МАТЕРІАЛИ VIII ВСЕУКРАЇНСЬКОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ

ІНФОРМАЦІЙНЕ СУСПІЛЬСТВО: «ПРОБЛЕМИ ТА ПЕРСПЕКТИВИ»

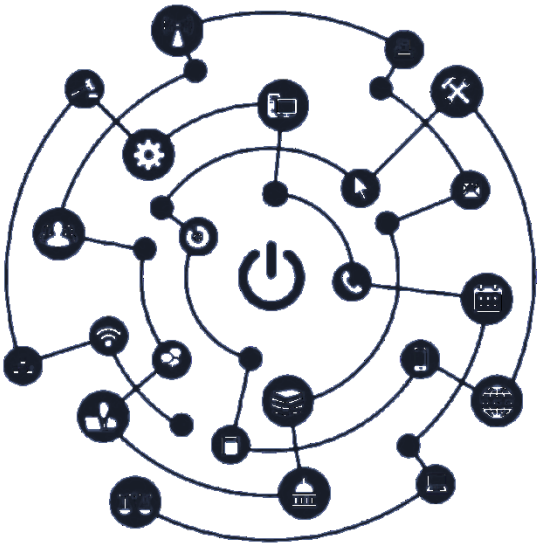
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»

ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ОДЕСА
2 червня 2023 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 VIII ВСЕУКРАЇНСЬКОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ

2 червня 2023 року, м. Одеса

Одеса, 2023

УДК 340.12:316.3

ББК 67.0

Відповідальній редактор:

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

Друкується за рішенням кафедри інформаційних технологій
Національного університету «Одеська юридична академія»
(протокол засідання кафедри № 10 від 19.05.2023 р.)

Матеріали VIII Всеукраїнської науково-практичної конференції «Інформаційне суспільство: проблеми та перспективи» (м. Одеса, 2 червня 2023 р.). Одеса, 2023. 146 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

ІНТЕГРАЦІЯ ФУНДАМЕНТАЛЬНИХ І ПРИКЛАДНИХ ЗНАНЬ ПРИ ПІДГОТОВЦІ СУЧАСНОГО СПЕЦІАЛІСТА

Баландіна Н. М.

старший викладач кафедри кібербезпеки
Національного університету «Одеська юридична академія»

Незважаючи на складні умови, в яких перебувають наука в Україні і люди, котрі нею займаються, майбутнє цивілізації, як і раніше, залишається нерозривно пов'язаним з долею науки. Багато питань, які стосуються долі української науки, обговорюються в пресі. Серед них особливу увагу привертає проблема співвідношення фундаментального і прикладного [1].

Множина дисциплін, маючих підрозділ на фундаментальні та прикладні, складає понад 1000, і кожна з них має власні особливості. Однак, незважаючи на такий поділ, існує чимало наукових дисциплін, з наявними ознаками одразу обох класів. Тому є ряд вчених, які не підтримують явний розділ на прикладні і фундаментальні дисципліни, наводячи приклад того що всі наукові галузі зазвичай виникають із теорії, а вже у ході процесу подальшого розвитку, вони досягають здатності бути застосованими практично.

У системі наук провідне місце належить фундаментальним наукам, які становлять передній край пізнавальної діяльності і власне тому називаються фундаментальними, що на їхній основі можна здійснювати різні прикладні дослідження. Ф. і п. н. перебувають у тісній взаємодії, що постійно посилюється. Відіграючи, поряд із соціальною практикою, вирішальну роль у визначенні осн. напрямів розвитку досліджень у прикладних науках, фундаментальні науки значною мірою впливають на прогрес самих цих наук. У свою чергу, відбувається зворотний вплив прикладних наук на фундаментальні, оскільки результати останніх на практиці можуть бути реалізовані лише через прикладні науки. Нерідко потреби практики виробництва, що виражаються через прикладні науки, викликають і стимулюють постановку відповідних досліджень у фундаментальних науках, приводять до відкриттів фундаментального характеру. Критерієм ефективності фундаментальних наук є не тільки успіх в досягненні об'єктивних знань про предметний світ, а й реальна чи потенціальна можливість практичного застосування цих знань; критерієм ефективності прикладних наук є як міра задоволення соціального замовлення, так і здатність пояснювати природні процеси та явища. Поділ на фундаментальні. і практичні науки є загальноприйнятим

і має принципово важливе значення, зокрема в плані регулювання оптимального співвідношення їхнього розвитку [2].

Взаємовідносини між прикладною та фундаментальною наукою досить складні та суперечливі. Прикладна наука без фундаментальних досліджень часто зазнає невдачі, так само як фундаментальні дослідження досить складно розвивати без прикладних, що впроваджують теоретичні розрахунки на практиці, підтверджуючи тим їх вірність.

Основна відмінність двох складових – це властивості їх результатів. Якщо під час прикладних досліджень вчені можуть спрогнозувати підсумки, то при фундаментальних — ні. У першому випадку випробування проводяться у зв’язку з очікуваннями людей, а в другому усталені теорії можуть руйнуватися. Але при цьому виникають набагато більш цінні знання та вміння [3].

Однак, деякі існуючі відмінності, жодним чином не виключають взаємодії базових і прикладних дисциплін у сучасної дослідницькій діяльності. Фундаментальні дослідження, поряд із прикладними, забезпечують розвиток наукової ідеї та сприяють розвитку технічного прогресу, надаючи значний вплив на великий спектр видів промислової діяльності та технологій виробництва.

Мета фундаментальної науки полягає в тому, щоб знати можливості та способи використання того, що вивчається, а мета прикладної науки в тому, щоб знати способи реалізації тих можливостей.

Фундаментальна наука розвиває наукове мислення та забезпечує виникнення наукових теорій, а прикладна наука розвиває технологічну думку і забезпечує збільшення ефективності та потенціалу промислового виробництва. Цей взаємозв'язок обумовлює ту важливу роль, яку фундаментальна наука відіграє у розвитку прикладної науки. Тому нехтування фундаментальними дослідженнями перешкоджає розвитку як фундаментальної, так і прикладної науки. Отже, необхідно розвивати фундаментальні дослідження та інтегрувати їх із відповідними прикладними дослідженнями.

Якщо фундаментальна наука є ефективним механізмом продукування нового знання, то пріоритетом прикладної науки є створення високопродуктивних технологій [4].

Фундаментальна чи теоретична наука використовує наявний досвід та знання у теоретичних дослідженнях для успішного виконання реальних завдань та втілення проектів – як прикладна математика, де проектування, будівництво та монтаж реальних технічних об'єктів розраховуються за допомогою математичних методів.

Розвиток комп'ютерної техніки розширює можливості математики при

використанні її в практичних завданнях і значно прискорює процес отримання результатів, завдяки високій швидкості та точності розрахунків.

Сучасний фахівець повинен вміти використовувати математичні методи при вирішенні професійних завдань. Це обумовлює неодмінну необхідність якісної математичної підготовки та навичок застосування математичного апарату.

Сенс взаємодії між математичними і прикладними пізнаннями, складається у розкритті спектру можливостей математики в якості базового знання.

На перший погляд, прикладна наука може здатися набагато кориснішою, ніж фундаментальна. Однак фундаментальні знання надають спроможність подальшого практичного застосування. Тобто, фундаментальні знання забезпечують теоретичну основу для прикладних досліджень.

У процесі підготовки спеціалістів з інформаційних технологій, такою основою служить математика. Фаховий рівень майбутніх професіоналів, їх конкурентоспроможність та компетентність, напряму залежать від якості отриманих базових знань з математики.

Від фундаментальності підготовки студентів, а потім і майбутніх фахівців, безпосередньо залежить їх професійний рівень, компетентність та конкурентоздатність. Разом з тим, вони повинні також мати спеціальні знання для вирішення практичних спеціалізованих завдань, що досягається за допомогою інтеграції базових та спеціальних дисциплін.

Отже, інтеграція фундаментальних та практичних досліджень об'єднує два види науки, формуючи єдиний масив знань, здатний надавати суттєву підтримку як майбутнім спеціалістам, так і вже досвідченим фахівцям.

Список використаних джерел:

1. Гончаренко С. Про фундаментальні і прикладні педагогічні дослідження, або «Не споруджують освіту на піску». URL: Про фундаментальні і прикладні пед.дослідження.pdf (iitta.gov.ua)
2. Мороз О.Я. Фундаментальні і прикладні науки. URL: https://leksika.com.ua/14960517/ure/fundamentalni_i_prikladni_nauki
3. Фундаментальні та прикладні науки – опис, функції, зв’язок. URL:https://naukozavr.info/biologiya/fundamentalni-ta-prykladni-nauky/
4. Павко А.І. Вітчизняна наука та європейські цінності - Голос України. URL: http://www.golos.com.ua › article

Ключові слова: фундаментальні науки, прикладні науки, наукові дослілження, практичнее застосування

Keywords: fundamental sciences, applied sciences, scientific research, practical application

ЦИФРОВА ДЕРЖАВА: ПРОБЛЕМИ ВПРОВАДЖЕННЯ SMART-ІНФРАСТРУКТУРИ В УКРАЇНІ

Бердникова М. О.

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

Для багатьох держав, процес впровадження цифровізації в галузі функціонування та управління державними апаратами, став основним напрямком сучасного розвитку. В Україні процес формування та подальшого розвитку інформаційного суспільства бере початок з 1998 року, а саме з прийняттям наступних законів – Закона України "Про концепцію Національної програми інформатизації" та Закону України "Про Національну програму інформатизації".

Поступовий процес цифровізації життєдіяльності суспільства складався з трьох етапів. Головною ідеєю першого етапу стала розробка національних стандартів у галузі інформатизації, створення інформаційно-пошукових систем, інформаційних ресурсів, запровадження систем захисту інформації. Наукові дослідження характерні для даної фази в більшості зосереджені на технічних аспектах побудови інформаційних систем і не включають аналіз теоретичних засад побудови інформаційного суспільства.

Другий етап розвитку інформаційного суспільства умовно бере початок з 2007 року, після прийняття законів України «Про захист інформації в інформаційно-телекомунікаційних системах» та «Про основні засади розвитку інформаційного суспільства в Україні на 2007–2015 рр.». Впродовж другого етапу було остаточно сформовано нормативно-правові засади розвитку інформаційного суспільств.

Основою метою третього етапу є запровадження та функціональне врегулювання системи "Електронного уряду".

Інформаційна система "Електронний уряд" є комплексною системою, яка забезпечує збір, введення, пошук, обробку, збереження та видачу інформаційних ресурсів користувачеві згідно з встановленими критеріями. Ідея е-уряду полягає у створенні умов надання широкого спектру державних послуг органами влади, бізнесу та громадянам шляхом використання інформаційно-телекомунікаційних засобів. Завдяки прозорості процесів проведення цифрових операцій, впровадження електронного управління має вагомий антикорупційний ефект, що призводить до унеможливлення проведення більшості застарілих бюрократично-корупційних схем.

Основною метою створення е-урядування є – впровадження урядового

порталу та регіональних порталів органів влади, інформаційного порталу, через який фізичні та юридичні особи можуть отримувати доступ до всіх необхідних ресурсів. Створення платформи для державних послуг дозволяє об'єднати різні системи та сервіси, забезпечуючи їх взаємодію та інтеграцію. Перехід систем державного управління на цифрові технології здійснюється відповідно до міжнародних стандартів, єдиних рекомендацій і критеріїв цифрового розвитку, що впроваджуються наднаціональними організаціями. З таких критеріїв можна виділити: положення Єдиного цифрового ринку ЄС, рекомендації з цифрового держуправління ОЕСР, а також домовленості щодо розвитку технологій в рамках G8 та G20 можуть слугувати прикладами.

Активна реалізація ідеї smart-міста на міжнародному рівні розпочинається на початку 2000-х років. В українських містах широко поширені технології розпізнавання обличь, які дозволяють безконтактної оплати в торговельних мережах та у громадському транспорті. Також впроваджується "розумна" система освітлення, яка реагує на рух транспорту та збирає інформацію про стан довкілля, таку як температура, якість повітря, рівень шуму і т.д. В багатьох містах вже встановлені smart-лічильники споживання води та газу, а також зростає кількість використання "розумних" сміттєвих контейнерів.

При розбудові smart-інфраструктури в українських містах необхідно використовувати наступні підходи:

1. Зосередження на потребах мешканців: smart-інфраструктура повинна бути спрямована на забезпечення комфортних умов життя для мешканців міста, важливо враховувати особливості їх способу життя, культури та моделей поведінки.

2. Забезпечення безпеки: smart-інфраструктура повинна враховувати небезпеки, з якими можуть зіштовхуватись міста. Технології націлені на забезпечення захисту повинні бути впроваджені для створення безпечних умов для громадян.

3. Гнучкість та адаптивність: smart-інфраструктура повинна мати можливість легко адаптуватися до подальших модифікацій та удосконалень. Технологічні рішення мають бути гнучкими, щоб відповідати змінюваним потребам міста та новим вимогам.

У реалізації проектів з впровадження smart-інфраструктури в Україні існує кілька проблем, що уповільнюють їх швидке втілення:

1. Відсутність стабільного доступу до Інтернету.

2. Недостатній рівень поширення концепції “smart-міста”, яка повинна розвиватися не лише на рівні місцевої влади та недержавного сектору, але й серед жителів.

3. Відсутність чітких показників для оцінки ефективності впровадження smart-інфраструктури.

4. Недостатня стандартизація та сертифікація smart-обладнання.

5. Інертність державних і муніципальних структур.

6. Низький рівень фінансування та відсутність сприятливого інвестиційного клімату для залучення необхідних інвестицій, включаючи іноземні.

Список використаних джерел:

1. Чукут С., Клименко І., Линьов К. Електронний уряд : Науково-практ. довід. С. 17–32. URL: http://surl.li/hfidw.
2. Цифрова економіка : підручник / Т. І. Олешко, Н. В. Касьянова, С.Ф. Смерічевський та ін. К. : НАУ, 2022. 200 с.
3. Маркевич К. Smart-інфраструктура у сталому розвитку міст: світовий досвід та перспективи України : Аналіт. доп. Київ : «Заповіт», 2021. 398 с.

Ключові слова: інформаційне суспільство, цифровізація, smart-інфраструктура
Keywords: information society, digitalization, smart-infrastructure

Науковий керівник: доцент Лобода Ю.Г.

МІСЦЕ ІТ СКЛАДОВОЇ У ПАРАДИГМІ СТАЛОГО РОЗВИТКУ

Горбаченко С. А.
доктор економічних наук, професор, завідувач кафедри кібербезпеки Національного університету «Одеська юридична академія»

Разінкін Н. С.
аспірант кафедри статистики та ММЕ Одеського національного економічного університету

У сучасному світі розробка і застосування інформаційних технологій (ІТ) виявляються невід'ємною частиною парадигми сталого розвитку. ІТ-індустрія має великий потенціал для покращення якості життя, забезпечення економічного зростання та збереження природних ресурсів.

По-перше, інформаційні технології впливають на економічну сторону сталого розвитку. Вони сприяють створенню нових робочих місць, збільшенню продуктивності та конкурентоспроможності підприємств. ІТ допомагають автоматизувати бізнес-процеси, оптимізувати управління ресурсами і знижувати витрати. Наприклад, впровадження хмарних технологій дозволяє підприємствам ефективніше використовувати обчислювальні потужності, зменшуючи споживання енергії та кількість потрібного обладнання [1].

По-друге, ІТ можуть бути важливим інструментом для забезпечення

екологічної сталості. Вони дозволяють моніторити стан навколишнього середовища, прогнозувати зміни клімату та ефективно використовувати ресурси. Застосування сенсорних мереж, інтернету речей (IoT) і систем штучного інтелекту дозволяє збирати дані про стан навколишнього середовища, виявляти екологічні проблеми та шукати раціональні рішення. Наприклад, ІТ можуть бути використані для керування енергоефективними системами, впровадження "розумних" мереж і транспортних систем, що допомагають зменшити викиди в атмосферу та покращити якість повітря [2].

По-третє, інформаційні технології мають значний соціальний вплив у сфері сталого розвитку. Вони сприяють зміні способу життя людей, поліпшенню доступу до освіти, охорони здоров'я та інших соціальних послуг. ІТ роблять можливими дистанційне навчання, телемедицину, електронне урядування та інші інноваційні рішення, які забезпечують рівний доступ до ресурсів та послуг незалежно від географічного розташування. Крім того, ІТ можуть використовуватися для сприяння соціальній взаємодії, підтримки громадської участі та співпраці, що сприяє розвитку демократії та сталих суспільств.

Додатково, варто відзначити деякі конкретні приклади використання ІТ для досягнення сталого розвитку.

В сфері енергетики, ІТ дозволяють ефективніше використовувати та керувати енергійними системами. Системи моніторингу та управління енергопостачанням на основі смарт-мереж (Smart Grids) допомагають зберігати енергію, знижувати втрати та оптимізувати її розподіл. Такі системи також підтримують інтеграцію відновлюваних джерел енергії, сприяючи зменшенню залежності від вуглеводневих палив та зниженню викидів парникових газів [3].

У сфері транспорту, ІТ використовуються для покращення мобільності та зменшення впливу на довкілля. Електронні системи управління транспортним рухом допомагають зменшити затори та витрати палива, а також покращити безпеку на дорогах. Технології "розумного" паркування та спільного використання автомобілів (car-sharing) зменшують кількість автомобілів на дорогах та сприяють ефективному використанню ресурсів.

В сфері сільського господарства, ІТ можуть бути використані для розвитку "розумних" агротехнологій та точного землеробства. Системи моніторингу ґрунту та рослин, дрони, сенсори та аналітичні програми допомагають збирати дані та надавати фермерам цінну інформацію для оптимального використання ресурсів, зниження впливу на довкілля та підвищення врожайності [4].

Крім того, ІТ можуть використовуватися для покращення управління водними ресурсами, усунення нерівностей у доступі до освіти та охорони здоров'я,

а також для створення нових екологічно чистих продуктів та послуг.

Проте, важливо пам'ятати, що використання ІТ для сталого розвитку повинно супроводжуватися урахуванням етичних, соціальних та екологічних аспектів. Забезпечення кібербезпеки, захисту приватності даних та врахування соціокультурних факторів є важливими вимогами для сталого впровадження ІТ-інновацій.

Заключно, ІТ складова виявляється необхідною для досягнення сталого розвитку у сучасному світі. Вони допомагають забезпечити економічне зростання, екологічну сталість та соціальну справедливість. Продовження інновацій та розвитку ІТ-галузі є важливим завданням для побудови стійкого майбутнього. При цьому, важливо враховувати етичні аспекти використання ІТ, захищати приватність даних та забезпечувати кібербезпеку, щоб мінімізувати негативний вплив на людей та навколишнє середовище.

Усе враховуючи, інформаційні технології відіграють ключову роль у парадигмі сталого розвитку. Їх вплив охоплює економічний, екологічний та соціальний аспекти сталого розвитку. Використання ІТ сприяє створенню ефективних та інноваційних рішень для вирішення глобальних викликів, забезпечує рівний доступ до ресурсів та послуг, а також сприяє збереженню природних ресурсів та покращенню якості життя людей. Оскільки ІТ галузь продовжує розвиватися, важливо забезпечити її сталий розвиток та використання з урахуванням цінностей сталого розвитку.

Список використаних джерел:

1. Василевська М.Д. Роль інформаційних технологій у бізнесі. Сучасні виклики та аспекти інноваційного розвитку економічної науки і практики. Київ, 2021. С.58-62.
2. А. В. Яцишин, О. О. Попов, В. О. Артемчук, В. О. Ковач, І. С. Зінов'єва, Автоматизовані інформаційні системи підтримки прийняття управлінських рішень у галузі екологічної безпеки. 2019.
3. Що таке Smart Grid. URL: https://nv.ua/ukr/ukraine/so-skorostyu-sveta/shcho-take-smart-grid-50055452.html
4. Каршерінг – що це таке, як працює і навіщо потрібен. URL: https://termin.in.ua/karsherinh/

Ключові слова: ІТ, сталий розвиток, технології, екологія, інновації
Keywords: IT, sustainable development, technology, environment, innovation

РОЛЬ ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ У СТАЛОМУ РОЗВИТКУ СУСПІЛЬСТВА

Спесівцев М.А.

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

Роль технологій у всіх сферах нашого життя стає все більш важливою. Вони прискорюють та спрощують різноманітні операції, що дозволяє ефективно здійснювати процеси. Тому все більше установ і організацій прагнуть використовувати передові технології, оскільки вони є вигідними як для них самих, так і для користувачів.

Закон України про Національну програму інформатизації прийнятий від 01.12.2022 [1] визначає подальшу стратегічну спрямованість країни у використанні інформаційно-комунікаційних технологій (далі - ІКТ) для досягнення сталого розвитку. Роль ІКТ у сталому розвитку суспільства виконує важливу функцію. ІКТ мають потенціал вирішувати екологічні, соціальні та економічні проблеми, з якими стикається людство.

Одним з основних напрямків використання ІКТ є ефективне управління ресурсами та енергією. Це включає в себе застосування сучасних технологій для моніторингу, аналізу та оптимізації використання різних ресурсів, таких як вода, енергія та матеріали. Завдяки сенсорним мережам, Інтернету речей та системам аналітики даних можна збирати та аналізувати інформацію, що дозволяє виявляти неефективність та займатися енергоефективним управлінням. Крім того, такі технології сприяють впровадженню, наприклад, розумних систем енергопостачання та управління електричними мережами, що сприяє ефективному використанню енергії та стабільному забезпеченню енергопостачанням.[3]

ІКТ також впливають на сталу мобільність та транспортні системи. Розумні транспортні системи, електромобільність, спільне використання транспорту та мобільні додатки сприяють ефективному використанню транспорту, зменшенню заторів, викидів шкідливих речовин та поліпшенню доступності транспорту для всіх верств населення.

ІКТ мають значний вплив на здоров'я та благополуччя людей. Технології телемедицини дозволяють надавати медичну допомогу на відстані, моніторити стан здоров'я та використовувати мобільні додатки для підтримки здорового способу життя. Це сприяє покращенню доступності медичних послуг, виявленню захворювань на ранній стадії та запобіганню їх поширенню.

Цифрові інновації істотно впливають на стале виробництво та споживання,

впровадження цифрових технологій у виробничі процеси, таких як Інтернет речей, автоматизація та штучний інтелект, допомагає ефективно використовувати ресурси, виробляти товари за запитом та зменшувати відходи. У сфері споживання, цифрові платформи та розумні додатки дозволяють споживачам зробити освічений вибір, споживати з урахуванням екологічних та соціальних аспектів.

Важливим аспектом ролі ІКТ у сталому розвитку є забезпечення доступу до інформації та освіти для всіх верств суспільства. Цифрові технології можуть допомогти подолати цифровий розрив та забезпечити рівні можливості для всіх у доступі до освіти, знань та інформації. Це сприяє інклюзивному розвитку та створенню рівних умов для всіх.

Інформатизація в Україні є передовою та має суттєвий вплив на життя громадян. Розробка системи "Дія" є одним з найяскравіших прикладів цього прогресу. Ця інноваційна платформа надає зручні та ефективні інструменти для взаємодії з державою.

Система "Дія" забезпечує онлайн доступ до багатьох сервісів, спрощуючи різні аспекти життя українців. За допомогою цієї платформи громадяни можуть звертатися до державних установ, здійснювати платежі за послуги, відстежувати стан своїх заявок та багато іншого. Це значно економить час та зусилля українців, дозволяючи їм отримувати необхідну допомогу та здійснювати необхідні дії швидше та зручніше.

Також варто відзначити, що інформатизація в Україні сприяє розвитку бізнесу та стимулює технологічну індустрію. Українські ІТ-компанії зарекомендували себе як провідні гравці на світовому ринку завдяки високоякісним програмним рішенням та інноваційним розробкам. Вони активно співпрацюють з партнерами з усього світу та створюють продукти, що впливають на різні сфери життя, включаючи медицину, освіту та економіку.

Інформатизація в Україні також відкрила нові можливості для громадської участі та залучення громадян до прийняття рішень. Електронні петиції, платформи для обговорення законопроектів та громадських ініціатив дають українцям можливість висловлювати свої думки та співпрацювати з державою у формуванні політики.

Загалом, роль ІКТ у сталому розвитку суспільства полягає в їх потенціалі для створення екологічно стійкого, соціально справедливого та економічно ефективного суспільства. ІКТ можуть вирішувати проблеми ефективного використання ресурсів, покращення якості життя людей, забезпечення сталого виробництва та споживання, а також забезпечення доступу до інформації та освіти. Для досягнення цього потрібні спільні зусилля уряду, бізнесу, громадськості та активної

взаємодії між різними секторами для розвитку та впровадження ІКТ-інновацій у всіх сферах життя.

Список використаних джерел:

1. Золотар О., Матвійчук І. Інформаційне право. Науковий часопис Національного педагогічного університету імені М.П. Драгоманова. 2018. Т. 18, № 33. С. 49–52. URL: https://cutt.ly/iwqp4iH4 (дата звернення: 21.05.2023).
2. Про Національну програму інформатизації : Закон України від 01.12.2022 р. № 2807-IX. URL: https://cutt.ly/EwqgCScB (дата звернення: 23.05.2023).
3. М-во економічного розвитку та торгівлі України. Цифрова адженда України – 2020. URL: https://cutt.ly/Ywqp7QS6 (дата звернення: 22.05.2023).

Ключові слова: технології, інформаційне суспільство, інформаційно-комунікаційні технології, інновації

Keywords: technologies, information society, information and communication technologies, innovations

Науковий керівник: доцент Лобода Ю.Г.

РОЗРОБКА ІНФОРМАЦІЙНОЇ СИСТЕМИ ЕЛЕКТРОННОГО ОБЛІКУ НАВЧАЛЬНИХ ПРОЦЕСІВ

Вівсяний В. Г.

студент 4-го курсу Інституту комп'ютерних систем Національного Університету «Одеська Політехніка»

Рудніченко М. Д.

кандидат технічних наук, доцент кафедри інформаційних технологій Національного Університету «Одеська Політехніка»

В сучасному світі, де технології проникають у всі сфери нашого життя, освіта також не залишається осторонь цих змін. У сучасних школах виникає все більша потреба в ефективному управлінні інформацією, організації навчального процесу та спрощенні адміністративних процедур.

З метою задоволення цих потреб, виникає необхідність в розробці і впровадженні інформаційних систем для шкіл. Ці системи допомагають автоматизувати процеси управління школою, взаємодії зі студентами, вчителями та адміністраторами, а також забезпечують збереження та доступ до важливої інформації [1].

Дана робота присвячена розробці та реалізації інформаційної системи для шкіл, яка дозволить школам управляти студентами, вчителями, класами, предметами, домашніми завданнями та оцінками. Користувачами даної системи є учні, вчителі та адміністратори шкіл.

Головні функціональні можливості даного програмного комплексу задовольняють найважливіші потреби шкіл, до них відноситься:

1. Створення користувачів системи (учнів, вчителів) адміністраторами.
2. Створення класів та додавання предметів.
3. Додавання домашнього завдання до предметів.
4. Виставлення оцінок вчителями.
5. Ведення вчителями журналу відвідуваності.
6. Перегляд домашнього завдання й оцінок учнями.

Мета цієї роботи – створити інформаційну систему, яка покращить ефективність та зручність управління шкільними процесами, сприятиме підвищенню якості навчання та розвитку освіти у сучасному світі.

Одним із ключових викликів у реалізації таких систем є відсутність необхідної інфраструктури та досвіду для розгортання та підтримки власного back-end сервера. Тому було вирішено використовувати сервіс Firebase, який допоможе вирішити дану проблему, з його функціями бази даних, хмарних функцій, хостингу тощо.

Серед основних переваг використання хмарних функцій Firebase є можливість повністю виключити back-end функціонал з клієнтського додатку. Замість того, щоб виконувати операції прямо на клієнті, ми можемо створити захисні середовища (Cloud Functions), в яких будуть виконуватися всі важливі операції. Клієнтський додаток тепер не бачить прямих шляхів доступу до бази даних, а отримує тільки назву функції, яку треба запустити. Наприклад, замість додавання нового запису безпосередньо до бази даних, клієнтський додаток викликає відповідну хмарну функцію, яка вже самостійно здійснить додавання запису. Такий підхід є важливим і повинен використовуватися в даній клієнт-серверній системі.

Іншим важливим аспектом використання хмарних функцій Firebase є можливість контролювати обмеження доступу до конкретних областей бази даних. За допомогою ідентифікатора користувача та перевірки його ролі, ми можемо встановити правила доступу до певних операцій. Наприклад, для читання домашніх завдань та оцінок доступ мають як вчителі, так і учні, але виставляти домашні завдання й оцінки можуть тільки вчителі. Завдяки цьому механізму, ми забезпечуємо відповідну безпеку та контроль над доступом до конфіденційної інформації.

В якості бази даних використовується Cloud Firestore, нереляційна документо-орієнтована база даних. Хоча в такій архітектурі бази даних не існує строгих взаємозв’язків між сутностями, у даній системі було прийнято рішення створити псевдо-модель та мати нормалізовану структуру даних [2]. Задля цього в кожному документі сутності, що має логічний зв’язок з іншою сутністю міститься посилання на відповідний документ у вигляді строки з UID такого документу. Наприклад, кожен документ (клас з учнями) колекції класів, в яких викладає вчитель, має посилання на оригінальний документ відповідного класу в загальній колекції класів школи. Таким чином, дані зберігаються не в денормалізованому

стані (який у випадку змін в одному місці змусив би робити багато операцій запису до БД), а у вигляді посилань на оригінальні документи. Дані посилання у випадку системи, що розробляється, використовуються хмарними функціями, які вставляють UID відповідного документу в запит й одержують дані, що потім будуть передані на клієнт.

До основних сутностей належать наступні: учні, вчителі, оцінки, домашні завдання, уроки, розклад уроку, предмети вчителя, класи вчителя.

Структуру такої документо-орієнтованої бази даних можна описати наступним чином. БД має дві головні колекції: школи та користувачі («users»). У колекції користувачів зберігаються всі користувачі зі всіх шкіл та їх основні облікові дані, а в колекції шкіл, власне, школи. Кожна школа має колекції загальних предметів (предмети за замовчуванням), вчителів, класів, студентів та адміністраторів. Наприклад, колекція класів має такі підколекції: студенти, предмети (які в свою чергу мають підколекції: розклад, домашні завдання, оцінки). Доступ до кожної з них контролюється за допомогою хмарних функцій, а не правил безпеки Firestore. Останні, в свою чергу, дають доступ тільки до відповідного документу користувача з колекції «users».

Враховуючи три ролі користувачів у системі (адміністратори, учні та вчителі), можна розділити програмний комплекс на два окремих клієнтських додатки. Перший буде націлений виключно на роль адміністратора, а другий буде поєднувати функціонал як учителя, так і учня. У додатку звичайного користувача роль буде визначатися під час процесу входу в систему за допомогою адреси електронної пошти та паролю. Система автоматично визначить, ким є даний користувач, збереже його дані й надалі буде відкривати домашню сторінку відповідно до його ролі.

В якості технології створення клієнтських додатків обрано фреймворк Flutter, який разом із мовою програмування Dart дозволяє створювати кросплатформні додатки, використовуючи єдину кодову базу [3]. Завдяки цьому, можливо створити мобільний додаток для вчителів та учнів і в майбутньому додати його веб-версію, зробивши декілька правок в основному коді, тим самим розширити кількість підтримуваних платформ для зручності кінцевого користувача.

Додаток для адміністраторів поставлятиметься у веб-версії й також створюється за допомогою Flutter. Таким чином, можна зосередитися на одній мові програмування та технології створення інтерфейсів, що значно прискорить процес розробки та зробить його більш продуктивним. Завдяки можливостям хостингу від Firebase додаток адміністратора буде розміщено за постійною URL-адресою в мережі Інтернет.

Однією з важливих функцій адміністратора є створення й додавання в систему нових користувачів, а саме учнів та вчителів. Адміністратор вносить необхідні облікові дані, обирає роль користувача, вводить його адресу електронної пошти та натискає кнопку «Додати». Після цього в колекції «users» бази даних буде створено новий запис, у відповідну ролі користувача колекцію (учні або

вчителі) буде додано новий документ, а на вказану електронну пошту буде відправлено реєстраційні дані, зокрема згенерований системою пароль, який користувачу необхідно буде ввести при першому вході в систему. Таким чином, усувається потреба в самостійній реєстрації, наприклад, учня, і посилюється захист облікового запису, адже доступ до поштової скриньки має тільки кінцевий споживач.

Особливістю цифрового обліку успішності учнів є можливість додавання вчителем коментарів до кожної оцінки. Таким чином, система дозволяє зберегти конфіденційність навчальних досягнень кожного учня за рахунок виключення необхідності оголошувати оцінку перед усім класом та підвищує успішність та мотивацію учнів [4].

Для зручності користувачів обох додатків можна реалізувати функції імпорту/експорту деяких даних, а саме: для адміністраторів – імпорт користувачів з Excel-документу для пакетного створення облікових записів та експорт необхідних таблиць для створення звітів, для вчителів – експорт таблиці оцінок та відвідуваності у документ формату Excel.

Висновки. Створення інформаційної системи для шкіл є важливим завданням у сучасному світі, де освіта набуває все більшої цифрової спрямованості. Результатом розробки описаного проекту є інформаційна система, яка сприятиме підвищенню якості навчання та ефективності управління шкільними процесами у сучасному освітньому середовищі.

Список використаних джерел

1. Лондар С. Л. Розвиток інформаційно-аналітичних систем управління у сфері загальної середньої освіти. Розвиток інформаційних систем управління освітою як інструмент реалізації державної освітньої політики : монографія / ред. С. Л. Лондар. Київ, 2020. С. 7–8.
2. Tkachuk I. Типи баз даних: особливості, відмінності та приклади. URL: https://dou.ua/lenta/articles/types-of-databases/ (дата звернення: 22.05.2023).
3. Miola A. Flutter complete reference: create beautiful, fast and native apps for any device. Independently Published, 2020. P. 24.
4. Про затвердження методичних рекомендацій щодо оцінювання результатів навчання учнів 1-4 класів закладів загальної середньої освіти : Наказ; МОН України від 13.07.2021 № 813 // База даних «Законодавство України» / Верховна Рада України. URL: https://zakon.rada.gov.ua/go/v0813729-21 (дата звернення: 22.05.2023)

Ключові слова: інформаційна система, управління школами, Firebase, кросплатформеність
Keywords: information system, school management, Firebase, cross-platform

ІНФОРМАЦІЙНА СИСТЕМА УПРАВЛІННЯ ПРОЄКТАМИ

Олефіренко О.Ю.

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

Кожна сучасна IT-компанія прагне стати кращою на ринку розробки програмного забезпечення. IT-сфера сьогодення є дуже динамічною, тому IT-компанії повинні постійно слідкувати за діяльністю конкурентів та вміти адаптуватися до змін у строках виконання проєкту, змін у бюджеті, який виділений для проєкту, реагувати на зміни вимог замовника тощо. Тому важко представити діяльність IT-компанії без системи для управління проєктами.

Система керування проєктами — визначення для комплексного програмного забезпечення, що охоплює програми для планування завдань, складання розпису, контролю ціни і керування бюджетом, розподілу ресурсів, спільної роботи, спілкування, швидкого керування, документування та адміністрування системи, яке використовуються спільно для керування великими проєктами [1].

Однією з найбільш поширених можливостей є можливість планування подій і керування завданнями. Вимоги можуть відрізнятися в залежності від того, як використовується інструмент. Найбільш поширеними є [2]:

- планування різних подій, що залежать одна від одної;
- планування розкладу роботи співробітників і керування ресурсами;
- розрахунок часу, необхідного на рішення кожної з задач;
- сортування завдань залежно від термінів їх завершення;
- керування декількома проєктами одночасно.

Інформаційна система для керування проєктами є актуальною і полягає у організації середовища, яке дозволить учасникам процесу розробки програмного забезпечення ефективно виконувати свої повсякденні задачі, а саме:

- керівник компанії буде мати можливість відстежувати всі дії, які виконуються в ході реалізації проєктів;
- менеджери проєктів за допомогою такої системи зможуть ефективно розподіляти людські ресурси для вчасного завершення проєкту;
- решта співробітників (розробники, дизайнери, тестувальники та ін.) зможе знати свій обсяг роботи та відповідно розподіляти свій робочий час.

Більшість компаній користується не однією, а декількома такими інформаційними системами з таких причин:

- одна система не може задовольнити всі потреби користувачів (наприклад, одна система використовується для основних потреб, а інша для

додаткових);

- деяке програмне забезпечення для керування проєктами має велику вартість, і компанії-початківці не мають можливості їх використовувати.

Предметна область, яка розглядається – це компанії з проєктною організаційною структурою. Проектна організаційна структура являє собою структуру для керування важливими видами діяльності в умовах обмеження витрат та строків на виконання [3]. Для виконання проєкту призначається менеджер проєкту та формується проєктна команда, яка займається його реалізацією. Схема проєктної організаційної структури наведена на рис. 1.

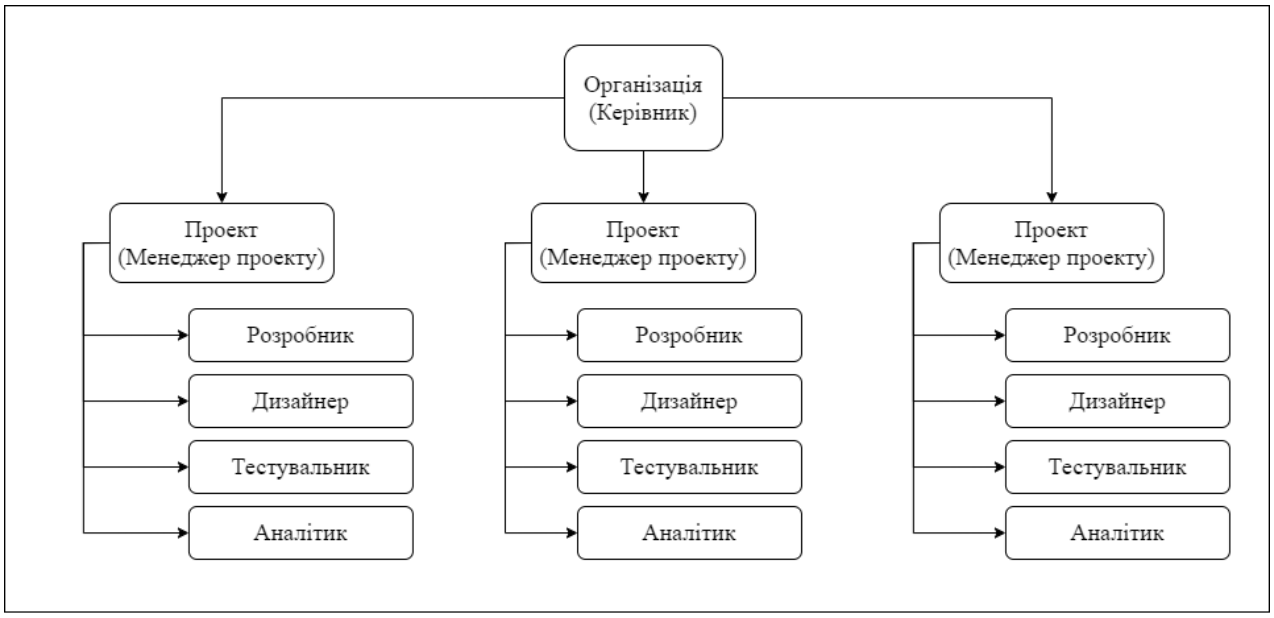


Рисунок 1 – Схема проєктної організаційної структури

Керівник відповідальний за призначення менеджера на проєкт та займається моніторингом усіх проєктів в організації. За потреби керівник виконує коригування строків та складу проєктної команди.

Менеджер проєкту наділений правами контролю відповідного проєкту та підпорядковується вищому керівництву. Саме менеджер відповідальний за всі види діяльності у проєкті, за його вчасне завершення та успіх. Менеджер звітує керівнику про результати виконання проєкту, виконує їх аналіз.

Проектна команда виконує поставлені задачі та звітує про їх виконання менеджеру проєкту, бере участь у нарадах та плануванні проєктної діяльності.

Проектна організаційна структура має наступні переваги:

- всі зусилля проєктної команди концентруються лише на певному проєкті;
- застосовується комплексний підхід для реалізації проєкту та рішення поставлених задач;
- велика гнучкість;
- активізація діяльності менеджерів проєктів разом з виконавцями проєкту

в процесі створення проєктних команд;

- зріст особистої відповідальності менеджерів проєктів за результат виконання проєкту, і як наслідок, формування їх лідерських якостей та особистої зацікавленості в успіху проєкту.

Проте така структура має ряд недоліків, які можуть змусити відмовитися від впровадження такої структури в організації:

- при існуванні великої кількості проєктів в організації, один співробітник може бути залучений до роботи над декількома проєктами водночас, що може призвести до втрати сконцентрованості над своїми задачами;
- застосування такої структури може призвести до труднощів в управлінні людськими ресурсами;
- у випадку, якщо кількість менеджерів є меншою за кількість проєктів, управління проєктами може стати неефективним з причини перевантаженості менеджера;
- затрати на виконання проєкту збільшуються, оскільки потрібно більша кількість персоналу.

Отже, можна зробити висновок про те, що така організаційна структура підходить для невеликих продуктових IT-компаній або для аутсорсингових компаній з невеликою кількістю співробітників, характер робіт яких є проєктним.

Список використаних джерел

1. Janssen, Cory. Project Management Software. Techopedia. URL: https://www.techopedia.com/definition/13132/project-management-software.
2. Nevogt, Dave (17 вересня 2013). 34 Project Management Solutions. Hubstaff. URL: https://blog.hubstaff.com/project-management-software-solutions.
3. Проектна структура управління. URL: https://studme.org/68680/ekonomika/proektnaya_struktura_upravleniya

Ключові слова: управління проєктами, проєктна організаційна структура, IT-компанія.
Keywords: project management, project organizational structure, IT company.

Науковий керівник: доцент Задерейко О.В.

WEB 3.0: РЕВОЛЮЦІЯ ІНТЕРНЕТУ

Сметана В.В.

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

Web 3.0 це третє покоління Інтернету, воно формується просто зараз із розвитком блокчейна і децентралізованих додатків (DApps) та інструментів (наприклад, DAO).

Покоління Web 1.0 з'явилося після поняття що таке Web 2.0, саме з 1991 по 2004 рік функціонував Web 1.0 і являв собою сайти статичної завантаженої інформації. Web 1.0 прозвали «Read only», адже взаємодія була односторонньою - користувачі могли дізнаватися новини, завантажувати музику, тобто тільки вживати контент, а не створювати його. У Web 1.0 не було алгоритмів для фільтрації інтернет-сторінок, через що користувачам було надзвичайно складно знайти потрібну інформацію [1].

Звичний більшості Web 2.0 - період після Web 1.0, відзначився появою соцмереж, таргетованої реклами та нових форматів взаємодії з сайтами, де генераторами контенту стали користувачі. Водночас це є ера нестачі приватності. Користувачі залишають велику кількість персональних даних із власниками різноманітних ресурсів, і ці дані можуть бути викрадені, вилучені або передані до контрольованих органів.

Web 3.0 - це нове покоління Інтернету, яке базується на технологічних інноваціях, таких як блокчейн, штучний інтелект, розподілені обчислення та інші. Воно покликане перетворити Інтернет зі статичної платформи на децентралізовану, інтерактивну та розумну екосистему. Це нова версія інтернету, в якій ресурси і контент належать усім користувачам, а не централізованим серверам великих компаній на кшталт Google і Amazon. При цьому в кожного користувача буде один ключ для входу в усі сервіси. Крім того, користувачі та машини зможуть взаємодіяти з даними безпосередньо. Але для того, щоб це сталося, програми повинні розуміти інформацію як концептуально, так і контекстуально. Маючи це на увазі, двома наріжними каменями Web 3.0 є семантичне павутиння і штучний інтелект [2].

Щоб по-справжньому зрозуміти наступний етап Інтернету, потрібно звернути увагу на ключові особливості Web 3.0:

Децентралізація: Одним з основних принципів Web 3.0 є децентралізація. У цій моделі користувачі не залежать від централізованих організацій або посередників для доступу до послуг та обміну даними. Замість цього, використовуються блокчейн-технології та розподілені мережі, де дані та функціональність розподіляються між багатьма вузлами, що забезпечує більшу прозорість, безпеку та надійність.

Контроль користувача над даними: Web 3.0 надає більший контроль користувачам над їхніми особистими даними. За допомогою криптографії та інших технологій, користувачі можуть вирішувати, які дані зберігаються, як вони використовуються і з ким вони спільно використовуються. Це сприяє збереженню приватності та захисту особистої інформації користувачів.

Розумні контракти: Розумні контракти є важливою частиною Web 3.0. Це програмні коди, які автоматизують і забезпечують виконання угод між різними сторонами без необхідності посередництва. Розумні контракти використовують блокчейн для забезпечення безпеки, достовірності та автоматичного виконання угод.

Інтероперабельність: Web 3.0 прагне до взаємодії та інтероперабельності між різними блокчейн-платформами та протоколами. Це дає змогу різним блокчейн-платформам і протоколам співпрацювати і взаємодіяти між собою, обмінюючись даними і послугами. Це забезпечує більшу гнучкість і ефективність використання блокчейн-рішень і дає змогу створювати складні децентралізовані додатки, що можуть використовувати ресурси та функціональність з різних блокчейн-платформ.

Розширена реальність (AR) і віртуальна реальність (VR): Web 3.0 включає розширену реальність і віртуальну реальність як важливі складові. Це дає змогу користувачам взаємодіяти з цифровим світом через інтерфейси AR і VR, що відкриває нові можливості для навчання, розваг і взаємодії з інформацією.

Штучний інтелект (AI): Web 3.0 використовує штучний інтелект для поліпшення функціональності та персоналізації користувацького досвіду. Штучний інтелект допомагає аналізувати і розуміти великі обсяги даних, прогнозувати і рекомендувати відповідні послуги та контент.

Революція фінансів: Web 3.0 вносить революцію в галузь фінансів, даючи змогу безпечним і децентралізованим фінансовим транзакціям. Завдяки блокчейну і криптовалютам, користувачі можуть здійснювати перекази коштів без посередників, отримувати позики, управляти активами і брати участь у фінансових екосистемах. Web3 використовує токени, такі як ETH, щоб надсилати гроші безпосередньо з браузера, і не вимагає довіряти третім особам. В одному тільки минулому році значно зріс інтерес до криптовалют, рішень "Layer 2", експериментів у царині нових форм колективного управління і революцій у сфері цифрової ідентичності. Зараз же основним двигуном розвитку web 3.0 є криптопроект, адже одна з концепцій 3 покоління є токенезованість контенту. Команди надають технологію для взаємодії з якою потрібні токени проєкту, як засіб обміну в мережі, а також для забезпечення консенсусу та безпеки блокчейну.

Ще одна з поширених функцій токена проєкту - управління. Власники токена беруть участь в управлінні додатком через DAO - орган децентралізованого управління, що працює за допомогою смарт-контрактів [3].

Сума токенів у володінні визначає "вагу" голосу під час голосувань. Члени DAO самі формулюють і обговорюють нові пропозиції, а також голосують за них.

Голосування DAO можуть стосуватися найрізноманітніших питань - починаючи від виділення коштів на якийсь проект і закінчуючи зміною якогось параметра застосунку. Існує багато компаній, які активно впроваджують і розвивають концепцію Web 3.0. Ось кілька прикладів таких компаній:

Cosmos (ATOM): Cosmos є блокчейн-платформою, що дає змогу побудувати децентралізовані додатки та блокчейн-інфраструктуру. Відмінною рисою Cosmos є ABCI протокол, що дає змогу Layer 1 проєктам взаємодіяти, маючи просту мову програмування і великий функціонал продукту, також впроваджується можливість надання іншим проєктам своїх валідаторів для забезпечення безпеки. Ethereum (ETH): Ethereum є однією з найбільш відомих платформ для розробки децентралізованих додатків та розумних контрактів. Вони активно працюють над розвитком Web 3.0 шляхом впровадження шарів масштабування, покращення швидкості транзакцій та забезпечення більшої ефективності.

Polkadot (DOT): Polkadot - це платформа, яка дозволяє взаємодіяти між різними блокчейн-платформами та протоколами. Вони розвивають інтероперабельність між блокчейнами, що допомагає побудувати багатоланкові додатки та прискорює інновації в галузі Web 3.0. XCM (Cross-Chain Messaging Protocol) є ключовим елементом блокчейн-платформи Polkadot. Цей протокол забезпечує механізм для взаємодії між різними блокчейнами в екосистемі Polkadot та сприяє інтероперабельності між ними. Технологія XCM дозволяє передавати активи, повідомлення та дані між блокчейнами, що базуються на Polkadot, безпосередньо та безпечно. Він створює єдиний стандарт комунікації, який дозволяє різним блокчейнам обмінюватися інформацією та взаємодіяти один з одним.

Stellar (XLM): Stellar - це відкрита блокчейн-платформа, яка прагне забезпечити миттєві та вартісні перекази грошей в будь-якій валюті. Вони використовують протокол Stellar Consensus Protocol (SCP), що гарантує безпеку, швидкість і надійність операцій. Враховуючи технологічну гнучкість та інтероперабельність Stellar, платформа може стати потенційною опцією для інтеграції CBDC. Stellar має широкий набір інструментів для емісії та обігу цифрових активів, в тому числі валют, та підтримку фінансових транзакцій з низькими комісіями та швидким підтвердженням.

Загалом, Web 3.0 є важливим кроком у розвитку Інтернету, він має потенціал перетворити спосіб, яким людство сприймає та використовує Інтернет, забезпечуючи більшу свободу, безпеку та контроль для користувачів, перспектива, яку надає Web 3.0, є захоплюючою. Він може перетворити багато сфер життя, включаючи фінанси, торгівлю, соціальні мережі, ланцюг постачання та багато інших. Забезпечення приватності, безпеки та інтероперабельності має потенціал

змінити спосіб, яким людство взаємодіє з технологією та один з одним. Однак, варто зазначити, що розвиток Web 3.0 ще продовжується, і багато з цих технологій знаходяться на етапі експериментів та вдосконалення.

Список використаних джерел:

1. Що таке Web 3.0. URL: https://coinmarketcap.com/alexandria/ru/article/what-is-web-3-0
2. Web 3.0. URL: https://ethereum.org/ru/web3/
3. Навіщо у Web3 потрібні криптовалюти? URL: https://forklog.com/cryptorium/chto-takoe-web3-web-3-0

Ключові слова: Web 3.0, блокчейн, децентралізація інтернет
Keywords: Web 3.0, blockchain, internet decentralization

Науковий керівник: доцент Задерейко О.В.

NEXT.JS: ФРЕЙМВОРК ДЛЯ СУЧАСНОЇ ВЕБРОЗРОБКИ НА REACT.JS

Гоцик В. А.

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

React.js є найпопулярнішою фронтенд JavaScript бібліотекою, про що свідчать тенденції авторитетних серед програмістів платформ GitHub та Stack Overflow [1]. React – це декларативна, ефективна і гнучка JavaScript-бібліотека, призначена для створення інтерфейсів користувача. Вона дозволяє компонувати складні інтерфейси з невеликих окремих частин коду –"компонентів" [2].

Головними перевагами цієї бібліотеки є те, що вона має:

- низький поріг входу;
- зворотну сумісність версій;
- численну спільноту розробників, завдяки підтримці якої можна знайти відповідь на будь-яке питання;
- велику кількість різних розширень тощо.

Але, попри численні переваги, вебзастосунки, розроблені за допомогою React, можуть мати проблеми з продуктивністю та пошуковою оптимізацією сайту (від англ. search-engine-optimization, SEO). Так, проблеми із SEO можуть виникати через те, що розробка вебсторінок на ReactJS відбувається у форматі односторінкового застосунку (від англ. single-page application, SPA), тобто контент підвантажується динамічно засобами JavaScript, а пошукові боти заради оптимізації або взагалі не виконують JavaScript код, або тільки частково. Через це, може

трапитись ситуація, коли пошуковий бот при перегляді вебсайту побачить лише порожню сторінку без контенту. Усунути такого роду проблеми допомагає фреймворк Next.js. Це передовий фреймворк для створення сучасних вебзастосунків, який поєднує потужність React і широкий спектр можливостей для продуктивної розробки застосунків із гарним користувацьким досвідом.

Next.js пропонує два різні методи рендеру вебзастосунку, які розв'язують проблеми з SEO:

- рендер на стороні сервера (від англ. server-side-rendering, SSR);
- генерація статичних сторінок (від англ. static-site-generation, SSG).
- SSR – це технологія, яка забезпечує рендер вебсторінки на сервері, а не в браузері [3]. Коли користувач заходить на сторінку, запит відсилається на сервер, JavaScript вебсайту виконується на сервері, а потім повністю відрендерена сторінка надсилається клієнту. Завдяки цьому клієнт отримує вже готову сторінку.
- SSG – це процес рендеру вебсайту під час його компіляції. Результатом є купа статичних файлів, зокрема файли для HTML, JavaScript і CSS коду. Оскільки сайт рендериться тільки один раз під час компіляції, то на сервері розташована вже готова сторінка. Клієнти при вході на сайт бачать вже відрендерену сторінку [4]. Перевагою цього підходу є швидкість завантаження сторінок, а недоліком є те, що при оновленні контенту сайту його треба перекомпільовувати.

Отже, SSR варто використовувати при розробці вебсайтів із великою кількістю контенту, який часто оновлюється, а в інших випадках – SSG, бо цей підхід дозволяє пришвидшити завантаження сайту.

Окрім усунення проблем із SEO, Next.js надає багато інших корисних функцій, які спрощують процес розробки вебзастосунків. Однією з них є автоматична оптимізація сторінок, яка дозволяє зменшити розмір завантажуваних ресурсів та прискорити швидкість завантаження. Крім цього, Next.js підтримує "гарячу" заміну модулів (від англ. Hot Module Replacement), що дозволяє розробникам бачити зміни в реальному часі без потреби перезавантаження сторінки. Це робить процес розробки більш зручним та ефективним.

Також Next.js надає потужну систему маршрутизації, яка дозволяє легко створювати вкладені та динамічні сторінки сайту. Крім того, він надає можливість розділення коду (від англ. code splitting), що дозволяє завантажувати лише необхідний код для кожної сторінки, що сприяє покращенню швидкості завантаження та продуктивності застосунку.

Врешті решт, Next.js – це фреймворк для сучасної веброзробки, який надає численні переваги у швидкості, продуктивності та SEO-оптимізації. Його підтримка серверного рендерингу, автоматична оптимізація сторінок,

масштабованість та розширюваність роблять Next.js ідеальним вибором для розробки сучасних вебзастосунків.

Список використаних джерел:

1. Stack Overflow trends. URL: https://gist.github.com/tkrotoff/b1caa4c3a185629299ec234d2314e190?permalink_comment_id=3935827
2. Посібник: знайомство з React. URL: https://uk.legacy.reactjs.org/tutorial/tutorial.html#what-is-react
3. What is server-side rendering: definition, benefits and risks. EPAM SolutionsHub. URL: https://solutionshub.epam.com/blog/post/what-is-server-side-rendering/
4. Colby Fayock What is Static Site Generation? How Next.js Uses SSG for Dynamic Web Apps. URL: https://www.freecodecamp.org/news/static-site-generation-with-nextjs/#what-is-static-generation

Ключові слова: Next.js, React.js, фронтенд, вебсайт, вебзастосунок.
Keywords: Next.js, React.js, frontend, website, web application.

Науковий керівник: доцент Трофименко О.Г.

СУЧАСНИЙ ІНСТРУМЕНТАРІЙ ВЕБРОЗРОБКИ

Трофименко О. Г.

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

Орел А. О.

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

Вибір технологій розробки програмних вебсайтів та вебзастосунків, з одного боку, залежить від багатьох критеріїв, а з іншого, саме цей вибір безпосередньо впливає на успішність кінцевого продукту.

Перш ніж розпочати розробку будь-якого вебзастосунку, зазвичай, команда розробників завчасно домовляється про стек технологій, що буде використовувати. Вибір технологічних засобів розробки (системи керування базою даних, мови програмування, відповідних фреймворків та патернів) – дуже важливий етап. Цей вибір залежить від вибору вебсервера, оскільки різні сервери підтримують різні мови програмування, бази даних для зберігання й обробки даних, інтерфейси взаємодії та різну архітектуру. Врешті-решт від вибору стека технологій залежить зручність реалізації функціоналу сайту та час розробки.

Фреймворки є своєрідним набором різноманітних інструментів для створення вебзастосунків, вебсайтів, як для десктопних, так і для мобільних

пристроїв. Станом на 2022 рік у світі серед веброзробників найбільш популярними є такі фреймворки та засоби розробки: Node.js – 47.12%; React.js – 42.62%; jQuery – 28.57%; Express – 22.99%; Angular – 20.39%; Vue.js – 18.82% [1]. Кожен із них має унікальний функціонал, структуру, програмні модулі і компоненти. Вибір того чи іншого фреймворка для розробки frontend частини залежить від складності та масштабу проєкту. Так, фреймворки Angular, React, Vue переважно використовуються для розробки інтерфейсу користувача. Однією з ключових особливостей React є можливість використовувати його в поєднанні з іншими технологіями, такими як Redux, Next.js та іншими. React забезпечує підтримку серверного рендерингу, що дозволяє забезпечувати швидку відповідь сервера на запити користувача [2]. Фреймворк React.js якнайкраще підходить для проєкту середньої складності, має гнучкий підхід та меншу кількість обов'язкових деталей, ніж Angular. Порівняно з Vue.js, React.js має численні бібліотеки, які надають численні, різноманітні функцій [3].

Backend частина сайту та сервера вважається складнішою, у порівнянні з frontend. По суті, це цілий програмно-апаратний комплекс, який стосується реалізації взаємодії з базою даних, API та іншими складовими, що забезпечують роботу серверної частини. На сьогодні є багато технологій, що дозволяють керувати внутрішніми процесами сайту. Серед них: CakePHP, Fiber, Django, Node.js та інші. За результатами опитування StackOverflow за 2022 рік розробники найчастіше використовують фреймворки Node.js (47,12%), React.js (42,62%) та jQuery (28,57%) [4]. Розробники React.js тісно пов'язані з іншими фреймворками та технологіями, особливо з Node.js. Цікаво, що 52,86% опитуваних професійних розробників починали своє навчання з Node.js, а 20,42% – з Django [5].

Найпопулярнішою браузерною мовою програмування є JavaScript (або JS). 67,9% професійних розробників використовують саме її [4]. Її засобами реалізується динамічна складова вебсторінки, а середовищем виконання JS-коду є сам веббраузер. Найпопулярніші фреймворки React.js і Node.js теж використовують мову програмування JavaScript і по суті є своєрідними середовищами розробки.

Перш ніж вибрати засоби розробки, варто зважити усі переваги та недоліки кожного з них. Так, стек технологій MERN складається з чотирьох базових технологій: MongoDB або MySQL, Express.js, React.js та Node.js [6]. Порівняно з MongoDB, база даних MySQL є реляційною, що дає можливість зберігати дані у взаємопов'язаних таблицях. MySQL краще підходить для проєктів, де потрібно зберігати структуровані дані зі строгими взаємозв'язками. MongoDB, в свою чергу, є документо-орієнтованою нереляційною базою даних, яка зберігає дані у вигляді документів формату JSON. Вона добре підходить для проєктів, де дані

можуть бути неструктурованими або мають змінну структуру, наприклад, коли йдеться про соціальні мережі, системи управлінням контентом тощо [7]. Використання React для клієнтської частини, Node.js та Express.js – для серверної частини, а також MySQL – для розробки бази даних, дозволить створити масштабований, швидкий вебзастосунок, що відповідатиме усім поставленим вимогам. Важливо, що стек технологій MERN є популярним і має численну спільноту підтримки, що допомагає находити відповіді на будь-які питання чи то проблеми, які можуть виникати під час роботи над проєктом.

Список використаних джерел:

1. Most used web frameworks among developers worldwide, as of 2022. URL: https://www.statista.com/statistics/1124699/worldwide-developer-survey-most-used-frameworks-web.
2. Керівництво з безпеки React.js: загрози, вразливості та способи їх усунення. URL: https://bestprogrammer.ru/izuchenie/rukovodstvo-po-bezopasnosti-react-js-ugrozy-uyazvimosti-i-sposoby-ih-ustraneniya.
3. Angular vs React vs Vue: Which Framework to Choose. URL: https://www.codeinwp.com/blog/angular-vs-vue-vs-react/#gref.
4. Developer Survey: Most popular technologies. URL: https://survey.stackoverflow.co/2022/#technology.
5. Орел А. О. Вибір технологій для розробки backend сайту. Кібербезпека в сучасному світі: актуальні виклики : матер. IV всеукр. наук.-практ. конф. (25 листопада 2022 р.). Одеса, 2022. С. 75-77. ISBN 978-966-992-297-7
6. MERN Stack Explained. URL: https://www.mongodb.com/mern-stack.
7. Can I use MySQL with Node.js? URL: https://dev.to/catchmareck/can-i-use-mysql-with-node-js-57m.

Ключові слова: вебзастосунок, Node.js, JavaScript, React.js, MySQL, MERN
Key words: web application, Node.js, JavaScript, React.js, MySQL, MERN

ДОСЛІДЖЕННЯ ТЕХНОЛОГІЙ NESTJS ТА EXPRESS ДЛЯ РОЗРОБКИ BACKEND САЙТУ

Козловський І. І.

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

Бекенд вебсайту – це серверна частина, яка забезпечує обробку запитів, обмін інформацією з базами даних та інтеграцію з іншими сервісами. Він відіграє важливу роль у функціонуванні сучасних вебзастосунків, забезпечуючи стабільність та ефективність роботи.

Серверні мови програмування використовуються для розробки серверної

частини вебзастосунків і дозволяють обробляти клієнтські запити, працювати з базами даних та реалізовувати бізнес-логіку. Основні серверні мови програмування, що найчастіше використовуються для розробки бекенду: JavaScript, PHP, Ruby, Python, Java, Rust, Solidity, Go, Kotlin [1]. Крім того, поширено використовується платформа Node.js, яка надає можливість виконувати JavaScript-скрипти на сервері.

Node.js призначена для виконання високопродуктивних мережевих застосунків, написаних мовою програмування JavaScript. В платформі використовується розроблений компанією Google рушій V8. Він забезпечує асинхронну, подійно-орієнтовану роботу, що покращує продуктивність сервера і робить Node.js популярним вибором для розробки бекенду [2].

Node.js з часом здобула значну популярність завдяки своїй продуктивності та широкому спектру використання. Це призвело до появи численних фреймворків, які спрощують розробку серверної частини на основі Node.js. Одними з найпопулярніших і найчастіше використовуваних є фреймворки Nest.js та Express [3].

Nest.js – це фреймворк для створення ефективних, масштабованих серверних програм засобами Node.js. Він підтримує мову TypeScript, оскільки створений на його основі. Це дозволяє поєднувати елементи об'єктно-орієнтованого, функціонального і функціонального реактивного програмування [4].

Express – це мінімалістичний та гнучкий фреймворк для програм Node.js, що надає широкий набір функцій вебзастосунків. Завдяки вбудованим HTTP-методам та проміжному програмному забезпеченню, створення надійного API є швидким і простим процесом [5].

Порівняємо різні характеристики Nest.js і Express, включаючи приклади використання кожного з них:

Архітектура: Nest.js пропонує структурований, модульний підхід, який сприяє організації коду та читабельності, особливо у великих проєктах. З іншого боку, Express.js є мінімалістичним та гнучким фреймворком, але може вимагати додаткових зусиль для створення зручної структури коду.

Погляди: Nest.js – це структура з твердими принципами. Він дотримується парадигми проєктування «угода над конфігурацією», яка дозволяє розробникам використовувати стандартні інструменти та код певним чином, таким чином зменшуючи потребу в явній конфігурації. Express.js – це фреймворк без твердих принципів. Це означає, що він не має набору заздалегідь визначених правил, яких слід дотримуватися.

Популярність: З понад 57 000 зірок на Github, Express є найпопулярнішим

фреймворком. Nest.js слідує за нею. Він має 47 000 зірочок на Github, займаючи друге місце серед кращих фреймворків Node.js за зірками Github.

Типізація: Nest.js вбудовує підтримку TypeScript, яка забезпечує статичну типізацію і покращує безпеку коду. Express працює з JavaScript, розробникам треба самостійно впроваджувати підтримку TypeScript, якщо це необхідно.

Підтримка тестування: Nest.js має вбудовану підтримку модульного тестування, що спрощує написання тестів для різних частин застосунку. У Express впровадження тестування потребує додатковий код, щоб впровадити модульне тестування [6].

Отже, Nest.js є гарним варіантом для розробки бекенду вебсайту, зокрема у ситуаціях, коли:

- проєкт корпоративного рівня;
- безпека та надійність коду є пріоритетом;
- команда розробників дотримується чітких правил написання коду;
- проєкти вимагають високого рівня тестування.
- В свою чергу, Express краще використовувати у ситуаціях, коли:
- проєкт не потребує масштабований код, орієнтований на продуктивність;
- швидкість розробки є пріоритетом;
- вимагається гнучкість та варіативність у виборі архітектури проєкту.

В кінцевому підсумку, вибір між Nest.js та Express залежить від конкретних потреб проєкту. Nest.js надає більшу структурованість, підтримку TypeScript та вбудовану підтримку тестування, що робить його популярним для великих та комплексних проєктів. З іншого боку, Express пропонує простоту та гнучкість, що робить його гарним варіантом для швидкої розробки простих або менш вимогливих проєктів.

Список використаних джерел:

1. The Best 10 Server Side Coding Languages. URL: https://blog.back4app.com/server-side-coding-languages/
2. Node.js documentation. URL: https://nodejs.org/uk/docs
3. 12 Best Nodejs Frameworks for App Development in 2023. URL: https://www.simform.com/blog/best-nodejs-frameworks/
4. Node.js framework documentation. URL: https://docs.nestjs.com/
5. Express 4.18.1. URL: https://expressjs.com/
6. Nest.js vs Express.js. URL: https://blog.logrocket.com/nestjs-vs-express-js/

Ключові слова: Backend, засоби для розробки вебсайту, JavaScript, Typescript, Node.js, Nestjs, Express.

Keywords: Backend, website development tools, JavaScript, Typescript , Node.js, Nestjs, Express.

Науковий керівник: доцент Трофименко О. Г.

ФАЙЛОВІ СИСТЕМИ

Бондар А.М.

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

Файлові системи є необхідною складовою операційних систем, які забезпечують доступ до файлів та даних на сховищі. Файлові системи включають в себе структуру каталогів та файлів, права доступу до файлів, методи зберігання даних на диску та інші функції, що допомагають зберігати та організовувати дані. Розглянемо різні типи файлових систем, їх особливості та застосування.

Одним з найбільш поширених типів файлових систем є файлові системи FAT (File Allocation Table) [1], розроблені компанією Microsoft. FAT-файлові системи були стандартом для дискового зберігання в операційних системах Windows. Найбільш відомими форматами FAT є FAT16 і FAT32.

FAT16 була поширеною в ранніх версіях Windows і мала обмеження щодо розміру файлу та обсягу дискового простору. Вона здатна працювати з файлами розміром до 2 ГБ, а обсяг дискового простору обмежений до 4 ГБ. З впровадженням FAT32 обсяг дискового простору збільшився до 2 ТБ, а максимальний розмір файлу - до 4 ГБ, що покращило сумісність з більшими накопичувачами та файлами.

Однак, з ростом розмірів дискового простору та потреб у зберіганні великих файлів, FAT32 став неефективним. Тому компанія Microsoft почала використовувати новішу файлову систему NTFS (New Technology File System) [2] для зберігання даних у своїх операційних системах Windows.

NTFS пропонує покращену надійність, безпеку та функціональність порівняно з FAT. Вона підтримує розширені права доступу до файлів та папок, шифрування файлів та папок для забезпечення конфіденційності даних, журналювання для покращення відновлення після аварійного відключення та інші розширені функції. NTFS також має вищу ефективність у керуванні дисковим простором та використовує більш ефективну структуру файлів. NTFS підтримує різні рівні доступу до файлів та папок, включаючи розширені права доступу для користувачів та груп, що дозволяє точно настроювати, хто має доступ до конкретних файлів і їхніх функцій. Це корисно у великих організаціях з багатьма користувачами та складними структурами доступу.

NTFS також надає можливість шифрування файлів та папок для захисту конфіденційної інформації. Шифрування забезпечує захист від несанкціонованого доступу до даних, навіть якщо фізичний носій (наприклад, жорсткий диск) потрапить у чужі руки.

Крім того, NTFS має журналювання, що дозволяє відновлювати систему до стану перед аварійним відключенням або іншими системними помилками. Журнальована файлова система зберігає інформацію про зміни, які відбуваються

на диску, що дозволяє системі точно відновлювати дані та стабільно працювати після відновлення.

Загалом, NTFS є потужною та розширеною файловою системою, яка надає покращену надійність, безпеку, керування доступом та інші розширені функції. Вона є стандартом для багатьох операційних систем Windows і є вибором для великих організацій та бізнес-середовищ, де потрібна надійна та безпечна система зберігання даних.

Іншою популярною файловою системою є файлова система HFS+ (Hierarchical File System Plus) [3], яка використовується в операційній системі Mac OS X. HFS+ підтримує розширені атрибути файлів, включаючи атрибути метаданих, які дозволяють зберігати додаткову інформацію про файл, таку як дата створення та дата зміни, а також дозволяє зберігати файли більше 4 ГБ. Її було представлено в 1998 році як наступницю оригінальної файлової системи HFS. HFS+ була основною файловою системою, яка використовувалася в комп'ютерах Apple Macintosh, доки її не замінила файлова система Apple (APFS) [4] у macOS High Sierra (10.13), випущеній у 2017 році.

HFS+ пропонує кілька покращень порівняно зі своїм попередником, зокрема підвищену продуктивність, збільшені обмеження на розмір файлу та покращену підтримку метаданих. Вона підтримує такі функції, як ведення журналу, що допомагає запобігти пошкодженню даних у разі несподіваного завершення роботи системи або збою живлення. HFS+ також підтримує Unicode, що дозволяє використовувати символи з різних мов і шрифтів у назвах файлів.

Однак HFS+ має певні обмеження, як-от неефективне виділення пам'яті для малих файлів і великі витрати метаданих файлів, що може призвести до марної втрати дискового простору. Ці обмеження було враховано в APFS, оптимізованому для сучасних технологій зберігання, включаючи твердотільні накопичувачі (SSD).

Однією з найбільш інноваційних файлових систем є ZFS (Zettabyte File System) [5], яка розробляється компанією Sun Microsystems (пізніше придбаною Oracle). ZFS була спочатку розроблена для операційної системи Solaris, але тепер її можна використовувати на багатьох інших платформах.

Одна з основних переваг ZFS полягає у її надійності та можливості відновлення даних. Вона використовує механізм контрольних сум для забезпечення цілісності даних. Кожен блок даних отримує свою контрольну суму, яка перевіряється під час читання, що дозволяє виявляти та виправляти приховані пошкоджені дані.

ZFS також пропонує функцію збереження даних на базі пулів. Вона дозволяє об'єднувати фізичні пристрої в один пул, який може бути розширюваний та забезпечує спільну потужність зберігання для всіх файлів і даних. Це спрощує управління даними та дозволяє легко додавати нові пристрої до системи.

Крім того, ZFS підтримує розширені функції, такі як снапшоти. Снапшоти дозволяють створювати миттєві знімки файлової системи, що дозволяє зберігати

резервні копії даних на конкретний момент у часі. Вони корисні для відновлення даних з попереднього стану або для відновлення системи у разі аварії або помилкових операцій.

Іншою важливою функцією ZFS є можливість копіювання даних. Вона дозволяє створювати точні копії файлів або даних безпосередньо на файловій системі. Це корисно, наприклад, для створення резервних копій або для створення незалежних копій даних для тестування або експериментів.

ZFS також підтримує можливість створення дзеркальних дисків або RAID-Z, що забезпечує надійність даних шляхом розподілу даних та контрольних сум на різних пристроях [6]. Це дозволяє системі відновлювати дані в разі відмови одного або кількох пристроїв.

Ще однією вагомою перевагою ZFS є підтримка масштабованості. Вона може працювати з великими обсягами даних, дозволяючи легко розширювати пам'ять і додавати нові пристрої до пулу збереження.

Узагалі, ZFS є потужною і надійною файловою системою, яка пропонує широкий набір функцій для управління даними, забезпечення цілісності та надійності, а також для спрощення резервного копіювання та відновлення даних. Вона є популярним вибором для багатьох професійних та високонавантажених середовищ, а також для домашнього використання з великими обсягами даних.

Ще однією популярною файловою системою є файлова система ext4 [7], яка використовується в операційній системі Linux. Ext4 є наступником файлової системи ext3 та має більшу швидкість та надійність. Вона також підтримує більші розміри файлів та дисків, а також можливість розподілу даних на різних рівнях зберігання, що дозволяє досягти кращої ефективності.

Окрім згаданих вище файлових систем, є ще багато інших типів, які використовуються у різних платформах та пристроях. Наприклад, файлова система exFAT, яка розроблялася для використання на з’йомних носіях даних та має підтримку файлів більше 4 ГБ. Існують також файлові системи для використання на мережевих пристроях, такі як NFS (Network File System) та SMB (Server Message Block).

У загальному, вибір файлової системи залежить від багатьох факторів, таких як розмір диску, тип пристрою, який буде використовуватися, та особливості конкретної операційної системи. Наприклад, якщо використовується операційна система Windows, то існує можливість використовувати файлову систему NTFS, яка підтримує розширені можливості доступу до файлів та захисту даних. Однак, якщо використовується операційна система Linux, існує можливість використовувати файлову систему ext4, яка є стандартною файловою системою для більшості дистрибутивів Linux.

Загалом, вибір файлової системи має бути зроблений з урахуванням різних факторів, які впливають на його ефективність та надійність. Наприклад, потрібно зберігати великі обсяги даних, то краще використовувати файлову систему, яка підтримує більші розміри дисків та файлів. Крім того, якщо потрібно зберігати

важливі дані, то краще використовувати файлову систему, яка має підтримку резервного копіювання даних та можливість відновлення даних в разі аварії.

Отже, файлові системи є важливою складовою будь-якої операційної системи та пристрою зберігання даних. Вони дозволяють зберігати, організовувати та керувати доступом до даних на диску. Вибір файлової системи залежить від багатьох факторів та має вплив на ефективність, та можливості надійного зберігання даних. Тому, вибираючи файлову систему, потрібно ретельно розглядати всі її характеристики та функції, щоб забезпечити найкращу продуктивність та надійність зберігання даних.

Список використаних джерел:

1. Файлові системи FAT, FAT32 та NTFS. URL: bit.ly/42I3s6y.
2. Файлові системи різних пристроїв та операційних систем. URL: bit.ly/3Om1BjI.
3. Алгоритм відновлення даних файлової системи HFS+. URL: bit.ly/3pTaSFG.
4. Огляд файлової системи APFS: чому вона краща за HFS+. URL: bit.ly/3BAwsBd.
5. What Is ZFS? URL: bit.ly/3Olu3Sx.
6. Stable IT: Що таке RAID-Z? URL: bit.ly/3pVNWp1.
7. Особливості Ext4. URL: bit.ly/3MAm8iW

Ключові слова: Файлові системи, FAT, Microsoft, Windows, FAT16, FAT32, NTFS, HFS+, Mac OS X, Apple, APFS, Unicode, SSD, ZFS, Sun Microsystems, Solaris, снапшоти, RAID-Z, пул збереження, ext4, Linux, ext3, exFAT, NFS, SMB

Key words: File systems, FAT, Microsoft, Windows, FAT16, FAT32, NTFS, HFS+, Mac OS X, Apple, APFS, Unicode, SSD, ZFS, Sun Microsystems, Solaris, snapshots, RAID-Z, save pool, ext4, Linux, ext3, exFAT, NFS, SMB

Науковий керівник: доцент Задерейко О.В.

ЗАСТОСУВАННЯ XAMARIN.FORMS ДЛЯ РОЗРОБКИ МУЛЬТИПЛАТФОРМОВИХ РІШЕНЬ

Чикунов П.О.

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

У сучасному світі, коли програмісти мають доступ до різних платних та безкоштовних платформ, розробка мультиплатформових рішень є важливою задачею. Цій процес може бути складною і затратною задачею для розробників додатків. Xamarin.Forms – це мультиплатформова платформа корпорації Microsoft, яка дозволяє розробникам створювати додатки для середовищ iOS, Android та Windows, використовуючи мову програмування C# та відому платформу .NET Framework.

Метою доповіді є обговорення доцільності опанування здобувачами платформи Xamarin.Forms для розробки мультиплатформових рішень. Необхідно розглянути переваги використання Xamarin.Forms, а також підходи до викладання цієї технології здобувачам.

При виконанні дослідження проведено аналіз переваг Xamarin.Forms. Встановлено, що до переваг використання Xamarin.Forms можна віднести наступне.

1. Ефективність розробки: Xamarin.Forms дозволяє розробникам створювати мультиплатформові додатки з використанням однієї кодової бази, що робить розробку більш ефективною та менш затратною. Розробники можуть ефективно використовувати свій час та ресурси, оскільки вони можуть написати майже один і той же код для кожної з платформ.

2. Однаковість інтерфейсу та функцій на різних платформах. Xamarin.Forms дозволяє розробникам створювати інтерфейс та функції додатків, що будуть працювати однаково на різних платформах, що полегшує розробку та знижує вартість підтримки.

3. Можливості розширення. Xamarin.Forms дозволяє розробникам розширювати та модифікувати існуючі бібліотеки та функціональність, що дозволяє створювати більш гнучкі та персоналізовані додатки.

4. Інтеграція з платформою .NET. Платформа Xamarin.Forms використовує мову програмування C#, яка має велику спільноту розробників та бібліотек. Розробники можуть швидко знайти рішення для більшості проблем, з якими вони можуть зіткнутися під час розробки.

5. Підтримка Microsoft. Xamarin.Forms є частиною екосистеми Microsoft та отримує підтримку від цього великого технологічного гіганта, що забезпечує стабільну та надійну розробку.

Однак, як і з будь-якою технологією, Xamarin.Forms також має свої недоліки, наприклад, більш високі вимоги до ресурсів, ніж багато інших рішень для розробки мобільних додатків. Xamarin.Forms може мати обмежену підтримку деяких функцій, які доступні лише для окремих платформ. Вивчення Xamarin.Forms може становити проблему для розробників, які не мають попереднього досвіду з мовою C# або з платформами Android, iOS та Windows. Іноді також можуть виникати проблеми зі сумісністю, коли нові версії платформ або бібліотек не сумісні зі старими версіями Xamarin.Forms. Це може призвести до труднощів у розробці та підтримці додатків.

Проаналізовано існуючі підходи до викладання Xamarin.Forms. Далі наведено декілька з них.

1. Онлайн-курси та туторіали. Існують безкоштовні та платні онлайн-курси

та туторіали, які надають студентам базові знання та вміння для розробки мультиплатформових додатків з використанням Xamarin.Forms. Ці курси можуть містити відеоуроки, текстові матеріали та приклади коду, які допомагають студентам зрозуміти основні концепції та принципи розробки з Xamarin.Forms.

2. Курси на кафедрах інформаційних технологій університетів. Університети можуть надавати курси з викладання Xamarin.Forms, як частину програми навчання з розробки мобільних додатків. Ці курси можуть включати лекції, лабораторні заняття та проекти, які допомагають студентам зрозуміти як працює Xamarin.Forms та як створювати мультиплатформові додатки з використанням цієї технології.

3. Онлайн-ресурси. Існують безкоштовні та платні онлайн-ресурси, які допомагають студентам зрозуміти та навчитися використовувати Xamarin.Forms. Ці ресурси можуть містити відеоуроки, блоги, форуми, документацію та приклади коду. До основних можна віднести Xamarin University, Xamarin.Forms Documentation, Xamarin University YouTube Channel.

4. Курси від компанії Xamarin, яка надає курси з викладання розробки мультиплатформових додатків з використанням цієї технології. Ці курси можуть бути платними та надають додаткові можливості для навчання та спілкування з експертами в галузі розробки мультиплатформових рішень.

Ось декілька відомих прикладів проектів на Xamarin.Forms. Storyo – це додаток, що дозволяє користувачам створювати відео-історії з фотографій та відео, які вони зробили. Додаток було створено з використанням Xamarin.Forms, щоб забезпечити мультиплатформову підтримку.

FreshDirect – це додаток онлайн-супермаркету, який дозволяє користувачам замовляти їжу та інші товари онлайн. Додаток було розроблено з використанням Xamarin.Forms, щоб забезпечити мультиплатформову підтримку.

JustWatch – це додаток, який допомагає користувачам знайти стрімінгові сервіси та трансляції на їхніх пристроях. Додаток було створено з використанням Xamarin.Forms, щоб забезпечити мультиплатформову підтримку.

iCircuit – це додаток для моделювання та аналізу електричних схем на мобільних пристроях. Додаток було створено з використанням Xamarin.Forms, щоб забезпечити мультиплатформову підтримку.

CA Mobile – це додаток для управління фінансовими послугами та вкладеннями клієнтів. Додаток було створено з використанням Xamarin.Forms, щоб забезпечити мультиплатформову підтримку.

Список використаних джерел:

1. Xamarin.Forms: Build Native Mobile Apps with C# : Платформа онлайн-курсів Udemy. URI: https://www.udemy.com.
2. Xamarin.Forms: Getting Started : Платформа онлайн-курсів Pluralsight. URI: https://www.pluralsight.com.

Ключові слова: мультиплатформові рішення, Xamarin.Forms
Keywords: multi-platform solutions, Xamarin.Forms

ШТУЧНИЙ ІНТЕЛЕКТ ТА ЙОГО ВПЛИВ НА ЖИТТЯ

Данилова Г.С.

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

Штучний інтелект (ШІ) - це галузь комп'ютерної науки, яка вивчає, як створювати алгоритми та програми, які можуть виконувати завдання, які зазвичай вимагають людського інтелекту. ШІ є важливою технологією, що дозволяє вирішувати складні проблеми та полегшує наше повсякденне життя.

ШІ змінює наше життя в багатьох сферах. У сфері медицини, ШІ використовується для діагностики та лікування різних захворювань. Наприклад, програми ШІ можуть аналізувати медичні зображення, щоб допомогти лікарям виявити ознаки захворювання. Крім того, ШІ може бути використано для створення імітаційних моделей людського тіла, що дозволить проводити експерименти з впливом лік, що автоматично зменшує необхідність у проведенні клінічних випробувань на людях [1].

У сфері транспорту, ШІ може допомогти зменшити кількість дорожньо-транспортних пригод, забезпечуючи безпечний та ефективний рух транспорту. Наприклад, системи допомоги при паркуванні та адаптивного круїз-контролю засновані на технології ШІ.

У сфері фінансів, ШІ використовується для прогнозування поведінки ринків, розпізнавання шахрайства та для створення інвестиційних портфелів. ШІ дозволяє фінансовим установам отримувати більш точні прогнози та знижувати ризики у своїх інвестиційних рішеннях.

Однак, разом зі своїми перевагами, ШІ також може мати негативний вплив на життя, зокрема у сферах праці та безпеки даних.

ШІ може призвести до автоматизації багатьох рутинних та простих завдань, що може звільнити багато людей від монотонної роботи. Однак, це також може

призвести до втрати робочих місць та зростання безробіття в окремих сферах. Для того, щоб запобігти цьому, необхідно забезпечити перекваліфікацію та навчання працівників у нових сферах, які з'являться завдяки розвитку ШІ [2].

Ще однією проблемою, пов'язаною з ШІ, є безпека даних. У цифровій сфері, де все більше і більше даних зберігається в електронному вигляді, необхідно забезпечити безпеку цих даних від зловмисників. ШІ може бути використано для створення програмного забезпечення, яке здатне виявляти та запобігати кібератакам. Однак, це також може створити нові можливості для кіберзлочинців, які будуть використовувати ШІ для зловмисних цілей.

У підсумку, ШІ є надзвичайно важливою технологією, яка впливає на багато аспектів нашого життя. ШІ може забезпечити більш точні та ефективні рішення у різних сферах, від медицини до транспорту та фінансів. Однак, ШІ також може давати негативний вплив на кількість робочих місць та безпеку даних, що вимагає уважності та заходів для їх запобігання. Тому, для того, щоб забезпечити максимальну вигоду від розвитку ШІ, необхідно продовжувати вивчати та розробляти етичні та безпечні стандарти використання ШІ, а також забезпечувати навчання та перекваліфікацію працівників, які можуть втратити свої робочі місця унаслідок впровадження ШІ.

Ще однією важливою проблемою, пов'язаною з ШІ, є етика та законність використання цієї технології. ШІ може бути використано для створення систем масового спостереження, що може порушувати приватність та права людини [3].

Тому, для того, щоб запобігти негативному впливу ШІ, необхідно розробляти етичні та правові стандарти для використання цієї технології. Наприклад, Європейський союз розробляє новий законопроект про штучний інтелект, який має забезпечити захист прав людини та приватності, а також визначити відповідальність за можливі наслідки використання ШІ.

У підсумку, ШІ є надзвичайно важливою технологією, яка може впливати на багато аспектів нашого життя. ШІ може забезпечити більш точні та ефективні рішення у різних сферах, але також може мати негативний вплив на робочі місця, безпеку даних, приватність та права людини. Для того, щоб забезпечити максимальну вигоду від розвитку ШІ та запобігти можливим наслідкам, необхідно продовжувати вивчати та розробляти етичні та безпечні стандарти використання ШІ, а також забезпечувати навчання та перекваліфікацію працівників, які можуть бути втягнуті у процес автоматизації [4].

Крім того, ШІ може бути використаний для розв'язання глобальних проблем, таких як боротьба зі зміною клімату, енергоефективність та виробництво продуктів без використання пестицидів. Такі застосування можуть мати великий

позитивний вплив на наше суспільство та навколишнє середовище [5].

У зв'язку з цим, ШІ відкриває перед людством безліч можливостей для розвитку та покращення життя, але також ставить перед ним виклики та проблеми, які потребують ретельного вивчення та розв'язання. Для того, щоб забезпечити максимальну користь від розвитку ШІ та зменшити його негативний вплив, необхідно співпрацювати на міжнародному рівні та розробляти етичні та правові стандарти використання цієї технології [6].

Загалом, ШІ може відігравати важливу роль у розвитку суспільства та забезпеченні його сталого розвитку, але це залежить від того, як людство використовуватиме цю технологію та як воно вирішить пов'язані з цим проблеми. Тому, для того, щоб забезпечити максимальну користь від ШІ, необхідно продовжувати вивчати та розробляти цю технологію відповідно до етичних та правових стандартів, а також забезпечувати навчання та перекваліфікацію працівників.

Список використаних джерел:

1. Wikipedia Artificial Intelligence. URL: https://uk.wikipedia.org/wiki/штучний_інтелект
2. Artificial intelligence: where it is used today and when it will replace humansю. URL: https://www.radiosvoboda.org/a/shtuchnyi-intelekt-zagrozy-i-mozhlyvisti/31145992.html
3. Wikipedia Application of artificial intelligence. URL: https://uk.wikipedia.org/wiki/Застосування_штучного_інтелекту
4. And again about artificial intelligence. Help, threat or empty talk? URL: https://yur-gazeta.com/publications/practice/inshe/i-znovu-pro-shtuchniy-intelekt-dopomoga-zagroza-chi-pusti-balachki.html
5. Bill Marczak, John Scott-Railton, Noura Al-Jizawi, Siena Anstis, and Ron Deibert. "The Great iPwn: journalists hacked with suspected NSO Group iMessage 'Zero-Click' Exploit". URL: https://citizenlab.ca/2020/12/the-great-ipwn-journalists-hacked-with-suspected-nso-group-imessage-zero-click-exploit/
6. Citizen Lab. "The Great iPwn: Journalists Hacked with Suspected NSO Group iMessage 'Zero-Click' Exploit". https://citizenlab.ca/2020/12/the-great-ipwn-journalists-hacked-with-suspected-nso-group-imessage-zero-click-exploit/

Ключові слова: штучний інтелект, використання штучного інтелекту, впровадження штучного інтелекту, вплив штучного інтелекту

Keywords: artificial intelligence, adoption of artificial intelligence, promotion of artificial intelligence, the introduction of artificial intelligence

Науковий керівник: к.т.н., доцент Задерейко О. В.

ОГЛЯД ВИДІВ ШТУЧНОГО ІНТЕЛЕКТУ В КОМП'ЮТЕРНИХ ІГРАХ

Задерейко О. В.

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

Толокнов А. А.

асистент кафедри інформаційних технологій Національного університету «Одеська юридична академія»

Штучний інтелект (ШІ) використовується у комп'ютерних іграх для генерації реактивної, адаптивної або розумної поведінки переважно у неігрових персонажей (NPC), подібно до людської інтелектуальності. Також використовується для створення вражаючих інтерактивних досвідів, де гральні персонажі та неперсонажні сутності можуть виявляти інтелектуальну поведінку. ШІ включає в себе алгоритми, які моделюють розумові процеси, прийняття рішень та взаємодію з оточуючим світом у грі. ШІ в комп'ютерних іграх є окремим підполем та відрізняється від академічного ШІ. Ось деякі з них:

Поведінковий ШІ (Behavioral AI). Цей вид ШІ використовується для моделювання поведінки неперсонажних сутностей у грі, таких як ворожі моби, тварини або розумні предмети. Вони можуть мати задані шаблони поведінки або реагувати на зовнішні події та стимули [1].

Основні концепції та компоненти поведінкового ШІ включають: поведінковий елемент має свій стан (State), який описує його поточну ситуацію та контекст; перехід між станами (State Transition) на основі певних умов або подій; поведінкові правила (Behavior Rules); поведінкові дерева (Behavior Trees); рандомізація (Randomization). Приклади комп'ютерних ігор, в яких використовується поведінковий ШІ: "The Sims", "Assassin's Creed", "The Last of Us" тощо.

ШІ прийняття рішень (Decision-making AI). Цей тип ШІ використовується для розробки гральних персонажів, які можуть аналізувати ситуацію в грі та приймати розумні рішення. Використовуються алгоритми, такі як дерева рішень, машинне навчання або еволюційні алгоритми, щоб гральні персонажі могли планувати свої дії [2].

Основні аспекти ШІ прийняття рішень включають: аналіз інформації; цілі та обмеження; алгоритми прийняття рішень; динамічне змінення рішень.

Деякі приклади використання ШІ прийняття рішень включають: поведінка NPC; автоматичне керування гравцем; розумні адаптивні вороги.

Деякі приклади комп'ютерних ігор, в яких використовується ШІ прийняття рішень: "F.E.A.R."; "The Sims"; серія "Civilization"; "XCOM 2"; "Hitman".

Нейромережевий ШІ (Neural Network AI). Цей вид ШІ використовує нейромережі для моделювання інтелектуальної поведінки в грі. Нейромережевий ШІ може використовуватись для тренування гральних персонажів на основі великої кількості даних або для створення генеративних моделей для генерації контенту в грі [1].

Основні концепції та компоненти нейромережевого ШІ включають: штучні нейромережі, навчання, розпізнавання зображень, прийняття рішень, адаптація. Деякі приклади використання нейромережевого ШІ включають: визначення поведінки ворожих сутностей; моделювання інтелектуальної поведінки союзників або командних товаришів, що співпрацюють з гравцем; розпізнавання мови та комунікація з гравцем за допомогою природних мовних інтерфейсів; персоналізація гри шляхом адаптації поведінки NPC до стилю гри гравця. Ось декілька прикладів комп'ютерних ігор, в яких використовується штучний інтелект на основі нейронних мереж: "AlphaGo", "Dota 2", "Forza Motorsport" тощо.

ШІ для розпізнавання мови та взаємодії (Natural Language Processing and Interaction AI). Цей вид ШІ використовується для розуміння та взаємодії з мовою гравця. Гральні персонажі можуть розпізнавати інструкції гравця або взаємодіяти з ним у природній мові [3].

Основні компоненти ШІ для розпізнавання мови та взаємодії включають: розпізнавання та розуміння мови; генерація мовних відповідей; діалоговий менеджмент, який визначає, як система взаємодіє з гравцем під час діалогу; обробка мовних команд: наприклад, гравець може надати інструкцію персонажу про зміну напрямку руху або виконання певної дії; аналіз контексту: ШІ може аналізувати контекст взаємодії, включаючи попередні запити гравця та інформацію про світ гри, для кращого розуміння інтенцій та надання більш адекватних відповідей.

Деякі приклади комп'ютерних ігор, де використовується ШІ обробки природної мови та взаємодії: "The Elder Scrolls V: Skyrim", серія "Mass Effect", "Westworld", "Her Story" тощо.

ШІ для розпізнавання мови та взаємодії в комп'ютерних іграх дозволяє створювати більш іммерсивний та інтерактивний геймплей, де гравець може спілкуватися з грою за допомогою мови, а система може відповідати на ці запити та вирішувати завдання на основі розуміння мови та контексту.

Машинне навчання та ШІ (Machine Learning and AI). Цей підхід використовує методи машинного навчання, такі як навчання з підкріпленням або нейромережеві моделі, для створення інтелектуальних систем у грі. Гральні персонажі можуть навчатися з досвіду, самостійно вдосконалювати свої навички та

адаптуватися до геймплею гравця [4].

Основні компоненти машинного навчання та ШІ включають: навчання на основі даних; розпізнавання патернів; адаптивність; генерація вмінь та контенту; управління поведінкою.

Деякі приклади використання: "OpenAI Five", "DeepMind's AlphaGo", серія "FIFA", "No Man's Sky".

В цілому, застосування штучного інтелекту у комп'ютерних іграх розширює можливості геймінгу, забезпечує більш складний та захоплюючий досвід гри та сприяє розвитку нових інновацій у галузі геймдеву. Зазначені види ШІ не є взаємовиключними. Їх можна комбінувати в комп'ютерних іграх для отримання більш реалістичного і багатого ігрового досвіду.

Список використаних джерел

1. Yannakakis G. Artificial Intelligence and Games / G. Yannakakis, J. Togelius. Berlin: Springer, 2018. 359 с.
2. Millington I. Artificial intelligence for game / Ian Millington. San Francisco: Morgan Kaufmann, 2006. 895 с.
3. Rabin S. AI Game Programming Wisdom / Steve Rabin. Hingham, Massachusetts, USA: Charles River Media, 2002. 704 с.
4. Rabin S. Game AI Pro: Collected Wisdom of Game AI Professionals / Steve Rabin. – Boca Raton, Florida, USA: CRC Press, 2013. 626 с.

Ключові слова: ШІ, штучний інтелект, комп'ютерні ігри
Keywords: AI, artificial intelligence, computer games

ШТУЧНИЙ ІНТЕЛЕКТ В МЕДИЦИНІ

Цьоць А.О.

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

Останнім часом стрімко зростає розвиток інформаційних технологій у світі в різних сферах життя, в тому числі й в медицині. Це стає важливим фактором розвитку науки. Технології штучного інтелекту та машинного навчання з кожним роком все сильніше закріплюються у цій галузі.

Активна розробка математичних методів вирішення медико-біологічних завдань розпочалася у середині XX столітті. Були розроблені методи аналізу, загальною ознакою яких стала наявність явних алгоритмів прийняття рішень. Але перший штучний нейрон з використанням простих бінарних порогових функцій

був створений МакКаллохом і Піттсом ще у 1943 р. [1].

В наш час все більшу популярність у медицині набирає техніка ANN (штучна нейронна мережа), яка являє собою обчислювальні аналітичні інструменти, схожі з біологічною нервовою системою. Вона складається з мереж взаємозв'язаних комп'ютерних процесорів, здатних виконувати обчислення для обробки даних. Їхнє вміння вчитися на історичних прикладах, аналізувати нелінійні дані, обробляти неточну інформацію зробили їх дуже привабливим аналітичним інструментом у галузі медицини [2].

Можна виокремити напрями застосування штучного інтелекту в медицині:

- застосунки та програмні продукти для розпізнавання медичних зображень (знімків, кардіограм, результатів комп'ютерної томографії);
- стартапи для розробки медичних препаратів (мікроскопічний аналіз, вивчення ефективності препаратів, дослідження вірусів та пошук ефективних вакцин) [3];
- використання технологій машинного навчання для створення протезів (інтелектуальні системи розробляють зручні протези з урахуванням анатомічних особливостей людини);
- застосунки для віддаленої допомоги пацієнту (за їх допомогою лікарі можуть у віддаленому режимі дати рекомендації для лікування застудних хвороб);
- застосунки з лікування онкологічних захворювань (аналізують стан пацієнта та пропонують ефективну схему лікування) [4].

Очікується, що штучний інтелект стане найважливішим елементом при діагностиці та уточненні захворювань. Завдяки здатності співставляти дані, збирати інформацію, участь штучного інтелекту у діагностиці допоможе мінімізувати лікарські помилки та зменшити кількість захворювань шляхом технологічного спостереження за пацієнтом.

Штучний інтелект у смартфоні дозволить зробити доступною медицину майбутнього для всіх: у користувача буде персональний медичний помічник, здатний надавати повну картину фізичного стану здоров'я.

В хірургічних процедурах все частіше застосовуються роботи зі штучним інтелектом. Роботизована хірургія – це технологія, яка стрімко розповсюджується медичних процедурах в неврології, гінекології, ортопедії та хірургії. Роботизовані технології дозволяють практикуючим лікарям, проводити лікування на рівні, якого вони не змогли б досягти навіть у результаті багаторічної практики. Допомога робота під час операції усуває випадкові рухи.

Нейронні мережі застосовуються у сферу радіології, заощаджуючи час медичних організацій. Після того, як медичне зображення було зроблене за

допомогою МРТ, комп'ютерної томографії, ультразвукового або рентгенологічного дослідження, лікар повинен перевірити його на ознак захворювань. Після навчання з використанням великої кількості даних застосунки на основі штучного інтелекту здатні аналізувати медичні зображення та повідомляти про виявлені ознаки і відхилення від норми.

Сьогодні в деяких шпиталях вже працюють комп'ютерні помічники, від яких можна отримувати поради та підказки. Завдяки комп'ютерним помічникам можна не тільки отримати медичні поради, а й записатися до лікаря.

Активну участь у розвитку медицини беруть також великі IT кампанії. Наприклад кампанія IBM створила цифровий інструмент під назвою Watson Health який здатний розпізнавати онкологічні захворювання, проблеми з судинною системою. Він збирає та структурує дані про стан пацієнта, історію хвороби, заощаджуючи час лікарів і допомагаючи встановити точний діагноз. Інший їх сервіс – Arterys здатний створювати візуалізацію серця людини на основі проаналізованих даних. Така система була створена на основі нейромережі [5]. Національна служба охорони здоров'я Великобританії використовує розробку від Google – DeepMind Health. Збирая дані про симптоми і скарги через мобільний застосунок вона відображає певні ризики для здоров’я [6].

Як бачимо, завдяки інформаційним технологіям відбувся прорив в розвитку медицини. Щоправда, зараз невідомо, коли штучний інтелект зможе давати без помилок рекомендації лікарям про необхідний спосіб лікування. Всі ці технології ще знаходяться у стадії розроблення, проте вже зараз вони здатні полегшити роботу лікарям та врятувати життя пацієнтів.

Список використаних джерел:

1. Основи нейромережних технологій. Вибрані розділи: Класифікаційні властивості персептронів [Електронний ресурс] : навч. посіб. для студ. спеціальності 172 «Телекомунікації та радіотехніка», освітньої програми «Інформаційнообчислювальні засоби радіоелектронних систем»/ П.О. Яганов; КПІ ім. Ігоря Сікорського. – Електронні текстові данні (1 файл: 0,437 Мбайт). – Київ : КПІ ім. Ігоря Сікорського, 2021. – 44 с.
2. Artificial intelligence in medicine URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1964229/
3. Застосування штучного інтелекту у медиціні: ефективна діагностика і створення нових ліків. URL: https://aiconference.com.ua/uk/news/primenenie-iskusstvennogo-intellekta-v-meditsine-effektivnaya-diagnostika-i-sozdanie-novih-lekarstv-92604#
4. Штучний інтелект у медицині. URL: https://bit.ua/blog_columns/shtuchnyj-intelekt-v-medytsyni/4) ИИ в медицине: тренды и примеры применения. URL: https://www.iksmedia.ru/articles/5850863-II-v-medicine-trendy-i-primery-prim.html
5. IBM Watson Health is now Merative. URL: https://www.ibm.com/watson-health

6. Google is absorbing DeepMind's health care unit to create an 'AI assistant for nurses and doctors'. https://www.theverge.com/2018/11/13/18091774/google-deepmind-health-absorbing-streams-team-ai-assistant-nurse-doctor

Ключові слова: штучний інтелект, медицина, технології машинного навчання

Keywords: artificial intelligence, medicine, machine learning technologies

Науковий керівник: доцент Логінова Н. І.

ПРОЕКТ AI ACT: ПЕРША СПРОБА ПРАВОВОГО РЕГУЛЮВАННЯ ВИКОРИСТАННЯ ТЕХНОЛОГІЙ ШТУЧНОГО ІНТЕЛЕКТУ В ЄВРОПЕЙСЬКОМУ СОЮЗІ

Чанишев Р. І.

кандидат юридичних наук, доцент,
доцент кафедри інформаційних технологій,
Національний університет «Одеська юридична академія»,

11 травня 2023 року Комітет ЄС з питань внутрішнього ринку та Комітет ЄС з громадянських свобод 84 голосами проти 7 (12 утрималися від голосування) погодили зміни до проекту The Artificial Intelligence Act, документа, завданням якого є правове регулювання штучного інтелекту на території країн Європейського Союзу [1].

Проект акту був представлений ще у квітні 2021 року, і за минулий час до нього були внесені значні зміни. Необхідність цих змін була викликана негативною реакцією на нього з боку низки провідних компаній-розробників систем штучного інтелекту (далі – ШІ).

Наприклад, генеральний директор OpenAI Сем Альтман (а саме ця компанія створила найвідомішу на даний момент у світі програму ChatGPT) попередив, що OpenAI може відмовити в наданні своїх послуг в країнах ЄС, якщо діяльність таких компаній буде надмірно регульована європейськими законодавцями. Підставою для такого припущення стало рішення депутатів Європарламенту, які внесли близько 3000 поправок у документ, у тому числі і поправку, що включає в сферу дії акта розробників «фундаментальних моделей», до яких входить і OpenAI.

Таким чином, компанії-розробники повинні будуть вказувати, крім усього іншого, джерела інформації, які служили для навчання систем ШІ. А оскільки для навчання систем ШІ активно використовувалися ресурси інтернет-сервісу

WWW, то неминуче постане питання захисту авторських прав осіб, які розмістили ці матеріали на веб-сайтах, що з великою часткою ймовірності призведе до подання численних позовів з боку правовласників.

Крім того, очевидна небезпека розкриття інформації про використані джерела та методи, оскільки ця інформація може бути використана конкуруючими компаніями. Коли відбувався процес розробки і навчання систем ШИ, ніхто не приділяв цьому питанню особливої уваги. Зараз, коли використання систем ШИ стало комерційно вигідним і приносить прибуток, такий підхід (розкриття інформації про джерела) був визнаний помилковим. Так, в інтерв'ю виданню The Verge співзасновник і головний науковий співробітник OpenAI Ілля Суцкевер сказав, що причини OpenAI не ділитися додатковою інформацією про GPT-4 — страх конкуренції та побоювання за безпеку — «самоочевидні» [2].

Попередження, що було висловлено генеральним директором компанії OpenAI, викликало різку критику з боку комісара з внутрішнього ринку Європейського Союзу Тьєррі Бретона та інших законодавців, і вже 26.05.2023 Сем Альтман поспішив спростувати свої слова, заявивши про те, що OpenAI не планує залишати Європу [3].

Тим самим стало очевидним, що європейський підхід до вирішення питань регулювання розробки та експлуатації систем ШІ є цілком правильним.

Поправки до Закону про штучний інтелект показують, що депутати Європарламенту прагнуть забезпечити безпеку, прозорість і контрольованість систем ШИ, як існуючих сьогодні, так і тих, які можуть бути створені в майбутньому. Крім того, необхідно забезпечити екологічну безпеку, так як в майбутньому промислові підприємства та транспорт будуть оперативно управлятимуться за допомогою систем ШІ.

Необхідність правового регулювання технологій, яких поки що немає, обумовлена тим, що, на думку експертів, найближчим часом стала можливою поява так званого «сильного» ШИ, тобто штучного розуму, який не поступається людському, і, найімовірніше - його перевершує. Те, що ще недавно було долею письменників-фантастів, раптом почало набувати реальних рис.

Проект AI Act, в залежності від можливих негативних наслідків, визначає чотири рівні ризику для систем ШИ: мінімальний, обмежений, високий і неприйнятний.

Системи ШИ, класифіковані як неприйнятний ризик, повинні бути суворо заборонені.

До них в даний час відносяться системи, завдання яких полягає в маніпулюванні людьми. Методи маніпуляції можуть бути різними, наприклад, вплив на

підсвідомість людини або маніпуляція за допомогою спрямованої дезінформації. Наприклад, експерти вже висловлюють побоювання, що ШИ може бути використаний для здійснення фішингових атак, створення діпфейків та генерування неправдивої, але дуже схожою на правдиву новинну інформацію, що поширюється в мережі Інтернет. Така інформація може генеруватися за заданим зловмисниками алгоритмом у вигляді текстової, графічної та аудіовізуальної інформації [4].

Крім того, повинні бути заборонені системи, що служать для автоматизованого збору інформації, яку після обробки можна буде використовувати для класифікації людей на основі їхньої соціальної поведінки, соціально-економічного статусу, особистих характеристик.

Серед поправок, внесених до проекту, можна виділити наступні:

- повна заборона дистанційних систем біометричної ідентифікації в громадських місцях, за винятком систем, що використовуються з дозволу суду правоохоронними органами;
- заборона збору та аналізу біометричних даних, пов'язаних зі статевими, етнічними, расовими, релігійними та національними особливостями людини;
- заборона правоохоронним органам використовувати прогностичні системи ШІ (такі системи використовують дані про минулу поведінку людини і на їх основі прогнозують її подальші дії);
- заборона на створення баз даних, що дозволяють розпізнавати обличчя за допомогою камер спостереження або шляхом збору інформації з соціальних мереж.

До систем рівня високого ризику були віднесені системи, які можуть завдати шкоди здоров'ю людини, її правам, безпеці або навколишньому середовищу. Крім того, до рівня високого ризику були віднесені системи ШІ, які можуть бути використані в ході проведення політичних і виборчих кампаній.

Системи обмеженого ризику підлягають обов'язковій сертифікації для того, щоб бути дозволеними до використання. Ці системи повинні будуть пройти процедуру тестування уповноваженими державними органами. До них відносяться, наприклад, системи безпеки, що застосовуються в автомобілях.

До систем ШІ, що самостійно генерують інформацію, таким, як ChatGPT, пред'являються додаткові вимоги:

- обов'язкове позначення того, що цей контент був створений ШІ;
- виключення можливості створення контенту, що порушує закони;
- наявність вказівки на те, які саме матеріали, захищені авторським правом, були використані під час навчання ШІ.

Законопроектом пропонується створити незалежні органи, відповідальні за регулювання використання ШИ. Вони будуть проводити оцінки відповідності, видавати сертифікати та застосовувати штрафні санкції до порушників законодавства.

Крім того, депутати Європарламенту збираються розширити право громадян подавати скарги на системи ШІ та отримувати роз'яснення дій, вжитих системами ШІ.

Однак, суворе регулювання процесу розробки та впровадження систем ШІ може загальмувати інновації в області ШІ.

З цієї причини депутати дозволили робити винятки з наведених вище правил – але виключно для проведення дослідницьких робіт у цій галузі.

У червні 2023 року планується голосування Європарламенту щодо прийняття даного проекту та його передача на розгляд Європейської Ради.

В цілому слід зазначити, що AI Act має забезпечити етичність, прозорість та безпеку використання систем ШІ в Європейському Союзі, захищати права та інтереси людей, а також підтримувати інновації та розвиток перспективних систем ШИ в рамках норм і стандартів.

Однак слід пам'ятати, що, по суті, даний проект є першим подібним проектом в історії людства і неминуче має ряд недоліків, які без досвіду роботи з подібними системами виявити на даному етапі неможливо. Згаданий вище глава OpenAI Сем Альтман у травні 2023 року був викликаний до Сенату США для надання свідчень щодо необхідності регулювання систем штучного інтелекту, що розробляються як його компанією, так і іншими техногігантами. Під час слухань він заявив: «Я думаю, якщо щось піде не так з цією технологією, то все буде зовсім не так. І ми хочемо про це констатувати. Ми хочемо працювати з урядом, щоб цього не допустити» [5].

Список використаних джерел:

1. AI Act: a step closer to the first rules on Artificial Intelligence URL: https://www.europarl.europa.eu/news/en/press-room/20230505IPR84904/ai-act-a-step-closer-to-the-first-rules-on-artificial-intelligence
2. OpenAI co-founder on company's past approach to openly sharing research: 'We were wrong' URL: https://www.theverge.com/2023/3/15/23640180/openai-gpt-4-launch-closed-research-ilya-sutskever-interview
3. ChatGPT-maker OpenAI says has no plans to leave Europe URL: https://www.reuters.com/technology/openai-has-no-plans-leave-europe-ceo-2023-05-26/
4. Faith Karimi, CNN Mom, these bad men have me': She believes scammers cloned her daughter's voice in a fake kidnapping URL: https://edition.cnn.com/2023/04/29/us/ai-scam-calls-kidnapping-cec/index.html

5. OpenAI's Sam Altman Urges A.I. Regulation in Senate Hearing URL: https://www.nytimes.com/2023/05/16/technology/openai-altman-artificial-intelligence-regulation.html?searchResultPosition=1

Ключові слова: закон ЕС про штучний інтелект, розробка та впровадження штучного інтелекту, OpenAI, ChatGPT, рівні ризику для систем ШИ

Keywords: The Artificial Intelligence Act (EC), artificial intelligence, the development and implementation of artificial intelligence, OpenAI, ChatGPT, risk levels for AI systems

ВИКОРИСТАННЯ INTERACTIVE PYTHON ДЛЯ МОДЕЛЮВАННЯ, АНАЛІЗУ ТА ОБРОБКИ ДАНИХ У НАВЧАЛЬНОМУ ТА ДОСЛІДНИЦЬКОМУ ПРОЦЕСАХ

Бойко В. Д.

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

Інструменти моделювання та аналізу даних є потужним інтегрованим середовищем, яке може бути задіяне як у навчальному, так і в дослідницькому процесі. Зокрема, у підході, який тривалий час існував під неофіційним терміном "дослідницьке програмування" (exploratory programming, EP), або якщо бути точним "моделювання та дослідницький аналіз даних". Збільшення обчислювальних потужностей, розширенням обсягу та характеру інформації, що збирається ([1], [2], [3]) робить актуальними використання спеціалізованих методів роботи з великими даними, датамайнінгу та дослідницького моделювання.

Характерною стороною цього підходу є орієнтація не на кінцевий продукт, а на процес обробки даних. При цьому важливо не лише зафіксувати результат, а й кроки, які привели до нього. В рамках підходу EP робота ведеться з метою аналізу даних, побудови та перевірки гіпотез, математичного, когнітивного та комп'ютерного моделювання. При цьому з одного боку широко застосовуються вже наявні інструменти, бібліотеки та фреймворки (наприклад, набір бібліотек для швидких матричних розрахунків), з іншого результат заздалегідь невідомий і ведеться методом проб і помилок. Дослідник часто працює з даними не знаючи заздалегідь яким чином він може отримати відповіді, що його цікавлять.

Таке завдання висуває певні вимоги до інструментів її вирішення і можна простежити, як їхня еволюція призвела до того, що в континуумі IT-продуктів, інструментів та рішень почали виявлятися тренди схожі на процеси конвергенції в біології та техніці. Ці процеси призводять до того, що конкретні рішення при різниці сімейств, що їх породили, або ценозів мають подібні ознаки, принципи роботи та ідеології. Найкраще такі еволюційні тренди простежуються на open

source проектах, які за рахунок своєї відкритості та за висловом Еріка Реймонда "роботи багатьох очей" [4], є повільно і довго еволюціонуючими системами.

Однією з характерних для EP особливостей є використання REPL-оболонок [5]. Як використання REPL-Шелла вихідно виникло в мові lisp [6] і так чи інакше вплинуло на безліч мов програмування. При цьому, більшість мов високого рівня (Perl, Ruby, GHCi в Haskell і так далі) включають цей інструмент, а коли для цілей EP використовуються мови середнього рівня - цей інструмент для них "винаходиться", прикладом чого може бути сам lisp, а крім нього – спеціально створений під табличну обробку awk, створений Чарльзом Муром fort, який займає проміжне положення серед мов програмування, будучи одночасно і низькорівневою мовою та мовою, що допускає розширення його до високорівневого. Автор зустрічав приклади, коли при використанні в окремих проектах для дослідницького програмування мови Сі для імітації роботи оболонки задіявся дебаггер gdb - в результаті квазі-REPL, який цілком при цьому справлявся зі своїми функціями. Більшість наукових інструментів та обчислювальних середовищ: MATLAB, SciLab, Octave, CAS Maxima тощо містять REPL-консоль у тому чи іншому вигляді.

Прикладом REPL-шеллу, який еволюціонував від простого доробки оболонки традиційної мови програмування спочатку в зручний інструмент EP, потім породив власну екосистему (стек) інструментів і форматів даних, а потім в незалежну від мов програмування оболонку є interactive python, який часто скорочують до IPython або ipython [7].

ipython виник спочатку, як простий доробок консольного шелла традиційного python. У його витоків у 2001 році був Фернандо Перес, який надихався синтаксисом REPL-системи Mathematica з широким використанням нумерації консольного введення та зверненням до історії команд за допомогою % синтаксису.

До цього додався проект IPP Янко Хаузера, який дозволив поліпшити інтерфейс традиційного REPL, додавши в нього інструменти, що вже добре зарекомендували себе, без яких складно працювати в будь-якому командному рядку: автодоповнення, історію команд і пошук по ній через Ctrl+r. Цей проект також дозволив тісно інтегрувати роботу REPL python з оболонкою операційної системи (bash/zsh/...) через нотацію !. Тобто з python REPL можна було викликати shell-команди, не виходячи з сеансу роботи в python. Крім того, через нотацію можна було викликати вбудовану допомогу звичайного python (pydoc/help/dir), що сприяло поліпшенню інтроспекції мови.

Наступним розширенням традиційного python REPL став LazyPython Натана Грея. LazyPython орієнтувався на зменшення роботи з введення шляхом автоматичного розміщення дужок і лапок ("квотування") таким чином, введений текст:

```
>>> ,function argument1 argument2 argument3
```

цей модуль перетворював на:

```
>>> function("argument1", "argument2", "argument3")
```

Таким чином суттєво спрощувалося введення команд, крім того, модуль автоматично розширював багато консольних команд (cp, rm, ls) і мав розширюваність - до нього можна було написати багато своїх власних варіантів поліпшення введення. Крім того інструменти Грея використовувалися для підсвічування та розмальовки виводу, оскільки вихідний python REPL був монохромним.

Інтеграція всіх перерахованих проектів дозволила отримати REPL-оболонку, яка мала такі властивості:

- добре інтегрувалася з операційною системою;
- дозволяла зберігати та зручно використовувати історію команд REPL;
- забезпечувала зручні інструменти інтроспекції.

Особливо добре ipython (навіть у такому максимально простому вигляді) підходив для EP. Усі операції з роботі з даними (імпорт, впорядкування, перетворення, візуалізація, моделювання, експорт) можна проводити не виходячи з консолі, використовуючи всі доступні засоби оболонки та операційної системи.

Наприклад, можна змінити робочу директорію (у сучасному ipython частіше використовується %cd, а не !cd):

```
In [5]: %cd ~
/home/vik

In [6]: %cd ~/rep/jupyter/
/home/vik/rep/jupyter
```

Переглянути її вміст:

```
In [7]: !ls
audit  crypto  maxima  networkx  pandas
...
```

При цьому команда перегляду вмісту може використовуватися з різними ключами – як і традиційна команда консолі (власне, ця команда по суті є консольною командою).

```
In [10]: %cd pandas/data
/home/vik/rep/jupyter/pandas/data
...

In [11]: !ls -1
access-log
air_quality_long.csv
air_quality_no2.csv
air_quality_no2_long.csv
air_quality_parameters.csv
air_quality_pm25_long.csv
air_quality_stations.csv
ted.csv
```

Доступні інші команди консолі, що зручно інтегрує всю систему в єдине ціле. Наприклад, досліджувані файли зручно переглядати за допомогою команд head та tail. Розглянемо файл access-log, взятий із доповіді [8]:

```
[@In] [19]: !head -n 5 access-log

140.180.132.213 - - [24/Feb/2008:00:08:59 -0600] "GET
```

```
/ply/ply.html HTTP/1.1" 200 97238
140.180.132.213 - - [24/Feb/2008:00:08:59 -0600] "GET /favicon.ico
HTTP/1.1" 404 133
75.54.118.139 - - [24/Feb/2008:00:15:40 -0600] "GET / HTTP/1.1"
200 4447
75.54.118.139 - - [24/Feb/2008:00:15:41 -0600] "GET
/images/Davetubes.jpg HTTP/1.1" 200 60025
75.54.118.139 - - [24/Feb/2008:00:15:42 -0600] "GET /favicon.ico
HTTP/1.1" 404 133

...

In [20]: !tail -n 5 access-log

80.161.85.77 - - [29/Feb/2008:07:47:40 -0600] "GET
/ply/example.html HTTP/1.1" 200 2359
156.63.68.202 - - [29/Feb/2008:07:49:28 -0600] "GET /ply/
HTTP/1.1" 200 8018
156.63.68.202 - - [29/Feb/2008:07:49:28 -0600] "GET
/ply/bookplug.gif HTTP/1.1" 200 23903
156.63.68.202 - - [29/Feb/2008:07:49:28 -0600] "GET /favicon.ico
HTTP/1.1" 404 133
80.161.85.77 - - [29/Feb/2008:07:52:46 -0600] "GET /ply/ply.html
HTTP/1.1" 200 97238
```

При цьому інтеграція з операційною системою дозволяє не лише переглядати файли, а й, наприклад, використовувати конвеєри для фільтрації потрібних даних:

```
In [28]: !head -n 5 access-log | awk '($9 == 404) {print $0}'
140.180.132.213 - - [24/Feb/2008:00:08:59 -0600] "GET /favicon.ico
HTTP/1.1" 404 133
75.54.118.139 - - [24/Feb/2008:00:15:42 -0600] "GET /favicon.ico
HTTP/1.1" 404 133
```

При цьому результати роботи такої команди можуть бути практично безшовно інтегровані в роботу python без будь-яких труднощів. Наприклад, можна зберегти список заходів на сервер, що отримали http-код 404 (сторінка недоступна) в окрему змінну і вже засобами python отримати з нього ip-адреси користувачів, що “зайшли не туди”.

```
In [33]: !head -n 5 access-log | awk '($9 == 404) {print $0}'
140.180.132.213 - - [24/Feb/2008:00:08:59 -0600] "GET /favicon.ico
HTTP/1.1" 404 133
75.54.118.139 - - [24/Feb/2008:00:15:42 -0600] "GET /favicon.ico
HTTP/1.1" 404 133

In [34]: users_404 = !head -n 5 access-log | awk '($9 == 404)
{print $0}'

In [35]: users_404
Out[35]:
['140.180.132.213 - - [24/Feb/2008:00:08:59 -0600] "GET
/favicon.ico HTTP/1.1" 404 133',
```

```
 '75.54.118.139 - - [24/Feb/2008:00:15:42 -0600] "GET /favicon.ico
HTTP/1.1" 404 133']
```

В даному випадку консольний висновок перетворився на python list, який можна легко розкласти на складові python засобами:

```
In [36]: for user in users_404:
    ...:     print(user.split())
    ...:

['140.180.132.213', '-', '-', '[24/Feb/2008:00:08:59', '-0600]',
'"GET', '/favicon.ico', 'HTTP/1.1"', '404', '133']
['75.54.118.139', '-', '-', '[24/Feb/2008:00:15:42', '-0600]',
'"GET', '/favicon.ico', 'HTTP/1.1"', '404', '133']

In [37]: for user in users_404:
    ...:     print(user.split()[0])
    ...:
140.180.132.213
75.54.118.139
```

Вже цей приклад показує, як зручно використовувати оболонку для завдань дослідницького програмування.

При цьому ipython у встановленому вигляді займає приблизно 77 Кбайт, може запускатися в консолі, а значить легко автоматизується і може використовуватися, наприклад, для аналізу даних на віддаленому сервері або в умовах, коли використання важких інструментів утруднено. Все це зумовило його популярність і призвело до еволюції ipython від простого доопрацювання базового середовища python до набору інструментів, який зараз називають scipy stack і який є де-факто одним із стандартів обробки та дослідження даних у науковому світі.

Список використаних джерел:

1. Jagadish H. V., Gehrke J., Labrinidis A., Papakonstantinou Y., Patel J. M., Ramakrishnan R., Shahabi C. Big data and its technical challenges // Communications of the ACM. Association for Computing Machinery (ACM), 2014. Vol. 57, no. 7. P. 86–94.
2. McFarland D. A., McFarland H. R. Big data and the danger of being precisely inaccurate // Big Data and Society. SAGE Publications, 2015. Vol. 2, no. 2. P. 205395171560249.
3. Kaisler S., Armour F., Money W., Espinosa J. A. Big data issues and challenges // Encyclopedia of information science and technology, third edition. IGI Global, 2023. P. 363–370.
4. Raymond E. The cathedral and the bazaar // Knowledge, Technology and Policy. Springer Science; Business Media LLC, 1999. Vol. 12, no. 3. P. 23–49.
5. Schröder M., Cito J. An empirical investigation of command-line customization // Empirical Software Engineering. Springer Science; Business Media LLC, 2021. Vol. 27, no. 2.

6. Seibel P. Lather, rinse, repeat: A tour of the REPL // Practical common lisp. Apress, 2005. P. 9–18.
7. Pérez F., Granger B. E. IPython: A system for interactive scientific computing // Computing in Science and Engineering. IEEE Computer Society, 2007. Vol. 9, no. 3. P. 21–29.
8. Beazley D. Generators: The final frontier. — 2014.

Ключові слова: відкрите програмне забезпечення, відкриті формати даних, відкриті протоколи взаємодії, python, ipython, interactive python, explore programming

Keywords: open software, open data formats, open communication protocols, python, ipython, interactive python, explore programming

ВИКОРИСТАННЯ МОВИ ПРОГРАМУВАННЯ PYTHON ДЛЯ ВИРІШЕННЯ ЗАВДАНЬ РОЗПІЗНАВАННЯ ОБРАЗІВ

Логінова Н. І.

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

Розпізнавання образів – це напрямок кібернетики та інформатики, який займається класифікацією та ідентифікацією предметів, процесів, явищ тощо. Розпізнати об'єкт означає класифікувати його та віднести до певного набору можливих класів, що ґрунтуються на аналізі значень ознак об'єкта.

Методи розпізнавання образів використовуються в різних галузях людини. Це й розпізнавання текстів та мовних команд, аналіз відбитків пальців, зображення обличь з відеокамер зовнішнього спостереження, медичних зображень для діагностування різних захворювань, у військових справах для обробки сигналів і радіолокаційних даних для прийняття рішень щодо ідентифікації повітряних цілій тощо.

Для розпізнавання актуальні питання цифрової обробки зображень, як-то: поліпшення якості зображення, визначення параметрів зображення, спектральний аналіз багатомірних сигналів, розпізнавання зображень та ін. Вагомою складової вирішення таких завдань є системи комп'ютерного зору, які за допомогою спеціальних алгоритмів та програм аналізують потік вхідних даних, виокремлюють об'єкти, ідентифікують їх та класифікують для подальшого розпізнавання. Комп'ютерний зір – це автоматичне отримання інформації із зображення. В якості інформації можуть бути 3D-моделі, зображення з відеокамер та будь-які графічні зображення. З практичної точки зору, комп'ютерний зір є поєднанням

програмування, моделювання та математичних методів.

Реальні зображення разом з корисною інформацією містять різні перешкоди: шуми фотоприймальних пристроїв та каналів зв'язку, геометричні та радіометричні спотворення тощо [1]. Тому не завжди отримуються зображення потрібної якості і виникає потреба в його обробці.

Загальні завдання обробки зображень – це відображення зображення та виконання різних операції з ним: кадрування, віддзеркалення, обертання, сегментація, класифікація, вилучення ознак, відновлення та розпізнавання.

Для вирішення таких завдань зручно застосовувати мову програмування Python, яка має багато різних бібліотек для обробки зображень.

Графіка у Python дозволяє малювати різні зображення, створювати анімацію та візуалізувати математичні обчислення. Бібліотеки зображень обробляють графіку та забезпечують підтримку форматів графічних файлів. Найбільш поширеними Python-бібліотеками є: Pillow, NumPy, Matplotlib, OpenCV та ін.

Pillow – безкоштовна бібліотека, яка дозволяє обробляти різні графічні зображення. Призначена для опрацювання даних, які зберігаються у піксельних форматах.

Для відкриття зображення функція `open()`, в яку передається шлях до файлу:

```
from PIL import Image

img = Image.open("cat.jpg")
```

Клас `Image` надає ряд атрибутів, які зберігають інформацію про зображення: ім'я та формат файлу, режим зображення (наприклад, "RGB" або "CMYK"), розмір зображення та ін.

Для збереження зображення застосовується метод `save()`.

```
img.save("cat_new.JPG")
```

Для обертання зображення застосовується метод `rotate()`, який в обов'язковий параметр передає кут повороту.

```
img2 = img.rotate(45)
img2.save("cat_new.jpg")
```

Метод `crop()` дозволяє вирізати частину зображення. В якості параметра він використовує кортеж з 4 чисел: координата_X_верхнього_лівого _кута, координата_Y_верхнього_лівого _кута, координата_X_правого_нижнього _кута, координата_Y_правого_нижнього _кута.

```
from PIL import Image
img = Image.open("cat.jpg")
img = img.crop((0, 0, img.width, img.height/2))
img.save("cat_new.jpg")
```

Для зміни розміру зображення застосовується метод `resize()`, який в якості параметру передає кортеж з двох чисел – нової висоти і ширини.

```
from PIL import Image
img = Image.open("cat.jpg")
img=img.resize((img.width//2, img.height//3))
img.save("cat_new_2.jpg")
```

Бібліотека має багато різних фільтрів для обробки зображень, які є частиною модуля `ImageFilter`.

OpenCV – бібліотека комп'ютерного зору з відкритим кодом, яка містить понад 2500 оптимізованих алгоритмів, які можна використовувати для виявлення та розпізнавання обличь, ідентифікації об'єктів, відстеження рухомих об'єктів на камерах, вилучення 3D-моделей об'єктів тощо [2].

Для роботи із зображеннями використовуються дві функції: `cv2.imread()` і `cv2.imshow()`.

Функція `cv2.imread()` зчитує в пам'ять комп'ютера масив даних із файлу із зображенням. Аргументом даної функції є ім'я файлу або шлях до файлу.

Друга функція `cv2.imshow()` використовується для відображення даних графічного файлу на екрані. Ця функція має два аргументи: ім'я вікна, в якому буде виведено зображення, та ім'я змінної, пов'язане з графічними даними, що відображаються.

Змінення розміру зображення OpenCV виконує за допомогою функції `cv2.resize()`, яка має три аргументи та результатом повертає трансформоване зображення: ім'я змінної, пов'язаної з вихідним зображенням; кортеж із двох елементів (послідовність) з новими розмірами зображення; алгоритм, який застосовується для перетворення зображення.

Наприклад:

```
f= float(wide)/image.shape[1]
new_size=(wide, int(image.shape[0]*f))
res=cv2.resize(image, new_size, interpolation=cv2.INTER_AREA)
```

Змінюючи параметр `wide` можна зменшувати або збільшувати розмір кінцевого зображення. При формуванні кортежу з розмірами зображення використовується команда `image.shape[0]`. Команда `image.shape[i]` повертає

ширину (при `i = 1`) або висоту (при `i = 0`) вихідного зображення.

Для розпізнавання образів, як правило, аналізують частину зображення, тому непотрібні частини зображення обрізають за допомогою зрізів. Зріз – це деякий об'єкт, який є набором індексів (`start, stop, step`), і метод, призначений для отримання певної частини послідовності. Зрізи часто використовують для представлення тієї чи іншої послідовності елементів деякого списку. Послідовність будується з елементів, починаючи з індексу `start` до індексу `stop` з кроком `step`.

Для створення зрізу застосовують розширений синтаксис індексування, використовуючи квадратні дужки «[]» і двокрапку «:» в якості роздільника. Наприклад, для одномірного списку `list [start: stop: step]`.

Наприклад:

```
crop=image[20:250, 200:390]
cv2.imshow("Crop image", crop)
cv2.imshow("Original image", image)
cv2.waitKey(0)
```

У результаті на екрані буде відображено два вікна: вихідне зображення та фрагмент вихідного зображення (рис. 1).

Рисунок 1 – вихідне зображення та зріз зображення

За допомогою бібліотеки OpenCV в Python можна працювати з різними кольоровими моделями, виділяти контури об'єктів, шукати об'єкти за формою, обробляти відеопотоки та ін., тобто виконувати різні завдання щодо розпізнавання образів.

Список використаних джерел:

1. Кобилін О.А., Творошенко І.С. Методи цифрової обробки зображень: навч. посібник. Харків: ХНУРЕ, 2021. 124 с.
2. About OpenCV. URL: https://opencv.org/

Ключові слова: розпізнавання образів, обробка зображень, Python, Pillow, OpenCv
Keywords: pattern recognition, image processing, Python, Pillow, OpenCv

ОСНОВНІ СЦЕНАРІЇ ВИКОРИСТАННЯ ФОРМАТУ ДАНИХ JSON У БАЗАХ ДАНИХ

Лобода Ю. Г.

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

JSON (JavaScript Object Notation) – формат обміну даними, котрий створений на мови програмування JavaScript стандарту EGMA-262 3rd Edition – December 1999 [1]. З 2017 року JSON стандартизовано як міжнародний стандарт ECMA-404 [2] та ISO/IEC 21778:2017 [3]. JSON – це популярний формат текстових даних, який застосовується для обміну даними у сучасних вебдодатках та мобільних додатках. Крім того, JSON використовується для зберігання неструктурованих даних у файлах журналів або баз даних NoSQL, наприклад, таких як Microsoft Azure Cosmos DB, MongoDB, CouchDB, Redis. Багато вебслужб повертають результати у форматі JSON або приймають дані у форматі JSON. Імена файлів JSON мають розширення .json.

JSON не залежить від мови реалізації, але використовує синтаксис С-подібних мов, таких як C, C++, C#, Java, JavaScript, Python та багатьох інших. Ці властивості роблять JSON ідеальною мовою для обміну даними. JSON є основним форматом обміну даними між вебсторінками та вебсерверами за допомогою AJAX [4].

Функції JSON, вперше представлені в SQL Server 2016, дозволяють поєднувати поняття NoSQL та реляційних баз даних в однієї базі даних. Тобто можливо поєднувати в одній таблиці класичні реляційні стовпці зі стовпцями, які містять документи у форматі тексту JSON, аналізувати та імпортувати документи JSON у

реляційні структури або форматувати реляційні дані у текст JSON.

Документи JSON можуть містити вкладені елементи та ієрархічні дані, які неможливо порівняти зі стандартними реляційними стовпцями. У цьому випадку можна виконати зведення ієрархії JSON за допомогою з'єднання батьківських елементів з вкладеними масивами.

JSON базується на двох загальних структурах даних, які підтримують сучасні мови програмування:

- колекція пар ключ/значення, реалізована як об’єкт, запис, структура, словник, хеш, іменований список або асоціативний масив;
- упорядкований список значень, реалізовано як масив, вектор, список або послідовність [4].

На деталізованому рівні JSON складається із типів даних: рядок, число, логічне вираження, нуль, об'єкт, масив.

Рядок у JSON складається із символів Unicode з екрануванням зворотною косою межею (\). Наприклад,

```
{ "surname" : "Gor" }
```

Число JSON відповідає формату JavaScript подвійної точності. Наприклад,

```
{ "n_1" : 210, "n_2" : 1021.08, "n_3" : 45.12}
```

Логічні значення виражаються як TRUE чи FALSE. Логічні значення не полягають у лапки та розглядаються як рядкові значення. Наприклад,

```
{ "acknowledged" : false }
```

Нуль – це пусте значення. Якщо ключу не надається значення, він може розглядатися як нуль. Наприклад,

```
{ "Instructions" : null }
```

Об'єкт – тип даних об'єкта JSON є набором пар імен або значень, укладених між фігурними дужками {}. Ключі мають бути унікальними рядками, розділеними комою. Наприклад,

```
{"Client" : { "name": "Alan Kim", "age": "38", "city" :
"Berlin", "car_model": "Chevrolet Corvette", "rating": 4.6 }}
```

Тип даних масив – це впорядкована колекція значень. У JSON значення масиву мають бути такі типи: рядок, число, об'єкт, масив, логічний вираз або нуль. Наприклад,

```
{ "Client" : [ { "name": "Jim Black", "age": 35, "car_model":
"BMW 5 Series"}, { "name": "Micha Tang", "age": 27, "car_model":
"Audi A6 }]}
```

Основні сценарії використання формату JSON.

1. Створення об'єкта JSON із даних, згенерованих користувачами.

JSON підходить для зберігання тимчасових даних. Тимчасовими даними, наприклад, можуть бути такі створені користувачем дані, як заповнена форма на вебсайті.

2. Використання JSON як формату даних для будь-якої мови програмування для забезпечення високого рівня взаємозамінності.

3. Перенесення даних між системами.

Перевірка через API вірогідності поштової адреси клієнта розташованої у базі даних вебсайту. Для цього необхідно надіслати адресні дані у форматі JSON в API служби перевірки адрес.

4. Конфігурування даних для додатків.

При розробці програм кожній програмі необхідні облікові дані для підключення до бази даних, а також шлях до файлу журналу. Ці облікові дані та шлях до файлу можуть бути вказані у файлі JSON, звідки можна вважати їх і отримати доступ до них.

5. Спрощення складних моделей даних.

JSON спрощує складні документи, доводячи їх до компонентів, які були визначені як значущі, шляхом перетворення процесу вилучення даних у передбачуваний та читаний людиною файл JSON.

Найпопулярніші реляційні бази даних мають можливість зберігання даних у форматі JSON. MySQL представила підтримку JSON, починаючи з версії 5.7, яка стала загальнодоступною у 2015 році. PostgreSQL підтримує тип даних JSON, починаючи з версії 9.2, випущеної у 2012 році. SQL Server підтримує JSON, починаючи з SQL Server 2016. Технології реляційних баз даних підтримують лише тип даних JSON. Це означає, що можливо додати один або кілька стовпців JSON у реляційні таблиці, а це не порушує реляційний підхід і є просто можливістю, корисною і потужною функцією.

JSON – це гнучкий формат передачі даних між додатками, аналогічний CSV-файлу. Однак замість рядків і стовпців об'єкти JSON є набором пар ключ/значення. За даними Stack Overflow, зараз JSON є найпопулярнішим форматом обміну даними, випереджаючи csv, yaml та xml [5]. Успіх JSON у його зручності читання як розробниками, так і комп’ютерами. Формат JSON простий для розуміння, але при цьому досить гнучкий для роботи як із примітивними, так і зі складними типами даних.

Список використаних джерел:

1. EGMA-262 3rd Edition – December 1999. ECMAScript Language Specification. URL: https://www.ecma-international.org/wp-content/uploads/ECMA-262_3rd_edition_december_1999.pdf (дата звернення 10.05.2023)
2. ECMA-404. The JSON data interchange syntax. 2nd edition, December 2017. URL: https://www.ecma-international.org/publications-and-standards/standards/ecma-404/ (дата звернення 10.05.2023)
3. ISO/IEC 21778:2017. Information technology — The JSON data interchange syntax.URL: https://www.iso.org/standard/71616.html (дата звернення 10.05.2023)
4. Introducing JSON. URL: https://www.json.org/json-en.html (дата звернення 15.05.2023)
5. Stack Overflow Trends. Json. Xml. Protocol-buffers. Yaml. Csv. URL: https://insights.stackoverflow.com/trends?tags=json%2Cxml%2Cprotocol-buffers%2Cyaml%2Ccsv&ref=arctype.com (дата звернення 15.05.2023)

Ключові слова: бази даних, JSON, NoSQL, реляційні бази даних, формат даних

Keywords: databases, JSON, NoSQL, relational databases, data type

РОЗРОБКА TELEGRAM-БОТА МОВОЮ PYTHON ДЛЯ ОЗНАЙОМЛЕННЯ ЗІ ЗМІСТОМ СИЛАБУСІВ ВИБІРКОВИХ ПРОФЕСІЙНИХ ДИСЦИПЛІН НА ФАКУЛЬТЕТІ

Шляхтицький Д. С.

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

Поширене використання Telegram-месенджера та Telegram-ботів серед студентів закладів вищої освіти (ЗВО) зумовлені їх зручністю, адже здобувачам набагато зручніше отримувати та опрацьовувати інформацію шляхом повідомлень від офіційних Telegram-ботів ЗВО, ніж вишукуючи цю інформацію окремо на сайті вишу. Це стосується, наприклад, ситуації багатокритеріального вибору численних вибіркових дисциплін здобувачами та ознайомлення зі змістом силабусів відповідних дисциплін. Замість пошуку силабусів з тої чи іншої вибіркової дисципліни, набагато легше було б відкрити відповідний чат-бот у Telegram та знайти необхідний документ, скориставшись критеріями відбору.

Складність розробки такого роду проєктів полягає в потребі завантаження численних документів силабусів. Використання для цього PDF-формат є недоцільним, оскільки такого типу файли будуть неявно завантажуватись на пристрій, тим самим засмічувати акаунт, пам’ять пристрою та споживати значні обсяги трафіку. Використання формату Telegraph дозволяє позбавитись потреби завантаження файлів і робить перегляд та змістовний пошук по файлу зручним.

Для розробки чат-бота для пошуку та ознайомлення зі змістом силабусів численних вибіркових дисциплін використовувалась мова програмування Python в інтегрованому середовищі розробки PyCharm Community [1-2]. Для підключення численних файлів силабусів у форматі Telegraph використовувалась базова бібліотека aiogram, яка є адаптованою для складноструктурованих Telegram-ботів і має великий набір функцій, які допоможуть зробити бот більш продуктивним та ефективним. Так, в aiogram є функції для:

- отримання та передавання повідомлень та файлів, включаючи зображення, відео, аудіофайли та документи
- робота з клавіатурами, які дозволяють користувачам взаємодіяти з ботом шляхом натискання кнопок;
- робота з командами, які дозволяють користувачам запускати певні дії за допомогою спеціально створених команд;
- робота з inline-запитами, які дозволяють користувачам швидко знаходити та надсилати інформацію безпосередньо з чату [3].

Крім того, aiogram має детальну документацію з прикладами, що допомагають новачкам швидко розібратися з функціоналом бібліотеки.

Розробленим чат-ботом для пошуку та ознайомлення зі змістом силабусів численних вибіркових дисциплін можна однаково ефективно користуватися в ОС Android, у Desktop-форматі Telegram застосунку тощо (рис. 1-2).

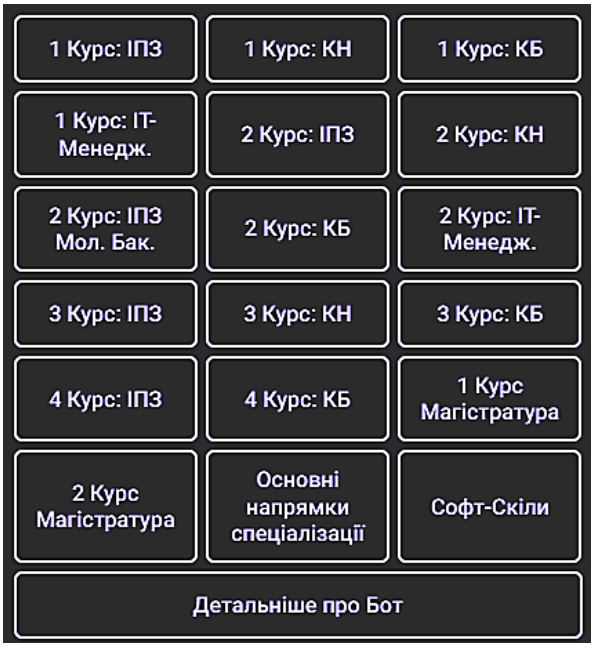


Рисунок 1 – Набір кнопок для вибору курсу, спеціальності та додаткової інформації на віртуальній клавіатурі ReplyKeyboard

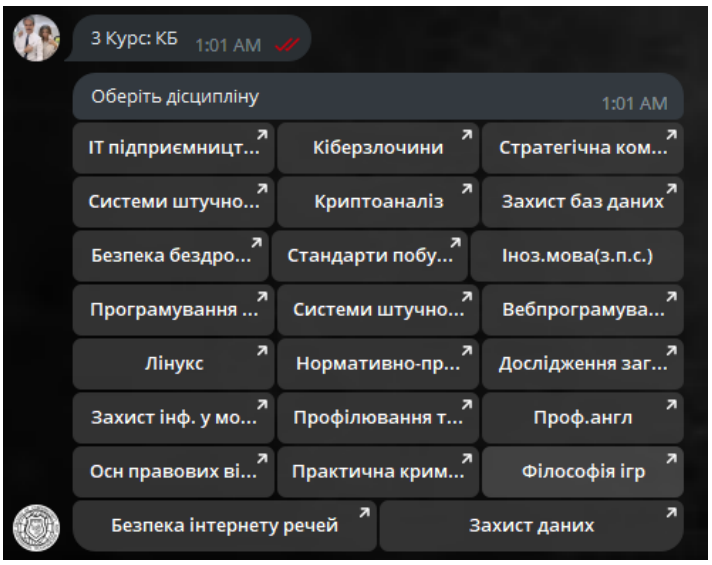


Рисунок 2 – Скріншот вибору дисциплін 3 курсу КБ у Desktop-форматі

Функціонал Telegram-бота подано у вигляді інтелектуальної мапи (рис. 3):

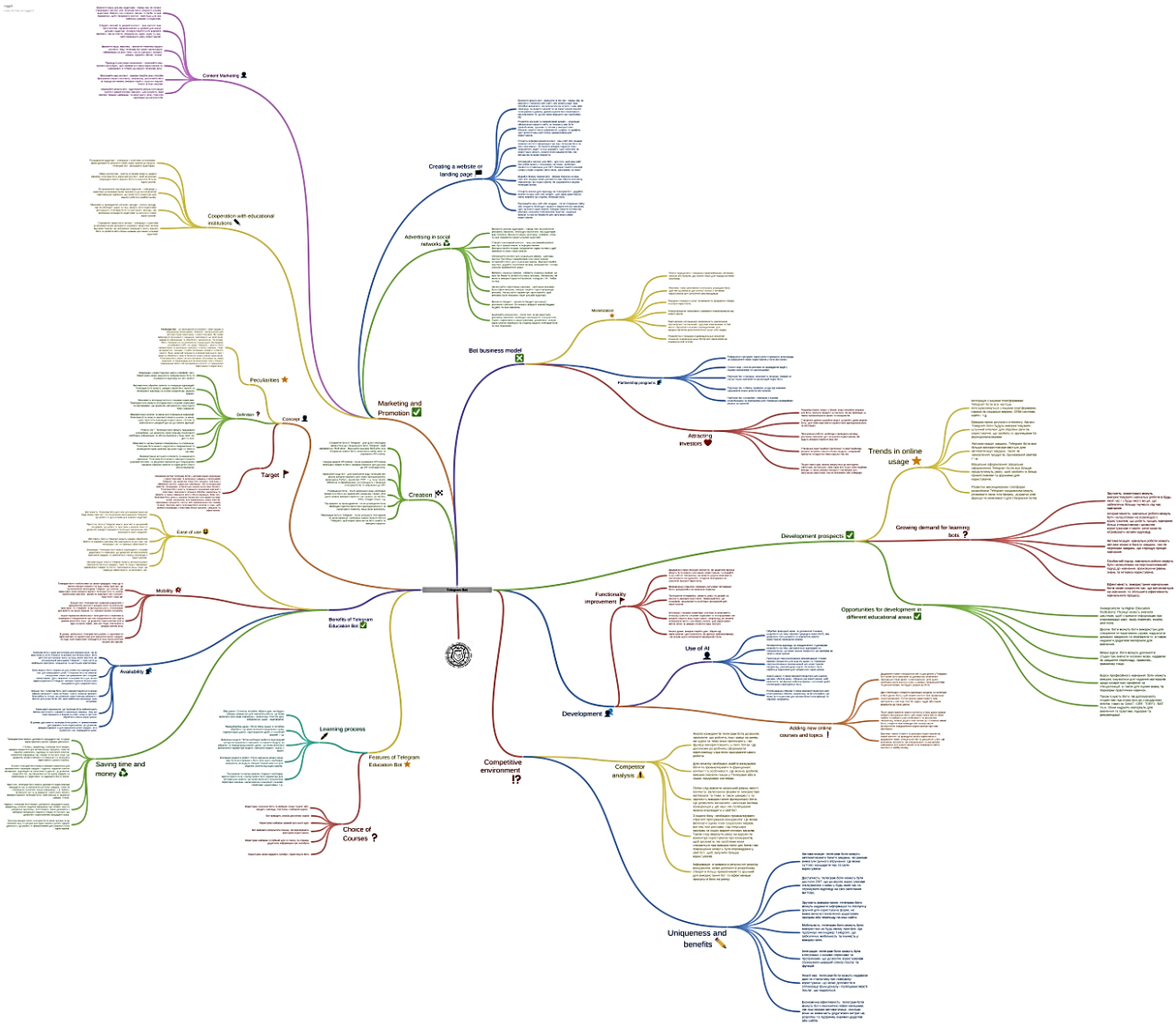


Рисунок 3 – Mind map розробленого чат-бота

Основні функції розробленого Telegram-бота:

- відображення усіх силабусів дисциплін відповідно до курсу: бот надає користувачеві перелік курсів, щоб вибрати той курс, на якому знаходиться силабус, що задовольняє його потребам;
- вибір силабусу: користувач може вибрати потрібний силабус за допомогою бота, натиснувши Inline-кнопку та перейшовши за посиланням;
- надання інформації про силабус: бот може надати користувачеві докладну інформацію про вибрану дисципліну, її зміст, теми, компетентності, програмні результати навчання тощо.

У роботі Telegram-бота використовуються файли формату Telegraph, які замість завантаження на пристрій і подальшого засмічування пам'яті та трафіку використовують гіперпосилання на відповідні Telegraph-файли.

Національний університет
"Одеська юридична академія"

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

«Фреймворки для Backend розробки»

Освітня програма	Рівень вищої освіти	Форма навчання	Курс вивчення	Семестр вивчення	Кількість кредитів
Інженерія програмного забезпечення	Перший рівень (бакалаврський)	Денна	3	5	4
		Заочна	3	6	4

Рисунок 3 – Приклад Telegraph-файлу із силабусом при натисканні на Inline-кнопку з посиланням на відповідну дисципліну

Розроблений чат-бот планується запровадити у навчальний процес на факультеті кібербезпеки та інформаційних технологій Національного університету «Одеська юридична академія».

Список використаних джерел:

7. Chai docs. URL: https://www.chaijs.com/guide/
8. JetBrains Inc. PyCharm. URL: https://www.jetbrains.com/ru-ru/pycharm/
9. aiogram dev team Welcome to aiogram. URL: https://aiogram.dev/

Ключові слова: чат-бот, Telegram-бот, PyCharm, aiogram, силабус, вибіркова дисципліна, Telegraph-файл

Keywords: chat bot, Telegram bot, PyCharm, aiogram, syllabus, selective discipline, Telegraph file.

Науковий керівник: доцент Трофименко О. Г.

ІНТЕРАКТИВНА ВІЗУАЛІЗАЦІЯ ГРАФІВ ЗАСОБАМИ БІБЛІОТЕКИ CYTOSCAPE.JS

Буділенко О. С.

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

Створення інтерактивних візуалізацій даних є важливим етапом в аналізі й вивченні складних зв'язків між об'єктами. Застосування бібліотеки cytoscape.js дозволяє створювати графічні подання даних та зв'язків, що дає можливість швидко й ефективно аналізувати дані та виявляти залежності між різними об'єктами [1].

Інтерактивна візуалізація даних із використанням бібліотеки cytoscape.js може бути застосована в різних галузях. Наприклад, у біоінформатиці вона використовується для відображення залежностей між генами та біологічними молекулами. У соціальних науках ця технологія дозволяє відображати зв'язки між людьми, організаціями та різними соціальними групами. Іншим прикладом використання бібліотеки cytoscape.js є візуалізація залежностей між різними показниками економічного розвитку країни. За допомогою графічного подання можна швидко й ефективно проаналізувати зв'язки між такими показниками, як ВВП, рівень безробіття, інвестиції, експорт та імпорт. Крім того, цю технологію можна використовувати для візуалізації залежностей між компонентами програмного забезпечення. Такий графічний підхід дозволяє легко визначати залежності між компонентами та шукати можливості для оптимізації коду.

Інтерактивна візуалізація даних за допомогою бібліотеки cytoscape.js має численні переваги. Вона допомагає аналізувати та відстежувати зміни у зв'язках між об'єктами, виявляти проблемні ділянки та шукати способи їх вирішення. При цьому, візуалізація даних здійснюється в зручному графічному вигляді, що дозволяє легко розуміти та сприймати інформацію [2]. Поширеним прикладом використання цієї технології є візуалізація мережі доріг та автомобільних маршрутів у місті. Це дає можливість швидко відстежувати залежності та вибудовувати маршрути між різними вулицями, досліджувати проблемні з точки зору пересування місця та шукати способи покращення транспортної розв’язки.

Саме засобами бібліотеки cytoscape.js та мови JavaScript в середовищі Visual Studio Code було розроблено програмний вебзастосунок для візуалізації графів та роботи базових алгоритмів роботи з ними. Функціонал цього застосунку дозволяє конструювати графи з додаванням та вилученням вузлів. Створення ребра між двома вершинами відбувається шляхом їх виділення і натискання на

відповідну кнопку, при цьому можна задати вагу ребра у разі зваженого графа. Також є можливість задавати чи то змінювати напрямок ребра для орграфа (рис. 1).

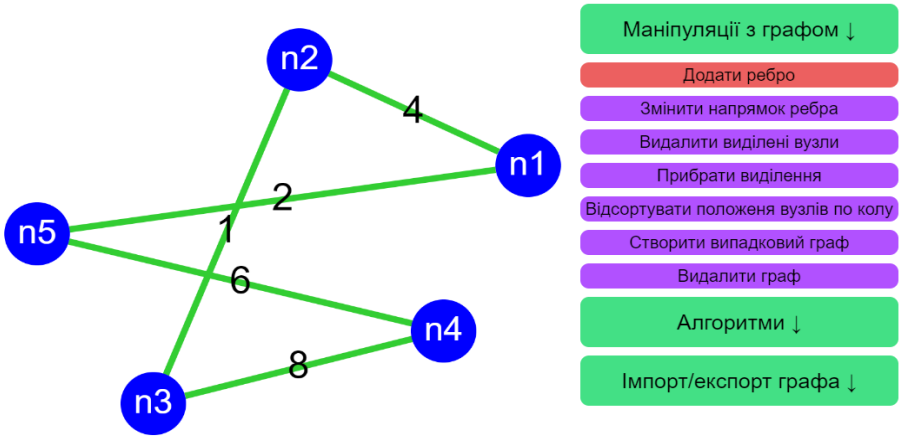
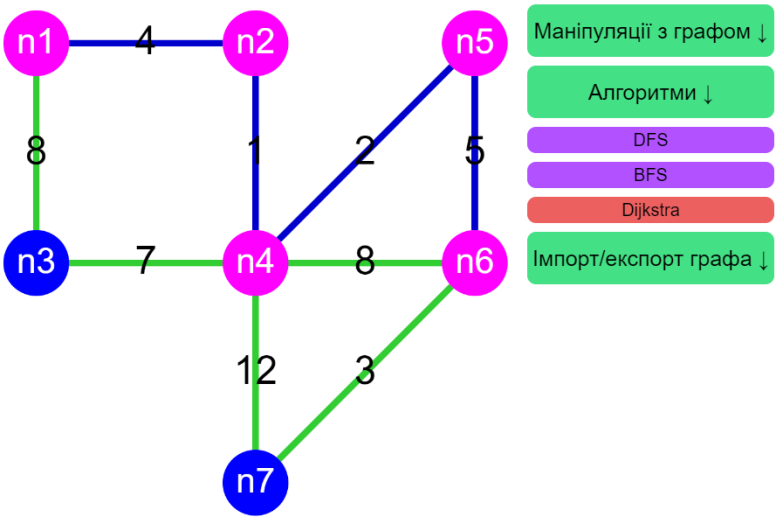


Рисунок 1 – Приклад створення зваженого орграфа

У розробленому застосунку є опція швидкого створення випадкового графа із заданого діапазону кількості вершин, що може бути зручним, особливо при вивченні алгоритмів роботи з графами. Є опція розміщення вузлів наявного графа по колу.

Розроблений застосунок візуалізує роботу трьох базових алгоритмів роботи з графами: BFS, DFS і Дейкстри [3, 4, 5]. Всі ці алгоритми мають наочну анімацію.

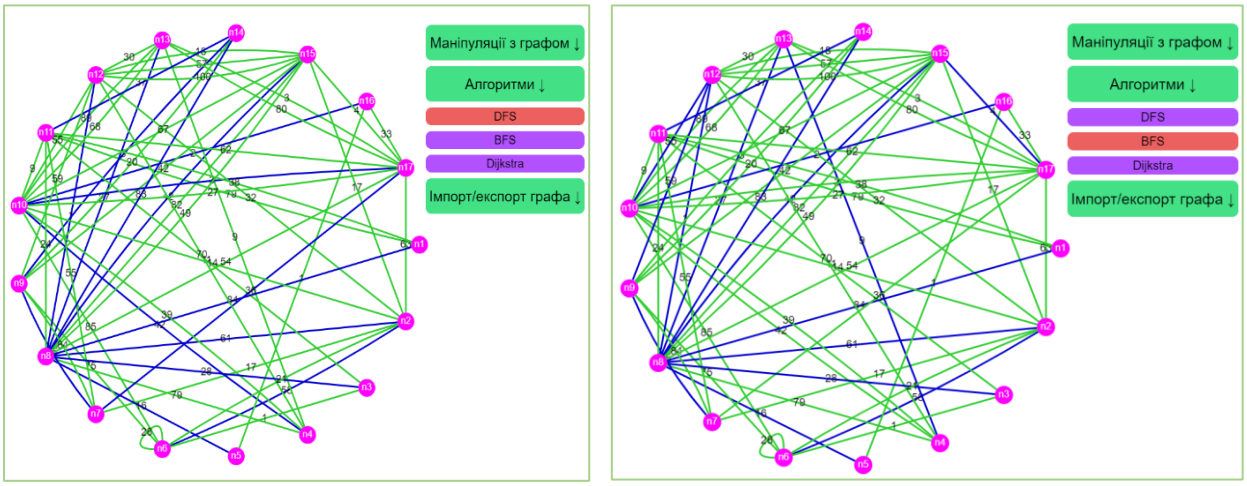
Результат візуалізації алгоритму Дейкстри для пошуку найкоротшого шляху між двома заданими вершинами n1 та n6 у графі показано на рис. 2.




```
Найкоротший шлях: n1 -> n2 -> n4 -> n5 -> n6
Відстань: 12
```

Рисунок 2 – Візуалізація алгоритму Дейкстри

Для обходів DFS та BFS користувач задає початкову вершину, а далі спостерігає роботу відповідного вибраного алгоритму (рис. 3).



а **б**

Рисунок 3 – Вигляд анімованої візуалізації алгоритмів на графах:

а) DFS; б) BFS

У застосунку є можливість зберігати та завантажувати графи за принципом парса структури графа засобами методів бібліотеки cytoscape.js. Для збереження графа у файлі використовується формат JSON (JavaScript Object Notation). За допомогою методу cy.json() структура графа може бути перетворена на JSON-рядок, який містить всю необхідну інформацію про граф. При завантаженні графа використовується метод JSON.parse(), який дозволяє розпарсити JSON-рядок та отримати його структуру.

Розроблений застосунок може бути корисним для дослідників, що працюють з даними, які мають багатовимірні зв'язки, наприклад: біологічні мережі, соціальні мережі, транспортні мережі тощо. Соціальні мережі, як-от: Facebook, Twitter, LinkedIn, мають складну структуру взаємодій між користувачами. Візуалізація таких мереж допомагає аналізувати соціальні зв'язки, виявляти групи, спільноти та впливових користувачів

Список використаних джерел:

1. Cytoscape.js. Graph theory (network) library for visualisation and analysis. URL: https://js.cytoscape.org.
2. Franz M., Lopes Ch., Fong D., Kucera M.,Cheung M., Siper M., Huck G., Dong Y., Sumer S., Bader G. Cytoscape.js 2023 update: a graph theory library for visualisation and analysis. Bioinformatics. 2023. Vol. 39. P. 1-3. DOI: 10.1093/bioinformatics/btad031.
3. Perform a depth-first search within the elements in the collection. URL: https://js.cytoscape.org/#eles.depthFirstSearch
4. Perform a breadth-first search within the elements in the collection. URL: https://js.cytoscape.org/demos/animated-bfs/

5. Dijkstra's algorithm on the elements in the collection. URL: https://js.cytoscape.org/#eles.dijkstra

Ключові слова: візуалізація, алгоритми, графи
Keywords: visualization, algorithms, graphs

Науковий керівник: доцент Трофименко О. Г.

ПРОГРАМНИЙ ПІДХІД ОНОВЛЕННЯ ПРОГРАМНИХ ЗАСТОСУНКІВ

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

Процес інсталяції програмного забезпечення (ПЗ) на кількох комп'ютерах у мережі з централізованого розташування відомий як розгортання ПЗ. При цьому відбувається встановлення, видалення та оновлення відповідного ПЗ [1]. Це стосується й оновлення актуальних версій застосунків на різних апаратних платформах та операційних системах. При оновленні програмних застосунків відбувається завантаження численних файлів та папок з відповідного онлайн-сервісу, який відповідає за цей процес. Оновлення може відбуватися в автоматичному режимі або після відповідного підтвердження користувача. Заради безпеки, дані та способи їх оновлення зазвичай не розміщують у відкритому доступі.

Більшість наявних інструментів (Total Software Deployment (TSD), Maestro AutoInstaller), які забезпечують схожу функціональність фонового викачування файлів з онлайн-ресурсів, переважно використовують велику кількість підходів до вирішення цієї проблеми [2].

Запропонований у роботі підхід виконує рекурсивний обхід, використовуючи структуру даних «дерево» для завантаження папок та файлів з вебсайту або сервера, що, завдяки ієрархічній структурі відповідних сховищ, суттєво прискорює процес обходу та завантаження, а також їх збереження на локальному диску. Розроблене програмне рішення дозволяє враховувати особливості процесу завантаження, в коді передбачені додаткові можливості перевірки наявності файлів на локальному диску перед завантаженням та обробки можливих помилок.

Розроблена програма покликана забезпечити зручну взаємодію з користувачем. Вона пропонує ввести адресу сайту, звідки будуть завантажуватися файли, а також шлях для збереження даних. За допомогою об'єкта WebClient [3] програма виконує завантаження та збереження каталогу з файлами. Ключовою

частиною коду є рекурсивна функція DownloadFolder. Вона отримує URL сайту та шлях для збереження файлів і виконує процес їх завантаження. Функція здійснює завантаження папок і файлів із сайту чи то сервера, а потім зберігає їх на локальному диску. Для отримання списку папок та файлів використовується об'єкт WebClient, який аналізує вебресурс для отримання посилань на файли та папки. Отримані посилання перевіряються та обробляються для завантаження кожного файлу окремо.

Запропоноване програмне рішення можна використовувати в різний спосіб, воно забезпечує автоматизоване завантаження великої кількості файлів, що може бути корисним у випадках, коли необхідно завантажити архіви, бази даних, зображення, відео та інші типи файлів із вебсайтів для подальшого аналізу, оброблення чи то збереження.

Основними цілями запропонованого програмного підходу є:

- забезпечення автоматичного завантаження папок та файлів із вебсайту чи то сервера;
- збереження завантажених файлів на локальний диск.
- здійснення перевірки наявності файлів на локальному диску перед завантаженням;
- обробка помилок, що можуть виникати під час завантаження файлів.

На практиці запропонований програмний підхід підтвердив свою ефективність, як зручний засіб завантаження файлів із вебсайтів, серверів та інших ресурсів. Подібного роду завантажувач файлів може бути застосований у багатьох сферах, як-от: веброзробка, аналітика даних, автоматизація процесів тощо.

Список використаних джерел:

1. Manually deploy software updates. URL: https://learn.microsoft.com/en-us/mem/configmgr/sum/deploy-use/manually-deploy-software-updates
2. Асинхронне оновлення програми на C#. URL: https://habr.com/ru/articles/217627/
3. WebClient Class. URL: https://learn.microsoft.com/en-us/dotnet/api/system.net.webclient?view=net-7.0 Class (System.Net) | Microsoft Learn

Ключові слова: автоматизація завантаження, оновлення програмного забезпечення, рекурсивне завантаження

Keywords: automation download, software update, recursive download

Науковий керівник: доцент Трофименко О. Г.

ОЦІНКА ЧАСОВОЇ СКЛАДНОСТІ ДЕЯКИХ АЛГОРИТМІВ СОРТУВАННЯ У ПОПУЛЯРНИХ МОВАХ ПРОГРАМУВАННЯ

Корнійчук М. М.

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

Трофименко О. Г.

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

Опрацювання даних лежить в основі роботи переважної більшості програмних застосунків, адже здебільшого є не достатнім просте відображення даних у необробленому вигляді. Доволі часто дані стають об'єктом різноманітних перетворень та предметом аналізу, який здійснюється з метою формування статистики та тенденцій. Ефективність здійснення таких операцій ускладнюється пропорційно зі збільшенням обсягів даних, особливо коли дані є невпорядкованими. В таких випадках пошук середнього значення (медіани), екстремумів (максимуму та мінімуму) чи то інших характеристик для набору даних може виявитися ресурсомісткою задачею, для виконання якої доведеться перебрати кожне значення з набору даних. Цю проблему можуть розв'язати алгоритми сортування.

Як відомо, сортування налічує численні алгоритми, кожен з них має свою специфіку, яка зумовлює відповідні переваги та недоліки. Для більшості людей, дотичних до розробки програмного забезпечення, знайоме поняття «Big O notation», яке у програмуванні вживається на позначення функції, що виражає залежність між обсягом вхідних даних та часом роботи алгоритму або необхідним обсягом пам'яті. Цю залежність вже встановлено та обґрунтовано для всіх відомих алгоритмів сортування і вона не залежить від мови, якою реалізовано алгоритм [1]. Так, для алгоритмів сортування бульбашкою та Шелла часову складність оцінюють як O(n^2), а для Merge та швидке сортування – O(n log(n)).

Предметом цього дослідження є прикладний аспект того, як реалізація конкретною мовою програмування впливає на фактичний час виконання алгоритму для масиву псевдовипадкових чисел. Для порівняння було зосереджено увагу на чотирьох найбільш поширених алгоритмах сортування: сортування бульбашкою (Bubble Sort), сортування Шелла (Shell Sort), сортування злиттям (Merge Sort) та швидке сортування (Shell Sort). Для імплементації було використано чотири найбільш популярні мови програмування: JavaScript, Java, C++ та PHP. Такий вибір обумовлено роллю, яку ці мови наразі відіграють наразі у сфері розробки

програмного забезпечення, Data Science, Data Analytics, Machine Learning тощо [2].

У дослідженні порівнювали фактичний час роботи кожного алгоритму для наборів даних від 100 до 100 000 елементів для кожної з вибраних мов програмування. Часом вважали середнє арифметичне значення п'яти вимірювань часу роботи алгоритму, оскільки цей підхід вже демонстрував свою ефективність в інших дослідженнях [3].

Алгоритми, реалізовані мовою PHP досліджувалися з використанням PHP версії 8.2.0, PHP Server extension for VS Code версії 3.0.2. Дослідження роботи алгоритмів, написаних мовою JavaScript (JS), проводилося з використанням Node.js версії 19.8.1. Для розрахунку часової складності використовувався метод now об'єкта performance, який повертає час у мілісекундах. В ході проведення вимірювань для алгоритмів, імплементованих мовою JS із використанням Node.js, було підтверджено проблему з алгоритмом quicksort при застосуванні рекурсії, а саме – неможливість ефективного сортування великих обсягів невпорядкованих даних через те, що зі зростанням числа елементів збільшується ризик переповнення стека викликів, внаслідок великої кількості рекурсивних викликів, які здійснюються під час роботи алгоритму. Так, переповнення стека викликів відбувалося при досягненні вихідним масивом розміру у понад 6 000 елементів. Аналогічна проблема спостерігалася під час дослідження роботи алгоритму, написаного мовою Java з використанням JDK 8, однак у цьому випадку переповнення стека відбувалося при досягненні масивом розміру 20 000 елементів. Водночас переповнення стека не відбувалося під час роботи цього самого алгоритму написаного мовою C++ і скомпільованого за допомогою g++.

Підсумовуючи результати дослідження варто зазначити, що алгоритми, імплементовані засобами мови Java, спрацювали швидше, ніж аналогічні алгоритми імплементовані мовою PHP, однак повільніше, ніж ці ж алгоритми, реалізовані мовою C++. Найшвидше працювали алгоритми, написані мовою JS у середовищі Node.js, що може бути зумовлено тим, що інтерпретатор для мови JS, яким користується середовище Node.js було написано засобами C/C++.

Подальші дослідження можуть стосуватися з'ясування факторів можливого впливу на роботу алгоритмів факторів залежності від архітектури комп'ютера, операційної системи, програмного середовища розробки та виконання програми, типів та обсягів даних, їх значень.

Список використаних джерел:

1. Big-O Complexity Chart. URL: https://www.bigocheatsheet.com/
2. Durrani O., Khan H. Asymptotic Performances Of Popular Programming Languages for Popular Sorting Algorithms. Bandaoti Guangdian/Semiconductor Optoelectronics. 2023. Vol. 42. P. 149-169.
3. Marcellino M., Pratama D. W., Suntiarko S. S., Margi K. Comparative of Advanced Sorting Algorithms (Quick Sort, Heap Sort, Merge Sort, Intro Sort, Radix Sort) Based on Time and Memory Usage. *1st International IEEE Conference on Computer Science and Artificial Intelligence* (ICCSAI). 2021. Vol. 1. P. 154-160.

Ключові слова: часова складність, алгоритми сортування, сортування бульбашкою, сортування Шелла, сортування злиттям, швидке сортування

Keywords: time complexity, sorting algorithms, bubble sort, Shell sort, Merge sort, quick sort

ОПИС СТРУКТУР ДАНИХ, ЩО НАЙЧАСТІШЕ ВИКОРИСТОВУЮТЬСЯ В РОЗРОБЦІ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Ковальський О.О.

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

Структури даних – це спосіб організації даних, з якими працює програмне забезпечення. Кожна структура унікально визначається тим, як його елементи зберігаються безпосередньо у пам’яті комп’ютера і набором операцій, за допомогою яких можна маніпулювати структурою або її окремими частинами. Для кожної структури даних обов’язкові дві операції: додавання нового елементу, та отримання вже наявного елементу.

Ефективність майже всього програмного забезпечення залежить від правильного вибору способу організації даних, з якими воно взаємодіє. Виразним прикладом впливу організації даних на швидкодію програмного забезпечення є пошук найменшого елементу серед даного набору цілих чисел. Уявімо собі, що є 100 000 000 цілих чисел. Якщо вони організовані як несортований масив, то у найгіршому випадку програмному забезпеченню доведеться перебрати всі 100 000 000 елементів. А якщо дані будуть структуровані як бінарне дерево пошуку, то тоді у найгіршому випадку доведеться перевірити лише [log2100 000 000] = 27 елементів. [x] позначає округлення x до +∞. Одними з найбільш поширених структур даних є [1]:

- масив;
- стек;
- черга;
- дерево;

Масив – це найбільш простий спосіб організації даних [2]. Він служить основою для деяких інших структур даних таких, як, наприклад, черга або стек. Елементи масиву зберігаються у послідовних комірках пам'яті комп'ютера, та кожен з них має свій унікальний для даного масиву індекс, що є цілим невід'ємним числом. Нумерація індексів починається з нуля. Такі особливості масиву дозволяють реалізувати миттєве повернення елементу масиву, якщо вказано його індекс. Типовими операціями з масивом є додавання елементу за вказаним індексом, видалення елементу за вказаним індексом, повернення елементу за вказаним індексом, повернення довжини вказаного масиву та пошук у масиві вказаного значення.

Стек – ще одна розповсюджена структура даних [3]. Вона використовується для організації пам'яті, виділеної для виконання певного програмного забезпечення. Ця виділена частина пам'яті називається – стеком. Також цей спосіб організації інформації використовується в інтегрованих середовищах розробки для відстеження відповідності відкритих дужок закритим дужкам. Стек працює за принципом LIFO – Last In First Out (останнім зайшов – першим вийшов). Його значення полягає у тому, що елемент доданий до стеку останнім, буде повернений з нього першим. Якщо користувачеві необхідно отримати довільний не перший елемент, йому потрібно повернути перед цим всі елементи, що були додані пізніше нього. У програмному забезпеченні стек реалізують на основі вже вбудованої у мову програмування реалізації масиву. Це означає, що дані стеку теж містяться у послідовних комірках пам'яті, але операції над цими даними відрізняються від тих, що притаманні масиву. Стек містить інформацію про поточний індекс. Коли користувач починає додавати елементи до пустого стеку, він починає з додавання елементу у асоційований з даним стеком масив за індексом 0. При кожному заштовхуванні елементу цей індекс збільшується на 1. При операції виштовхування зі стеку, з масиву повертається елемент за індексом, на 1 меншим від поточного індексу і одразу ж видаляється, а поточний індекс зменшується на 1.

Черга, як і стек, реалізується на основі масиву [4]. Проте вона функціонує за принципом, протилежним принципу стека. Він називається FIFO – First In First Out (перший зайшов – перший вийшов). Черга використовується у тих випадках, коли потрібно спланувати порядок виконання запитів. Наприклад, запити клієнтів до агенту має сенс організувати у вигляді черги, для того, щоб справедливо обслуговувати клієнтів. Черга містить інформацію про два індекси: один для заштовхування елементів у асоційований із чергою масив, а інший для виштовхування елементів з відповідного масиву. При заштовхуванні елементу у пусту чергу відповідний індекс дорівнює 0. Після кожного заштовхування цей індекс

збільшується на один. Для виштовхування елементів використовується інший індекс, що не залежить від операції заштовхування, має початкове значення 0 і збільшується на один із кожною операцією виштовхування.

Структура даних дерево використовується для зберігання елементів, що знаходяться у ієрархічних відносинах [5]. У реалізації дерева на практиці, кожен елемент містить безпосередньо деяку інформацію, яка називається вузлом дерева та вказівники на інші вузли. Ці вказівники називаються гранями дерева. Якщо елемент дерева містить вказівник на інший елемент дерева, то перший з них називається, батьківським а другий – дочірнім. Дочірній елемент не може вказувати на батьківський елемент.

Елемент, що не має батьківського елементу називається коренем дерева. У дереві може бути лише один корінь. Елементи, що не мають дочірніх елементів називаються листками дерева, вони є кінцевими елементами. Типовими операціями з деревом є додавання елементу до дерева, видалення елементу з дерева, пошук по дереву у ширину та пошук по дереву у глибину.

Отже, існують різні структури даних, і кожна з них використовується для організації різної інформації у різних випадках. Кожній структурі притаманні свої операції для роботи з нею. Деякі структури реалізовані на основі вже існуючих, а деякі реалізуються без використання вже наявних.

Список використаних джерел:

1. The top data structures you should know for your next coding interview. URL: https://www.freecodecamp.org/news/the-top-data-structures-you-should-know-for-your-next-coding-interview-36af0831f5e3/. (Дата звернення 20.05.2023).
2. Цмоць І. Г., Парубчак В. О., Антонів В. Я. Паралельно-вертикальне сортування одновимірних масивів даних методом злиття з використанням підрахунку //Збірник наукових праць інституту проблем моделювання в енергетиці ім. ГЄ Пухова. – 2013. №. 68. С. 92-100.
3. Операційні системи : навчальний посібник [Електронне видання] / О. В. Задерейко, С. Л. Зіноватна, А. А. Толокнов. Одеса : Фенікс, 2022. 140 с. ISBN 978-966-928-828-8. – URL: https://hdl.handle.net/11300/22701.
4. Бугаєва Л. М., Ковалюк Д. О. Алгоритми та структури даних. Комп'ютерний практикум. URL: https://ela.kpi.ua/bitstream/123456789/55685/1/Alhorytmy_ta_struktury_danykh.pdf. (Дата звернення 20.05.2023).
5. Бершов Є. В. Порівняльний аналіз реалізації структур даних «Зв'язний список» та «Бінарне дерево пошуку» в прикладних застосуваннях. 2021. URL: https://essuir.sumdu.edu.ua/bitstream-download/123456789/84745/1/Bershov_bac_rob.pdf

Ключові слова: структури даних, масив, стек, черга, дерево

Keywords: data structures, array, stack, queue, tree

Науковий керівник: доцент Задерейко О.

ОПРАЦЮВАННЯ ДЕЯКИХ ФУНКЦІЙ ІЗ МАТРИЦЯМИ В MICROSOFT EXCEL

Шляхтицький Д.С.

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

Опрацювання матриць в Microsoft Excel є актуальним і корисним інструментом для багатьох користувачів. Excel має ряд функцій і можливостей, що спрощують обробку та аналіз даних в матричному форматі.

Ось декілька причин, чому опрацювання матриць в Excel є актуальним:

Обчислення і аналіз даних: Excel дозволяє використовувати формули та функції для виконання різних операцій з матрицями. Ви можете виконувати арифметичні операції, знаходити мінімуми, максимуми, середні значення, сортувати та фільтрувати дані в матрицях.

Аналіз та візуалізація даних: Excel надає можливості для створення графіків та діаграм, які допомагають візуалізувати дані з матриць. Це дозволяє легше сприймати та аналізувати великі обсяги інформації.

Робота з великими наборами даних: Excel може обробляти великі обсяги даних в матричному форматі. Це дозволяє ефективно працювати зі значними таблицями, базами даних або іншими джерелами, які містять багато рядків та стовпців.

Матричні функції: Excel має набір спеціальних матричних функцій, які дозволяють виконувати різні операції з матрицями. Наприклад, ви можете обчислювати визначники, знаходити обернені матриці, розв'язувати системи лінійних рівнянь та багато іншого.

Автоматизація і макроси: Excel дозволяє автоматизувати операції з матрицями за допомогою макросів та VBA (Visual Basic for Applications).

Транспонування матриці. Дана матриця з 5 рядків та 8 стовпців.

	A	B	C	D	E	F	G	H	I
1	Довільна матриця								
2	23	60	59	42	23	85	80	22	
3	67	92	52	21	2	31	98	25	
4	95	28	83	96	69	86	30	8	
5	10	71	11	33	41	48	81	44	
6	45	75	5	9	40	25	45	55	
7									

Як транспонувати матрицю? Необхідно замінити всі рядки матриці на відповідні (з тими ж номерами) стовпці цієї ж матриці. Дану операцію можна здійснити двома способами.

1 спосіб – це використати базову функцію Excel “ТРАНСП( )”. Виділяємо місце для транспонованої матриці.

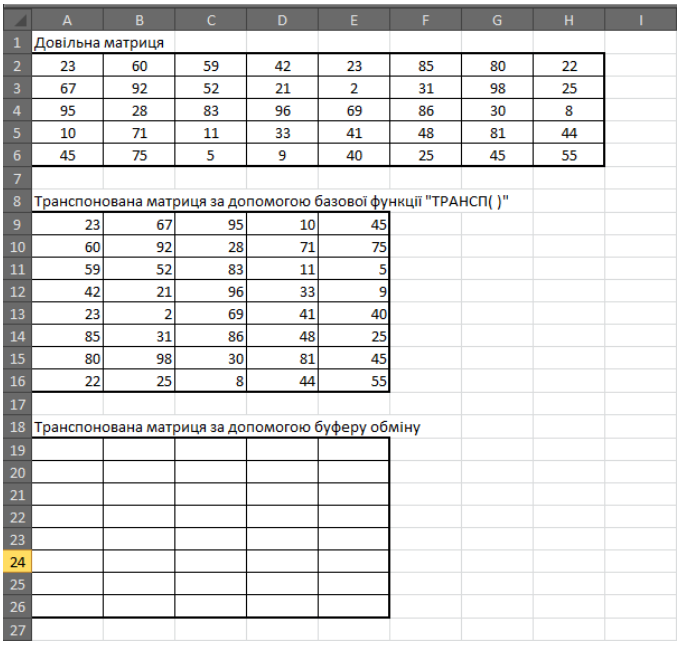
Натискаємо на комірку A8 та переходимо до Мастера функцій. Там серед повного списку функцій необхідно знайти функцію ТРАНСП( ). Як вхідні дані виділити усю довільну матрицю. Але тут виникає питання:

	A	B	C	D	E	F	G	H	I
1	Довільна матриця								
2	23	60	59	42	23	85	80	22	
3	67	92	52	21	2	31	98	25	
4	95	28	83	96	69	86	30	8	
5	10	71	11	33	41	48	81	44	
6	45	75	5	9	40	25	45	55	
7									
8	Транспонована матриця за допомогою базової функції "ТРАНСП( )"								
9	#ЗНАЧ!								
10									
11									
12									
13									
14									
15									
16									
17									
18									

Excel вважає що формат клітинки B9 є помилковим. Для вирішення цієї проблеми необхідно виділити усю транспоновану матрицю починаючи з клітинки B9. Після виділення матриці натискаємо клавішу F2 та переходимо в режим редагування таблиці. Далі необхідно ввести комбінацію клавіш Ctrl+Shift+Enter. Після натискання комбінації матриця, що є транспонованою до довільної буде виведена в таблиць з початком у комірці, де була введена формула.

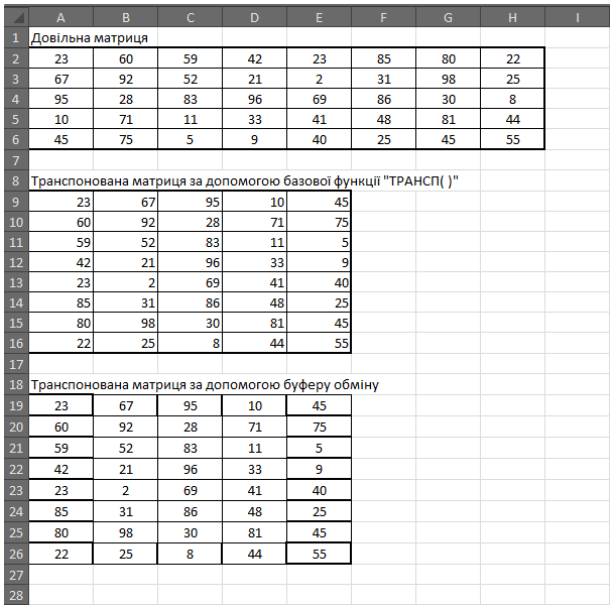
	A	B	C	D	E	F	G	H	I
1	Довільна матриця								
2	23	60	59	42	23	85	80	22	
3	67	92	52	21	2	31	98	25	
4	95	28	83	96	69	86	30	8	
5	10	71	11	33	41	48	81	44	
6	45	75	5	9	40	25	45	55	
7									
8	Транспонована матриця за допомогою базової функції "ТРАНСП( )"								
9	23	67	95	10	45				
10	60	92	28	71	75				
11	59	52	83	11	5				
12	42	21	96	33	9				
13	23	2	69	41	40				
14	85	31	86	48	25				
15	80	98	30	81	45				
16	22	25	8	44	55				
17									

Також, є інший спосіб – за допомогою буферу обміну, тобто операцію копіювати-вставити.



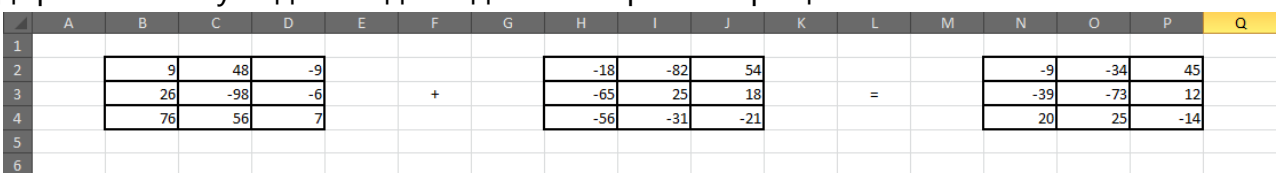
	A	B	C	D	E	F	G	H	I
1	Довільна матриця								
2	23	60	59	42	23	85	80	22	
3	67	92	52	21	2	31	98	25	
4	95	28	83	96	69	86	30	8	
5	10	71	11	33	41	48	81	44	
6	45	75	5	9	40	25	45	55	
7									
8	Транспонована матриця за допомогою базової функції "ТРАНСП( )"								
9	23	67	95	10	45				
10	60	92	28	71	75				
11	59	52	83	11	5				
12	42	21	96	33	9				
13	23	2	69	41	40				
14	85	31	86	48	25				
15	80	98	30	81	45				
16	22	25	8	44	55				
17									
18	Транспонована матриця за допомогою буферу обміну								
19									
20									
21									
22									
23									
24									
25									
26									
27									

Виділяємо довільну матрицю та копіюємо її. Вставити звичайним чином не вдасться, бо умова транспонування не виконається, тобто нам необхідна певний формат вставки. Виділяємо комірку A19 та відкриваємо у меню “Головна” спадаючий список “Вставити” та пункт “Спеціальна вставка”. Дані у віконці обрати пункт “Транспонувати”.



	A	B	C	D	E	F	G	H	I
1	Довільна матриця								
2	23	60	59	42	23	85	80	22	
3	67	92	52	21	2	31	98	25	
4	95	28	83	96	69	86	30	8	
5	10	71	11	33	41	48	81	44	
6	45	75	5	9	40	25	45	55	
7									
8	Транспонована матриця за допомогою базової функції "ТРАНСП( )"								
9	23	67	95	10	45				
10	60	92	28	71	75				
11	59	52	83	11	5				
12	42	21	96	33	9				
13	23	2	69	41	40				
14	85	31	86	48	25				
15	80	98	30	81	45				
16	22	25	8	44	55				
17									
18	Транспонована матриця за допомогою буферу обміну								
19	23	67	95	10	45				
20	60	92	28	71	75				
21	59	52	83	11	5				
22	42	21	96	33	9				
23	23	2	69	41	40				
24	85	31	86	48	25				
25	80	98	30	81	45				
26	22	25	8	44	55				
27									
28									

Додавання матриць. Додавання матриць досить тривіальне та елементи дорівнюють сумі двох відповідних комірок матриці.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1																	
2		9	48	-9				-18	-82	54				-9	-34	45	
3		26	-98	-6		+		-65	25	18		=		-39	-73	12	
4		76	56	7				-56	-31	-21				20	25	-14	
5																	
6																	

Множення матриці на число. Для множення матриці на число необхідно кожен елемент матриці помножити на це число.

Множення матриці на число

				-23	34	-29			-46	68	-58		
	2	*		-11	-30	-34	=		-22	-60	-68		
				-56	-59	-92			-112	-118	-184		

Множення матриці на матрицю. Для множення матриці на матрицю необхідно виконувати умову – кількість стовпців 1 матриці має дорівнювати кількості рядків 2 матриці. Для множення ми використовуємо базову функцію Excel "МУМНОЖ( )". Як вхідні дані вводимо дві наші матриці. Маємо пам'ятати, що розмір шуканої матриці буде таким: кількість стовбців 1 матриці та рядків 2. В початковій клітинці буде схожа ситуація як і з транспонуванням. Знову F2 та Ctr+Shift+Enter.

Множення матриці одна на одну

4	-37	46	20	-1		-7	44	-74		809	-3828	-7652	
11	35	-82	88	93		-35	-10	37		2484	2164	-6776	
96	71	-83	93	-66		7	-57	-88		-12273	-3092	-2447	
18	-97	65	-4	94	*	-35	-85	-100	=	11384	3285	-15975	
49	-20	-33	45	58		80	52	-61		3191	3428	-9500	
-98	1	-24	-38	29						4133	1784	11432	
100	35	-6	-14	55						2923	8442	-7532	

Обернена матриця. Для знаходження оберненої матриці ми знову ж використовуємо функцію від Excel "МОБР( )". Маємо пам'ятати, що кількості рядків та стовпців мають збігатись. Розміри довільної та оберненої матриць будуть ідентичними. Беремо довільну матрицю як вхідні дані для функції та дотримуємось вже звичного механізму "F2, Ctrl+Shift+Enter".

99	-74	-71	-25	-23	-83	
79	-5	-47	85	16	94	
97	17	15	8	-60	-84	
-63	-79	-38	1	32	51	
-38	-43	26	21	-50	60	
-71	71	-82	-69	22	-67	

Обернена

0.010882	0.001202	-0.0162	-0.01824	0.001292	-0.00421
-0.00178	0.002751	-0.00285	-0.00888	0.001169	0.003931
-0.00163	-0.00543	-0.00303	-0.00381	-0.00306	-0.00744
-0.02373	0.005503	0.043167	0.040266	-0.01213	0.002779
-0.00321	-0.00107	-0.00098	0.006883	-0.01542	-0.00487
0.011948	0.002266	-0.02692	-0.02463	0.011043	-0.00165

Визначник Матриці. Для знаходження визначника матриці Excel пропонує функцію "МОПРЕД( )". Як вхідні дані обираємо нашу матрицю та отримуємо визначник. Не забуваємо, що для виведення коректного результату матриця має бути квадратною.

Отже, опрацювання матриць в Microsoft Excel є актуальним і корисним

інструментом для аналізу та обробки даних. Використання матриць у Excel надає можливість виконувати різноманітні операції з даними, що дозволяє ефективно обчислювати та аналізувати інформацію великих обсягів за допомогою власно створених або базових функцій Excel; візуалізувати її шляхом графіків та діаграм.

Список використаних джерел:

1. Функція ТРАНСП. URL: https://support.microsoft.com/office/ТРАНСП
2. Функція МУМНОЖ. URL: https://support.microsoft.com/office/МУМНОЖ
3. Функція МОБР. URL: https://support.microsoft.com/office/МОБР
4. Функція МОПРЕД. URL: https://support.microsoft.com/office/МОПРЕД

Ключові слова: матриці в Excel, Microsoft Excel, матриця, транспонування
Keywords: matrix in the Excel, Microsoft Excel, matrix, transposition

Науковий керівник: старший викладач Баландіна Н.М.

ЗАКОН ЄС ПРО ЗАПРОВАДЖЕННЯ УНІВЕРСАЛЬНОГО ЗАРЯДНОГО ПРИСТРОЮ

Жоржа В. О.

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

23 вересня 2021 року Єврокомісія офіційно представила законопроєкт про єдиний зарядний пристрій для всіх мобільних телефонів, планшетів, камер, навушників, колонок та ігрових консолей.

Основні соціальні наслідки, очікувані від цієї ініціативи, пов'язані зі зручністю для споживачів. Також очікуваними наслідками є зменшення/мінімізація електронних відходів за рахунок зменшення необхідності придбання різних типів зарядних пристроїв та надання можливості повторно використовувати вже належні.

Комісар ЄС з питань конкуренції Маргрете Вестагер прокоментувала законопроект наступним чином: «Європейські споживачі вже давно розчаровані несумісними зарядними пристроями, що накопичуються в їх шухлядах. Ми дали учасникам ринку достатньо часу, щоб напрацювати власні рішення, але тепер настав час законодавчих ініціатив» [1].

Цьому законопроекту передувала постанова про єдиний стандарт зарядки мобільних пристроїв, яку Європарламент схвалив ще в січні 2020 року. І тоді тільки Apple виступила проти цього - виробник iPhone заявив, що це «задушить інновації» і ненавмисно створить ще більше електронних відходів.

Згідно з дослідженням Єврокомісії, станом на 2018 рік половина гаджетів в

ЄС були оснащені роз'ємом microUSB, ще 29% - роз'ємом USB Type-C, а 21% - роз'ємом Lightning, який використовується в пристроях Apple. У тому ж 2018 Apple перевела iPad Pro на універсальний роз'єм USB Type-C замість Lightning, але очікуваної відмови від фірмового роз'єму Lightning в iPhone так і не відбулося, незважаючи на регулярні повідомлення інсайдерів і аналітиків. Актуальна лінійка iPhone залишилася на Lightning.

У вересні 2022 року Європейська Рада остаточно затвердила закон, що визначає USB Type-C як обов'язковий та універсальний порт для всіх електронних пристроїв за окремими винятками. З кінця 2024 року всі гаджети, що будуть продаватися на території Європейського Союзу, повинні мати USB Type-C.

Після отримання підпису Президента Європарламенту та Президента Європейської Ради закон буде опублікований в офіційному журналі Європейського Союзу та набере чинності через 20 днів після публікації. Компаніям-виробникам електроніки надається ще два роки на адаптацію до нових вимог [2].

Затвердження USB Type-C як юридично обов'язкового універсального порту вплине перш за все на компанію Apple, оскільки вона широко використовує роз'єм Lightning замість USB Type-C на багатьох своїх пристроях. Члени Європарламенту стверджують, що цей крок зменшить електронне сміття та зробить використання різних пристроїв більш зручним. За даними ЄС, ці вимоги допоможуть скоротити обсяги електронних відходів на 11 тис. тонн щороку.

USB Type-C буде обов'язковим для мобільних телефонів, планшетів та електронних книг, цифрових камер, навушників, бездротових колонок, ігрових консолей, бездротових мишей та клавіатур. Крім того, USB Type-C повинні мати всі нові ноутбуки, але період адаптації для них становитиме 40 місяців замість 24. Винятки будуть застосовуватися до пристроїв, які занадто малі для порту USB Type-C, таких як смарт-годинники та деяке спортивне обладнання.

ЄС також планує забезпечити сумісність рішень для бездротового заряджання. Закон вимагає, щоб до кінця 2024 року Єврокомісія розробила акти, які зобов'язуватимуть компанії створювати універсальні рішення для бездротового заряджання. Невідомо, чи стосуватиметься ця вимога системи Apple MagSafe, яка базується на стандарті бездротового заряджання Qi.

Рішення про універсальний порт для заряджання Єврокомісія спробувала прийняти ще в 2018 році, але зіткнулася з активним опором з боку Apple, яка попередила, що примус використовувати спільний зарядний порт придушить інновації. Обговорення закону відновилося минулого року. У червні 2022 року Комітет ЄС з внутрішнього ринку та захисту прав споживачів вимагав внести закон на обговорення до Європейського парламенту, який потім переважною

більшістю голосів проголосував за нього.

Новий законопроєкт вимагає однакової швидкості зарядки для будь-якого сумісного зарядного пристрою. Він обов'язково передбачає використання USB Power Delivery разом з портом Type-C, якщо струм перевищує 5 В та сила становить 3 А. Це означає, що Apple більше не може обмежувати швидкість зарядки, посилаючись на сертифікацію MFI. Однак, журналісти вказують на можливу обхідну дію Apple, яка може змусити користувачів купувати сертифіковані кабелі для функцій передачі даних через універсальний роз'єм [3].

В Європейському Союзі з 28 грудня наступного року всі смартфони, включаючи майбутні айфони, повинні мати порт USB Type-C. Закон також стосується інших пристроїв, таких як планшети, навушники та електронні книги, за винятком невеликих гаджетів, в які неможливо встановити USB Type-C. Очікується, що подібні стандарти будуть введені для ноутбуків, але це може зайняти до 2026 року. Те ж саме стосується стандартів швидкої зарядки.

Отож, USB Type-C - це роз'єм, який використовується для підключення різних пристроїв, таких як смартфони, планшети, ноутбуки, периферійні пристрої і зарядні пристрої. Він має кілька переваг, які роблять його важливим. USB Type-C підтримує швидкий передачу даних і зарядку, може використовуватися для підключення до відео та аудіо пристроїв, а також має зворотну сумісність з попередніми версіями USB. Це означає, що ви можете використовувати один кабель для різних пристроїв, що спрощує використання технологій і заряджання.

Прямим наслідком прийняття цього закону стала ініціатива сенаторів США, які заявили, що США мають наслідувати приклад Європейського Союзу і вимагати від виробників електроніки впровадити єдиний зарядний пристрій для мобільних телефонів. «Політика ЄС може значно скоротити електронні відходи та допомогти споживачам, які втомилися ритися в ящиках, повних заплутаних зарядних пристроїв, у спробі знайти сумісне чи купити нове», — йдеться у листі сенаторів на ім’я міністерки торгівлі Джини Раймондо [4].

Успішне впровадження стандарту USB Type-C допомагає забезпечити єдиною стандартною системою зарядки та підключення пристроїв, що полегшує життя користувачам і сприяє зменшенню кількості різних кабелів та зарядних пристроїв.

Список використаних джерел:

1. Скрипін Володимир Єврокомісія запропонувала зробити USB-C обов'язковим для зарядки всіх мобільних пристроїв, включаючи iPhone [Електронний ресурс]. – URL: https://itc.ua/news/evrokomissiya-predlozhila-sdelat-obyazatelnym-usb-c-dlya-zaryadki-vseh-mobilnyh-ustrojstv-vklyuchaya-iphone/

2. У ЄС виробники смартфонів матимуть два роки для запровадження роз'ємів USB Type-C. 2022. [Електронний ресурс]. – URL: https://suspilne.media/337776-u-es-virobniki-smartfoniv-matimut-dva-roki-dla-zaprovadzenna-rozemiv-usb-type-c//
3. ЄС заборонить Apple обмежувати швидкість роботи USB Type-C в нових iPhone. 2023. [Електронний ресурс]. – URL: https://www.unian.ua/techno/yes-zaboronit-apple-obmezhuvati-shvidkist-roboti-USBType-C-v-novih-iphone-zmi-12178752.html/.
4. Карпусь Вадим У США слідом за ЄС пропонують запровадити універсальні заряднi пристрої для мобільних гаджетів [Електронний ресурс]. – URL: https://itc.ua/ua/novini/u-ssha-slidom-za-yes-proponuyut-zaprovaditi-universalni-zaryadni-pristroyi-dlya-mobilnih-gadzhetiv/

Ключові слова: зарядний пристрій, стандарт USB Type-C, універсальний порт
Keywords: charger, USB Type-C standard, universal port

Науковий керівник: доцент Чанишев Р.І.

ПОРІВНЯННЯ ТЕХНОЛОГІЙ SIM, eSIM ТА iSIM

Лєсняков М. Р.
студент 1 курсу, факультету судового та міжнародного права
Національного університету «Одеська юридична академія»

Сім карти в своїй основі відомі кожному, але в епоху сучасних технологій розвиток та модернізація навіть такої системи не стоїть на місці.

SIM-карта була впроваджена з появою мереж 2G, а її головним завданням стало спростити ідентифікацію абонентів мережі. Абревіатура SIM (Subscriber Identity Module) розшифровується як модуль ідентифікації абонента.

SIM-картка дозволяє підключатися до послуг оператора мобільного зв'язку, що дозволяє виходити в інтернет або дзвонити. Це стара модель, але чи має вона альтернативу? Так, на її заміну прийшла нова технологія, це- eSIM.

eSIM — це цифрова SIM-картка, яка дає змогу використовувати стільниковий план оператора зв'язку без використання фізичної картки nano-SIM. eSIM є аналогом пластикової картки, так само, як і вона, дозволяючи отримувати послуги операторів мобільного зв'язку, але кардинальна різниця полягає у форм-факторі. Вона являє собою мікро чіп встановлений у гаджеті на материнські платі.

Також чіп eSIM може бути встановлений не тільки у смартфони а й у смарт-годиники, планшети, ноутбуки тощо. По суті, картки працюють за однаковим принципом, відрізняючись форм-фактором, способом встановлення та підключення. Різниця в цих параметрах суттєва, але кардинальних відмінностей у способі передачі немає. Я сам користуюся цим, і можу сказати що це як мінімум

комфортніше, вона в мене виступає у вигляді другої картки яка підключена до іншого оператора, не було б цієї системи, потрібно було би кожного рази виймати фізичну карту першу, і замінювати її на другу. Як би здалося на перший погляд, це вершина технологічного прориву, але ні, не так давно було винайдено ще одну систему карток, її називають iSIM.

iSIM відповідає специфікаціям GSMA і надає ряд переваг, таких як підвищена продуктивність і поліпшена системна інтеграція [1].

Вперше, цю систему впровадили в життя компания Apple, в своїй останній моделі айфону 14. Винайшла цю сім-карту компанія Qualcomm, яка вбудовується безпосередньо у процесор пристрою. iSIM розшифровується як integrated SIM або, простіше, інтегрована.

Вона знаходиться всередині чіпсету і не потребує окремого живлення та з'єднання з модемом. Площа, яку займає iSIM всього 1 мм2, що набагато менше, ніж площа картки формату nanoSIM.

Виходячи з цього, можна одразу зрозуміти, що у смартфоні з'явиться вільне місце. Як мінімум, це дозволить розмістити акумулятор трохи більшого розміру, а як максимум - придумати щось цікаве з камерами, просто змістивши інші компоненти. «Ну і, як зазначає Qualcomm, завдяки тому, що окреме живлення для iSIM не потрібно, то витрата акумулятора на стільникову мережу стане нижчою».

Що до переваг, вони звичайно не такі колосальні як між звичайною фізичною карткою та eSIM, але теж мають місце бути.

Технологія iSIM забезпечує переваги як для споживачів, так і для операторів. Це дозволить легко інтегрувати мобільні сервіси в різні пристрої, включаючи ноутбуки, планшети, платформи віртуальної реальності, пристрої Інтернету речей, носимі пристрої тощо.

Технологія iSIM дозволяє оператору віддалено використовувати існуючу інфраструктуру eSIM. Крім того, оператор може відкрити можливість підключення до мобільних послуг для ряду пристроїв, які раніше не могли мати вбудованих можливостей SIM [2].

Завдяки Qualcomm ми зрозуміли що прогрес не стоїть на місці, і бачимо яких висот досягнув. iSIM це новий вид сім-карт, і він не дуже перевірений часом. У нас є всі вхідні данні по ній, але як воно працює в повсякденному житті не відомо, тому віддамо один бал eSIM, а що ж казати до сучасних технологій то iSIM сильно попереду і за це йому +, але це все так звучить багатообіцяюче.

Як сказав Гійом Лафе, віцепрезидент із вбудованих продуктів Thales Mobile: «Перша у світі сертифікація безпеки iSIM від GSMA стала результатом кількох років інтенсивної розробки Qualcomm Technologies та Thales. Поряд з eSIM, що

набирає популярності, Thales 5G iSIM дає виробникам пристроїв і операторам мобільного зв'язку ще більшу свободу пропонувати своїм клієнтам бездротове підключення, а також цікавіші та доступніші дизайни продуктів» [3].

Отже, треба розібратися та «пограти́ся» з цією системою, і вже тоді об'єктивно оцінювати цих суперників.

Список використаних джерел

1. Карпус Вадим Qualcomm зайнялася розробкою технології iSIM (SIM в процесорі) і провела демонстрацію з робочим смартфоном. URL: https://itc.ua/news/qualcomm-zanyalas-razvitiem-tehnologii-isim-sim-v-proczessore-i-provela-demonstracziyu-s-rabotayushhim-smartfonom
2. Ferreno Julian eSIM and iSIM: what they are and how they differ. URL: https://crast.net/8369/esim-and-isim-what-they-are-and-how-they-differ/
3. Даньшина Катерина eSIM більше не потрібна. Смартфони на Snapdragon 8 Gen 2 отримають інтегровану SIM-карту – iSIM. URL: https://itc.ua/ua/novini/smartfony-na-snapdragon-8-gen-2-otrymayut-integrovanu-sim-kartu-isim/

Ключові слова: технологія, SIM-карта, оператор мобільного зв'язку
Keywords: technology, SIM card, mobile operator

Науковий керівник: доцент Чанишев Р.І.

ВИБІР СУЧАСНИХ ЗАСОБІВ РОЗРОБКИ МОБІЛЬНИХ ЗАСТОСУНКІВ

Манаков С. Ю.

к.т.н., доцент кафедри інформаційних технологій
факультету кібербезпеки та інформаційних технологій
Національного університету «Одеська юридична академія»

Дика А. І.

асистент кафедри інформаційних технологій
факультету кібербезпеки та інформаційних технологій
Національного університету «Одеська юридична академія»

Завдяки поширеності використання мобільного зв'язку та нарощуванню потенціалу смартфонів, нині розробка застосунків для мобільних платформ є одним із перспективних напрямків розвитку сфери IT. Кількість завантажень мобільних застосунків у всьому світі постійно зростає, перевищивши 255 мільярдів 2019 року. Очікується, що у 2025 році мобільні застосунки принесуть понад 613 млрд доларів доходу [1]. Тому вибір технологій та засобів розробки мобільних застосунків є актуальним питанням для переважної кількості майбутніх фахівців цієї галузі.

На вибір засобів розробки мобільних застосунків впливають такі фактори:

- масштаб та складність проєкту;
- встановлений бюджет на терміни завершення проєкту;
- внутрішня орієнтованість та минулий досвід компанії-розробника;
- вимоги безпеки до проєкту.

Якщо розглядати розробку програм для iOS, то беззаперечним лідером є мова Swift [2] у середовищі Xcode для нативної розробки. Swift було представлено 2014 року на заміну Objective-C. Xcode містить інструменти, необхідні для розробки та налагодження коду, а також дизайнер графічного інтерфейсу для створення інтерфейсів користувача. Завдяки цьому розробники мають доступ до всього, що їм потрібне для створення високопродуктивних, надійних застосунків під iOS.

Для розробки програм під Android офіційним середовищем для нативної розробки від Google є Android Studio [3]. Це середовище характеризується відкритим доступом, частими оновленнями та підтримкою мов програмування Java, Kotlin та C++. Воно має потужний редактор коду з підказками синтаксису й інструментами рефакторингу, розширені можливості налагодження, монітори пам'яті та вбудований емулятор.

Наразі IT-сфера переживає зліт популярності кросплатформних інструментів. Найпопулярніші з них – фреймворки Flutter (мова Dart), React Native (JavaScript), .NET MAUI (C#) та технологія Kotlin MultiPlatform. З часом потреба в Android- та iOS-розробниках на ринку праці не стає меншою.

При виборі між нативною та кросплатформною розробкою слід пам’ятати, що платформи-конкуренти (Android та iOS) не зацікавлені у підтримці один одного, а мультиплатформні проєкти зазвичай вимагають у 1,5-2 рази більше ресурсів, порівняно з нативними.

.NET MAUI (раніше відома як Xamarin) від Microsoft орієнтована більше на кросплатформність, ніж на мобільну розробку. Основним недоліком платформи .NET MAUI є її нездатність запропонувати складну графіку та легкі програми.

Фреймворк React Native використовує поширено відому мову програмування JavaScript та бібліотеку React з можливістю використовувати власні функції та API-інтерфейси. Переваги React Native зумовлені високою продуктивністю, подібною до продуктивності нативних застосунків, та гарною підтримкою спільноти. Основні недоліки цього фреймворку стосуються недолугого керування пам'яттю, необхідності мати нативних розробників і проблем з безпекою JavaScript.

Flutter [4] використовує власну віртуальну машину та мову програмування Dart, має швидкий та продуктивний механізм рендерингу та пропонує узгоджений зовнішній вигляд на різних платформах. Однак, Flutter все ще не дуже

поширений і має великі розміри програмних застосунків.

Найбільшу динаміку зростання популярності набуває мова Kotlin. І Kotlin, і Flutter підтримуються Google. Нині і надалі їх порівнюватимуть дедалі частіше. Останнім часом Flutter надають перевагу тільки тому, що вона більш відома і має ширше ком'юніті. Проте, за прогнозами [5] з часом Kotlin стане наступним лідером. У Google навіть позиціонують Kotlin, як перспективну мову для backend-розробки.

Особливості Kotlin для мобільної розробки:

- статично типізована мова програмування, яка працює на платформі Java Virtual Machine (JVM);
- підтримує функціональне програмування, що дозволяє створювати більш короткі та читабельні програми;
- зручний і сучасний синтаксис, який дозволяє швидко та легко створювати високоякісні застосунки;
- численні вбудовані функції, як-от: null-безпека, розширення функцій, лямбда-вирази, які допомагають спростити розробку;
- підтримка багатопоточності, яка дозволяє створювати багатопоточні програми без потреби використання блокувань та синхронізації;
- відкрита ліцензія Apache 2.0, яка робить мову Kotlin доступною для використання у будь-яких проєктах;
- хороша інтеграція з Java, яка дозволяє використовувати наявний код Java в проєктах Kotlin і навпаки;
- менша швидкість компіляції, порівняно з Java;
- широка підтримка в інструментах розробки IntelliJ IDEA та Android Studio.

Окремо варто відзначити використання в Kotlin декларативного інтерфейсу користувача засобами інструментарію Jetpack Compose, замість громіздкої XML-розмітки View-елементів для Java та Xamarin. Декларативний підхід визначає стан екрану й оновлює елементи інтерфейсу користувача при зміненні стану. Такий підхід не новий і запозичений з Flutter та SwiftUI, але саме він дозволяє додатково суттєво зменшити обсяг коду при використанні Kotlin, замість Java.

Jetpack Compose побудований на так званих Composable-функціях. Ці функції дозволяють програмно визначати інтерфейс користувача безпосередньо у коді Kotlin, описуючи, як він має виглядати, і надаючи залежності даних, а не зосереджуючись на процесі побудови інтерфейсу користувача (ініціалізація елемента, приєднання його до батька тощо). Compose спрощує процес розробки інтерфейсу користувача, що дозволяє розробникам створювати якісні програми швидше. Розробники можуть використовувати функціональний підхід для

створення інтерфейсу користувача, що спрощує процес тестування і налагодження.

Jetpack Compose є офіційною бібліотекою Google. Вона має велику спільноту розробників та високу ймовірність підтримки в майбутньому. Щодо продуктивності, результати досліджень [6, 7] показують, що Compose має більш високу частоту кадрів, але повільніше завантаження сторінок. При цьому, після завантаження компонентів Compose частота заморожених кадрів і тривалість завантаження сторінки значно покращилися.

Незважаючи на вищеописані особливості інструментів мобільної розробки, всі вони так чи інше присутні на ринку, а їх вибір часто залежить від конкретних обставин.

Список використаних джерел:

1. Number of mobile app downloads worldwide from 2016 to 2022. URL: https://www.statista.com/statistics/271644/worldwide-free-and-paid-mobile-app-store-downloads.
2. Swift. URL: https://developer.apple.com/swift
3. Android Studio. URL: https://developer.android.com/studio
4. Flutter. URL: https://flutter.dev
5. Kotlin. URL: https://kotlinlang.org
6. Jetpack Compose. URL: https://developer.android.com/jetpack/compose
7. Comparing Jetpack Compose performance with XML. URL: https://medium.com/okcredit/comparing-jetpack-compose-performance-with-xml-9462a1282c6b

Ключові слова: мобільна розробка, Java, Dart, Kotlin, Jetpack Compose
Keywords: mobile development, Java, Dart, Kotlin, Jetpack Compose

КРИПТОВАЛЮТНІ ГАМАНЦІ

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

Криптогаманець – це спеціальне програмне (або апаратне) забезпечення, яке дає змогу зберігати цифрову валюту, а також здійснювати різні операції з нею. При цьому безпосередній контроль доступу до криптовалюти здійснюється з допомогою технології блокчейн. Тому для підтвердження права управління коштами необхідно мати спеціальні ключі – це унікальний набір символів, що складається з літер і цифр. Існують два різновиди ключів:

1. Публічний ключ можна порівняти з розрахунковим рахунком у банку. Він використовується для отримання платежів від інших користувачів та поповнення

балансу гаманця. Адреса в блокчейні є скороченою версією публічного ключа, яку може бачити будь-який користувач.

2. Приватний ключ можна уявити як віртуальний ключ до банківської скриньки сейфу. Він надає повний контроль над коштами, що зберігаються на рахунку. Приватний ключ використовується для створення цифрових підписів та перевірки транзакцій. Він є конфіденційним і відомий лише власнику адреси, оскільки надає доступ до його активів.

Особливу увагу потрібно звернути на те, що володіння криптовалютою не залежить від наявності гаманця, а від наявності приватного ключа. Гаманець є спеціалізованим інструментом, який використовується для взаємодії з блокчейном. Приватний ключ є головним елементом, який визначає володіння криптовалютою.

Кастодіальний гаманець - це додаток, який використовується для зберігання і передачі криптовалют, а його оператор (кастодіан) має доступ до адрес користувачів або приватних ключів. Клієнти кастодіана також повинні пройти процедуру верифікації особистості (KYC). Ці гаманці часто використовуються на централізованих криптобіржах, де всі кошти зберігаються на декількох адресах, керованих біржею, що спрощує торгівлю і зменшує витрати на транзакції всередині платформи. Індивідуальні користувачі рідко використовують кастодіальні гаманці поза спеціалізованими додатками. Однак такі рішення часто використовуються інституціональними інвесторами, які оперують великими сумами. Вони довіряють зберігання своїх криптовалют професійним компаніям з підвищеним рівнем безпеки та іншими додатковими послугами, наприклад, страхуванням [1]. Деякі з відомих кастодіанів: Xapo, BitGo, Gemini і Coinbase Custody.

Основний недолік кастодіальних криптогаманців - можливість для кастодіана отримати доступ до криптоактивів клієнтів. Адміністрація біржі володіє приватними і публічними ключами від адрес, на яких зберігаються криптовалюти клієнтів. Централізоване зберігання великого обсягу коштів робить торгові платформи частою метою хакерських атак. У результаті зламів окремі користувачі можуть позбутися своїх криптовалют. Крім того, криптобіржа - юридична організація, яка зобов'язана підкорятися законодавству і вимогам правоохоронних органів. За їхнім запитом вона може як надавати дані про клієнтів, так і заморожувати їхні кошти в гаманці. Наприклад, у разі санкцій або арешту майна за рішенням суду. Неприємним моментом може також стати відсутність доступу до власних активів під час проведення технічних робіт, адже кастодіан може за необхідності відключити доступ для користувачів.

Некастодіальний криптовалютний гаманець зберігає за власником адреси

повний контроль своїх коштів, оскільки не передає будь-кому їхні приватні ключі. Такий застосунок не може заморожувати кошти користувачів або керувати ними, але водночас не несе відповідальності за їхнє збереження [2].

Зазвичай це додаток, який можна завантажити на ПК, мобільний пристрій або в браузер. Для створення адреси в блокчейні через некастодіальний застосунок не потрібно проходити KYC. Додаткову надійність популярним некастодіальним гаманцям дає публікація їхнього вихідного коду. Це дає змогу незалежним експертам переконатися в тому, що додаток дійсно безпечний. Також такі проєкти часто підтримує ціла спільнота програмістів. Кошти, задіяні в торгівлі або інвестиціях (у тому самому стейкінгу), можете зберігати в кастодіальному гаманці стороннього додатка (наприклад, на біржі). Однак вам слід посилити безпеку профілю, зокрема налаштувати двофакторну автентифікацію. А невикористовувані цифрові активи рекомендовано тримати у власному некастодіальному криптогаманці, доступ до якого на рівні приватного ключа має лише власник.

Усі гаманці поділяються на два види: "гарячі", що існують тільки як цифрові додатки, вони постійно (або дуже часто) підключені до мережі., і "холодні", в основі яких фізичний носій та працюють у режимі офлайн.

Основна перевага гарячих гаманців - це швидкий (або безперервний) зв'язок з блокчейном через інтернет. Користувачі можуть поповнювати баланс або знімати кошти в будь-який момент. Приклад «гарячіх" гаманців:

- MetaMask;
- Trust Wallet;
- binance.com

"Холодні", або апаратні гаманці - це пристрої розміром з картку флеш-пам'яті, в яких криптоактиви зберігаються офлайн. Це дає максимальний захист від хакерських зламів. Для здійснення переказів "холодний" гаманець потрібно синхронізувати з блокчейном через комп'ютер, підключений до інтернету. Приклад "холодних гаманців":

- Ledger;
- Trezor;
- Keepkey

Якщо активи власника зберігаються на біржі, то йому не потрібен приватний ключ, лише логін та пароль від свого акаунта, в інших випадках скоріш за все буде потрібна сід-фраза. Це унікальна послідовність із 12 або 24 слів англійською мовою, яка слугує паролем для відновлення доступу до адреси або її перенесення в інший гаманець. Сід-фраза, як і приватний ключ - тільки для власника адреси, більше їх нікому передавати не можна. Якщо сід-фраза буде

загублена або вкрадена, можна втратити доступ до коштів.

Для забезпечення безпеки власного криптогаманця слід враховувати наступні аспекти [3]:

1. Збереження секретної фрази (сід-фрази): Важливо зберігати секретну фразу у безпечному місці, такому як папір або металева пластина. Не слід повідомляти її нікому і не слід робити цифрові копії, оскільки це може призвести до крадіжки гаманця.

2. Слід уникати фішингу: треба бути обережними з посиланнями, які отримуються електронною поштою або через повідомлення. Треба перевіряти URL-адреси сайтів, перш ніж вводити свої дані, і уникати переходу за сумнівними посиланнями.

3. Слід бути уважними з електронними листами: Не можна передавати конфіденційну інформацію про свій криптогаманець через електронну пошту. Треба бути обережними з запитами, які просять виконати незвичайні дії або надати свої активи.

4. Слід розподіляти власні активи: Не потрібно зберігати всі свої заощадження криптовалюти на одному гаманці. Рекомендовано розподілити свої активи залежно від цілей: використовувати кастодіанський гаманець для швидких операцій, а довгострокове зберігання виконувати на некастодіанському гаманці.

5. Слід налаштувати безпеку облікового запису: якщо використовувати криптобіржу, рекомендовано встановити додаткові заходи безпеки для свого облікового запису, такі як двофакторна автентифікація і підтвердження за допомогою номера телефону або електронної пошти.

Список використаних джерел:

1. Кастодіальні та некастодіальні крипто гаманці. URL: https://forklog.com/cryptorium/chto-takoe-kastodialnye-i-nekastodialnye-kriptokoshelki/.
2. Гаманець для крипто валюти URL: https://www.binance.com/en/blog
3. Носії крипто валют URL: https://corporatefinanceinstitute.com/resources/cryptocurrency/cryptocurrency-wallet/

Ключові слова: криптовалюта, блокчейн, ключ, адреса
Keywords: cryptocurrency, blockchain, key, address

Науковий керівник: доцент Задерейко О.В.

ЗАСОБИ HARDHAT ДЛЯ АВТОМАТИЗОВАНОГО ТЕСТУВАННЯ ПРОГРАМНИХ ЗАСТОСУНКІВ У МЕРЕЖІ БЛОКЧЕЙН

Дацко І. А.

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

Безпека є обов'язковою вимогою до кожного smart-контракту через специфіку роботи блокчейну та Solidity. Є багато різних атак, які можуть бути реалізовані тільки у smart-контрактах. Наприклад, атака reentrancy, найвідомішим прикладом якої є злом фонду DAO, унаслідок якого зловмисники змогли вкрасти криптовалюти на 50 мільйонів доларів [1]. Суть такої атаки полягає у тому, щоб повторно викликати функцію та виводити криптовалюту до того, як smart-контракт встигає оновити баланс користувача. Такої вразливості можна уникнути простим способом, використовуючи у своїх контрактах патерн програмування CEI. CEI розшифровується як check, effect, iteraction. Мета його полягає у правильній послідовності виконання дій будь-якої функції у Smart-контракті, а саме:

1) пройти усі передбачені перевірки;

2) встановити коректні значення змінних, з якими працює сам контракт;

3) змінити дані у самому блокчейні.

Hardhat – це платформа для розробки ПЗ у середовищі Ethereum. Воно складається з різних компонентів для редагування, компіляції, налагодження та розгортання смарт-контрактів, що разом створюють повноцінне середовище для розробки мовою Solidity [2]. Також Hardhat надає можливість розробнику розгорнути свій локальний блокчейн.

Взагалі існує багато тестових блокчейнів, де можна безпечно налагоджувати свої контракти, але для невеликих застосунків легше використовувати свій, локальний блокчейн, який по суті є змодельованою мережею etherium, яка працює локально на машині користувача. Це безпечне та ізольоване середовище, в якому можна тестувати свої контракти. Такий локальний блокчейн має увесь функціонал мережі etherium, від розгортання контрактів до виконання транзакцій. Він створюється з набором попередньо визначених розробником облікових записів із криптовалютою на рахунку, які можна використовувати для взаємодії зі smart-контрактами, що дозволяє тестувати їх за різними сценаріями. Також розробник має повний контроль над своїм блокчейном, тобто може змінювати складність майнінгу, встановлювати власні ліміти газу, щоб перевіряти, як контракт поводить себе за різних умов. Тож Hardhat – це потужний інструмент для розробки, тестування та налагодження своїх контрактів.

Hardhat дозволяє писати розробнику скрипти на JavaScript, які своєю чергою за допомогою декількох бібліотек можуть використовуватись для написання модульних тестів. Одна з таких – Chai – забезпечує користувача чистим синтаксисом для написання тверджень у тестах, що полегшує як саме їх написання, так і читання написаного коду. Вона впроваджує декілька різних шляхів для написання тверджень за допомогою ключових слів assert, should, expect, що надає можливості написання тестів саме так, як зручно розробнику [3]. Суттєво, що Chai можна використовувати з різними платформами для тестування, наприклад, такими як Mocha, яка використовується для написання модульних тестів, використовуючи функції it(), describe(), beforeEach() [4].

У контексті Smart-контрактів автоматизоване тестування використовувати вигідніше з декількох причин:

Автоматичні тести можуть бути виконані швидше та ефективніше, ніж ручні. Вони можуть бути запущені багаторазово і без затримок. У разі ручного тестування Smart-контрактів, воно зазвичай проходить у так званих тестових блокчейнах, які нічим не відрізняються від своїх реальних аналогів, але криптовалюта там тестова. Це чудовий інструмент для тестування роботи функцій, які мають взаємодіяти з іншими Smart-контрактами. При цьому для smoke тестування, тобто для перевірки працездатності основного функціонала автоматичні тести є набагато ефективнішими;

Автоматичне тестування може знизити витрати на тестування Smart-контрактів, оскільки воно не потребує для виконання великої кількості часу та людських ресурсів;

Автоматичне тестування набагато точніше, ніж ручне. Написані тести можна використовувати багато разів, щоб спостерігати за поведінкою функцій за різних умов та виявляти помилки з більшою надійністю.

Список використаних джерел:

1. What is a Re-Entrancy Attack? URL: https://quantstamp.com/blog/what-is-a-re-entrancy-attack
2. Hardhat docs. URL: https://hardhat.org/hardhat-runner/docs/getting-started#overview
3. Chai docs. URL: https://www.chaijs.com/guide/
4. Mocha docs. URL: https://mochajs.org

Ключові слова: Solidity, Hardhat, блокчейн, тестування.

Keywords: Solidity, Hardhat, blockchain, testing.

Науковий керівник: доцент Трофименко О. Г.

Секція 2. ІНФОРМАЦІЙНА БЕЗПЕКА ТА ЗАХИСТ ІНФОРМАЦІЇ: СОЦІАЛЬНИЙ, ПРАВОВИЙ І ТЕХНІЧНИЙ АСПЕКТ

КОМПЛЕКСНИЙ ПІДХІД ДО ВИКЛАДАННЯ ПРОФЕСІЙНИХ ДИСЦИПЛІН ЗІ СПЕЦІАЛЬНОСТІ 125 «КІБЕРБЕЗПЕКА»

Ахмаметьєва Г. В.

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

Останні роки спеціальність 125 «Кібербезпека» набуває все більшої популярності серед молоді, що обумовлено широким впровадженням цифрових технологій в сфери людського життя, починаючи від реєстрації майна в державних і приватних нотаріальних конторах, здійснення електронного документообігу в організаціях будь-якої форми власності до відстеження власних коштів в особистих кабінетах банків, комунальних установ, тощо. З кожним роком обсяг оцифрованих даних зростає за експоненціальним законом, що потребує постійного удосконалення існуючих засобів захисту, впровадження новітніх розробок, здатних впоратися із забезпеченням надійного та захищеного зберігання та обробки конфіденційної інформації.

Звісно, кібербезпека охоплює широкий спектр напрямів, зокрема:

- вивчення існуючих законів і стандартів у сфері безпеки, підготовка заощаджень і пропозицій до законів, здійснення експертизи проектів законів і стандартів;
- організаційні питання з проведення аудитів, виявлення ризиків як шляхом огляду технічного оснащення приміщень, аналізу документації або тестових випробувань системи захисту, введенням політик безпеки, посадових інструкцій та інше;
- впровадження і організація технічного захисту приміщень, територій, розробка проектів і схем пропускного режиму, відеоспостереження, управління доступом з використання апаратних пристроїв;
- використання програмних засобів захисту інформаційного середовища від серверів і систем зберігання даних до персональних комп'ютерів користувачів.

І це не повний перелік можливих аспектів спеціальності «Кібербезпека», зокрема потребують дослідження системи впровадження і управління захистом хмарними ресурсами.

Кожна освітня програма зі спеціальності відповідно до затвердженого стандарту [1] має максимально охопити базові знання і компетенції та поглибити знання здобувачів у вибіркових дисциплінах відповідно до професійної спрямованості відповідних кафедр вищих навчальних закладів. Гарним комплексним підходом до поглиблення професійних навичок здобувачів є впровадження

вибіркових освітніх траєкторій, які містять послідовний ланцюжок спеціальних дисциплін певного напряму. На факультеті кібербезпеки та інформаційних технологій НУ «ОЮА» освітня програма зі спеціальності 125 «Кібербезпека» [2] пропонує три траєкторії:

- «Криптографія та стеганографії» з поглибленим вивченням криптографічних і стеганографічних технологій, особливостей їх побудови і застосування, спрямованих на здійснення шифрування і приховування даних, організацію безпечної передачі конфіденційної інформації;
- блок «Cyberforses» спрямований на вивчення технологій з безпеки комп'ютерних, мобільних та бездротових мереж, тестування на проникнення, виявлення вразливостей інформаційних систем, моніторинг та аналіз інформаційних впливів;
- «Cisco certified networking and security specialist» передбачає вивчення дисциплін за допомогою платформи академії Cisco [3] з можливістю отримати відповідні сертифікати. Блок спрямований на безпечні засади побудови комп'ютерних мереж і мережних з'єднань, аналіз і контроль їх ефективності, хмарну безпеку, безпеку інтернету речей, системи управління доступом.

Кожна з траєкторій має логічно вивірену послідовність навчальних дисциплін, що дозволяє послідовно, базуючись вже на отриманих знаннях, поглиблювати знання і отримувати практичні навички з обраного напряму.

В той же час обов'язкові компоненти освітніх програм хоч і відповідають стандарту, забезпечуючи відповідні компетенції та результати навчання, однак в деяких дисциплінах теми можуть дублюватись, що фактично означає повторення вже пройденого матеріалу, кожна дисципліна існує окремо одна від одної. В тому випадку, коли послідовні дисциплін викладає один лектор, то ще можна сформувати логічно побудовану систему отримання здобувачами знань і навичок, однак дисципліни спеціальності читають різні викладачі, і ось тут логічна послідовність може бути порушена. Здобувачі кафедри кібербезпеки НУ «ОЮА» відзначають, що в навчальних матеріалах недостатньо висвітлено застосування теоретичного базису в практичних ситуаціях професійної сфери.

Зважаючи на виявлені недоліки вважаю доцільним сформулювати принципи при підготовці матеріалів навчальних дисциплін, які дозволили би забезпечити комплексний підхід у викладанні обов'язкових освітніх компонентів.

1. Взаємозв'язок між викладачами з формування програм навчальних дисциплін з метою уникнення дублювання тем.

При формуванні програми навчальної дисципліни необхідною складовою є її погодження з гарантом освітньої програми. Як показує практика, завжди можуть виникати ситуації, коли тема, що розглядається, може бути складовою декількох дисциплін, тому важливо саме між викладачами узгодити, до якої дисципліни доцільно її включити з урахуванням розподілу дисциплін по семестрах, складності сприйняття теми, її необхідності при вивченні інших освітніх

компонентів. Цей принцип вимагає злагодженої співпраці всіх співробітників кафедри і факультету.

2. Практична орієнтованість лабораторних і практичних робіт.

Факультет кібербезпеки та інформаційних технологій НУ «ОЮА» застосовує практичну орієнтованість професійних освітніх компонентів, виділяючи більшість часів саме на практичні і лабораторні роботи. В процесі вивчення спеціалізованих технологій, що застосовуються в професійній діяльності, цей принцип є очевидним. Однак є дисципліни, які закладають теоретичний базис для вивчення подальших компонентів, і тут вже вважаю доцільним звернути увагу на ті практичні ситуації, в яких застосовується та чи інша теорія. Наприклад, при вивченні теорії ризиків доцільно було б розглядати ситуації, які виникали в реальному житті, та аналізувати ризики з використанням різних методів їх оцінки, які застосовують фахівці. Бажано змоделювати процес аудиту здобувачами комп'ютерних лабораторій факультету, звернути увагу на роботу з документацією. В математичних основах криптографії можна давати завдання, безпосередньо пов'язані з використанням математичного апарату, наприклад, визначити параметри лінійного конгруентного генератора, маючи лише чотири послідовних числа, та інші нескладні задачі.

3. Використання навичок, отриманих в інших дисциплінах.

Якщо дублювання тем в різних дисциплінах є небажаним, то спонукання здобувачів застосовувати навички, отримані в попередніх курсах, сприяє необхідності самостійно пригадати вже пройдений матеріал та прокачати свої знання при розв'язанні практичних завдань. Гарним прикладом є поєднання вивчення методів аналізу ризиків, кодування та шифрування з їх програмною реалізацією, що з одного боку дозволяє досконально розібрати принципи функціонування методів, що вивчаються, а з іншого удосконалити навички з програмування. Вважаю необхідним максимально задіяти математичний апарат в процесах моделювання, спеціалізованих задачах зберігання, передачі та обробки інформації. Математична теорія широко може застосовуватися при обробці мультимедійних даних, експертизі і виявленні слідів порушення цілісності цифрових даних.

4. Спрямування здобувачів на самостійне освоєння і використання близькоспоріднених технік з метою підвищення якості і ефективності результатів роботи.

Вважаю необхідним підтримувати прагнення здобувачів до самостійного пошуку матеріалів відповідно до дисципліни, спонукати їх до використання додаткових методів, що значно спрощують і пришвидшують результати їх роботи. Наприклад, застосування методів розподілу і організації часу в процесі навчання, що дозволить ефективно використовувати власний час та вчасно виконувати завдання. Криптографічні асиметричні методи вимагають складних обчислень над багаторозрядними числами, отож слід спонукати здобувачів до використання швидких операцій для програмної реалізації з метою прискорення часу роботи програми. Слід вимагати від здобувачів самостійно обирати і

застосовувати найбільш оптимальні методи для розв'язання задач з обґрунтуванням свого вибору.

5. Організація міждисциплінарних курсових проєктів.

В інших вищих навчальних закладах вже практикуються міждисциплінарні курсові проєкти, зокрема в Київському національному університеті імені Тараса Шевченка кафедра теорії та технології програмування залучає до спільних ІТ-проєктів здобувачів, які поділяються по групах з 5-6 осіб та реалізують більш складну комплексну задачу зі спеціальності. Результати роботи доповідають на захисті та публікують у спеціальному збірнику [4]. На факультеті кібербезпеки та інформаційних технологій НУ «ОЮА» найкращі здобувачі вже залучаються до окремих проектів, однак вважаю, що необхідно залучати максимальну кількість студентів до спільних проєктів з обмеженням у часі виконання та обов'язковим звітуванням щодо виконаної роботи. Це дозволить прокачати їх навички роботи в команді, підвищить відповідальність та рівень самооцінки.

Таким чином, вважаю, що застосування сформульованих принципів спроможне значно підвищити зацікавленість здобувачів, наблизити їх до умов, близьких до реальної роботи в команді в ІТ-компаніях, покращити якість викладання дисциплін, в свою чергу інтерес до розв'язання спеціалізованих задач сприяє більш легкому та ефективному засвоєнню матеріалу.

Список використаних джерел:

1. Стандарт вищої освіти. Перший (бакалаврський) рівень. Галузь знань 12 Інформаційні технології, спеціальність 125 Кібербезпека. Затверджено Наказом МОН України № 1074 від 04.10.2018 р. Режим доступу: https://mon.gov.ua/storage/app/uploads/public/5bb/626/1a8/5bb6261a84776166409164.pdf
2. Освітня програма зі спеціальності 125 Кібербезпека Національного університету «Одеська юридична академія» для здобувачів першого (бакалаврського) рівня вищої освіти. 2022. 19 с.
3. Cisco Networking Academy. Режим доступу: https://www.netacad.com/
4. Програмування: теорія та практика. Збірник матеріалів за результатами ІТ-проєкту міждисциплінарної інтеграції / За редакцією Л.Л. Омельчук, О.М. Ткаченка, О.В. Шишацької. Київ, 2022. 155 с.

Ключові слова: кібербезпека, підходи до викладання дисциплін, комплексний підхід
Keywords: cybersecurity, approaches to teaching disciplines, integrated approach

ВИКОРИСТАННЯ ІНТЕРАКТИВНИХ ОНЛАЙН-ПЛАТФОРМ ДЛЯ ОТРИМАННЯ НАВИЧОК У СФЕРІ КІБЕРБЕЗПЕКИ

Дрига А. В.

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

У сучасному цифровому світі, де велика частина нашого повсякденного життя переноситься в онлайн-простір, захист особистої інформації стає надзвичайно важливим завданням, а попит на фахівців у сфері кібербезпеки зростає з кожним днем. Саме тому проблема розвитку методів навчання навичкам кібербезпеки стає дедалі актуальнішою, і особливої популярності набирають інтерактивні онлайн-майданчики для практики етичного хакінгу. Такі платформи дають змогу не лише отримати знання, а й легально закріпити їх практично.

Найчастіше, такі платформи надають вам доступ до спеціально вразливих машин різної складності, які потрібно зламати. Також вони можуть містити завдання у форматі Capture The Flag (CTF), які зі свого боку можуть мати різну тематику, наприклад криптографія, веб-безпека тощо. Найпопулярнішим ресурсом такого типу є HackTheBox, який виділяється величезною кількістю вразливих машин, що оновлюються щотижня, а також активною спільнотою, що може допомогти у розв'язанні завдань [1]. Також варто відзначити систему рейтингу, яка показує ваш рівень майстерності зі злому машин. Другою за популярністю платформою для вивчення етичного хакінгу є Tryhackme [2]. Величезною перевагою цього сайту є те, що він містить велику кількість навчальних матеріалів за різними напрямками. Два ці сайти мають платну підписку, яка відкриває доступ до деяких машин, але і без неї є можливість випробувати величезну їх кількість. Крім HackTheBox і Tryhackme, існує також і безліч інших подібних платформ, наприклад VulnHub, Root-Me, Pentesterlab та інші, які мають свої унікальні особливості.

Існують також спеціально вразливі веб-сайти, які орієнтовані на експлуатацію популярних веб-вразливостей, наприклад OWASP Top Ten [3]. Такі сайти вимагають встановлення веб-сервера і тих технологій, які використовувалися під час його розробки. Існують віртуальні машини, в яких уже встановлено все необхідне для правильної роботи вразливих сайтів. Прикладом такої машини є Web Security Dojo, який містить у собі такі застосунки:

- Damn Vulnerable;
- Web Application OWASP;
- Mutillidae II;
- WebGoat Insecure;

- Web App Hacme;
- Casino Juice Shop та інші [4].

Крім цього, Web Security Dojo має вбудовані інструменти для проведення тестування на проникнення, такі як Burp Suite, sqlmap, Metasploit, Zed Attack Proxy, а також всю необхідну документацію для вразливих застоссунків.

Для тих, хто хоче попрактикуватися у зворотній розробці додатків (reverse engineering) існують так звані crackme. Це програми невеликого розміру, код яких можна легально проаналізувати. Крім цього, існують keygenme – програми, для яких потрібно написати генератор ключів активації. Існують сайти з прикладами таких завдань, такий як Reversing.Kr [5]. На ньому зібрано безліч crackme, які написані різними мовами програмування та для різних платформ.

Для підготовки к змаганням Capture The Flag під назвою "UA30CTF" було обрано платформу HackTheBox по наступним причинам: велика різноманітність завдань на різні тематики та рівні складності дозволять знайти ті завдання, які тобі дійсно цікаво проходити; завдання в стилі CTF дозволяють потренуватися перед реальними змаганнями з кібербезпеки та зрозуміти, як вони взагалі проходять. Через свою добру репутацію, при наявності високого рейтингу на цьому сайті роботодавці будуть віддавати тобі більше пріоритет. Використання цієї платформи дозволило мінімізувати трудовитрати на організацію змагань та успішно їх провести.

Список використаних джерел:

1. Hack The Box: Hacking Training For The Best. URL: https://www.hackthebox.com/
2. TryHackMe | Cyber Security Training URL: https://tryhackme.com/
3. OWASP Top Ten. URL: https://owasp.org/www-project-top-ten/
4. Web Security Dojo - Maven Security Consulting. URL: https://www.mavensecurity.com/resources/web-security-dojo/
5. Reversing.Kr. URL: http://reversing.kr/

Ключові слова: кібербезпека, хакінг, CTF, веб-безпека, зворотна розробка
Keywords: cyber security, hacking, CTF, web security, reverse engineering

Науковий керівник: доцент Бойко В.Д.

КОМПЛЕКС ПРИНЦИПИВ ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

Чепурна О. Є.
доцент кафедри кібербезпеки
Національного університету «Одеська юридична академія»,

Соловйов А. С.
студент 1 курсу магістратури факультету кібербезпеки
та інформаційних технологій
Національного університету «Одеська юридична академія»,

Захист інформаційної безпеки передбачає захист конфіденційності, цілісності та доступності конфіденційних даних від несанкціонованого доступу, використання, розкриття, перекручення, модифікації або знищення. Ефективний захист інформаційної безпеки вимагає впровадження комплексного набору принципів і практик, які забезпечують конфіденційність, цілісність і доступність інформаційних активів.

Деякі з ключових принципів захисту інформаційної безпеки:

1. Конфіденційність: Забезпечення доступу до конфіденційної інформації лише уповноваженим особам та запобігання несанкціонованому розголошенню або доступу до неї.

2. Цілісність: Забезпечення того, що інформація є точною, повною та послідовною, і що вона не була змінена або знищена без дозволу.

3. Доступність: Забезпечення доступу до інформації уповноваженим особам, коли це необхідно, та запобігання перебоям у доступі чи наявності інформації.

4. Аутентифікація: Перевірка особи осіб, які отримують доступ до інформації або систем, та забезпечення того, що вони мають відповідний рівень повноважень.

5. Авторизація: Визначення рівня доступу, який мають особи до певної інформації або систем, виходячи з їхньої ролі, обов'язків та необхідності знати.

6. Неприпустимість відмови: Забезпечення того, що особи не можуть заперечувати свої дії або діяльність, а також наявність чіткого аудиторського сліду всієї діяльності.

7. Поглиблений захист: Впровадження багаторівневих засобів контролю та заходів безпеки для забезпечення комплексного та надійного захисту.

8. Управління ризиками: Виявлення та оцінка потенційних ризиків і вразливостей, а також впровадження відповідних заходів для їх зменшення або управління ними.

Дотримуючись принципів і впроваджуючи відповідні засоби контролю та

заходи безпеки, організації можуть забезпечити захист своїх чутливих інформаційних активів від різних загроз і ризиків.

Для більш цінних даних, які можуть зберігатися на серверах компаній, або данні які можуть містити секретну або конференційну інформацію, треба підтримувати високий рівень практики з інформаційної безпеки. Такий як:

1. Шифрування: Впровадження методів шифрування для захисту конфіденційних даних під час їх передачі та зберігання.
2. Контроль доступу: Забезпечення доступу до конфіденційної інформації, систем та об'єктів лише уповноваженим особам.
3. Реагування на інциденти: Розробка та впровадження плану реагування на інциденти для швидкого виявлення, локалізації та відновлення після інцидентів безпеки.
4. Навчання з питань безпеки: Проведення регулярних тренінгів з підвищення обізнаності працівників про безпеку, щоб вони знали про ризики та найкращі практики безпеки.
5. Регулярне оцінювання безпеки: Проведення регулярних оцінок безпеки для виявлення вразливостей і ризиків та впровадження відповідних заходів безпеки.
6. Дотримання норм і стандартів: Забезпечення відповідності чинному законодавству, нормативним актам і галузевим стандартам, пов'язаним з інформаційною безпекою.
7. Постійний моніторинг та вдосконалення: Постійний моніторинг та вдосконалення засобів контролю та заходів безпеки, щоб забезпечити їхню ефективність у боротьбі із загрозами та ризиками, що постійно змінюються.

Впровадження цих практик дуже сильно допоможе у підсиленні та усуненню несанкціонованого витоку інформації. Вона буде захищена для люде які не мають дозволу до ознайомлення з документами або іншою інформацією, та завжди буде інформація хто та коли мав контакт з інформацією.

Але основним захистом інформації ці її шифрування. Завдяки цьому данні можна краще сховати, без ключа її неможливо прочитати, або редагувати. Для забезпечення безпеки при шифруванні варто дотримуватися таких речей як:

1. Надійні алгоритми шифрування: Алгоритми шифрування - це математичні формули, які використовуються для перетворення відкритого тексту в зашифрований. Надійні алгоритми шифрування використовують складні математичні функції і призначені для того, щоб протистояти атакам хакерів та інших зловмисників. Прикладами стійких алгоритмів шифрування є Advanced Encryption Standard (AES), Rivest-Shamir-Adleman (RSA) та Elliptic Curve Cryptography (ECC).

2. Управління ключами: Ключі шифрування використовуються для шифрування та розшифрування даних. Дуже важливо захистити секретність і цілісність цих ключів. Управління ключами передбачає безпечне створення, зберігання, розповсюдження та відкликання ключів. Практики управління ключами також включають ротацію ключів, резервне копіювання ключів та депонування ключів.

3. Автентифікація та цілісність: Одного лише шифрування недостатньо для забезпечення інформаційної безпеки. Механізми автентифікації та цілісності необхідні для того, щоб гарантувати, що зашифровані дані не будуть змінені або підроблені, і що доступ до них матимуть лише уповноважені особи. Механізми автентифікації включають цифрові підписи та коди автентифікації повідомлень (MAC).

4. Використання безпечних протоколів: Шифрування має здійснюватися з використанням безпечних протоколів для захисту від атак, які використовують вразливості мережевих протоколів. Прикладами безпечних протоколів є протокол захищених сокетів (SSL) та протокол безпеки транспортного рівня (TLS).

5. Дотримання норм і стандартів: Організації повинні дотримуватися чинних законів, нормативних актів і галузевих стандартів, що стосуються шифрування інформації. Прикладами таких стандартів є Федеральний стандарт обробки інформації (FIPS) та Стандарт безпеки даних індустрії платіжних карток (PCI-DSS).

Дотримуючись цих моментів можна забезпечити конфіденційність та цілісність файлів. Але шифрування не зможе повність захистити інформацію, тому треба використовувати повний комплекс необхідних для захисту інформації, такі як контроль доступу, та реагування на інциденти з витоками.

Шифрування інформації є важливим інструментом для забезпечення конфіденційності та цілісності даних. Шифрування забезпечує надійний захист від атаки зловмисників, або зміни конфіденційної інформації. Але треба зазначити що без комплексів захисту, лише одне шифрування не зможе дати сто відсотковий захист даних. Тому організації та самі користувачі повинні постійно оновлювати та контролювати свої методи шифрування. Дотримуючись таких правил можна зміцнити свою інформаційну безпеку від різних загроз та ризиків.

Список використаних джерел:

1. DataMI. URL: https://datami.ua/informatsijna-bezpeka-vidi-zagroz-i-metodi-yih-usunennya/
2. IITD. URL: https://iitd.com.ua/shifruvannja-ta-zahist-baz-danih/

Ключові слова: безпека, інформаційна безпека, шифрування, заходи, алгоритми, використання, зловмисники, документи, інформація

Keywords: security, information security, encryption, measures, algorithms, use, attackers, documents, information

ПРИВАТНІСТЬ ТА ЗАХИСТ ПЕРСОНАЛЬНИХ ДАНИХ

Панкін Н.М.

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

У сучасну цифрову епоху особисті дані цінніші, ніж будь-коли раніше. Їх можна використовувати для відстеження переміщень, покупок, інтересів і навіть наших думок. Ці дані можуть бути використані для націлювання реклами, для прийняття рішень щодо кредитоспроможності та навіть для впливу на політичні погляди.

Як наслідок, зараз дуже важливо захищати свої персональні дані. Користувачам потрібно знати, як збираються та використовуються їх дані, і які потрібно вжити заходи, щоб захистити їх від несанкціонованого доступу [1].

Персональні дані - це будь-яка інформація, яка може бути використана для ідентифікації особи. Вона може включати особисте ім’я, адресу, номер телефону, електронну адресу, номер соціального страхування та номер кредитної картки [2].

Конфіденційність - це поняття, що описує право та стан особи, яке визначають, що їхні персональні дані та приватна інформація є конфіденційними та повинні бути захищені від незаконного доступу, використання або розголошення. Це означає, що особа має контроль над тим, хто має доступ до її особистих даних і як вони використовуються [3].

Конфіденційність включає в себе різні аспекти, такі як:

1. Збереження приватності особистих даних: Це означає, що персональні дані, такі як ім'я, адреса, номери телефонів, електронна пошта, фінансова інформація, медичні записи та інша особиста інформація, мають бути захищені від крадіжки і не повинні бути доступні неправомірним особам.

2. Контроль над використанням даних: Особа повинна мати право контролювати, як її персональні дані збираються і використовуються. Вона може визначати, кому дозволяє доступ до своїх даних і з якою метою вони використовуються.

3. Захист від незаконного доступу: Конфіденційність передбачає використання заходів безпеки для запобігання несанкціонованого доступу до особистих даних. Це може включати використання паролів, шифрування даних, фізичний захист і контроль доступу до систем та інші технічні і організаційні заходи.

4. Дотримання правових норм і регуляцій: У багатьох юрисдикціях існують закони та норми, що регулюють збір, використання та розголошення персональних даних. Конфіденційність передбачає дотримання цих правових норм і забезпечення відповідності з ними.

Важливо пам'ятати, що конфіденційність не обмежується лише цифровими даними, але також охоплює будь-яку форму особистої інформації, незалежно від форми її збереження або передачі. Захист конфіденційності є важливим аспектом дотримання приватності і особистих прав кожної особи.

Персональні дані можна збирати різними способами. Їх можна отримати, коли створюється обліковий запис в Інтернеті, коли робиться купівля або використовуються платформи соціальних мереж.

Персональні дані також можна збирати за допомогою стеження. Уряди та підприємства все частіше використовують технології спостереження для збору даних про пересування громадян, їх комунікації та іншу діяльність.

Персональні дані також можуть бути використані для скоєння злочинів. Зловмисники можуть використовувати особисті дані, щоб відкривати нові рахунки на вкрадене ім’я, а хакери можуть використовувати особисті дані, щоб отримати доступ до фінансових рахунків.

Необхідно зробити наступні дії, щоб захистити свої особисті дані [4]:

- треба бути обережними з інформацією, яка публікується від імені користувача в Інтернеті;
- треба використовувати надійні паролі та регулярно їх змінювати;
- треба слідкувати за тим, якє програмне забезпечення завантажується на комп’ютер користувача та які веб-сайти відвідуються;
- треба використовувати VPN, коли використовується загальнодоступні бездротові мережі Wi-Fi;
- треба бути в курсі налаштувань конфіденційності власних акаунтів у соціальних мережах;
- треба регулярно переглядати власний кредитний звіт на наявність ознак шахрайства.

У сучасну цифрову епоху як ніколи важливо захищати особисті дані. Кожному користувачу потрібно знати, як збираються та використовуються його дані, і які заходи необхідно вжити для захисту їх від несанкціонованого доступу. Дотримуючись наведених вище порад, користувачі зможуть підвищити захист своїх особистих даних їх від тих, хто може використовуватиме їх для нанесення будь-якої шкоди.

Список використаних джерел:

1. Що таке приватність? URL: https://khpg.org/1604922460
2. Персональні дані: використання, захист і відповідальність. URL: https://jurliga.ligazakon.net/news/201367_personaln-dan-vikoristannya-zakhist--vdpovdalnst---shcho-potrbno-znati

3. Брижко, В. М., В. Г. Пилипчук. Приватність, конфіденційність та безпека персональних даних. Інформація і право №1(32), 2020, С. 33-46.
4. 5 шляхів захисту особистих даних в Інтернеті. URL: https://beetroot.academy/blog/general/5-shlyahiv-zahistu-osobistih-danih-vinterneti
5. Безпека в Інтернеті, що потрібно знати. URL: https://pon.org.ua/novyny/5427-bezpeka-v-nternet-scho-potrbno-znati.html

Ключові слова: персональні дані, конфіденційність, приватність особистих даних, захист
Keywords: personal data, privacy, privacy of personal data, protection of personal data

Науковий керівник: доцент Задерейко О.В.

КІБЕРБЕЗПЕКА ТА ВПЛИВ ЇЇ НА ВІЙСЬКОВИЙ СТАН

Шагалін К.А.

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

У сучасному світі, де цифрові технології проникли в усі сфери життя, кібербезпека стає критично важливим аспектом для захисту державної безпеки. Військовий стан є особливо вразливим у контексті кібератак, оскільки успішність операцій та захист інформації відіграють ключову роль у забезпеченні ефективності та успіху військових дій. Розглянемо, як кібербезпека впливає на військовий стан і які заходи можуть бути прийняті для його поліпшення [1, 2].

Захист критичної інфраструктури: Військовий стан залежить від роботи критично важливих систем, таких як комунікаційні мережі, електронна пошта, системи управління військовим обладнанням та інші. Кібербезпека забезпечує захист цих систем від кібератак, спрямованих на зниження ефективності чи порушення нормального функціонування. Шляхи досягнення цього полягають у використанні захисних технологій, шифруванні даних, двофакторній аутентифікації та регулярному аудиті систем безпеки.

Кіберрозвідка противника: Кібербезпека допомагає військовим силам отримувати доступ до інформації про противника та його військові можливості. Шляхи досягнення цього включають проведення кіберрозвідки, злам та збір інформації з ворожих систем, інтернет-підприємств та соціальних медіа. Ця інформація дозволяє виявити противника, оцінити його наміри та дії, що допомагає приймати обґрунтовані стратегічні рішення та реагувати на загрози.

Кібератаки та кібервійна: Кібератаки можуть бути важливим інструментом у військових операціях, що впливають на військовий стан. Різні види кібератак включають атаки на комунікаційні мережі, знищення даних, відмову в обслуговуванні, розповсюдження дезінформації та інші. Постійний моніторинг та

виявлення кіберзагроз, розробка захисних стратегій та практик, а також підвищення рівня кіберзахисту важливі для запобігання та вчасні реагування на кібератаки.

Обмін інформацією та співпраця: Кібербезпека вимагає співпраці та обміну інформацією між військовими структурами та організаціями. Важливо забезпечити обмін даними щодо кіберзагроз та кібератак, таким чином створюючи спільну базу знань і досвіду. Це також сприяє розвитку кіберзахисту та прогнозуванню майбутніх загроз.

Безпека й захист у сфері кіберпростору є невід'ємною складовою військової доктрини багатьох країн. Досягнення військової переваги включає не тільки фізичну силу, але й володіння захист кіберпростору [3]. Основні аспекти, що впливають на військовий стан у контексті кібербезпеки, включають [4]:

1. Навчання та підготовка: Військовим структурам необхідно забезпечувати високий рівень навчання та підготовки щодо кібербезпеки. Це охоплює тренування персоналу в розпізнаванні і виявленні потенційних кіберзагроз, використання захисних технологій, поведінку в мережі та реагування на інциденти. Регулярні навчальні програми та симуляційні тренування допомагають військовим силам бути готовими до викликів у кіберпросторі.

2. Кіберінтелект: Розвиток кіберінтелекту є важливим засобом для підвищення кібербезпеки військового стану. Кіберінтелект охоплює збір, аналіз та інтерпретацію інформації про кіберзагрози та дії противників. Це допомагає усунути вразливості, передбачити можливі атаки та прийняти заходи щодо попередження загроз.

3. Міжнародне співробітництво: Забезпечення кібербезпеки вимагає співпраці та обміну інформацією між державами. Міжнародне співробітництво дозволяє об'єднати зусилля у виявленні, протидії та розслідуванні кібератак. Розробка спільних норм та стандартів для кібербезпеки сприяє створенню стійких та безпечних кіберсистем у військовій сфері.

4. Стійкість до кібератак: Важливим аспектом кібербезпеки військового стану є створення стійких кіберсистем, які можуть витримати атаки та продовжувати свою роботу. Це охоплює використання передових технологій шифрування, механізмів виявлення та реагування на аномальну активність, а також резервне копіювання та відновлення даних.

5. Кіберзахист військових пристроїв та обладнання: Забезпечення кібербезпеки вимагає захисту військових пристроїв та обладнання, таких як бойові системи, беспілотні апарати, супутникові системи та інші. Використання захищених протоколів зв'язку, апаратного забезпечення з вбудованими захисними

механізмами та регулярне оновлення програмного забезпечення є важливими кроками для запобігання злому та зловживанню.

6. Кібергібридні загрози: В сучасному військовому стані постійно зростає роль кібергібридних загроз, які поєднують кібератаки з інформаційною воєнною діяльністю та іншими нестандартними методами. Запобігання та виявлення таких загроз вимагає комплексного аналізу та інтеграції кіберінтелекту, інформаційної воєнної діяльності та фізичної безпеки.

7. Експертна кадрова підтримка: Розвиток кібербезпеки військового стану потребує наявності кваліфікованих кадрів, здатних аналізувати, виявляти та вирішувати кіберзагрози. Залучення експертів з кібербезпеки, проведення програм підготовки та створення кар'єрних можливостей у цій сфері є важливими кроками для забезпечення ефективності та сталості кібербезпеки.

Комплексний підхід до кібербезпеки військового стану передбачає поєднання захисних, превентивних та реагувальних заходів. Забезпечення стійкої кібербезпеки стає невід'ємною складовою стратегії військових сил у сучасному світі [5, 6].

Загальність цих заходів полягає в поєднанні технічних, організаційних та людських аспектів кібербезпеки. Співробітництво між державними, військовими та приватними структурами, розробка стандартів та нормативних актів, постійне оновлення та підвищення кваліфікації персоналу є ключовими складовими забезпечення успішної кібербезпеки військового стану.

Отже, кібербезпека має великий вплив на військовий стан, забезпечуючи захист критичних інфраструктур, забезпечуючи доступ до інформації про противника, запобігаючи та відповідаючи на кібератаки та сприяючи співпраці та обміну інформацією. Розвиток кібербезпеки є невід'ємною частиною підвищення військової ефективності та безпеки у сучасному світі.

Список використаних джерел:

1. Кібербезпека: виклики і загрози. URL: https://www.ukrinform.ua/rubric-other_news/3394912-kiberbezpeka-vikliki-i-zagrozi.html
2. Кібербезпека в армії: проблеми та рішення. URL: https://www.nato.int/nato_ukraine/uk/news_181689.htm
3. Кібербезпека військових систем: поточний стан та виклики. URL: https://www.defence-ua.com/index.php/analitika/6549-kiberbezpeka_vijskovikh_sistem_potochnij_stan_ta_vikliki
4. Кібератаки на військові системи: тенденції та наслідки. URL: https://www.cybersecurity.com.ua/kiberatakyy-na-vijskovi-systemy-tendentsiyi-ta-naslidky/

5. Кібербезпека та воєнна безпека: взаємозв'язок і виклики. URL: https://www.nisproject.org.ua/news/kiberbezpeka-ta-vojenna-bezpeka-vzajemozvjazok-i-vikliki
6. Кібербезпека військових даних: важливість та заходи захисту. URL: https://www.cyberdefensemagazine.com.ua/kiberbezpeka-vijskovikh-danikh-vazhlivist-ta-zakhodi-zakhystu/

Ключові слова: кібербезпека, військовий стан, кібератаки, захист кіберпростору, потенційні кіберзагрози, кіберінтелект, кіберсистеми

Keywords: cybersecurity, martial law, cyber attacks, cyberspace Protection, potential cyber threats, cyber intelligence, cyber systems

Науковий керівник: доцент Задерейко О. В.

OSINT – БАЗОВІ ЗНАННЯ У РОЗВІДЦІ

Задерейко О.В.

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

Ханов Б. В.

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

OSINT (відкритий збір інформації) є методом збору та аналізу інформації, яка є загальнодоступною для громадськості. Використання OSINT дозволяє отримати цінні дані про особу, організацію, подію або будь-який інший об'єкт. Цей підхід широко застосовується у сферах кібербезпеки, розвідки, правоохоронної діяльності та інших сферах [1].

Один з головних аспектів OSINT - це пошук інформації в Інтернеті. Це може включати перегляд веб-сайтів, соціальних мереж, форумів, блогів та інших джерел інформації. Різноманітність даних, що можна знайти за допомогою OSINT, включає особисті дані, адреси, номери телефонів, електронні адреси, зображення, новини та багато іншого [2].

Основні принципи OSINT базуються на активному та пасивному зборі інформації. Активний збір включає взаємодію з джерелами інформації шляхом використання запитів, веб-сканування та інших технік. Пасивний збір полягає у збиранні інформації, яка вже доступна публічно, без прямого контакту з джерелом [3].

Один з найбільш важливих аспектів OSINT – аналіз зібраної інформації. Це

включає оцінку достовірності та надійності джерел, перевірку фактів, зв'язків та залежностей. Аналіз результатів OSINT може надати важливі висновки та допомогти прийняти рішення на основі отриманих даних [4].

Застосування OSINT може бути досить різноманітним. В сфері кібербезпеки OSINT може використовуватися для виявлення потенційних загроз, ідентифікації шкідливого програмного забезпечення та вразливостей, аналізу активності зловмисників та виявлення кібератак. Збір і аналіз публічної інформації може допомогти виявити зловмисників, їх методи та мотивацію, а також розуміти їхні цілі та наміри [4].

У сфері розвідки OSINT є важливим інструментом для збору розвідувальної інформації про потенційних противників. Він дозволяє отримати важливі дані про зовнішній світ, такі як політична ситуація, військові дії, геополітичні конфлікти, соціальні настрої тощо. Ця інформація може бути використана для розробки стратегії безпеки та прийняття політичних рішень.

У правоохоронній діяльності OSINT може допомогти в розслідуванні злочинів. Збір публічної інформації про підозрюваних, свідків, місця злочину та інші важливі деталі може допомогти правоохоронним органам отримати підстави для проведення детального розслідування та затримання злочинців [6].

OSINT також може бути використана для моніторингу і раннього виявлення загроз у сфері бізнесу та політики. Шляхом аналізу публічної інформації про компанії, ринки, політичні події та інших факторів, можна виявити потенційні ризики, зробити прогнози та прийняти стратегічні рішення.

Основні етапи OSINT:

1. Складання плану дослідження або визначення мети.

2. Підготовка обладнання та програм, необхідних для вирішення поставленого завдання.

3. Виконання пошуку за всіма доступними ідентифікаторами.

4. Збір інформації.

5. Аналіз отриманих даних.

6. Підготовка висновку і результатів.

7. Архівування або очищення обладнання, на якому виконувався пошук.

Основні етичні принципи OSINT:

1. Використання відповідно до законодавства і в легальному полі.

2. Використання тільки для вирішення поставлених завдань.

3. Нерозголошення даних, знайдених за допомогою OSINT.

4. Не нанесення шкоди за допомогою OSINT.

5. Не використання OSINT в злочинній діяльності.

Як результат OSINT дає змогу користувачу зберегти час і грошові витрати під час пошуку необхідної інформації, а вивчені інструменти і техніки будуть корисні за межами професійної діяльності. Нарешті, OSINT може бути важливим інструментом для індивідуального захисту.

Отже, технології OSINT є потужним інструментом для збору, аналізу та використання відкритої інформації з різних джерел. Завдяки доступу до соціальних мереж, новинних сайтів, публічних баз даних та інших ресурсів, можна отримати цінні уявлення та інсайти щодо подій, осіб, організацій та ситуацій.

Основний принцип OSINT полягає в систематичному і збалансованому підході до збору інформації, перевірці її достовірності та аналізу з різних джерел. Застосування OSINT може бути широким - від сери кібербезпеки і розвідки до бізнес-аналітики та правоохоронних органів.

Оскільки ландшафт інформаційної безпеки постійно змінюється, важливо постійно оновлювати свої знання про OSINT та використовувати сучасні інструменти і методи для максимально ефективного збору та аналізу відкритої інформації.

Загалом, OSINT відіграє значну роль у сучасному світі, допомагаючи в різних галузях роботи, які потребують доступу до відкритої інформації.

Список використаних джерел:

1. OSINT Framework. URL: https://osintframework.com/
2. Bellingcat. URL: https://www.bellingcat.com/
3. Intelligence Fusion. URL: https://www.intelligencefusion.co.uk/
4. OSINT Curious Project. URL: https://osintcurio.us/
5. European Union Intelligence and Situation Centre. URL: https://www.intelligence.europa.eu/
6. OSINT Techniques. URL: https://osinttechniques.com/

Ключові слова: OSINT, технології, інструменти, потенційні загрози, відкриті джерела інформації
Keywords: OSINT, technologies, tools, potential threat detection, open sources of information

МОДЕЛЮВАННЯ І АНАЛІЗ ЗАГРОЗ ІНФОРМАЦІЙНИМ ОБ'ЄКТАМ БЮРО ПЕРЕКЛАДІВ

Лабур Я.А.

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

Моделювання і аналіз загроз інформаційним об'єктам бюро перекладів є важливим аспектом забезпечення безпеки інформаційних систем. У зв'язку зі зростаючою кількістю кібератак і поширенням шкідливих програм, компанії, які

займаються перекладом, повинні бути готові до виявлення, оцінки та запобігання потенційним загрозам.

Перш за все, для моделювання і аналізу загроз інформаційним об'єктам бюро перекладів необхідно провести оцінку потенційних загроз та визначити вразливості існуючих систем. Цей процес можна розподілити на кілька етапів [1].

Починаючи зі збору інформації, слід дослідити типи загроз, з якими може стикнутися бюро перекладів. Це можуть бути такі загрози, як віруси, злам, крадіжка даних, фішинг атаки тощо. Дослідження цих загроз дозволить визначити потенційні точки вразливості та визначити, які ресурси можуть бути під загрозою.

Далі, необхідно провести оцінку вразливостей інформаційних систем бюро перекладів. Це можна зробити шляхом перевірки систем на виявлення слабких місць, які можуть бути використані зловмисниками для доступу до конфіденційної інформації. До цього етапу можна включити проведення тестів на проникнення, щоб виявити, наскільки системи вразливі до потенційних атак [2].

Після виявлення загроз і вразливостей, можна перейти до розробки моделей загроз. Це означає створення сценаріїв можливих атак і визначення наслідків цих атак на роботу бюро перекладів. Моделі загроз допоможуть визначити, які заходи безпеки потрібно впровадити для запобігання цим загрозам.

Наступним кроком є розробка плану заходів забезпечення безпеки. Цей план повинен включати рекомендації з щодо політики безпеки, шифрування даних, використання багаторівневої аутентифікації, регулярне резервне копіювання даних та інші заходи безпеки, які допоможуть зменшити ризик загроз інформаційним об'єктам бюро перекладів.

Останнім етапом є впровадження та моніторинг заходів безпеки. Заходи безпеки, розроблені на попередньому етапі, повинні бути реалізовані в роботі бюро перекладів. Крім того, необхідно забезпечити постійний моніторинг та оновлення заходів безпеки з метою виявлення нових загроз і вирішення існуючих проблем безпеки [3].

Загалом, моделювання і аналіз загроз інформаційним об'єктам бюро перекладів є важливим процесом для забезпечення безпеки даних та інформаційних систем. Цей підхід допоможе виявити потенційні загрози, вразливості систем, розробити відповідні заходи безпеки та забезпечити безперебійну роботу бюро перекладів.

Крім вищевказаного, важливо також враховувати інші аспекти моделювання і аналізу загроз інформаційним об'єктам бюро перекладів, а саме [4]:

1. Оцінка ризиків: Під час аналізу загроз необхідно оцінювати ризики, пов'язані з кожною загрозою. Це допоможе визначити пріоритетність заходів

безпеки і розподілити обмежені ресурси відповідно до важливості ризиків.

2. Законодавчі вимоги: Бюро перекладів повинно враховувати відповідні законодавчі вимоги щодо захисту конфіденційності, цілісності та доступності інформації. Наприклад, якщо бюро працює з особистими даними клієнтів, необхідно дотримуватись вимог Загального регламенту про захист персональних даних (GDPR) або інших аналогічних нормативних актів.

3. Культура безпеки: Важливо створити культуру безпеки серед співробітників бюро перекладів. Проведення навчань та свідомої освіти щодо кібербезпеки допоможе зрозуміти працівникам загрози та методи їх запобігання. Регулярне нагадування про правила безпеки та встановлення внутрішніх процедур також сприятимуть підвищенню рівня захисту інформації.

4. Моніторинг та реагування: Бюро перекладів повинно мати механізми моніторингу та реагування на загрози. Це включає використання системи інцидентного управління, щоб швидко виявляти та реагувати на події безпеки, а також забезпечення резервного копіювання даних та відновлення після інцидентів.

5. Зовнішні аудити безпеки: Регулярне проведення зовнішніх аудитів безпеки допоможе перевірити ефективність заходів безпеки та виявити можливі проблеми. Експерти з кібербезпеки можуть допомогти виявити вразливості та розробити рекомендації для подальшого поліпшення безпеки.

Всі ці аспекти допоможуть створити комплексний підхід до моделювання і аналізу загроз інформаційним об'єктам бюро перекладів. Врахування цих факторів і впровадження відповідних заходів безпеки дозволять забезпечити надійний захист інформації та уникнути можливих проблем, пов'язаних з кібербезпекою.

Задля більш повного опису моделювання і аналізу загроз інформаційним об'єктам бюро перекладів, розглянемо деякі конкретні аспекти цього процесу [5].

1. Системи інформаційної безпеки: Бюро перекладів повинно розглядати впровадження спеціалізованих систем інформаційної безпеки. Ці системи можуть включати в себе механізми виявлення та запобігання вторгнень, системи керування ідентифікацією та доступом, системи моніторингу мережі та системи контролю доступу до даних. Вибір конкретних систем повинен відповідати потребам бюро перекладів і його інфраструктурі.

2. Резервне копіювання даних: Важливим аспектом безпеки інформаційних об'єктів є регулярне резервне копіювання даних. Це забезпечує можливість відновлення даних у випадку їх втрати або пошкодження внаслідок кібератаки або іншого інциденту. Резервні копії повинні зберігатися в безпечному місці і

періодично перевірятися на цілісність.

3. Управління правами доступу: Ефективне управління правами доступу до інформації є важливим елементом безпеки. Бюро перекладів повинно встановити чіткі правила щодо надання доступу до різних рівнів інформації та забезпечити контроль за цим доступом. Використання ролей та обмеження прав доступу допоможе запобігти несанкціонованій модифікації або втраті даних.

4. Оновлення та патчі: Бюро перекладів повинно забезпечувати регулярне оновлення програмного забезпечення та встановлення важливих патчів безпеки. Це дозволить закрити вразливості, які можуть бути використані зловмисниками для зламу системи.

5. Залучення зовнішніх експертів: У разі потреби бюро перекладів може залучити зовнішніх експертів з кібербезпеки для проведення аудиту безпеки, пенетраційного тестування або оцінки загроз. Це допоможе отримати об'єктивну оцінку системи безпеки та виявити можливі уразливості, які не були виявлені внутрішніми ресурсами.

Врахування цих додаткових аспектів допоможе бюро перекладів створити комплексний і ефективний підхід до моделювання і аналізу загроз власним інформаційним об'єктам. При цьому важливо постійно оновлювати та вдосконалювати заходи безпеки, оскільки сучасни тренди кібербезпеки постійно еволюціонують.

Список використаних джерел:

1. Cybersecurity: Challenges and Threats. URL: https://www.ukrinform.ua/rubric-other_news/3394912-cybersecurity-challenges-and-threats.html
2. Cybersecurity in the Military: Issues and Solutions. URL: https://www.nato.int/nato_ukraine/en/news_181689.htm
3. Cybersecurity of Military Systems: Current Status and Challenges. URL: https://www.defence-ua.com/index.php/analitika/6549-cybersecurity-of-military-systems-current-status-and-challenges
4. Cyber Attacks on Military Systems: Trends and Consequences. URL: https://www.cybersecurity.com.ua/cyber-attacks-on-military-systems-trends-and-consequences/
5. Cybersecurity and National Security: Interconnection and Challenges. URL: https://www.nisproject.org.ua/news/cybersecurity-and-national-security-interconnection-and-challenges.

Ключові слова: моделювання загроз, аналіз загроз, інформаційні об'єкти, бюро перекладів, безпека інформації

Keywords: threat modeling, threat analysis, information assets, translation bureau, information security

Науковий керівник: доцент Задерейко О.В.

АДИТИВНО-МУЛЬТИПЛІКАТИВНІ ПЕРЕТВОРЕННЯ В СИСТЕМАХ ШИФРУВАННЯ

Матвійчук І.

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

Підвищення безпеки інформаційного обміну в сучасних системах автоматизованого управління складними технологічними і бізнесовими процесами передбачає широке втілення криптографічних засобів захисту даних. Постійне зростання об'єму інформаційних потоків в таких системах вимагає пошуку нових більш ефективних способів шифрування. З урахуванням цього, сьогодні актуальною задачею є розроблення швидкісних обчислювально стійких способів поточного шифрування, які допускають компактну і надійну програмну реалізацію.

В сучасних автоматизованих системах для захисту великих об'ємів даних використовують методи блочного симетричного шифрування, але це складний обчислювальний процес, що вимагає значних обчислювальних ресурсів і помітно уповільнює інформаційний обмін між його суб'єктами.

Рішенням означеної проблеми може бути розроблення криптографічних ситем поточного шифрування, які забезпечують більшу продуктивність, про що свідчить їх включення до складу таких стандартів як GSM, GPRS, UMTS тощо.

Інтерес до поточних шифрів пов'язаний з роботою Клода Шеннона, присвяченій аналізу одноразових гамм-блокнотів, тобто іменованих шифром Вернама [1]. Пізніше, ця концепція лягла в основу сучасних поточних шифрів, алгоритми яких намагається імітувати потік випадкових, рівномірно розподілених чисел, джерелом яих є комп'ютерні рекурентні алгоритми. Такі шифруючі послідовності часто іменують псевдовипадковими послідовностями (ПВП), або шифруючими гамами. Враховуючи переваги поточного шифрування в системах комунікації, що працюють в реальному маштабі часу, зусилля фахівців зосереджені на створені алгоритів, які забезпечують високий ступень рівномірності розподілення ймовірностей псевдовипадкових чисел у складі шифруючих гам. При цьому, стоїть задача зробити їх максимально простими.

Шифруюча гама вважається випадковою, якщо про бінарні символи у її складі не можна зробити ніяких прогнозів, і для них не можна знайти ніякого простого описання. Але якщо шифруюча гамма може генеруватися ефективно, то таке просте описання існує [2].

Ключ до поточного шифру повинен бути утворений таким складним способом, щоб було абсолютно неможливо його відновити. Рівень випадковості

послідовності визначають за допомогою тестів, які здатні виявити її статистичні нерегулярності.

Для випробувань розроблених алгоритмів на стійкість застосовують різні статистичні тести "на випадковість", які за допомогою достатньо нескладних обчислень дають відповідь на питання про відповідність створеної послідовності заданим вимогам [3].

На практиці Що стосується вимог, які ставляться до формувачів криптографічних ПВП, то вони повинні бути непередбачуваними, якщо кожен їх елемент буде взято з рівноймовірного розподілу.

Практично всі генератори псевдовипадкових чисел, що використовують в криптографії можуть бути схематично представлені як функція від двох змінних $f(s,i)$, де i – номер утвореного випадкового бінарного слова фіксованої довжини, а s – параметр або секретний ключ, який часто називають «зерном». Зазвичай, таке бінарне слово, обирається з випадкового джерела.

Будь-який генератор повинен забезпечити, щоб:

- по-перше, створена псевдовипадкова послідовність виглядала як статистично абсолютно випадкова;
- по-друге, знання якої-небудь початкової частини послідовності не повинно дозволяти передбачити наступний біт цієї послідовності з вірогідністю, що відрізняється від 0,5.

Зазвичай, алгоритм поточного шифрування уявляє собою набір лінійних операцій, що утворюють рекурентну послідовність чисел. Оскільки таким способом забезпечити необхідну рівномірність розподілення чисел на виході програмного генератора важко, доводиться вносити в алгоритм додаткові нелінійні перетворення. Найбільший інтерес у рамках такого підходу викликає мультиплікативно-адитивне перетворення, описане в [4]. Суть такого перетворення зводиться до того, що значення вхідних байт перетворюються шляхом поетапного застосування до них операцій додавання за модулем 28 (модулем 256), що позначається символом $\oplus$, і операцій множення по модулю 28 – 1 (по модулю 255), що позначається символом $\otimes$.

Обидві ці операції не підпорядковуються законам дистрибутивності.

$$a \oplus (b \otimes c) \neq (a \oplus b) \otimes (a \oplus c)$$

і асоціативності

$$a \otimes (b \otimes c) \neq (a \otimes b) \otimes c$$

або

$$a \oplus (b \oplus c) \neq (a \oplus b) \oplus c$$

Використання таких неузгоджених операцій у комбінації забезпечує високу

складність перетворення вхідного значення, що, у свою чергу, ускладнює криптоаналіз шифру загалом.

Висновки. Генератор шифруючої гами повинен бути виконаний таким чином, щоб вид сформованої ним послідовності максимально залежав від інформації, яка міститься в ключі, і при цьому виключалася б можливість отримання однакових послідовностей з різних ключів. Використання додаткових адитивно-мультипликативних перетворень на виході простого рекурентного алгоритма, наприклад, лінійного конгруентного або іншнго подібного алгоритму, суттєво підвищує стійкість поточного шифрування без помітних додаткових витрат обчислювальних ресурсів.

Список використаних джерел:

1. Bruce Schneier. Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd Edition 662. Pages 1995.
2. Фільштінський В. А. Математичні основи криптографії: конспект лекцій / В.А. Фільштінський, А.В. Бережний. Суми: Сумський державний університет, 2011. 138 с.
3. Knuth, D. E. The Art of Computer Programming. Volume 2. Seminumerical Algorithms. 3rd edition / D. E. Knuth. Boston, Mass, USA : Addison-Wesley, Longman Publishing, 1997. 762 p.
4. William Stallings. Cryptography and Network Security Principles and Practices, Fourth Edition p. 592.

Ключові слова: адитивно-мультиплікативні перетворення, поточне шифрування, лінійний конгруентний генератор, псевдовипадкові числа, шифруюча гама

Keywords: additive-multiplicative transformations, current encryption, linear congruent generator, pseudorandom numbers, encryption gamma

Науковій керівник: доцент Щербина Ю.

ПОРОГОВА СХЕМА РОЗПОДІЛУ СЕКРЕТУ

Онацький О. В.

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

Жарова О. В.

кандидат фізико-математичних наук, доцент кафедри вищої математики Національного університету «Одеська політехніка»

Онацька А. О.

студентка 4 курсу Міжнародного гуманітарного університету

Для захисту секретної інформації від втрати і компрометації необхідно підвищити надійність зберігання секретної інформації. Одним з методів підвищення

надійності зберігання секретної інформації, є використання схем розподілу секрету (secret sharing scheme) [1, 2] (рис. 1).

Схеми розподілу секрету знайшли широке застосування в системах зберігання, передачі, обробки й захисту інформації, які забезпечують зберігання розподіленої інформації. У криптографії під схемою розподілу секрету розуміють будь-який метод поділу секрету серед групи абонентів (учасників), кожному з яких дістається частка секрету [1]. Секрет може бути відновлений тільки певною кількістю абонентів (учасників). Загальна класифікація схем розподілу секрету показано на рис. 1.

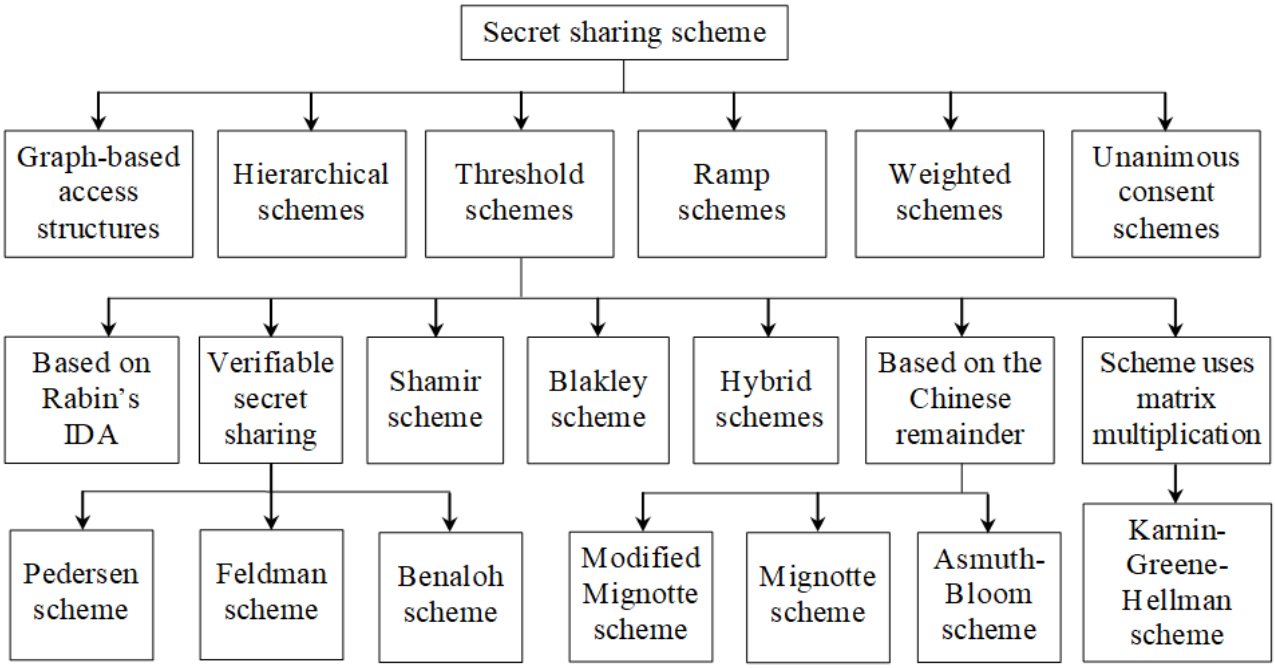


Рисунок 1 – Загальна класифікація схем розподілу секрету

В роботі запропоновано порогова схема розподілу секрету (threshold secret sharing scheme) на основі математичного апарату еліптичних кривих (elliptic curves – EC) [3, 4], структурна схема яка показано на рис. 2. Безпека криптосистем на EC, заснована на труднощах розв'язання задачі дискретного логарифмування в групі точок еліптичної кривої [3].

У криптосистем на EC запропоновано використовувати криптоперетворення, що базуються на перетвореннях у групі точок еліптичних кривих над полями $GF(p)$, $GF(2^m)$, $GF(p^m)$ [3-5]. У криптосистемах над розширеним полем $GF(2^m)$ рівняння EC має вигляд [4, 5]:

$y^2 + xy \equiv (x^3 + ax^2 + b) \bmod (f(x), 2)$ – несуперсингулярна крива;

$y^2 + cy \equiv (x^3 + ax + b) \bmod (f(x), 2)$ – суперсингулярна крива,

де x, y – точки EC, $x, y \in E(GF(2^m))$; a, b, c – коефіцієнти EC; $f(x)$ – незведений поліном над полем $GF(2)$.

Нехай $E_p(a, b)$ – еліптична крива, відома учасникам інформаційного процесу; $f(x)$ – незведений поліном над полем $GF(2)$; G – попередньо погоджена точка цієї кривої; $\#E_p(a, b) = n$ – порядок групи еліптичної кривої; k_a і k_b – секретні ключі абонентів A, B; r_a і r_b – випадкові числа абонентів A, B; $M = P_M$ – секрет. Абонент A обчислює значення відкритого ключа Y_a та R_a, які передає абоненту B. Абонент B обчислює значення відкритого ключа Y_b та R_b, який передає абоненту A. Абоненти A, B обчислюють спільний ключ K. Наступним кроком, абонент A розподіляє секрет M на m кількість часток $F(m)$, які мають порогове значення t (наприклад, порога схема розділення секрету Шаміра (t, m)). Далі, абонент A перетворює кожну з часток у значення P_i, які передаються абоненту B. Абонент B маючи різні частки $F(j)$ може відновити секрет M. Приклад порогової схеми розділення секрету Шаміра на ЕС розглянуто в статті [6].

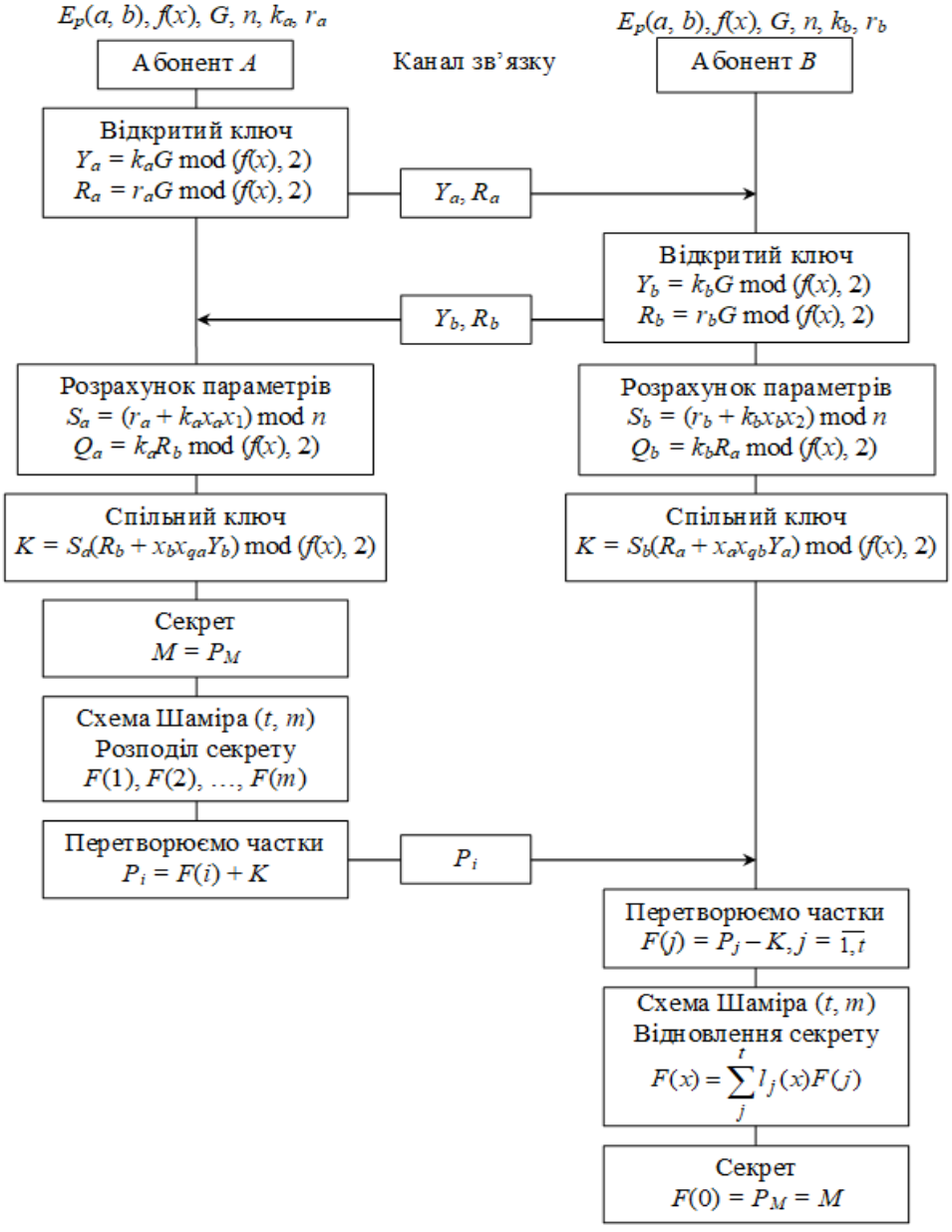


Рисунок 2 – Запропонована порогова схема розподілу секрету

Таким чином, у роботі запропоновано порогова схема розподілу секрету на основі математичного апарату еліптичних кривих над розширеним полем $GF(2^m)$. Схема дозволяє розділяти спільні ресурси, розподіляючи їх між різними хмарними системами, а потім відновити їх.

Список використаних джерел:

1. Calkavur S., Sole P., Bonnecaze A. Code based secret sharing schemes. Applied combinatorial coding. World Scientific Publishing Co. Pte. Ltd., 2022. 215 p.
2. Schneier B. Applied Cryptography: Protocols, Algorithms and Source Code in C: 20th Anniversary Edition. Wiley, 2015. 784 p.
3. Stavroulakis P., Stamp M. Handbook of Information and Communication Security. Berlin: Springer-Verlag, 2010. 863 p.
4. Горбенко І. Д., Горбенко Ю. І. Прикладна криптологія. Теорія. Практика: монографія, Харків: Видавництво «Форт», 2012. 880 с.
5. Hankerson D., Menezes A., Vanstone S. Guide to Elliptic Curve Cryptography. Springer-Verlag, 2004. 358 p.
6. Onatskyi O., Dykyi O., Zharova O., Yona L. Modification of the Shamir and Blakley threshold secret sharing schemes with elliptic curves. Інфокомунікаційні та комп'ютерні технології. № 1(03), 2022. С. 267–283.

Ключові слова: схема розподілу секрету, частка секрету, фаза розподілу секрету, фаза відновлення секрету, порогова схема розподілу секрету, еліптичні криві

Keywords: secret sharing scheme, secret share, secret sharing, secret reconstruction, threshold secret sharing scheme, elliptic curves

ТЕХНІЧИЙ АСПЕКТ ЗАХИСТУ ДІАЛОГІВ З CHATGPT ВІД ВИТОКУ

Слатвінська В.М.

асистент кафедри кібербезпеки
Національного університету «Одеська юридична академія»

Не секрет, що вся інформація, яку надає ChatGPT, знаходиться у вільному доступі в Інтернеті. По суті це індивідуалізований інструмент парсингу інформації оскільки він дає можливість використовувати модель для надання конкретних кроків, ставлячи контекстні запитання [1].

З огляду на наявні скарги на ChatGPT на підставі порушень конфіденційності, в тому числі GDPR [2] постає питання щодо захищеності діалогів користувачів від витоку інформації. Після того, як Samsung Semiconductor дозволила інженерам використовувати ChatGPT для допомоги, і вони почали використовувати його для швидкого виправлення помилок у своєму вихідному коді, стався витік конфіденційної інформації, такої як нотатки з внутрішніх нарад і дані про бізнес-

показники [3]. ChatGPT є потужним інструментом, але його функція відкритого навчання потенційно може надавати конфіденційну інформацію третім особам, що є неприйнятним у висококонкурентній IT галузі.

Добре відомо, що поки ChatGPT не має фізичної форми, але його програмний код та параметри зберігаються на сервері. У зв'язку з цим і з метою захисту діалогів з ChatGPT від витоку необхідно вжити наступних заходів:

Забезпечити шифрування даних. Це потрібно для захисту даних під час передачі їх між користувачем та сервером можна використовувати шифрування. Крім того, це забезпечить захист від злочинного доступу до даних в транзиті.

Передбачити аутентифікацію користувача. Перед початком діалогу з ChatGPT користувач може бути попрошений надати інформацію для аутентифікації, наприклад, логін та пароль. Це допоможе встановити ідентичність користувача та захистити від несанкціонованого доступу до діалогів.

Контроль доступу. Доступ до діалогів з ChatGPT може бути обмежений лише авторизованим користувачам або групам користувачів з відповідними дозволами. Це дозволить зменшити ризик несанкціонованого доступу до діалогів.

Моніторинг та аналіз діяльності користувачів. Це потрібно для того, щоб виявити можливі порушення безпеки. Для цього можна використовувати моніторинг та аналіз діяльності користувачів. Це може допомогти виявити незвичні зразки поведінки та вчасно прийняти відповідні заходи.

Резервне копіювання даних. Це потрібно для захисту діалогів від випадкового видалення або пошкодження. Можна виконувати їх резервне копіювання на інших серверах. Це забезпечить можливість відновлення діалогів у разі втрати даних на основному сервері.

Перелічені вище заходи можуть бути поєднані та доповнені іншими методами.

Самі по собі діалоги в ChatGPT можуть бути технічно захищеними на високому рівні, якщо будуть застосовані всі необхідні заходи для захисту даних та діалогів від несанкціонованого доступу та витоку.

Однак, необхідно розуміти, що жодна система не може гарантувати 100% захист від атак та витоків. Завжди існує ризик, що хакер або кіберзлочинець знайде спосіб обійти захист та отримати доступ до даних. Тому, окрім технічних заходів, необхідно також вживати організаційних та правових заходів для забезпечення безпеки даних та діалогів.

У будь-якому випадку, технічні заходи, які можуть бути використані для захисту діалогів в ChatGPT, включають шифрування даних під час передачі, аутентифікацію користувача, контроль доступу, моніторинг та аналіз діяльності

користувачів, резервне копіювання даних та інші. Тож використання цих заходів може допомогти забезпечити високий рівень захисту діалогів в ChatGPT.

Кожен користувач діалогу в ChatGPT може бути захищений від витоку персональних даних, якщо будуть застосовані необхідні технічні та організаційні заходи для забезпечення безпеки даних.

У зв'язку з тим, що в діалогах з ChatGPT можуть обговорюватись особисті (персональні) дані, такі як імена, адреси, електронні адреси, номери телефонів, то їх захист від витоку має бути дуже важливим завданням для розробників та провайдерів послуг.

Технічні заходи для захисту персональних даних можуть включати шифрування даних під час передачі, контроль доступу до даних, захист від атак типу "людина посередині", які дозволяють злочинцям отримати доступ до обміну даними між користувачем і сервісом.

Організаційні заходи можуть включати дотримання правил конфіденційності та безпеки даних, використання стандартів безпеки, підтримку гарантійних зобов'язань щодо захисту даних користувачів, та інші.

Незважаючи на це, користувачі також повинні дотримуватись власних правил безпеки і захисту своїх особистих даних, наприклад, не розголошувати конфіденційну інформацію в діалогах з ненадійними сторонами.

Захист персональних даних в GPT-4, на відміну від попередньої версії – ChatGPT, має бути однією з важливих функцій в цій новій моделі мови. Очікується, що GPT-4 буде вдосконаленою версією попередніх моделей GPT, яка матиме значно більше параметрів та можливостей, а також буде вдосконалена щодо технічних та організаційних заходів для захисту персональних даних.

Однак, варто зазначити, що захист персональних даних від витоку залежить не тільки від самої моделі мови, але і від технічних та організаційних заходів, які використовуються для зберігання та обробки даних. Тому, безпека та конфіденційність даних будуть залежати від того, наскільки добре будуть дотримуватись розробники відповідних заходів для захисту даних.

При цьому частота створення резервних копій діалогів залежить від того, які дані потрібно зберегти та наскільки важливо зберігати їх для майбутнього використання або дослідження. Однак, як правило, великі компанії та організації, що використовують моделі штучного інтелекту, включаючи ChatGPT, зазвичай створюють резервні копії регулярно.

Для прикладу, OpenAI, компанія, що розробляє ChatGPT, зазначає, що вони мають механізми автоматичної резервного копіювання та відновлення системи, а також резервування даних, включаючи тексти відгуків користувачів. Крім того,

вони проводять регулярні аудити безпеки, щоб переконатися в тому, що дані користувачів зберігаються в безпеці.

Точна частота створення резервних копій може залежати від різних факторів, таких як обсяг даних, кількість користувачів та інші технічні параметри. Однак, зазвичай резервні копії створюються регулярно, щоб забезпечити захист даних у разі випадкового втрати або пошкодження.

ChatGPT не розуміє важливості діалогу з користувачем самостійно. Однак, компанії, які використовують моделі штучного інтелекту, включаючи ChatGPT, можуть мати встановлені правила та процедури збереження даних, які вони збирають, включаючи діалоги з користувачами.

Такі правила та процедури можуть базуватися на різних факторах, таких як тип інформації, яку вони збирають, рівень чутливості цієї інформації, відповідність законодавству про захист персональних даних та інші. Зокрема, якщо компанія працює в області медицини, то може бути встановлене правило збереження всіх діалогів з користувачами для майбутніх досліджень, або якщо компанія працює з фінансовою інформацією, то може бути встановлене правило збереження діалогів з користувачами протягом певного періоду часу.

Підсумовуючи, збереження діалогів з користувачами може бути частинами загальної стратегії збереження даних компанії, і відповідні правила та процедури можуть бути встановлені для забезпечення належного захисту та збереження інформації.

Список використаних джерел:

1. Europol warns against potential criminal uses for ChatGPT and the likes URL: https://www.euractiv.com/section/artificial-intelligence/news/europol-warns-against-potential-criminal-uses-for-chatgpt-and-the-likes/
2. ChatGPT вступає у світ регуляторних проблем у Європі URL: https://daycom.com.ua/chatgpt-vstupaie-u-svit-rehuliatornykh-problem-u-ievropi/
3. Використання ChatGPT призвело до трьох епізодів витоку даних в Samsung URL: https://mc.today/uk/vikoristannya-chatgpt-prizvelo-do-troh-epizodiv-vitoku-danih-v-samsung/

Ключові слова: ChatGPT, витік даних, діалог, дані.
Keywords: ChatGPT, data leakage, dialog, data.

КІБЕРБЕЗПЕКА МАРКЕТИНГОВИХ КОМУНІКАЦІЙ СУБ'ЄКТІВ ІНТЕРНЕТ-ТОРГІВЛІ

Горбаченко С. А.

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

Скідан В. С.

маркетолог

У сучасному цифровому світі найбільш популярною формою торгівлі є торгівля за допомогою мережі Інтернет. Вона дозволяє покупцям здійснювати покупки з будь-якого місця та в будь-який час і при цьому забезпечує широкий асортимент та конкурентні ціни. Станом на кінець 2021 року загальний обсяг продажів товарів онлайн досяг 2,992 трлн дол. [1].

В регіональному аспекті лідером за обсягами продажів в мережі Інтернет виступає Китай. За даними eMarketer на 2021 рік, його частка в онлайн-торгівлі становила 52,1% (рис. 1).

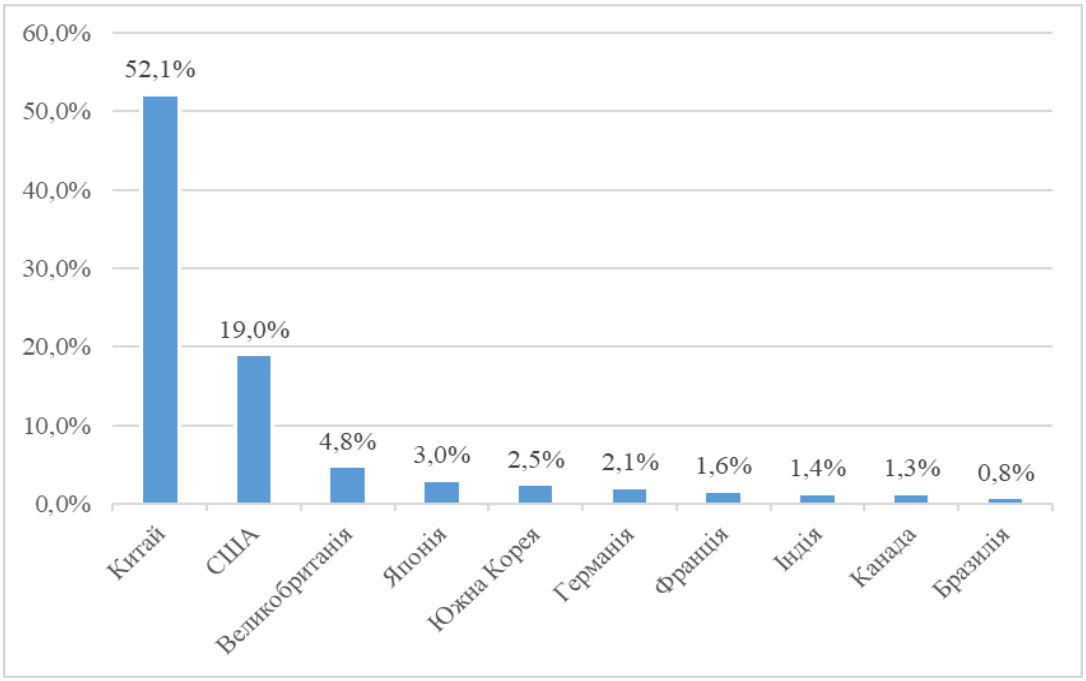


Рисунок 1 – Структура світового ринку Інтернет-торгівлі за 2021 рік
Джерело: складено за даними [2]

Глобальний характер Інтернет-торгівлі, крім іншого, ставить перед підприємцями й завдання пошуку найбільш ефективних каналів комунікації для кожного зі своїх клієнтів. Під маркетинговими комунікаціями розуміють комплексний вплив підприємства на внутрішнє й зовнішнє середовище для створення сприятливих умов для стабільної прибуткової діяльності [3, с. 28]. Основними

завданнями використання маркетингових комунікацій суб'єктів Інтернет-торгівлі є своєчасне інформування споживачів про нові товари, формування лояльних груп споживачів, підвищення власної конкурентоспроможності та покращення іміджу тощо.

Відтак, з одного боку розвиток інформаційних технологій вимагає від суб'єктів Інтернет-торгівлі використання нових підходів до формування маркетингових комунікацій, які передбачають більш глибоку інтеграцію клієнтських даних. Наприклад, під час взаємодії з Інтернет-магазинами, клієнти надають свої особисті дані, такі як ім'я, адреса, номер кредитної карти. Отже суб'єкти Інтернет-торгівлі повинні гарантувати безпеку цих даних та запобігати несанкціонованому доступу до них з боку зловмисників. Для цього можна використовувати шифрування даних, двофакторну аутентифікацію, захищені протоколи передачі інформації та інші заходи безпеки.

З огляду на вищевказане цифровізація маркетингових комунікацій передбачає й критичну важливість забезпечення кібербезпеки усіх бізнес-процесів. Серед конкретних прикладів кіберзагроз можна зазначити промислове шпигунство, хакерські атаки, використання піратських програмних продуктів, можливість несанкціонованого втручання в бази даних, недобросовісне використання інформації працівниками, відсутність копіювання важливих документів на матеріальних носіях, відсутність практики ведення протоколів змін у програмному забезпеченні тощо [4]

При цьому кіберзлочинці постійно шукають нові способи зламування систем безпеки з метою отримання доступу до грошей, нанесення шкоди репутації підприємця, крадіжки конфіденційних даних клієнтів. Наприклад, в останні роки найбільшу загрозу в сегменті Інтернет-торгівлі становить фішинг, коли шахраї намагаються обдурити користувачів та викрасти їхні гроші або конфіденційні дані, використовуючи різні методи маніпуляції та соціального інжинірингу. У відповідь на наявну загрозу суб'єкти Інтернет-торгівлі намагаються інформувати про неї своїх клієнтів та навчати їх виявляти фішингові атаки, використовувати антивірусне програмне забезпечення, спеціальні фільтри для електронної пошти та інші засоби захисту.

Побудова сучасної системи кіберзахисту в процесі здійснення маркетингових комунікацій суб'єктом Інтернет-торгівлі передбачає вживання низки відповідних заходів (табл. 1).

Таблиця 1

Основні заходи забезпечення кібербезпеки маркетингових комунікацій

№	Вид заходів	Опис
1.	Захист мережі	Використання брандмауерів, віртуальних приватних мереж (VPN), систем виявлення вторгнень (IDS) та інших технологій, що допомагають виявляти та запобігати несанкціонованому доступу до мережі.
2.	Захист від шкідливих програм	Встановлення та регулярне оновлення антивірусного програмного забезпечення.
3.	Захист від фішингу	Навчання співробітників впізнавати фішингові атаки та уникати небезпечних посилань або прикріплених файлів. Оновлювати паролі та використовувати двофакторну аутентифікацію для забезпечення безпеки облікових записів.
4.	Захист від DDoS-атак	Постійний моніторинг трафіку, застосування хмарних технологій та технічних систем розподіленого захисту.
5.	Захист конфіденційності даних користувачів	Забезпечення шифрування даних, обмежений доступ до них, встановлення політик конфіденційності та відповідний контроль доступу.
6.	Резервне копіювання даних	Регулярне резервне копіювання даних маркетингових комунікацій є важливим для запобігання втраті даних в результаті випадкового видалення, технічних проблем або кібератак.
7.	Аудит безпеки	Оцінка систем безпеки, перевірка вразливостей та тестування на проникнення для виявлення потенційних проблем.

Джерело: складено автором за даними [5]

Основні заходи кібербезпеки спрямовуються на захист від хакерських атак на вебсайти та інші онлайн-ресурси. Зловмисники можуть спробувати зламати сайт, знищити або модифікувати вміст, або викрасти конфіденційну інформацію.

З огляду на це суб'єкти інтернет-торгівлі повинні регулярно оновлювати спеціальне програмне забезпечення, використовувати надійні паролі, резервні копії даних та системи, а також встановлювати системи виявлення вторгнень для вчасного виявлення і реагування на потенційні атаки.

В цьому сенсі кібербезпека маркетингових комунікацій є постійним та безперервним процесом, оскільки кіберзагрози також постійно еволюціонують. Всі співробітники, які займаються маркетинговими комунікаціями та, відповідно, мають доступ до конфіденційних даних, повинні бути ознайомлені з основними принципами кібербезпеки. Вказаний аспект включає усвідомлення загроз, які можуть виникнути, розуміння процедур безпеки, які слід дотримуватися, та навички виявлення потенційних атак. Регулярне навчання та оновлення знань персоналу щодо нових загроз та методів захисту також є критично важливими.

Нарешті, розуміння та дотримання законодавства з питань кібербезпеки є невід'ємною частиною безпечної діяльності суб'єктів Інтернет-торгівлі. Вони повинні враховувати місцеві та міжнародні закони щодо захисту даних, конфіденційності, інтелектуальної власності та інших аспектів кібербезпеки. Дотримання законодавства допомагає забезпечити законність та надійність взаємодії з клієнтами в процесі здійснення торгівельних операцій.

Таким чином лише шляхом впровадження комплексного підходу до питань кібербезпеки в процесі маркетингових комунікацій суб'єкти Інтернет-торгівлі можуть захистити власні інтереси, інтереси клієнтів та суспільства в цілому.

Список використаних джерел:

1. Статистика розвитку e-commerce у найбільших регіонах світу. 2022. URL: https://magazine.ukr-china.com/statystyka-rozvytku-e-commerce-u-najbilshyh-regionah-svitu/
2. Top 10 US ecommerce companies for 2021. URL: https://www.insiderintelligence.com/chart/247783/Top-10-Countries-Ranked-by-Retail-Ecommerce-Sales-Share-2021-of-total-worldwide-retail-ecommerce
3. Васильченко Л. С. Сутність та сучасні тенденції розвитку маркетингових комунікацій підприємства. Причорноморські економічні студії. 2019. Вип. 48-2. С. 27-30.
4. Панченко В. А. Менеджмент інформаційної безпеки комерційного підприємства. URL: http://economics.kntu.kr.ua/pdf/3(36)/23.pdf
5. Основи кібербезпеки та кібероборони: підручник / Ю.Г. Даник, П.П. Воробієнко, В.М. Чернега. Одеса.: ОНАЗ ім. О.С. Попова, 2019. 320 с.

Ключові слова: кібербезпека, інтернет-торгівля, дані, захист

Keywords: cybersecurity, e-commerce, data, protection

CAIN AND ABEL - ІНСТРУМЕНТ ХАКЕРА І КРИМІНАЛІСТА

Лукашенко О. О.

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

Програми для зламування паролів можуть використовуватися не тільки для отримання несанкціонованого доступу до комп'ютерного пристрою, а й для тестування безпеки комп'ютерних пристроїв та мереж. Вони використовують різні методи атаки, такі як метод грубої сили, підбір паролів за словниками та методи соціальної інженерії. Однією з найпопулярніших програм «для злому паролів» є програма Cain and Abel.

«Cain and Abel» (українською «Каїн і Авель») – це назва програмного забезпечення, розробленого компанією Massimiliano Montoro, яке спочатку створювалося як програма відновлення паролів для Windows, нині поширюється як безкоштовний інструмент для тестування безпеки мереж, а також використовується і для виконання різноманітних атак на мережеву безпеку та розшифровки паролів [1].

«Cain and Abel» надає різноманітні можливості з взлому паролів, перехопленню мережевого трафіку, аналізу мережевої безпеки та розшифруванню зашифрованих паролів. Деякі з основних функцій цього програмного забезпечення включають перехоплення паролів через атаки Man-in-the-Middle (MITM), дешифрування паролів збережених на локальному комп'ютері, перехоплення аутентифікаційної інформації з мережі, а також атаки на протоколи безпеки, такі як ARP (Address Resolution Protocol) poisoning [2, с.170].

Однак, важливо зауважити, що використання «Cain and Abel» для будь-яких злочинних дій, які порушують закони про безпеку і приватність, є незаконним і не етичним. Це програмне забезпечення повинно використовуватись лише для тестування безпеки в рамках законних дозволів та згоди власника системи.

«Cain and Abel» є потужним інструментом для виконання тестування на проникнення та аудиту безпеки мереж і систем. Він надає можливість аналізувати слабкі місця в мережевій інфраструктурі та ідентифікувати потенційні вразливості.

Одна з ключових функцій «Cain and Abel» - це перехоплення мережевого трафіку, включаючи перехоплення паролів, обмінюваних по мережі. Це можливо завдяки методам, таким як атаки MITM (Man-in-the-Middle), де програма працює як проміжний вузол між двома комунікуючими сторонами і перехоплює їхні дані, включаючи паролі.

«Cain and Abel» також має здатність розшифровувати зашифровані паролі, збережені на локальному комп'ютері. Він використовує різні методи, такі як атаки на хеш-функції, підбір паролів та використання словникових атак, щоб отримати доступ до зашифрованої інформації.

Зокрема, «Cain and Abel» підтримує такі види атак:

Атаки на протоколи безпеки: Програма може виконувати атаки, такі як ARP poisoning, DNS spoofing та інші, для перехоплення та зловживання мережевими протоколами.

Атаки на паролі: «Cain and Abel» надає можливість виконувати атаки на хеш-функції, підбір паролів і використання словникових атак для розшифрування паролів, які зберігаються на локальному комп'ютері.

Аналіз мережевої безпеки: Програма дозволяє сканувати мережі, виявляти активні пристрої, порти та потенційні вразливості мережевої інфраструктури [3, с. 66].

Важливо розуміти, що використання «Cain and Abel» для будь-яких нелегальних або злочинних дій, таких як несанкціонований доступ до системи або викрадення паролів, є суворо забороненим і неприпустимим. Зловживання цим програмним забезпеченням може призвести до порушення закону, порушення приватності та завдання шкоди іншим користувачам або мережевим системам.

«Cain and Abel» повинен використовуватись виключно для законних цілей, таких як тестування безпеки мереж і систем, покращення власної безпеки та згоди з власником системи. Під час використання програми для цих цілей, необхідно дотримуватись вимог законодавства та етичних норм.

Наголошується, що здійснення будь-яких незаконних дій з використанням «Cain and Abel» може мати серйозні правові наслідки. У разі потреби в тестуванні безпеки або консультуванні з питань інформаційної безпеки, рекомендується звертатися до відповідних професіоналів та експертів у сфері кібербезпеки.

Існує кілька проблем, пов'язаних з використанням «Cain and Abel» або подібного програмного забезпечення:

Легальність: Використання «Cain and Abel» для незаконних дій, таких як несанкціонований доступ до системи або викрадення паролів, є неприпустимим і незаконним. Це може призвести до серйозних правових наслідків і кримінального переслідування [4, с.225].

Етика: Зловживання програмою для порушення приватності і завдання шкоди іншим особам або мережевим системам є неетичним і несправедливим. Важливо дотримуватись етичних принципів та поважати права та приватність інших користувачів.

Законна відповідність: Використання «Cain and Abel» повинно здійснюватися в рамках законів та норм, що регулюють безпеку та приватність. Здійснення дій, які порушують ці закони, може мати серйозні юридичні наслідки.

Порушення безпеки: Використання «Cain and Abel» без належних знань та досвіду може призвести до небезпеки та порушення безпеки. Недостатній розуміння використання програми може призвести до помилок або недостатньо захищеного тестування, що може вразити мережеву інфраструктуру або системи.

Відповідальність: Власники систем і мереж повинні бути відповідальними за власну безпеку та захист. Використання «Cain and Abel» повинно бути здійснене тільки з дозволу власника системи і в межах законних вимог.

У висновку до теми «Cain and Abel» можна сказати, що це програмне забезпечення є потужним інструментом для тестування безпеки мереж і систем. Воно надає можливість перехоплювати мережевий трафік, розшифровувати паролі та аналізувати мережеву безпеку. Однак, використання «Cain and Abel» для злочинних або нелегальних дій є неприпустимим і незаконним [5, с.124].

Важливо дотримуватись етичних і законних принципів під час використання програмного забезпечення, такого як «Cain and Abel». Воно повинно використовуватись тільки з дозволу власника системи та в рамках законів і норм, що регулюють безпеку і приватність. Використання програми для тестування безпеки та покращення власної безпеки є доцільним, проте завжди слід дотримуватись принципів законності і етики. Наслідування неправомірних дій з використанням «Cain and Abel» може призвести до серйозних правових наслідків, порушення приватності та завдання шкоди іншим. В разі потреби в тестуванні безпеки або консультацій з питань інформаційної безпеки рекомендується звертатися до професіоналів та експертів у цій галузі.

Список використаних джерел:

1. Mohamed Ahmed Password cracking using Cain & Abel [Electronic resource]. – URL: https://resources.infosecinstitute.com/topic/password-cracking-using-cain-abel.
2. Anderson, L., & Thompson, R. (2015). The ethical implications of using Cain and Abel for network security testing. Journal of Information Ethics, 24(3), 167-182.
3. Smith, J., & Johnson, A. (2010). Analyzing network security using Cain and Abel. In Proceedings of the International Conference on Cyber Security (ICCS), 65-70.
4. Abel, P., & Cain, M. (2005). Cain & Abel: A versatile platform for password recovery. In International Journal of Information Security, 4(3-4), 224-227.
5. Brown, R., & Davis, S. (2012). Evaluating the effectiveness of Cain and Abel in penetration testing. Journal of Computer Security, 20(2), 123-137.

Ключові слова: безпека, паролі, атака, програмне забезпечення

Keywords: security, passwords, attack, software

Науковий керівник: доцент Чанишев Р.І.

МЕТОДИ ЗАХИСТУ АВТЕНТИФІКАЦІЇ КОРИСТУВАЧІВ

Мозговий М. В.

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

На сьогодні безпека є однією з найбільш важливих та актуальних тем у світі інформаційних технологій. З повсюдним застосуванням вебтехнологій рівень кіберзагроз та вразливостей зростає, тому важливо підтримувати безпеку створюваних вебзастосунків на високому рівні. Нині сучасні компанії та агентства переважно у своїй роботі покладаються на безпеку хмарних технологій для організації робочого середовища та зберігання даних.

Від безпеки доступу до акаунта залежить можливість безпечного здійснення різних фінансових операцій, роботи з електронною поштою, спілкування у соціальних мережах тощо. Інформаційна безпека будь-якого застосунку повинна підтримувати конфіденційність, цілісність та доступність відповідних даних. Конфіденційність означає, що інформація повинна залишатися приватною і недоступною для несанкціонованого доступу. Цілісність означає, що інформація повинна залишатися незмінною та повністю точною. Доступність означає, що інформація має бути доступною для використання, коли це потрібно. Для конфіденційності інформації необхідно захищати облікові записи адміністраторів та людей, які можуть отримати ці дані, варто унеможливити усі спроби атак на інфраструктуру системи.

Для збільшення вірогідності, що пароль не буде скомпрометовано, варто зберігати дані автентифікації у гешованому вигляді і при цьому додавати сіль до гешу. Соління паролів використовують для захисту паролів користувачів, які зберігаються в базах даних, шляхом додавання рядка з 32-х або більше символів, а потім їх гешування [1]. Соління паролів запобігає швидкому брутфорсу хакерами, зворотній інженерії паролів і крадіжці їх із бази даних.

Від адміністратора вебсайту чи то вебзастосунку багато в чому залежить не лише безпека даних, а й можливість доступу до програмної системи та навіть, в певній мірі, працездатність компанії. Для безпеки акаунта адміністратора варто застосовувати додаткові методи захисту:

- SMS-автентифікація;
- двофакторна автентифікація;
- складні паролі можуть бути у вигляді текстового або графічного подання;
- шифрування даних автентифікації;
- регулярне оновлення паролів;

- зберігання паролів у захищеному середовищі;
- моніторинг входу будь-яких користувачів програмної системи.

Застосування одного чи то комбінації додаткових методів захисту знижуватиме ризики несанкціонованого доступу. Здебільшого для контролю процесу автентифікації користувачів використовують комбінацію цих методів.

Метод SMS-автентифікації базується на використанні одноразового пароля. Перевага цього підходу, в порівнянні з постійним паролем, полягає в тому, що пароль не можна використовувати повторно. Навіть якщо припустити, що зловмисник може перехопити дані під час обміну інформацією, він не зможе ефективно використовувати викрадений пароль для доступу до системи. По суті, метод SMS-автентифікації може бути складовою частиною багатофакторної автентифікації. Завдяки зручності і надійності, SMS-автентифікація є доволі поширеною. Так, 97% респондентів онлайн-банкінгу використовують саме SMS-автентифікацію [2]. Менш поширеними методами є автентифікація через ідентифікатор мобільного телефону (IMEI, serial number) або через sim-картку. Так, 91% респондентів користуються мобільними застосунками, ідентифікуючи себе через ID телефону та додаткові біометричні фактори, а 47% при щоденному користуванні банківськими послугами застосовують ідентифікатор власної sim-картки [2]. Проте, SMS-автентифікації має значну проблему, пов’язану з тим, що дані мобільного зв’язку не шифруються та передаються у відкритому вигляді. Тому через шкідливе програмне забезпечення можливо отримати SMS-дані та виконати вхід зловмисника у систему.

Якщо розглядати інший метод захисту – графічні паролі, то вони як різновид складних паролів, мають певні переваги перед текстовими. Адже в силу людського, переважно візуального сприйняття розпізнати та запам’ятати якесь зображення набагато легше, аніж текст, що може бути певним вирішенням проблеми із запам’ятовуванням складних текстових паролів. Однак графічні паролі більш вразливі до атак, які можна реалізувати зловмисником через спостереження за діями користувача, коли той вводить дані під час автентифікації (паролі, пін-коди або номери банківських карток). Поширено такого роду спостереження і подальші кібератаки реалізують у людних місцях, наприклад, біля банкомата, у кав'ярні або в громадському транспорті. Крім того, автентифікація за допомогою графічного пароля є відносно дорогою у реалізації, що ускладнює її поширення. Для усунення слабких сторін текстових і графічних паролів використовують гібридну схему автентифікації, яка є комбінацією текстових і графічних паролів [3].

Запропонований у цій роботі алгоритм захисту автентифікації користувачів пропонує: для входу в акаунт вводити не тільки свій логін і пароль, а й

використовувати згенероване адресне посилання з геш-секретом – додатковим ключем у формі текстових даних. Доступ до акаунта можливий тільки у разі збігу всіх трьох значень (логін, пароль, геш-секрет). Геш-секрет є текстовими даними, які долучаються до генерованого рядка пошуку. Його використання запобігає злому через підбір паролів шляхом перебору всіх можливих комбінацій (брутфорсу, bruteforce). Запропонований метод може використовувати різні алгоритми гешування рядка геш-секрету (MD5, SHA-1, SHA-256, SHA-512 [4, 5]) для створення додаткових перевірок при вході до акаунта.

Для зберігання в базі даних структура ідентифікаційних даних для запропонованого алгоритму може бути такою:

- логін – найчастіше це номер телефону або електронна пошта;
- пароль – переважно зберігається в зашифрованому вигляді, методи шифрування краще вибирати криптостійкі до злому або брутфорсу;
- геш-секрет – текстове поле, яке додаватиметься до рядка адреси;
- додаткові поля, в яких може зберігатися додаткова інформація про користувача.

На відміну від звичайного алгоритму автентифікації, в алгоритмі, запропонованому в роботі, гешування додаткового поля геш-секрету в поєднанні з полями автентифікації сприяє досягненню безпечного доступу до акаунта користувача. Розроблений алгоритм добре контролює порушення цілісності геш-секрету, використовуючи криптографічні методи, тим самим контролює безпечну автентифікацію користувача у системі.

Список використаних джерел:

1. Awati R. Password salting. URL: https://www.techtarget.com/searchsecurity/definition/salt.
2. Yevseiev S., Kots H., Liekariev Y. Developing of multi-factor authentication method based on niederreiter-mceliece modified crypto-code system. Eastern-European Journal of Enterprise Technologies. 2016. Vol. 6/4 (84). P. 11-23. DOI: 10.15587/1729-4061.2016.86175.
3. Mackie I., Yıldırım M. A Novel Hybrid Password Authentication Scheme Based on Text and Image. 32th IFIP Annual Conference on Data and Applications Security and Privacy (DBSec). 2018. Bergamo, Italy. P. 182-197. DOI:10.1007/978-3-319-95729-6_12.
4. The MD5 Message-Digest Algorithm. URL: https://www.ietf.org/rfc/rfc1321.txt.
5. Secure Hash Standard (SHS). URL: https://nvlpubs.nist.gov/nistpubs/fips/nist.fips.180-4.pdf

Ключові слова: гешування, геш, автентифікація, соління пароля, гібридний пароль, акаунт, брутфорс

Keywords: hashing, hash, authentication, password salting, hybrid password, account, bruteforce

Науковий керівник: доцент Трофименко О. Г.

ПРИНЦИПИ АНАЛІЗУ ВРАЗЛИВОСТЕЙ ВЕБ-СЕРВЕРІВ

Стаднік Д. А.

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

Веб-сервер - комп'ютерна система, яка відповідає за обслуговування веб-сторінок і веб-додатків для клієнтів через Інтернет, а також, на якій розміщуються веб-сайти та веб-додатки і яка робить їх доступними для користувачів через Інтернет. Веб-сервер прослуховує вхідні запити від клієнтів, таких як веб-браузери, і відповідає на них, відправляючи назад запитувану веб-сторінку або додаток. Такі, веб-сервери вразливі до низки загроз, які можуть поставити під загрозу їхню безпеку та викрити конфіденційну інформацію або дозволити зловмисникам отримати контроль над сервером[1].

Для зв'язку з клієнтами веб-сервери зазвичай використовують протокол передачі гіпертексту (HTTPS). Коли клієнт запитує веб-сторінку або додаток, веб-сервер отримує запитуваний вміст зі свого сховища, наприклад, файлової системи або бази даних, і надсилає його назад клієнту у вигляді HTTP-відповіді.

Існує кілька видів веб-серверів, включаючи[2]:

1) Apache HTTP Server - це найбільш поширене програмне забезпечення для веб-сервера, яке використовується більш ніж на 50% всіх сайтів. Воно є відкритим вихідним кодом і може працювати на декількох платформах, включаючи Windows, Linux та Unix.

2) Microsoft Internet Information Services (IIS) - це програмне забезпечення для веб-сервера, створене компанією Microsoft, і зазвичай використовується для хостингу веб-сайтів на серверах Windows.

3) NGINX - це високопродуктивний веб-сервер, проксі-сервер та зворотний проксі-сервер, який може ефективно обробляти великі обсяги трафіку.

4) Lighttpd - це легкий та швидкий веб-сервер, який часто використовується для обслуговування статичного контенту.

5) Node.js - це середовище виконання JavaScript, побудоване на движку JavaScript V8 від Chrome, яке можна використовувати для створення швидких та масштабованих веб-серверів.

6) Google Web Server (GWS) - це власний веб-сервер, який використовується Google для обслуговування сайтів.

7) Sun Java System Web Server - це веб-сервер, створений компанією Sun Microsystems, який часто використовується для хостингу веб-застосунків,

написаних на Java.

До найпоширеніших загроз для веб-серверів належать[3]:

1) DDoS-атаки: Розподілені атаки на відмову в обслуговуванні призначені для того, щоб перевантажити веб-сервер трафіком, роблячи його недоступним для легальних користувачів. DDoS-атаки можуть бути запущені з використанням мережі скомпрометованих комп'ютерів або шляхом використання вразливостей у веб-додатках, що працюють на сервері.

2) SQL-ін'єкція - це тип уразливості веб-додатків, який дозволяє зловмисникам виконувати шкідливі SQL-запити до бази даних веб-сервера. Це може дати зловмисникам доступ до конфіденційної інформації, що зберігається в базі даних, наприклад, облікових даних користувача, фінансової інформації або інших важливих даних.

3) Міжсайтовий скриптинг (XSS): XSS-атаки полягають у впровадженні шкідливого коду на веб-сторінку, який потім виконується в браузері користувача. Це може бути використано для крадіжки даних користувача, перехоплення сеансів користувача або запуску інших типів атак.

4) Шкідливе програмне забезпечення - це тип програмного забезпечення, призначеного для проникнення в комп'ютерну систему і заподіяння шкоди. Шкідливе програмне забезпечення може інфікувати веб-сервери через уразливості в серверному програмному забезпеченні або веб-додатках, розміщених на сервері. Після встановлення шкідливе програмне забезпечення може використовуватися для крадіжки даних, здійснення атак на інші системи або інших шкідливих дій.

5) Несанкціонований доступ: Несанкціонований доступ до веб-сервера може статися, коли зловмисники отримують доступ до адміністративного інтерфейсу сервера або інших чутливих областей сервера. Це може дозволити зловмисникам викрасти дані, змінити веб-вміст або отримати контроль над сервером.

Аналіз уразливості веб-сервера - це важливий аспект веб-безпеки, який охоплює виявлення й оцінювання потенційних уразливостей і слабких місць у веб-серверах. Під час проведення аналізу вразливості веб-сервера слід дотримуватися таких принципів:

1) Визначте цільовий веб-сервер: Першим кроком в аналізі вразливостей веб-сервера є визначення цільового веб-сервера. Це включає визначення типу сервера, операційної системи та використовуваної платформи веб-додатків.

2) Провести тестування на проникнення: Тестування на проникнення включає в себе спробу використання вразливостей веб-сервера, щоб визначити, чи

можуть вони бути успішно використані зловмисниками. Це тестування має проводитися в контрольованому середовищі, щоб не завдати шкоди веб-серверу або навколишній інфраструктурі.

3) Проведіть сканування портів: Сканування портів проводиться для визначення відкритих портів на цільовому веб-сервері. Ця інформація використовується для визначення потенційних точок входу для зловмисників.

4) Проведіть сканування вразливостей: Перевірка вразливостей проводиться для виявлення відомих вразливостей і слабких місць у цільовому веб-сервері. Для цього використовуються автоматизовані інструменти для сканування сервера на наявність таких поширених вразливостей, як SQL-ін'єкції, міжсайтовий скриптинг (XSS) та інші типи вразливостей веб-додатків.

5) Проаналізуйте результати: Після завершення сканування вразливостей необхідно проаналізувати отримані результати, щоб визначити ступінь серйозності виявлених вразливостей. Аналіз має включати оцінку потенційного впливу кожної вразливості, а також імовірність її використання.

6) Усунення вразливостей: Наступним кроком є усунення всіх виявлених вразливостей. Це може включати застосування виправлень, оновлення програмного забезпечення або налаштування параметрів безпеки.

7) Моніторинг і обслуговування: Нарешті, для того щоб веб-сервер залишався безпечним протягом довгого часу, необхідно здійснювати постійний моніторинг і обслуговування. Це охоплює застосування оновлень і виправлень у міру їхньої появи, моніторинг нових вразливостей і впровадження додаткових заходів безпеки в міру необхідності.

Існує багато інструментів і методів, які можна використовувати для аналізу вразливостей веб-серверів, в тому числі[4]:

1) Сканери вразливостей: Це автоматизовані інструменти, які можуть сканувати веб-сервер на наявність відомих вразливостей і проблем з безпекою. Деякі популярні сканери вразливостей включають OpenVAS, Nessus і Nikto.

2) Інструменти тестування на проникнення: Інструменти тестування на проникнення використовуються для імітації атак на веб-сервер з метою виявлення вразливостей і слабких місць, які можуть бути використані зловмисниками. Прикладами інструментів тестування на проникнення є Metasploit, Burp Suite та Acunetix.

3) Брандмауери веб-додатків (WAF): WAF призначені для захисту веб-додатків від цілого ряду атак, включаючи SQL-ін'єкції, міжсайтовий скриптинг (XSS) та інші вразливості. Деякі популярні WAF включають ModSecurity, Barracuda WAF і Cloudflare WAF.

4) Інструменти аналізу логів: Інструменти аналізу логів можна використовувати для моніторингу логів веб-сервера на наявність ознак підозрілої активності, таких як незвичні шаблони трафіку, спроби доступу до вразливих областей сервера або відомі сигнатури атак.

5) Ручне тестування: Ручне тестування передбачає використання комбінації автоматизованих інструментів і ручних методів для виявлення вразливостей на веб-сервері. Це може включати такі методи, як перегляд коду, фаззінг і соціальна інженерія.

Захист веб-сервера - це важливий аспект, який необхідно враховувати під час проектування, розробки та експлуатації веб-додатків. Рекомендується дотримуватися рекомендацій виробників веб-серверів та забезпечувати постійний моніторинг та аналіз захисту сервера для мінімізації ризиків та забезпечення безпеки веб-сервера. Важливо відзначити, що жоден інструмент або метод не може гарантувати повну безпеку веб-сервера. Комплексний підхід до безпеки веб-сервера передбачає поєднання інструментів, процесів і найкращих практик для мінімізації ризику вразливостей і атак. Також необхідно переконатися, що веб-сервер використовує останні версії програмного забезпечення та оновлення безпеки для запобігання вразливості та атакам. Регулярне резервне копіювання даних та перевірка наявності вразливостей також допоможуть забезпечити безпеку веб-сервера.

Список використаних джерел:

1. MDN Web Docs: What is a web server? URL: https://developer.mozilla.org/en-US/docs/Learn/Common_questions/Web_mechanics/What_is_a_web_server/
2. Types of Web Servers. URL: https://iq.opengenus.org/types-of-web-servers/.
3. Маєвський О. В. ВРАЗЛИВОСТІ ВЕБ-СЕРВЕРІВ. URL: https://elartu.tntu.edu.ua/bitstream/123456789/13457/2/VseukrStud_2012v1_Shved_Kh-Vrazlyvosti_veb_serveriv_102.pdf.
4. Website analysis: your go-to optimization resource. URL: https://www.hotjar.com/website-analysis/.

Ключові слова: Веб-сервер, інструмент, вразливості, загроза, сканування.
Keywords: Web server, tool, vulnerabilities, threat, scan.

Науковий керівник: доцент Бойко В.Д.

Секція 3. УПРАВЛІНСЬКІ ТА ЕКОНОМІЧНІ АСПЕКТИ РОЗВИТКУ СУЧАСНОЇ ІТ ІНДУСТРІЇ

ДИРЕКТИВА ЄВРОПЕЙСЬКОГО СОЮЗУ 2000/31/ЕС ПРО ЕЛЕКТРОННУ ТОРГІВЛЮ

Бугаєвська Є. Є.

студентка 1-го курсу факультету цивільної та господарської юстиції Національного університету «Одеська юридична академія»

17 червня 2022 року Європейська комісія рекомендувала Європейській раді надати Україні статус кандидата на вступ до Європейського Союзу, а 23 червня 2022 Європейський парламент ухвалив резолюцію із закликом невідкладно надати статус кандидата на членство в Європейському Союзі для України. Найважливішою умовою майбутнього членства України у ЄС є адаптація законодавства України до законодавства Європейського Союзу и наразі Україна активно працює над адаптацією свого законодавства до вимог ЄС.

У зв'язку з цим актуальним є докладне вивчення напрацьованих за роки існування ЄС законодавчих ініціатив та рішень, вивчення досвіду їх практичної реалізації та проведення порівняльного аналізу законодавства України та Європейського Союзу.

Розвиток електронної комерції стає все більш важливим, і відбувається дуже швидко. Найбільші компанії світу починають використовувати електронну комерцію, поряд із традиційним бізнесом.

Суспільство входить в еру складних комунікаційних процесів, які визначають місце держав у світі та на світовому ринку товарів і послуг. Розвиток інформаційно-комунікаційних технологій кардинально змінив світ і надав нові можливості розвитку людської цивілізації.

Однією з основних тенденцій інформатизації суспільства є стрімкий розвиток Інтернету, який впливає на практично всі сфери життєдіяльності, включаючи економіку, державне управління, науку та мистецтво.

Електронна комерція – це галузь цифрової економіки, яка включає в себе всі фінансові та комерційні операції, що здійснюються через комп'ютерні мережі, а також бізнес-процеси, що пов'язані з цими операціями.

Незважаючи на те, що електронна комерція має глобальний та розгалужений характер у сучасному світі, багато відносин між її учасниками не регулюються спеціальними законами або іншими джерелами права.

Суспільство характеризується певним ступенем організованості, що вимагає узгодження потреб та інтересів окремих людей та співтовариства. Необхідність

правового регулювання виникає тоді, коли з'являються нові соціально значущі відносини, які при їх відсутності можуть виникнути проблеми. Перехід від паперової до електронної форми інформаційних відносин суттєво змінив економічні відносини та виробництво. Ця трансформація має величезне значення та масштаб.

Україна, ратифікувавши Угоду про асоціацію з Європейським Союзом 16 вересня 2014 року, зобов'язалася забезпечити поступову адаптацію свого законодавства до законодавства ЄС відповідно до напрямків, визначених в Угоді, включаючи електронну комерцію.

Інституції Європейського Союзу вперше звернули увагу на правові аспекти електронної комерції в середині 1990-х років.

У 1997 році Європейська Комісія видала Повідомлення про електронну комерцію, у якому повідомляється про необхідність створення спільних правових засад для регулювання електронної комерції в межах ЄС до 2000 року. Внаслідок роботи Комісії, стало прийняття в 1999 році Директиви про електронні підписи, у 2000 році були прийняті Директиви про електронну комерцію, а згодом у 2001 році - Директиви про Інформаційне Суспільство [1].

Серед зазначених вище Директив, що відносяться до правового регулювання ЄС у сфері електронної комерції, Директива про електронну комерцію посідає центральне, головне місце.

Вона має рамковий характер і включає в себе норми з різних питань, що потребували вирішення в ЄС на момент її прийняття, що зазвичай потребує кількох нормативних актів в інших правових системах.

Ця Директива регулює надання послуг інформаційного суспільства як між юридичними особами, так і між юридичними особами та фізичними особами-споживачами.

Директива охоплює широкий спектр послуг, включаючи продаж фізичних товарів (книг, компакт-дисків, ліків), цифрових продуктів (електронних книг, програмного забезпечення) та інформаційних і фінансових послуг [2].

Основною метою Директиви є усунення правових перешкод, які можуть виникати при створенні провайдерів Інтернет-послуг в межах ЄС та наданні ними послуг на спільному ринку, щоб розвиток електронної комерції в ЄС став можливим. Директива застосовується широко і торкається всіх галузей права - як публічного, так і приватного – що стосуються надання послуг інформаційного суспільства.

Електронна комерція, що розвивається в рамках інформаційного суспільства, відкриває широкі можливості для зайнятості на малих та середніх

підприємствах в ЄС, стимулює економічне зростання та інвестиції в нові вироби від європейських компаній, а також підвищує конкурентоспроможність промисловості ЄС завдяки загальному доступу до Інтернету.

Поширеним є також розуміння, що електронна комерція охоплює укладання різноманітних угод за допомогою електронних документів, таких як купівля-продаж, поставка, розподіл продукції, агентські відносини, факторинг, лізинг, консалтинг, інжиніринг, страхування, банківські послуги, транспортні послуги та біржові угоди.

Тобто, в юридичній літературі немає єдиного визначення електронної комерції, оскільки це поняття охоплює широке коло відносин. Однак, ми дотримуємося думки, що загалом, електронна комерція відноситься до купівлі-продажу товарів та послуг через електронні мережі, включаючи Інтернет.

Електронна торгівля має багато переваг, однак вона також має свої ризики, такі як ухилення від податків, шахрайство та порушення прав інтелектуальної власності. Незважаючи на це, розвинені країни успішно регулюють ці ризики, щоб не завдати шкоди електронній торгівлі. Для цього уряди та приватні особи формують коаліції та співтовариства [3].

В Україні існують спеціальні підрозділи електронної розвідки та протидії комп'ютерним злочинам, які борються з електронною злочинністю, включаючи електронну комерцію. Однак, важливо звернути увагу на питання підготовки правоохоронних органів до використання нових інформаційних технологій, оскільки комп'ютерні технології можуть фіксувати докази злочинів не тільки в Інтернеті.

Таким чином, незважаючи на те, що Інтернет є світовою інформаційною системою, електронна комерція з погляду права не набула настільки "світового" характеру і традиційно продовжує залишатися в рамках національної юрисдикції. Проте, з ростом складності складу учасників процесу, проблема вибору права стає більш актуальною.

Для того, щоб розвивати електронну комерцію в Україні, потрібно закріплювати її основи у законодавстві, розвивати правове регулювання та вносити зміни до чинного законодавства. Необхідно створити режим, який найбільше сприятиме розвитку електронної комерції, що буде забезпечувати необхідний фундамент для подальшого розвитку суспільства в XXI столітті.

Список використаних джерел

1. A European Initiative on Electronic Commerce, COM(97), 157 final. Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures.
2. Directive 2001/29/EC of the European Parliament and of the Council of 22 May 2001 on the harmonisation of certain aspects of copyright and related rights in the information society.
3. Директива 2000/31/ЄС Європейського парламенту та Ради "Про деякі правові аспекти інформаційних послуг, зокрема, електронної комерції, на внутрішньому ринку" ("Директива про електронну комерцію") від 8 червня 2000 року

Ключові слова: електронна комерція, Директива, Європейській Союз

Keywords: electronic commerce, Directive, European Union

Науковий-керівник: доцент Чанишев Р. І.

ЗМІСТ

ІНФОРМАЦІЙНЕ СУСПІЛЬСТВО: ПРОБЛЕМИ ТА ПЕРСПЕКТИВИ

МАТЕРІАЛИ VIII ВСЕУКРАЇНСЬКОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ

2 червня 2023 року
м. Одеса

Відповідальній редактор: Н. І. Логінова

Дизайнер: Р. Р. Дробожур

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
«ОДЕСЬКА ЮРИДИЧНА АКАДЕМІЯ»

ФАКУЛЬТЕТ КІБЕРБЕЗПЕКИ
ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА ІНФОРМАЦІЙНИХ
ТЕХНОЛОГІЙ

МАТЕРІАЛИ VIII ВСЕУКРАЇНСЬКОЇ НАУКОВО-ПРАКТИЧНОЇ КОНФЕРЕНЦІЇ

ІНФОРМАЦІЙНЕ СУСПІЛЬСТВО: «ПРОБЛЕМИ ТА ПЕРСПЕКТИВИ»

2 червня 2023 року, м. Одеса