protection system that takes into account network packet anomalies. The detection and processing of such anomalies allows you to quickly identify and neutralize threats where DDos attacks occupy a special place. This article describes how to analyze network traffic in real time based on statistical methods and machine learning algorithms and classify network packets according to their behavioral characteristics [1]. The system implements a multi-layered approach to server protection, which includes three main stages: initial data filtering, statistical analysis, and the use of machine learning models. At the first stage, malicious packets are excluded based on simple criteria, such as forbidden IP addresses and incorrect packet formats.[2] In Phase 2, statistical analysis is used to detect deviations in the traffic distribution, for example, a sharp increase in the number of requests or a change in packet size [3]. The third stage involves the use of classifiers trained with historical data to identify anomalies in network operation. The list of presented models allows you to adapt to new types of attacks by automatically updating [4]. The advantages of the presented system are: It detects both traditional DDoS attacks (port scans, exploits of network protocol vulnerabilities, and SQL injection attempts) and other types of threats. Second, integration with existing monitoring tools and firewalls. Integration with existing monitoring tools and firewalls [5] also

Communication Technologies, Kyiv, Ukraine

makes it easy to implement without significant cost increases. The system is characterized by high attack detection accuracy, low false positive rate. It provides efficient real-time server protection to ensure business continuity and prevent financial and reputation loss.

Keywords: anomalies in packets, DDoS attacks, machine learning, traffic analysis, server protection.

Поночовний Петро Михайлович, аспірант кафедри Технічних систем кіберзахисту Державного університету інформаційно-комунікаційних технологій, Київ, Україна.

Ponochovny Petro Mykhailovych, PhD student of department Technical system cyber security of the State University of Information and Communication Technologies, Kyiv, Ukraine.

E-mail: petja9186@gmail.com

ORCID ID 0009-0008-6480-6990

Пепа Юрій Володимирович, к.т.н., доцент, професор кафедри Технічних систем кіберзахисту Державного університету інформаційно-комунікаційних технологій, Київ, Україн.а

Pepa Yuriy Volodymyrovych, Ph.D., docent, professor of department Technical system cyber security of the State University of Information and

E-mail: yurka14@ukr.net.

ORCID ID 0000-0003-2073-1364

УДК 004.056.5

СИСТЕМНИЙ ПІДХІД ДО БЕЗПЕКИ ВЕБДОДАТКІВ: АНАЛІЗ ЗАГРОЗ ТА МЕТОДИ КІБЕРЗАХИСТУ

Анна Ільєнко, Денис Спис, Лілія Галата, Олена Дубчак

Дослідження присвячене аналізу поширених вразливостей вебдодатків, їх впливу на безпеку інформаційних систем, економічні, репутаційні та правові наслідки, а також методи їх виявлення та усунення. Проведено всебічний огляд сучасного стану безпеки вебдодатків, включаючи статистичні дані щодо актуальних загроз, аналіз тенденцій розвитку атак та розгляд найвідоміших інцидентів останніх років. Особливу увагу приділено порівнянню різних підходів до класифікації вразливостей, зокрема OWASP Top 10, CWE Top 25, MITRE ATT&CK, NIST SP 800-53 та інших стандартів, з метою оцінки їх ефективності та практичного застосування. У роботі розглянуто методи тестування безпеки вебдодатків, зокрема статичний (SAST), динамічний (DAST) та інтерактивний аналіз (IAST), а також можливості застосування штучного інтелекту та машинного навчання (AI/ML) для автоматизованого виявлення загроз. Досліджено переваги та недоліки різних методів кіберзахисту, а також практичні аспекти їх використання у реальних умовах. Крім того, проведено детальний аналіз впливу вразливостей на організації, зокрема їх економічні наслідки, репутаційні ризики, а також правові наслідки. Результати дослідження дозволяють сформувати комплексний підхід до мінімізації ризиків, що включає впровадження передових методів аналізу безпеки, використання рекомендацій міжнародних стандартів та застосування сучасних інструментів кіберзахисту. Це забезпечує ефективніше виявлення та усунення загроз на ранніх етапах розробки та експлуатації вебдодатків.

***Ключові слова**: кібербезпека, вразливості вебдодатків, OWASP Top 10, статичний аналіз коду, динамічний аналіз, машинне навчання, автоматизоване виявлення вразливостей, кіберзагрози.*

Вступ

У сучасному цифровому середовищі вебдодатки стали невід'ємною складовою бізнес-процесів та повсякденного життя – від фінансових операцій до соціальних комунікацій. Більшість компаній використовують вебсервіси для підтримки своєї діяльності, що зумовлює зростання попиту на веброзробку. Водночас із розширенням функціональності вебтехнологій зростає і кількість вразливостей у них, що ставить під загрозу конфіденційність, цілісність і доступність даних користувачів.

За даними OWASP та інших досліджень, саме вразливості вебдодатків часто стають причиною успішних кібератак. Зловмисники активно націлюються на вебдодатки, оскільки ті нерідко містять критично важливі персональні чи фінансові дані і водночас можуть мати слабкі місця через помилки розробки або некоректну конфігурацію. Вразливості вебдодатків призводять до значних збитків компаній – як прямих фінансових, так і репутаційних втрат, а також правових наслідків. Однак, попри усвідомлення загрози, ефективне виявлення та усунення вебвразливостей залишається складною проблемою. Багато вебдодатків містять прорахунки в архітектурі чи конфігурації, що створює небезпечні лазівки для атак. Класичним прикладом є атаки типу SQL-ін'єкцій або міжсайтового скриптингу (XSS), що роками залишаються одними з найпоширеніших загроз. Систематизація знань про такі вразливі місця є критичною для розробки ефективних методів їх своєчасного виявлення та усунення. Забезпечення безпеки вебдодатків виступає ключовою умовою їх надійного функціонування та збереження довіри користувачів. Таким чином, актуальною науково-практичною проблемою є розробка ефективніших методів своєчасного виявлення та попередження вразливостей вебдодатків з урахуванням їх різноманітності й еволюції методів атак. Оскільки проблема безпеки вебдодатків є багатогранною, її вирішення потребує комплексного підходу, що поєднує розробку технологічних рішень, вдосконалення процесів безпечної розробки та підвищення обізнаності фахівців у сфері кібербезпеки. У цьому контексті важливе місце займає аналіз існуючих вразливостей, оцінка їхнього впливу та розробка ефективних методів їхнього виявлення та запобігання. Ця стаття спрямована на всебічний розгляд поширених вразливостей вебдодатків, їхню класифікацію, оцінку сучасних тенденцій у сфері атак та аналіз методів мінімізації ризиків. Отримані результати можуть слугувати основою для подальших досліджень у напрямку автоматизації процесів виявлення вразливостей та розробки інноваційних підходів до підвищення рівня безпеки вебдодатків.

Постановка задачі

Аналіз поширених вразливостей вебдодатків та огляд ефективних підходів до їхнього виявлення є основним об'єктом даного дослідження. Незважаючи на активні зусилля у сфері кібербезпеки, більшість атак на вебдодатки експлуатують добре відомі вразливості, які виникають через помилки в архітектурі, розробці чи конфігурації. Важливим аспектом є не лише виявлення цих вразливостей, а й мінімізація їх ризиків за допомогою сучасних методів аналізу, таких як статичний (SAST), динамічний (DAST), інтерактивний (IAST) аналіз та машинне навчання

Мета дослідження – здійснити аналіз поширених вразливостей вебдодатків, визначити їхній вплив на безпеку інформаційних систем та бізнес-процеси, а також дослідити сучасні методи їхнього виявлення, класифікації та мінімізації ризиків. Досягнення поставленої мети передбачає вирішення таких завдань: провести огляд сучасного стану безпеки вебдодатків, визначити ключові тенденції розвитку кіберзагроз та їхній вплив на різні галузі; проаналізувати наукові дослідження з цієї тематики для оцінки поточного рівня безпеки вебдодатків, виявлення основних підходів до ідентифікації та попередження вразливостей, а також визначення недостатньо досліджених аспектів і перспективних напрямів подальших досліджень; дослідити розвиток методів атак і появу нових загроз, розглянути реальні випадки кібератак на вебдодатки, їхні причини та наслідки; оцінити вплив типових вразливостей вебдодатків на діяльність

організацій, зокрема економічні збитки, репутаційні ризики та правові наслідки; дослідити класифікацію та основні категорії поширених вразливостей вебдодатків, зокрема модель OWASP Top 10 та альтернативні підходи (CWE, MITRE ATT&CK тощо); проаналізувати методи виявлення вразливостей, зокрема статичний (SAST), динамічний (DAST) та інтерактивний (IAST) аналіз, а також можливості застосування штучного інтелекту та машинного навчання (AI/ML) для автоматизації процесу виявлення загроз.

1. Огляд сучасного стану безпеки вебдодатків

1.1 Статистичні дані і тенденції атак

Вебдодатки сьогодні належать до найбільш атакованих компонентів інформаційних систем. За останніми дослідженнями Verizon, майже половина всіх інцидентів кібербезпеки пов'язана саме з вебдодатками. Згідно з Verizon DBIR 2024, вебдодатки стали основним вектором початкового проникнення: їх залучено приблизно у 50% проаналізованих порушень безпеки [1]. У 2023 році, за даними звіту Verizon, атаки через вебдодатки були використані у 80% зафіксованих інцидентів і призвели до 60% витоків даних. Така статистика підкреслює критичність захисту вебсервісів. Масштаб загрози підтверджується і кількісно: наприклад, лише за 2023 рік засобами компанії Barracuda було заблоковано понад 18 млрд атак на вебдодатки та API (у середньому ~1,7 млрд атак на місяць) [2]. Це вказує на те, що в середньому вебресурси піддаються безперервним автоматизованим спробам зламу.

Частота атак на вебдодатки неухильно зростає в усіх галузях (рис. 1).

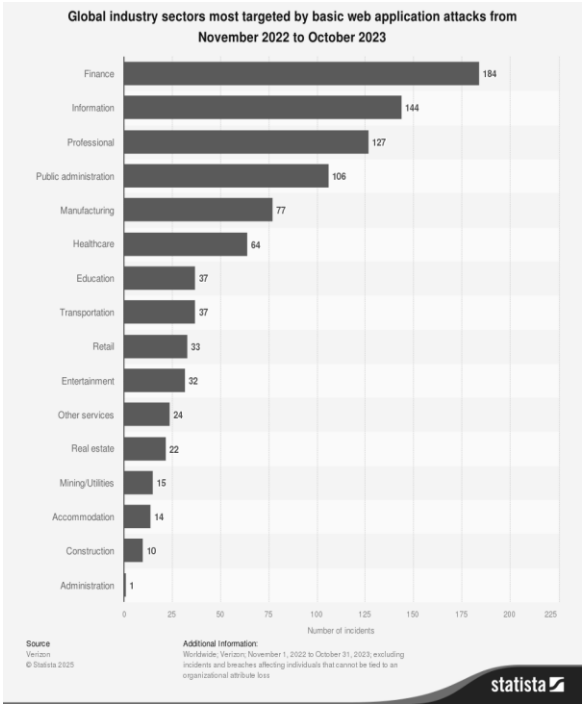

Рисунок 1 Галузі, які найбільше постраждали від атак на вебдодатки в період з листопада 2022 по жовтень 2023 року

Статистика свідчить, що зловмисники найбільше націлюються на фінансовий сектор: так, у період з листопада 2022 по жовтень 2023 року у фінансовій індустрії було зафіксовано 184 інциденти атак через вебдодатки – більше, ніж у будь-якій іншій галузі [3]. Другим за кількістю став інформаційний сектор (144 атаки), а третє місце посіли професійні та комерційні послуги (близько 127 інцидентів). Ці дані корелюють з мотивами нападників: фінансові системи обробляють грошові транзакції, а отже є найпривабливішими; інформаційні служби і бізнес-послуги володіють значними масивами персональних даних, що теж становлять цінність.

Окрім зростання абсолютної кількості атак, спостерігається тривожна тенденція збільшення частки інцидентів, у яких зловмисники експлуатують відомі вразливості для проникнення. За даними Verizon DBIR 2024, кількість успішних зламів, здійснених через експлуатацію вразливостей, зросла майже

втричі (на 180%) порівняно з попереднім роком [4]. Такий різкий стрибок пов'язаний як із широким використанням 0-day експлойтів, так і з тим, що багато систем залишаються невиправленими належним чином. Іншими словами, вразливості, про які відомо і для яких вже існують патчі, продовжують залишатися «відкритими дверима» для нападників. Наприклад, комплекс вразливостей ProxyShell у Microsoft Exchange (виявлений у 2021 р.) і надалі активно використовується кіберзлочинцями, спричиняючи численні компрометації та випадки зараження ransomware [1]. Це свідчить про недостатній рівень своєчасного оновлення програмних компонентів і виправлення відомих багів безпеки.

1.2 Огляд наукових публікацій щодо виявлення вразливостей вебдодатків

Сучасні вебдодатки зазвичай є ціллю різноманітних атак через їхні поширені вразливості. За останні п'ять років було опубліковано чимало досліджень, присвячених аналізу таких вразливостей та методів їх виявлення і усунення. Нижче представлено огляд ключових публікацій 2019–2024 рр. з цієї тематики з оцінкою використаних у них підходів, а також їх сильних і слабких сторін. На основі цього огляду сформульовано висновки про недостатньо досліджені аспекти безпеки вебдодатків, що підкреслює актуальність нових підходів.

Огляди стану веббезпеки. «Web Application Security: A Pragmatic Exposé» (C. C. Aladi, 2024) – новітній огляд останніх чотирьох років досліджень у сфері безпеки вебдодатків. В цій роботі автор відзначає, що атаки вводу коду (injection) залишаються одними з найпоширеніших загроз для вебдодатків [5]. На основі аналізу попередніх досліджень сформульовано практичні рекомендації для захисту від найбільш розповсюджених атак. Зокрема, стаття пропонує практичний дороговказ

для фахівців із безпеки щодо впровадження перевірених заходів захисту, тим самим поєднуючи теорію з практикою. Перевагою роботи є актуальний фокус на реальних загрозах і практичних рішеннях; недоліком – обмеження огляду лише останніми роками та акцент здебільшого на ін'єкційних атаках, без детального розгляду інших категорій вразливостей.

«Deep learning trends and future perspectives of web security and vulnerabilities» (M. S. Chughtai et al., 2024) – ґрунтовний огляд застосування штучного інтелекту для захисту вебдодатків. У статті визначено перелік критичних вразливостей вебдодатків, проаналізовано засоби їх виявлення – статичні, динамічні та гібридні – а також новітні інструменти на основі AI [6]. Зокрема, огляд систематизує напрацювання з використання машинного й глибинного навчання для виявлення атак (в т.ч. методи інтрузивного виявлення) та виявляє прогалини у нинішніх підходах. Наприклад, автори відзначають нестачу об'єднаних рішень, що поєднують статичний та динамічний аналіз із AI, а також проблеми із виявленням більш прихованих атак. Серед сильних сторін цієї роботи – всеохопний характер (проаналізовано 129 джерел) та визначення перспективних напрямів (як-от потреба у вдосконалених моделях глибинного навчання для пошуку вразливостей). Водночас недоліком є відсутність експериментальної перевірки висновків (робота є оглядовою) і залежність від якості досліджень, включених до аналізу.

Статистичний аналіз поширених вразливостей. «Classifying common security vulnerabilities by software type» (O. Ezenwoye, Y. Liu, W. Patten, 2020) – дослідження, що пропонує класифікацію вразливостей за типом програмного забезпечення. Автори проаналізували 51 110 записів CVE (2015–2019) з Національної бази даних вразливостей, розподіливши їх за категоріями: операційні системи, веббраузери, сервери, вебдодатки тощо [7]. Це дозволило виявити, наскільки різняться типи слабких місць залежно від сфери застосування. З'ясувалося, що лише 7 найбільш поширених типів вразливостей (зокрема, переповнення буфера та Cross-Site Scripting) становлять понад 53% всіх випадків. Важливо, що саме у вебдодатках спостерігається найвища повторюваність одних і тих самих вразливостей між різними версіями продукту – тобто певні слабкі місця (наприклад, XSS) з року в рік знову виявляються в нових вебпродуктах. Це підкреслює проблему: типові помилки безпеки вебдодатків не зникають повністю попри виправлення, можливо через повторне допущення розробниками. Перевага роботи – аналіз великого масиву даних, що дає статистично обґрунтовану картину ландшафту загроз. Недоліком є описовий характер: автори не пропонують рішень для зменшення цих

вразливостей, а лише констатують факти. Пропущеним залишається питання причин такої повторюваності і шляхів запобігання їй (наприклад, вдосконалення процесів розробки з урахуванням безпеки).

Оцінка ефективності сканерів вразливостей. «Evaluation of Black-Box Web Application Security Scanners in Detecting Injection Vulnerabilities» (M. Althunayyan et al., 2022) – експериментальне дослідження ефективності наявних автоматичних сканерів безпеки для вебдодатків. Автори протестували п'ять популярних сканерів (OWASP ZAP, Burp Suite Pro та ін.) на сучасному вразливому вебдодатку e-commerce з впровадженими помилками типу SQL-ін'єкцій, NoSQL- ін'єкцій та серверних шаблонних ін'єкцій (SSTI) [8]. Результати показали тривожну картину: більшість ін'єкційних вразливостей залишились не знайденими автоматичними сканерами – у деяких режимах роботи інструменти пропустили всі наявні вразливості. Наприклад, жоден зі сканерів не зміг виявити атаки NoSQL чи SSTI, а виявлення навіть типових SQL-ін'єкцій було фрагментарним. Хоча хибнопозитивних спрацювань майже не було, показник повноти (recall) для більшості засобів виявився дуже низьким – у межах 0–29%. Перевагою цього дослідження є об'єктивне виявлення недоліків актуальних сканерів безпеки на реалістичному сценарії – воно чітко вказує, які типи атак недостатньо покриваються існуючими інструментами. Слабка сторона – сфокусованість лише на ін'єкційних атаках (OWASP Top 10 A1), без оцінки інших видів вразливостей (наприклад, XSS, CSRF), а також використання одного тестового застосунку. Пропущеним аспектом є питання, як саме покращити ці сканери – автори лише констатували проблеми, не запропонувавши конкретних рішень у цій роботі.

«Artificial Intelligence and Dynamic Analysis-Based Web Application Vulnerability Scanner» (M. A. Yalçınkaya, E. U. Küçüksille, 2024) – дослідження, що представляє удосконалений сканер вразливостей з елементами AI. У цій роботі автори інтегрували методи штучного інтелекту з динамічним аналізом для автоматизованого пен-тестування вебдодатків. Зокрема, реалізовано підхід з нечіткою класифікацією вебсторінок перед скануванням: на першому етапі сканер аналізує тип сторінки (наприклад, форма логіну, сторінка пошуку тощо) та прогнозує, які вразливості можуть бути притаманні даному компоненту. На основі цієї класифікації далі вибираються найбільш релевантні тести (ін'єкції, XSS тощо) для даної сторінки [9]. Розроблений сканер було перевірено на спеціально створеному полігоні з 262 тестовими вебсторінками, що містили відомі вразливості різних типів. За результатами експерименту, AI-сканер виявив значну частину вразливостей (краще, ніж типові інструменти без AI), причому класифікація сторінок дозволила знизити кількість непотрібних перевірок і хибних спрацювань. Перевага підходу – інтелектуальне сканування, яке враховує контекст сторінки: це підвищує ефективність і повноту тестування у порівнянні зі стандартними «сліпими» сканерами. До недоліків можна віднести необхідність попереднього навчання моделі на даних про вебсторінки; також метод потребує перевірки на реальних вебдодатках у промислових умовах. Пропущеним напрямом дослідження є поширення такого AI-підходу на інші типи атак і адаптація до новітніх вебтехнологій (SPA, мікросервіси тощо), які автори не розглядали безпосередньо.

Статичний аналіз коду для виявлення вразливостей. «Splendor: Static Detection of Stored XSS in Modern Web Applications» (He Su et al., 2023) – представляє новий метод статичного аналізу для пошуку міжсайтового скриптінгу (XSS) у вебдодатках. Це перша комплексна статична методика виявлення Stored XSS-вразливостей у сучасних вебдодатках, яка аналізує вихідний код з урахуванням взаємодії з базою даних [10]. Інструмент Splendor, запропонований авторами, здатен відстежувати «розірвані» потоки даних: він поєднує прямий аналіз від джерела введення та зворотний аналіз від точки виходу (sink), а також відновлює зв'язки через БД завдяки спеціальному евристичному зіставленню токенів. Це дозволяє статично побудувати повний ланцюг передачі даних, навіть якщо частина вводу тимчасово зберігається у базі (типова ситуація для Stored XSS). На вибірці з 5 реальних вебдодатків Splendor значно перевершив існуючі статичні та динамічні інструменти: з його допомогою вдалося виявити всі закладені XSS-вразливості, тоді як аналоги пропускали частину з них. Більше того, виявлено 17 раніше невідомих вразливостей (0-day) у популярних open-source проєктах, що демонструє практичну цінність

підходу. До переваг роботи належать висока точність (наприклад, точність ідентифікації місць доступу до БД склала 91.3% і здатність знаходити вразливості, недосяжні для наявних сканерів. Недоліком є вузька спеціалізація: метод спрямовано лише на Stored XSS і реалізовано для PHP-застосунків, тож він не охоплює інші типи XSS (відбитий чи DOM-базований) або інші мови програмування. Відповідно, подальші дослідження потрібні для поширення цього підходу на інші класи вразливостей і середовища (наприклад, підтримка JavaScript-фреймворків, де XSS також поширені).

Публікації, розглянуті вище, узагальнено в таблиці 1, де порівнюються їхні методи, результати, сильні та слабкі сторони, а також окреслюються не до кінця вивчені питання.

Таблиця 1

Результати порівняння

Назва роботи (рік) та автори	Основний метод дослідження	Основні результати
Web Application Security: A Pragmatic Exposé ladi (2024) DOI: 10.1145/3644394	Огляд літератури (2018–2023); аналіз рекомендацій з практики безпеки веб	Визначено, що ін'єкційні атаки залишаються найбільш поширеною загрозою; узагальнено перелік практичних заходів протидії типовим вебатакам
Deep learning trends and future perspectives of web security and vulnerabilities . Chughtai et al. (2024) I: 10.3233/jhs- 230037	Огляд (+ часткова систематизація) 129 джерел; порівняння статичних, динамічних і гібридних засобів захисту; аналіз застосувань AI/ML	Складено таксономію поширених вразливостей вебдодатків; порівняно існуючі інструменти сканування (статичні/динамічні); проаналізовано підходи з AI та виявлено кілька напрямів, де досліджень бракує (наприклад, інтеграція ML в процес розробки)
Classifying common security vulnerabilities by software type Ezenwoye et al. (2020) DOI: 10.18293/SEKE2020-047	Статистичний аналіз CVE (51 110 записів, 2015–2019); класифікація вразливостей за типом ПЗ	Візуалізовано розподіл вразливостей: ~7 ключових типів (напр., Buffer Overflow, XSS) становлять >50% всіх випадків; для різних сфер ПЗ спектр вразливостей різниться. вебдодатки виділяються найвищою повторюваністю тих самих вразливостей між роками
Evaluation of Black-Box Web App Security Scanners in Detecting Injection Vulnerabilities lthunayyan et al. (2022) DOI: 10.3390/electronic s11132049	Експериментальне тестування (ZAP, Burp тощо) на спеціально підготовленому вразливому вебзастосунку; порівняльна оцінка точності (TP/FP/FN)	Виявлено низьку ефективність сучасних сканерів щодо ін'єкцій: стандартні інструменти пропустили більшість навмисно вставлених вразливостей (SQLi/NoSQL/SSTI); деякі сканери не знайшли жодної з новітніх (NoSQL, SSTI) атак mdpi.com
Artificial Intelligence and Dynamic Analysis-Based Web App Vulnerability Scanner Yalçınkaya & E. U. Küçüksille (2024) DOI: 10.22042/isecure. 2023.367746.847	Розробка власного вебсканера з AI: поєднання динамічного сканування з алгоритмами машинного навчання (нечітка класифікація сторінок, адаптивний вибір тестів); випробування на наборі з 262 тестових сторінок	Продемонстровано, що AI-сканер може ефективніше виявляти вразливості: класифікація дозволила спрямувати атаки туди, де вони найбільш ймовірні, підвищивши коефіцієнт виявлення у порівнянні зі стандартними інструментами. Зменшено кількість помилкових спрацювань завдяки контекстному підходу
Splendor: Static Detection of Stored XSS in Modern Web Apps He Su et al. (2023) DOI: 10.1145/3597926. 3598116	Статичний аналіз коду + аналіз БД: побудова повного тракту даних від вводу до виводу для виявлення stored XSS; розробка інструменту (PHP) і порівняльне тестування зі статичними та динамічними детекторами	Успішно виявлено всі закладені XSS-вразливості у тестових застосунках, що істотно краще за інші сканери; інструмент також знайшов 17 нових XSS-вразливості у популярних проєктах (які раніше не були відомі). Показано принципову здійсненність статично виявляти складні вразливості, пов'язані з БД

Аналіз публікацій показує, що, попри прогрес у засобах захисту, багато типових вразливостей вебдодатків залишаються актуальними. Дослідники приділили значну увагу ін'єкційним атакам (SQL/NoSQL injection, XSS), розробивши як статичні, так і динамічні методи їх виявлення. Переваги сучасних підходів полягають у підвищенні автоматизації (сканери, інструменти аналізу коду) та застосуванні AI для покращення точності і швидкості. Наприклад, глибинне навчання дозволяє ефективніше знаходити вразливості у великому обсязі коду, а інтелектуальні сканери можуть адаптивно досліджувати складні вебзастосунки. Однак, огляд виявив і низку недоліків та прогалин. Більшість наявних інструментів мають обмеження у покритті нових видів атак (як-то NoSQL- ін'єкції, SSTI), або фокусуються лише на окремих вразливостях, залишаючи осторонь інші ризики (CSRF, помилки бізнес логіки, помилки конфігурації тощо). Статичні та динамічні методи часто існують відокремлено: відсутня інтегрована платформа, що поєднала б переваги обох підходів – це відзначено як напрям для майбутніх досліджень. Крім того, практична повторюваність тих самих вразливостей у вебдодатках натякає на прогалини в процесі розробки (відсутність достатнього впровадження secure coding практик), що мало досліджено. Недостатньо уваги приділено й автоматизованому виправленню знайдених вразливостей – нинішні роботи зосереджені переважно на виявленні.

Таким чином, існує чітке обґрунтування для розвитку нових підходів у цій сфері. Необхідні більш розумні сканери та аналізатори, здатні виявляти широке коло вразливостей, включно з тими, що важко діагностуються традиційними методами. Поєднання методів (статичний + динамічний аналіз + AI) обіцяє підвищити покриття і точність. Дослідження, що інтегрують безпеку на ранніх етапах розробки (DevSecOps) і автоматично попереджають розробників про помилки, також є перспективними. Недостатньо вивчені аспекти – такі як безпека односторінкових застосунків, мікросервісів, а також виявлення логічних вразливостей – відкривають можливості для нових наукових рішень. Враховуючи виявлені прогалини, розробка нових підходів до забезпечення безпеки вебдодатків є актуальною та необхідною для випередження зловмисників і зменшення ризиків кібербезпеки.

1.3 Сучасні методи атак, відомі інциденти та наслідки

Методи атак на вебдодатки постійно еволюціонують, стаючи дедалі більш витонченими. Традиційні види атак (SQL-ін'єкції, XSS, CSRF тощо) залишаються актуальними, але до них додаються складніші багатокрокові атаки, що поєднують кілька векторів і технік. Одна з ключових сучасних тенденцій – активне залучення засобів штучного інтелекту (ШІ) для підготовки та проведення нападів. Злочинці дедалі частіше застосовують алгоритми машинного навчання для автоматизації збору інформації про ціль і пошуку слабких місць. ШІ допомагає більш точно адаптувати атаки під конкретних жертв: зокрема, генерувати правдоподібні фішингові повідомлення, персоналізовані на основі поведінки користувача [11]. Як відзначають дослідники, роль ШІ полягає в тому, щоб пристосувати атаку до профілю конкретної цільової аудиторії. Наприклад, з'явилися інструменти для автоматизованого фішингу, що аналізують дописи користувача в соцмережах і генерують індивідуальні підроблені повідомлення від знайомих йому осіб. Такі високоточні атаки соціальної інженерії значно підвищують імовірність успіху.

Окрім фішингу, ШІ використовується зловмисниками і для технічних атак – зокрема, для автоматизованого виявлення вразливостей у коді або обходу засобів захисту. Застосування нейромереж для сканування вихідного коду та конфігурацій дозволяє швидше знаходити невиправлені помилки безпеки. Також фіксуються випадки використання ШІ для підробки голосу чи зображень (deepfake) з метою отримання несанкціонованого доступу (наприклад, голосове шахрайство vishing). Таким чином, розвиток ШІ дає нові інструменти як захисникам, так і нападникам, проте в руках останніх він суттєво підсилює ефективність атак.

Варто зауважити, що загальний рівень кіберзагроз досяг історичного максимуму. За оцінками Європолу та інших організацій, у 2024 році кіберзлочинність спричинила глобальні збитки на суму близько €10 млрд, що вдвічі більше ніж попереднього року [11]. Кібератаки становлять дедалі більшу частку всіх правопорушень: до 20% злочинів у світі зараз скоюються з використанням IT-

інфраструктури. Ці цифри підкреслюють, що питання безпеки вебдодатків є надзвичайно актуальним, і виклики в цій сфері будуть тільки зростати.

Останні роки багаті на гучні інциденти, пов'язані з успішною експлуатацією вразливостей вебдодатків. Одним з найбільш резонансних випадків став витік даних компанії Equifax (США) у 2017 році. В результаті експлуатації вразливості в вебкомпоненті (відомий баг в Apache Struts) зловмисники отримали доступ до персональних даних 147 мільйонів користувачів [12]. Цей інцидент завдав компанії колосальної шкоди: окрім репутаційних втрат, Equifax довелося виплатити щонайменше $575 млн (а загалом до $700 млн) у межах мирової угоди з регуляторами та позовниками. Випадок Equifax показав, що одна критична вебвразливість у широко доступному сервісі може призвести до системної кризи безпеки із багаторічними правовими та фінансовими наслідками.

Інший приклад – масштабний витік даних з Twitter, виявлений на початку 2023 року. В результаті хакерської акції в Інтернет було викладено базу, що містила електронні адреси більш ніж

200 млн користувачів Twitter [13]. Хоча Twitter заперечував наявність нової вразливості та припускав, що дані могли бути зібрані з попередніх зламів або парсингу, факт залишається: величезний масив облікових записів став публічно доступним. Експерти відзначили, що цей витік створює підвищені ризики для користувачів – зокрема, масову розсилку таргетованого фішингу, спроби викрадення облікових записів через функцію скидання паролю (reset password) тощо. Фірма Hudson Rock назвала цей інцидент «одним з найзначніших витоків, що я коли-небудь бачив». Випадок Twitter підкреслює, що навіть найбільші технологічні компанії не застраховані від компрометації даних користувачів, а наслідки таких подій – підрив довіри аудиторії та увага регуляторів (наприклад, розслідування, розпочаті ЄС та FTC щодо дотримання Twitter вимог захисту даних після витоку).

Серед інших прикладів можна згадати компрометацію компанії SolarWinds у 2020 р., коли через вразливість та бекдор у їх програмному забезпеченні було уражено ланцюг постачання й скомпрометовано мережі сотень організацій по всьому світу. Атака на SolarWinds стала тривожним сигналом щодо необхідності посилення контролю безпеки на всіх етапах розробки та доставки програмних продуктів. Також показовими є інциденти з витоком даних користувачів Facebook (533 млн записів у 2021 р.), Yahoo (2013–2014 рр., всі 3 млрд акаунтів) – ці випадки мали суттєві негативні наслідки для компаній, включно зі знеціненням бренду і багатомільйонними видатками на врегулювання претензій.

Таким чином, реальні кейси підтверджують: поширені вразливості вебдодатків не є абстрактною загрозою, а призводять до масштабних інцидентів з реальними втратами. Далі розглянемо класифікацію таких вразливостей та підходи до їх систематизації, що допомагає краще розуміти природу ризиків.

2. Дослідження вразливостей вебдодатків

Одним із найавторитетніших джерел класифікації вебвразливостей є проект OWASP Top 10 (Open Worldwide Application Security Project Top 10). OWASP – це некомерційна спільнота експертів з усього світу, що займається покращенням безпеки софтверних продуктів. Результатом роботи цієї спільноти став, зокрема, перелік Топ-10 найбільш критичних ризиків безпеки вебдодатків, який оновлюється кожні кілька років на основі колективного досвіду і статистики виявлених вразливостей [14]. Звіт OWASP Top 10 надає розробникам і фахівцям з безпеки узагальнену інформацію про найпоширеніші типи вразливостей, ранжовані за поширеністю, небезпечністю та потенційним впливом. Цей перелік фактично слугує де-факто стандартом галузі: неврахування вимог OWASP Top 10 при розробці часто розглядається аудиторами як ознака слабкої безпекової практики організації. Навпаки, інтеграція принципів Top 10 у життєвий цикл розробки (SDLC) демонструє прихильність компанії до передових практик захищеного програмування. Багато регуляторних стандартів (наприклад, PCI DSS для платіжних систем) фактично вимагають усунення або контролю вразливостей зі списку OWASP Top 10.

Остання актуальна версія OWASP Top 10 була випущена у 2021 році (попередня – 2017 р.). У ній відбулися суттєві зміни порівняно з версією 2017, що відображають еволюцію загроз. По-перше, на 1-е місце піднялася категорія Broken Access Control («Порушення контролю доступу»), яка витіснила з верхньої

позиції Injection (що трималася лідером з 2010 р.) [15]. Це означає, що проблеми з контролем доступу (авторизацією) стали найпоширенішими і найнебезпечнішими на сьогодні. По-друге, було перейменовано та переосмислено деякі категорії: так, Cryptographic Failures («Криптографічні помилки») прийшли на зміну колишній категорії Sensitive Data Exposure, розширивши її до ширшого кола питань захисту даних. Injection (ін'єкції) залишилися у списку, але тепер охоплюють також XSS, оскільки міжсайтовий скриптинг було інтегровано як різновид ін'єкційної атаки. З'явилися і нові категорії – зокрема, Insecure Design («Небезпечне проектування»), що робить акцент на необхідності врахування безпеки ще на етапі архітектурного дизайну систем. Крім того, вперше до Top 10 було додано категорію Server-Side Request Forgery (SSRF) – атаки з підробкою серверних запитів, які останніми роками спричиняли резонансні інциденти. Включення SSRF під номером 10 свідчить, що хоча частота таких атак поки відносно

невисока, їхній потенційний вплив великий (особливо у хмарних середовищах). Отже, OWASP Top

10 (2021) відобразив сучасні тренди, приділивши більше уваги питанням дизайну, інтеграції компонентів та новим видам атак.

OWASP Top 10 (2021) окреслює найбільш небезпечні типи вразливостей, проте він не є вичерпним переліком усіх проблем безпеки. Організації повинні використовувати його як базис, доповнюючи іншими галузевими стандартами та специфічними аналізами загроз для своїх систем. З огляду на швидкі зміни у світі кіберзагроз, OWASP вже працює над черговим оновленням списку – очікується, що в ньому буде приділено увагу, зокрема, загрозам, пов'язаним із застосуванням ШІ та ланцюгом постачання програмного забезпечення. Це забезпечить актуальність моделі Top 10 у найближчі роки.

Поряд з OWASP Top 10 існує ряд інших класифікацій вразливостей і пов'язаних з ними стандартів безпеки. Кожен з них має свою специфіку і мету застосування. Розглянемо найвідоміші з них.

CWE Top 25 (SANS/CWE 25). CWE (Common Weakness Enumeration) – це каталог типових помилок програмування, підтримуваний MITRE. На його основі щорічно формується список Топ- 25 найбільш небезпечних програмних помилок, відомий як SANS/CWE Top 25. На відміну від OWASP, що фокусується на високорівневих ризиках вебдодатків, CWE Top 25 охоплює ширший спектр низькорівневих вразливостей у ПЗ загалом (включно з проблемами пам'яті в C/C++, помилками конкуренції, тощо). OWASP Top 10 і CWE Top 25 мають значну спільну область (багато вебвразливостей відповідають певним CWE-записам), однак CWE 25 надає більш детальну технічну класифікацію і орієнтована швидше на розробників системного ПЗ, архітекторів і дослідників безпеки [16]. OWASP Top 10 є зручним «чеклістом» для веброзробників, тоді як CWE Top 25 слугує довідником конкретних збоїв, які призводять до вразливостей, і оновлюється щороку під впливом нових загроз. Обидва стандарти доповнюють один одного: практики рекомендують враховувати OWASP Top 10 як базову мінімальну вимогу, а для глибокого аудиту безпеки коду звертатися до переліку CWE.

MITRE ATT&CK. Це не перелік вразливостей, а база знань про тактики і техніки атаки, підтримувана організацією MITRE. Фреймворк MITRE ATT&CK описує поетапно дії зловмисників (від розвідки до ексфільтрації даних) і включає сотні технік з унікальними ідентифікаторами. З точки зору веббезпеки, ATT&CK містить, зокрема, техніку T1190 Exploit Public-Facing Application – експлуатація вразливого публічного вебдодатку як метод початкового проникнення [17]. Таким чином, ATT&CK надає контекст: показує, як саме конкретні вразливості можуть бути використані в реальних атаках, і які подальші дії злочинця (привілейоване розширення, рух по мережі тощо). ATT&CK широко застосовується для моделювання загроз і побудови захисних сценаріїв (coverage) – наприклад, організація може перевіряти, чи здатна вона виявити і заблокувати всі техніки з ATT&CK, які актуальні для її середовища. У контексті вебдодатків ATT&CK корисний тим, що прив'язує OWASP/CWE-вразливості до сценаріїв реальних атак.

NIST SP 800-53. Документ NIST Special Publication 800-53 – це каталог із ~1000 контролів безпеки (засобів, процедур, політик), рекомендованих для інформаційних систем. На відміну від OWASP/CWE, NIST 800-53 не лише

про вразливості, а про всеохопний захист системи. Він містить контрольні заходи, згруповані у 20 категорій (контроль доступу, шифрування, безпека програмного забезпечення, реагування на інциденти тощо). Для вебдодатків у NIST 800-53 передбачено ряд конкретних вимог – наприклад, контролі SI-10 «Validation of Inputs» (валідація вводів) або SC-7

«Boundary Protection», що включає використання вебApplication Firewall. NIST 800-53 часто використовується урядовими установами США та підрядниками для побудови систем управління безпекою. Фактично, він задає framework для Secure SDLC: у ньому є розділи про безпечне проектування (SA-17), про аналіз коду (SA-11), про тестування безпеки (RA-5) тощо [18-19]. Тому для забезпечення відповідності строгим вимогам (наприклад, при обробці персональних даних у державному секторі) одних лише OWASP Top 10 може бути недостатньо – слід імплементувати ширший набір контролів на основі NIST 800-53.

PCI DSS. Стандарт безпеки індустрії платіжних карт (Payment Card Industry Data Security Standard) – обов'язковий для всіх компаній, що зберігають чи обробляють дані карток. PCI DSS містить 12 основних вимог, кілька з яких безпосередньо пов'язані з безпекою вебдодатків [20]. Зокрема, Requirement 6 – «Develop and maintain secure systems and applications» – зобов'язує впроваджувати процеси безпечної розробки, регулярно виправляти вразливості (патчі протягом 1 місяця після виходу) та проводити сканування/пен-тестування вебдодатків на наявність OWASP Top 10 ризиків. Також PCI DSS вимагає використовувати веб Application Firewall (WAF) або інші засоби захисту вебдодатків (Requirement 6.6). Таким чином, PCI DSS фактично інституціоналізує OWASP Top 10: невиправлення, скажімо, SQL-ін'єкції чи XSS у платіжному сервісі розглядатиметься як невідповідність стандарту з можливістю штрафних санкцій. Для бізнесу, який працює з платіжними даними, дотримання PCI DSS є обов'язковим, тому ця галузь переважно зріла у плані впровадження захисту вебдодатків.

ISO/IEC 27034. Міжнародний стандарт ISO/IEC 27034-1:2011 «Безпека додатків» – надає рекомендації щодо впровадження процесу Application Security – забезпечення безпеки на всіх етапах життєвого циклу застосунків. Він пропонує концепцію Організаційної нормативної бази (ONF), що включає політики, методики, контрольні списки безпечної розробки, і Програмної нормативної бази (ANF) для конкретного додатку [21]. ISO 27034 менш відомий і використовується рідше, ніж OWASP чи NIST, проте для великих організацій може бути корисним як формальний фреймворк інтеграції безпеки у процеси розробки згідно з міжнародно визнаними практиками. Він узгоджується із сімейством стандартів ISO 27000 (системи управління інформаційною безпекою) і доповнює їх прикладним рівнем.

OWASP Top 10 забезпечує практичну базу для розробників вебдодатків, акцентуючи увагу на конкретних типах вразливостей, яких потрібно уникати. CWE Top 25 дає деталізоване технічне бачення проблем у коді, особливо корисне для низькорівневого програмування. MITRE ATT&CK зміщує фокус з вразливостей на процес атаки, допомагаючи зрозуміти, як слабкі місця можуть бути використані в реальності, і вибудувати захист за принципом «kill chain». Стандарти на кшталт NIST, PCI, ISO пропонують системний підхід – інтеграцію контролів і процесів, щоб запобігати появі вразливостей та швидко реагувати на інциденти. Кожен з підходів не взаємовиключний, а швидше доповнює інші. На практиці організації часто використовують OWASP Top 10 як мінімум та орієнтир при розробці, CWE/CVE – для внутрішнього трасування та інвентаризації конкретних багів, а ATT&CK – для навчання персоналу реагувати на цільові атаки. Дотримання регуляторних стандартів (таких як PCI DSS або вимоги GDPR) є додатковим стимулом забезпечувати захищеність вебдодатків на належному рівні. У таблиці нижче представлено ключові характеристики цих моделей, їх особливості та сфери застосування.

2.1 Вплив вразливостей на безпеку вебдодатків

Вразливості вебдодатків несуть комплексні загрози для організацій та користувачів, охоплюючи економічні, репутаційні та правові аспекти. Компанії зазнають значних фінансових втрат як через прямі витрати на усунення наслідків атак, так і через втрату довіри клієнтів та партнерів. Нижче представлена таблиця з основними категоріями впливу веб

вразливостей та їх ключовими наслідками.

Таблиця 2

Порівняння класифікацій безпеки

Методологія	Основна мета	Ключові особливості
OWASP Top 10	Перелік найбільш критичних вразливостей вебдодатків для розробників	Фокус на практичних вразливостях веброзробки
CWE Top 25 (SANS/CWE 25)	Каталог найбільш небезпечних програмних помилок у різних типах ПЗ	і деталі вразливостей, корисні для розробників та дослідників
MITRE ATT&CK	Опис тактик та технік атак для аналізу загроз і тестування безпеки	ак, допомагає розуміти методи зловмисників
NIST SP 800-53	талог заходів для захисту інформаційних систем	Орієнтований на контрольні заходи безпеки
PCI DSS	Обов'язковий стандарт для компаній, що працюють з платіжними даними	Регламентує процеси безпеки для карткових даних
ISO/IEC 27034	овий стандарт для інтеграції безпеки у розробку ПЗ	у на всіх етапах життєвого циклу ПЗ

Таблиця 3

Результати порівняння

Категорія впливу	Фактор	Опис наслідків
Економічн і наслідки	Прямі витрати на усунення інциденту	Компанії витрачають мільйони доларів на розслідування атак, відновлення систем і впровадження додаткових заходів безпеки
Економічні наслідки	Втрачена інформація та простої	Втрати через викрадені платіжні дані та інтелектуальну власність, а також збитки через зупинку сервісу
Економічні наслідки	Зниження ринкової вартості	Середній спад акцій після великого витоку даних складає 5–7%, що може призвести до багатомільярдних збитків [22]
Економічні наслідки	Збільшення витрат на кіберстрахування	Компанії стикаються з підвищенням вартості кіберстрахування або взагалі втрачають можливість страхового покриття
Репутаційні ризики	Втрата довіри клієнтів	До 65% користувачів знижують рівень довіри після інциденту, а 31% повністю припиняють користування сервісом [22]
Репутаційні ризики	Підрив бренду та іміджу	Бренд асоціюється з ненадійністю, що може впливати на довгострокову конкурентоспроможність
Репутаційні ризики	Зниження лояльності клієнтів	Користувачі можуть рідше заходити на сайт, взаємодіяти з контентом або паралельно шукати альтернативи
Репутаційні ризики	Вплив на ділові партнерства	Бізнес-партнери можуть розірвати контракти або змінити умови співпраці, що позначається на доходах
Репутаційні ризики	Морально-психологічний клімат	ітники можуть втрачати мотивацію, що знижує продуктивність компанії
Правові наслідки	Адміністративні штрафи	Глобальні компанії можуть отримати штрафи до 4% світового обороту відповідно до регламентів, таких як GDPR [23]
Правові наслідки	Судові позови та компенсації	Користувачі можуть подавати колективні позови, що призводить до багатомільйонних виплат та судових витрат
Правові наслідки	Нагляд і обмеження діяльності	Регулятори можуть накладати обмеження на діяльність компанії, що впливає на операційну ефективність
Правові наслідки	Контракти і відповідальність перед клієнтами	Компанії можуть бути зобов’язані виплачувати компенсації клієнтам за порушення умов SLA
Правові наслідки	Ризик регуляторних перевірок	Посилений контроль з боку регуляторів після інциденту може спричинити додаткові витрати на аудит та відповідність стандартам

Економічні, репутаційні та правові наслідки вразливостей вебдодатків підкреслюють критичну важливість кібербезпеки у сучасному цифровому середовищі. Попередження атак та впровадження надійних систем захисту є не просто рекомендацією, а необхідністю для збереження фінансової стабільності, ринкової вартості компанії та її довгострокової конкурентоспроможності. Вкладення у превентивні заходи, такі як регулярне тестування безпеки, впровадження політик безпечної розробки та використання автоматизованих засобів виявлення вразливостей, є ключем до мінімізації ризиків та запобігання масштабним втратам.

3. Методи аналізу та виявлення вразливостей

Виявлення вразливостей на ранніх етапах – ключ до їх ефективного усунення. Існує декілька підходів до аналізу безпеки вебдодатків, що застосовуються протягом циклу розробки та експлуатації: статичний, динамічний та інтерактивний аналіз, а також їх комбінації. Розглянемо особливості кожного.

3.1 Статичне тестування безпеки

Статичний аналіз коду (SAST, Static Application Security Testing). Це метод перевірки безпеки, при якому аналізується вихідний код або двійкові файли програми без її запуску. Інструменти SAST сканують код на наявність відомих шаблонів вразливостей (правил), таких як необмежені SQL-запити, небезпечні виклики функцій тощо. Переваги SAST**:** виявлення вразливостей на ранніх стадіях SDLC – ще на етапі розробки, інтеграція в IDE або CI/CD, детальна вказівка місця в коді й причини проблеми [24]. SAST підходить для пошуку типових слабких місць (SQL-ін'єкції, XSS, слабка криптографія) і дає змогу розробникам виправити їх до релізу. Він

«бачить» внутрішню логіку додатку, недоступну іншим підходам. Недоліки**:** можливі помилково- позитивні спрацьовування, оскільки аналіз здійснюється без інформації про реальне оточення; складність аналізу динамічних конструкцій (рефлексії, генерованого коду); необхідний доступ до коду/збірки, тому менш придатний для перевірки сторонніх продуктів. Попри це, SAST є фундаментом концепції Shift Left – зміщення безпекового тестування вліво (тобто на більш ранні етапи розробки). Приклади SAST-інструментів**:** SonarQube, Checkmarx, Veracode (статичний модуль), Fortify тощо.

3.2 Динамічне тестування безпеки

Динамічний аналіз (DAST, Dynamic Application Security Testing). На відміну від SAST, динамічне тестування виконується на працюючому додатку, без доступу до його коду. Типовий DAST – це сканер вебвразливостей, який здійснює блек-бокс тестування: надсилає різноманітні вхідні дані до вебдодатку (через HTTP-запити, форми) і аналізує відповіді на ознаки вразливостей. Наприклад, сканер пробує вставляти ' OR '1'='1 у поля вводу і шукає в відповіді ознаки успішної SQL-ін'єкції. Переваги DAST: не потребує доступу до коду, дозволяє імітувати дії реального атакера ззовні, тому виявляє проблеми конфігурації, які SAST не побачить (наприклад, неправильні HTTP-заголовки безпеки). DAST можна застосовувати до будь-якого вебдодатку, розгорнутого в тестовому середовищі. Він добре знаходить вразливості, що проявляються у рантаймі: XSS, SQLi, неправильні доступи. Недоліки: DAST здійснюється пізно, на етапі готового продукту, тому виявлені помилки виправляти дорожче [25]. Крім того, динамічне сканування може пропустити вразливості бізнес логіки, не очевидні через інтерфейс, та часто займає багато часу на великих застосунках. DAST не знає структури коду, тому здатен ігнорувати внутрішні точки входу. Також є ризик хибнопозитивних результатів і навпаки, хибнонегативних, якщо сканер не обійшов якусь частину застосунку. Приклади DAST-інструментів: OWASP ZAP, Acunetix, Netsparker, Burp Suite (сканер) тощо. В ідеалі DAST виконується на стадії тестування перед релізом і періодично на продакшені (наприклад, щоквартально), як цього вимагає PCI DSS.

3.3 Інтерактивне тестування безпеки

Інтерактивне тестування (IAST, Interactive Application Security Testing). IAST – відносно новіший підхід, що намагається поєднати плюси SAST і DAST. Для цього в працюючий застосунок впроваджується спеціальний агент (сенсор), який моніторить виконання коду зсередини під час тестування. Іншими словами, IAST відстежує роботу вебдодатку на сервері в реальному часі (подібно до профайлера) і ловить потенційні вразливості, аналізуючи як сам код, так і його поведінку при певних вхідних даних [25]. Зазвичай IAST використовується у зв'язці з DAST або ручним тестуванням: поки зовнішній сканер або тестер взаємодіє з

додатком, IAST-агент

«всередині» відслідковує, наприклад, спроби створити SQL-запит із некоректними параметрами, виконати небезпечний системний виклик тощо. Переваги IAST: отримання дуже детальної інформації про вразливість (стек викликів, рядок коду, де сталася проблема), менше хибних спрацьовувань завдяки поєднанню контексту коду і фактичних даних. IAST працює у режимі

реального часу і може інтегруватися в середовище QA/CI. Це дозволяє, наприклад, автоматично перевіряти безпеку при запуску набору юніт-тестів: якщо якийсь тест вводить некоректні дані, IAST-агент всередині одразу вкаже на вразливість, якщо вона є. Недоліки: IAST-інструменти часто прив'язані до конкретних мов та платформ (обмежена підтримка технологій), їх впровадження може вимагати модифікації середовища (додавання агента на сервер). Крім того, вони перевіряють тільки ті ділянки коду, які були виконані в ході тестування – тобто покриття залежить від сценаріїв тесту. Якщо якась функціональність не була активно протестована, IAST її не проаналізує. Вартість таких систем також поки що досить висока, а ринок – вузький (комерційні рішення типу Contrast Security, Seeker (Synopsys), HCL AppScan). Тим не менш, комбінування інтерактивного підходу з традиційним статичним і динамічним тестуванням дає найкращі результати – покращується охоплення та точність виявлення вразливостей.

3.4 Використання AI/ML для виявлення вразливостей

З ростом складності програмного забезпечення і обсягу коду, що пишеться, виникла потреба в більш інтелектуальних підходах до аналізу безпеки. Традиційні інструменти (засновані на сигнатурах і жорстко закодованих правилах) іноді не встигають за креативністю розробників і хакерів. Тому в останнє десятиліття активно досліджуються й впроваджуються методи штучного інтелекту та машинного навчання (AI/ML) для автоматичного виявлення вразливостей.

Машинне навчання для аналізу вихідного коду. Науковці і компанії експериментують з моделями ML (насамперед глибинні нейронні мережі), які навчаються на великих масивах коду і даних про відомі вразливості. Ідея полягає в тому, щоб алгоритм самостійно виявляв приховані патерни небезпечного коду, без явних правил, заданих людиною. Дослідження показали, що найбільш ефективним є навчання з учителем (supervised ML), коли нейромережа тренується на наборі функцій/фрагментів коду, маркованих як вразливі чи безпечні [26]. Наприклад, модель може проаналізувати тисячі прикладів SQL-ін'єкцій у PHP і потім з певною ймовірністю передбачати, чи є вразливість у новому шматку коду. Такі підходи вже демонструють перспективні результати. Зокрема, дослідники з Microsoft і MIT в проєкті *Devign* розробили GNN-модель (графові нейронні мережі) для пошуку вразливостей у C-коді, яка досягла точності понад 60% виявлення раніше невідомих багів [27]. Інша модель на основі Transformer (підхід NLP, обробки природної мови) показала ще вищу ефективність при пошуку дефектів у рядках коду на GitHub. Комерційно ці технології лише починають з'являтися: наприклад, інструмент GitHub Copilot (GitHub Advanced Security) використовує ML не лише для підказки коду, а й для виявлення місць з потенційною вразливістю (Copilot може попередити розробника, якщо той пише небезпечний фрагмент). Очікується, що у майбутньому ML доповнить SAST-аналізатори, зменшивши кількість хибних спрацьовувань і виявляючи складні логічні вразливості, на які не написано явних правил.

AI для динамічного тестування і фаззінгу. Fuzzing (фазз-тестування) – техніка подачі великої кількості випадкових або мутованих даних у додаток з метою викликати виняткову ситуацію – теж еволюціонує завдяки ML. Зокрема, розробляються інтелектуальні фаззери, які використовують генетичні алгоритми або reinforcement learning для генерації саме таких тестових випадків, що мають вищий шанс знайти дефект. Алгоритм аналізує реакцію програми на попередні ввідні дані і

«навчається» генерувати нові ввідні, щоб просуватися глибше у код. Це особливо ефективно для пошуку багів пам'яті та логічних помилок. Наприклад, проект AFL++ має модулі, що інтегрують елементи ML, а компанія Google застосовує ML у своїй інфраструктурі ClusterFuzz для оптимізації фазз-тестів. Для вебдодатків AI може допомогти у розумінні допустимих структур HTTP-запитів і більш розумному комбінуванні параметрів, ніж простий перебір.

Аналіз бінарних даних і аномалій. Машинне навчання також використовується для виявлення аномальної активності, що може вказувати на експлуатацію вразливостей. Наприклад, системи WAF наступного покоління (Web Application Firewall з ML) навчаються на легітимному трафіку вебдодатку і блокують ті запити, які статистично відхиляються від норми. Якщо зловмисник спробує SQL-ін'єкцію з незвичним payload – модель класифікації може це визначити, навіть якщо немає явної сигнатури. Так само системи моніторингу (SIEM, IDS) впроваджують ML-модулі для виявлення патернів, що свідчать про атаку на вебсервіси (нетипова послідовність дій користувача, підозріле зростання помилок 500 тощо).

Обмеження і виклики AI/ML. Попри досягнуті успіхи, слід зазначити, що машинне навчання все ще не є універсальним вирішенням проблем. Його результати залежать від якості даних: потрібні великі вибірки коду з позначками вразливостей для навчання, а такі датасети складно зібрати (багато коду закриті, а помилки стають відомі не завжди). Крім того, моделі часто функціонують як «чорні скриньки», які надають результат без пояснень, що ускладнює розуміння причинно-наслідкових зв'язків і труднощів для розробника. Також ML-системи можуть пропускати вразливості, якщо ті не схожі на нічого, бачене раніше (проблема zero-day для ML). В літературі наголошується і на ризику зловживання ШІ нападниками: ті самі технології можуть використовуватися для пошуку нових вразливостей (наприклад, генерування експлойтів), тому гонка «озброєнь» триває.

Попри зазначені виклики, тенденція очевидна: штучний інтелект стає невід'ємною частиною інструментарію кібербезпеки. У контексті вебдодатків це означає більш автоматизоване і проактивне виявлення вразливостей. Вже зараз деякі провайдери пропонують SaaS-рішення, що сканують код репозиторію з використанням ML і виводять пріоритезований список підозрілих місць. В майбутньому можна очікувати появу повністю автоматизованих «багхантерів», які будуть переглядати кожен коміт програміста і, можливо, навіть автоматично створювати pull request з виправленням (є прототипи від Microsoft Research щодо автоматичного патчінгу вразливостей на основі ML). Для бізнесу це обіцяє зменшення витрат на ручний аудит і швидшу розробку безпечного коду, але поки що людина-експерт залишається важливою ланкою, що підтверджує і аналізує результати, отримані ШІ.

3.5 Практичне застосування

У практиці Secure SDLC великі організації часто використовують обидва типи аналізу – SAST і DAST – паралельно, аби компенсувати недоліки кожного. SAST застосовується на етапі розробки/кодування (наприклад, аналіз pull-запитів на GitHub), а DAST – перед розгортанням та під час експлуатації (наприклад, регулярне сканування на продакшн-серверах). IAST може інтегруватися на етапі тестування (QA). Такий багаторівневий підхід дозволяє спіймати більшу частину вразливостей до того, як продукт потрапить до кінцевих користувачів. Практика показує, що IAST здатний виявляти більше різновидів OWASP-ризиків, ніж суто DAST або SAST окремо, а комбінація методів дає максимально повну картину безпеки додатку.

Окрім автоматизованих методів тестування, важливу роль відіграють ручні аудитори та penetration testing. Досвідчені фахівці з безпеки можуть провести поглиблений аналіз, виявляючи логічні вразливості, недоліки бізнес-логіки, які автоматичні сканери не знайдуть. Зокрема, метод Threat Modeling на стадії дизайну допомагає ще до написання коду виявити потенційні слабкі місця і передбачити контрзаходи. Поєднання автоматизованих сканерів з експертизою людини дає найкращий результат у забезпеченні якості безпеки. Втім, у сучасних масштабах розробки автоматизація SAST/DAST стала невід'ємною частиною процесу – без неї підтримувати високий рівень безпеки на постійній основі вкрай важко.

Висновки

Проведений аналіз підтверджує, що безпека вебдодатків залишається одним з найактуальніших і найскладніших напрямків кібербезпеки. Кількість атак на вебресурси продовжує зростати разом із розширенням сфер застосування вебтехнологій – від мобільних додатків до хмарних сервісів і API для IoT. Відсутність належного захисту вебсистем може призвести до серйозних наслідків, таких як витоки персональних даних, фінансові втрати, підрив репутації компаній та правові санкції. Сучасні атаки стають все більш цілеспрямованими і складними,

використовуючи інноваційні методи (в тому числі зловживання можливостями штучного інтелекту) для обходу традиційних механізмів захисту безпеки.

У цих умовах організаціям потрібно впроваджувати комплексний підхід до безпеки вебдодатків. Такий підхід має включати як технічні засоби, так і процеси: починаючи від дотримання кращих практик безпечного програмування та регулярного аналізу коду (SAST/DAST) на етапах розробки, до інтеграції засобів безпеки в CI/CD конвеєри та проведення постійного моніторингу трафіку. Модель OWASP Top 10 залишається важливим орієнтиром для визначення найкритичніших вразливостей, і її останнє оновлення 2021 р. відображає сучасні пріоритети (контроль доступу, належне шифрування, безпечний дизайн). Ймовірно, у наступних редакціях OWASP Top 10 з'являться нові категорії, що відображають сучасні загрози, такі як атаки на ланцюг постачання та зловживання ШІ, захист хмарних сервісів тощо.

Забезпечення безпеки вебдодатків є безперервним процесом, що вимагає випереджувального мислення. Нові вразливості неминуче з'являтимуться, поки розробляється новий код. Однак застосування зазначених методів та практик значно підвищує стійкість вебсистем до більшості атак. Глибока багаторівнева оборона (defense-in-depth), побудована на основі правильної архітектури, безпечному коді, потужних засобах тестування і пильному моніторингу, дозволяє зводити ризик успішної атаки до мінімуму. Подальші дослідження і розвиток технологій кібербезпеки допоможуть впоратися з новими викликами – зокрема, зробити безпеку більш автоматизованою, проактивною і адаптивною. Це, своєю чергою, сприятиме розвитку надійного цифрового середовища, де вебдодатки можуть безпечно виконувати свої функції, не наражаючи на небезпеку дані користувачів та бізнес-процеси. У майбутньому забезпечення безпеки вебдодатків потребуватиме ще більшої уваги до інтеграції новітніх технологій та підходів, а також постійного адаптування до нових типів загроз. Це стане основою для створення більш стійких і безпечних інфраструктур, здатних витримати виклики еволюції технологій та кіберзагроз.

ЛІТЕРАТУРА

[1]. Verizon, «2024 Data Breach Investigations Report». [Електронний ресурс]. Режим доступу: https://www.verizon.com/business/resources/T807/reports/2024-dbir-data-breach-investigations-report.pdf.

[2]. Barracuda. «Threat Spotlight: Attackers Targeting Web Applications Right Now», 2024. [Електронний ресурс]. Режим доступу: https://blog.barracuda.com/2024/02/07/threat-spotlight- attackers-targeting-web-applications-right-now.

[3]. Statista, «Cybercrime Target Industries 2024». [Електронний ресурс]. Режим доступу: https://www.statista.com/statistics/221293/cyber-crime-target-industries/.

[4]. Cloud Security Alliance, «What we know about vulnerability exploitation in 2024 so far». [Електронний ресурс]. Режим доступу: https://cloudsecurityalliance.org/blog/2024/06/12/what-we- know-about-vulnerability-exploitation-in-2024-so-far.

[5]. C. C. Aladi, «Web Application Security: A Pragmatic Exposé», Electronics, Vol. 11, Issue 13, 2024. DOI: 10.1145/3644394. [Електронний ресурс]. Режим доступу: https://www.mdpi.com/2079- 9292/11/13/2049.

[6]. M. S. Chughtai et al., «Deep learning trends and future perspectives of web security and vulnerabilities», 2024. DOI: 10.3233/JHS-230037. [Електронний ресурс]. Режим доступу: https://ouci.dntb.gov.ua/en/works/4y6dzKgl/.

[7]. O. Ezenwoye, Y. Liu, W. Patten, «Classifying common security vulnerabilities by software type», Proceedings of the 32nd International Conference on Software Engineering and Knowledge Engineering (SEKE 2020), 2020. DOI: 10.18293/SEKE2020-047. [Електронний ресурс]. Режим доступу: https://ksiresearch.org/seke/seke20paper/paper047.pdf.

[8]. M. Althunayyan et al., «Evaluation of Black-Box Web Application Security Scanners in Detecting Injection Vulnerabilities», Electronics, Vol. 11, Issue 13, 2022. DOI: 10.3390/electronics11132049. [Електронний ресурс]. Режим доступу: https://www.mdpi.com/2079-9292/11/13/2049.

[9]. M. A. Yalçınkaya, E. U. Küçüksille, «Artificial Intelligence and Dynamic Analysis-Based Web Application Vulnerability Scanner», ISECURE Journal, Vol. 16, Issue 1, 2024. DOI: 10.22042/isecure.2023.367746.847. [Електронний ресурс]. Режим доступу: https://www.isecure-journal.com/article_183555_f52954f44ac33e6b456

862c7a8ad3ad5.pdf.

[10]. He Su et al., «Splendor: Static Detection of Stored XSS in Modern Web Applications», Proceedings of the 2023 International Symposium on Software Testing and Analysis (ISSTA), 2023. DOI: 10.1145/3597926.3598116.[Електронний ресурс]. Режим доступу: https://2023.issta.org/details/issta-2023-technical-papers/75/Splendor-Static-Detection-of-Stored-XSS-in-Modern-Web-Applications.

[11]. El País, «2024 bate récords históricos en ciberataques que, con ayuda de la IA, son cada vez más precisos», 2024. [Електронний ресурс]. Режим доступу: https://elpais.com/tecnologia/2024-12- 31/2024-bate-records-historicos-en-ciberataques-que-con-ayuda-de-la-ia-son-cada-vez-mas-precisos.html.

[12]. Federal Trade Commission (FTC), «Equifax Data Breach Settlement». [Електронний ресурс].

Режим доступу:

https://www.ftc.gov/ enforcement/refunds/equifax-data-breach-settlement.

[13]. Reuters, «Twitter hacked: 200 million user email addresses leaked, researcher says», 2023. [Електронний ресурс]. Режим доступу: https://www.reuters.com/technology/twitter-hacked-200- million-user-email-addresses-leaked-researcher-says-2023-01-05.

[14]. OWASP Foundation, «OWASP Top 10». [Електронний ресурс]. Режим доступу: https://www.blackduck.com/glossary/what-is-owasp-top-10.htm.

[15]. Outpost24, «OWASP Top 10 2021: What is new?». [Електронний ресурс]. Режим доступу: https://outpost24.com/blog/owasp-top-10-2021-what-is-new.

[16]. SecOp Solution, «OWASP Top 10 vs SANS 25». [Електронний ресурс]. Режим доступу:

https://www.secopsolution.com/blog/owasp-top-10-vs-sans-25.

[17]. MITRE, «ATT&CK Framework». [Електронний ресурс]. Режим доступу: https://attack.mitre.org/.

[18]. Kiuwan, «How NIST SP 800-53 Revision 5 Affects Application Security». [Електронний ресурс]. Режим доступу: https://www.kiuwan.com/blog/how-nist-sp-800-53-revision-5-affects-application- security.

[19]. Hyperproof, «NIST 800-53 Compliance Guide». [Електронний ресурс]. Режим доступу: https://hyperproof.io/nist-800-53/.

[20]. Stripe, «PCI Compliance Guide». [Електронний ресурс]. Режим доступу: https://stripe.com/ie/guides/pci-compliance.

[21]. ISO/IEC 27034, «Application Security Guidelines». [Електронний ресурс]. Режим доступу:

https://www.iso27001security.com/html/27034.html.

[22]. Rippleshot. «How Data Breaches Impact Brand Value», [Електронний ресурс]. Режим доступу: https://www.rippleshot.com/post/how-data-breaches-impact-brand-value.

[23]. GDPR EU. «What does it mean to be GDPR compliant in 2025?», [Електронний ресурс]. Режим доступу: https://www.gdpreu.org/what-does-it-mean-to-be-gdpr-compliant-2025/.

[24]. Morillo, C., «97 Things Every Information Security Professional Should Know: Collective Wisdom from the Experts», 2021.

[25]. Imperva, «SAST, IAST, DAST – Understanding the Differences». [Електронний ресурс]. Режим доступу: https://www.imperva.com/learn/application-security/sast-iast-dast/.

[26]. Nature, «Latest Cybersecurity Threats and Trends», 2024. [Електронний ресурс]. Режим доступу:

https://www.nature.com/articles/s41598-024-56871-z.

[27]. IEEE, «Web Application Vulnerability Detection Methods», 2024. [Електронний ресурс]. Режим доступу: https://ieeexplore.ieee.org/abstract/document/8614145.

SYSTEM APPROACH TO WEB APPLICATION SECURITY: ANALYSIS OF THREATS AND METHODS OF CYBER PROTECTION

The study focuses on analyzing common vulnerabilities in web applications, their impact on information system security, economic, reputational, and legal consequences, as well as methods for their detection and mitigation. A comprehensive review of the current state of web application security is conducted, including statistical data on current threats, analysis of attack trends, and an overview of the most notable incidents in recent years. Special attention is given to comparing different approaches to vulnerability classification, including OWASP Top 10, CWE Top 25, MITRE ATT&CK, NIST SP 800-53, and other standards, to

evaluate their effectiveness and practical applicability. The study examines web application security testing methods, including static (SAST), dynamic (DAST), and interactive application security testing (IAST), as well as the potential of artificial intelligence (AI) and machine learning (ML) for automated threat detection. The advantages and limitations of different cybersecurity methods are analyzed, along with practical aspects of their implementation in real-world scenarios. Additionally, a detailed analysis of the impact of vulnerabilities on organizations is presented, covering economic consequences (direct financial losses, response costs, market value decline), reputational risks (loss of user trust, brand damage), and legal repercussions (fines, lawsuits, regulatory compliance requirements). The results of the study contribute to forming a comprehensive approach to risk minimization, including the implementation of advanced security analysis methods, adherence to international security standards, and the application of modern cybersecurity tools. This ensures more efficient detection and mitigation of threats at the early stages of web application development and operation.

Keywords: cybersecurity, web application vulnerabilities, OWASP Top 10, static code analysis, dynamic analysis, machine learning, automated vulnerability detection, cyber threats.

Ільєнко Анна Вадимівна, к.т.н., доцент, завідувач кафедри кібербезпеки, Державного некомерційного підприємства «Державний університет «Київський авіаційний інститут», м.Київ, Україна.

Ilienko Anna, Ph.D., Associate Professor, Head of the Department of Cybersecurity, of the State non-commercial company state university «Kyiv aviation institute», Kyiv, Ukraine.

E-mail: anna.ilienko@npp.kai.edu.ua

Orcid ID: https://orcid.org/0000-0001-8565-1117

Спис Денис Сергійович, аспірант кафедри кібербезпеки, Державного некомерційного підприємства «Державний університет «Київський авіаційний інститут», м.Київ, Україна.

Spys Denys, PhD student at the Department of Cybersecurity, of the State non-commercial company state university «Kyiv aviation institute», Kyiv, Ukraine.

E-mail: 5771857@stud.kai.edu.ua

Orcid ID: 0009-0003-5316-8878

Галата Лілія Павлівна, доктор філософії, доцент кафедри кібербезпеки Державного університету «Київський авіаційний інститут»

Halata Liliia, PhD, associate professor of the Department of Cybersecurity, State University «Kyiv Aviation Institute»

E-mail: liliia.halata@npp.kai.edu.ua

Orcid ID: 0000-0002-7978-3954

Дубчак Олена Вікторівна, старший викладач кафедри кібербезпеки, Державного некомерційного підприємства «Державний університет «Київський авіаційний інститут», м. Київ, Україна.

Olena Dubchak, senior Lecturer at the Department of Cybersecurity of the State non-commercial company state university «Kyiv aviation institute», Kyiv, Ukraine.

E-mail: 3915922@npp.kai.edu.ua

Orcid ID: 0000-0001-9739-3960