) не лише стає незамінним інструментом з точки зору функціональної цінності, але повністю змінює парадигму визначення власної функціональної природи та навіть суспільної ніші. Активне застосування інформаційних систем з елементами штучного інтелекту істотним чином перетворює сучасну повсякденну реальність та формує особливий тип світовідчуття людини.

Штучний інтелект можна визначити як галузь комп'ютерної науки, що займається автоматизацією розумної поведінки [1]. Сучасна техніка, що містить ознаки штучного інтелекту, стала повсякденним засобом підвищення рівня комфорту та безпеки. Зазначені закономірності призвели до появи інформаційних систем нового типу під назвою інтелектуальні інформаційні системи, під якими розуміються автоматизовані інформаційні системи, засновані на знаннях, або комплекс програмних, логіко-математичних і лінгвістичних засобів для здійснення підтримки діяльності людини.

Разом зі збільшенням комфорту та якості життя сучасні технології штучного інтелекту порушують питання нанесення шкоди людині умисно чи ні та процесів регулювання у нормативно-правовому полі таких випадків [2, 3]. Одним з таких прикладів може стати смертельна аварія, спричинена Autopilot Tesla у 2018 році, та подальша судова справа проти компанії.

Так, у березні 2018 року Вей «Волтер» Хуан загинув після того, як його електромобіль Model X з активованою системою Autopilot врізався в захисний бар'єр уздовж шосе 101 в Маунтін-В'ю, Каліфорнія. Пізніше розслідування, проведене Національною радою з безпеки на транспорті, виявило, що аварії сприяли автопілот і відволікання від водіння, оскільки телефонні дані вказували, що Хуан грав у мобільну гру [4].

У 2019 році сім'я Хуана подала проти Tesla позов про смерть, стверджуючи, що на той час автомобіль не мав належних функцій безпеки, включаючи автоматичне гальмування.

У своєму позові родина стверджувала, що автопілот не задіяв необхідні функції безпеки, такі, як автоматичне екстрене гальмування. «Tesla проводить бета-тестування свого програмного забезпечення Autopilot на живих водіях. Сім'я Хуан хотіла захистити інших автомобілістів, які керують безпілотними або іншими напівавтономними автомобілями Tesla, від трагедії», – зазначав адвокат родини Марк Фонг [5].

Адвокати мали намір показати, що Волтера Хуанга помилково ввели в оману, буцімто він може довіряти своїй Model X їхати по шосе, не будучи готовим взяти керування на себе

Також родичі загиблого водія, раніше повідомляли, що незадовго до інциденту він вже помічав дивну поведінку автопілота на цій же ділянці траси і навіть скаржився на це офіційному дилеру Tesla.

Представник Tesla відмовився коментувати позов, пославшись на офіційну заяву компанії про смертельну аварію. У ньому йдеться, що перед аварією водій нібито не стежив за дорогою та ігнорував сигнали системи, які вимагали взяти керування. У компанії також припустили, що аварія була смертельною, тому що перед відбійником не було спеціального захисного бар'єру, який міг би пом'якшити удар.

З цього приводу варто зазначити, що раніше Tesla неодноразово стикалася з судовими позовами через свої функції допомоги водієві. Але вона жодного разу не була визнана винною. У 2021 році автовиробник виграв два позови, які звинувачували Autopilot у причетності до двох аварій: смертельній ДТП «без людей за кермом» [6] та ще одній смертельній аварії 2019-го року, в якій загинув водій Model 3 [7].

Також Tesla вже виграла суд у Лос-Анджелесі, де компанія стверджувала, що попереджує водіїв про те, що її технологія вимагає постійного моніторингу з боку людини, незважаючи на назву «Autopilot» і «Full Self-Driving». Судовий розгляд стосувався аварії, в якій автомобіль Model S виїхав на обочину і травмував водія.

Тесла продовжувала тестувати і впроваджувати системи Autopilot і більш продуктивну систему Full Self-Driving (FSD), яку генеральний директор Ілон Маск вважає критично важливою для майбутньої компанії, але яка викликає регуляторну та юридичну увагу.

Разом з тим у 2024 році федеральні регулятори США посилили перевірку Tesla, оскільки в жовтні 2023 року Міністерство юстиції розширило кримінальне розслідування щодо функцій Autopilot. Tesla також оголосила про відкликання 2 млн автомобілів Tesla та випустила оновлення, яке мало ускладнити водіям неправильне використання Autopilot. Але експерти виявили, що виправлення здебільшого не відповідають вимогам [8].

Проте у 2024 році справа про смертельну ДТП була офіційно закрита, оскільки автовиробник врегулював позов. Це сталося у той самий день, коли мав відбутися відбір присяжних. За інформацією медіа, адвокати попросили суд засекретити деталі мирової угоди, щоб точна сума, яку компанія заплатила, не стала надбанням громадськості .

Відтак, на основі проаналізованих даних судової справи проти компанії Tesla у питанні вини штучної технології Autopilot у смерті людини можна зазначити, що в майбутньому такі прецеденти можуть повторюватися. Все це матиме місце в силу інтеграції та появи новітніх автоматизованих технологій з використанням штучного інтелекту. Компанія Tesla, через свої лідерські позиції у питаннях розробки та впровадження інновацій, повинна дбати про якість своїх продуктів та безпеку їх користувачів. Всі ці аспекти розкривають широке поле для подальшого дослідження відповідальності розробників технологій на базі штучного інтелекту.

Список використаних джерел:

1. Корєєва С.Р. Теоретичні підходи до визначення поняття та правового регулювання штучного інтелекту. Науковий вісник Ужгородського Національного Університету. 2021. Вип. 66. С. 50-55.
2. Кронівець Т.М. Правові та ціннісні особливості феномену штучного інтелекту як елементу правової дійсності. Науковий вісник Херсонського державного університету. 2022. №2. С. 21-23.
3. Бортник С.М. Правове регулювання використання штучного інтелекту. Застосування інформаційних технологій у правоохоронній діяльності: матеріали Круглого столу (м. Харків, 14 грудня 2023 р.). Харків: ХНУВС, 2023. С. 11-14.
4. Карпусь В. Tesla постане перед судом через роль Autopilot у смертельній аварії 2018 року. URL: https://itc.ua/ua/novini/tesla-postane-pered-sudom-cherez-rol-autopilot-u-smertelnij-avariyi-2018-roku/
5. На Tesla подала до суду сім'я водія, який вибухнув у безпілотнику URL: http://surl.li/trlzu.
6. Даньшина Катерина Автопілот Tesla не винен у сенсаційній смертельній ДТП «без людей за кермом» — насправді водій керував авто під дією алкоголю та медикаментів. URL: http://surl.li/trmya.

7. Даньшина Катерина Автопілот Tesla не винен у смертельній аварії 2019-го року, в якій загинув водій Model 3 (ще два пасажири отримали важкі травми). URL: http://surl.li/trmzg.
8. Tesla виграє перший судовий процес в США щодо смертельної аварії з Autopilot. URL: https://autogeek.com.ua/tesla-vygraye-pershyj-sudovyj-proczes-v-ssha-shhodo-smertelnoyi-avariyi-z-autopilot.

Ключові слова: Tesla, автопілот, Model X
Keywords: Tesla, Autopilot, Model X

Науковий керівник: доцент Р. Чанишев

АНАЛІЗ МЕТОДІВ ПОСТОБРОБЛЕННЯ ПОТОКІВ ВИПАДКОВИХ ЧИСЕЛОВИХ

Щербина Юрій

кандидат технічних наук, доцент, доцент кафедри інформаційних технологій Національного університету «Одеська юридична академія»

Казакова Надія

доктор технічних наук, професор, професор кафедри кібербезпеки Національного університету «Одеська юридична академія»

Однією з проблем сучасного програмування є створення генераторів псевдовипадкових чисел, що використовуються в таких областях як аналіз, моделювання та криптографія. Її вирішення до цих пір залишається достатньо складною задачею. Особливо це стосується криптографії. На даний момент створено багато ефективних і стійких шифрів. Але для забезпечення надійного захисту необхідно мати ще й ефективну систему генерації та розподілення ключів [1]. А для цього потрібні справжні рівномірно розподілені бінарні послідовності. Що ж стосується випадкових чисел для потреб моделювання, то тут єдиною вимогою є рівномірність розподілення ймовірностей чисел на виході генератора, але ця вимога ставиться достатньо жорстко. Будь-яка нерівномірність на виході джерела випадковості автоматично переноситься на процес, що моделюється на його основі.

Генератори випадкових чисел ділять на генератори псевдовипадкових чисел (ГПВЧ або pseudorandom number generators – PRNG), і генератори справжніх випадкових чисел (ГСВЧ або true random number generators – TRNG).

ГПВЧ – це детермінований генератор, який утворює числа алгоритмічним способом, що виглядають як випадкові. Такі числа повністю визначаються своїм

початковим значенням і використовуються в криптографічних системах та моделюванні.

Оскільки міжсимвольна кореляція на виході ГСВЧ майже відсутня, саме їх використовують для формування ключів до шифрів в системах управління ключовими даними [2].

Загальна схема генератора справжніх випадкових чисел, де джерелом випадковості з реальний фізичний процес, наведена на рисунку 1.

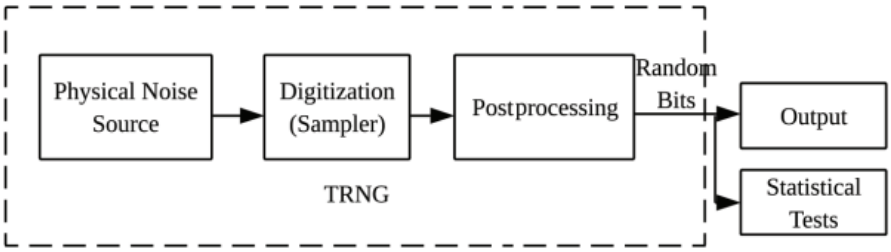


Рисунок 1 – Складові частини ГСВЧ

Цей генератор містить джерело фізичного шуму (physical noise source), що дає аналоговий випадковий сигнал, блок дискретизації (digitization), в якому відбувається оцифровування сигналу та перетворення його на послідовність бітових символів, а також засоби виконання постпроцесінгу (postprocessing) – додаткового оброблення вихідної послідовності з метою усунення зміщення. Під зміщенням *p* розуміється відхилення ймовірностей нульових і одиничних символів у вихідній послідовності від величини 0.5

$$b = \frac{|p_1 - p_0|}{2},$$

де p_0, – це ймовірність нульового символу у складі вихідної послідовності, а p_1 – одиночного символу.

Причиною використання постоброблення справжніх випадкових послідовностей, отриманих від фізичних джерел, є те, що фізичні джерела, зазвичай, вкрай нестабільні і величина бітового зміщення в процесі роботи може змінюватись. Постоброблення передбачає екстракцію з вихідного потоку тих символів, що "вписуються" у рівномірне розподілення ймовірностей. Зрозуміло, що частина вихідного потоку втрачається і це є головним недоліком алгоритмів, що реалізують такий спосіб отримання справжніх випадкових чисел.

На даний момент, найбільш поширеними є наступні алгоритми постоброблення.

Корректор XOR. Алгоритм, що передбачає розділення вхідного потоку на послідовні *n*-бітні групи символів та їх складання по модулю два. Якщо кількість одиниць в групі парна, на вихід видається нуль, у протилежному випадку –

одиниця. Усунення зміщення обходиться зменшенням продуктивності джерела випадковості в *n* раз. Позитивною стороною коректора є забезпечення постійної швидкості генерації символів.

Корректор фон Неймана. Алгоритм фон Неймана передбачає розділення вхідного потоку на пари символів і, якщо ці символи однакові (00, 11), то на вихід видається 0, у протилежному випадку (01, 10) – 1. Такий коректор наближає ентропію потоку до 1, але швидкість вихідного потоку повністю визначається джерелом випадкового фізичного процесу і, таким чином, не є постійною. Вона може складати до 1/4 від швидкості вхідного потоку.

Функція H. Цей алгоритм для усуненням зміщення було запропоновано Дихтлем [3]. Він заснований на перетворені квазігрупової послідовності.

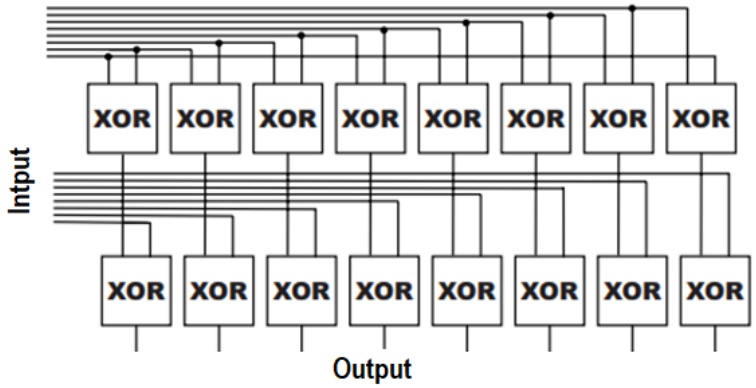


Рисунок 2 – H-функція для постоброблення

Схему алгоритму, закладеного в H-функцію, показано на рисунку 3. Він передбачає перетворення кожних 16 вхідних символів вхідної послідовності у 8 біт вихідного потоку за наступними правилами. Вхідний блок $a_0, a_1, \dots, a_{15}$ перетворюється у вихідний блок $b_0, b_1, \dots, b_7$, де $b_i = a_i \oplus a_{(i+1) \bmod 8}$. Слід зауважити, що відображення $a_0, a_1, \dots, a_7$ в $b_0, b_1, \dots, b_7$ не є бієктивним. Існує тільки 128 варіантів таких комбінацій. Таким чином, на виході H-функції спостерігаються символи, створені за правилом $c_i = b_i \oplus a_{i+8}$.

Наведені вище способи усунення зміщення рівномірності є досить простими з точки зору кількості обчислювальних ресурсів, що витрачаються на їх реалізацію. Інші, більш складні способи постоброблення, здебільшого, базуються на використанні хешування, яке значно знижує продуктивність обраного джерела випадковості.

Сама ідея постоброблення передбачає використання фізичного джерела випадкового процесу, оскільки бітові символи з його виходу не мають між собою аналітичної залежності. Але джерела такого типу, зазвичай, нестабільні і рівномірність символів на їх виході має помітне зміщення. Через це, екстракція

залишається єдиним способом рандомізації вихідного потоку [4].

Виникає питання: чи можна застосувати пост оброблення для генераторів псевдовипадкових чисел? Очевидно, що так робити можна, наприклад, коли такий генератор використовують для потреб моделювання, де міжсимвольна кореляція не так важлива. Правда, в такому випадку, можна просто удосконалити алгоритм самого ГПВЧ. Тут треба шукати компромісне рішення, враховуючи швидкість генерації випадкових чисел, розмір обчислювальних ресурсів та відсутність міжсимвольної кореляції.

Список використаних джерел:

1. Bruce Schneier. Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd Edition 662. Pages 1995.
2. Avaroglu E., Tuncer T. A novel S-box-based postprocessing method for true random number generation / E.Avaroglu, T. Tuncer // Turk J Elec Eng & Comp Sci. – 2020. – №28. – P. 288-301. URL: https://journals.tubitak.gov.tr/cgi/viewcontent.cgi?article=1431&context=elektrik.
3. Dichtl M. Bad and good ways of post-processing biased physical random numbers. In: Biryukov A (editor). Fast Software Encryption. FSE 2007. Lecture Notes in Computer Science, vol 4593. Berlin, Germany: Springer, 2007. pp. 137–152. URL : https://link.springer.com/content/pdf/10.1007/978-3-540-74619-5_9.pdf.
4. John von Neumann Various Techniques Used in Connection with Random Digits. J. Res. Nat. Bur. Stand. Appl. Math. Series 3, 36-38 (1951). URL : https://mcnp.lanl.gov/pdf_files/InBook_Computing_1961_Neumann_JohnVonNeumannCollectedWorks_VariousTechniquesUsedinConnectionwithRandomDigits.pdf

Ключові слова: Генератори випадкових чисел, моделювання, алгоритм, криптографія, рандомізація, постоброблення

Keywords: Random number generators, modeling, algorithm, cryptography, randomization, post-processing

ПРИКЛАД ЗАСТОСУВАННЯ АЛГОРИТМУ КРУСКАЛА ДЛЯ ОПТИМІЗАЦІЇ ЕЛЕКТРИЧНИХ МЕРЕЖ

Шапіро Дмитро

студент 2-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Побудова магістральних ліній електропередач (ЛЕП) потребує дві важливі складові: час та ресурси. Наприклад, вартість прокладки високонавантаженої лінії електропередачі в США становить приблизно 3,1-4,5 млн доларів за кілометр

згідно з проєктом [1]. Вартість прокладки залежить від використаних технологій під час проєктування, матеріалів та місцевості, тому такі роботи мають бути точно прораховані та належним чином обґрунтовані.

Як приклад можливої оптимізації ЛЕП вибрано алгоритм Крускала. Цей алгоритм часто використовується для побудови бюджетних з'єднань мереж, планування кабелів та усунення менш прибуткових сполучень зі збереженням працездатності усієї мережі. Незалежно від сфери діяльності, цей алгоритм дозволяє оптимізувати будь-яке підключення [2].

Метою алгоритму Крускала є побудова дерева, яке з'єднує всі вершини графа, використовуючи мінімально можливу суму ваг ребер. Побудова мінімального кістякового дерева відбувається через початкове сортування усіх вершин графа за вагою у новому масиві. Далі відбувається вибір двох вершин з найменшою вагою та їх перевірка на належність до спільного дерева. Якщо вони не належать до нього, вони об'єднуються, інакше алгоритм іде далі [3]. Результаті отримано підграф без циклів з такою самою кількістю вершин та ребрами з мінімальною сумою ваг [3].

Функція обчислення мінімального кістякового дерева по алгоритму Крускала в JavaScript має вигляд:

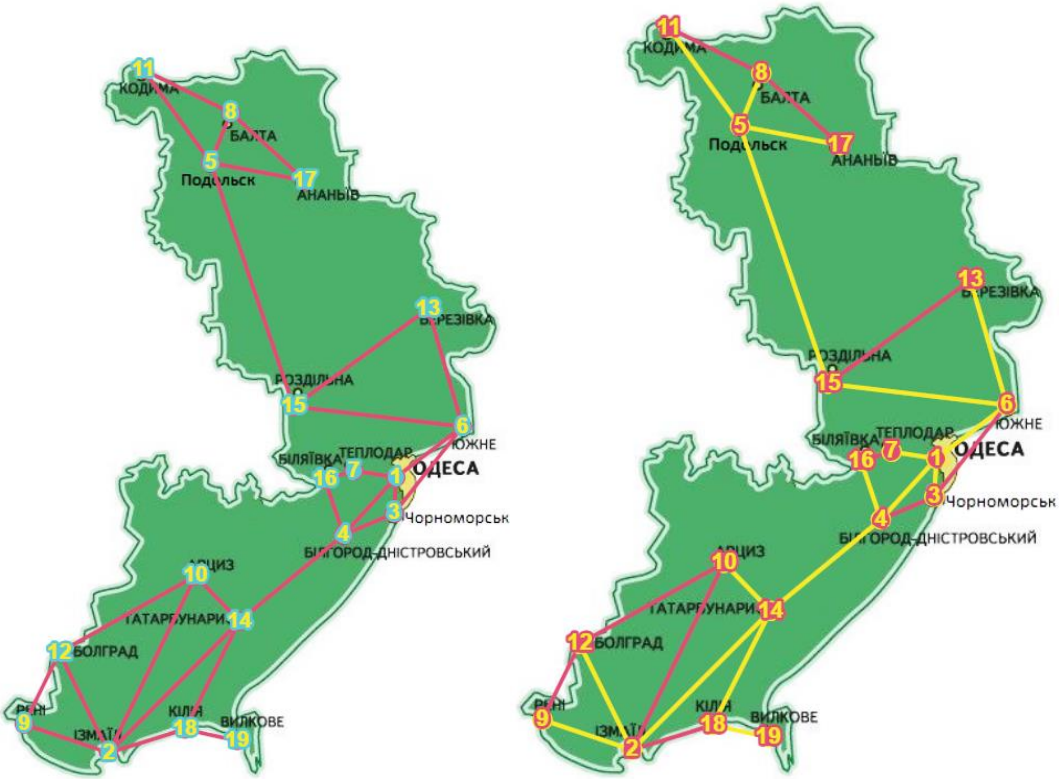
```
// Побудова мінімального кістякового дерева
function kruskal(graph) {
  let disjointSet = new DisjointSet(graph.length);
  let edges = [];
  let result = [];

  for (let i = 0; i < graph.length; i++) {
    for (let j of graph[i].connectedWith) {
      edges.push([i, j - 1, getRandomValue(10)]);
    }
  }
  edges.sort((a, b) => a[2] - b[2]);
  for (let [u, v, weight] of edges) {
    if (disjointSet.find(u) !== disjointSet.find(v)) {
      disjointSet.union(u, v);
      result.push({ from: u + 1, to: v + 1, weight });
    }
  }
  return result;
}
```

Додатково використано структуру даних DisjointSet. Вона має на меті розбиття усіх елементів на неперетинні підмножини з подальшим об'єднанням двох підмножин, а також пошук цих елементів у множинах. Ця структура даних використовується саме при створенні мінімального кістякового дерева. І вона відіграє ключову роль в алгоритмі Крускала, бо поліпшує часову складність алгоритму до O(E * log V) де (V – множина вершин, E – множина ребер) [4].

Клас, який презентує структуру даних DisjointSet в JavaScript, має вигляд:

```
class DisjointSet {
  find(x) {
    if (this.parent[x] !== x)
      this.parent[x] = this.find(this.parent[x]);
    return this.parent[x];
  }
  union(x, y) {
    let rootX = this.find(x);
    let rootY = this.find(y);
    if (rootX !== rootY) {
      if (this.rank[rootX] > this.rank[rootY]) {
        this.parent[rootY] = rootX;
      } else if (this.rank[rootX] < this.rank[rootY]) {
        this.parent[rootX] = rootY;
      } else {
        this.parent[rootY] = rootX;
        this.rank[rootX]++;
       }
    }
  }
}
```

У дослідженні взято карту Одеської області з основними містами районного значення, на якій було побудовано ЛЕП з'єднання між містами у вигляді списків суміжності. Дистанція між містами становить 1. У нашому прикладі вхідні дані ані ЛЕП, ані їх з'єднання не відповідають дійсності і є лише демонстраційними, для перевірки дієвості алгоритму в ході дослідження. На рис. 1, а наведено карту з уявними магістральними лініями.



а) б)

Рисунок 1 – Граф дослідження з умовною прив'язкою до місцевості:

а) візуалізація зв'язків у графі на карті;

б) побудоване мінімальне кістякове дерево

Після побудови графа проведено його перетворення в мінімальне кістякове дерево. Після цього список суміжності перетворено у список ребер з вершинами та вагою ребра (рис. 2, б).

Алгоритм Крускала визначив основні центри електропередачі, з яких буде найдоцільніше проводити ЛЕП, при цьому зберігши наявну енергосистему з можливістю подальшого розширення по містах.

Переваги застосування алгоритму Крускала:

– *практичне застосування:* даний алгоритм підходить для різнопланових задач логістики, будівництва мереж різних сфер та концентрації основних потужностей у визначених місцях;

– *простота реалізації:* є можливість модифікації алгоритму Крускала через початковий поділ масиву вершин на дві рівноподільні частини за середнім значенням, через що буде покращена швидкість дії алгоритму тільки за нормальних умов [5];

– *гнучкість сортування ребер:* крім основного сортування ребер, можна додати критерії для відбору ребер перед початком обчислення.

Недоліки алгоритму Крускала:

– *не підходить для щільних графів:* графи з великою кількістю ребер будуть обчислюватися значно повільніше, бо алгоритм потребує первинного сортування ребер за вагою, на відміну від алгоритму Прима;

– *ребра з однаковою вагою:* у таких ситуаціях буде вибиратися довільне ребро кожного разу, що може відрізнятися від бажаного результату;

– *використання пам'яті:* через використання структури даних disjoint Set, алгоритм потребує додаткову пам'ять для зберігання вершин.

Властивості мінімальних кістякових дерев можуть дійсно змінити вже наявні мережі і системи, наприклад електромережі, інтернет-мережі, газові та інші трубопроводи тощо. Їх оптимізація – це ключ для подальшого вдосконалення, підтримки та масштабування.

Список використаних джерел:

1. Minnesota high-voltage 180-mile power line cost. URL: https://www.mprnews.org/story/2023/08/04/proposed-minnesota-highvoltage-180mile-power-line-could-cost-1b
2. Kruskal Algorithm: Definition and Purpose. URL: https://datascientest.com/en/kruskal-algorithm-definition-and-purpose
3. Minimum Spanning Tree. URL: https://yuminlee2.medium.com/kruskals-algorithm-minimum-spanning-tree-db96e91d0aed
4. Vynnychuk S. Analysis steps of the time complexity of Kruskal algorithm with sorted scales with work time O(E + VLogV) for its C-implementation. Collection "Information Technology and Security". 2012. Vol. 1(2). P. 10–13. DOI: https://doi.org/10.20535/2411-1031.2012.1.2.54120
5. Haiming Li, Qiyang Xia, Yong Wang. Research and Improvement of Kruskal Algorithm. Journal of Computer and Communications. 2017. Vol. 5 No.12. DOI: 10.4236/jcc.2017.512007.
6. Jiayu Chen. The analysis and application of Prim algorithm, Kruskal algorithm, Boruvka algorithm. ACE. 2023. Vol. 19. P. 84-89. DOI: 10.54254/2755-2721/19/20231012.

Ключові слова: алгоритм Крускала, комунікації, розробка
Keywords: Kruskal Algorithm, Communications, development

Науковий керівник: доцент О. Трофименко

СОЦІАЛЬНО-ЕКОНОМІЧНІ АСПЕКТИ ВИКОРИСТАННЯ ЦЕНТРІВ ОБРОБКИ ДАНИХ

Чанишев Рашид

кандидат юридичних наук, доцент, доцент кафедри інформаційних технологій Національного університету «Одеська юридична академія»

Повсюдне впровадження хмарних технологій і швидкий розвиток систем штучного інтелекту призвело до масового будівництва центрів обробки даних (далі -ЦОД) по всьому світу. За даними міжнародної консалтингової компанії JLL, попит на послуги ЦОД перебуває на рекордно високому рівні, водночас темпи будівництва ЦОД досягли максимуму у 2023 році, а 83 відсотки об'єктів, які будують, уже здано в попередню оренду. За словами Карла Бітса (Karl Beets), старшого менеджера компанії JLL, «розвиток центрів оброблення даних поширюватиметься скрізь, де є достатньо енергії та доступної землі» [1].

Якщо спочатку ЦОД намагалися розміщувати в тих географічних регіонах, де були достатні електрогенерувальні потужності та можна було забезпечити ефективне відведення тепла (наприклад, у північних країнах, таких, як Норвегія, Ісландія і т. ін. регіонах), то на теперішній час ЦОД будують скрізь, де є для цього фізична можливість.

Підвищений попит на послуги ЦОД пов'язаний зі швидким розвитком систем штучного інтелекту, наприклад, таких, як ChatGPT. Однак усі ці послуги збільшують потребу не тільки в ЦОД, а й в електроенергії. За даними JLL, уже в листопаді 2023 року в ChatGPT було 100 мільйонів активних користувачів, при цьому ця система щодня споживала стільки ж електроенергії, скільки споживає 33 000 домогосподарств.

За даними Synergy Research Group, до кінця 2026 року в експлуатації буде близько 1200 ЦОД. З них 53 % ЦОД розташовані в США, приблизно 16 % ЦОД розташовані в Європі, 11 % - у країнах Азіатсько-Тихоокеанського регіону (APAC), 15 % - у Китаї, і тільки 5 % - у решті світу [2]. При цьому, за оцінками фахівців, тільки в одній Європі необхідно побудувати 3000 нових ЦОД.

Одній тільки великій трійці хмарних провайдерів (для позначення таких компаній зазвичай використовується термін «гіперскейлери», від «hyperscaler») - компаніям Amazon, Google і Microsoft належить понад 400 великих ЦОД.

Більше третини ЦОД у США розташовані в одному штаті - Північній Вірджинії. ЦОД у цьому штаті більше, ніж у Європі та Китаї. Така концентрація ЦОД в одному штаті пов'язана з особливостями законодавства штату, що надають значні податкові пільги власникам ЦОД.

Несподіваним наслідком такої концентрації ЦОД в обмеженому регіоні стали соціальні проблеми - домовласники і громадські активісти округу Прінс-Вільям (Prince William County) в Північній Вірджинії (США) стали подавати масові скарги на «катастрофічний» шум, що видається місцевими ЦОД, які належать компанії Amazon. За словами активістів, безперервний шум спричинений цілодобовою роботою систем повітряного охолодження на дахах ЦОД, щодо усунення якого не вживають жодних заходів. Водночас місцеві жителі скаржаться на проблеми, що виникли не тільки з їхнім здоров'ям, а й зі здоров'ям домашніх тварин. Аналогічні скарги в Аризоні 2018 року призвели до рішення місцевої влади заборонити будівництво нових ЦОД у цьому штаті [3].

Після тривалих протестів жителів Північної Вірджинії компанія Amazon була змушена розробити заходи щодо зниження шуму від систем вентиляції [4].

Проблему поглиблює той факт, що використовувати для ЦОД водяне охолодження стало важко, оскільки наявна інфраструктура міст не призначена для перероблення великої кількості стоків, що надходять від ЦОД із водяним охолодженням.

Так, компанія Microsoft була змушена відмовитися від водяного охолодження своїх нових ЦОД у місті Фенікс (США) через те, що інфраструктура цього міста не може переробляти більше стічних вод і очищати стоки з дата-центрів компанії до безпечного для людей стану. Microsoft і влада міста домовилися про те, що всі майбутні центри обробки даних, будівництво яких заплановано в цьому місті, матимуть повітряне охолодження [5].

Однак, як було зазначено вище, використання повітряного охолодження може призвести до масових протестів місцевого населення.

З огляду на ступінь важливості проблеми, компанія Microsoft провела експеримент із розміщення ЦОД на дні моря (UDC). Цей експеримент виявився вдалим, і 2023 року китайська компанія Highlander побудувала перший повноцінний ЦОД, повністю розміщений на дні моря [6].

Розміщення ЦОД під водою дає змогу одночасно розв'язати дві проблеми - з охолодженням ЦОД (охолодження здійснюється морською водою) і зі зниженням рівня шуму. Однак залишається незрозумілим, як підводні ЦОД впливатимуть на екологію, якщо таких ЦОД буде побудовано багато.

Крім того, існує варіант будівництва плаваючих ЦОД (обладнання яких розміщується на спеціальних баржах, розташованих біля берега моря). Плавучі або підводні ЦОД досить мобільні, що дає змогу розгорнути їх у будь-якій точці світу, де є необхідна інфраструктура (електропостачання та підводні кабелі зв'язку).

Платити за купівлю або оренду землі (а це дуже велика стаття витрат) при цьому теж не потрібно.

Існують і більш екзотичні проекти розміщення ЦОД - наприклад, проекти розміщення ЦОД на орбітальних космічних станціях [7].

Але все ж основною проблемою, пов'язаною з ЦОД, є проблема нестачі електроенергії. За останній час потужність великих ЦОД збільшилася від 30 МВт до 60-90 МВт, і в перспективі вона буде тільки зростати. Одні тільки ЦОД компанії Apple у 2023 році витратили понад 2,3 TW-h електроенергії, а сумарна потужність ЦОД компанії Microsoft перевищила потужність, споживану такою країною, як Португалія.

Ця проблема має системний характер і є спільною для всіх регіонів розміщення ЦОД - і в США, і в Європі, і в Азії.

Так, у Великій Британії, де найближчим часом планується будівництво близько 300 нових ЦОД, у 2022 году вибухнув скандал після того, як з'ясувалося, що будівництво житла в Західному Лондоні призупинено, оскільки вся доступна на місцевих електропідстанціях потужність була зарезервована для ЦОД.

Крім того, проблема посилюється тим, що в багатьох країнах законодавство вимагає переходу від традиційних способів виробництва електроенергії до використання поновлюваних джерел енергії.

Так, у 2022 році з'явилися перші ЦОД, що використовують сонячні фотоелектричні панелі. Використання таких панелей передбачається і для плавучих ЦОД. Того ж року компанією Plug Power для ЦОД Microsoft було створено систему резервного енергопостачання, що використовує водневі паливні комірки сумарною потужністю в 1,5 МВт.

Як один із варіантів розв'язання проблеми розглядають варіант використання малих модульних ядерних реакторів (SMR), що розміщуються в безпосередній близькості від майданчиків ЦОД. SMR на швидких нейтронах мають потужність до 15 МВт кожен і здатні працювати до 10 років без перезарядки, що дасть змогу ЦОД працювати без підключення до магістральних електромереж [8].

Крім того, подібні рішення просуваються як «екологічні», оскільки сприяють зниженню викидів вуглецю в атмосферу.

Використання SMR може вирішити проблему з електропостачанням ЦОД, але при цьому з'являються значні ризики, пов'язані з проблемою забезпечення безпеки таких ядерних реакторів. До того ж масове будівництво SMR майже неминуче призвести до масових протестів місцевого населення, при цьому до

таких протестів буде залучено набагато більшу кількість людей, ніж це було в Північній Вірджинії.

Загострення пристрастей може дещо знизити використання тепла, що виробляється ЦОД, для централізованого опалення житлових будинків. Оскільки це тепло є «відходом виробництва», вартість такого опалення може бути дуже низькою, аж до безкоштовної. Наприклад, компанія Microsoft планує у 2025-2026 роках створити систему, що дає змогу опалювати понад 6000 житлових будинків в околицях Копенгагена [9]. Аналогічний проект Microsoft реалізується і в Гельсінкі. За повідомленням Datacenter Dynamics, і уряд Великої Британії виділив £36 млн ($44,5 млн) на модернізацію системи центрального опалення Західного Лондона, яка дасть змогу використовувати «сміттєве» тепло ЦОД для опалення 10 тис. будинків.

З огляду на все вищесказане, можна припускати, що соціально-економічні проблеми, пов'язані з використанням ЦОД, у найближчому майбутньому стрімко зростатимуть і неминуче призведуть до необхідності вжиття обмежувальних заходів на законодавчому рівні.

Список використаних джерел:

1. Kari Beets. North America Data Center Report The quest for power moves development into new frontiers. URL: http://surl.li/tumfz.
2. Virginia Still Has More Hyperscale Data Center Capacity Than Either Europe or China. URL: http://surl.li/tumkm.
3. Peter Judge. Prince William residents complain of "catastrophic noise" from data centers. URL: http://surl.li/tumkr.
4. Amazon Tones Down Its Data Center Noise After Residents Sound the Alarm. URL: http://surl.li/tummm.
5. Dan Swinhoe. Microsoft to swap water for air cooling at campus in Phoenix, Arizona. URL: http://surl.li/tumno.
6. Peter Judge. China's Highlander completes first commercial underwater data center, looks for exports. URL: http://surl.li/tuysj.
7. Ascend: Thales Alenia space to lead European feasibility study for data centers in space. URL: http://surl.li/tuytg.
8. Sebastian Moss. Sam Altman-backed nuclear reactor firm Oklo hiring data center lead as it looks to power "AI and cloud computing". URL: http://surl.li/tuyvd.
9. Peter Judge. Microsoft to warm homes with waste heat from Danish data center. URL: http://surl.li/tuyww.

Ключові слова: розвиток хмарних обчислень і штучного інтелекту, ЦОД, соціальні проблеми, економічні проблеми, екологічні проблеми, методи вирішення

Keywords: development of cloud computing and artificial intelligence, data centres, social problems, economic problems, environmental problems, methods of solution

РОЗРОБКА ВЕБСАЙТУ ФАКУЛЬТЕТУ КІБЕРБЕЗПЕКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Чикунов Павло

кандидат технічних наук, доцент, доцент кафедри інформаційних технологій Національного університету «Одеська юридична академія»

Касян Павло

студент 4-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Метою виконання дослідження є розробка, тестування та SEO-оптимізація вебсайту факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія» (ОНЮА) на платформі WordPress.

Для досягнення мети дослідження необхідно розробити функціональний, безпечний та легкій у використанні вебсайт, який відображатиме академічні стандарти ОНЮА та престиж факультету, та відповідає наступним вимогам:

1. забезпечення інтуїтивної навігації: вебсайт повинен мати чітку та зрозумілу структуру, яка дозволить легко знаходити потрібну інформацію та ресурси;

2. наявність адаптивного дизайну: вебсайт має коректно відображатися на різних пристроях, забезпечуючи комфортний перегляд як на настільних комп'ютерах, так і на гаджетах;

3. функціональність спеціалізованих модулів: інтеграція модулів, таких як системи зворотнього зв’язку, організації наукових та виховних заходів, публікація новин;

4. забезпечення достатнього рівень безпеки: застосування сучасних технологій та кращих практик для забезпечення захисту персональних даних студентів та викладачів.

5. простота управління контентом: можливість легкого оновлення та редагування контенту менеджментом факультету;

6. інтеграція з соціальними мережами та іншими онлайн-сервісами: забезпечення розповсюдження інформації та залучення аудиторії з популярних платформ.

Огляд літератури виявив кілька ключових аспектів, зокрема сучасні підходи до викладання інформаційних технологій, стратегії підготовки фахівців з кібербезпеки, ефективні методики забезпечення захисту інформаційних систем. Особлива увага приділена аналізу існуючих освітніх платформ і порівнянню їхніх особливостей та функціоналу.

Проведено огляд існуючих вебсайтів закладів вищої освіти, що дозволило визначити ефективні методи представлення інформації, наявність інтерактивних елементів, що залучають студентів та сприяють їхньому активному навчанню.

В результаті розроблено концепцію вебсайту, яка відповідає сучасним тенденціям веб-дизайну та орієнтована на кращі практики в галузі освіти та кібербезпеки. Основний упор зроблено на створенні ресурсу, який би водночас слугував освітньою платформою та вітриною факультету, підкреслюючи його репутацію та професіоналізм у сфері кібербезпеки та інформаційних технологій.

Вибір системи управління контентом (CMS) є одним із вирішальних аспектів при створенні вебсайту, оскільки від неї залежать можливості розширення функціоналу, зручність управління сайтом, а також швидкість та безпека.

WordPress виявився найпопулярнішою CMS завдяки своїй гнучкості, великій спільноті та легкості використання. Ця система є відкритою для користувачів різного рівня знань – від новачків до професійних розробників. WordPress пропонує величезний вибір плагінів та тем. Зважаючи на потреби факультету кібербезпеки та інформаційних технологій, вибір пав на WordPress через його широкі можливості персоналізації, велику кількість готових рішень для освітніх установ та зручність управління сайтом для адміністраторів і кінцевих користувачів.

Моделювання архітектури вебсайту на WordPress є ключовим етапом у процесі його розробки, оскільки забезпечує зручність і логічність навігації для кінцевих користувачів. Вибір структури вебсайту впливає на усі аспекти його функціонування, від легкості доступу до інформації до зручності управління контентом з боку адміністратора.

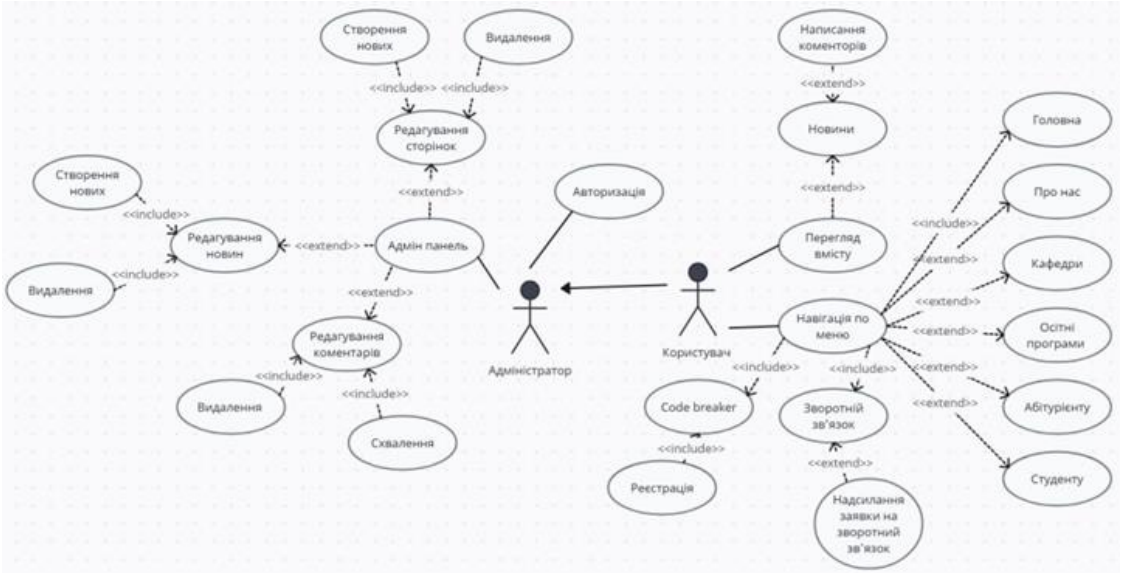


Рисунок 1 – Діаграма варіантів використання вебсайту

Забезпечення адаптивності дизайну також відіграє важливу роль у процесі моделювання архітектури. Адаптивний дизайн гарантує, що веб-сайт коректно відображається на різних пристроях, що є необхідним для забезпечення широкої доступності і комфорту використання.

Процес вибору теми для вебсайту має вирішальне значення, адже це безпосередньо впливає на загальний зовнішній вигляд, відчуття від сайту та взаємодію користувачів з ним. Після ретельного аналізу доступних опцій та врахування цілей веб-проєкту, прийняте рішення використати тему Phlox.

У процесі розробки вебсайту на WordPress одним з ключових моментів є планування використання плагінів для розширення функціоналу. Yoast SEO є одним з найпопулярніших плагінів для оптимізації веб-сайтів під пошукові системи. Він надає інструменти для керування мета-тегами, створення карти сайту, підказки щодо виправлення помилок SEO та багато іншого. Його аналітичні функції допомагають зрозуміти, як можна поліпшити текст на сторінці для кращого ранжування в пошукових системах.

Початковий етап просування здійснювався через генерацію XML-карт сайту за допомогою Yoast SEO, що сприяє швидшому виявленню та індексуванню сторінок пошуковими системами. Встановлені фокусні ключові слова для кожної сторінки, використовуючи аналітичні інструменти плагіну, які дозволяють побачити, як сторінка може ранжуватися за даним ключовим словом.

Yoast SEO автоматично аналізує читабельність контенту, надаючи рекомендації щодо поліпшення структури речень та використання проміжних заголовків для кращого розуміння тексту користувачами і пошуковими роботами. Виконана оптимізація мета-тегів тайтлів і описів, забезпечуючи їх залученість та відповідність вибраним ключовим словам.

Завдяки Yoast SEO були внесені метадані для соціальних мереж, які включають заголовки, описи та зображення, оптимізовані для Facebook та Twitter, що підвищує ефективність соціального просування.

Важливою частиною SEO-стратегії стало використання внутрішнього лінкування для покращення навігації на сайті та збільшення часу, який користувачі проводять на ресурсі. Ці заходи, у комплексі з іншими функціями Yoast SEO, сприяли поліпшенню видимості сайту в пошукових системах та зростанню органічного трафіку.

Плагін All In One WP Security обраний для забезпечення безпеки веб-сайту на WordPress. Цей плагін включає широкий спектр функцій безпеки, що покривають різні аспекти захисту сайту від потенційних загроз.

Використання All In One WP Security дозволило мінімізувати ризики безпеки сайту і створити надійний бар'єр проти багатьох типів загроз, від простих до складних атак, що забезпечує спокій як для мене, так і для користувачів сайту.

Під час тестування вебсайту виконана перевірка функціональності та користувацького інтерфейсу. Починаючи з перевірки відповідності дизайну та розміщення елементів на різних пристроях, встановлено, що веб-сайт відображається коректно та зручно як на настільних комп'ютерах, так і на мобільних пристроях. Зокрема перевірена робота різних функцій, таких як форми зв'язку, кнопки, меню навігації, анімації.

Протестована працездатність функцій вебсайту на веббраузерах Chrome, Firefox, Safari та Edge, та встановлено, що він працюють однаково ефективно. У процесі тестування перевірено час завантаження сторінок, щоб вони відкривалися швидко та без затримок навіть при підвищеному навантаженні.

Загалом, після вичерпного тестування, автори дійшли до висновку, що веб-сайт готовий до публікації та відповідає меті дослідження.

Список використаних джерел:

1. Hedengren T.D. WordPress for Beginners: A Comprehensive Guide. O'Reilly Media, USA, 2021.
2. Yoast SEO. The Definitive Guide To Higher Rankings For WordPress Sites. URL: https://yoast.com/wordpress-seo.
3. WebDevStudios. Security Best Practices for WordPress. URL: https://webdevstudios.com.
4. WPBeginner. Beginner's Guide for WordPress. URL: https://wpbeginner.com.
5. Torque. All You Need to Know About WordPress Security. URL: https://torquemag.io.

Ключові слова: ФКІТ ОНЮА, вебсайт, WordPress, плагін, Yoast SEO, All In One WP Security
Keywords: FCIT OLA, website, WordPress, plagin, Yoast SEO, All In One WP Security.

ПРОЦЕДУРНА ГЕНЕРАЦІЯ ІГРОВИХ РІВНІВ ЗАСОБАМИ ГРАФІВ

Шляхтицький Дмитро

студент 2-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Наразі жанр roguelike входить у ранг популярних жанрів ігор. Особливістю таких ігор є процедурна генерація рівнів гри. Процедурна генерація зазвичай використовується для створення вмісту для відеоігор або анімаційних фільмів,

наприклад пейзажів, 3D-об'єктів, дизайну персонажів, анімації або діалогів неігрових персонажів [1].

Є різні варіанти реалізації процедурної генерації, одним з яких є застосування графів. У процедурній генерації графи використовуються для конструювання ігрових рівнів.

Для створення графо-орієнтованої генерації рівнів необхідно декларувати певний перелік можливих вершин у вигляді кімнат рівнів. Також треба виділити певне поле, в межах якого випадковим чином генеруватиметься граф рівня. Як приклад ігрового поля, автором було вибрано розмір поля 30 х 30, тобто 30 на 30 можливих кімнат (вершин графа). Кількість кімнат не є тотожною розмірам поля, це залежить від врахування розмірів і кількості кімнат, а також від рівня складності гри (рис. 1). Для поля було визначено буфер відступів, тобто віддаленість між кімнатами, щоб складалось візуальне враження про кімнати як окремі ігрові простори.

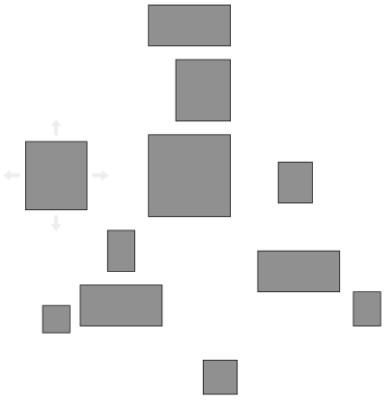


Рисунок 1 – Вершини графа (кімнати) ігрового поля

Створені випадковим чином кімнати треба поєднати в єдине ігрове поле, тобто у планарний граф за допомогою концепції тріангуляції Делоне (рис. 2). Для реалізації цього поєднання було використано алгоритм Бойера-Ватсона [2].

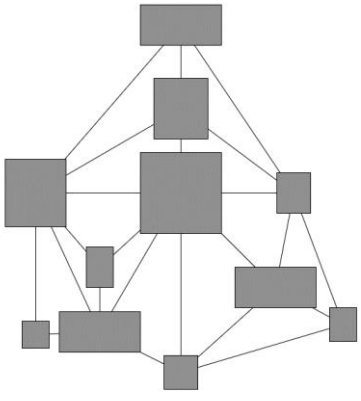


Рисунок 2 – Тріангуляція Делоне з кімнат

Для структуризації графа кімнат ігрового поля з тріангуляції треба створити мінімальне остове дерево. Для цього в роботі було використано алгоритм Прима [3], який дозволив прибрати дублюючі коридори (ребра) між кімнатами (вершинами графа) (рис. 3).

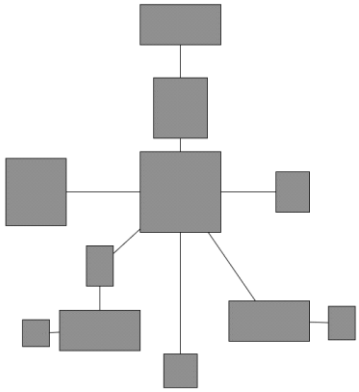


Рисунок 3 – Мінімальне остове дерево коридорів

Далі для кожного ребра (коридора) з-посеред інших у множині всіх наявних ребер було застосовано алгоритм A*, щоб побудувати можливий маршрут руху гравця від однієї кімнати до інших [4]. Після знаходження траєкторії руху між кімнатами алгоритм A* буде змінювати стан рівня так, щоб попередні (вже пройдені) коридори не «налазили» візуально на поточний.

Приклад отриманого ігрового поля (рівня) при застосуванні усіх цих алгоритмів засобами мови C# наведено на рис. 4. Отримана схема утворена за структурою згенерованого випадковим чином графа. Такий підхід дозволяє кожного разу генерувати різні за формою та розміром рівні, що урізноманітнює досвід гравця і перегравaність для самої гри, тобто бажання проходити навіть вже пройдені рівні повторно.

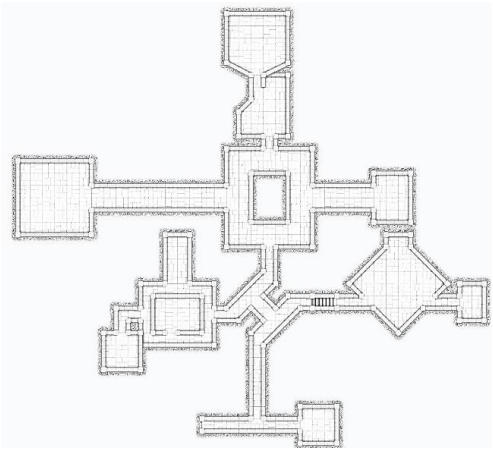


Рисунок 4 – Згенерований рівень(ігрове поле) у формі графа

Відомими прикладами подібної генерації ігрових рівнів є ігри “The Binding of Isaac” та “Rogue Legacy”. Тим самим процедурна генерація ігрових рівнів за допомогою графів є потужним інструментом для створення різноманітного візуального контенту в комп’ютерних іграх. Завдяки своїй гнучкості й ефективності, цей підхід дозволяє знаходити нові застосування у сфері комп’ютерних ігор.

Список використаних джерел:

1. Procedural generation: creating 3D worlds with deep learning. URL: https://www.mit.edu/~jessicav/6.S198/Blog_Post/ProceduralGeneration.html..
2. Bowyer-Watson Algorithm for Delaunay Triangulation. URL: https://www.gorillasun.de/blog/bowyer-watson-algorithm-for-delaunay-triangulation/
3. Prim’s Algorithm for Minimum Spanning Tree (MST). URL: https://www.geeksforgeeks.org/prims-minimum-spanning-tree-mst-greedy-algo-5/
4. A* Search Algorithm. URL: https://www.geeksforgeeks.org/a-search-algorithm/

Ключові слова: генерація, тріангуляція, мінімальне остове дерево, графи, roguelike
Keywords: generation, triangulation, MST, graphs, roguelike

Науковий керівник: доцент О. Трофименко

ВІЗУАЛІЗАЦІЯ МЕРЕЖЕВИХ ПРОТОКОЛІВ У ВИГЛЯДІ ГРАФІВ

Задерейко Олександр

кандидат технічних наук, доцент, доцент кафедри інформаційних технологій Національного університету “Одеська юридична академія”

Шляхтицький Дмитро

студент 2-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Теорія графів є важливою галузю дискретної математики, яка вивчає графи. Графи є множинами точок (вершин, вузлів), які з’єднуються множинами ліній (ребер, дуг). Графи використовуються для моделювання різних систем, включаючи комп’ютерні мережі [1].

Комп’ютерні мережі широко використовують теорію графів для аналізу, проектування та оптимізації їх структури та функціонування. Наведені 5 основних аспектів використання теорії графів у комп’ютерних мережах:

1. Топологія мережі: теорія графів дозволяє моделювати різноманітні топології мережі, такі як зірка, дерево, кільце (цикл), меш (сітка), та інші. Це

допомагає розуміти структуру мережі та її характеристики, такі як масштабованість, надійність, та ефективність.

2. Маршрутизація та передача даних: графи використовуються для моделювання шляхів маршрутизації в мережі та оптимізації передачі даних. Алгоритми, які базуються на теорії графів, такі як алгоритм Дейкстри та алгоритм Беллмана-Форда, використовуються для пошуку найкоротших шляхів у мережах.

3. Аналіз мережі: теорія графів дозволяє аналізувати властивості мережі, такі як централізованість, вузлова важливість, густота мережі та інші метрики. Це допомагає у виявленні слабких місць та вдосконаленні мережевих архітектур.

4. Оптимізація розподілених систем: графи використовуються для оптимізації розподілених систем, таких як хмарні обчислення або системи обробки даних у реальному часі. Вони допомагають визначити оптимальне розташування ресурсів та мінімізувати затрати на передачу даних.

5. Мережевий дизайн: теорія графів використовується для проектування нових мереж та вдосконалення існуючих. Вона дозволяє враховувати різні фактори, такі як пропускна здатність, завантаження, вартість та надійність, для створення оптимальних мережевих рішень.

Важливо візуалізувати протоколи як графи, оскільки це дозволяє краще розуміти їхню структуру та взаємозв'язки між різними компонентами[2]. Графічне представлення допомагає уявити, як дані пересилаються від одного вузла до іншого через мережу, як вони обробляються та які перешкоди можуть виникати на цьому шляху [2].

Наприклад, в графі можна відобразити взаємодію між різними протоколами та їхні ролі в мережевому обміні даними. Також можна показати, як протоколи взаємодіють з різними аспектами мережевого середовища, такими як маршрутизатори, комутатори та сервери.

Графічне представлення також допомагає візуалізувати потенційні проблеми та вразливості в мережевій інфраструктурі, що може бути корисним для розробників програмного забезпечення та адміністраторів мережі. Такий підхід дозволяє швидше виявляти та вирішувати проблеми з мережевими з'єднаннями та забезпечувати більш ефективне функціонування мережі в цілому. Візуалізація мережевих протоколів у вигляді графів може бути виконана за допомогою різних методів. Були обрані мова Python та допоміжнні бібліотеки matplotlib та network [3].

Ці бібліотеки, розроблені для використання з мовою програмування Python, дозволяють візуалізувати мережеві графи, представлені у вигляді різних

баз даних. Цей метод є одним з найсучасніших та найбільш актуальних для широкого спектра операцій. Їх використання дозволяє гнучко налаштовувати процес візуалізації та інтегрувати його з іншими модулями Python. Однак цей метод може вимагати певних навичок програмування та розуміння теорії графів. Програмний код, що реалізує візуалізацію мережевих протоколів у виглядів графів показана нижче (рис. 1) та результат роботи (рис. 2).

```
import networkx as nx
import matplotlib.pyplot as plt

G = nx.DiGraph()

nodes = ['PC', 'localhost', 'HABR', 'MS OFFICE', 'HABR eternal server 1',
         'HABR eternal server 2', 'HABR eternal server 3', 'HABR Storage',
         'MS Storage', 'Google API', 'Google', 'Bing', 'Bing Traffic Manager',
         'GStatic', 'Telegram', 'Proxy', 'Spotify', 'Google Play']

G.add_nodes_from(nodes)

edges = [
]

G.add_edges_from(edges)

pos = nx.shell_layout(G)
nx.draw(G, pos, with_labels=True, node_size=1500, node_color='lightblue', font_size=8, font_weight='bold', arrowsize=5)

edge_labels = nx.get_edge_attributes(G, name: 'protocol')
for edge, label in edge_labels.items():
    nx.draw_networkx_edge_labels(G, pos, edge_labels={(edge[0], edge[1]): label}, font_color='red', font_size=6)

plt.title("Граф компонентів мережі та протоколів")
plt.show()
```

Рисунок 1 – Програмний код візуалізації мережевих з'єднань

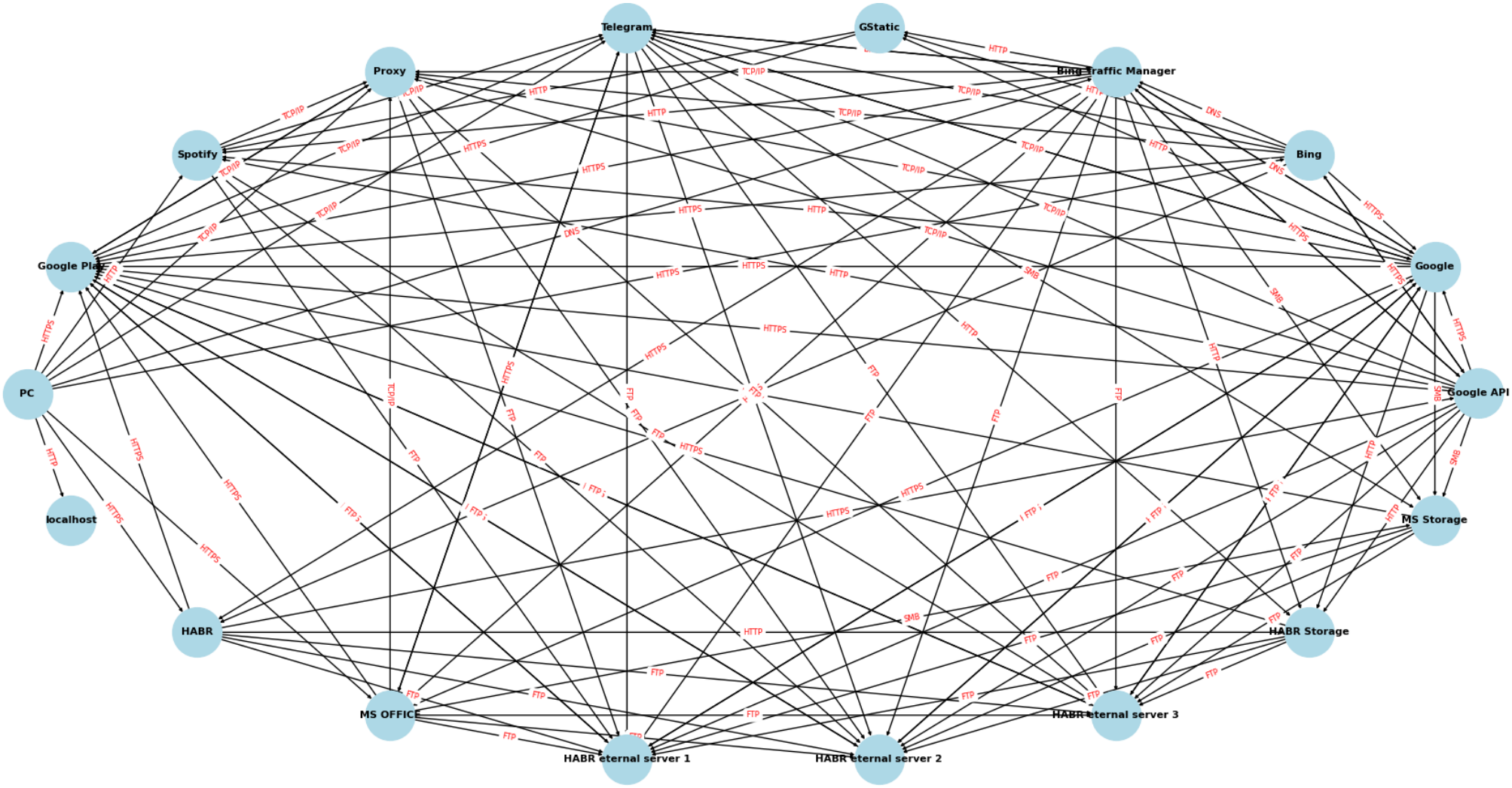


Рисунок 2 – Візуалізація мережевих з'єднань (результат виконання програми)

Зважаючи на зростаюче значення безпеки в комп’ютерних мережах, аналіз даних мережевих протоколів у вигляді графів може бути надзвичайно корисним для виявлення потенційних загроз та вразливостей у комп'ютерних мережах [4]. Представлення мереж у вигляді графів дозволяє відобразити взаємозв'язки між різними вузлами і сегментами мереж та ідентифікувати аномальну (не стандартну) взаємодію, що опосередковано може свідчити про зловмисні дії або витік конфіденційної інформації. Використання графів у цьому контексті дозволяє адміністраторам мереж оперативно реагувати на поточний стан безпеки мереж, при необхідності коригувати його з метою забезпечення захисту від потенційних кібератак.

Список використаних джерел:

1. Кузьменко, Ігор Миколайович. "Теорія графів." (2020). URL: https://ela.kpi.ua/server/api/core/bitstreams/fb0a4251-74d9-470b-88da-71abb4e85f93/content.
2. Басюк, Тарас, and Дмитро Тисько. "Методи візуалізації мережі в стандарті ZigBee." (2017). URL: https://oldena.lpnu.ua/bitstream/ntb/38385/1/101_212-213.pdf.
3. MatPlotLib. URL: https://matplotlib.org/.
4. NetWorkX. URL: https://networkx.org/.

Ключові слова: комп’ютерні мережи, інтернет, програмування, графічне представлення мереж

Keywords: computer networks, Internet, programming, graphical representation of networks.

Секція 2. ІНФОРМАЦІЙНА БЕЗПЕКА ТА ЗАХИСТ ІНФОРМАЦІЇ: СОЦІАЛЬНИЙ, ПРАВОВИЙ І ТЕХНІЧНИЙ АСПЕКТ

ІНСТРУМЕНТИ МОНІТОРИНГУ ЯК ОБОВ'ЯЗКОВИЙ КРИТЕРІЙ ЕКСПЛУАТАЦІЇ СЕРВЕРА

Астахов Даниїл

студент 4-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

На сьогоднішній день інформаційні системи стають все складнішими та більшими, тому необхідно ретельно вивчити їхні слабкі сторони та недоліки. Забезпечення стабільної роботи цих систем має бути головним пріоритетом для будь-якої організації.

Одним із способів виявлення потенційних проблем в інформаційній системі є впровадження сервісів моніторингу роботи системи, які націлені на відстеження подій, пов'язаних з наступними категоріями:

1. Моніторинг доступу
2. Моніторинг мережі
3. Моніторинг системних ресурсів
4. Виявлення вразливостей
5. Журналювання подій

Вони видають сповіщення або повідомлення, коли перевищуються попередньо визначені порогові значення або виявляються аномалії. Кожен із цих сервісів моніторингу відіграє важливу роль, оскільки дозволяє інженерам запобігати виникненню серйозних проблем, оперативно реагуючи на потенційні негативні події. Забезпечення надійної, безпечної та ефективної роботи IT-інфраструктури є кінцевою метою інфраструктурного моніторингу.

Кращими практиками моніторингу є:

1. Встановлення та перегляд базових метрик
2. Налаштування комплексних сповіщень
3. Організація та пріоритезація сповіщень

Є безліч інструментів, які включають у собі збір даних, але є основні, які інженери використовують найчастіше, а саме:

1. Prometheus+Loki+Grafana Stack
2. ELK Stack
3. Datadog

Перші два інстументи є класичними сервісами для збору даних, в той час, як Datadog стає все більш та більш популярним в використанні серед багатьох компаній та навіть великих корпорацій. Вони представляють зрозумілі інструменти для спостереження та документування подій. Розгортання застосунку та сервера в цілому обов'язково включає встановлення хоча б одного інструмента.

В той час, ELK Stack це комбінація трьох відкритих програмних продуктів: Elasticsearch, Logstash та Kibana.

1. Elasticsearch забезпечує швидкий та масштабований доступ до даних, ідеально підходить для пошуку, аналізу та збереження великих обсягів журналів.

2. Logstash - це інструмент обробки журналів, який використовується для збору, обробки та нормалізації журналів з різних джерел.

3. Kibana: Це візуалізаційний інтерфейс, який дозволяє користувачам створювати різноманітні графіки, діаграми та інші візуальні елементи на основі даних, збережених у Elasticsearch.

Він підходить більше всього для аналізу великих обсягів даних, тобто для великих застосунків або великої кількості маленьких мікросервісів.

Моніторинг середовища Kubernetes є надзвичайно важливим для забезпечення оптимальної продуктивності, надійності та доступності ваших додатків. По мірі зростання популярності Kubernetes, стає надзвичайно важливим розроблення ефективної стратегії моніторингу. Імплементація інструментів моніторингу та інших інструментів моніторингу дозволить зменшити можливі проблеми та забезпечити безперервне функціонування вашого середовища Kubernetes.

Список використаних джерел:

1. Інструменти моніторингу Kubernetes [Електронний документ] – URL: http://surl.li/phugt.
2. Infrastructure monitoring: An introduction [Електронний документ] – URL: https://mrintegrity.medium.com/monitoring-from-scratch-ea2b83a8f8a5.
3. Kubernetes Monitoring | Tools + 9 Best Practices [Електронний документ] – URL: https://www.cherryservers.com/blog/kubernetes-monitoring-tools
4. What is infrastructure monitoring? [Електронний документ] – URL: https://www.ibm.com/topics/infrastructure-monitoring#:~:text=Infrastructure%20monitoring%20helps%20identify%20faults,network%20outages%20or%20application%20errors.

Ключові слова: моніторинг, журналювання, вразливості, доступ
Keywords: monitoring, logging, vulnerabilities, vulnerabilities.

Науковий керівник: доцент В. Бойко

ОГЛЯД СТЕГАНОГРАФІЧНИХ ЗАСТОСУНКІВ ДЛЯ ЦИФРОВИХ ЗОБРАЖЕНЬ

Ахмаметьєва Ганна

кандидат технічних наук, доцент, доцент кафедри кібербезпеки Національного університету «Одеська юридична академія»

Пискун Микола

студент 5-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

У зв'язку з широким поширенням моніторингу та контролю відкритих каналів зв'язку, з яких збирають персональні дані особи, її області інтересів, круг спілкування, місцезнаходження, наслідком чого може бути виток чутливих даних до зловмисників. Це призводить до необхідності використання засобів захисту, зокрема криптографічні і стеганографічні алгоритми та програмні застосунки, що реалізують ці перетворення.

Обмін мультимедійними даними (зображень, відео, аудіо та голосових повідомлень) в месенджерах, соціальних мережах не є чимось підозрюваним, за рахунок можливості забезпечення високої пропускної здатності можливо організувати канал прихованого зв'язку. При використанні стеганографічних перетворень слід звертати увагу на стійкість до стиснення вбудованих в контейнер даних та можливість їх відновлення, адже деякі реалізації стеганографічних перетворень здійснюють вбудовування секретної інформації в просторову область контейнера, що часто не забезпечує стійкості до стиснення. Розмір стеганоповідомлень в форматі без втрат часто займає набагато більший розмір, ніж файлів в форматі з втратами, крім того в процесі завантаження медіафайлів на деякі ресурси відбувається автоматичне стиснення даних, що може зруйнувати стеганоповідомлення. З форматів без втрат останнім часом найпоширенішими є формат PNG.

Далі буде проведений огляд деяких програмних застосунків, які корисно використовувати для приховування даних.

Hallucinate [1]. Програма не потребує встановлення, має простий зрозумілий інтерфейс. В якості зображення-контейнеру можна обрати довільний формат, при цьому стеганоконтейнер можна зберегти лише в форматі BMP та PNG. Повідомленням може бути файл довільного формату. Застосунок реалізує вбудовування секретної інформації методом LSB та дає можливість обрати бажану якість підсумкового зображення, всього доступні вісім варіантів: Best, Better,

Good, Fair, Poor, Bad, Worse, None. Слід зазначити, що чим нижче якість вихідного зображення, тим більше виникає візуальних артефактів, в той же час можна вбудувати більший обсяг конфіденційної інформації.

OpenPuff [2]. Застосунок реалізує дві задачі стеганографії: вбудовування і вилучення секретного повідомлення в мультимедійні файли, вбудовування і перевірку цифрового водяного знаку як захист авторських прав. В якості контейнерів можуть виступати зображення (формати BMP, JPEG, PCX, PNG, TGA), аудіо (формати AIFF, MP3, NEXT/SUN, WAVE) і відео (формати 3GPP, FLV, MP4, MPG, SWF, VOB), а також файли PDF. Методом вбудовування є метод LSB, тому стеганоконтейнери потрібно зберігати в формат без втрат.

Додатковою перевагою програми можна відзначити можливість виділення необхідної кількості ядер процесора і кількість потоків у випадку, коли використовуються великі файли. Крім того в застосунку реалізований складний парольний захист повідомлень при їх вбудовуванні і вилученні – можна ввести три різні паролі довжиною від 8 до 32 символів, на їх основі буде згенерований унікальний ключ за допомогою криптографічно стійкого генератору псевдовипадкових чисел CSPRNG (Cryptographically secure pseudorandom number generator), який задаватиме стеганографічний шлях.

OpenStego [3]. Програма підтримує контейнери в форматі BMP, PNG, JPG, GIF і WBMP, однак заповнений контейнер завжди зберігається ф форматі PNG. В якості повідомлення можна взяти будь-які файли. Застосунок підтримує парольний захист, вбудовування/вилучення прихованих повідомлень і вбудовування/перевірку цифрових водяних знаків.

Hide ‘N’ Send [4]. Компактний і простий у використанні інструмент для вбудовування секретних повідомлень в файлах зображень. Застосунок підтримує кілька алгоритмів приховування (M-F5, M-LSB, F5, LSB), хешування (SHA512, RIPEMD, MD5) і шифрування (AES, RC2, RC4). Слід зазначити, що можливе лише використання зображень в форматі JPG, під час процесу приховування також можна ввести пароль. Однак програма не дає можливості задати якість результату вбудовування, крім того, під час вбудовування контейнер переписується стеганоконтейнером.

RedJPEG [5]. Програма має простий інтерфейс та призначена для приховування інформації в файли формату JPEG. В якості додаткової інформації можна обрати файли довільного формату. Також є можливість використання перевірки цілісності даних, використовуючи алгоритми AMPRNG і Cartman II DDP4 в режимі

хеш-функції, і стиснення даних з використанням LZMA-компресії. Обов'язковою умовою стеганографічного перетворення є введення паролю.

Характерною рисою програми є наявність рядку стану, в якому відображаються назва файлу, наявність або відсутність даних та ємність файлу. При вбудовуванні файлу також відображений розмір даних для приховування та співвідношення розмірів у відсотках, що дозволяє одразу побачити, чи вміститься файл у контейнер. Застосунок зберігає результат в форматі JPEG, однак жодних налаштувань якості стеганоповідомлення при збереженні програмою не передбачено.

Steghide UI [6]. Програма призначена для приховування конфіденційних даних в зображеннях (BMP, JPG) і аудіо (WAV, AU). Серед параметрів можна задати алгоритм шифрування (Blowfish, Cast-128, Cast-256, DES, Enigma, AES 256/128/192, Saferplus, Serpent, TripleDES, Twofish та інші), режим шифрування, ввести пароль та задати рівень стиснення (0 – без стиснення, 1 – мінімальне стиснення, 9 – максимальне стиснення).

Отже описані основні можливості найбільш коректних і функціонально наповнених стеганографічних застосунків для вбудовування конфіденційної інформації в мультимедійні дані.

Список використаних джерел:

1. Hallucinate. SourseForge [Електронний ресурс]. Режим доступу: http://sourceforge.net/projects/hallucinate/
2. OpenPuff. SoftLot [Електронний ресурс]. Режим доступу: https://www.softslot.com/software-1007-openpuff.html
3. OpenStego [Електронний ресурс]. Режим доступу: http://www.openstego.com/
4. Hide 'N' Send. Softpedia [Електронний ресурс]. Режим доступу: https://www.softpedia.com/get/Security/Encrypting/Hide-N-Send.shtml
5. Цифрова стеганографія: Програми та інші засоби. SPY-SOFT.NET [Електронний ресурс]. Режим доступу: https://spy-soft.net/cifrovaya-steganografiya-sposoby-realizacii/
6. Steghide UI. Softpedia [Електронний ресурс]. Режим доступу: https://www.softpedia.com/get/Security/Encrypting/Steghide-UI.shtml

Ключові слова: стеганографія, цифрові зображення, область перетворень, просторова область

Keywords: steganography, digital images, transform domain, spatial domain

ПОРІВНЯЛЬНИЙ АНАЛІЗ ВІДСТАНІ НЕЛІНІЙНОСТІ ВИПАДКОВОЇ БУЛЕВОЇ ФУНКЦІЇ ТА ВИПАДКОВОГО S-БЛОКА

Бакуніна Олена

кандидат фізико-математичних наук, доцент, доцент кафедри кібербезпеки Національного університету «Одеська юридична академія»

Основною характеристикою конструктів сучасних криптографічних алгоритмів, яка характеризує степінь реалізації даним конструктом концепції конфузії [1], є відстань нелінійності. В сучасній літературі представлено чимало робіт, присвячених дослідженню фундаментальної природи нелінійності криптографічних алгоритмів, а також методів синтезу криптографічних конструктів, що володіють значним рівнем нелінійності. Тим не менш, у загальному випадку, математичні об'єкти, що характеризуються значним рівнем нелінійності, лишаються малодослідженими та такими, що приховують у собі ще багато таємниць. Їх подальше дослідження являє суттєвий теоретичний інтерес, а також важливе з точки зору вдосконалення та розробки нових криптографічних конструкцій.

Зокрема, цікавими лишаються питання дослідження нелінійності випадкових криптографічних конструкцій. Відомо, що для випадково згенерованих булевих функцій, а також і для криптографічних S-блоків, рівень нелінійності суттєво зростає при зростанні їх довжини. При цьому відомо, що S-блоки довжини N формуються з $\log_2 N$ компонентних булевих функцій, які мають об'єднуватися спеціальним чином для формування бієктивного S-блока. При цьому особливості такого об'єднання лишаються малодослідженими, невідомі повністю регулярні методи побудови S-блоків на основі множини булевих функцій. Описані обставини роблять актуальним завдання порівняльного дослідження випадково згенерованих булевих функцій та випадково згенерованих S-блоків при зростанні їх довжини.

Метою роботи є експериментальне порівняння відстані нелінійності випадково згенерованих булевих функцій та S-блоків при зростанні їх довжини.

Введемо необхідні для проведення досліджень визначення.

Визначення 1. Відстанню нелінійності булевої функції називається мінімальна відстань Геммінга

$$N_f = \min(dist(f, A_j)), \quad j = 1,2,...,2^{k+1}, \tag{1}$$

між булевою функцією f і всіма кодовими словами афінного $A(N,k)$–коду [2].

Для довільного натурального k, афінним $A(N,k)$-кодом довжини $N=2^k$ називається множина всіх рядків Ω_f тих булевих функцій, степінь нелінійності яких не перевищує 1, тобто $A(N,k)=\left\{\Omega_f \mid f\in F_k, \deg f \le 1\right\}$ [3].

Визначення 2. Відстанню нелінійності S-блока називається мінімальне значення серед відстаней нелінійності його компонентних булевих функцій.

Визначення 3. Збалансованою булевою функцією називається така булева функція, у таблиці істинності якої кількість значень «1» дорівнює кількості значень «0», тобто така булева функція, вага Геммінга якої дорівнює $wt(f)=N/2$, де $N=2^k$, k — кількість змінних, від яких залежить булева функція.

За побудовою всі компонентні булеві функції бієктивних S-блоків, а також всі їх лінійні комбінації, є збалансованими [4].

Відзначимо також, що максимально можливі значення нелінійності збалансованих булевих функцій, а, таким чином і S-блоків, є меншими від максимально можливих значень незбалансованих булевих функцій. Зокрема, гранично нелінійні булеві функції — бент-функції, є незбалансованими за побудовою.

Для досягнення мети роботи були проведені наступні експерименти: випадково згенеровані $J=100000$ булевих функцій, збалансованих булевих функцій, а також S-блоків підстановки довжини $N=16,32,64,128,256$. Для даних математичних об'єктів відповідно до Визначення 1 і Визначення 2 були вимірені значення відстаней нелінійності, після чого були обраховані середні арифметичні значення для кожної зазначеної довжини. Результати досліджень представлені в табл. 1, де у круглих дужечках вказано значення відставанні відстані нелінійності випадково згенерованих збалансованих булевих функцій та S-блоків від випадково згенерованих булевих функцій.

Таблиця 1 — Результати експериментальних досліджень

Конструкція/Довжина	16	32	64	128	256
Булева функція	3.8325	9.2983	21.5616	47.9066	103.5374
Збалансована булева функція	3.6885 (0.144)	9.2270 (0.0713)	21.4717 (0.0899)	47.8955 (0.0111)	103.4833 (0.0541)
S-блок	3.0288 (0.8037)	7.8886 (1.4097)	19.4664 (2.0952)	44.8370 (3.0696)	99.0570 (4.4804)

Аналіз даних, представлених у табл. 1 дозволяє дійти наступних висновків. Нелінійність випадково згенерованої збалансованої булевої функції є меншою від нелінійності випадково згенерованої булевої функції, при цьому це

відставання є меншим для булевих функцій непарної кількості змінних, тобто довжин $N = 32,128,$ та має тенденцію до зниження із зростанням довжини як для булевих функцій парної так і непарної кількості змінних.

При цьому при порівнянні нелінійності випадково згенерованих S-блоків і нелінійності випадково згенерованих булевих функцій ми спостерігаємо зворотну картину — із зростанням довжини, відставання нелінійності S-блока зростає.

У роботі представлені результати проведеного порівняльного експериментального дослідження нелінійності випадково згенерованої булевої функції, збалансованої булевої функції та S-блока. Встановлено, що із зростанням довжини булевої функції, відставання нелінійності збалансованої булевої функції від нелінійності булевої функції, зменшується. При цьому, значення такого відставання є меншими для булевих функцій непарного числа змінних. Дослідження також дозволили встановити, що відставання нелінійності випадково згенерованого S-блока від нелінійності випадково згенерованої булевої функції збільшується із зростанням довжини.

Отримані результати є важливими для розуміння фундаментальних особливостей нелінійності таких криптографічних примітивів, як булеві функції та криптографічні S-блоки підстановки, та можуть стати основою для вдосконалення існуючих методів створення нелінійних S-блоків, а також розробки нових.

Список використаних джерел:

1. C.E. Shannon, A: Mathematical Theory of Cryptography, Bell System Technical Memo, 1945.
2. Соколов А.В. Новые методы синтеза нелинейных преобразований современных шифров. Lap Lambert Academic Publishing, Germany 2015. 100 с.
3. Logachev O. A., Salnikov A. A., Yashchenko V. V. Boolean Functions in Coding Theory and Cryptography. Translations of Mathematical Monographs, 2012, Vol. 241. 334 P.
4. Kim K. Construction of DES-like S-boxes Based on Boolean Functions Satisfying the SAC. Proc. of Asiacrypt'91, Springer Verlag. 1991. P. 59-72.

Ключові слова: криптографія, нелінійність, булева функція, S-блок
Keywords: cryptography, nonlinearity, Boolean function, S-box

РОЗРОБКА ГЕНЕТИЧНОГО АЛГОРИТМУ ДЛЯ ПОБУДОВИ ВИСОКОНЕЛІНІЙНИХ S-БЛОКІВ ПІДСТАНОВКИ

Баландіна Наталія

старший викладач кафедри кібербезпеки
Національного університету «Одеська юридична академія»

Одним з основних компонентів сучасних шифрів, який визначає їх ефективність, є криптографічний S-блок. Побудова криптографічних S-блоків має відбуватися таким чином, щоб вони відповідали базовим критеріям криптографічної якості: критерію максимізації алгебраїчної та дистанційної нелінійності, критерію розповсюдження помилки, критерію мінімізації кореляційного зв'язку між векторами виходу та входу S-блока підстановки.

На сьогодні створено чимало методів синтезу криптографічних S-блоків, що засновані на різноманітних математичних конструкціях, серед яких особливе місце займають конструкції полів Галуа через значні їх можливості щодо максимізації нелінійних властивостей синтезованих S-блоків. Наприклад, відома конструкція Ніберг [1] довжини $N = 256$, яка застосовується у криптографічному алгоритмі AES, дозволяє отримати значення дистанційної нелінійності $N_S = 112$. Відомі і інші методи синтезу криптографічно високоякісних S-блоків на основі наступних конструкцій полів Галуа: коди степеневих лишків, багаторівневі лінійні рекурентні послідовності (БЛРП), GF-перетворення та інші.

Тим не менш, незважаючи на значні переваги S-блоків, заснованих на конструкціях полів Галуа, вони не полишені і недоліків, що пов'язані із детермінованим характером полів Галуа, а також обмеженими потужностями множин синтезованих S-блоків. Так, у роботі [2] проводяться дослідження, які показують існування лінійної залежності між компонентними булевими функціями S-блоків конструкції Ніберг, що є їх недоліком.

Як зазначають чимало сучасних дослідників в області криптографії, найкращими з точки зору імплементації концепцій дифузії і конфузії є (псевдо)випадкові S-блоки, що, тим не менш, характеризуються відповідністю основним критеріям криптографічної якості. Як показують дослідження, побудова таких криптографічних S-блоків, що засновані на (псевдо)випадковому характері перестановки елементів, а також володіють високим рівнем криптографічної якості, може бути здійснена із застосуванням генетичних алгоритмів.

Метою даної роботи є підвищення ефективності криптографічних алгоритмів шляхом синтезу S-блоків, що володіють високим рівнем криптографічної якості на основі генетичних алгоритмів.

Одним з основних критеріїв криптографічної якості, що застосовуються до сучасних S-блоків, є критерій максимізації дистанційної нелінійності, яку можна визначити як мінімальну відстань Геммінга

$$N_f = \min(dist(F_i, A_j)), \quad i = \overline{1,k}, \quad j = 1, 2^{k+1}, \tag{1}$$

між компонентними функціями перетворення і всіма кодовими словами афінного $A(N,k)$–коду [3].

Для довільного натурального k, афінним $A(N,k)$-кодом довжини $N = 2^k$ називається множина всіх рядків Ω_f тих булевих функцій, степінь нелінійності яких не перевищує 1, тобто $A(N,k) = \{\Omega_f \mid f \in F_k, \deg f \le 1\}$.

Даний критерій покладений в основу запропонованого генетичного алгоритму синтезу криптографічних S-блоків, який викладемо у вигляді наступних конкретних кроків:

Крок 1. Задати початкову популяцію з $n/2$ S-блоків довжини N, кожний з яких визначається тривіальною монотонно-зростаючою послідовністю виду $0,1,...,N-1$. Задати значення лічильника ітерацій генетичного алгоритму $i = 0$.

Крок 2. На основі популяції з $n/2$ S-блоків створити ще $n/2$ S-блоків, здійснюючи перестановку двох, випадковим чином обраних елементів їх Q-послідовностей, для кожного з них.

Крок 3. Провести вимірювання дистанційної нелінійності N_S кожного з n отриманих S-блоків, згідно до (1).

Крок 4. Видалити з популяції $n/2$ S-блоків, що володіють найменшими значеннями нелінійності N_S.

Крок 5. Якщо досягнуто останньої ітерації генетичного алгоритму, тобто $i = i_0 - 1$, зупин, інакше — перейти на Крок 2.

Для значення довжини S-блока $N = 256$, загальної кількості S-блоків у популяції $n = 100$, а також кількості ітерацій генетичного алгоритму $i_0 = 1000$, були отримані наступні результати щодо зростання максимального значення дистанційної нелінійності серед S-блоків у популяції в залежності від номеру ітерації генетичного алгоритму (рис. 1, де продемонстровані ітерації від 0 до 400).

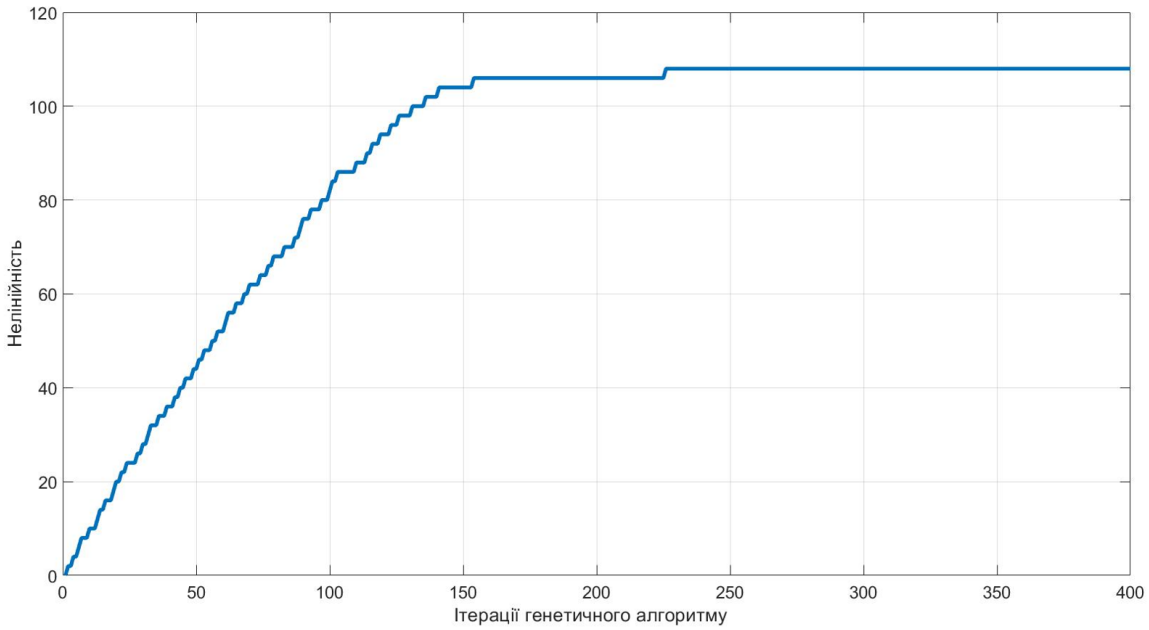


Рисунок 1 – Графік залежності максимального значення дистанційної нелінійності серед S-блоків у популяції від номеру ітерації генетичного алгоритму

Аналіз даних рис. 1 показує стрімке зростання максимального значення дистанційної нелінійності серед S-блоків у популяції до 154 ітерації, після чого, на 227 ітерації максимальне значення дистанційної нелінійності стабілізується на значенні $\max(N_S) = 108$, що складає ~96.43% від максимального значення нелінійності бієктивних S-блоків довжини $N = 256$.

Ми наводимо приклад одного із згенерованих за допомогою запропонованого генетичного алгоритму S-блоків у вигляді його Q-послідовності, що кодує.

$$S = \{86,150,43,44,105,93,12,107,196,206,161,138,217,13,120,59,29,\\ 199,19,189,110,139,26,230,80,25,220,203,68,125,227,145,235,4,232,\\ 183,42,155,211,148,253,245,201,163,84,99,66,39,210,54,90,78,100,\\ 9,11,252,147,160,181,135,186,30,177,133,57,193,\ 89,55,149,27,38,136,\\ 239,20,209,170,65,144,122,205,250,40,154,46,168,56,21,251,184,69,\\ 95,231,248,223,187,35,104,119,101,88,60,106,204,197,36,81,126,137,\\ 109,28,79,32,172,34,117,175,179,229,17,153,166,247,207,85,164,240,\\ 214,\ 33,47,188,31,16,131,185,94,146,128,130,\ 18,202,58,225,75,91,71,\\ 212,114,208,7,2,0,226,140,169,82,162,238,171,233,213,45,165,167,218,\\ 174,37,121,190,180,127,115,123,243,64,50,151,134,6,191,158,192,73,97,\\ 228,112,52,1,51,156,222,53,157,83,22,249,10,221,215,198,195,241,74,\\ 113,124,132,5,159,98,254,236,224,116,62,141,194,152,176,111,182,200,\\ 70,\ 15,237,118,\ 63,\ 76,173,72,178,77,108,61,96,103,129,14,102,67,\\ 143,242,3,92,23,255,234,24,246,41,48,49,216,244,87,142,8,219\}. \quad (2)$$

Зазначений S-блок володіє дистанційною нелінійністю $N_S = 108$. Наведемо також інші показники криптографічної якості зазначеного S-блока. Його алгебраїчний степінь нелінійності складає $\deg(S) = 7$.

Наведемо матрицю ваг Геммінга похідних компонентних булевих функцій S-блока (2)

e_j	$wt\left(D_{1,k}\right)$	$wt\left(D_{2,k}\right)$	$wt\left(D_{3,k}\right)$	$wt\left(D_{4,k}\right)$	$wt\left(D_{5,k}\right)$	$wt\left(D_{6,k}\right)$	$wt\left(D_{7,k}\right)$	$wt\left(D_{8,k}\right)$
10000000	144	132	128	120	136	124	128	128
01000000	132	128	132	120	132	140	112	116
00100000	132	136	140	120	112	120	128	116
00010000	128	128	132	136	148	116	148	136
00001000	120	120	116	136	132	108	140	112
00000100	132	116	120	136	128	124	132	124
00000010	120	132	128	124	132	132	124	124
00000001	92	124	124	116	120	132	140	132

, (3)

а також його матрицю коефіцієнтів кореляції

$$R=\begin{bmatrix} 0.05 & 0.08 & 0.05 & 0.06 & -0.08 & 0.03 & -0.02 & -0.09 \\ 0 & 0.09 & 0 & -0.03 & 0 & -0.05 & -0.08 & 0.08 \\ 0.05 & -0.11 & -0.05 & -0.06 & 0.03 & 0.11 & 0.06 & 0.02 \\ 0.03 & 0.03 & 0.06 & -0.13 & -0.05 & -0.09 & 0.05 & -0.06 \\ -0.05 & 0.06 & -0.06 & 0.02 & 0.06 & 0 & 0 & 0 \\ -0.08 & -0.02 & -0.06 & 0 & 0.02 & 0.13 & 0.06 & -0.02 \\ -0.06 & 0.03 & 0 & 0.08 & -0.05 & -0.03 & 0.09 & 0.02 \\ 0.02 & 0.06 & 0 & 0.06 & 0.13 & -0.05 & -0.05 & 0 \end{bmatrix}. \quad (4)$$

Аналіз (3) і (4) показує, що окрім високих як алгебраїчної, так і дистанційної нелінійності, S-блок (2) володіє припустимими лавинними і кореляційними властивостями, що у поєднанні з його псевдовипадковою природою дозволяє рекомендувати його до практичного застосування.

Висновки

Відзначимо основні результати проведених досліджень:

1. Представлено генетичний алгоритм для синтезу множин довільної потужності високонелінійних (псевдо)випадкових S-блоків підстановки довільної довжини N.

2. Дослідження криптографічних властивостей синтезованих за допомогою розробленого генетичного алгоритму S-блоків довжини $N=256$ дозволило встановити високий рівень їх криптографічної якості, зокрема, дистанційної та алгебраїчної нелінійності, а також припустимі лавинні і дистанційні властивості. У поєднані із їх псевдовипадковою природою це дозволяє рекомендувати такі S-блоки до практичного застосування.

Список використаних джерел:

1. Nyberg K. Differentially uniform mappings for cryptography. I Advances in cryptology. Proceedings of EUROCRYPT'93 (1994) vol.765, Lecture Notes in Compuer Springer-Verlag, Berlin, Heidelberg, New York. P.55-65.

2. Sokolov A.V., Barabanov N.A. Algorithm for removing the spectral equivalence of component Boolean functions of Nyberg-design S-boxes. Radioelectronics and Communications Systems. 2015. Vol. 58, No. 5. P. 220-227.
3. Соколов А. В. Новые методы синтеза нелинейных преобразований современных шифров. Lap Lambert Academic Publishing, Germany 2015. 100 с.

Ключові слова: криптографія, S-блок, нелінійність, генетичний алгоритм
Keywords: cryptography, S-box, nonlinearity, genetic algorithm

КОНТРОЛЬ ЦІЛІСНОСТІ СИСТЕМНИХ ЖУРНАЛІВ ТА ЗАГАЛЬНИЙ ЗАХИСТ РОЗПОДІЛЕНИХ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ СИСТЕМ ВІД ВТОРГНЕНЬ

Бойко Віктор
кандидат технічних наук, доцент кафедри кібербезпеки
Національного університету «Одеська юридична академія»

Одним з основних засобів як первинної, так і вторинної діагностики вторгнення є журнали подій (логи) програмного забезпечення та системні журнали (логи операційної системи) - набори діагностичних повідомлень, які генеруються штатно або позаштатно функціонуючого програмного забезпечення та можуть бути ефективно використані для виявлення факту вторгнення - З прийняттям подальших дій [1].

Логи часто є єдиним способом для розкриття інцидентів, їх аналіз може проводитися різними способами - як ручним, так і автоматизованим - шляхом підбору сигнатур і евристичних правил, однак застосовуваних не до аналізу бінарних файлів, а до аналізу повідомлень у логах. Логи є цінним матеріалом для аналізу характеру атак, поведінки атакуючої сторони та інформуванням у разі прориву захисту. Робота з логами та системними журналами є важливим компонентом безпеки системи, тому, якщо розгортається інтегрована система безпеки - до неї обов'язково включається SIEM - Security information and event management [2], який раніше був поєднанням SIM (Security information management) - системи управління інформацією про безпеку, та SEM (Security event management) - системи управління подіями безпеки[3].

При цьому, якщо йдеться про системи низової ланки, адміністратори рідко беруть на себе працю організувати бекап, обробку та аналіз логів [4], [5]. При цьому бекап, навіть якщо він реалізований, рідко перевіряється на цілісність і

консистентність - чим можна скористатися для маскування злому системи шляхом підміни або спотворення логів, які можуть його виявити.

У роботах [6] та [7] показано необхідність створення системи моніторингу цілісності системних журналів на основі технології зв'язаних списків (блокчейн) для забезпечення виявлення системних вторгнень. Порівняно з альтернативними рішеннями (наприклад, повним копіюванням логів системи на резервний сервер) у масштабах веб-серверів середнього рівня запропонована система проста, невибаглива у використанні та невибаглива до ресурсів. Однак при масштабуванні такої системи рівня великих кластерів виникають деякі труднощі через розподіленого характеру зберігання даних у таких системах. Крім того, збільшення обсягу даних, що зберігаються та оброблюються в системних журналах, може пропорційно збільшити час перевірки системних журналів, що може негативно вплинути на час виявлення вторгнення до системи. Для перевірки системних журналів необхідні ефективніші структури даних. Для більш розвинених і розподілених систем, зокрема, що використовують оркестрацію контейнерних додатків, використання такої системи може спричинити низку складнощів, пов'язаних з дистрибутивністю та динамічним характером функціонування систем [3]. Оскільки контейнерні програми та оркестратори стають все більш поширеними, пропоновану систему управління необхідно вдосконалити, щоб врахувати ці особливості.

В рамках завдання комп'ютерну мережу або сукупність віртуального або реального обладнання можна розглядати як граф вузлів, пов'язаних з'єднаннями та генеруючи повідомлення журналу [8]. Існує також центральний, довірений вузол мережі (CN – центральний вузол), який може з високою надійністю отримувати, зберігати та обробляти інформацію від вузлів мережі. Системний журнал має текстову форму (або може бути зведений до неї), повідомлення для нього є лінійними текстовими записами, кожна з яких має мітку даних, опис події і, як правило, займає один рядок. З погляду криптографії ми можемо називати вузол CN «верифікатором», а будь-який із вузлів, який має довести істинність своїх журналів, «коммітером». В описаній схемі, завдання аудиту та перевірки журналів зводиться до завдання аудиту та перевірки дайджестів журналів. У роботі [7] запропоновано механізм, що ґрунтується на технології простого зв'язаного списку (простий блокчейн). Однак цей механізм не дуже зручний для роботи у розподіленій мережі вузлів.

Розподілена мережа вузлів потребує рішення, яке дозволяло б контролювати цілісність баз даних дайджестів і не вимагало б зберігання та передачі самих

дайджестів через мережу між вузлами або мінімізувало б такий обмін даними [9]. Перспективним є механізм, що ґрунтується на використанні т.зв. Дерев Меркла [10] або їх модифікацій, заснований на побудові дерева дайджестів, де дайджести об'єднані спільним батьком, який є дайджестом з їхніх дайджестів, у свою чергу, батьківських вузлів дерева об'єднуються іншими дайджестами і так дане. У можливій схемі нижній рівень такого дерева зберігатиметься на вузлах мережі, а решта, отримана шляхом обчислення хеш-функцій, зберігатиметься на централізованому вузлі. Якщо зловмисник змінить вміст будь-якого з дайджестів, це покаже контрольний розрахунок хешів його батьків, оскільки збережені та розраховані значення не співпадуть.

Використання даної схеми роботи дозволить з одного боку зберегти всі переваги вихідної системи (невимогливість до ресурсів, відсутність необхідності у побудові та обслуговуванні складної інфраструктури), з іншого, шляхом незначного збільшення надмірності дозволить отримати такі переваги, як зменшення часу верифікації та можливість роботи з розподіленими ресурсами що покращить загальну безпеку системи.

Список використаних джерел:

1. Holt T. J., Leukfeldt R., Van De Weijer S. An examination of motivation and routine activity theory to account for cyberattacks against Dutch web sites // Criminal Justice and Behavior. — SAGE Publications Sage CA: Los Angeles, CA, 2020. Vol. 47, no. 4. P. 487–505.
2. González-Granadillo G., González-Zarzosa S., Diaz R. Security information and event management (SIEM): Analysis, trends, and usage in critical infrastructures // Sensors. MDPI AG, 2021. Vol. 21, no. 14. P. 1–28.
3. VentureBeat. 2023 cybersecurity forecasts: Zero trust, cloud security will top spending. — https://venturebeat.com/security/2023-cybersecurity-forecasts-zero-trust-cloud-security-will-top-spending/, 2023.
4. Aslan Ö. A., Samet R. A comprehensive review on malware detection approaches // IEEE Access. Institute of Electrical; Electronics Engineers (IEEE), 2020. Vol. 8. P. 6249–6271.
5. Bryant B. D., Saiedian H. Improving SIEM alert metadata aggregation with a novel kill-chain based classification model // Computers & Security. Elsevier, 2020. Vol. 94. P. 1–23.
6. Boyko V., Vasilenko M., Slatvinska V. Linked list systems for system logs protection from cyberattacks / "Information technologies in education, science and technology" (ITEST-2022) june 23-25, 2022 cherkasy. 2022. P. 81–82.
7. Boyko V., Vasilenko M., Slatvinska V. Linked list systems for system logs protection from cyberattacks // Information technology for education, science, and technics / ed. by Faure E., Danchenko O., Bondarenko M., Tryus Y., Bazilo C., Zaspa G. Springer Nature Switzerland. P. 224–234.

8. Ozer M., Varlioglu S., Gonen B., Adewopo V., Elsayed N., Zengin S. Cloud incident response: Challenges and opportunities / 2020 international conference on computational science and computational intelligence (CSCI). IEEE, 2020. P. 49–54.
9. Aßmuth A., Duncan R., Liebl S., Söllner M. A secure and privacy-friendly logging scheme / Cloud computing 2021 / под ред. Bob Duncan, Yong Woo Lee, Manuela Popescu. — International Academy, Research,; Industry Association (IARIA), 2021. 8–12.
10. Angelo M. di, Durieux T., Ferreira J. F., Salzer G. Evolution of automated weakness detection in ethereum bytecode: A comprehensive study // Empirical Software Engineering. — Springer Science; Business Media LLC, 2024. Vol. 29, no. 2. P. 1–44.

Ключові слова: руткіти, виявлення зловмисного програмного забезпечення, зловмисне програмне забезпечення, кібератака, блокчейн, журнал, SIEM, перевірка, хеш, merkle trees
Keywords: rootkits, malware detection, malware, cyberattack, blockchain, log, SIEM, verification, hash, merkle trees

НАЛАШТУВАННЯ VIRTUALBOX ДЛЯ ДОСЛІДЖЕННЯ ТА АНАЛІЗУ MALWARE

Величканич Юрій
студент 4 курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Malware є постійною загрозою котра завдає шкоди як звичайним користувачам так і малим або великим компаніям, та з часом шкідливе програмне забезпечення стає все більш просунутим та більш агресивним, розробники шкідливого програмного забезпечення завжди вдосконалюють його та його методи протидії аналізу[1].

Дослідження та аналіз malware є дуже важливим аспектом в кібебезпеці котре допомагає з формуванням захисту проти шкідливого програмного забезпечення, та дає можливість зрозуміти по яким принципам працює шкідливе програмне забезпечення[2].

Віртуальна машина – це програмне забезпечення котре дозволяє створювати віртуальний пристрій, тобто імітувати роботу фізичного персонального комп'ютеру, тощо. Вони можуть запускати власні операційні системи та програмне забезпечення, що не залежить від системи хоста.

В першу чергу для дослідження шкідливого програмного забезпечення потрібно організувати віртуальне середовище, за для цього досить хорошим провайдером віртуального середовища є VirtualBox[3] так як це програмне забезпечення є безкоштовним та воно має відкритий код. Віртуальна машина – це

ізольоване від основного пристрою середовище котре можна використовувати для різних цілей, в тому числі і для дослідження шкідливе програмне забезпечення.

Аналіз актуальний, але утруднений тим, що malware захищається від спроб аналізу. Це є проблемою, оскільки виникає протиріччя. З одного боку для динамічного аналізу необхідно запустити malware в середовищі з мережним доступом - щоб звести нанівець міри malware із захисту від аналізу. З іншого боку потрібно запускати malware в ізольованому та безпечному оточенні, щоб не завдати шкоди системі аналітика. Для організації робочого місця динамічного аналізу malware оптимальне використання вкладених систем віртуалізації, при цьому може знадобитися автоматизація конфігурування та налаштування даних систем[4][5].

Перевагами використання саме VirtualBox є те що

– Ця віртуальна машина є ізольованою від системи, що в свою чергу дає перевагу перед поширенням шкідливе програмне забезпечення на основний пристрій.

– Віртуальна машина дає змогу створювати контрольовані середовища в котрих можна комфортно досліджувати шкідливе програмне забезпечення.

– VirtualBox є досить простий в використанню так як має досить простий та інтуїтивно зрозумілий інтерфейс, та це робить його досить доступним для різних користувачів з різним досвідом використання віртуальних машин.

– Також VirtualBox має досить широкий спектр можливостей бо підтримує досить багато різновидів операційних систем, вдосталь апаратних конфігурацій, а також дає можливість широкого налаштування мережевих конфігурацій для дослідження різних типів шкідливого програмного забезпечення.

Для початку роботи порібно завантажити VirtualBox а вже потім потрібно обрати операційну систему котра буде працювати зі стеком LAMP / LAPP / LNPP (L – Linux; A/N – Apache2 / Nginx; P/M – Postgres / MySQL; P – Perl / PHP / Python). Першочергово потрібно обрати дестрибутиву Linux на котрій буде комфортно працювати, на приклад є пару популярних варіантів, як Ubuntu чи Debian, вони є досить надійними варіантами для віртуалізації. Наступним кроком потрібно налаштувати сам стек, тобто інсталювати Apache2 або Nginx та Postgres або MySQL та Perl або PHP або Python.

Також після вище виконаних кроків, задля безпеки, важливим кроком є налаштування конфігурацій мережі, тому потрібно почати з підбору мережевого режиму, краще за все буде обрати NAT(*Network Address Translation*) mode, цей

мережевий режим створює віртуальну мережу для віртуальної машини зі своїм власним IP-адресом, та це вже забезпечує більшу ізоляцію, та дозволяє керувати трафік.

Також потрібно на основній віртуальній машині створити ще одне вкладене середовище віртуалізації, а саме знову завантажити програмне забезпечення котре буде забезпечувати віртуальне середовище, тобто VirtualBox, та налаштувати його мережеві конфігурації наступним чином, що потрібно організувати доступ до мережі але не давати доступ до основної системи, тож для цієї функції також підійде такий режим мережевої конфігурації як NAT(*Network Address Translation*) mode тому що як вище вказано він створює віртуальну мережу для віртуальної машини котре має свій особистий IP-адрес, що забезпечує ізоляцію від основної операційної системи

Список використаних джерел:

1. Jithin, R., & Chandran, P. (2014). Virtual machine isolation: A survey on the security of virtual machines. In Recent Trends in Computer Networks and Distributed Systems Security: Second International Conference, SNDS 2014, Trivandrum, India, March 13-14, 2014, Proceedings 2 (pp. 91-102). Springer Berlin Heidelberg.
2. Miwa, S., Miyachi, T., Eto, M., Yoshizumi, M., & Shinoda, Y. (2007). Design issues of an isolated sandbox used to analyze malwares. In Advances in Information and Computer Security: Second International Workshop on Security, IWSEC 2007, Nara, Japan, October 29-31, 2007. Proceedings 2 (pp. 13-27). Springer Berlin Heidelberg.
3. Oracle VM VirtualBox URL: https://www.virtualbox.org
4. Monovm. What is Nested Virtualization? 101 Guide URL: https://monovm.com/blog/nested-virtualization/
5. IEEE Xplore. What are virtual environments URL: https://ieeexplore.ieee.org/abstract/document/250914

Ключові слова: віртуалізація, аналіз malware, malware, аналіз мережної поведінки malware, вразливості, virtualbox

Keywords: virtualization, malware analysis, malware, analysis of network behavior malware, vulnerabilities, virtualbox

Науковий керівник: доцент В. Бойко

ВИКОРИСТАННЯ РЕКУРЕНТНИХ НЕЙРОННИХ МЕРЕЖ З ДОВГОЮ КОРОТКОЧАСНОЮ ПАМ'ЯТТЮ (LSTM) ДЛЯ НАДІЙНОГО ВИЯВЛЕННЯ ДОМЕНІВ АЛГОРИТМУ ГЕНЕРАЦІЇ ДОМЕНІВ (DGA)

Гуренко Марк

студент 4-го курсу факультету кібезбезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Домени, створенні алгоритмом генерації доменів (DGA) становлять значну загрозу кібербезпеці через їхню здатність уникати виявлення традиційними методами. Різні сімейства шкідливих програм використовують алгоритми генерації доменів (DGA) для створення численних псевдовипадкових доменних імен для підключення до серверів управління (C2). Щоб заблокувати цей трафік, організації з безпеки зазвичай проводять реінжиніринг шкідливого програмного забезпечення, щоб виявити алгоритм, а потім складають список доменів, які потім попередньо реєструються, блокуються, або додаються до чорного списку DNS. Однак цей процес є громіздким і легко оминається творцями шкідливого програмного забезпечення.

Альтернативним підходом є перехоплення DNS-запитів і визначення того, чи є домен згенерованим за допомогою DGA. Перший основний метод виявлення DGA полягає в аналізі груп схожих доменів і використанні їхніх статистичних властивостей. Однак ці методи повільні і не підходять для виявлення в режимі реального часу. Крім того, вони часто покладаються на дорогу інтеграцію та контекстну інформацію, таку як пасивний DNS, що може бути нездійсненним у багатьох системах, наприклад, для виявлення кінцевих точок [1].

Ще складнішою проблемою є виявлення генерації DGA на рівні домену без додаткової інформації. Початкові спроби в цьому напрямку показали низьку ефективність, оскільки вони значною мірою покладалися на імперативні механізми, які легко обходилися шкідливим програмним забезпеченням.

Новий підхід інтегрує машинне навчання і нейронні мережі глибокого навчання, зокрема мережі з довгою короткочасною пам'яттю (LSTM), для прогнозування DGA в реальному часі без потреби в контекстній інформації або створених вручну ознаках.

Мережі з довгою короткочасною пам'яттю (LSTM) – це особливий тип рекурентних нейронних мереж (RNN), які призначені для вирішення проблеми "зникаючих градієнтів" (vanishing gradients). Ця проблема може виникнути при навчанні RNN на довгих послідовностях даних, оскільки градієнти функції помилки можуть стати дуже малими, що ускладнює навчання мережі [3].

LSTM виявилися дуже ефективними для широкого спектру завдань, включаючи обробку природної мови, розпізнавання мови і підписів до зображень. Вони також добре підходять для завдань, які передбачають навчання на основі послідовних даних, таких як прогнозування часових рядів і виявлення аномалій.

До ключових переваг LSTM входять:

– можливість навчатися на довгих послідовностях даних, не страждаючи від проблеми зникаючих градієнтів [2];

– здатність вловлювати як короткострокові, так і довгострокові залежності в даних;

З іншого боку недоліком ресурсоємність навчання: воно може бути дуже дорогим з точки зору обчислювальних ресурсів, особливо для великих наборів даних. Крім того, результати навчання важко інтерпретувати, оскільки може бути важко зрозуміти, як LSTM приймає рішення.

Слід також зазначити, що така архітектура буває дуже чутлива до налаштувань гіперпараметрів, Від того модель потребує ретельного налаштовування гіперпараметрів під час навчання.

Основними ж відмінними рисами LSTM є:

– Довготривалі залежності: LSTM призначені для обробки довготривалих залежностей у послідовних даних за допомогою комірки пам'яті, яка може зберігати інформацію протягом тривалих періодів часу. Це особливо корисно для таких задач, як розпізнавання мови, машинний переклад та прогнозування часових рядів, де розуміння контексту має вирішальне значення [4, 5, 6];

– Зникаючі градієнти: LSTM можуть уникнути проблеми зникаючого градієнта, яка виникає у традиційних RNN, використовуючи механізм вентилювання, який вибірково запам'ятовує або забуває інформацію. Це дозволяє LSTM більш ефективно вивчати довгострокові залежності [7, 8];

– Обробка відсутніх даних: LSTM можуть обробляти відсутні дані в часовому ряді, використовуючи раніше вивчену інформацію для прогнозування. Це особливо корисно для задач, де дані є неповними або неструктурованними [7];

– Продуктивність: Доведено, що LSTM перевершують традиційні RNN у багатьох додатках, особливо в тих, що передбачають довгострокові залежності [7];

– Обробка складних часових залежностей: LSTM призначені для виявлення складних закономірностей у послідовних даних, що робить їх

особливо корисними для таких задач, як розпізнавання мови та машинний переклад [7].

На противагу цьому, GRU (Gated Recurrent Unit, Вентильний рекурентний вузол) є простішими та ефективнішими за LSTM, але можуть бути не настільки ефективними у виявленні довготривалих залежностей. PNN (Probabilistic neural network, Ймовірнісна нейронна мережа) – це імовірнісні моделі, які можуть обробляти послідовні дані, але без виявлення довгострокових залежностей. CNN (Convolutional neural network, Згорткова нейронна мережа) у першу чергу призначені для обробки просторових даних і не настільки ефективні в роботі з послідовними даними. KNN (k-nearest neighbors algorithm, Метод k-найближчих сусідів) – це тип алгоритму машинного навчання, який може обробляти послідовні дані; є непараметричною моделю, яка переважно використовується для задач класифікації та регресії проте не так ефективно, як LSTM. Decision Tree (Дерева рішень) і Random Forest (Випадкові ліси) – це ансамблеві методи, які можуть обробляти послідовні дані, однак вони не настільки потужні. Logistic Regression (Логістична регресія) – це лінійна модель, яка може обробляти послідовні дані, але не може впоратися з відсутніми даними в часовому ряді, використовуючи раніше вивчену інформацію для прогнозування. [4, 5, 6, 7, 8, 9].

Отже, мережі LSTM мають механізм вентилювання, який дозволяє їм вибірково запам'ятовувати або забувати інформацію, що дозволяє їм ефективно фіксувати довгострокові залежності в послідовних даних. Вони добре підходять для задач, де розуміння контексту довгих послідовностей має вирішальне значення. LSTM можуть обробляти вхідні послідовності довільної довжини, не страждаючи від проблеми зникаючого градієнта, яка характерна для традиційних RNN. їхня здатність зберігати релевантну інформацію протягом тривалих періодів часу робить їх потужним інструментом для моделювання складних часових зв'язків.

Список використаних джерел:

1. Woodbridge J. et al. Predicting domain generation algorithms with long short-term memory networks //arXiv preprint arXiv:1611.00791. – 2016.
2. Vij P., Nikam S., Bhatia A. Detection of algorithmically generated domain names using LSTM //2020 International Conference on COMmunication Systems & NETworkS (COMSNETS). – IEEE, 2020. – С. 1-6.
3. Weizhe Q. Y. Z. B. Z., Hualong A. K. S. W. DGA Domain Name Classification Method Based on Long Short-Term Memory with Attention Mechanism //Peng Cheng Laboratory Commumications. – 2020. – Т. 1. – №. 2. – С. 64.

4. What is the advantage of adding CNN to LSTM for forecasting sequential data? URL: https://ai.stackexchange.com/questions/35879/what-is-the-advantage-of-adding-cnn-to-lstm-for-forecasting-sequential-data
5. What is The Main Difference Between RNN and LSTM- (RNN vs LSTM). URL: https://www.theiotacademy.co/blog/what-is-the-main-difference-between-rnn-and-lstm/
6. Deep Learning | Introduction to Long Short Term Memory. URL: https://www.geeksforgeeks.org/deep-learning-introduction-to-long-short-term-memory/
7. What are the advantages and challenges of using LSTM or GRU for long-term dependencies in time series? URL: https://www.linkedin.com/advice/1/what-advantages-challenges-using-lstm-gru-long-term
8. CNN Long Short-Term Memory Networks. URL: https://machinelearningmastery.com/cnn-long-short-term-memory-networks/
9. Exploring the Potential of Long Short-Term Memory (LSTM) Networks in Time Series Analysis. URL: https://www.linkedin.com/pulse/exploring-potential-long-short-term-memory-lstm-networks-pandey/

Ключові слова: Алгоритм генерації доменів (DGA), довготривала короткочасна пам'ять (LSTM), рекурентні нейронні мережі (RNN), кібербезпека, машинне навчання, глибоке навчання

Keywords: Domain Generation Algorithm (DGA), Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNNs), cybersecurity, machine learning, deep learning

Науковий керівник: доцент В. Бойко

РОЛЬ ШТУЧНОГО ІНТЕЛЕКТУ У ТЕСТУВАННІ БЕЗПЕКИ ВЕБЗАСТОСУНКІВ

Дика Анастасія

аспірант кафедри інформаційних технологій
Національного університету «Одеська юридична академія»

Тестування безпеки вебзастосунків є важливою складовою забезпечення якості програмного забезпечення (ПЗ). З розвитком кіберзагроз, з одного боку, та зростанням складності вебзастосунків, з іншого, традиційні методи тестування безпеки подекуди виявляються недостатніми для ефективного виявлення вразливостей ПЗ. Останнім часом все частіше для тестування ПЗ пропонують застосовувати штучний інтелект (ШІ). Адже він пропонує нові підходи до генерації тестових даних, що може значно підвищити ефективність і точність тестування безпеки. Основні методи ШІ охоплюють машинне навчання (МН, Machine Learning, ML), генетичні алгоритми, еволюційні стратегії, агентні моделі та інші інноваційні

підходи. Використання ШІ для генерації тестових даних дозволяє автоматизувати процес створення складних, реалістичних сценаріїв тестування, що підвищує ефективність тестування безпеки.

Іншою важливою перевагою застосування ШІ у тестуванні ПЗ є реалістичність тестових даних. ШІ може створювати сценарії тестування, що імітують реальні користувацькі взаємодії, завдяки чому покращується якість тестування. Це допомагає виявляти вразливості, які можуть бути пропущені при використанні менш реалістичних тестових даних.

Ще однією перевагою є значне розширення покриття тестами. Використання ШІ дозволяє створювати велику кількість різноманітних тестових випадків, що підвищує ймовірність виявлення вразливостей. Це дуже важливо для комплексних вебзастосунків, де ручне тестування не може забезпечити достатнього покриття.

Крім того, ШІ забезпечує адаптивність тестових даних. Алгоритми ШІ можуть автоматично пристосовуватися до змін у вебзастосунках і створювати нові тестові випадки для виявлення нових вразливостей. Це робить тестування більш динамічним та ефективним у довгостроковій перспективі.

Водночас є певні виклики використання ШІ для генерації тестових даних. Один із них – це якість даних, адже ефективність ШІ значною мірою залежить від якості та обсягу тренувальних даних. Недостатня кількість даних або дані низької якості можуть призвести до неточних або неефективних тестових випадків.

Ще одним викликом є складність моделей. Алгоритми машинного навчання можуть бути доволі складними для розуміння та налаштування, що вимагає висококваліфікованих фахівців. Це може стати бар'єром для невеликих команд розробки або компаній, які не мають доступу до відповідних ресурсів.

Фальшиві спрацьовування – ще один важливий виклик. ШІ-системи можуть генерувати фальшиві спрацювання (позитиви), що потребує додаткових ресурсів для перевірки та аналізу. Це може спричинити додаткові витрати часу та зусиль зі зниженням загальної ефективності тестування.

На сьогодні науковці шукають нові підходи до генерації тестових даних за допомогою ШІ та МН задля покращення ефективності та точності генерації тестових даних для тестування безпеки вебзастосунків.

МН є доволі ефективною технологією для генерації тестових даних. Алгоритми МН можуть аналізувати великі обсяги даних та виявляти патерни, які можуть бути використані для створення нових тестових випадків. В дослідженні [1] оцінюються можливості генеративного ШІ для формування тестових даних у

різних доменах. Сформовано три типи запитів до великих мовних моделей (LLMs), які виконують завдання генерації тестових даних на різних рівнях інтеграції: 1) генерація сирих тестових даних; 2) синтез програм конкретною мовою, які генерують корисні тестові дані; 3) створення програм, які використовують сучасні бібліотеки-фейкери. Застосування глибокого навчання для виявлення вразливостей дозволило отримати результати, значно вищої точності, аніж результати тестування, досягнуті традиційними методами.

Генетичні алгоритми (ГА, Genetic Algorithm) є еволюційними методами, які використовуються для пошуку оптимальних рішень за допомогою природного відбору та мутацій. ГА можуть бути ефективно застосовані для автоматичної генерації тестових даних, оскільки вони можуть створювати складні, непередбачувані сценарії тестування з урахуванням вразливостей у вебзастосунках. Системи з ГА для створення складних сценаріїв атаки виявилися ефективними для виявлення вразливостей, які не були виявлені традиційними методами [2].

Використання агентних моделей (Agent-based Models) є ще одним підходом, що використовується для генерації тестових даних. Ця техніка моделювання створює віртуальну екосистему агентів, взаємодія яких породжує складні структури даних, тим самим віддзеркалюючи їхні людські аналоги в реальних програмах [3]. У контексті тестування безпеки вебзастосунків агентні моделі можуть бути використані для симуляції поведінки різних типів користувачів та потенційних зловмисників. Це дозволяє створювати реалістичні сценарії взаємодії з вебзастосунком, що допомагає виявляти вразливості, які можуть бути пропущені традиційними методами.

Отже, методи ШІ починають відігравати важливу роль в автоматизації генерації тестових даних для тестування безпеки вебзастосунків. Використання МН, ГА та агентних моделей дозволяє покращити ефективність і точність генерації тестових даних. Попри певні виклики щодо якості даних та складності моделей, які потребують вивчення та врахування, переваги використання ШІ у тестуванні безпеки є вагомими. Подальші дослідження та розробки у цій галузі сприятимуть створенню більш інтелектуальних та адаптивних систем захисту, які можуть ефективно протистояти сучасним кіберзагрозам.

Список використаних джерел:

1. Baudry B., Etemadi K., Fang S., Gamage Y., Liu Y., Liu Y. Generative AI to Generate Test Data Generators. arXiv e-prints, arXiv-2401. 2024. DOI: https://doi.org/10.48550/arXiv.2401.17626. URL: https://arxiv.org/pdf/2401.17626.

2. Esnaashari M., Damia A. H. Automation of software test data generation using genetic algorithm and reinforcement learning. Expert Systems with Applications. 2021. Vol. 183. Issue C. P. 115446. DOI: https://doi.org/10.1016/j.eswa.2021.115446
3. Clark A., Walkinshaw N., Hierons R. Test case generation for agent-based models: a systematic literature review. Information and Software Technology. 2021. Vol. 135. P. 106567. DOI: https://doi.org/10.1016/j.infsof.2021.106567.

Ключові слова: тестування безпеки, тестування програмного забезпечення, штучний інтелект, генерація тестових даних, машинне навчання, генетичні алгоритми, агентні моделі
Keywords: security testing, software testing, artificial intelligence, test data generation, machine learning, genetic algorithm, agent-based models

Науковий керівник: доцент О. Трофименко

СТВОРЕННЯ ВІРТУЛЬНОЇ ЛАБОРАТОРІЇ ДЛЯ МОДЕЛЮВАННЯ ПОТЕНЦІЙНИХ АТАК

Дрига Артем
студент 4-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

У житті більшості людей технології посідають важливе місце, наприклад це може бути спілкування в соціальних мережах, замовлення товарів через інтернет або ж робота з цифровими документами. Усе це значно спрощує щоденні дії людини, але водночас створює нові виклики для збереження конфіденційності, доступності та цілісності інформації, адже чим більше суспільство залежне від технологій - тим більше воно вразливе перед зловмисниками.

Для того, щоб захистити інформацію від несанкціонованого втручання, фахівцям з кібербезпеки потрібно не тільки налаштовувати захисні інструменти, а й використовувати більш проактивні методи, такі як моделювання потенційних дій зловмисника. Для цих цілей створюються спеціальні віртуальні лабораторії, які складаються з систем з підготовленими інструментами і навмисно вразливих цілей для імітування атак.

Для створення такої лабораторії було обрано:

1. Дистрибутив Linux під назвою Kali Purple як операційна система для моделювання атак. У той час, як Kali Linux призначений насамперед для проведення тестувань на проникнення, Kali Purple крім інструментів для атаки має в собі велику кількість засобів для захисту [1];

2. Віртуальну машину Metasploitable у ролі жертви для атак. Metasploitable являє собою навмисно вразливий сервер на Linux [2];

3. З метою захисту лабораторії від можливих зовнішніх атак на вразливу машину було також підключено дистрибутив pfSense, який виступає в ролі міжмережевого екрану [3]. З його допомогою було налаштовано внутрішню мережу, яка з'єднала між собою всі три віртуальні машини;

4. Гіпервізор VirtualBox для налаштування та роботи з віртуальними машинами [4].

Як вже було зазначено, pfSense був використаний для створення окремої внутрішньої мережі, до якої підключені всі машини.

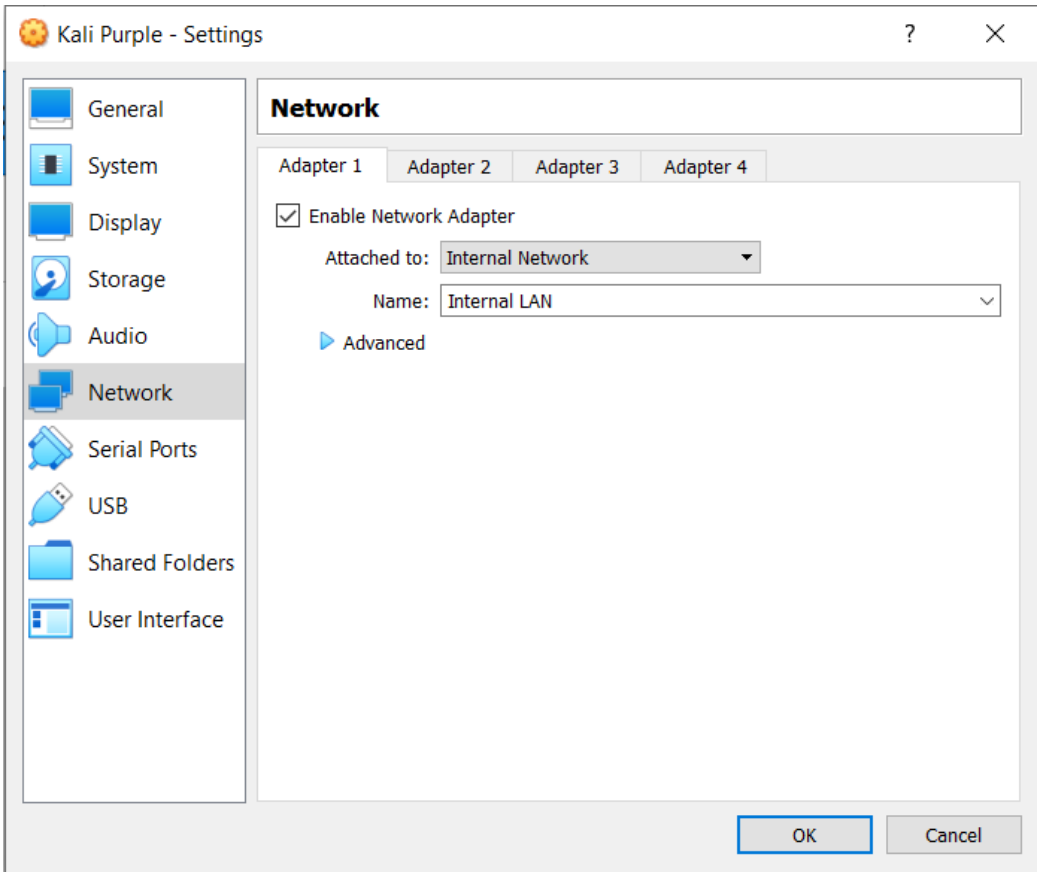


Рисунок 1 – Kali Purple, що підключений до внутрішньої мережі

Наступним кроком буде визначення IP-адреси машини Metasploitable. За допомогою Kali Purple було виконано команду netdiscover, яка просканувала мережу та вивела список адрес внутрішньої мережі.

```
Currently scanning: 192.168.10.0/16   |   Screen View: Unique Hosts

2 Captured ARP Req/Rep packets, from 2 hosts.   Total size: 120

  IP            At MAC Address     Count     Len  MAC Vendor / Hostname

192.168.1.1     08:00:27:15:1c:bf      1      60  PCS Systemtechnik GmbH
192.168.1.102   08:00:27:58:17:8a      1      60  PCS Systemtechnik GmbH
```

Рисунок 2 – Результат роботи netdiscover

В списку є дві адреси. У більшості випадків, менша IP-адреса - це мережевий екран або маршрутизатор, тому перша адреса була визначена як pfSense, а друга адреса - як Metasploitable.

Тепер потрібно перевірити, чи все було правильно налаштовано. Для підтвердження цього в браузері Kali Purple було введено 192.168.1.102.

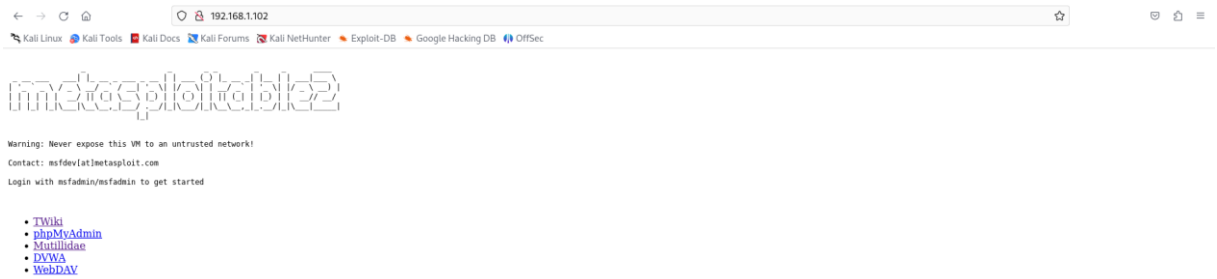


Рисунок 3 – Головна сторінка Metasploitable

Таким чином було доведено, що віртуальна лабораторія була правильно налаштована, що дає змогу в подальшому її використовувати для проведення моделювання атак.

Список використаних джерел:

1. Kali Linux 2023.1 Release (Kali Purple & Python Changes) [Електронний ресурс]. – Режим доступу: https://www.kali.org/blog/kali-linux-2023-1-release/#kali-purple
2. Metasploitable - Browse /Metasploitable2 at SourceForge.net [Електронний ресурс]. – Режим доступу: https://sourceforge.net/projects/metasploitable/files/Metasploitable2/
3. pfSense® - World's Most Trusted Open Source Firewall net [Електронний ресурс]. – Режим доступу: https://www.pfsense.org/
4. Oracle VM VirtualBox [Електронний ресурс]. – Режим доступу: https://www.virtualbox.org/

Ключові слова: Kali Purple, Metasploitable, pfSense, VirtualBox
Keywords: Kali Purple, Metasploitable, pfSense, VirtualBox

Науковий керівник: доцент В. Бойко

АТАКИ БАЗ ДАНИХ ЗА ДОПОМОГОЮ ІН'ЄКЦІЙ

Жук Максим

студент 4-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

У сучасному світі бази даних (БД) відіграють критичну роль у функціонуванні практично будь-якої інформаційної системи. Вони містять величезні обсяги конфіденційної інформації, що робить їх пріоритетною мішенню для зловмисників. Захист даних від несанкціонованого доступу, модифікації чи видалення є ключовим завданням для забезпечення інформаційної безпеки будь-якої організації.

Однією з найпоширеніших загроз для БД є ін'єкції. Ін'єкція - це метод атаки, при якому зловмисник вводить шкідливий код у запит до БД. Це може призвести до отримання несанкціонованого доступу до конфіденційної інформації, зміни або видалення даних, а також до порушення працездатності всієї системи.

Найпоширеніші типи ін'єкцій:

– SQL-ін'єкції: Цей тип атаки націлений на БД, керовані системами управління базами даних (СУБД), такими як MySQL, PostgreSQL, Oracle.

– XSS-ін'єкції: Цей тип атаки націлений на веб-додатки, які використовують JavaScript для обробки введених користувачами даних.

– LDAP-ін'єкції: Цей тип атаки націлений на служби каталогів, такі як Active Directory.

В прикладі приведемо роботу «найпопулярнішої» ін’єкційної атаки – SQL-ін’єкції. SQL-ін'єкція - це метод атаки, під час якого зловмисник вводить шкідливий SQL-код у поле введення веб-додатка.

Розглянемо веб-сайт, який запитує у користувача ім'я користувача та пароль для входу в систему. Зловмисник може ввести наступний текст у поле "Ім'я користувача":

' OR 1=1; --

У результаті цього код SQL-запиту, який генерує веб-додаток, виглядатиме так:

SELECT * FROM users WHERE username = '' OR 1=1; --' AND password = 'password';

Через оператор OR ця конструкція завжди повертатиме TRUE, що дасть змогу зловмиснику отримати доступ до будь-якого запису в таблиці users.

Статистика по атакам на БД

– Згідно зі звітом компанії Verizon, у 2022 році 43% усіх витоків даних було пов'язано з веб-додатками, а 22% - із SQL-ін'єкціями [1].

– За даними компанії Positive Technologies, у 2022 році кількість веб-вразливостей, пов'язаних із SQL-ін'єкціями, зросла на 13% [2].

– Дослідження компанії Imperva показує, що в середньому на одну компанію припадає 144 SQL-вразливості [3].

Існує безліч методів захисту баз даних від ін'єкцій.

Найбільш ефективні методи:

– Використання параметризованих запитів: — цей метод дає змогу відокремлювати користувацьке введення від SQL-коду, що унеможливлює впровадження шкідливого коду.

– Веб-фільтрація: — допомагає блокувати шкідливий код на рівні веб-додатка.

– Обмеження прав доступу — необхідно надавати користувачам тільки ті права доступу до БД, які їм необхідні.

– Шифрування даних — допомагає захистити дані від несанкціонованого доступу в разі компрометації БД.

– Регулярне оновлення ПЗ — важливо своєчасно встановлювати оновлення безпеки для СУБД і веб-додатків.

Додаткові методи захисту баз даних:

– Застосування брандмауерів і систем запобігання втручанням (IDS/IPS): Ці системи допомагають блокувати несанкціонований доступ до БД ззовні та зсередини мережі.

– Використання маскування даних: Цей метод приховує конфіденційну інформацію, роблячи її нечитабельною для несанкціонованих осіб.

– Резервне копіювання та відновлення даних: Регулярне створення резервних копій даних БД дає змогу відновити їх у разі збою або атаки.

– Контроль і аудит доступу: Слід постійно відстежувати й аналізувати активність користувачів у БД, щоб виявляти підозрілу діяльність.

– Навчання персоналу: Важливо, щоб усі співробітники, які мають доступ до БД, були обізнані про загрози безпеці та знали, як правильно захищати дані.

Важливо зазначити, що не існує єдиного універсального методу захисту баз даних. Найефективніший підхід до захисту залежатиме від конкретних потреб та особливостей організації. Тому рекомендується використовувати комплексний підхід до захисту, який поєднує в собі кілька різних методів.

Захист баз даних - це важливий аспект інформаційної безпеки.

Використовуючи комплексний підхід до захисту, який включає в себе використання різних методів захисту, можна значно підвищити рівень безпеки БД і знизити ризики витоку даних.

Список використаних джерел:

1. Verizon Data Breach Investigations Report 2022 URL: https://www.verizon.com/business/resources/reports/dbir/
2. Positive Technologies Global Security Report 2023 URL: https://www.ptsecurity.com/ww-en/analytics/cybersecurity-threatscape-2023-q3/
3. Imperva Cyber Attacks Landscape Report 2023 URL: https://www.imperva.com/resources/resource-library/reports/ddos-threat-landscape-report-2023/
4. Web Application Firewall Based on Anomaly Detection using Deep Learning URL: https://cdn.istanbl.edu.tr/file/JTA6CLJ8T5/10AB7DC605CB478CBE21104E72E95F5D
5. Let's talk about SQL Injection URL: https://www.zerocopter.com/articles/lets-talk-about-sql-injection

Ключові слова: пентест, ризики, кібербезпека, SQL-ін'єкція, XSS, LDAP, база даних
Keywords: pentest, risks, cybersecurity, SQL injection, XSS, LDAP, database

Науковий керівник: доцент В. Бойко

WORDPRESS, ТА ЙОГО БЕЗПЕКА

Касян Павло

студент 4 курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

WordPress є відкритою платформою для створення сайтів, яка починала свій шлях як проста система для ведення блогів, але з часом перетворилася на одну з найпопулярніших систем управління контентом у світі. Це дозволяє користувачам створювати і управляти сайтами різного ступеня складності, від простих блогів до складних корпоративних порталів і інтернет-магазинів.

Однією з головних переваг WordPress є його гнучкість і масштабованість. Завдяки великій кількості тем і плагінів, кожен може адаптувати сайт під свої потреби без глибоких знань у програмуванні. Теми визначають вигляд сайту, тоді як плагіни додають нові функції або розширюють існуючі можливості.

Крім того, WordPress відомий своєю спільнотою користувачів та розробників, яка постійно працює над удосконаленням платформи. Ця спільнота допомагає користувачам з розв'язанням проблем, обміном знаннями та розробкою

нових інструментів.

З точки зору SEO, WordPress має вбудовані засоби оптимізації, які можуть допомогти сайтам краще ранжуватися в пошукових системах. Це включає підтримку SEO-дружніх URL, наявність плагінів для оптимізації контенту та зображень, а також можливості для налаштування метаданих і тегів.

Щодо безпеки, WordPress також пропонує ряд рішень для захисту веб-сайтів. Регулярні оновлення, включаючи патчі безпеки, допомагають захистити сайти від вразливостей. Важливо регулярно оновлювати ядро системи, плагіни та теми, а також використовувати надійні інструменти для захисту від злому і шкідливих програм.

Безпека є однією з найважливіших складових при розробці та управлінні веб-сайтами на WordPress. Через свою популярність, WordPress часто стає ціллю для хакерів, тому розробка безпечного сайту вимагає ретельного підходу та реалізації ряду захисних заходів.

Перш за все, основа безпеки сайту на WordPress починається з вибору надійного хостингу. Надійні хостинг-провайдери пропонують різноманітні технології безпеки, включаючи резервне копіювання даних, автоматичні оновлення програмного забезпечення та захист від DDoS-атак.

Другий важливий крок – це регулярне оновлення WordPress, тем та плагінів. Кожне оновлення часто містить виправлення помилок та покращення безпеки, що можуть запобігти потенційним вразливостям. Залишати старі версії може відкрити шляхи для атак.

Крім технічних аспектів, важливо також встановити сильні паролі та використовувати двофакторну аутентифікацію для захисту адміністративного доступу до сайту. Ці прості кроки можуть значно зменшити ризик несанкціонованого доступу.

Для додаткового захисту можна встановити спеціалізовані плагіни безпеки, які здійснюють моніторинг і блокування підозрілих активностей, сканування на наявність шкідливих кодів та виправлення вразливостей. Плагіни, такі як Wordfence або iThemes Security, забезпечують комплексний захист та можуть автоматично реагувати на загрози.

Також варто розглянути застосування SSL сертифіката, який не тільки шифрує дані, передані між користувачем та сайтом, але й покращує довіру користувачів та може підвищити позиції сайту в пошукових системах.

Плагіни для WordPress грають ключову роль у розширенні функціоналу і забезпеченні безпеки сайту. Вони додають нові можливості, оптимізують роботу

сайту, а також допомагають захистити його від зовнішніх загроз. Щодо безпеки, існує цілий ряд плагінів, які можуть значно підвищити захист вашого сайту на WordPress [1]. Існує декілька плагінів безпеки:

1. **Wordfence Security** – один з найбільш відомих і широко використовуваних плагінів безпеки для WordPress, який включає функції, такі як фаєрвол, сканер вірусів, інструмент для блокування IP-адрес та здатність виявляти і блокувати шкідливий трафік. Wordfence також моніторить всі спроби входу на сайт і надсилає сповіщення при виявленні підозрілої активності.

2. **Sucuri Security** – плагін, який забезпечує комплексний захист, включаючи моніторинг безпеки, сканування на шкідливі програми та журнал безпеки для відстеження всіх змін, що відбуваються на сайті. Цей плагін відомий своєю здатністю відновлювати сайти, що були зіпсовані або заражені.

3. **iThemes Security** (раніше відомий як Better WP Security) – плагін, який надає понад 30 способів захисту сайтів. Він змінює URL-адреси для областей входу та адміністрування, блокує небажані спроби входу та зміцнює файл .htaccess для підвищення безпеки.

4. **All In One WP Security & Firewall** – плагін, що пропонує легке в налаштуванні, але потужне рішення для захисту WordPress сайтів. Включає функції, які зменшують ризик хакерських атак, захищають облікові записи користувачів, блокують шкідливі запити і забезпечують захист файлів.

5. **Jetpack** – багатоцільовий плагін, що включає різноманітні функції для збільшення продуктивності, соціальних мереж та контенту. Також плагін пропонує важливі інструменти безпеки, такі як захист від брутфорс-атак, сканування на шкідливі програми та автоматичні резервні копії.

Перелічені плагіни не тільки допомагають захистити сайт від потенційних загроз, але й забезпечують інструменти для реагування на інциденти безпеки, що дозволяє швидко відновити нормальну роботу вашого сайту у разі атаки.

Однак, слід більш детально розглянути плагін All In One WP Security & Firewall [2], який забезпечує широкий спектр захисних заходів, спрямованих на підвищення безпеки сайту. Він розроблений таким чином, щоб бути доступним для користувачів різного рівня технічних знань і пропонує зрозумілі інструкції та налаштування, які можна легко адаптувати до потреб конкретного сайту.

Основними функціями та можливостями цього плагіна є:

Базова безпека

Захист облікових записів: Плагін має можливість перевіряти слабкі паролі користувачів та змушувати їх використовувати сильні паролі, що підвищує

загальний рівень безпеки акаунтів на сайті.

Блокування користувачів: Функція блокує IP-адреси, які здійснюють надто багато невдалих спроб входу, що допомагає запобігти атакам брутфорс.

Розширена безпека

Файл .htaccess: Плагін може автоматично генерувати та оптимізувати файл .htaccess, що додає додатковий рівень захисту на серверному рівні.

Файл wp-config.php: Забезпечує захист основного конфігураційного файлу WordPress, що є критично важливим для забезпечення безпеки сайту.

Безпека файлів

Сканер файлів: Плагін сканує і знаходить файли, що були змінені або є підозрілими, та дозволяє вам швидко реагувати на можливі загрози.

Захист від виконання PHP: У певних каталогах сайту блокується можливість виконання PHP-скриптів, що знижує ризик виконання шкідливих скриптів.

Брандмауер (Firewall)

Базові налаштування брандмауера: Захист від різних видів атак, таких як SQL-ін'єкції та кросс-сайт скриптинг.

Розширені правила брандмауера: Надає можливість налаштувати більш специфічні правила для захисту від складніших атак.

Безпека бази даних

Захист бази даних: Автоматичне перейменування префіксу таблиць бази даних для ускладнення процесу їх вгадування потенційними зловмисниками.

Резервне копіювання: Можливість створювати резервні копії бази даних прямо з плагіна, що дозволяє швидко відновити дані у випадку їх втрати або пошкодження.

Візуальний режим охорони

Security Meter: На головній панелі плагіну відображається індикатор рівня безпеки, який змінюється в залежності від того, скільки захисних заходів впровадили.

Отже, плагін All In One WP Security & Firewall є відмінним вибором для власників сайтів на WordPress, які хочуть підвищити рівень безпеки своїх ресурсів без необхідності глибоких технічних знань. Завдяки своїм широким можливостям, плагін забезпечує один з найкращих рівнів захисту в рамках доступних плагінів для WordPress.

Список використаних джерел:

1. Поради щодо безпеки WordPress: 10 способів захистити свій сайт. URL: http://surl.li/uldxa

2. All-In-One Security (AIOS) – Security and Firewall. URL: https://wordpress.org/plugins/all-in-one-wp-security-and-firewall/

Ключові слова: WordPress, плагін, безпека, сайт
Keywords: WordPress, plugin, security, site

Науковий керівник: доцент Н. Логінова

ЗАХОДИ ЗАХИСТУ ІНФОРМАЦІЇ В ІНФОРМАЦІЙНИХ СИСТЕМАХ

Кухаренко Сергій
кандидат технічних наук, доцент, доцент кафедри інформаційних технологій Національного університету «Одеська юридична академія»

Розвиток інформаційних технологій збільшує кількість можливих атак на інформаційні системи. Для забезпечення ефективного протидії неконтрольованому витоку та несанкціонованому доступу до інформації, а також для зменшення збитків від таких загроз, необхідно систематично розробляти заходи та обирати засоби захисту інформації. Особливо важливо мати розуміння основних законодавчих положень у цій області та уміти ефективно впроваджувати організаційні, програмні, технічні та інші заходи забезпечення інформаційної безпеки.

Тому проектування систем захисту інформації та вибір засобів захисту потребують детального аналізу загроз і ризиків. Це складне завдання, яке вимагає врахування різних типів потенційних загроз, витрат на впровадження та інтересів різних зацікавлених сторін.

Процес аналізу загроз є важливим кроком у забезпеченні захисту інформаційних систем. Цей аналіз дозволяє визначити потенційні загрози та їхній вплив на інформацію, що обробляється системою. Засоби захисту інформації слід обирати лише після детального аналізу ризиків і загроз. Це дозволяє об’єктивно оцінити наслідки реалізації загроз і збитки від можливих втрат і визначити найбільш прийнятні методи і засоби забезпечення належного рівня безпеки інформаційних систем організації. [1].

Розглядаючи загальні принципи, доречно зазначити, що комплексна система захисту передбачає використання специфічних правових, фізичних та організаційних аспектів та програмно-технічні засоби, призначені для забезпечення ідентифікації та автентифікації користувачів, розподілу прав доступу до

технічних інформаційних ресурсів і сервісів, а також реєстрації та фіксації спроб несанкціонованого доступу.

Організаційні заходи зазвичай спрямовані на чіткий розподіл обов'язків персоналу в процесах обробки інформації, створення кількох ліній контролю, запобігання зовнішнім і внутрішнім загрозам, навмисному чи випадковому знищенню та модифікації інформації.

Відповідно до чинного законодавства технічному захисту підлягає інформація, що становить державну чи іншу таємницю.

Захист інформації спрощено відбувається в кілька етапів: перший етап – виявлення та аналіз загроз; другий – розвиток ІТ-системи; третій – реалізація плану захисту; четвертий – контроль функціонування та управління системою захисту.

На першому етапі проводиться комплексний аналіз об'єктів захисту, ситуаційного плану та умов функціонування інформаційної системи для оцінки можливості виникнення загроз та очікуваних втрат після її реалізації та підготовки даних для побудови моделі загроз.

Загрози можуть здійснюватися через: технічні канали, у тому числі через побічні канали електромагнітного випромінювання і наведення, акустичні, оптичні, радіо-, радіотехнічні, хімічні тощо, а також канали спеціального впливу за рахунок формування полів і сигналів з метою руйнування захисних систем або порушення цілісності інформації.

Отримання несанкціонованого доступу здійснюється шляхом підключення пристроїв до ліній зв'язку, маскування під зареєстрованих користувачів, подолання заходів захисту веб-ресурсів, використання закладних пристроїв і програм, комп'ютерних вірусів тощо.

На другому етапі розробляється план, який містить організаційні, технічні заходи захисту ІзОД, визначаються зони безпеки інформації. Організаційні заходи регламентують порядок інформаційної діяльності з урахуванням норм і вимог технічного захисту для всіх періодів життєвого циклу інформаційної системи [2].

Технічні заходи повинні бути адекватними до загроз та розроблені з урахуванням можливих збитків, а також забезпечувати задану ефективність протягом всього життєвого циклу системи. Вони передбачають блокування загроз як без використання засобів технічного захисту так і навпаки. Мінімально необхідний рівень захисту забезпечують обмежувальними і фрагментарними заходами протидії найнебезпечнішій загрозі.

На третьому етапі мають бути здійснені організаційно-технічні заходи щодо захисту інформації з обмеженим доступом, створені необхідні зони захисту інформації та сертифіковані технічні засоби забезпечення відповідності інформаційної діяльності вимогам безпеки інформації [3].

На четвертому етапі здійснюється контроль за функціонуванням та управлінням системою захисту на об'єктах інформаційної діяльності з метою визначення й удосконалення стану захисту, виявлення та запобігання порушенням системи захисту.

Отже, на даний час існує досить багато, як теоретичних підходів так і розвинутих моделей за допомогою яких можливо описати практично всі аспекти системи захисту інформації, а методи та заходи захисту інформації включають в себе організаційні, технічні та правові заходи, що вживаються при процесі побудови та функціонування системи для забезпечення захисту інформації.

Список використаних джерел:

1. Про Концепцію (основи державної політики) національної безпеки України: Постанова Верховної Ради України від 16.01.1997. Поточна редакція від 21.12.2000. URL: http://surl.li/tvmwt
2. ДСТУ 3396.2-97 Захист інформації. Технічний захист інформації. Основні положення. [Чинний від 1997.01.01]. Київ, 1997. 6с. URL: https://tzi.com.ua/downloads/DSTU%203396.0-96.pdf
3. Про затвердження Правил забезпечення захисту інформації в інформаційних, електронних комунікаційних та інформаційно-комунікаційних системах: Постанова КМУ від 29.03.2006 № 373. Поточна редакція від 10.21.2022. URL: https://zakon.rada.gov.ua/laws/show/373-2006-%D0%BF#Text

Ключові слова: захист інформації, технічний захист інформації, безпека даних, інформаційні системи, комплексна система захисту інформації

Keywords: information protection, technical information protection, data security, information systems, comprehensive information protection system

МЕТОД ВБУДОВУВАННЯ ПОВІДОМЛЕНЬ У ВІДЛІКИ ІМПУЛЬСНО-КОДОВОЇ МОДУЛЯЦІЇ АУДІОСИГНАЛУ

Овчинников Микола

аспірант кафедри кібербезпеки
Національного університету «Одеська юридична академія»

В даний час існує безліч методів стеганографії приховування інформації в аудіо файлах. Методи відрізняються складністю реалізації та різним рівнем

стійкості до атак. Розглянемо один із можливих методів, який дозволяє приховати інформаїцю в аудіофайлі, наприклад, це буде інформація про автора музичного твору. Це метод вбудовування повідомлень у відліки імпульсно-кодової модуляції аудіосигналу з використанням двопозиційної частотної маніпуляції.

Цифровий запис аудіосигналів заснований на виконанні двох ключових операцій над аналоговими сигналами: дискретизації та квантування. У реальних пристроях ці операції виконуються одночасно. В результаті дискретизації аналоговий аудіосигнал перетворюється на послідовність відліків, а внаслідок квантування кожен відлік замінюється послідовністю бітів. Таким чином, початковий аналоговий сигнал перетворюється на масив цілих чисел, кожне з яких обмежене числом розрядів, що дорівнює кількості бітів квантування.

Для відновлення аналогової форми сигналу з цього масиву необхідно знати кількість рівнів квантування та інтервал дискретизації.

Часто цифрове представлення аудіосигналу називають імпульсно-кодовою модуляцією аудіосигналу [1]. Це пов'язано з тим, що грубу копію аудіосигналу можна представити як послідовність прямокутних імпульсів однакової тривалості, яка дорівнює інтервалу дискретизації, при цьому амплітуда поточного імпульсу дорівнює поточному значенню елемента масиву цифрового запису.

Пропонується використовувати масив відліків аудіосигналів як контейнер для вбудовування повідомлень. Враховуючи різноманітність додатків, приховані повідомлення можуть мати різну природу і структуру, тому домовимося, що будь-яке повідомлення перед вбудовуванням у контейнер перетворюється на послідовність бітів [2-3].

Єдине обмеження, що висувається до цієї послідовності, — це кількість її елементів, яке обмежується ємністю контейнера і може варіюватися залежно від методу вбудовування.

При частотної маніпуляції кожному можливому значенню символу, що передається, зіставляється своя частота (рис. 1). Протягом кожного символьного інтервалу передається гармонійне коливання із частотою, що відповідає поточному символу [4-5].

Сам алгоритм вбудовування можна описати в такий спосіб. Аудіофайл-контейнер розбивається на відрізки, що містять кількість звітів, кратне ступеня двійки. Проводиться швидке перетворення Фур'є відрізка. Для вбудовування даних використовують вузькі смуги частот поблизу вибраних частот. Для цих вузьких смуг оцінюється енергія, виходячи з якої вибирається амплітуда модульованого повідомлення, що приховується.

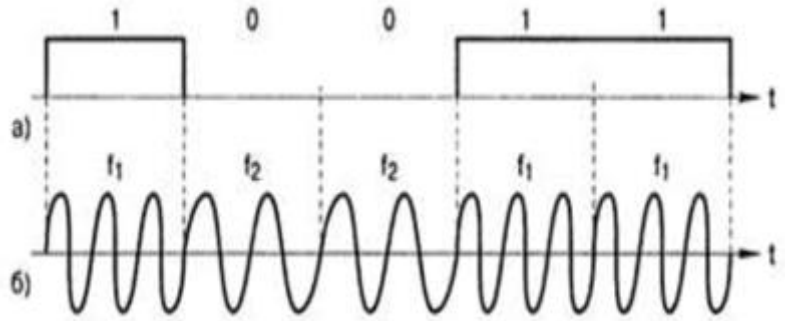


Рисунок 1 – Форма сигналу при частотної маніпуляції:
(а) маніпулюючий сигнал;
(б) частотно-маніпулюючий сигнал - радіосигнал частотної маніпуляції

Модульоване повідомлення формується так:

1) вибираються дві частоти, що лежать усередині смуг вбудовування;

2) оцінюється тривалість вбудовування одного біта повідомлення - в цей час має укластися кілька періодів частоти вбудовування;

3) формується додаткове повідомлення: вбудовуванню одиниці відповідає коливання нижньої частоти зі смуги вбудовування, а нулю - верхньої частоти зі смуги вбудовування;

4) амплітуда модульованого повідомлення вибирається такою, щоб енергія повідомлення, що вбудовується, відповідала енергії контейнера в смузі вбудовування.

Після формування повідомлення смуги вбудовування режектуються із спектру вихідного відрізка та проводиться зворотне перетворення Фур'є. Далі отримані відліки сумуються з відліками модульованого повідомлення. Ширина смуг вбудовування, а також частоти вбудовування оцінюються виходячи з необхідної швидкості передачі біт повідомлення, що приховується, а також з міркувань непомітності вбудовування.

Для отримання прихованого повідомлення необхідно знати довжину відрізка вбудовування, частоту вбудовування, а також ширину смуг вбудовування. Аудіофайл-стегоконтейнер також розбивається на відрізки відомої довжини та виробляється швидке перетворення Фур'є. Аналізуються смуги частот поблизу смужок вбудовування. Смуги вбудовування відфільтровуються, і проводиться зворотне перетворення Фур'є отриманого вузькосмугового сигналу. Таким чином отримуємо модульоване повідомлення. Далі оцінюється тривалість передачі одного біта повідомлення. Модульоване повідомлення розбивається на відрізки передачі одного біта, усередині кожного з яких оцінюється частота і за цією

частотою приймається рішення про переданий біт. Таким чином приймається повідомлення, що приховується.

Для отримання прихованого повідомлення необхідно знати довжину відрізка вбудовування, частоту вбудовування, а також ширину смуг вбудовування. Аудіофайл-стегоконтейнер також розбивається на відрізки відомої довжини та виробляється швидке перетворення Фур'є. Аналізуються смуги частот поблизу смужок вбудовування. Смуги вбудовування відфільтровуються, і проводиться зворотне перетворення Фур'є отриманого вузькосмугового сигналу. Таким чином отримуємо модульоване повідомлення. Далі оцінюється тривалість передачі одного біта повідомлення. Модульоване повідомлення розбивається на відрізки передачі одного біта, усередині кожного з яких оцінюється частота і за цією частотою приймається рішення про переданий біт. Таким чином приймається повідомлення, що приховується.

Список використаних джерел:

1. Torrieri, D. 2005. Principles of spread-spectrum communication systems. Boston, MA: Springer Science. 456 p.
2. Bender, W., B. Gruhl, N. Morimoto, and A. Lu. 1996. Techniques for data hiding. IBM Syst. J. 35(3).
3. Nishimura, R. Audio watermark based on periodical phase shift [Text] / R. Nishimura, Y. Suzuki // J. Acoust. Soc. Japan. – 2004. – Vol. 60. – No 5. – P. 268–272.
4. Кубів Р. В. Дослідження адаптивної диференціальної імпульсно-кодової модуляції голосового сигналу : дис. – Тернопільський національний технічний університет імені Івана Пулюя, 2013.
5. Лукьянчіков С. Дослідження сучасних методів стиснення звукової інформації з втратами та без втрат //Геометричне моделювання та інформаційні технології. – 2017. – №. 2. – С. 62-67.

Ключові слова: стеганографія, аудіофайли, метод імпульсно-кодової модуляції, частотно-маніпулюючий сигнал

Keywords: steganography, audio files, pulse code modulation method, frequency shift keying signal

Науковий керівник: доцент Г. Ахмаметьєва

АВТОМАТИЗАЦІЯ ТЕСТУВАННЯ НА ПРОНИКНЕННЯ ТА ХАРДЕНІНГУ CMS WORDPRESS

Плаксін Олександр

студент 5-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

WordPress - це веб-платформа для створення і управління веб-сайтами та блогами. Він поєднує в собі зручний інтерфейс для користувачів з потужною функціональністю, що робить його популярним серед початківців та досвідчених веб-розробників. Користувачі WordPress можуть легко створювати, редагувати та публікувати контент на своєму веб-сайті без необхідності великих знань програмування. Він надає широкий вибір тем та плагінів, що дозволяє користувачам налаштовувати свій сайт з урахуванням їхніх унікальних потреб.

Згідно інформації із ресурсу HostingAdvice[1], у світі існує понад 800.000.000 веб-сайтів, створених на WordPress. Згідно дослідженням ресурсу w3techs [2], 43% усіх існуючих веб-сайтів використовують WordPress. Згідно даним ресурсу Hostinger [3]– щодня створюється понад 500 веб-сайтів на WordPress. В пошуковій мережі IoT Shodan [4] існує понад 1.500.000 веб-сайтів на WordPress, що вільно індексуються (багато корпоративних сайтів мають захист від індексування подібними системами накшталт Shodan, тому результатів лише півтора мільони). Подібна Shodan пошукова система Censys [5] показує приблизно той самий результат. Згідно даним з офіційного веб-сайту WordPress [6], CMS має майже 60.000 безкоштовних плагінів (додадків до CMS, які покращують досвід використання).

Однією з найбільших проблем WordPress є безпека. Велика популярність платформи робить її цілью для зловмисників, які намагаються використовувати різноманітні вразливості для отримання доступу до веб-сайтів та інформації користувачів.

До найпоширеніших вразливостей WordPress можна віднести:

– **Крос-сайтовий скриптінг (XSS)**: Приклад: Вразливість у плагіні Contact Form 7, яка дозволяє злоумисникам вставляти зловісний JavaScript у поля форми контактів, що може призвести до виконання шкідливого коду на боці клієнта.

– **SQL ін’єкція**: Приклад: Вразливість у плагіні WordPress SEO by Yoast, яка дозволяє зловмисникам виконувати SQL-запити на базу даних, що може призвести до витоку конфіденційної інформації.

– **Витік інформації**: Приклад: Вразливість у плагіні GDPR Cookie Consent, яка дозволяє зловмисникам отримувати доступ до конфіденційної інформації про користувачів, такої як їх IP-адреси.

– **Вразливості в сторонніх темах та плагінах**: Приклад: Вразливість у темі Divi, що дозволяє зловмисникам отримати доступ до адміністративного інтерфейсу веб-сайту та вносити зміни в контент.

– **Витік адміністративних прав**: Приклад: Вразливість у плагіні WP Super Cache, яка дозволяє зловмисникам отримати адміністративний доступ до веб-сайту без належних дозволів.

Ці приклади демонструють, як важливо регулярно оновлювати та контролювати теми, плагіни та ядро WordPress, а також вживати заходів безпеки, щоб захистити веб-сайт від потенційних загроз.

Проста в налаштуванні CMS WordPress так само проста й в місконфігурації (недолік із класу вразливостей через людський фактор, коли адміністративний користувач банально забуває / не є компетентним, або неправильно із точки зору безпеки здійснює налаштування веб-сайту, що може привести до швидкого зламу ресурсу із подальшим витоком чутливих даних).

Найбанальніший приклад місконфігурації – це залишення файлу wp-config.php (конфігураційного файлу WordPress) із таким поширенням, через який його можна буде "прочитати" будь-якому користувачу.

Наприклад: якщо у конфігурації поширення ".php", то веб-сервер не дасть його прочитати звичайному користувачу, однак деякі "адміністратори" роблять резервні копії конфігурацій та присвоюють їм інші поширення, які із легкістю можуть бути прочитані ким завгодно (наприклад, wp-config.php.bak, wp-config.php.txt тощо)

В такій конфігурації міститься наступна інформація:

– Інформація про базу даних (де розташована, логін та пароль)

– Інформація про адміністративних користувачів (логін/пароль від поштового серверу)

– "солі" (елемент, що посилює паролі) та додадкові ключи від WordPress API

Таким чином, для мінімалізації таких ризиків, як наприклад, людський фактор, потрібно створити рішення, яке зможе в автоматичному режимі раз в деякий період часу проводити певне "сканування" веб-сайту із внутрішньої сторони (тобто із середини серверу, а не із зовні), знаходити вразливості (місконфігурації, застаріле ПЗ тощо) та сигналізувати адміністратору про це. Після

сканування, адміністратор буде проінформований про ті чи інші вразливості, що були знайдені, а далі повинен сам прийняти рішення та зробити певні кроки для їх виправлення.

Список використаних джерел:

1. How Many Websites Use WordPress? (2024 Stats) URL: https://www.hostingadvice.com/how-to/how-many-websites-use-wordpress/
2. Usage Statistics and Market Share of Content Management Systems, May 2024 URL: https://w3techs.com/technologies/overview/content_management/
3. Top 22 WordPress Statistics: Defining Trends and Insights for 2024 URL: https://www.hostinger.com/tutorials/wordpress-statistics/
4. Shodan Search Engine URL: https://shodan.io/
5. Attack Surface Management and Threat Hunting Solutions | Censys URL: https://censys.io/
6. WordPress Plugins | WordPress.org URL: https://wordpress.org/plugins/

Ключові слова: WordPress, CMS, SQL, XSS, Cybersecurity, risks, SQL injections
Keywords: WordPress, CMS, SQL, XSS, Cybersecurity, risks, SQL injections

Науковий керівник: доцент В. Бойко

ЗАБЕЗПЕЧЕННЯ ПІДТРИМКИ ІНФРАСТРУКТУРИ СИСТЕМ КІБЕРБЕЗПЕКИ ТА ЗАХИСТУ ДАНИХ В УМОВАХ ЗБОЇВ ЕНЕРГОПОСТАЧАННЯ

Пшеничний Євгеній

студент 4-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

Згідно зі статистичними дослідженнями існує тренд на збільшення щорічної частоти відключень електроенергії після 1998 року. Середня частота до 1998 р. становила 11,5 подій на рік, тоді як після 1998 р. вона становила 23,5 подій на рік. Найбільш широко обговорюваним у літературі відключенням електроенергії є блекаут, що стався в серпні 2003-го року в енергосистемах США і Канади, який торкнувся 50 мільйонів осіб і призвів до втрат у розмірі 6 мільярдів доларівх[1]. У [2] описується два послідовних великих відключення електроенергії в енергосистемі Індії 30 і 31 липня 2012 року, що сталося через неправильну роботу реле, спричинену неправильними налаштуваннями (прихованою відмовою) на тлі серйозних перебоїв у подачі енергії. Ці блекаути торкнулися 620 мільйонів осіб і

на 2022-й рік є найбільшими з коли-небудь зареєстрованих із погляду кількості постраждалих людей[3].

Відключення електроенергії створює ідеальні умови для кіберзловмисників для використання вразливостей. Це відбувається тому, що системи безпеки та інструменти моніторингу, які потребують безперервного живлення, стають тимчасово неефективними". Звернути уваги що проблему для кібербезпеки робить не відсутність електро постачання а те що системи мають послаблення тому не можуть працювати на повну потужність внаслідок чого швидкість виявлення вторгнень буде знижена."Ризик виходить за рамки безпосереднього збою. Під час відключення електроенергії затримка з виявленням вторгнень дає зловмисникам більше часу для доступу до конфіденційної інформації, встановлення шкідливого програмного забезпечення або програм-вимагачів, а також для підготовки майбутніх атак[4], [5].

Перебої в електропостачанні можуть мати значний вплив на інфраструктуру систем кібербезпеки та захисту даних: відключення електроенергії може призвести до зупинки серверів, мережевого обладнання та систем зберігання даних. Це може призвести до втрати даних, порушення роботи інформаційних систем та інших критичних послуг. Зловмисники можуть використовувати перебої з електропостачанням для здійснення кібератак. Наприклад, вони можуть розпочати DDoS-атаки або спробувати проникнути в системи коли відсутнє живлення. Відсутність електроенергії може ускладнити або зробити неможливим реагування на кібератаки. Це може призвести до того, що зловмисникам вдасться завдати більшої шкоди.

Важливо також зазначити, що кібербезпека та захист даних - це не просто технічні питання. Це також питання культури та управління ризиками. Тому для того, щоб ефективно вирішувати ці проблеми, необхідний комплексний підхід, який буде включати в себе як технічні, так і організаційні заходи.

Досвід безперебійної роботи інформаційно-комунікаційних мереж часто призводить до того, що багато різних служб - у тому числі служби, що забезпечують інфраструктуру кібербезпеки так чи інакше використовують їх, як частину своєї інфраструктури, при цьому не приділяючи належної уваги стійкості до збоїв в енергопостачанні. При цьому часто ігнорується відразу кілька факторів, що створюють ризики, а саме:

– загальна зношеність інфраструктури, що визначає наростання кількості збоїв у системах енергопостачання;

– подорожчання енергоносіїв, що опосередковано уповільнює чи відкладає модернізацію технічних систем;

– низьку практичну підготовленість технічного персоналу та невміння планувати та прогнозувати дії для екстрених ситуацій - наприклад, далеко не кожен технічний фахівець має уявлення про те, що різні моделі дизельних генераторів мають різні граничні терміни функціонування, після яких їм потрібна перерва та технічне обслуговування.

Досвід блекаутів зими 2023 року говорить про те, що системи кірбербезпеки та системи інформаційно-комунікаційних мереж загалом вимагають додаткових заходів та технічних засобів, що дозволяють мінімізувати збитки від відключень електроенергії. Практика показала, що найбільшою проблемою є невомзожність управління та контролю енергетичної складової інфраструктури дистанційно, що часто ускладнювалася запровадженням комендантської години, що додатково ізолювало технічний персонал.

Частою була ситуація, коли системному адміністратору доводилося гадати про причини збою в системі – це блекаут чи просто помилка у програмному забезпеченні? Особливо складною була ситуація у разі, якщо доводилося підтримувати парк розподілених у просторі серверів.

З урахуванням цього, складовими мінімальної системи можуть бути:

– система моніторингу стану електромережі;

– система моніторингу джерел безперебійного живлення,

– система моніторингу стану живлення серверів та робочих станцій (за даними їх сенсорних систем);

– система дистанційного увімкнення відключеного обладнання при подачі електроенергії (зокрема зручною є технологія Wake-On-LAN.

Як показує практика, такі заходи, реалізовані з урахуванням можливості доступу до самих серверних систем (у найпростішому випадку - за протоколом ssh), дозволять вчасно вжити заходів штатного реагування на відключення світла:

– контролювати стан машинного парку;

– контролювати ступінь зношеності систем резервного живлення;

– задіяти альтернативні варіанти роботи при відключенні живлення;

– штатно відключати "безнадійні" чи некритичні системи, що дозволить уникнути втрати даних;

– при відновленні живлення запускати всі відключені системи, контролюючи при цьому стан системи резервування.

Все вищеперелічене передбачає необхідність підвищеної уваги до систем управління інфраструктурою інформаційно-комунікаційних мереж, як до важливої складової кібернетичної безпеки.

Список використаних джерел:

1. Hines, P., Apt, J., & Talukdar, S. (2009). Large blackouts in North America: Historical trends and policy implications. Energy Policy, 37(12), 5249–5259. doi:10.1016/j.enpol.2009.07.049
2. Lai, L. L., Zhang, H. T., Mishra, S., Ramasubramanian, D., Lai, C. S., & Xu, F. Y. (2012, November). Lessons learned from July 2012 Indian blackout. In 9th IET International Conference on Advances in Power System Control, Operation and Management (APSCOM 2012) (pp. 1-6). IET.
3. Simonoff, J. S., Restrepo, C. E., & Zimmerman, R. (2007). Risk-management and risk-analysis-based decision tools for attacks on electric power. Risk Analysis: An International Journal, 27(3), 547-570.
4. Risks power outages pose to cyber security URL: https://www.utilitybidder.co.uk/compare-business-energy/risks-power-outages-pose-to-cyber-security/
5. Haes Alhelou, H., Hamedani-Golshan, M. E., Njenda, T. C., & Siano, P. (2019). A survey on power system blackout and cascading events: Research motivations and challenges. Energies, 12(4), 682.

Ключові слова: ризики, інфраструктура кібербезпеки, енергопостачання, критична інфраструктура, системне адміністрування, кібербезпека, блекаути

Keywords: risks, cybersecurity infrastructure, energy supply, critical infrastructure, system administration, cybersecurity, blackouts

Науковий керівник: доцент В. Бойко

ОЦІНКА БЕЗПЕКИ ДРОТОВИХ, БЕЗДРОТОВИХ ТА ХМАРНИХ КОМП'ЮТЕРНИХ МЕРЕЖ

Саблуков Костянтин

студент 2-го курсу факультету кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія»

В сучасному світі існує багато типів комп'ютерних мереж, які використовуються для обміну даними. Ця робота зосереджена на порівнянні трьох основних типів: дротових, бездротових та хмарних мереж.

Дротові мережі використовують фізичні проводи для передачі даних. Це можуть бути медні кабелі, оптоволоконні кабелі або інші типи проводів. Основний протокол, який використовується в дротових мережах, - це Ethernet. Ethernet

використовує протокол контролю доступу до носія з контролем зіткнень (CSMA/CD), який регулює, як комп'ютери передають дані по мережі. TCP/IP - це набір протоколів, який використовується для передачі даних в Інтернеті. TCP/IP включає в себе протоколи, такі як TCP, який забезпечує надійну передачу даних, і IP, який відповідає за маршрутизацію даних по мережі.

Надійність та стабільність: Дротові мережі менше вразливі до кібератак, що базуються на безпосередньому доступі до мережевого обладнання.

Контроль доступу: Фізичний доступ до мережевих кабелів краще контрольований, зменшуючи можливість несанкціонованого доступу.

Але, мобільність обмежена. Фізичне підключення через кабель обмежує мобільність користувачів та пристроїв. Потенційною точкою атаки є фізичний доступ до кабелів може бути використаний для атак або підміни обладнання.

Бездротові мережі використовують радіохвилі або інфрачервоне випромінювання для передачі даних. Wi-Fi - це найпоширеніший стандарт для бездротових мереж, який використовує протоколи IEEE 802.11. Ці протоколи визначають, як дані кодуються і передаються по радіохвилях. Для захисту даних в бездротових мережах використовуються протоколи безпеки, такі як WEP, WPA, WPA2 та WPA3. Ці протоколи шифрують дані, щоб запобігти несанкціонованому доступу [1].

Мобільність: Бездротові мережі надають значну мобільність, оскільки пристрої можуть підключатися з будь-якого місця в межах діапазону сигналу.

Легкість встановлення: Бездротові мережі зазвичай легше встановлювати та налаштовувати, оскільки вони не вимагають прокладання кабелів.

Загрози перехоплення: Бездротові мережі відкриваються на більше ризиків з точки зору безпеки через можливість перехоплення радіосигналів.

Обмежена швидкість та пропускна здатність: Швидкість передачі даних може бути нижчою, особливо при великій кількості підключених пристроїв.

Хмарні мережі використовують інтернет для зберігання та обробки даних на віддалених серверах. Це дозволяє користувачам мати доступ до своїх даних з будь-якого місця, де є доступ до Інтернету. Протоколи SSL/TLS використовуються для захисту даних під час їх передачі між клієнтом та сервером. Ці протоколи використовують криптографію для шифрування даних, що передаються по мережі, щоб запобігти перехопленню. Інші протоколи, такі як IPsec та SSH, також можуть використовуватися для захисту даних в хмарних мережах [2].

Криптографічний захист: Хмарні мережі можуть використовувати протоколи шифрування для захисту даних під час їх передачі через Інтернет.

Захищений доступ: Протоколи безпеки, такі як SSL/TLS, можуть забезпечувати захищений доступ до хмарних ресурсів.

Приватність даних: Приватність може стати питанням через те, що дані зберігаються на серверах, які контролюються постачальниками хмарних послуг.

Залежність від Інтернет-з'єднання: Доступність та продуктивність хмарних мереж можуть залежати від швидкості Інтернету та надійності постачальника хмарних послуг.

Однією з основних переваг дротових мереж є висока швидкість передачі даних. Вони також зазвичай більш надійні і стабільні, ніж бездротові мережі, оскільки вони менш схильні до перешкод. Однак головним недоліком дротових мереж є те, що вони обмежені в плані мобільності. Для підключення до дротової мережі пристрій повинен бути фізично підключений до мережі через кабель.

Бездротові мережі, з іншого боку, використовують радіохвилі для передачі даних, що дозволяє пристроям зв’язуватися без фізичних кабелів. Це надає значну мобільність, оскільки пристрої можуть підключатися до мережі з будь-якого місця в межах діапазону сигналу. Бездротові мережі також легше встановлювати та налаштовувати, оскільки вони не вимагають прокладання кабелів. Однак бездротові мережі можуть мати проблеми з безпекою, оскільки радіосигнали можуть бути перехоплені. Крім того, швидкість передачі даних в бездротових мережах може бути нижчою, ніж у дротових мережах, особливо якщо є багато пристроїв, які використовують мережу одночасно.

Нарешті, хмарні мережі використовують інтернет для зберігання та обробки даних на віддалених серверах. Це забезпечує високу масштабованість, оскільки користувачі можуть легко збільшувати або зменшувати обсяг використовуваних ресурсів в залежності від своїх потреб. Однак хмарні мережі також можуть мати питання щодо приватності та безпеки, оскільки дані зберігаються на серверах, які контролюються постачальниками хмарних послуг, а не самими користувачами. Крім того, доступність та продуктивність хмарних мереж залежать від швидкості Інтернету та надійності постачальника хмарних послуг.

Список використаних джерел:

1. Романов, Д. В. "найбільш популярні вразливості мережевих протоколів у корпоративній мережі." Матеріали XV-ої Міжнародної науково-практичної конференції «Free and Open Source Software», Харків, 13-14 лютого 2024 р.–Харків: Харківський національний економічний університет імені Семена Кузнеця, 2024.–148 с.: 139.
2. Павленко, Є. Р., and Дмитро Володимирович Великодний. Мережева безпека на основі протоколів IPsec і SSL VPN. Diss. Сумський державний університет, 2017.

Ключові слова: інтернет-мережа, протоколи, бездротові мережі, дротові мережі, хмарні мережі
Keywords: internet network, protocols, wireless networks, wired networks, cloud networks.

Науковий керівник: доцент О. Задерейко

УГОДА МІЖ США ТА СПОЛУЧЕНИМ КОРОЛІВСТВОМ ПРО ЗАБЕЗПЕЧЕННЯ БЕЗПЕКИ СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ

Мазур Дмитро

студент 1 курсу факультету адвокатури та антикорупційної діяльності Національного університету «Одеська юридична академія»

У квітні 2024 року США і Велика Британія підписали угоду про перевірку безпеки великих мовних моделей (LLM), що лежать в основі систем штучного інтелекту. Ця угода дасть змогу обом країнам працювати над узгодженням своїх наукових підходів і тісно співпрацювати в розробці наборів оцінок для моделей, систем і агентів штучного інтелекту.

Відповідні розробки будуть виконані нещодавно створеним британським Інститутом безпеки штучного інтелекту (AISI) та їхніми американськими колегами [1].

Тема розвитку штучного інтелекту (далі - ШІ) є актуальною сьогодні з кількох причин: ШІ має потенціал революціонізувати багато аспектів нашого життя. ШІ може автоматизувати багато завдань, які в даний час виконуються людьми, звільняючи наш час для більш творчої та продуктивної роботи. ШІ також може допомогти нам вирішити деякі з найнагальніших проблем світу, таких як зміна клімату, бідність і хвороби.

У останні роки технології ШІ розвивалися дуже швидкими темпами. Це пов`язано з низкою факторів, включаючи збільшення доступності даних, обчислювальних потужностей і талантів.

Науковці активно досліджують методи забезпечення безпеки ШІ. Це включає розробку методів виявлення та пом'якшення ризиків, пов'язаних з ШІ, а також розробку більш надійних та прозорих систем ШІ.

Однак, при цьому існує багато потенційних ризиків, пов'язаних з ШІ. Незважаючи на потенційні переваги ШІ, існує також багато потенційних ризиків, пов'язаних з його розвитком.

У травні 2023 року сотні лідерів технологічної галузі, вчених та інших громадських діячів підписали відкритий лист із попередженням про те, що еволюція ШІ може призвести до вимирання людства [2].

Наприклад, ШІ може бути використаний для створення автономної зброї, яка може становити загрозу для людства. ШІ також може призвести до масового безробіття, оскільки машини стають здатними виконувати багато завдань, які в даний час виконуються людьми.

Тому важливо обговорити етичні наслідки ШІ, щоб максимізувати переваги ШІ та мінімізувати ризики, важливо обговорити етичні наслідки його розвитку. Ця дискусія повинна включати експертів з різних галузей, включаючи етику, філософію, право та технології. Потрібно розробити політику для регулювання ШІ. У міру того, як ШІ продовжує швидко розвиватися, потрібно розробити політику для його регулювання. Ця політика повинна бути спрямована на те, щоб гарантувати, що ШІ використовується на благо людства, а не завдає шкоди. [3]

Потенційні ризики, пов`язані з використанням штучного інтелекту (ШІ): системи ШІ можуть успадковувати та посилювати упередження, що існують у даних, на яких вони навчаються. Це може призвести до дискримінації певних груп людей.

Також ШІ може використовуватися для створення автономної зброї, яка може становити загрозу для людства. ШІ також може використовуватися для створення систем спостереження, які можуть використовуватися для відстеження та контролю людей без їхньої згоди.

Крім того, ШІ може призвести до масового безробіття, оскільки машини стають здатними виконувати багато завдань, які в даний час виконуються людьми.

Якщо системи ШІ стануть занадто складними, ми можемо втратити контроль над ними. Це може призвести до того, що системи ШІ створять шкоди людям або навіть захоплять світ (що поки що є улюбленою темою у фантастів).

ШІ може використовуватися зловмисними цілями, такими як кібератаки, дезінформація та маніпулювання людьми. Використання ШІ може призвести до низки етичних проблем, таких як питання про те, чи мають машини права, і як ми повинні ставитися до штучного інтелекту.

Важливо зазначити, що це лише деякі з потенційних ризиків, пов'язаних з використанням ШІ. Важливо продовжувати досліджувати та розробляти ШІ відповідальним чином, щоб мінімізувати ці ризики та максимізувати переваги ШІ [4].

І саме через ці причини та ризики деякі країни вирішили підписати угоду про те, щоб зробити AI захищеним проектом. На даний момент були зроблені такі кроки для того, щоб зробити AI більш безпечним: У 2019 році 26 країн, включаючи Україну, підписали Декларацію про відповідальне розроблення штучного інтелекту, заснованого на принципах етики та науковості. Ця декларація описує ключові принципи, які повинні керувати розробкою та використанням ШІ, такі як повага до прав людини, безпека, прозорість та підзвітність.

У 2020 році Європейська Комісія опублікувала «Етичні принципи для штучного інтелекту», які описують рекомендації щодо розробки та використання ШІ відповідно до європейських цінностей.

У вересні 2023 року у Великій Британії відбувся глобальний саміт із безпеки ШІ, на якому були присутні представники ЄС, США, Китаю, Індії, Німеччини, Франції та інших країн, зокрема й України. За підсумками саміту було підписано декларацію, що отримала назву «Декларація Блетчлі» [5].

Тим самим угода між США та Сполученим Королівством стала ще одним (і, ймовірно) далеко не останнім етапом на шляху забезпечення безпеки систем штучного інтелекту. Розробка рівнів захисту ШІ дозволить забезпечити безпеку, етичність та стійкий розвиток цієї технології. Такі угоди про ШІ є важливим інструментом для забезпечення безпечного, етичного та стійкого розвитку цієї технології і в Україні і світі також.

Список використаних джерел:

1. Ghoshal Anirban US and UK sign agreement to test the safety of AI models URL: https://www.cio.com/article/2076986/us-and-uk-sign-agreement-to-test-the-safety-of-ai-models.html.
2. Pause Giant AI Experiments: An Open Letter URL: https://futureoflife.org/open-letter/pause-giant-ai-experiments.
3. Vallance Chris Artificial intelligence could lead to extinction, experts warn. URL: https://www.bbc.com/news/uk-65746524
4. Artificial Intelligence and Emerging Technology Initiative URL: https://www.brookings.edu/projects/artificial-intelligence-and-emerging-technology-initiative/
5. МакФарланд Алекс Що таке Декларація Блетчлі, підписана 28 країнами? URL: http://surl.li/trsrs/.

Ключові слова: штучний інтелект, визначення ключових загроз, стандарти та регулювання в галузі штучного інтелекту, співпраця в області досліджень та розвитку нових технологій
Keywords: artificial intelligence, identification of key threats, standards and regulation in the field of artificial intelligence, cooperation in research and development of new technologies.

Науковий керівник: доцент Р. Чанишев

ЗАСТОСУВАННЯ НЕЙРОННИХ МЕРЕЖ У ЗАДАЧАХ ЗАХИСТУ ІНФОРМАЦІЇ

Заставнюк Мирослав

студент 4 курсу факультету кібербезпеки та інформаційних технологій Національного університету "Одеська юридична академія"

Черевко Євген

кандидат фізико-математичних наук, доцент, доцент кафедри кібербезпеки Національного університету "Одеська юридична академія"

Нейронні мережі – це набір алгоритмів, який намагається розпізнати фундаментальні взаємозв'язки у наборі даних за допомогою процесів, які імітують роботу людського мозку. У цьому сенсі нейронні мережі - це система нейронів органічного або штучного походження.

З розвитком сучасних технологій і збільшенням обсягу цифрових даних питання безпеки та захисту даних стають все більш важливими. Нейронні мережі досягли величезного прогресу в широкому спектрі завдань, таких як порівняння та розпізнавання ідентичних даних, що використовуються в системах безпеки аеропортів. Це робиться шляхом захоплення облич людей і порівнянням їх із базою даних злочинців. Також прикладом є функція GoogleLens яка шукає подібні фото. Достатньо завантажити фото і система знайде усі зображення подібні до завантаженої картинки.[1] Ще одним прикладом є reCapcha яка навчається шляхом того що користувач обирає унікальні або задані картинки серед дев'яти зображень.

Штучний інтелект – є надбанням технологічної цивілізації, складною системою зв'язків та основою для створення об'єктів інформаційних систем. У найближчому майбутньому активне застосування технологій штучного інтелекту зможе дозволити переробити людську поведінку, змінити суспільні відносини та вплинути на існуючі особисті характеристики.

Штучний інтелект – це, насамперед, технології. Саме технологіям, які застосовуються для розробки систем та продуктів прогнозують велике майбутнє. До таких технологій відноситься:

- машинне навчання;
- комп'ютерний зір;
- когнітивістика;
- Natural Language Processing, NLP (Обробка природної мови);
- глибоке навчання та інші.

Сучасні технології штучного інтелекту реалізуються такими напрямами:

- розпізнавання та синтез мови;
- інтелектуальні системи підтримки прийняття рішень;
- перспективні методи штучного інтелекту [2].

Використання нейронних мереж для виявлення загроз та реагування на них також може автоматизувати процес виявлення та реагування на потенційні атаки. Моделі глибокого навчання можуть аналізувати дані в мережі, виявляти аномалії та вживати заходи безпеки для захисту критично важливих даних. Наприклад, існують глобальні компанії, які аналізують в мережі великий об’єм інформації, що може вказувати на нові загрози або, наприклад, передбачити атаки нульового дня. Ці компанії мають в своїх активах системи, які збирають масиви даних, аналізують їх за допомогою технологій штучного інтелекту, виявляють закономірності, проводять кластеризацію даних та прогнозують загрози. Без таких технологій обробити такий обсяг інформації було б практично неможливо. Тут широко використовуються нейронні мережі та кластеризація. Штучний інтелект також активно використовується для відстеження і прогнозування загроз інформаційній безпеці на основі отриманих з відкритих і закритих джерел. Таким чином сфера застосування та масштаб завдань штучного інтелекту в сфері інформаційної безпеки значно зросли за останні десятиліття.

Криптографія є ключовим елементом захисту інформації в цифровому світі. Вона використовується для шифрування даних з метою збереження їх конфіденційності та забезпечення їхньої цілісності під час передачі через ненадійні канали зв'язку.

Застосування нейронних мереж у криптографії може включати такі аспекти:

1. Розшифрування шифрів

Нейронні мережі можуть бути використані для розшифрування шифрів, що допомагає зламувачам отримати доступ до зашифрованих даних без ключа. Зокрема, вони можуть бути застосовані для аналізу шифрованого тексту та виявлення патернів, які допомагають відновити оригінальний текст.

2. Криптографічний аналіз

Нейронні мережі можуть допомагати у вирішенні складних криптографічних проблем, таких як аналіз відкритого тексту, шифрів та протоколів. Вони можуть використовуватися для знаходження слабких місць у криптографічних системах та розробки більш надійних алгоритмів шифрування.

3. Створення нових криптографічних методів

Нейронні мережі можуть бути використані для розробки нових методів шифрування та аутентифікації, які базуються на навчанні з використанням великих обсягів даних. Це може включати розвиток алгоритмів шифрування, які стійкі до атак з використанням машинного навчання.

Варто також відзначити розвиток теорії нейронних мереж. Зараз це зокрема втілюється у так званих графових нейронних мережах – GNN (Graph Neural Network). Графова нейронна мережа характерна тим, що вона описує об'єки і взаємозв'зки між ними за допомогою графів – точок, що є з'єднаними лініями [3]. Графові нейронні мережи також є перспективними для застосування їх у задачах захисту інформації.

Список використаних джерел:

1. М. Підгородецький, М. Петрик, докт. фіз.-мат. наук; проф.(Тернопільський національний технічний університет імені івана Пулюя, Україна) «Застосування нейронних мереж для вирішення задач класифікації об'єктів на зображеннях»
2. URL:https://elartu.tntu.edu.ua/bitstream/lib/37564/2/iMST_2021_Pidhorodetskyi_M-application_of_neural_175.pdf
3. З.В. Гбур, д. держ. упр., професор.професор кафедри управління охороною здоров'я та публічного адміністрування, Національний університет охорони здоров'я України імені П.Л. Шупика «Використання штучного інтелекту в інформаційній безпеці України»
4. URL: http://www.dy.nayka.com.ua/pdf/1_2022/4.pdf
5. 3. Wu, Lingfei; Cui, Peng; Pei, Jian; Zhao, Liang (2022). "Graph Neural Networks: Foundations, Frontiers, and Applications". Springer Singapore: 725.

Ключові слова: нейронні мережі, безпека інформації, штучний інтелект, криптографія, машинне навчання

Keywords: neural networks, information security, artificial intelligence, cryptography, machine learning

АНАЛІЗ ЗАГРОЗ ТА РИЗИКІВ ДЛЯ ВЕЛИКИХ БАЗ ДАНИХ: ВІД СТОРОННІХ АТАК ДО ВНУТРІШНІХ ЗАГРОЗ

Ханов Богдан

студент 3 курсу факультету кібербезпеки та інформаційних технологій Національного Університету «Одеська юридична академія»

В сучасному цифровому світі, де дані є одним з найцінніших ресурсів, великі бази даних відіграють критичну роль у функціонуванні різноманітних організацій. Зберігаючи великі обсяги інформації про клієнтів, продукти, фінансові

транзакції та інші аспекти бізнесу, ці бази даних стають об'єктом підвищеної уваги для атак та загроз зовнішнього та внутрішнього походження.

Аналіз загроз та ризиків для великих баз даних стає важливим завданням для забезпечення безпеки, цілісності та конфіденційності інформації, яка вони містять. Від сторонніх кіберзлочинців, що шукають слабкі місця у системі, до внутрішніх загроз, що можуть виникнути в результаті недбалого ставлення до безпеки або зловживання довіри, великі бази даних потенційно вразливі до широкого спектру небезпек.

У цій роботі ми детально розглянемо різноманітні аспекти безпеки великих баз даних, включаючи загрози зовнішнього та внутрішнього походження, технічні та організаційні заходи безпеки, управління ризиками та стратегії відновлення після інциденту. Ця робота пропонує глибокий аналіз та розуміння складних взаємозв’язків між безпекою даних та ефективним функціонуванням великих баз даних у сучасному цифровому середовищі.

Загрози зовнішнього походження

1. Хакерські атаки: є однією з найбільш серйозних загроз для великих баз даних. Ці атаки можуть бути виконані з метою несанкціонованого доступу до даних, їх викрадення, модифікації або навіть видалення. Хакери можуть експлуатувати вразливості в програмному забезпеченні або мережевих протоколах, а також використовувати соціальну інженерію, щоб отримати доступ до облікових записів або інших конфіденційних даних.

Існує кілька типів хакерських атак, включаючи SQL ін’єкції, кросс-сайт скриптінг, атаки переповнення буфера, а також атаки на переповнення стеку. Кожен з цих видів атак може використовувати вразливості у програмному забезпеченні для впровадження шкідливого коду або отримання доступу до системи.

Для захисту від хакерських атак важливо вжити ряд заходів безпеки, таких як регулярне оновлення програмного забезпечення, встановлення вогнепровідних систем, впровадження механізмів аутентифікації та авторизації, а також моніторинг активності користувачів для виявлення незвичайних чи підозрілих дій.

Невідповідність цим заходам може призвести до серйозних наслідків, включаючи втрату даних, порушення конфіденційності інформації, а також втрату довіри користувачів і клієнтів. Тому важливо вжити всі можливі заходи для захисту баз даних від хакерських атак [1].

2. DDoS атаки: (розподілені атаки на відмову в обслуговуванні) є серйозною загрозою для великих баз даних. Ці атаки спрямовані на перевантаження

серверів, мереж та інфраструктури, що призводить до припинення нормального функціонування ресурсів та послуг.

У DDoS атак велика кількість запитів надходить до сервера з однієї або кількох джерел одночасно. Це може бути досягнуто за допомогою ботнетів (мережі комп'ютерів, які були заражені шкідливим програмним забезпеченням і підконтрольні зловмисникам), а також за допомогою зомбі-комп'ютерів (комп'ютерів, які були скомпрометовані і використовуються без дозволу власників) [2].

Наслідком DDoS атак може бути відмова в обслуговуванні для легітимних користувачів, тобто сервери перестають відповідати на запити або працюють з великими затримками, що призводить до недоступності сервісу. Це може мати серйозні наслідки для бізнесу, зокрема, втрату прибутку, порушення репутації та втрату довіри користувачів.

Для захисту від DDoS атак важливо вжити ряд технічних та організаційних заходів. До них можуть відноситися використання спеціалізованих пристроїв для фільтрації трафіку, налаштування вогнепровідних систем, використання CDN (мережі доставки контенту) для розподілення трафіку, а також планування реагування на інциденти та резервне копіювання даних.

Недостатній захист від DDoS атак може призвести до серйозних втрат для бізнесу та порушення нормального функціонування послуг та систем. Тому важливо вжити всі можливі заходи для захисту великих баз даних від цієї форми атаки [**Ошибка! Источник ссылки не найден.**].

3. Фішинг та соціальний інженерінг: Фішинг та соціальний інженерінг - це загрози, що використовуються зловмисниками для отримання конфіденційної інформації шляхом маніпулювання користувачами.

Фішинг - це метод атаки, коли зловмисники відправляють шахрайські повідомлення, які намагаються переконати людей надати конфіденційні дані, такі як ім'я користувача, пароль, номер кредитної картки тощо. Ці повідомлення можуть виглядати як офіційні листи від відомих компаній чи організацій. Фішингові атаки можуть бути спрямовані і на працівників, які мають доступ до баз даних, використовуючи приховані схеми, щоб надати зловмисникам доступ до системи [4].

Соціальний інженерінг - це метод атаки, коли зловмисники використовують маніпуляцію людьми для отримання конфіденційної інформації або отримання доступу до системи. Наприклад, вони можуть намагатися отримати доступ до бази даних, вимовляючися як інші працівники чи користувачі, або

використовуючи соціальні інженерні методи, такі як приховані камери або фальшиві ідентифікаційні документи.

Для захисту від фішингу та соціального інженерінгу важливо проводити навчання персоналу з безпеки інформації, встановлювати строгі політики щодо обміну конфіденційною інформацією, а також використовувати механізми перевірки ідентифікації і автентифікації, щоб ускладнити доступ для зловмисників. Також важливо підтримувати свідомість персоналу щодо потенційних загроз та небезпек, пов'язаних з фішингом та соціальним інженерінгом [**Ошибка! Источник ссылки не найден.**].

Внутрішні загрози

Внутрішні загрози для великих баз даних можуть виявитися такими, що походять від самого персоналу, який має доступ до системи і даних. Це може включати в себе недбале ставлення до безпеки, спроби використати доступ для незаконних дій, недоліки в знаннях про безпеку та навіть навмисні атаки.

Співробітники можуть стати причиною витоку конфіденційної інформації, будь то намірено чи несвідомо. Це може відбутися через недбале ставлення до захисту конфіденційності даних або через зловживання довіри, що мають співробітники.

Внутрішні помилки також можуть призвести до порушень безпеки. Наприклад, неправильне налаштування прав доступу або випадкове видалення даних може мати серйозні наслідки для безпеки інформації [**Ошибка! Источник с сылки не найден.**].

Крім того, співробітники, які мають доступ до системи, можуть намагатися використовувати цей доступ для несанкціонованих дій, таких як видалення або зміна даних, або навіть зловживання привілеями.

Для захисту від внутрішніх загроз важливо вжити ряд заходів, таких як обмеження доступу до конфіденційних даних, моніторинг активності персоналу, навчання персоналу з питань безпеки, а також впровадження строгих політик безпеки та контрольних механізмів. Тільки комплексний підхід до безпеки може надати відповідний захист від внутрішніх загроз для великих баз даних [6].

1. Службові порушення: Зловживання привілеями, недбале ставлення до безпеки та інші службові порушення можуть стати серйозною загрозою для великих баз даних. Ці види внутрішніх загроз можуть бути спричинені співробітниками, які мають доступ до системи та даних.

Перш за все, недбале ставлення до безпеки може призвести до витоку конфіденційної інформації або порушення цілісності даних. Наприклад,

використання слабких паролів або відкрите зберігання конфіденційних даних може відкрити систему для атак ззовні або зловмисних внутрішніх дій [6].

Крім того, зловживання привілеями є ще однією серйозною загрозою. Співробітники з розширеними привілеями можуть намагатися використовувати свої права для несанкціонованого доступу до даних або зміни конфіденційної інформації.

Недостатня підготовка з питань безпеки також може призвести до ненавмисних порушень безпеки. Співробітники, які не розуміють правил безпеки або не свідомі потенційних загроз, можуть ненавмисно створити вразливості в системі або виконати дії, які призведуть до порушення безпеки даних.

Нарешті, внутрішні злочини також можуть бути причиною службових порушень. Деякі співробітники можуть намагатися скомпрометувати систему або дані для власної користі або для завдання шкоди компанії.

Щоб запобігти цим внутрішнім загрозам, необхідно вжити ряд заходів, включаючи навчання персоналу з питань безпеки, встановлення строгих політик безпеки та контрольних механізмів доступу, а також постійний моніторинг активності співробітників. Тільки шляхом вжиття цих заходів можна забезпечити ефективний захист від службових порушень та знизити ризик внутрішніх загроз для великих баз даних [6].

2. Витік конфіденційної інформації. Внутрішній витік конфіденційної інформації може мати декілька причин:

Намірне розголошення: Співробітники можуть намагатися незаконно використовувати чутливі дані для особистої користі або для завдання шкоди компанії. Це може включати продаж інформації конкурентам або використання інформації для шантажу.

Недбале ставлення до безпеки: Співробітники можуть випадково розголошувати конфіденційну інформацію через необережне поводження з даними або використання необхідних заходів безпеки.

Крадіжка облікових даних: Співробітники можуть намагатися красти облікові дані користувачів або іншу конфіденційну інформацію з метою використання її для незаконних цілей.

Неавторизований доступ: Співробітники можуть отримати несанкціонований доступ до системи або даних і, в результаті, отримати можливість розголошення конфіденційної інформації.

Витік конфіденційної інформації може мати серйозні наслідки для організації, включаючи втрату довіри клієнтів, порушення законодавства про

конфіденційність даних, втрату конкурентної переваги і фінансові втрати. Тому для запобігання витокам конфіденційної інформації важливо вжити ряд заходів безпеки, таких як обмеження доступу до конфіденційних даних, моніторинг активності користувачів і забезпечення відповідності внутрішніх політик безпеки [7].

Аналіз цих загроз допомагає виявити слабкі місця в безпеці баз даних і розробити стратегії для їх запобігання та управління.

Список використаних джерел:

1. Пашорін В. І. Безпека інформаційних систем : навч. посіб. / В. І. Пашорін, Ю. В. Костюк. – Київ : Держ. торг.-екон. ун-т, 2023. – 376 с. ISBN 978-966-918-101-5
2. Доценко С. І. Організація та системи керування базами даних: навч. посібник. – Харків: УкрДУЗТ, 2023. – 117 с.
3. Інформаційна безпека та захист даних в комп'ютерних технологіях і мережах: навч. посіб. / В.П. Полторак. – Київ : КПІ ім. Ігоря Сікорського, 2020. – 78 с.
4. Павленко О.А. Методи та засоби захисту інформації в інформаційних системах: Навч. посібник. – К.: Вид-во НУБіП України, 2019. – 288 с.
5. Інформаційна безпека. Підручник / В. В. Остроухов, М. М. Присяжнюк, О. І. Фармагей, М. М. Чеховська та ін.; під ред. В. В. Остроухова – К.: Видавництво Ліра-К, 2021. – 412 с.
6. Жилін А. В. Технології захисту інформації в інформаційно-телекомунікаційних системах : навч. посіб. / А. В. Жилін, О. М. Шаповал, О. А. Успенський ; ІСЗЗІ КПІ ім. Ігоря Сікорського. – Київ : КПІ ім. Ігоря Сікорського, Вид-во «Політехніка», 2021. – 213 с.
7. Мікула М. П., Коцюк Ю. А., Мікула О. М. Організація баз даних та знань: навч. посіб. – Острог: Видавництво Національного університету «Острозька академія», 2021. –194 с.

Ключові слова: безпека даних, методи безпеки, бази даних, загрози, ризики
Keywords: data security, security methods, databases, threats, risks

Науковий керівник: доцент С. Кухаренко

СУЧАСНІ МЕТОДИ ЗАБЕЗПЕЧЕННЯ КІБЕРБЕЗПЕКИ НА ДЕРЖАВНОМУ РІВНІ ТА МІЖНАРОДНЕ СПІВРОБІТНИЦТВО

Чепурна Олена

кандидат фізико-математичних наук, доцент, доцент кафедри кібербезпеки Національного університету «Одеська юридична академія»

Кошева Катерина

здобувачка 1-го курсу магістратури Національного університету «Одеська юридична академія»

Сучасність характеризується неабиякими викликами у сфері кібербезпеки. Частота та різноманітність кібератак на бізнес, ланцюги постачання та державні установи неперервно зростали. Згідно з оцінками IBM[1], кожен витік даних призводить до збитків в розмірі понад 4,3 мільйонів доларів і вимагає щонайменше 200 днів для виявлення та усунення наслідків. Як можна забезпечити безпеку в непередбачуваному цифровому світі?

Стабільне та ефективне функціонування сучасного суспільства неможливе без вдосконалення методів та засобів захисту від кібератак.

Кібербезпека на державному рівні вимагає комплексного та цілеспрямованого підходу для захисту інформаційних ресурсів, інфраструктури та критичних систем від кіберзагроз. Держави знаходяться під постійним тиском через зростаючі загрози кібернападів, які можуть впливати на економічну стабільність, національну безпеку та довіру громадян до державних установ. Саме тому на сьогоднішній день акцент у багатьох країнах зроблено на розвиток та впровадження ряду ключових напрямків забезпечення кібербезпеки на державному рівні.

Розробка стратегій і політик у сфері кібербезпеки є першочерговим завданням. Вони визначають пріоритети, цілі та стратегії держави у сфері кібербезпеки. Так, Указом Президента № 447/2021 від 26 серпня 2021 року затверджено рішення Ради національної безпеки і оборони України від 14 травня 2021 року «Про Стратегію кібербезпеки України».

Стратегія кібербезпеки України визначає пріоритети національних інтересів у сфері кібербезпеки, наявні та потенційно можливі кіберзагрози, цілі та завдання забезпечення кібербезпеки України з метою створення умов для безпечного функціонування кіберпростору, його використання в інтересах особи, суспільства і держави [11].

Основні виклики для України у сфері кібербезпеки:

- активне використання кіберзасобів у міжнародній конкуренції;

– змагальний характер розвитку засобів кібербезпеки в умовах швидких прогресуючих змін інформаційно-комунікаційних технологій, зокрема хмарних та квантових обчислень, 5G-мереж, великих даних, Інтернету речей, штучного інтелекту тощо;

– мілітаризація кіберпростору та розвиток кіберзброї, що дає можливість приховано проводити кібератаки для підтримки бойових дій і розвідувально-підривної діяльності у кіберпросторі;

– вплив пандемії COVID-19 на економічну діяльність та соціальну поведінку, що спричинив стрімку трансформацію і організацію значного сегмента суспільних відносин у дистанційному режимі з широким використанням електронних сервісів та інформаційно-комунікаційних систем;

– упровадження нових технологій, цифрових послуг та механізмів електронної взаємодії громадян з державою, що здійснюється безсистемно в частині заходів з кібербезпеки та без належної оцінки ризиків [12].

Особлива увага приділяється захисту об`єктів критичних інфраструктур, таких як енергетика, транспорт, фінанси, охорона здоров'я та інші. Впровадження заходів з кіберзахисту в цих галузях важливе для економічної та національної безпеки.

Одним з ключових напрямів можна виділити необхідність створення та підтримку національних центрів кібербезпеки. Ці центри відіграють ключову роль у координації дій та реагуванні на кіберзагрози, а також у виявленні та аналізі інцидентів. Також у зв'язку зі зростанням загроз кібератак на виборчі системи, особлива увага приділяється заходам забезпечення безпеки виборчих процесів. Важливою складовою кібербезпеки є наявність кваліфікованих кадрів. Розвиток програм навчання та підвищення кваліфікації у сфері кібербезпеки допомагає забезпечити національну безпеку.

Законодавчий аспект грає важливу роль у забезпеченні кібербезпеки. Розробка та впровадження відповідного законодавства, включаючи закони про кіберзлочинність та захист персональних даних, є необхідною складовою. Таким прикладом є Закон України “Про основні засади забезпечення кібербезпеки України” [13].

У зв'язку з тим, що кіберзагрози перетинають межі, співробітництво на міжнародному рівні стає надзвичайно важливим. Обмін інформацією та спільна боротьба з кіберзагрозами стають ключовими елементами стратегії забезпечення кібербезпеки.

У своїй суті, кіберзлочини "не мають кордонів", тому міжнародні організації закликають держави до співпраці з іншими зацікавленими сторонами у розробці ефективних механізмів адміністративно-правового регулювання у сфері кібербезпеки. Це означає не лише розробку та ухвалення необхідного законодавства, але й проведення спільних розслідувань таких дій застосовуючи існуюче міжнародне право [4].

Серед органів, які здійснюють регулювання кіберпростору, варто зазначити такі: у Європейському Союзі функціонує Агентство з мережевої та інформаційної безпеки (European Network and Information Security Agency, ENISA), у Сполучених Штатах Америки кібербезпекою займається Агентство Національної Безпеки, у НАТО створений Комітет з кібернетичної оборони (The Cyber Defence Committee), а також Спільний центр з кібернетичної оборони (Cooperative Cyber Defence Centreof Excellence), Спеціалізований центр з оборони в сфері кібербезпеки НАТО (CCDCOE), Міжнародний альянс із забезпечення кібербезпеки (ICSPA), Інтерпол (INTERPOL), Міжнародне багатостороннє товариство проти кіберзагроз (IMPACT) та ін. [14].

Вагому роль у виробленні єдиних підходів щодо забезпечення кібербезпеки, як складової національної безпеки країн, відіграє Організація Північноатлантичного договору (НАТО), партнерство з якою для України є пріоритетним у сфері зовнішньополітичної діяльності.

Сама якісна система кіберзахисту критично важливої інфраструктури успішно діє в Сполучених Штатах. Там розроблені стандарти безпеки, такі як Фреймворк кібербезпеки NIST, які активно використовуються багатьма приватними організаціями по всьому світу. Ці стандарти дозволяють виявляти, реагувати та навіть уникати кіберінцидентів. Для України важливо вивчити досвід США у сфері кібербезпеки, зокрема, щодо створення відповідної нормативно-правової бази.

Міжнародне співробітництво України та здобуття досвіду в адміністративно-правовому забезпеченні кібербезпеки є надзвичайно важливими для удосконалення законодавства у цій області та підвищення ефективності національної системи кібербезпеки України.

Список використаних джерел:

1. 6 трендів кібербезпеки[Електронний ресурс]. – Режим доступу: https://wezom.com.ua/ua/blog/6-trendiv-kiberbezpeki-v-2024-rotsi
2. Boardroom Cybersecurity Report 2023. https://cybersecurityventures.com/cybersecurity-boardroom-report-2023/.
3. United States - The Economist Intelligence Unit. https://country.eiu.com/united-states.

4. Cyber Ventures Boardroom Cybersecurity Report 2023. https://www.secureworks.com/resources/rp-boardroom-cybersecurity-report-2023.
5. The 2023 Global Cybercrime Report: A look at the key takeaways. https://www.tripwire.com/state-of-security/2023-global-cybercrime-report-look-key-takeaways.
6. eSentire | 2023 Official Cybercrime Report. https://www.esentire.com/resources/library/2023-official-cybercrime-report.
7. Cybersecurity Statistics in 2024 - USA TODAY. https://www.usatoday.com/money/blueprint/business/vpn/cybersecurity-statistics/.
8. Economic Forecast for the US Economy - The Conference Board. https://www.conference-board.org/research/us-forecast.
9. (8) An Update to the Economic Outlook: 2023 to 2025. https://www.cbo.gov/publication/59431.
10. US GDP Set to Rise 5.6% Annually to 2025 from 2020 Pandemic-Induced .... https://www.prnewswire.com/news-releases/us-gdp-set-to-rise-5-6-annually-to-2025-from-2020-pandemic-induced-recession-301504970.html.
11. Стратегія кібербезпеки України: цілі та пріоритети [Електронний ресурс]. – Режим доступу: https://armyinform.com.ua/2021/08/27/ strategiya-kiberbezpeky-ukrayiny-czili-ta-priorytety/
12. Про Стратегію кібербезпеки України: Указ Президента № 447/2021 від 26 серпня 2021 р.
13. Про основні засади забезпечення кібербезпеки України: Закон України від 5.10.2017 р. № 2163-VIII. ВВР України

Ключові слова: Кібербезпека; кіберзахист; стратегія захисту; державне регулювання;міжнародне співробітництво

Keywords: Cyber security; cyber protection; defense strategy; state regulation; international cooperation

Секція 3. УПРАВЛІНСЬКІ ТА ЕКОНОМІЧНІ АСПЕКТИ РОЗВИТКУ СУЧАСНОЇ ІТ ІНДУСТРІЇ

ВИКОРИСТАННЯ СТАТИСТИЧНИХ МЕТОДІВ В ПРОЦЕСІ ОЦІНКИ ЕФЕКТИВНОСТІ МАРКЕТИНГОВОЇ ДІЯЛЬНОСТІ

Горбаченко Станіслав

доктор економічних наук, професор, завідувач кафедри кібербезпеки Національного університету «Одеська юридична академія»

Разінкін Нікіта

асистент кафедри кібербезпеки Національного університету «Одеська юридична академія»

Сучасні ринки товарів та послуг характеризуються здебільшого високим рівнем конкуренції, отже суб'єктам підприємництва необхідно оцінювати та, за необхідністю, корегувати власні маркетингові заходи. При цьому, з високою ефективністю можуть застосовуватися статистичні методи, які надають кількісні дані та необхідне підґрунтя для аналітичних висновків.

Чи не найвідомішим статистичним методом для оцінки маркетингової діяльності є регресійний аналіз (рис. 1). Він дозволяє виявити залежність між маркетинговими змінними, такими як витрати на рекламу, ціни на продукти, обсяги продажів та інші фактори. Завдяки регресійному аналізу можна визначити, які фактори найбільше впливають на обсяги продажів та яку частку змін у продажах можна пояснити змінами у витратах на маркетинг. Це дозволяє прогнозувати майбутні продажі на основі аналізу минулих даних, що допомагає компаніям ефективніше планувати свої маркетингові бюджети [1].

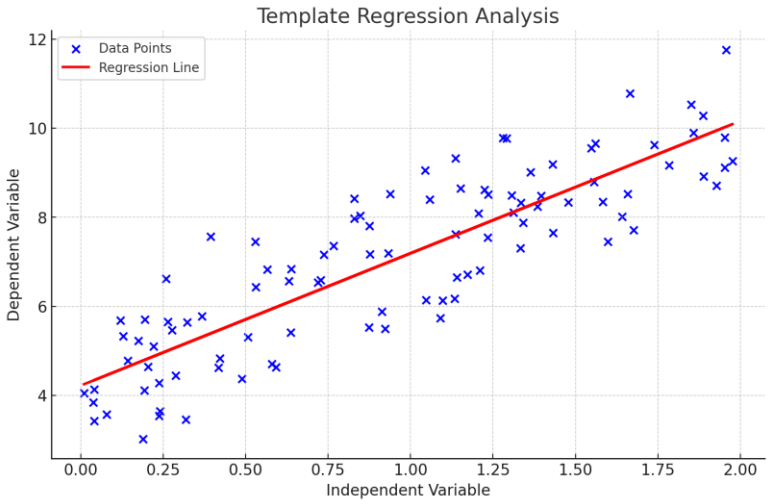


Рисунок 1 – Приклад регресійного аналізу

Іншим важливим методом є кореляційний аналіз, який дозволяє визначити ступінь взаємозв'язку між різними маркетинговими змінними. Наприклад, він може довести існування зв'язку між збільшенням витрат на рекламу та зростанням продажів. Високий коефіцієнт кореляції свідчить про сильний зв'язок між змінними, що дозволяє суб'єктам підприємництва виявляти найефективніші маркетингові канали та концентрувати свої зусилля на них.

Сегментація ринку є одним із важливих елементів маркетингової діяльності. В цьому сенсі використання кластерного аналізу дозволяє групувати споживачів за різними ознаками, такими як вік, дохід, поведінкові характеристики тощо. Це дозволяє суб'єктам підприємництва точніше налаштовувати маркетингові інструменти, з орієнтуванням чи таргетуванням їх на окремі сегменти споживачів. Наприклад, можна розробити різні рекламні повідомлення для різних сегментів споживачів і потім порівняти конверсію.

Дисперсійний аналіз також відіграє значну роль у маркетинговому аналізі, допомагаючи визначити, чи існують статистично значущі відмінності між різними групами споживачів або маркетинговими заходами. Завдяки цьому суб'єкти підприємництва можуть оцінювати ефективність різних маркетингових стратегій та коригувати їх відповідно до отриманих результатів. Дисперсійний аналіз може показати, яка рекламна кампанія є найбільш ефективною для певної групи споживачів, що дозволяє в майбутньому оптимізувати маркетингові витрати.

Застосування часових рядів дозволяє аналізувати зміни у маркетингових показниках протягом часу. Це корисно для виявлення трендів та сезонних коливань, що дозволяє суб'єктам підприємництва планувати свої маркетингові заходи та ресурси та спрямовувати їх у потрібний час. Наприклад, аналіз часових рядів може показати, які періоди року є найбільш сприятливими для проведення рекламних кампаній, що дозволяє максимально ефективно формувати рекламний бюджет.

Аналіз коваріації дозволяє оцінити взаємозв'язок між кількома маркетинговими змінними та визначити їхній спільний вплив на продажі. Це дозволяє суб'єктам підприємництва краще розуміти, як різні маркетингові заходи взаємодіють між собою та як вони впливають на кінцевий результат [3].

Застосування статистичних методів також дозволяє оцінювати ефективність маркетингових кампаній у реальному часі. Наприклад, суб'єкти підприємництва можуть використовувати A/B-тестування для порівняння ефективності різних рекламних повідомлень або маркетингових стратегій. Це дозволяє швидко

виявляти найбільш ефективні підходи та коригувати маркетингові кампанії в режимі реального часу, що, у свою чергу, підвищує їхню ефективність.

Щодо прикладів та кейсів з практики найбільших світових корпорацій можна виокремити наступні.

1. *Регресійний аналіз в компанії Procter & Gamble (P&G):* Procter & Gamble, один з найбільших виробників споживчих товарів у світі, використовує регресійний аналіз для оптимізації своїх маркетингових витрат. Завдяки цьому методу вони змогли визначити найбільш ефективні канали реклами та оптимізувати свої витрати на маркетинг. Як результат, P&G збільшили свої продажі на 20% у конкретних продуктових лініях, скоротивши при цьому витрати на рекламу на 15% [4].
2. *Кореляційний аналіз в Amazon:* Amazon використовує кореляційний аналіз для оцінки ефективності своїх рекламних кампаній. Вони виявили, що збільшення витрат на онлайн-рекламу значно корелює із зростанням продажів у певних категоріях товарів. Це дозволило Amazon перенаправити ресурси на найбільш ефективні маркетингові канали, що призвело до збільшення рентабельності інвестицій (ROI) на 25% [5].
3. *Кластерний аналіз у Starbucks:* Starbucks використовує кластерний аналіз для сегментації своїх клієнтів. Вони поділили своїх клієнтів на декілька сегментів за демографічними та поведінковими ознаками. Це дозволило їм створити цілеспрямовані маркетингові кампанії для кожного сегмента, що призвело до збільшення лояльності клієнтів та підвищення продажів на 15% [5].
4. *Дисперсійний аналіз у Coca-Cola:* Coca-Cola застосовує дисперсійний аналіз для оцінки ефективності своїх рекламних кампаній у різних регіонах. Вони змогли визначити, що певні рекламні кампанії є більш ефективними в одних регіонах, ніж в інших. Це дозволило Coca-Cola адаптувати свої маркетингові стратегії для кожного ринку, що призвело до збільшення продажів на 10% у глобальному масштабі [5].
5. *Аналіз часових рядів у McDonald's:* McDonald's використовує аналіз часових рядів для прогнозування попиту на свої продукти у різні періоди року. Вони змогли виявити сезонні тренди та коливання у попиті, що дозволило їм краще планувати свої маркетингові заходи та запаси. Це призвело до зниження витрат на зберігання та збільшення ефективності рекламних кампаній [5].

Таким чином, використання статистичних методів для оцінки ефективності маркетингової діяльності є необхідним для забезпечення конкурентоспроможності суб'єктів підприємництва. Вони дозволяють отримати об'єктивні дані, які можуть бути використані для оптимізації маркетингових стратегій, підвищення ефективності рекламних кампаній та досягнення інших стратегічних цілей.

Список використаних джерел:

1. Стренковська А. Ю., Панченко М. О. Аналіз методичних підходів до оцінки ефективності маркетингової діяльності підприємства. Маркетинг і цифрові технології, 2023. №7.1. С. 149-157.
2. Деділова Т. В. Актуальні аспекти підвищення ефективності маркетингової діяльності підприємства / Т. В. Деділова, І. В. Осьмірко // Проблеми і перспективи розвитку підприємництва : зб. наук. пр. / ХНАДУ. – Харків, 2023. № 1 (30). С. 117-125.
3. Муштай В. А. Використання матричних методів в оцінюванні організації маркетингового планування діяльності підприємства. Інфраструктура ринку, 2021. Вип. 51. С. 194-198.
4. Procter & Gamble Japan (A) Regression Analysis / MBA Resources. URL: https://embapro.com/frontpage/regressioncase/4014-japan-1983 (дата звернення: 10.05.2024)
5. The Strategic Value of Regression Analysis in Marketing Research. URL: https://blog.marketresearch.com/the-strategic-value-of-regression-analysis-in-marketing-research (дата звернення: 10.05.2024)

Ключові слова: статистичні методи, ефективність, маркетингова діяльність, аналіз, оцінка
Keywords: statistical methods, efficiency, marketing activity, analysis, evaluation

ЗМІСТ

ІНФОРМАЦІЙНЕ СУСПІЛЬСТВО: ПРОБЛЕМИ ТА ПЕРСПЕКТИВИ

МАТЕРІАЛИ IX ВСЕУКРАЇНСЬКОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ

24 травня 2024 року
м. Одеса

Відповідальній редактор: Н. І. Логінова

Дизайнер: Р. Р. Дробожур

Електронне видання

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
«ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»

ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

МАТЕРІАЛИ IX ВСЕУКРАЇНСЬКОЇ
НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ

ІНФОРМАЦІЙНЕ СУСПІЛЬСТВО: «ПРОБЛЕМИ ТА ПЕРСПЕКТИВИ»

24 травня 2024 року, м. Одеса