УДК 004.56

**А. В. Притула**
**Л. М. Куперштейн**

# Аналіз проблем безпеки веб-застосунків

Вінницький національний технічний університет

**Анотація**
*В сучасному світі безпека веб-застосунків є одним із ключових аспектів інформаційної безпеки. Цей аналіз зосереджено на основних вразливостях веб-застосунків, визначених OWASP TOP 10 та динаміці їх розвитку за останні роки.*
**Ключові слова:** веб-застосунок, owasp, атака, вразливість.

**Abstract**
*In today's world, web application security is one of the key aspects of information security. This analysis focuses on the main vulnerabilities of web applications identified by OWASP TOP 10 and the dynamics of their development in recent years.*
**Keywords:** web-application, owasp, attack, vulnerability.

## Вступ

В сучасному світі, де технології набувають все більшого значення у житті суспільства, безпека веб-застосунків стає критично важливою. З ростом залежності від онлайн-сервісів, користувачі передають величезні обсяги персональної інформації через веб-застосунки, починаючи від особистих даних і закінчуючи банківською інформацією. Це робить веб-застосунки основною мішенню для зловмисників, які прагнуть отримати несанкціонований доступ до даних або завдати шкоди.

OWASP – міжнародна некомерційна організація, яка зосереджена на покращенні безпеки програмного забезпечення, регулярно публікує список 10 загроз безпеки веб-застосунків [1]. Цей список вважається стандартом у галузі безпеки веб-застосунків та використовується як основний джерело для аналізу поточного стану веб-безпеки.

Зміни у рейтингу загроз показують, як розвиваються тактика та стратегія зловмисників. Розуміння цих динамік дозволяє професіоналам у сфері безпеки адаптуватися до нових загроз і розробляти нові ефективні засоби захисту.

## Результати дослідження

В сучасному цифровому світі веб-застосунки стали ключовим інструментом для багатьох аспектів нашого життя: від соціальних мереж до банківських операцій. Проте разом із зростаючою залежністю від цих додатків збільшується й кількість потенційних загроз безпеки. Кожен день тисячі веб-застосунків стають мішенями для атак, і з кожним роком методи зловмисників стають все хитрішими.

Для професіоналів у сфері безпеки, а також для розробників веб-застосунків, важливо розуміти актуальні загрози та тенденції їхньої зміни. Це допомагає правильно налаштовувати системи безпеки, адаптувати методи розробки та, в кінцевому підсумку, захищати користувачів від можливих атак.

Однією з ключових організацій, яка працює над моніторингом та аналізом загроз для веб-застосунків, є OWASP [1]. Їхній щорічний список TOP 10 відображає найбільш актуальні та розповсюджені загрози для веб-застосунків. Вивчення цього списку дозволяє не лише зрозуміти, які загрози на даний момент є найбільш актуальними, але й простежити за динамікою їхньої зміни протягом часу.

Детальний огляд рейтингу OWASP TOP 10 відкриває можливість аналізу кожної загрози окремо, зосереджуючись не тільки на її основі та потенційному впливі, але й на методах, які зловмисники використовують для адаптації своїх атак у відповідь на зміни технологій та заходів безпеки. Розглянемо та проаналізуємо більш детальніше основні проблеми безпеки web-застосунків.

Broken Access Control – порушення контролю доступу виникає, коли атакуючий може отримати доступ до даних або функцій веб-застосунку, до яких йому не повинно бути доступу [2]. Це може включати отримання даних користувача, зміну контенту або виконання певних дій без відповідних

прав. Зазвичай це відбувається через неправильну конфігурацію систем контролю доступу або їх відсутність. Однією з причин зростання порушень контролю доступу є загальна тенденція до розподіленої архітектури та мікросервісів. Це створює додаткові точки входу та потенційні слабкі місця для атак. Крім того, збільшення обсягу даних, які обробляються веб-застосунками, а також більша інтеграція з іншими системами, збільшили ризик неналежного доступу до даних.

Cryptographic Failures – помилки в області криптографії виникають коли криптографічні механізми використовуються неправильно, коли вони є слабкими або застарілими [3]. Це може призвести до несанкціонованого доступу до конфіденційної інформації або підміни даних. На практиці це може бути використання слабких алгоритмів шифрування, неналежне зберігання ключів чи неправильне використання API для криптографічних операцій. Протягом останнього десятиліття криптографічні помилки стали все більш видимими в списку OWASP. Ростуча залежність від цифрових технологій і зберігання даних в цифровому форматі зробила захист цих даних все більш критичним.

Injection – ін'єкції є однією з найбільш поширених вразливостей в веб-застосунках. Вони виникають, коли додаток відправляє неконтрольований або неперевірений вхідний запит до інтерпретатора [1, 4]. Це може дозволити зловмисникам вставляти або "ін'єктувати" зловмисний код, який буде виконаний від імені додатка. Ці атаки можуть призвести до різних наслідків, залежно від додатка, включаючи несанкціонований доступ до бази даних, виконання команд на сервері або віддалене керування системою. Технології розвиваються, і також зростає кількість додатків, які використовують різноманітні бази даних та сервіси. Це створює більше можливостей для зловмисників проводити атаки ін'єкцій. Додатково, з появою нових мов програмування та технологій, з'являються нові способи реалізації атак ін'єкцій. Тому важливість розуміння та захисту від таких загроз лише зростає.

Insecure Design – небезпечне проектування відноситься до вад у проектуванні додатка, які можуть призвести до вразливостей [5]. Це базова проблема, що лежить в основі багатьох інших загроз. В разі неправильного проектування додатка з початку, можуть виникнути серйозні проблеми з безпекою в майбутньому, незалежно від того, наскільки добре реалізований код. З часом більше уваги стало приділятися безпеці на етапі проектування. Впровадження практики безпечного кодування та безпечного проектування стали більш поширеними в індустрії. Однак, протягом років, додатки ставали все більш складними, що збільшило ймовірність виникнення прогалин в безпеці.

Security Misconfiguration – неправильна конфігурація безпеки є результатом недостатнього або неправильного налаштування параметрів безпеки на рівні додатка, бази даних, мережі, платформи тощо [6]. Це може включати в себе все, від відсутності патчів безпеки до залишення за замовчуванням адміністративних паролів або відображення докладних помилок користувачам. Зловмисник постійно шукають нові способи експлуатації слабких місць, що змушує організації постійно залишатися в курсі змін і підтримувати свої системи належним чином налаштованими.

Vulnerable and Outdated Components – використання вразливих або застарілих компонентів може зробити додаток вразливим до атак [1]. Ці компоненти можуть включати бібліотеки, фреймворки, модулі чи інші зовнішні залежності, які використовуються у додатку. Ця загроза стала більш актуальною з роками, оскільки розробка програмного забезпечення стала більш модульною. Розробники часто покладаються на готові рішення з відкритим кодом для швидкої розробки, замість створення компонентів "з нуля".

Identification and Authentication Failures – помилки в ідентифікації та автентифікації стосуються недоліків у процесах розпізнавання користувача (ідентифікація) і переконання у його справжності (автентифікація) [7]. Ці помилки можуть призвести до того, що несанкціоновані користувачі отримають доступ до системи або привілейованих ресурсів. Причина, чому ця загроза стає більш важливою, полягає у тому, що користувачі стають все більш залежними від онлайн-сервісів у своєму повсякденному житті, в той час як зловмисники розробляють все більше інструментів для атак. Збільшення кількості додатків і сервісів, які користувачі використовують щодня, збільшує кількість точок входу для потенційних атак. Це, у свою чергу, збільшує необхідність в захищених системах ідентифікації та автентифікації.

Software and Data Integrity Failures – помилки цілісності програмного забезпечення та даних відносяться до ненадійності або втрати цілісності даних або коду програмного забезпечення [1]. Це може бути результатом вразливостей, що дозволяють зловмисникам змінювати або знищувати інформацію без відома власника або системи. З роками, з поширенням хмарних технологій і все більшої кількості даних, які обробляються онлайн, цілісність даних стає важливішою. Наприклад,

впровадження DevOps та CI/CD може призвести до швидшого випуску коду, але також може збільшити ризик помилок у цілісності.

Security Logging and Monitoring Failures – ця вразливість відноситься до недостатньої або відсутньої реєстрації подій безпеки, а також до відсутності адекватного моніторингу цих записів [1]. Якщо інциденти безпеки не відстежуються або на них не реагують належним чином, це може призвести до невиявлених порушень та додаткових ризиків. З ростом кіберзлочинності та розвитком технік атак, вимоги до журналювання та моніторингу безпеки збільшились. У минулому існував підхід, коли "не відомо — не болить", але зараз організації усвідомлюють, що невиявлені порушення можуть мати катастрофічні наслідки.

Server-Side Request Forgery (SSRF) – це вектор атаки, який змушує сервер виконувати запити від імені атакуючого [8]. Атакуючий може використовувати SSRF для зондування внутрішньої мережі, взаємодії з іншими службами та отримання даних, до яких він не має прямого доступу. Ростуча популярність хмарних сервісів призвела до збільшення експозиції SSRF-атакам. Багато організацій переміщують свої додатки та інфраструктуру в хмару без повного розуміння архітектурних та безпекових викликів, що це може створити. Це, в свою чергу, створює можливості для атакуючих експлуатувати SSRF вразливості.

Порівняючи загрози протягом останнього десятиліття, можна побачити, що деякі вразливості, такі як SQLi та XSS, залишаються стійкими у списку OWASP TOP 10. Однак їх позиції змінювались. Наприклад, з 2010 по 2020 рік ін’єкції займали найвищі місця в списку, але у 2021 році їх позиція змінилась [1, 4, 9].

Ці зміни можна пояснити декількома факторами:

- технологічний розвиток – нові технології і підходи до розробки можуть зменшувати ризик певних вразливостей, але також можуть вносити нові типи вразливостей;
- зростання обізнаності – як тільки спільнота розробників стає більш обізнаною щодо певної вразливості, з'являються кращі інструменти та практики для її запобігання;
- зміна ландшафту загроз – зловмисники також адаптуються, змінюючи свої тактики та цілі. Наприклад, якщо раніше основний акцент було зроблено на витягування інформації, то зараз може бути більше акценту на втручання в роботу системи або її знищення.

Зростаюча складність веб-застосунків, використання третіх бібліотек та зовнішніх залежностей також можуть впливати на динаміку загроз. Нові технології, такі як контейнери, мікросервіси або серверний код, можуть вносити нові вразливості в додатки, які раніше вважалися безпечними [11].

Тому важливо розуміти, що незалежно від позиції вразливості в списку OWASP TOP 10, кожна з них заслуговує на увагу розробників і професіоналів з безпеки.

## Висновки

Актуальність і важливість веб-безпеки продовжують рости. З огляду на постійно змінювані тактики та методи атак, важливо постійно слідкувати за змінами в рейтингу загроз та адаптувати свої методи захисту відповідно. OWASP TOP 10 є чудовим ресурсом для вивчення сучасних вразливостей та методів їх запобігання.

Однією з ключових складових в захисті веб-застосунків є їх тестування. Регулярне тестування додатків на безпеку, зокрема використання тестування на проникнення та автоматизованих сканерів безпеки, допомагає виявити та усунути потенційні вразливості до того, як їх можуть використати зловмисники. Відсутність відповідного тестування може призвести до втрати конфіденційної інформації, фінансових збитків та інших негативних наслідків для організації.

Крім того, зростання популярності DevSecOps [12], що поєднує процеси розробки, експлуатації та безпеки, підкреслює важливість інтеграції безпеки на всіх етапах розробки додатку. Такий підхід визначає безпеку важливою частиною життєвого циклу додатку, а не додатковою функцією, яка накладається після завершення розробки.

## СПИСОК ВИКОРИСТАННОЇ ЛІТЕРАТУРИ

1. OWASP Top 10:2021. OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation. URL: https://owasp.org/Top10/ (дата звернення: 25.10.2023).

2. What is broken access control vulnerability and how to prevent it - authgear. Authgear - Secure and Simple User Management. URL: https://www.authgear.com/post/what-is-broken-access-control-vulnerability-and-how-to-prevent-it (дата звернення: 25.10.2023).
3. A02 Cryptographic Failures - OWASP Top 10:2021. OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation. URL: https://owasp.org/Top10/A02_2021-Cryptographic_Failures/ (дата звернення: 25.10.2023).
4. Voitovych O. P., Yuvkovetskyi O. S., Kupershtein L. M. SQL injection prevention system. 2016 International Conference "Radio Electronics & Info Communications" (UkrMiCo), Kyiv, Ukraine, 11–16 September 2016. 2016. URL: https://doi.org/10.1109/ukrmico.2016.7739642 (дата звернення: 25.10.2023).
5. Insecure design | tutorials & examples | snyk learn. Snyk Learn. URL: https://learn.snyk.io/lesson/insecure-design/ (дата звернення: 25.10.2023).
6. Eshete B., Villafiorita A., Weldemariam K. Early Detection of Security Misconfiguration Vulnerabilities in Web Applications. 2011 Sixth International Conference on Availability, Reliability and Security (ARES), м. Vienna, Austria, 22–26 серп. 2011 р. 2011. URL: https://doi.org/10.1109/ares.2011.31 (дата звернення: 26.10.2023).
7. A07 identification and authentication failures - OWASP top 10:2021. OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation. URL: https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures/ (дата звернення: 26.10.2023).
8. Server-Side request forgery (SSRF). Imperva. URL: https://www.imperva.com/learn/application-security/server-side-request-forgery-ssrf/ (дата звернення: 26.10.2023).
9. OWASP Top 10 Vulnerabilities in 2013 | Indusface Blog. Indusface. URL: https://www.indusface.com/blog/owasp-top-10-vulnerabilities-2013/ (дата звернення: 26.10.2023).
10.OWASP Top Ten 2017 | 2017 Top 10 | OWASP Foundation. OWASP Foundation, the Open Source Foundation for Application Security | OWASP Foundation. URL: https://owasp.org/www-project-top-ten/2017/Top_10 (дата звернення: 26.10.2023).
11.Dejwakh V. Docker and the rise of microservices. DEV Community. URL: https://dev.to/dej/docker-and-the-rise-of-microservices-16lc (дата звернення: 27.10.2023).
12.What is DevSecOps? | IBM. IBM in Deutschland, Österreich und der Schweiz | IBM. URL: https://www.ibm.com/topics/devsecops#:~:text=DevSecOps– short%20for%20development,%20security,, %20deployment,%20and%20software%20delivery (дата звернення: 27.10.2023).

***Притула Андрій Вікторович*** – студент групи 125-23а, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, Вінниця, e-mail: andrik.pritula@gmail.com.

***Куперштейн Леонід Михайлович*** **–** к.т.н., доцент кафедри захисту інформації, Вінницький національний технічний університет, м. Вінниця email: kupershtein.lm@gmail.com

***Prytula Andrii V.*** **–** Student of Faculty of Information Technologies and Computer Engineering, Vinnytsia National Technical University, e-mail: andrik.pritula@gmail.com.

***Kupershtein Leonid M.*** – PhD, Associated Professor of Information Protection Chair, Vinnytsia National Technical University, Vinnytsia, email: kupershtein.lm@gmail.com