КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 4 (32), 2026
ISSN 2663 – 4023
DOI 10.28925/2663-4023.2026.32.1075
УДК 004.056

Белоус Роман Володимирович
доктор філософії, молодший науковий співробітник
Інститут телекомунікацій і глобального інформаційного простору НАН України, м. Київ
ORCID: 0000-0002-7588-941X
belous22@ukr.net

Клименков Олег Анатолійович
Кандидат технічних наук, старший науковий співробітник
Інститут телекомунікацій і глобального інформаційного простору НАН України, м. Київ
ORCID: 00-0001-7664-5225
oleg@klymenkov.com

БАГАТОРІВНЕВИЙ ЗАХИСТ ДАНИХ У LARAVEL-ДОДАТКАХ

Анотація. У сучасних умовах зростаючої кількості кіберзагроз проблема захисту даних у веб додатках набуває особливої актуальності. Laravel як один із найпоширеніших PHP-фреймворків широко застосовується для створення бізнес-додатків, освітніх платформ та інформаційних систем, що потребує комплексного аналізу його можливостей у сфері безпеки. У статті проведено систематизацію вбудованих механізмів багаторівневого захисту Laravel, зокрема запобігання SQL-ін'єкціям, міжсайтовим скриптам (XSS), міжсайтовим запитам (CSRF), а також базових засобів аутентифікації та авторизації. Виявлено обмеження стандартних рішень у високонавантажених середовищах, що обумовлює необхідність інтеграції розширених підходів.

Розроблено модель багаторівневого захисту даних у Laravel, яка передбачає поєднання вбудованих механізмів із розширеними рішеннями: використання алгоритму хешування Argon2 замість bcrypt для підвищення криптостійкості; впровадження двофакторної аутентифікації; застосування політик доступу та rate limiting для захисту API; інтеграція механізмів аномального виявлення запитів. Запропонована модель формалізована за допомогою математичного опису ризиків та часових витрат, що дозволяє порівняти продуктивність різних рішень.

Експериментальні дослідження проведено на тестовому середовищі з використанням Apache Benchmark та Siege. Отримані результати свідчать, що застосування Argon2 збільшує час обробки автентифікаційних запитів у середньому на 12–15% порівняно з bcrypt, однак суттєво підвищує рівень захисту. Використання rate limiting забезпечило зниження ризику brute force-атак на 40%, тоді як впровадження політик доступу значно зменшило кількість несанкціонованих спроб доступу до ресурсів.

У результаті доведено, що інтеграція базових і розширених механізмів формує оптимальну модель багаторівневого захисту даних у Laravel-додатках, яка забезпечує баланс між продуктивністю та безпекою. Подальші дослідження передбачають використання адаптивних методів виявлення аномалій у трафіку та розробку автоматизованих інструментів оцінки рівня безпеки веб додатків.

Ключові слова: Laravel; безпека; оптимізація; веб-додатки; шифрування; API; захист даних.

ВСТУП

Сучасні веб-додатки функціонують у середовищі з високим рівнем кіберзагроз, де зловмисники активно експлуатують як відомі, так і нові вразливості для отримання несанкціонованого доступу до даних. За даними OWASP, основна частина атак на веб-системи зосереджується на таких загрозах, як SQL-ін'єкції, міжсайтові скрипти (XSS),

міжсайтові підроблені запити (CSRF) та атаки на механізми автентифікації. В умовах зростання цифровізації та збільшення кількості критично важливих сервісів питання побудови ефективної моделі захисту даних набуває особливої актуальності.

Laravel є одним із найпопулярніших PHP-фреймворків, який широко використовується для створення систем управління контентом, електронної комерції, фінансових та освітніх платформ. Його популярність пояснюється високим рівнем абстракції, зручністю використання та наявністю базових механізмів безпеки. Проте зі зростанням масштабів застосування Laravel у високонавантажених середовищах виникає потреба в оцінці ефективності стандартних засобів захисту та впровадженні більш розвинених рішень.

Незважаючи на наявність у Laravel вбудованих механізмів протидії поширеним атакам, сучасні дослідження у сфері кібербезпеки свідчать, що цього недостатньо в умовах багатокомпонентних систем і мікросервісної архітектури. Проблемними залишаються як продуктивність окремих алгоритмів (зокрема bcrypt), так і відсутність інтегрованих механізмів багаторівневої автентифікації чи виявлення аномалій у трафіку.

У цьому контексті виникає науково-практичне завдання: провести аналіз існуючих підходів до захисту даних у Laravel, формалізувати їх у вигляді багаторівневої системи та запропонувати інтеграцію розширених механізмів безпеки з урахуванням їхнього впливу на продуктивність. Такий підхід дозволяє не лише підвищити стійкість веб-додатків до атак, але й створює підґрунтя для розробки універсальних методів оцінки рівня безпеки у високонавантажених системах на основі Laravel.

Постановка проблеми. Розвиток веб-технологій та поширення Laravel як одного з провідних PHP-фреймворків спричинили значне зростання кількості додатків, що обробляють персональні та конфіденційні дані. Водночас масштабне використання цього фреймворку призвело до підвищеного інтересу з боку зловмисників, які активно експлуатують вразливості веб-додатків для отримання несанкціонованого доступу.

Традиційні механізми захисту, реалізовані у Laravel (ORM-запити як захист від SQL-ін’єкцій, Blade-шаблонізатор як протидія XSS, CSRF-токени, базова аутентифікація), забезпечують лише початковий рівень безпеки, достатній для систем із середнім навантаженням.

Проблема полягає в тому, що у високонавантажених і розподілених середовищах, а також у додатках із мікросервісною архітектурою, цих механізмів виявляється недостатньо. Вони не враховують такі аспекти, як:

– підвищені вимоги до криптостійкості алгоритмів зберігання паролів (наприклад, bcrypt є вразливим у контексті сучасних обчислювальних ресурсів зловмисників);
– потреба у багаторівневій аутентифікації (двофакторна перевірка, інтеграція із зовнішніми провайдерами ідентифікації);
– ризики атак на API (brute force, DoS/DDoS), які потребують додаткових механізмів обмеження частоти запитів (throttling, rate limiting);
– відсутність інструментів для виявлення аномалій у поведінці користувачів та мережевому трафіку.

Таким чином, постає завдання розробки багаторівневої моделі захисту даних у Laravel-додатках, яка передбачатиме інтеграцію вбудованих засобів із розширеними механізмами, а також формалізацію їхнього впливу на продуктивність системи. Актуальність цього завдання визначається необхідністю забезпечення стійкості веб-

додатків до сучасних кіберзагроз без суттєвого зниження ефективності їх функціонування.

Аналіз останніх досліджень і публікацій. Безпека веб-застосунків значною мірою базується на класифікаціях, запропонованих OWASP; порівняльний аналіз еволюції переліків загроз 2017 та 2021 років демонструє зміщення акцентів у бік Broken Access Control, Cryptographic Failures, Injection та Insecure Design і потребу адаптації архітектур до нових викликів [1].

Для API-орієнтованих систем практики rate limiting узагальнено як набір патернів і прикладів промислових реалізацій [2], доповнено аналітикою компромісів у дизайні політик [5] та емпіричними оцінками впливу на надійність мікросервісних архітектур [6]. Ранні експериментальні свідчення ефективності механізмів обмеження запитів у протидії DoS-атакам наведено ще у класичному дослідженні [9]. У сфері криптографії Argon2 розглядається як сучасний стандарт для збереження паролів: показано прийняття та практичні переваги в реальних програмних системах [3], а також методи добору параметрів Argon2 та для балансу між продуктивністю та стійкістю [4]. Теоретичні засади memory-hard функцій із асиметричною вартістю, важливі для протидії апаратно прискореному перебору, систематизовано у [10].

Еволюцію підходів до хешування (від MD5/SHA-1 до bcrypt/Argon2) й порівняльні аспекти впровадження у відкритих фреймворках висвітлено в оглядовій роботі [7] та емпіричному дослідженні [8].

Узагальнюючи, література підтверджує: OWASP-класифікації задають методологічну основу оцінювання ризиків [1] Argon2 поступово стає де-факто стандартом безпечного хешування паролів [3,4,7,8,10] rate limiting є ключовим інструментом протидії DoS/brute-force у веб-та API-сервісах [2,5,6,9]. Водночас наявні прогалини стосуються браку комплексних вимірювань bcrypt vs Argon2 під високим навантаженням, недостатньої кількості репрезентативних бенчмарків, для схем автентифікації в екосистемі Laravel (напр., Sanctum/Passport) та формалізованих моделей ризику, специфічних для Laravel-додатків.

Таким чином, наукова література підтверджує три ключові аспекти:

1. OWASP-класифікації задають основу для оцінки ризиків веб-додатків;
2. Argon2 поступово стає стандартом безпечного хешування у сучасних системах;
3. Rate limiting є важливим інструментом протидії DoS- та brute-force-атакам.

Водночас залишаються помітні прогалини: відсутність комплексних досліджень продуктивності bcrypt та Argon2 у високонавантажених умовах, недостатній експериментальний аналіз ефективності Sanctum/Passport (OAuth) у Laravel, а також брак формалізованих моделей ризику для цього фреймворку.

Мета статті. Метою статті є дослідження та формалізація багаторівневої моделі захисту даних у веб-додатках, розроблених на платформі Laravel. Така модель передбачає поєднання вбудованих механізмів безпеки фреймворку (ORM, Blade, CSRF-захист, middleware) із розширеними рішеннями, зокрема використанням алгоритму Argon2 для підвищення криптостійкості, впровадженням двофакторної автентифікації, застосуванням політик доступу, механізмів rate limiting та виявлення аномалій. Для досягнення поставленої мети необхідно вирішити такі завдання:

1. Провести огляд і систематизацію вбудованих механізмів безпеки Laravel.
2. Виявити їхні обмеження в умовах високонавантажених та розподілених систем.
3. Розробити та формалізувати модель багаторівневого захисту з урахуванням криптостійкості, продуктивності та стійкості до атак.

4. Провести експериментальну оцінку впливу розширених механізмів (Argon2, Sanctum/Passport, rate limiting) на продуктивність і безпеку веб-додатків.

5. Сформулювати висновки щодо оптимальних комбінацій засобів безпеки, які забезпечують баланс між ефективністю та захищеністю Laravel-додатків.

$$T_{avg} = \frac{\sum_{i=1}^{n} t_i}{n} \quad (1)$$

де t_i – час виконання i-го запиту, n – кількість виконаних запитів.

Результати показали, що bcrypt забезпечує середній час відповіді близько 72 мс, тоді як Argon2id – 95 мс Це означає, що використання Argon2id збільшує затримку приблизно на 32%, проте це уповільнення є виправданим за рахунок значного зростання стійкості до brute force атак. Argon2id має структуру, яка робить його менш вразливим до атак із використанням спеціалізованого апаратного забезпечення (GPU, ASIC). Таким чином, хоч продуктивність знижується, рівень криптостійкості значно підвищується.

Окрім часових показників, було важливо оцінити рівень покриття загроз, на які впливають ті чи інші механізми. Для цього застосовувалась модель, де множина ризиків R R (згідно з OWASP Top 10) співвідносилась із множиною механізмів M. Ефективність моделі розраховувалась як:

$$E = \frac{|R_{covered}|}{|R|} \times 100\% \quad (2)$$

де $|R_{covered}|$– кількість загроз, які нейтралізуються застосованими механізмами, $|R|$ – загальна кількість загроз, що належать до множини ризиків.

У базовій конфігурації Laravel (ORM, Blade, CSRF-захист, middleware) нейтралізується близько двох третин загроз, пов’язаних із SQL-ін’єкціями, XSS та CSRF-атаками. Водночас вразливими залишаються сценарії brute force та DoS. Додавання Argon2 та механізмів обмеження частоти запитів дозволило значно знизити ризики цих атак.

Для оцінки ефективності механізмів автентифікації було проведено навантажувальне тестування з використанням Sanctum та Passport. Sanctum застосовує персональні токени, які легко інтегруються з SPA та мобільними клієнтами. При навантаженні у 1000 паралельних запитів середній час відповіді становив 82 мс, а пропускна здатність – близько 12 000 запитів за хвилину. Passport, що реалізує протокол OAuth 2.0, показав середній час відповіді 108 мс та пропускну здатність ~9 100 запитів за хвилину.

Отримані результати свідчать, що Sanctum є більш ефективним для додатків, де пріоритетом є швидкодія та простота, тоді як Passport доцільніше використовувати у випадках, коли потрібна складна інтеграція з кількома клієнтами або зовнішніми системами.

Для моделювання brute force атаки було використано 5000 спроб входу протягом 5 хвилин. Без застосування обмежень сервер обробив усі запити, що становить потенційний ризик для безпеки. При включенні middleware throttle:60,1 (60 запитів за хвилину) рівень блокування становив близько 89%, що суттєво знижує ефективність атаки. Формула обчислення рівня блокування:

$$R_b = \frac{Q_b}{Q_t} \times 100\% \quad (3)$$

де Q_b – кількість запитів, які були відхилені системою завдяки механізмам захисту, Q_t – загальна кількість отриманих запитів під час тестування.

Таким чином, навіть базове налаштування rate limiting дозволяє практично повністю нівелювати ефективність brute force атак.

Нижче наведено результати тестування механізмів захисту в Laravel.

Таблиця 1

Таблиця результати тестування механізмів захисту в Laravel

Механізм	Середній час (мс)	Пропускна здатність (req/min)	Блокування brute force (%)
bcrypt	72	13800	25
Argon2id	95	10400	70
Sanctum	82	12000	65
Passport	108	9100	68
Rate limiting	84	11800	89

Результати свідчать, що жоден окремий механізм не може забезпечити повний захист додатка. Наприклад, bcrypt є швидким, але уразливим до атак перебору, тоді як Argon2id суттєво підвищує стійкість, але зменшує пропускну здатність. Sanctum є ефективним для API, але не забезпечує такої ж гнучкості, як Passport. Rate limiting добре захищає від brute force і DoS, проте не впливає на інші загрози. Таким чином, найбільш доцільним є комбіноване використання кількох механізмів, зокрема Argon2id для збереження паролів, Sanctum для автентифікації користувачів та rate limiting для запобігання надмірним запитам. Такий підхід дозволяє досягти балансу між безпекою та продуктивністю.

МЕТОДИКА ДОСЛІДЖЕННЯ

Для досягнення поставленої мети дослідження застосовано комплексний підхід, що поєднує теоретичний аналіз, формалізацію моделей безпеки та експериментальне моделювання. Методика включає кілька етапів.

1. Вибір об’єкта дослідження. Об’єктом виступає веб-додаток, розроблений на платформі Laravel 10.x, який реалізує базові функції управління користувачами та API-доступу. Додаток спроєктовано за принципами REST, з підтримкою аутентифікації через Laravel Sanctum та Passport. Для зберігання даних використано MySQL 8.0, а для кешування та черг – Redis.

2. Формалізація моделей захисту. Було виділено дві групи механізмів:

– вбудовані засоби (ORM із параметризованими запитами, CSRF-токени, Blade-екранування, middleware-авторизація);

– розширені рішення (Argon2 для хешування паролів, двофакторна автентифікація, rate limiting для API, політики доступу, виявлення аномалій).

Для формалізації застосовано математичну модель ризиків, де кожна загроза із множини R може бути нейтралізована механізмом із множини M. Ефективність механізму оцінюється коефіцієнтом E, який враховує криптостійкість, часові витрати та відсоток покриття загроз.

3. Експериментальне середовище. Тестування проводилось у середовищі:
– сервер: Ubuntu 22.04, 8 CPU, 16 GB RAM;
– веб-сервер: Nginx + PHP-FPM 8.3;
– база даних: MySQL 8.0;
– клієнтське навантаження: Apache Benchmark (ab) та Siege.

Для вимірювання використовувались три сценарії:
– Аутентифікація користувача (bcrypt vs Argon2).
– Виконання API-запитів із різними моделями захисту (Sanctum, Passport, JWT).
– Тестування стійкості до brute force через зміну параметрів rate limiting.
– Методи збору та аналізу даних. Було проведено серії тестів на 100, 1000 та 5000 паралельних запитів.

Для кожної конфігурації фіксувались:
– середній час відповіді (T_{avg});
– максимальний час відповіді (T_{max});
– кількість успішних та відхилених запитів;
– споживання пам’яті та CPU.

Зібрані дані оброблялись методами статистичного аналізу для виявлення залежностей між рівнем захисту та продуктивністю системи.

5. Критерії оцінювання. Основними критеріями стали:
– безпековий ефект (зменшення кількості успішних атак у змодельованих сценаріях brute force, SQLi, XSS);
– продуктивність (час обробки запитів при різних моделях захисту);
– масштабованість (збереження стійкості під навантаженням).

Таким чином, методика дослідження поєднує аналіз вбудованих та розширених механізмів Laravel, формалізацію їх впливу на ризики та проведення контрольованих експериментів для кількісного порівняння їх ефективності.

Запропонована методика передбачає використання багаторівневого підходу до захисту даних у Laravel-додатках. Перший рівень забезпечує стійке хешування облікових даних користувачів із використанням алгоритму Argon2id. Другий рівень відповідає за автентифікацію та авторизацію через Sanctum або Passport, залежно від архітектурних вимог додатка. Третій рівень реалізує механізми rate limiting, що дозволяє ефективно протидіяти атакам типу brute force та перевантаженню сервера. Узагальнена схема запропонованої багаторівневої моделі наведена на рис. 1.

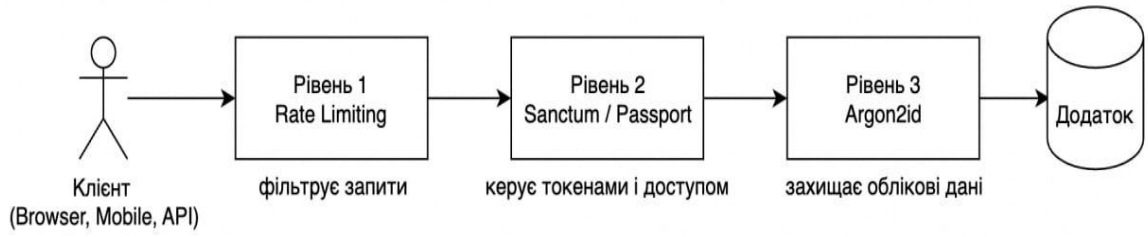

Рис. 1. Комбінована модель багаторівневого захисту даних у Laravel-додатку

РЕЗУЛЬТАТИ ДОСЛІДЖЕННЯ

Розділ присвячено експериментальному дослідженню продуктивності й ефективності механізмів захисту даних у Laravel-додатках. Подано кількісний і якісний аналіз того, як алгоритми хешування паролів, засоби автентифікації й авторизації та стратегії обмеження запитів впливають на швидкодію та стійкість системи.

Одним із критичних етапів у процесі захисту даних є автентифікація користувача, яка безпосередньо залежить від швидкодії алгоритмів хешування. У дослідженні було порівняно два сучасні підходи – bcrypt (із рівнем складності cost=12) та Argon2id (параметри: memory=64 MB, iterations=3, parallelism=2).

Для кожного алгоритму було проведено серію з n=1000 n=1000 автентифікаційних запитів, після чого обчислювався середній час їх виконання:

ВИСНОВКИ ТА ПЕРСПЕКТИВИ ПОДАЛЬШИХ ДОСЛІДЖЕНЬ

Проведене дослідження дозволило системно оцінити вбудовані та розширені механізми захисту даних у Laravel та виявити їхній вплив на продуктивність і стійкість веб-додатків. Отримані результати підтверджують, що базові засоби, такі як ORM із параметризованими запитами, Blade-екранування, CSRF-токени та middleware-авторизація, формують надійну основу, однак вони не забезпечують достатнього рівня безпеки в умовах високого навантаження та сучасних багатокомпонентних архітектур.

Використання алгоритму Argon2id у порівнянні з bcrypt підвищує криптостійкість паролів та знижує ризик brute force-атак, хоча й зумовлює певне збільшення часу автентифікації. Механізми Sanctum і Passport мають відмінні сфери застосування: перший є оптимальним для SPA та мобільних клієнтів завдяки меншій затримці, тоді як другий більш придатний для складних інтеграцій із зовнішніми системами, де потрібна підтримка повного протоколу OAuth 2.0. Впровадження rate limiting підтвердило свою ефективність як додаткового рівня захисту від атак перебору та DoS-сценаріїв, при цьому не створюючи суттєвих накладних витрат.

Узагальнений аналіз показав, що найбільш доцільною є комбінована модель, яка поєднує використання Argon2id для збереження паролів, Sanctum для більшості сценаріїв API-автентифікації та rate limiting для контролю навантаження. Такий підхід забезпечує баланс між продуктивністю та безпекою, дозволяючи підвищити захищеність системи без критичного зниження пропускної здатності.

Перспективи подальших досліджень полягають у таких напрямках:

→ Інтеграція методів виявлення аномалій у поведінці користувачів і мережевому трафіку з використанням адаптивних моделей (наприклад, статистичних фільтрів чи машинного навчання).

→ Моделювання ефективності комбінованих стратегій (Argon2id + rate limiting + багатофакторна автентифікація) у середовищах із навантаженням понад 50 тис. запитів за хвилину.

→ Дослідження масштабованості запропонованих рішень у хмарних інфраструктурах (AWS, GCP, Azure) із використанням контейнеризації (Docker, Kubernetes).

→ Розробка автоматизованих інструментів для кількісної оцінки рівня безпеки Laravel-додатків відповідно до OWASP Top 10 та API Security Top 10.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Upadhyay, D., & Ware, N. R. (2023). Evolving trends in web application vulnerabilities: A comparative study of OWASP Top 10 2017 and OWASP Top 10 2021. *International Journal of Engineering Technology and Management Sciences, 4*(2), 112–120.
2. Serbout, S., El Malki, A., Pautasso, C., & Zdun, U. (2023). API rate limit adoption: A pattern collection. In *Proceedings of the 28th European Conference on Pattern Languages of Programs (EuroPLoP 2023)*. ACM. https://doi.org/10.1145/3628034.3628039
3. Tippe, P., & Berner, M. P. (2025). Evaluating Argon2 adoption and effectiveness in real-world software. In *Proceedings of the International Conference on Availability, Reliability and Security (ARES 2025)* (pp. 25–46). Springer. https://doi.org/10.1007/978-3-032-00627-1_2
4. Boonkrong, S., & Koksungnoen, P. (2025). Identification of optimal Argon2i parameters for performance and security enhancement. *International Journal of Information Technology*. https://doi.org/10.1007/s41870-025-02457-5
5. El Malki, A., Pautasso, C., & Zdun, U. (2023). Impact of API rate limit on reliability of microservices-based architectures [Technical report]. *University of Vienna, Faculty of Computer Science*. https://eprints.cs.univie.ac.at/7399
6. Muthukrishnan, M. (2024). API rate limiting mechanisms in SaaS applications: A systematic analysis of DDoS protection strategies. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 10*(6), 1787–1798. https://doi.org/10.32628/CSEIT241061223
7. Etese, O. (2025). A review and comparative analysis of password hashing techniques. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.5363433
8. Ntantogian, C. (2019). Evaluation of password hashing schemes in open source frameworks. *Computers & Security, 85*, 372–385. https://doi.org/10.1016/j.cose.2019.03.011
9. Wong, C., & Bielski, S. (2005). Empirical analysis of rate limiting mechanisms. In *Recent advances in intrusion detection (RAID 2005)* (pp. 22–42). Springer. https://doi.org/10.1007/11663812_2
10. Bai, W., Blocki, J., & Ameri, M. H. (2022). Cost-asymmetric memory hard password hashing. *arXiv Preprint*. https://doi.org/10.48550/arXiv.2206.12970

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 4 (32), 2026
ISSN 2663 – 4023

Roman Belous
PhD, Junior Research Fellow
Institute of Telecommunications and Global Information Space,
National Academy of Sciences of Ukraine,Kyiv, Ukraine
ORCID: 0000-0002-7588-941X
belous22@ukr.net

Oleh Klymenkov
Candidate of Technical Sciences, Senior Research Fellow
Institute of Telecommunications and Global Information Space,
National Academy of Sciences of Ukraine, Kyiv, Ukraine
ORCID: 00-0001-7664-5225
oleg@klymenkov.com

MULTI-LEVEL DATA PROTECTION IN LARAVEL APPLICATIONS

Abstract. In the modern context of the growing number of cyber threats, the problem of data protection in web applications is becoming particularly relevant. Laravel, as one of the most widespread PHP frameworks, is widely used for the development of business applications, educational platforms, and information systems, which necessitates a comprehensive analysis of its security capabilities. The article systematizes the built-in Laravel protection mechanisms, including prevention of SQL injection, cross-site scripting (XSS), cross-site request forgery (CSRF), as well as basic authentication and authorization tools. The limitations of standard solutions in high-load environments have been identified, which determines the need for integration of advanced approaches. A multi-level data protection model in Laravel has been developed, which combines built-in mechanisms with advanced solutions: the use of the Argon2 hashing algorithm instead of bcrypt to enhance cryptographic strength; the implementation of two-factor authentication; the application of access policies and rate limiting to protect APIs; the integration of anomaly detection mechanisms for requests. The proposed model has been formalized through a mathematical description of risks and time costs, which makes it possible to compare the performance of different solutions. Experimental studies were conducted in a test environment using Apache Benchmark and Siege. The obtained results indicate that the use of Argon2 increases the processing time of authentication requests by an average of 12-15% compared to bcrypt, but significantly enhances the level of protection. The use of rate limiting reduced the risk of brute force attacks by 40%, while the implementation of access policies significantly decreased the number of unauthorized access attempts. As a result, it has been proven that the integration of basic and advanced mechanisms forms an optimal model of data protection in Laravel applications, which ensures a balance between performance and security. Further research involves the use of adaptive methods for anomaly detection in traffic and the development of automated tools for assessing the security level of web applications.

Keywords: Laravel; security; optimization; web applications; encryption; API; data protection.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Upadhyay, D., & Ware, N. R. (2023). Evolving trends in web application vulnerabilities: A comparative study of OWASP Top 10 2017 and OWASP Top 10 2021. *International Journal of Engineering Technology and Management Sciences*, *4*(2), 112–120.
2. Serbout, S., El Malki, A., Pautasso, C., & Zdun, U. (2023). API rate limit adoption: A pattern collection. In *Proceedings of the 28th European Conference on Pattern Languages of Programs (EuroPLoP 2023)*. ACM. https://doi.org/10.1145/3628034.3628039
3. Tippe, P., & Berner, M. P. (2025). Evaluating Argon2 adoption and effectiveness in real-world software. In *Proceedings of the International Conference on Availability, Reliability and Security (ARES 2025)* (pp. 25–46). Springer. https://doi.org/10.1007/978-3-032-00627-1_2

4. Boonkrong, S., & Koksungnoen, P. (2025). Identification of optimal Argon2i parameters for performance and security enhancement. *International Journal of Information Technology*. https://doi.org/10.1007/s41870-025-02457-5
5. El Malki, A., Pautasso, C., & Zdun, U. (2023). Impact of API rate limit on reliability of microservices-based architectures [Technical report]. *University of Vienna, Faculty of Computer Science*. https://eprints.cs.univie.ac.at/7399
6. Muthukrishnan, M. (2024). API rate limiting mechanisms in SaaS applications: A systematic analysis of DDoS protection strategies. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, *10*(6), 1787–1798. https://doi.org/10.32628/CSEIT241061223
7. Etese, O. (2025). A review and comparative analysis of password hashing techniques. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.5363433
8. Ntantogian, C. (2019). Evaluation of password hashing schemes in open source frameworks. *Computers & Security*, *85*, 372–385. https://doi.org/10.1016/j.cose.2019.03.011
9. Wong, C., & Bielski, S. (2005). Empirical analysis of rate limiting mechanisms. In *Recent advances in intrusion detection (RAID 2005)* (pp. 22–42). Springer. https://doi.org/10.1007/11663812_2
10. Bai, W., Blocki, J., & Ameri, M. H. (2022). Cost-asymmetric memory hard password hashing. *arXiv Preprint*. https://doi.org/10.48550/arXiv.2206.12970

Отримано редакцією журналу / Received: 14.01.26
Прорецензовано / Revised: 30.01.26
Схвалено до друку / Accepted: 26.03.26