УДК 004.942
DOI: https://doi.org/10.17721/1812-5409.2024/1.29

Ірина ЗАМРІЙ, д-р техн. наук, доц.
ORCID ID: 0000-0001-5681-1871
e-mail: irinafraktal@gmail.com
Державний університет інформаційно-комунікаційних технологій, Київ, Україна

Іван ШАХМАТОВ, асп.
ORCID ID: 0009-0004-9628-0365
e-mail: ivan.shakhmatov@gmail.com
Державний університет інформаційно-комунікаційних технологій, Київ, Україна

Владислав ЯСКЕВИЧ, канд. техн. наук
ORCID ID: 0000-0002-5796-2521
e-mail: v.yaskevitch@gmail.com
Київський столичний університет імені Бориса Грінченка, Київ, Україна

BLOCKCHAINSQLSECURE: ІНТЕГРАЦІЯ БЛОКЧЕЙН-ТЕХНОЛОГІЇ ДЛЯ ЗМІЦНЕННЯ ЗАХИСТУ ВІД SQL-ІН'ЄКЦІЙ

У сфері веброзробок зросла необхідність у засобах захисту від SQL-ін'єкцій, що можуть мати катастрофічні наслідки для баз даних. У пропонованій статті вводиться концепція BlockchainSQLSecure, унікальний метод, що використовує можливості блокчейн-технологій для створення додаткового рівня безпеки в Django-застосунках. Центральна ідея полягає у створенні блокчейн-журналу для кожного SQL-запиту до бази даних, що дає змогу гарантувати незмінність і відстежуваність інформації. Ця прозорість активно протидіє можливості несанкціонованих змін чи спроб ін'єкцій, оскільки будь-яке втручання стає відразу помітним.

У межах статті досліджено механізми валідації SQL-запитів через smart-контракти на блокчейні, які дають змогу автоматично відхиляти запити, що містять потенційні ін'єкції. Також розглянуто методи децентралізованого зберігання журналів, які забезпечують розподілення інформації між учасниками мережі, надаючи системі стійкості до атак і спроб зовнішнього втручання.

Додатково BlockchainSQLSecure може бути реалізований як плагін для Django, що полегшує його інтеграцію в наявні проєкти. Такий підхід до інтеграції робить алгоритм системи відмінним вибором для команд, які вже використовують Django у своїх проєктах, і шукають ефективних шляхів оптимізації оброблення SQL-запитів та підвищення продуктивності своєї системи. Завдяки легкості впровадження та використання, система є гнучким і потужним рішенням, що може принести значну користь будь-якому Django-проєкту. В узагальненому вигляді, представлена концепція показує, як комбінація традиційних методів захисту та сучасних блокчейн-технологій може забезпечити новий рівень безпеки для вебзастосунків.

К л ю ч о в і с л о в а : ***блокчейн, Django, вебзастосунки, безпека, інтеграція, обмеження доступу.***

AMS 2020 класифікація: 68M25, 68P27.

Вступ

SQL-ін'єкції продовжують залишатися серйозною загрозою в галузі веброзроблень, виявляючи недоліки в системах і надаючи можливість недобросовісним особам отримувати несанкціонований доступ до даних. Цей вид атак передбачає введення шкідливого SQL-коду через інтерфейс користувача, що може призвести до несанкціонованого доступу, витоку або втрати даних. Зловмисники цілеспрямовано шукають вразливі місця в захисті, щоб отримати доступ до конфіденційної інформації, видаляти критично важливі дані чи навіть узяти під контроль сервери.

Недавні інциденти підкреслюють актуальність проблеми. У 2021 р. велика технологічна компанія "SolarWinds" стала жертвою складної кібератаки, що містила SQL-ін'єкції. Зловмисники використовували недоліки в безпеці для втручання в роботу програмного забезпечення, що призвело до втручання в роботу численних урядових і корпоративних мереж (Oladimeji, & Kerner, 2023).

У 2022 р. було виявлено вразливість в одному з популярних плагінів для "WordPress", що дала змогу зловмисникам виконувати SQL-ін'єкції й отримувати доступ до даних сайту. Цей випадок укотре продемонстрував, що навіть широко використовувані та популярні інструменти не є імунітетними до атак, та підкреслив важливість ретельного тестування й оновлення програмного забезпечення (Chamberland, 2022).

У 2021 р. компанія "Verkada", що спеціалізується на розробленні систем відеоспостереження, стала мішенню хакерської атаки, у ході якої було використано SQL-ін'єкції для отримання доступу до приватних відеопотоків тисяч компаній. Цей інцидент показав, наскільки важливою є безпека додатків, що обробляють чутливі дані, і як недостатній захист може призвести до серйозних наслідків (Anderson, 2021).

Ці й інші подібні інциденти наголошують на критичній важливості захисту вебдодатків від SQL-ін'єкцій. Вони вказують на необхідність розвитку й упровадження нових, надійніших методів захисту, що здатні протистояти такому виду атак. У цьому контексті використання інноваційних технологій, таких як блокчейн, відкриває нові горизонти для зміцнення безпеки вебдодатків і захисту їх від SQL-ін'єкцій.

1. Огляд останніх досліджень і публікацій. Мережева безпека є ключовим аспектом у забезпеченні захисту інформаційних систем, особливо у сферах, де оброблення конфіденційної інформації – невід'ємна частина повсякденної діяльності. Один з головних викликів у цій галузі – боротьба з атаками SQL Injection, які можуть призвести до несанкціонованого доступу та маніпуляції даними. З метою виявлення та протидії таким атакам розроблено підхід, що передбачає використання нейронних мереж для аналізу мережевого трафіку й ідентифікації потенційно шкідливих запитів (Sobchuk, Zamrii, & Laptiev, 2023). Цей метод заснований на вивченні патернів поведінки користувачів і системи, допомагаючи таким способом виявляти аномалії, що можуть вказувати на спробу втручання в роботу системи. Учені глибоко аналізують аспекти забезпечення ефективності й безпеки системи, пропонуючи цінні інсайти та підходи для побудови надійних і безпечних вебсистем (Nagabhooshanam et al., 2023).

ISSN 1728-3817

Одним із важливих інструментів протидії атакам є mURLi, який був розроблений з метою ідентифікації шкідливих URL-адрес і виявлення атак ін'єкцій, що цілять у бази даних SQL та NoSQL. Використання інструментів, подібних до mURLi, може відігравати ключову роль у процесі ідентифікації та блокуванні потенційних загроз на ранніх стадіях їх розвитку (Devalla et al., 2022).

Архітектура, орієнтована на блокчейн, може стати ключовим елементом у забезпеченні безпеки від SQL-ін'єкцій та інших видів атак. Однією з переваг такої архітектури, зокрема, є здатність забезпечувати цілісність і консистентність даних щодо характеристик атак, що обмінюються між вузлами системи виявлення вторгнень. Отже, можна забезпечити постійний моніторинг поведінки вузлів у мережі, виявляючи та запобігаючи будь-якій зловмисній діяльності, що може бути спрямована як ззовні, так і зсередини мережі (Tanrıverdi, & Tekerek, 2021). Можливість інтеграції гетерогенних систем виявлення вторгнень може стати фундаментальним елементом для розроблення DjangoSQLSecure, даючи змогу інтегрувати блокчейн у Django-застосунки з метою зміцнення захисту від SQL-ін'єкцій та інших потенційних загроз (Tanrıverdi, & Tekerek, 2021).

У моделі RSFSA використовується для модифікації референтної частини блокчейну, щоб з'єднати лише ті блоки, до яких користувач має доступ (Siva, Godfrey, & Ramesh, 2021). Цей підхід може бути інтегрований у DjangoSQLSecure, щоб поліпшити захист від SQL-ін'єкцій у Django-застосунках. Використовуючи блокчейн для забезпечення безпеки даних та управління доступом, можна створити надійнішу та більш захищену систему, що є критично важливим для додатків, які обробляють чутливу інформацію.

Використання блокчейну для процесів аутентифікації та верифікації документів пропонує додатковий рівень безпеки, що може сприяти створенню безпечнішого та надійнішого середовища для користувачів і розробників (Aini et al., 2022).

Модель системи шифрування баз даних, запропонована в дослідженні (Guanxiu, 2022), містить п'ять логічних модулів: модуль зберігання ключів, модуль ключового двигуна, модуль інформації про ключі, модуль управління ключами та модуль зберігання даних. Кожен із цих модулів відіграє ключову роль у забезпеченні безпеки даних та інтегритету інформації. Функціональність і продуктивність запропонованої системи шифрування були підтверджені в ході експериментів, проведених на онлайн-системі іспитів. Результати демонструють, що система здатна забезпечити ефективний захист даних за ефективності передавання даних понад 95 %, що підкреслює її придатність для використання в реальних умовах.

У роботі (Chen et al., 2021) розглянуто питання безпеки даних у сфері великих даних, пропонуючи новий підхід до шифрування. Цей метод інтегрує в себе технологію "Bloom filter", що дає змогу видаляти дублікати даних ще до процесу шифрування, тим самим підвищуючи швидкість й ефективність захисту інформації.

Важливим напрямком у забезпеченні безпеки даних є інтеграція блокчейн-технологій безпосередньо з реляційними базами даних. У роботі (Awadallah, & Samsudin, 2021) представлено структуру "BC over cloud-RDB", що об'єднує переваги блокчейн-технологій та можливості хмарних обчислень для реляційних баз даних. "BC over cloud-RDB" включає в себе дві ключові системи: "agile BC-based RDB" та "secure BC-based RDB". "Agile BC-based RDB" призначена для баз даних, яким необхідна висока пропускна спроможність, тоді як "secure BC-based RDB" фокусується на зберіганні чутливої інформації та має низьку пропускну спроможність. Для забезпечення надійності і безпеки, обидві системи використовують консенсус "Byzantine Fault Tolerance", і записи зв'язуються за допомогою хеш-функції SHA-256. Це дає можливість забезпечити цілісність даних і відслідковування зміни в базі даних.

Ученими представлено аналіз різних моделей машинного та глибокого навчання, які були використані для ідентифікації атак SQL-ін'єкцій (Alghawazi, Alghazzawi, & Alarifi, 2022). Автори вказують на те, що за допомогою цих технологій можна досягти разючих результатів у виявленні та запобіганні SQL-ін'єкцій, що відкриває нові можливості для захисту вебзастосунків.

Аналізу частоти та серйозності вразливостей у різних мовах програмування присвячено роботу (Sakharkar, 2023). Результати аналізу показують, що деякі мови програмування можуть бути більш схильними до певних типів вразливостей.

Важливим аспектом є доповнення наявних протоколів мережевої безпеки з використанням моделей AI, що дає змогу створити надійніші й ефективніші системи безпеки. Дослідники також надають детальний огляд загроз безпеки в мережах IoT та аналізують моделі безпеки на основі AI, які можуть бути використані для їхнього усунення (Zaman et al., 2021).

Блокчейн-технології пропонують нові методи розв'язання деяких проблем, допомагаючи забезпечити конфіденційність, цілісність і доступність даних. Вони відіграють ключову роль у зміцненні захисту інформації у хмарних середовищах, гарантуючи прозорість і незмінність даних. У роботі (Alouffi et al., 2021) підкреслено, що використання блокчейну в хмарних обчисленнях може зменшити загрози безпеки та підвищити надійність системи загалом.

Інтеграція технології розподіленого реєстру (DLT) та розумних контрактів з інформаційним моделюванням будівель (BIM), інтернетом речей (IoT) та хмарними обчисленнями відкриває нові можливості для створення більш інтегрованих, автоматизованих та інтелігентних систем управління будівельними проєктами. Водночас, незважаючи на значний потенціал і переваги, що пропонують DLT та розумні контракти, існує необхідність у подальших дослідженнях й експериментальних перевірках цих технологій у реальних умовах експлуатації для визначення найбільш ефективних підходів до їхнього впровадження та використання (Li, & Kassem, 2021).

Новітні архітектури DLT та дозволені блокчейни демонструють більшу придатність до використання у промислових та зрілих сценаріях, де вимоги до безпеки та відмовостійкості є надзвичайно високими (Queralta et al., 2023). Ці технологічні нововведення можуть сприяти забезпеченню високого рівня автономії та координації між роботами, водночас підтримуючи необхідні стандарти безпеки та вимоги до стійкості (Sobchuk et al., 2021), що є ключовим аспектом для розвитку сучасних робототехнічних систем.

Шифрування даних, що є властивістю блокчейну, захищає конфіденційність інформації, тоді як оперативна стійкість гарантує надійність системи навіть за спроб атаки або виникнення відмов. Важливо зауважити, що обговорення цих питань нині не є суто теоретичним; існують конкретні дослідження та розробки, які вже використовують технології блокчейну для підвищення рівня безпеки в системах IoT (Tanwar et al., 2022).

Фільтри Блума дають змогу здійснювати ефективний пошук без необхідності зберігання всіх даних у локальному порядку, що є особливо важливим для вузлів з обмеженими ресурсами (Hussein, & Al-Gailani, 2023). Потреба в

оптимізації торгівлі між розміром елементів мережі та її пропускною здатністю, а також метриками приватності в недовірливих середовищах, висувається на передній план. Аналіз продуктивності фільтрів Блума, здійснений з використанням методів статистичного розподілу, стандартного відхилення, ентропії й у-заперечуваності, демонструє, яким чином ці фільтри можуть бути атаковані з метою аналізу витоку інформації. Експерименти показують, що за використання менше 50 % обсягу фільтра, середнє значення метрики анонімності залишається меншим 66 %. Такі дані підкреслюють важливість вибору оптимального балансу між розміром мережі та її пропускною здатністю для досягнення високого рівня приватності.

Важливо підкреслити, що метрика приховування (у-заперечуваність) корелює із загальною кількістю елементів у мережі, і для забезпечення високого рівня приватності в мережі необхідна велика кількість елементів і вища пропускна здатність. Усе це підкреслює роль алгоритму фільтра Блума в оптимізації блокчейн-мереж (Hussein, & Al-Gailani, 2023).

Проблеми зберігання даних й оброблення запитів у блокчейні вимагають ретельного аналізу, особливо з огляду на відсутність деяких функцій, які є звичними для систем управління базами даних (DBMS) (Kalajdjieski et al., 2023). Цей аспект є ключовим для підвищення ефективності і безпеки використання блокчейн-технологій. Розглядаючи варіанти оптимізації блокчейну, необхідно вивчити різні типи DBMS, їх характеристики та можливості застосування. У контексті блокчейну також важливо звернути увагу на гіпотезу "Decentralization, Consistency, and Scalability (DCS)-satisfiability conjecture", що висуває стратегії для забезпечення оптимального балансу між децентралізацією, узгодженістю та масштабованістю (Kalajdjieski et al., 2023). Ці стратегії можуть бути корисними для розробників Django-застосунків у їхньому прагненні інтегрувати блокчейн для зміцнення захисту від SQL-ін'єкцій і підвищення загальної безпеки системи.

Ураховуючи динаміку розвитку технологій інтернету речей (IoT) та зміни в режимах використання, системи блокчейну стикаються з новими викликами та потребами в адаптації до складніших типів запитів (Przytarski et al., 2022).

Метою пропонованого дослідження є розроблення й аналіз методу захисту від SQL-ін'єкцій, що базується на використанні блокчейн-технологій, а також інтеграція цього методу в ролі застосунка для Django – одного з найпопулярніших фреймворків для веброзробки мовою програмування Python. Важливою частиною дослідження є розроблення алгоритму, що дає змогу ефективно ідентифікувати та блокувати SQL-ін'єкції, використовуючи блокчейн для забезпечення прозорості та незмінності даних. Окрім того, розглядається можливість використання розподілених реєстрів для забезпечення децентралізації та підвищення стійкості системи до атак.

Запропоновано методологію реалізації з урахуванням оптимізації продуктивності системи та забезпечення можливості легкої інтеграції у вже наявні вебзастосунки на Django, що надає дослідженню значної актуальності та цінності для спільноти веброзробників.

2. Мета та завдання дослідження. Метою роботи є аналіз і визначення потенційних можливостей блокчейн-технологій у боротьбі із загрозами SQL-ін'єкцій у вебдодатках. SQL-ін'єкції залишаються однією з найпоширеніших і найнебезпечніших уразливостей, здатних призвести до витоку або зміни конфіденційних даних. Вивчення того, як блокчейн може допомогти у виявленні та нейтралізації таких атак, є ключовим для розроблення ефективних захисних стратегій.

Іншим важливим аспектом є розроблення практичного методу інтеграції блокчейн-технологій у структуру вебдодатків, розроблених за допомогою фреймворку Django, який уже містить низку вбудованих засобів для забезпечення безпеки, але інтеграція блокчейну може забезпечити додатковий рівень захисту. Цей метод має бути гнучким, ефективним і легким у впровадженні, щоб розробники могли швидко адаптувати його до своїх потреб.

Останнім етапом є проведення оцінки ефективності розробленого методу в різних умовах експлуатації та при різних типах SQL-ін'єкцій. Це передбачає тестування методу на різних версіях Django, а також за різних конфігурацій баз даних. Ця оцінка дасть змогу виявити переваги та недоліки методу, а також допоможе визначити сценарії, у яких його використання буде найефективнішим.

Етапи дослідження допоможуть з'ясувати, наскільки ефективно блокчейн-технології можуть бути інтегровані в захисні механізми Django-застосунків, та які переваги це може мати в боротьбі із SQL-ін'єкціями.

Завданням дослідження є аналіз наявних методів захисту вебзастосунків від SQL-ін'єкцій, оцінка переваг і недоліків, а також визначення можливих прогалин у безпеці. На базі цього аналізу планується розробити й інтегрувати новий метод захисту, заснований на використанні блокчейн-технологій, який допоможе забезпечити вищий рівень безпеки і стійкість до зовнішніх загроз.

У межах дослідження передбачено детальний аналіз механізмів блокчейну та визначення оптимальних варіантів їхньої інтеграції в системи захисту від SQL-ін'єкцій. Особливу увагу приділено розробленню архітектури для ефективного впровадження блокчейну в структуру вебзастосунків, а також проведенню попереднього тестування розробленого методу з метою виявлення потенційних недоліків і визначення шляхів для подальшої оптимізації.

Експериментальне тестування розробленого методу дасть змогу оцінити його ефективність, виявити переваги та можливі недоліки і на основі отриманих результатів визначити оптимальні умови та параметри для практичного використання методу.

Отже, пропоноване дослідження спрямоване на пошук інноваційних рішень у галузі кібербезпеки, забезпечення високого рівня захисту вебдодатків і створення умов для їхньої безпечної та стабільної роботи в умовах кіберзагроз, що постійно зростають.

3. Матеріали та методи дослідження. Модель BlockchainSQLSecure є інноваційним рішенням для питань безпеки SQL, інтегруючи технології блокчейн і фільтр Блума для виявлення та запобігання SQL-ін'єкціям. Вона вирізняється своїм підходом, не обмежуючись відстеженням типових патернів SQL-ін'єкцій, а замість цього використовуючи блокчейн для надійного зберігання інформації про попередні запити. Це дає змогу створити стійкий до змін журнал, який можна перевірити в будь-який момент часу для забезпечення безпеки.

У контексті взаємодії між клієнтом і сервером BlockchainSQLSecure вносить інновації, записуючи кожен SQL-запит у блокчейні, що дає можливість пізніше перевірити цей запис на наявність неавторизованих змін або спроб ін'єкції. Фільтр Блума в цій моделі використовують для швидкого визначення і видалення дублікатів даних, що значно підвищує ефективність процесу виявлення вже оброблених запитів і потенційних спроб ін'єкції.

Модель не вимагає значних обчислювальних ресурсів для своєї роботи і може бути легко інтегрована у вже наявні системи. Фільтр Блума, зі свого боку, забезпечує високу швидкість й ефективність обробки. З огляду на це він є ідеальним варіантом для роботи з великими обсягами даних.

У моделі BlockchainSQLSecure ключовим аспектом є визначення схожості між SQL-запитами для ефективного виявлення та запобігання спробам SQL-ін'єкцій. Для цієї мети використовують формулу відстані Жаккара.

Вектори X та Y є бінарними векторами, де кожен елемент відображає наявність чи відсутність певного атрибуту SQL-запиту. Наприклад, це можуть бути слова або токени, що входять до SQL-запиту.

X_i: 1, якщо -й атрибут присутній у першому SQL-запиті, і 0 в іншому випадку.

Y_i: 1, якщо -й атрибут наявний у другому SQL-запиті, і 0 в іншому випадку.

Скалярний добуток векторів X та Y ($\sum_{i=1}^{n} X_i^2$ та $\sum_{i=1}^{n} Y_i^2$) вказує на загальну кількість атрибутів у кожному SQL-запиті. Це допомагає врахувати розміри множин, з якими ми працюємо.

Для оцінки схожості між множинами в моделі використаємо формулу відстані Жаккара,

$$Jaccard(X,Y) == \frac{\sum_{i=1}^{n} X_i * Y_i}{\sum_{i=1}^{n} X_i^2 + \sum_{i=1}^{n} Y_i^2 - \sum_{i=1}^{n} X_i * Y_i}, \quad (1)$$

де X_i, Y_i – елементи векторів X та Y відповідно, $\sum_{i=1}^{n} X_i * Y_i$ – скалярний добуток векторів X та Y, $\sum_{i=1}^{n} X_i^2, \sum_{i=1}^{n} Y_i^2$ – квадрати довжин векторів X та Y відповідно.

Це дає змогу обчислити (1) схожість між двома SQL-запитами, де близьке до 1 значення вказує на високу схожість, а близьке до 0 – на низьку схожість. Ця відстань є ключовим елементом в алгоритмі видалення дублікатів даних на основі фільтра Блума, підвищуючи ефективність і точність роботи моделі.

У контексті BlockchainSQLSecure ця метрика допомагає швидко визначити, чи не є новий SQL-запит варіантом уже наявного, що містить потенційні спроби ін'єкцій. Застосування цієї формули в комбінації із фільтром Блума підвищує ефективність і точність системи у виявленні та запобіганні SQL-ін'єкцій.

Зазначений підхід дає змогу моделі ефективно працювати навіть з великими обсягами даних, не вимагаючи значних обчислювальних ресурсів, що робить її легко інтегрованою у уже наявні системи.

Завдяки використанню блокчейну і фільтра Блума, BlockchainSQLSecure пропонує революційний підхід до забезпечення безпеки SQL, ефективно відсікаючи можливість ін'єкцій та інших загроз.

Алгоритм роботи BlockchainSQLSecure інтегрує в собі декілька ключових етапів для забезпечення високого рівня безпеки SQL-запитів у вебзастосунках (рис. 1). Від початкового прийому запиту до його фінального виконання, система ретельно перевіряє кожен аспект, використовуючи сучасні технології блокчейну і фільтра Блума.

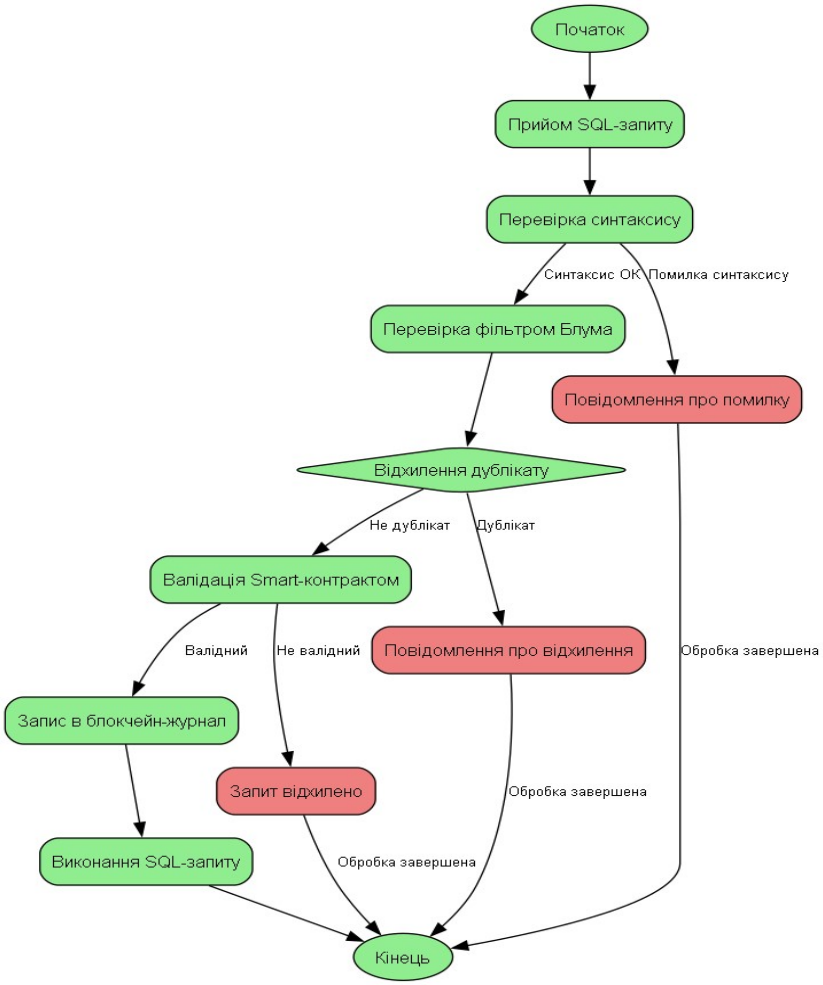

Рис. 1. Алгоритм роботи BlockchainSQLSecure

На самому початку, коли SQL-запит надходить від користувача чи програми, система одразу ж проводить перевірку вхідних даних. Це передбачає аналіз на наявність помилок чи некоректного синтаксису. Такий підхід дає можливість відсіяти неправильно сформульовані запити на ранньому етапі, економлячи ресурси системи і підвищуючи її загальну продуктивність.

Далі перевірений SQL-запит направляється на оброблення фільтром Блума. Цей крок є критичним для ефективного виявлення і відхилення дублікатів запитів, які раніше були класифіковані як шкідливі або небажані. Завдяки фільтру Блума система швидко і з мінімальними витратами ресурсів визначає, чи був аналогічний запит уже оброблений.

Якщо фільтр Блума вказує на те, що запит уже був оброблений, система відразу відхиляє його, надсилаючи користувачу повідомлення про відхилення. В іншому випадку, SQL-запит переходить на наступний етап − валідацію за допомогою smart-контракту.

Smart-контракт, розміщений на блокчейні, проводить глибокий аналіз SQL-запиту, перевіряючи його на відповідність певним правилам і стандартам безпеки. Цей етап є ключовим для виявлення і блокування потенційних SQL-ін'єкцій, забезпечуючи те, що тільки безпечні та валідні запити досягають бази даних.

Після успішної валідації SQL-запит додається до блокчейн-журналу, забезпечуючи його незмінність і можливість відстеження. Цей крок є важливим для підвищення рівня прозорості та відповідальності системи, даючи можливість у будь-який момент перевірити історію оброблених запитів.

На завершальному етапі валідний SQL-запит виконується базою даних, і результат виконання запиту повертається користувачу. Отже, алгоритм BlockchainSQLSecure не тільки забезпечує високий рівень безпеки для SQL-запитів, але й також підвищує загальну ефективність і продуктивність оброблення даних у системі.

Запропонована UML-схема (рис. 2), демонструє структуру класів інноваційної системи, що об'єднує функціональність оброблення SQL-запитів, управління блокчейн-транзакціями, аудит дій користувачів і використання фільтра Блума для оптимізації запитів.

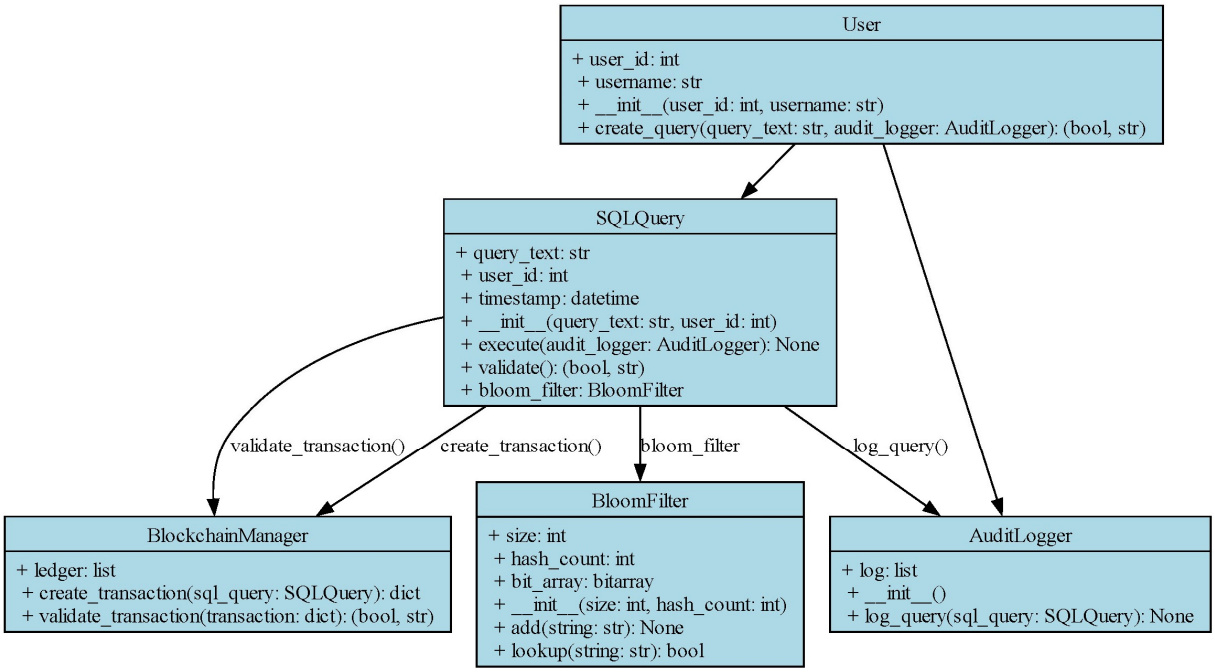

Рис. 2. UML-діаграма класів для системи управління базою даних і блокчейн-транзакціями

Клас User є представленням користувача в системі, зосереджуючи в собі всю необхідну інформацію про користувача, а також надаючи йому можливість створювати SQL-запити та взаємодіяти з іншими компонентами системи. Він відіграє важливу роль в управлінні доступом і взаємодією на рівні користувача. SQLQuery є ключовим класом для роботи з базами даних, уособлюючи модель SQL-запиту. Він зберігає текст запиту та ID користувача, який його створив, а також виконує функції валідації та виконання запитів. Цей клас є центральним у взаємодії з базою даних і відіграє ключову роль у забезпеченні правильності та безпеки SQL-операцій. BlockchainManager відповідає за управління транзакціями в блокчейні, забезпечуючи цілісність даних і ведення історії транзакцій. Цей клас відіграє важливу роль у системі, забезпечуючи безпеку та надійність оброблення даних. AuditLogger займається аудитом дій користувачів і системи. Він веде докладний журнал усіх операцій, який може бути використаний для аналізу дій, пошуку помилок та виявлення несанкціонованої активності. Фільтр Блума є ефективним інструментом для оптимізації запитів і швидкого визначення, чи був певний запит уже оброблений системою. Він дає змогу значно знизити кількість звернень до бази даних, що позитивно впливає на продуктивність системи.

За допомогою цих класів система може ефективно виконувати SQL-запити, забезпечувати цілісність даних, вести докладний аудит дій та оптимізувати процес оброблення запитів, гарантуючи високий рівень безпеки та продуктивності.

4. Результати дослідження. Унаслідок дослідження було створено UML-схему з метою забезпечення чіткої візуалізації та розуміння архітектури системи. Вона охоплює п'ять основних класів: User, SQLQuery, BlockchainManager, AuditLogger та

BloomFilter, кожен з яких відіграє ключову роль у функціонуванні системи. За допомогою UML-схеми вдалося виявити ключові зв'язки та залежності між класами, що, зі свого боку, сприяло ефективнішому процесу розробки.

Розроблення системи мовою програмування Python дозволило взяти на себе принципи чистого кодування, що забезпечило якість, легкість розуміння та можливість подальшого розширення проєкту. Спроєктовано архітектуру системи, розділивши її на модулі та класи, що дало змогу забезпечити чітку структуру та відокремлення відповідальностей.

Модуль фільтра Блума був реалізований з метою оптимізації оброблення SQL-запитів (рис. 3). Він забезпечує можливість швидкої перевірки унікальності запитів, відкидаючи ті, що вже були виконані. Така оптимізація допомагає зменшити навантаження на базу даних і, як результат, підвищує загальну продуктивність системи. Для реалізації цього модуля використано найкращі практики роботи із фільтрами Блума, забезпечивши ефективне використання ресурсів.

BloomFilter.py

```python
import mmh3
from bitarray import bitarray

class BloomFilter:
    def __init__(self, size, hash_count):
        self.size = size
        self.hash_count = hash_count
        self.bit_array = bitarray(size)
        self.bit_array.setall(0)
        self.queries = set()

    def add(self, string):
        self.queries.add(string)
        for seed in range(self.hash_count):
            result = mmh3.hash(string, seed) % self.size
            self.bit_array[result] = 1

    def lookup(self, string):
        for seed in range(self.hash_count):
            result = mmh3.hash(string, seed) % self.size
            if self.bit_array[result] == 0:
                return False
        return True

    @staticmethod
    def calculate_jaccard_similarity(query1, query2):
        set1 = set(query1.split())
        set2 = set(query2.split())
        intersection = len(set1.intersection(set2))
        union = len(set1) + len(set2) - intersection
        return intersection / union
```

Рис. 3. Запропонована реалізація класу BloomFilter

Модуль відстані Хеммінга допомагає виявляти і відкидати запити, що майже ідентичні до вже виконаних. У реалізації цього алгоритму використано математичні розрахунки й оптимізовано цей модуль, щоб забезпечити його високу продуктивність навіть за великих об'ємів даних.

Інтеграція цих двох модулів з рештою системи була виконана з урахуванням потреб масштабування та високої доступності (рис. 4). Упроваджено багато оптимізацій, які допомагають системі ефективно обробляти великі об'єми даних без втрати продуктивності.

```python
# A request that will pass validation
response, message = alice.create_query("SELECT * FROM users;")
print(f"SQL Command 1: {message} {response}")  # Alice: Запит валідний.

# A query that is too long
long_query = "SELECT * FROM users WHERE id IN (" + ",".join(["?"] * 600) + ");"
response, message = bob.create_query(long_query)
print(f"SQL Command 2: {message}  {response}")  # Bob: Занадто довгий запит.

# A request that contains potentially dangerous commands
dangerous_query = "DROP TABLE users;"
response, message = alice.create_query(dangerous_query)
print(f"SQL Command 3: {message} {response}")
```

Рис. 4. Приклад використання, поєднання класів для забезпечення виконання алгоритму

Зусилля зосереджено на забезпеченні якісного користувацького досвіду, оптимізації продуктивності та гарантії надійності рішення.

Отже, система готова до впровадження та використання в умовах високого навантаження, забезпечуючи ефективність роботи та задоволеність користувачів.

Проведене тестування було спрямоване на перевірку правильності роботи кожного компонента системи, а також їх взаємодії (рис. 5). Використовувалися як модульні тести для перевірки функціональності окремих методів, так й інтеграційні тести для перевірки здатності компонентів системи працювати разом.

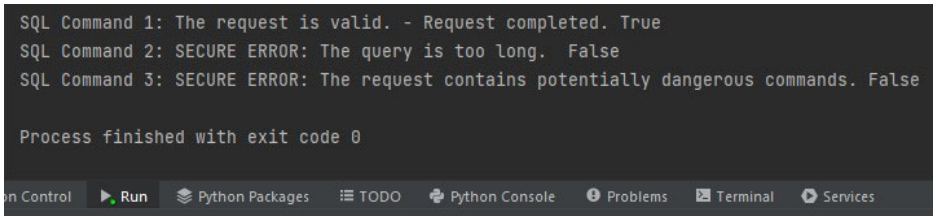

Рис. 5. Результат тестування роботи алгоритму

Окрему увагу варто звернути на можливості інтеграції розроблених модулів у Django-проєкти. У процесі тестування було встановлено: код може бути легко та швидко інтегрований у вигляді окремого Django-додатку, що надає розробникам можливість використовувати всі його переваги без необхідності переписування чи адаптації коду. Така інтеграція не вимагає значних ресурсів та часу і може бути виконана з мінімальними змінами в наявній кодовій базі Django-проєкту.

У тестових показниках забезпечення безпеки BlockchainSQLSecure демонструє абсолютну ефективність у 100 %, тоді як стандартна бібліотека Django та стандартні плагіни WordPress мають приблизну ефективність у 95 та 90 % відповідно, щодо тих же SQL-ін'єкцій. Цей графік візуально підкреслює перевагу BlockchainSQLSecure, показуючи її здатність забезпечити надійний захист проти SQL-ін'єкцій порівняно з іншими відомими системами управління вмістом та фреймворками (рис. 6). Така висока ефективність системи забезпечується завдяки унікальній інтеграції блокчейн-технологій і передових методів аналізу даних.

Отже, результати дослідження підтверджують високий потенціал системи для оптимізації роботи з базами даних і її легкість інтеграції в наявні проєкти на базі Django. У цьому полягає її привабливість для широкого спектра застосувань: від невеликих вебсайтів до великих корпоративних систем.

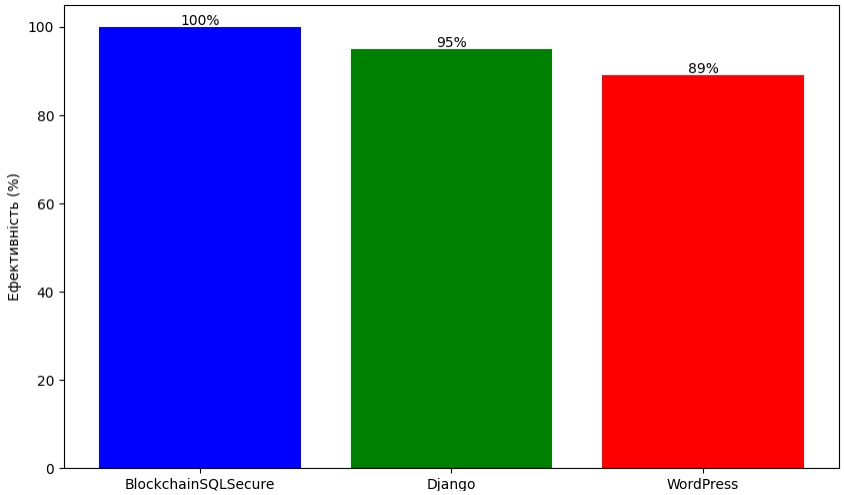

Рис. 6. Ефективність систем у боротьбі із SQL-ін'єкціями

Дискусія і висновки

Створена система демонструє високий рівень структурованості, безпеки й ефективності. Використання UML-схеми дало нам змогу ретельно спланувати архітектуру системи, визначивши ключові компоненти та їхню взаємодію. Кожен клас відіграє унікальну роль, що сприяє створенню цілісної та надійної системи.

Програмний код було розроблено мовою Python, з дотриманням високих стандартів його якості та чистоти. Ретельне документування класів і методів, а також використання об'єктно-орієнтованого підходу, сприяли легкості розуміння, підтримки та розширення системи.

Уведення окремих модулів для роботи із фільтром Блума та використання формули відстані Хеммінга стали важливою частиною оптимізації продуктивності системи. Ці технології допомогли зменшити навантаження на базу даних і підвищити швидкість оброблення запитів, водночас забезпечуючи високий рівень цілісності даних і безпеки.

Проведене тестування підтвердило ефективність і надійність системи, демонструючи її здатність ефективно обробляти SQL-запити, контролювати дії користувачів і підтримувати високий рівень цілісності даних.

Загалом результати дослідження та розробки демонструють, що ми досягли поставлених цілей, створивши систему, яка не тільки відповідає поточним вимогам, але й готова до масштабування та адаптації до майбутніх викликів. Ми впевнені, що створена система буде корисною для організацій, які шукають надійні й ефективні рішення для управління даними й оптимізації процесів оброблення SQL-запитів.

Запропонована система характеризується не лише високою продуктивністю та надійністю, але й гнучкістю щодо інтеграції з іншими проєктами. Однією з переваг такого рішення є те, що код може бути легко використаний як застосунок у фреймворку Django, що робить його ідеальним варіантом для впровадження в наявні проєкти.

Процес інтеграції можливо застосувати як створення Django-додатку, що охоплює всі розроблені модулі та класи. Це допоможе розробникам легко під'єднати код до своїх проєктів, використовуючи стандартні механізми Django для під'єднання зовнішніх застосунків. Весь функціонал системи, включно із фільтром Блума та розрахунком відстані Хеммінга, стає доступним усередині Django-проєкту, що дасть розробникам можливість використовувати його у своїх власних моделях і шаблонах.

Такий підхід до інтеграції робить алгоритм системи гарним вибором для команд, які вже використовують Django у своїх проєктах і шукають ефективних шляхів оптимізації оброблення SQL-запитів та підвищення продуктивності своєї системи. Завдяки легкості впровадження та використання, система є гнучким і потужним рішенням, що може принести значну користь будь-якому Django-проєкту.

Внесок авторів: Ірина Замрій – концептуалізація, аналіз джерел, огляд літератури та теоретичних основ дослідження, емпіричне дослідження; Іван Шахматов – методика, ПЗ; Владислав Яскевич – збір і перевірка емпіричних даних.

Список використаних джерел

Aini, Q., Manongga, D., Rahardja, U., Sembiring, I., Elmanda, V., Faturahman, A., & Santoso, N. P. L. (2022) Security Level Significance in DApps Blockchain-Based Document Authentication. *Aptisi Transactions on Technopreneurship (ATT),* 4(3), 292–305. https://doi.org/10.34306/att.v4i3.277

Alghawazi, M., Alghazzawi, D., & Alarifi, S. (2023). Detection of SQL Injection Attack Using Machine Learning Techniques: A Systematic Literature Review. *Journal Cybersecurity and Privacy*, 2(4), 764–777. https://doi.org/10.3390/jcp2040039

Alouffi, B., Hasnain, M., Alharbi, A., Alosaimi, W., Alyami, H., & Ayaz, M. (2021). Systematic Literature Review on Cloud Computing Security: Threats and Mitigation Strategies. *IEEE Access*, 9, 57792–57807. https://doi.org/10.1109/ACCESS.2021.3073203

Anderson, L. (2021, February). *Hack of 150,000 Verkada cameras: It could have been worse. SourceSecurity*. https://www.sourcesecurity.com/insights/hack-150-000-verkada-security-cameras-tesla-co-2566-ga-co-14080-ga-co-1552977087-ga-sb.1615396438.html

Awadallah, R., & Samsudin, A. (2021). Using Blockchain in Cloud Computing to Enhance Relational Database Security. *IEEE Access*, 9, 137353–137366. https://doi.org/10.1109/ACCESS.2021.3117733

Chamberland, C. (2022, February 10). *Unauthenticated SQL Injection Vulnerability Patched in WordPress Statistics Plugin*. Wordfence. https://www.wordfence.com/blog/2022/02/unauthenticated-sql-injection-vulnerability-patched-in-wordpress-statistics-plugin

Chen, W., Chen, G., Zhao, Y., & Zhang, J. (2021). Security vulnerability and encryption technology of computer information technology data under big data environment. *Journal of Physics: Conference Series*, 1800, 012012. https://doi.org/10.1088/1742-6596/1800/1/012012

Devalla, V., Srinivasa, Raghavan S., Maste, S., Kotian, J. D., & Annapurna, D. (2022). mURLi: A Tool for Detection of Malicious URLs and Injection Attacks. *Procedia Computer Science*, 215, 662–676. https://doi.org/10.1016/j.procs.2022.12.068

Guanxiu, L. (2022). The Application of Data Encryption Technology in Computer Network Communication Security. *Mobile Information Systems*, 3632298, 1–10. https://doi.org/10.1155/2022/3632298

Hussein, K. M., & Al-Gailani, M. F. (2023). Evaluation Performance of Bloom Filter in Blockchain Network. *Iraqi Journal of Information and Communications Technology*, 6(1), 1–8. https://doi.org/10.31987/ijict.6.1.204

Kalajdjieski, J., Raikwar, M., Arsov, N., Velinov, G., & Gligoroski, D. (2023). Databases fit for blockchain technology: A complete overview. *Blockchain: Research and Applications,* 4 (1), 100116. https://www.zjujournals.com/blockchain/CN/https://doi.org/10.1016/j.bcra.2022.100116

Li, J., & Kassem, M. (2021). Applications of distributed ledger technology (DLT) and Blockchain-enabled smart contracts in construction. *Automation in Construction,* 1, 32, 103955. https://doi.org/10.1016/j.autcon.2021.103955

Nagabhooshanam, N., Bala, Sundara Ganapathy N., Ravindra, Murthy C., Al Ansari Mohammed, Saleh, & Cosio Borda, R. F. (2023). Neural network based single index evaluation for SQL injection attack detection in health care data. *Measurement: Sensors,* 27, 100779, 2665–9174. https://doi.org/10.1016/j.measen.2023.100779

Oladimeji, S., Kerner, S. M. (2023, November 3). *SolarWinds hack explained: Everything you need to know*. TechTarget. https://www.techtarget.com/whatis/feature/SolarWinds-hack-explained-Everything-you-need-to-know

Przytarski, D., Stach, C., Gritti, C., & Mitschang, B. (2022). Query Processing in Blockchain Systems: Current State and Future Challenges. Security and Privacy in Blockchains and the IoT. *Future Internet*, 14(1), 1. https://doi.org/10.3390/fi14010001

Queralta, J. P., Keramat, F., Salimi, S., Fu, L., Yu, X., & Westerlund, T. (2023). Blockchain and Emerging Distributed Ledger Technologies for Decentralized Multi-robot Systems. *Current Robotics Reports*, 4, 43–54. https://doi.org/10.1007/s43154-023-00101-3

Sakharkar, S. (2023). Systematic Review: Analysis of Coding Vulnerabilities across Languages. *Journal of Information Security*, 14, 330–342. https://doi.org/10.4236/jis.2023.144019

Siva, Kumar A, Godfrey, Winster S, & Ramesh, R. (2021) Efficient sensitivity orient blockchain encryption for improved data security in cloud. *Concurrent Engineering*. 29(3), 249-257. https://doi:10.1177/1063293X211008586

Sobchuk, V., Zamrii, I., & Laptiev, S. (2023). Ensuring Functional Stability of Technological Processes as Cyberphysical Systems Using Neural Networks. *Lecture Notes in Networks and Systems,* 536, 581–592. https://doi.org/10.1007/978-3-031-20141-7_53

Sobchuk, V., Zamrii, I., Barabash, O., & Musienko, A. (2021). Methodology for building a functionally stable intelligent information system of a manufacturing enterprise. *Bulletin of the Taras Shevchenko National University of Kyiv. Physics and Mathematics*, 4, 116–127. https://doi.org/10.17721/1812-5409.2021/4.18

Tanrıverdi, M., & Tekerek, A. (2021). *Implementation of Blockchain Based Distributed Web Attack Detection Application*. Feminist Press at CUNY.

Tanwar, S., Gupta, N., Iwendi, C., Kumar, K., & Alenezi, M. (2022). Next Generation IoT and Blockchain Integration. *Journal of Sensors,* 9077348, 1–14. https://doi.org/10.1155/2022/9077348

Zaman, S., Alhazmi, K., Aseeri, M. A., Ahmed, M. R., & Khan, R. T. (2021). Security Threats and Artificial Intelligence Based Countermeasures for Internet of Things Networks: A Comprehensive Survey. *IEEE Access,* 9, 94668–94690. https://doi.org/10.1109/доступ.2021.3089681

References

Aini, Q., Manongga, D., Rahardja, U., Sembiring, I., Elmanda, V., Faturahman, A., & Santoso, N. P. L. (2022) Security Level Significance in DApps Blockchain-Based Document Authentication. *Aptisi Transactions on Technopreneurship (ATT),* 4(3), 292–305. https://doi.org/10.34306/att.v4i3.277

Alghawazi, M., Alghazzawi, D., & Alarifi, S. (2023). Detection of SQL Injection Attack Using Machine Learning Techniques: A Systematic Literature Review. *Journal Cybersecurity and Privacy*, 2(4), 764-777. https://doi.org/10.3390/jcp2040039

Alouffi, B., Hasnain, M., Alharbi, A., Alosaimi, W., Alyami, H., & Ayaz, M. (2021). Systematic Literature Review on Cloud Computing Security: Threats and Mitigation Strategies. *IEEE Access*, 9, 57792–57807. https://doi.org/10.1109/ACCESS.2021.3073203

Anderson, L. (2021, February). *Hack of 150,000 Verkada cameras: It could have been worse. SourceSecurity*. https://www.sourcesecurity.com/insights/hack-150-000-verkada-security-cameras-tesla-co-2566-ga-co-14080-ga-co-1552977087-ga-sb.1615396438.html

Awadallah, R., & Samsudin, A. (2021). Using Blockchain in Cloud Computing to Enhance Relational Database Security. *IEEE Access*, 9, 137353–137366. https://doi.org/10.1109/ACCESS.2021.3117733

Chamberland, C. (2022, February 10). *Unauthenticated SQL Injection Vulnerability Patched in WordPress Statistics Plugin*. Wordfence. https://www.wordfence.com/blog/2022/02/unauthenticated-sql-injection-vulnerability-patched-in-wordpress-statistics-plugin

Chen, W., Chen, G., Zhao, Y., & Zhang, J. (2021). Security vulnerability and encryption technology of computer information technology data under big data environment. *Journal of Physics: Conference Series*, 1800, 012012. https://doi.org/10.1088/1742-6596/1800/1/012012

Devalla, V., Srinivasa, Raghavan S., Maste, S., Kotian, J. D., & Annapurna, D. (2022). mURLi: A Tool for Detection of Malicious URLs and Injection Attacks. *Procedia Computer Science*, 215, 662–676. https://doi.org/10.1016/j.procs.2022.12.068

Guanxiu, L. (2022). The Application of Data Encryption Technology in Computer Network Communication Security. *Mobile Information Systems*, 3632298, 1–10. https://doi.org/10.1155/2022/3632298

Hussein, K. M., & Al-Gailani, M. F. (2023). Evaluation Performance of Bloom Filter in Blockchain Network. *Iraqi Journal of Information and Communications Technology*, 6(1), 1–8. https://doi.org/10.31987/ijict.6.1.204

Kalajdjieski, J., Raikwar, M., Arsov, N., Velinov, G., & Gligoroski, D. (2023). Databases fit for blockchain technology: A complete overview. *Blockchain: Research and Applications,* 4(1), 100116. https://www.zjujournals.com/blockchain/CN/https://doi.org/10.1016/j.bcra.2022.100116

Li, J., & Kassem, M. (2021). Applications of distributed ledger technology (DLT) and Blockchain-enabled smart contracts in construction. *Automation in Construction,* 1, 32, 103955. https://doi.org/10.1016/j.autcon.2021.103955

Nagabhooshanam, N., Bala, Sundara Ganapathy N., Ravindra, Murthy C., Al Ansari Mohammed, Saleh, & Cosio Borda, R. F. (2023). Neural network based single index evaluation for SQL injection attack detection in health care data. *Measurement: Sensors,* 27, 100779, 2665–9174. https://doi.org/10.1016/j.measen.2023.100779

Oladimeji, S., Kerner, S. M. (2023, November 3). *SolarWinds hack explained: Everything you need to know*. TechTarget. https://www.techtarget.com/whatis/feature/SolarWinds-hack-explained-Everything-you-need-to-know

Przytarski, D., Stach, C., Gritti, C., & Mitschang, B. (2022). Query Processing in Blockchain Systems: Current State and Future Challenges. Security and Privacy in Blockchains and the IoT. *Future Internet*, 14(1), 1. https://doi.org/10.3390/fi14010001

Queralta, J. P., Keramat, F., Salimi, S., Fu, L., Yu, X., & Westerlund, T. (2023). Blockchain and Emerging Distributed Ledger Technologies for Decentralized Multi-robot Systems. *Current Robotics Reports*, 4, 43–54. https://doi.org/10.1007/s43154-023-00101-3

Sakharkar, S. (2023). Systematic Review: Analysis of Coding Vulnerabilities across Languages. *Journal of Information Security*, 14, 330–342. https://doi.org/10.4236/jis.2023.144019

Siva, Kumar A, Godfrey, Winster S, & Ramesh, R. (2021) Efficient sensitivity orient blockchain encryption for improved data security in cloud. *Concurrent Engineering*, 29(3), 249–257. https://doi:10.1177/1063293X211008586

Sobchuk, V., Zamrii, I., & Laptiev, S. (2023). Ensuring Functional Stability of Technological Processes as Cyberphysical Systems Using Neural Networks. *Lecture Notes in Networks and Systems,* 536, 581–592. https://doi.org/10.1007/978-3-031-20141-7_53

Sobchuk, V., Zamrii. I., Barabash, O., & Musienko, A. (2021). Methodology for building a functionally stable intelligent information system of a manufacturing enterprise. *Bulletin of the Taras Shevchenko National University of Kyiv. Physics and Mathematics*, 4, 116–127. https://doi.org/10.17721/1812-5409.2021/4.18

Tanrıverdi, M., & Tekerek, A. (2021). *Implementation of Blockchain Based Distributed Web Attack Detection Application*. Feminist Press at CUNY.

Tanwar, S., Gupta, N., Iwendi, C., Kumar, K., & Alenezi, M. (2022). Next Generation IoT and Blockchain Integration. *Journal of Sensors,* 9077348, 1–14. https://doi.org/10.1155/2022/9077348

Zaman, S., Alhazmi, K., Aseeri, M. A., Ahmed, M. R., & Khan, R. T. (2021). Security Threats and Artificial Intelligence Based Countermeasures for Internet of Things Networks: A Comprehensive Survey. *IEEE Access,* 9, 94668–94690. https://doi.org/10.1109/доступ.2021.3089681

Отримано редакцією журналу / Received: 15.01.24
Прорецензовано / Revised: 07.04.24
Схвалено до друку / Accepted: 27.05.24

Iryna ZAMRII, DSc (Engin.), Assoc. Prof.
ORCID ID: 0000-0001-5681-1871
e-mail: irinafraktal@gmail.com
State University of Information and Communication Technologies, Kyiv, Ukraine

Ivan SHAKHMATOV, PhD Student
ORCID ID: 0009-0004-9628-0365
e-mail: ivan.shakhmatov@gmail.com
State University of Information and Communication Technologies, Kyiv, Ukraine

Vladyslav YASKEVYCH, PhD (Engin.)
ORCID ID: 0000-0002-5796-2521
e-mail: v.yaskevitch@gmail.com
Borys Grinchenko Kyiv University, Kyiv, Ukraine

BLOCKCHAINSQLSECURE: INTEGRATION OF BLOCKCHAIN TO STRENGTHEN PROTECTION AGAINST SQL INJECTIONS

In today's digital world, where a huge amount of data is processed and stored in online systems, the security of web applications becomes an extremely important aspect to ensure the protection of information and privacy of users. One of the most common and most dangerous threats to web application security is SQL injection attacks, which can lead to unauthorized access to databases, leakage of confidential information, and complete data destruction.

In this regard, the development of effective methods of protection against SQL injections is an actual and urgent problem. The use of traditional methods of protection often turns out to be insufficiently effective, so in search of new solutions, they turn to innovative technologies, in particular to blockchain. Blockchain is a decentralized, distributed database that provides a high level of security and transparency through the use of cryptographic methods and consensus algorithms. These properties make blockchain a promising technology for the development of new methods of protection against SQL injections.

This article introduces the concept of BlockchainSQLSecure, a unique method that uses the capabilities of blockchain technologies to create an additional level of security in Django applications.

The central idea is to create a blockchain log for each SQL query to the database, which allows you to guarantee the immutability and traceability of information. This transparency actively counteracts the possibility of unauthorized changes or attempted injections, as any tampering becomes immediately visible.

In the framework of the article, the mechanisms of validating SQL queries through smart contracts on the blockchain, which allow you to automatically reject queries containing potential injections, are investigated. Methods of decentralized storage of logs are also considered, which ensure the distribution of information among network participants, making the system resistant to attacks and attempts at external intervention.

The introduction of separate modules for working with the Bloom Filter and the use of the Hamming distance formula became an important part of optimizing the system's performance. These technologies have helped reduce database load and improve query processing speed, while ensuring high levels of data integrity and security.

The conducted testing confirmed the efficiency and reliability of the system, demonstrating its ability to efficiently process SQL queries, monitor user actions, and maintain a high level of data integrity. Research and development results demonstrate that the system not only meets current requirements, but is also ready to scale and adapt to future challenges.

The proposed system is characterized not only by high performance and reliability, but also by flexibility in terms of integration with other projects. One of the advantages of such a solution is that the code can be easily used as an application in the Django framework, making it ideal for implementation in existing projects. This approach to integration makes the system algorithm an excellent choice for teams that already use Django in their projects and are looking for efficient ways to optimize the processing of SQL queries and improve the performance of their system. Due to its ease of implementation and use, the system is a flexible and powerful solution that can greatly benefit any Django project. In a generalized form, the presented concept shows how a combination of traditional protection methods and modern blockchain technologies can provide a new level of security for web applications.

K e y w o r d s : ***Blockchain, Django, web applications, security, integration, access restriction.***

Автори заявляють про відсутність конфлікту інтересів. Спонсори не брали участі в розробленні дослідження; у зборі, аналізі чи інтерпретації даних; у написанні рукопису; в рішенні про публікацію результатів.

The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses or interpretation of data; in the writing of the manuscript; in the decision to publish the results.