Отже, розроблений програмний засіб демонструє можливість практичного використання СЗК для модифікації класичних криптографічних алгоритмів. Поєднання криптографічного перетворення з багатомодульним представленням даних дозволяє ускладнити структуру шифротексту, розподілити інформацію між кількома модулями та створити основу для паралельної обробки. Подальші дослідження доцільно спрямувати на поглиблений криптоаналіз запропонованого методу, оптимізацію вибору модулів, порівняння з сучасними симетричними алгоритмами та реалізацію обчислень на апаратних платформах, зокрема FPGA або GPU.

1. 1. Nieles M., Dempsey K., Pillitteri V. *An Introduction to Information Security*. Gaithersburg : NIST, 2017. 101 p..
2. 2. Kasianchuk M. M., Yakymenko I. Z., Nykolaychuk Y. M. Symmetric Cryptoalgorithms in the Residue Number System. *Cybernetics and Systems Analysis*. 2021. Vol. 57. P. 329–336. https://doi.org/10.1007/s10559-021-00358-6.
3. 3. Nykolaychuk Ya. M., Yakymenko I. Z., Vozna N. Ya., Kasianchuk M. M. Residue Number System Asymmetric Cryptoalgorithms. *Cybernetics and Systems Analysis*. 2022. Vol. 58, No. 4. P. 611–618. https://doi.org/10.1007/s10559-022-00494-7.

Проєктування захищеної архітектури для оцінювання ігор LUDARA з використанням технології Node.js та принципів Security-by-Design

УДК 004.42

Мороз Даниїл[1], Мудрик Іван[2]

Тернопільський національний технічний університет імені Івана Пулюя,
[1]danyil_moroz1301@tntu.edu.ua, [2]i1mudryk@tntu.edu.ua

Теперішній ринок відеоігор демонструє стрімке зростання: за даними аналітичних агентств, кількість активних гравців у світі перевищує 3 мільярди осіб, а обсяг галузі щорічно збільшується на 8–10% [1]. Попри це, існуючі платформи для оцінювання ігор стикаються з низкою проблем інформаційної безпеки: ризиками компрометації персональних даних користувачів, вразливостями відкритих API, а також маніпуляцією рейтингами за допомогою автоматизованих скриптів (ботів). Метою роботи є проєктування та розробка безпечної архітектури веб-платформи «Ludara», що базується на принципах Security-by-Design із використанням технологічного стеку Node.js.

Архітектура платформи реалізована за принципом чіткого розмежування рівнів доступу та відповідальностей. Серверна частина побудована на Node.js з фреймворком Express.js, клієнтська — на React з використанням Vite [2]. Для гарантування цілісності даних обрано реляційну СУБД PostgreSQL. Взаємодія з базою даних здійснюється через ORM-бібліотеку Prisma, що автоматично параметризує запити, унеможливлюючи атаки типу SQL-ін'єкцій, та забезпечує типобезпечність. Автентифікація реалізована за протоколом на основі JWT-

токенів (JSON Web Tokens). Для протидії атакам типу XSS та підміні даних застосовується сувора валідація та санітизація всіх вхідних параметрів за допомогою бібліотеки Zod.

З метою мінімізації площі можливих атак (attack surface) та оптимізації зберігання даних, інтеграція з RAWG API реалізована за принципом лінивого завантаження (lazy loading). Система не зберігає локально масив із сотень тисяч ігор [3], а виконує живі запити до зовнішнього API. При першій взаємодії користувача з грою система перевіряє її наявність у локальній базі, після чого безпечно отримує та кешує метадані. Такий підхід не лише зменшує обсяг потенційно вразливих даних у власній БД, а й забезпечує актуальність інформації.

Для управління доступом у платформі реалізовано строгу модель Role-Based Access Control (RBAC) із трирівневою системою: звичайний користувач, критик та адміністратор. Це дозволяє криптографічно на рівні токенів розмежувати права на виставлення оцінок, публікацію рецензій та модерацію контенту, чітко відділяючи експертну думку від користувацької. Завантаження та зберігання медіафайлів (зображень профілів) ізольовано від основного сервера та делеговано захищеному хмарному сервісу Cloudinary [4].

Ключовим архітектурним рішенням є інтеграції з RAWG API. Каталог ігор та пошук працюють у режимі живих запитів до зовнішнього API, що дозволяє уникнути необхідності зберігати та синхронізувати масив із сотень тисяч ігор локально. Натомість власна база даних платформи містить лише ті ігри, з якими користувачі вже взаємодіяли. При першій взаємодії — виставленні оцінки чи зміні статусу — система перевіряє наявність гри в локальній базі за унікальним ідентифікатором RAWG, і якщо запис відсутній, автоматично отримує метадані з API та зберігає їх. Такий підхід суттєво зменшує обсяг даних у власній БД і водночас забезпечує актуальність інформації про ігри. Додатково реалізовано фільтрацію результатів: виключаються DLC, доповнення, моди та інший небажаний контент за допомогою параметрів запиту та перевірки тегів на стороні сервера.

Платформа реалізує трирівневу систему ролей: звичайний користувач, критик та адміністратор. Користувачі можуть виставляти оцінки за шкалою 1–10, писати рецензії та лайкати відгуки інших. Рецензії критиків виділяються окремим блоком на сторінці гри, що дозволяє чітко розмежувати експертну та користувацьку думку. Реалізовано систему статусів ігор («Хочу пройти», «Проходжу», «Пройдено»), персональну статистику профілю з аналізом улюблених жанрів, а також рейтинг Топ-100 ігор за оцінками користувачів платформи. Зображення профілів зберігаються через хмарний сервіс Cloudinary із автоматичною оптимізацією розміру [4].

Окрему увагу в розробці приділено прикладному захисту веб-інфраструктури. Реалізовано захист від несанкціонованого доступу через спеціалізовані middleware-компоненти для верифікації JWT-токенів на кожному маршруті. Для запобігання атакам типу Brute-force та DDoS на рівні API-ендпоінтів впроваджено механізм обмеження частоти запитів (rate limiting). Адміністративна панель забезпечує безпечний моніторинг інцидентів:

управління життєвим циклом користувачів, застосування банів та модерацію підозрілого контенту.

Розроблена платформа демонструє ефективність застосування підходу Security-by-Design при побудові сучасних веб-застосунків на основі Node.js. Запропонована захищена архітектура із застосуванням RBAC, строгим контролем вхідних даних та ізоляцією медіа-контенту забезпечує стійкість системи до поширених веб-вразливостей і може бути масштабована для інших високонавантажених соціальних платформ.

1. Newzoo Global Games Market Report 2024. — Режим доступу: https://newzoo.com/resources/trend-reports/newzoos-global-games-market-report-2024-free-version
2. Cantelon M., Harter M. Node.js in Action. — Manning Publications, 2017. — 392 p.
3. RAWG Video Games Database API Documentation. — Режим доступу: https://rawg.io/apidocs
4. Martin R. C. Clean Architecture: A Craftsman's Guide to Software Structure and Design. — Prentice Hall, 2017. — 432 p.
5. Bryk O., Mudryk I., Holubovskyi M., Stoianov Y. Machine learning models and methods aspects of processing unstructured data. Proceedings of the 1st International Workshop on Bioinformatics and Applied Information Technologies (BAIT 2024), Zboriv, Ukraine, 2024. 2024. P. 64–74.

Конвергенція кіберсуб'єктів національних держав та організованої кіберзлочинності

УДК 004.056:355.4

Світлана Легомінова[1], Тетяна Капелюшна[2], Тетяна Мужанова[3]

Державний університет інформаційно-комунікаційних технологій,
[1]s.legominova@duikt.edu.ua, [2]t.kapeliushna@duikt.edu.ua,
[3]t.muzhanova@duikt.edu.ua

Як свідчать реалії, в сучасному швидкозмінному кіберландшафті відмінності між кіберсуб'єктами національних держав і організованими кіберзлочинцями стають дедалі більш розмитими. На попередніх етапах ці суб'єкти мали різні мотиви: національні держави прагнули досягти довгострокових геополітичних переваг за допомогою шпигунських і розвідувальних операцій, у той час як кіберзлочинці прагнули отримати фінансову вигоду, експлуатуючи вразливості ІКС для вимагання, крадіжок і шахрайства.

Однак упродовж кількох останніх років відзначено тривожну тенденцію щодо злиття тактик, методів і цілей цих суб'єктів, що ускладнює їх розмежування, можливості виявлення і притягнення винних до відповідальності (кібератрибуції), а також виносить на порядок денний критичні питання щодо стрімкої еволюції кіберзагроз та її критичних наслідків для глобальної безпеки.