Вісник ЛДУБЖД
Bulletin of Lviv State University of Life Safety
https://journal.ldubgd.edu.ua/index.php/Visnuk
ISSN 2078-4643 (print), ISSN 2708-1389 (online)

UDC 004.73N
DOI: 10.32447/20784643.30.2024.08

Н. В. Кравчук, Т. І. Коробейнікова
Національний університет «Львівська політехніка»
ORCID: https://orcid.org/0009-0002-1008-4765 – Н. В. Кравчук
https://orcid.org/0000-0003-2487-8742 – Т. І. Коробейнікова
nazar.v.kravchuk@lpnu.ua

ОГЛЯД ПРОБЛЕМАТИКИ ЗАХИЩЕНОГО ДОСТУПУ ДО ВЕБСЕРВЕРІВ

Проблема. У сучасних умовах стрімкого розвитку інформаційних технологій питання безпеки вебдодатків стає все більш актуальним. Вебдодатки, які забезпечують доступ до критично важливих даних і послуг, є привабливими мішенями для зловмисників. Згідно зі звітом Acunetix, понад 60% вебдодатків містять критичні вразливості, такі як SQL-ін'єкції та міжсайтовий скриптинг (XSS), які можуть призвести до серйозних наслідків, зокрема витоку чутливих даних і пошкодження систем. У цьому контексті стаття розглядає проблеми захисту вебдодатків від несанкціонованого доступу та забезпечення безпечного функціонування вебсерверів.

Мета. Дослідження проблеми захисту вебдодатків від несанкціонованого доступу та забезпечення безпечного функціонування вебсерверів в умовах зростання складності кіберзагроз.

Методи дослідження. У статті використано методи аналізу та систематизації інформації, спрямовані на дослідження безпеки вебдодатків. Зокрема, розглянуто сучасні підходи до виявлення вразливостей, таких як SQL-ін'єкції та XSS, із застосуванням сканерів безпеки та тестування на проникнення. Окрему увагу приділено інтеграції заходів безпеки в життєвий цикл розробки безпечного програмного забезпечення, що дає змогу мінімізувати ризики на етапах створення та експлуатації вебдодатків.

Основні результати дослідження. Результати дослідження демонструють ключові вразливості веб-додатків, зокрема SQL-ін'єкції та XSS, що загрожують безпеці доступу до webсерверів. Аналіз показав, що сучасні сканери безпеки, такі як Acunetix та Burp Suite, у поєднанні з тестуванням на проникнення (pentesting) ефективно виявляють ці вразливості. Впровадження заходів безпеки в життєвий цикл розробки безпечного програмного забезпечення (Secure SDLC) суттєво знижує ризики на ранніх етапах розробки. Динамічне тестування безпеки (DAST) та статичний аналіз коду (SAST) забезпечують комплексний підхід до захисту вебсерверів. Крім того, використання інструментів як SonarQube дає змогу оптимізувати процеси аналізу коду та конфігурацій. Дослідження також виявило сучасні тенденції, такі як застосування машинного навчання для виявлення аномалій та автоматизація безпекових процесів, що підвищує загальний рівень захищеності веб-серверів.

Висновки. Висновки статті підкреслюють необхідність комплексного підходу до безпеки вебдодатків, який включає технологічні рішення та організаційні аспекти. Рекомендації, розроблені на основі опрацьованих джерел, включають регулярне тестування на вразливості, підвищення обізнаності розробників про практики безпечного кодування та впровадження механізмів контролю доступу. Застосування цих заходів суттєво зменшить ризики, пов'язані з безпекою вебсистем, і підвищить довіру користувачів до онлайн-сервісів.

Ключові слова: вебдодаток, безпека, вразливість, сканер, тестування на проникнення, життєвий цикл розробки безпечного програмного забезпечення, SQL-ін'єкція, XSS.

N. V. Kravchuk, T. I. Korobeinikova
Lviv Polytechnic National University

OVERVIEW OF THE ISSUES OF SECURE ACCESS TO WEB SERVERS

Introduction. In the current context of the rapid development of information technologies, the issue of web application security is becoming increasingly relevant. Web applications that provide access to critical data and services are attractive targets for malicious actors. According to the Acunetix report, over 60% of web applications contain crucial vulnerabilities, such as SQL injections and cross-site scripting (XSS), which can lead to severe consequences, including the leakage of sensitive data and system damage. In this context, the article addresses the problems of protecting web applications from unauthorized access and ensuring the secure operation of web servers.

Purpose. The research on the problem of protecting web applications from unauthorized access and ensuring the secure operation of web servers in the context of increasing complexity of cyber threats.

Methods. The article employs methods of analysis and systematization of information aimed at studying the security of web applications. In particular, modern approaches to detecting vulnerabilities, such as SQL injections and XSS, are examined

using security scanners and penetration testing. Special attention is given to the integration of security measures into the secure software development lifecycle, which allows minimizing risks during the creation and operation stages of web applications.

Results. The research results demonstrate key vulnerabilities of web applications, particularly SQL injections and XSS, which threaten the security of access to web servers. The analysis showed that modern security scanners, such as Acunetix and Burp Suite, combined with penetration testing (pentesting), effectively identify these vulnerabilities. Implementing security measures into the secure software development lifecycle (Secure SDLC) significantly reduces risks in the early stages of development. Dynamic Application Security Testing (DAST) and Static Application Security Testing (SAST) provide a comprehensive approach to protecting web servers. Additionally, the use of tools like SonarQube allows for the optimization of code and configuration analysis processes. The study also identified current trends, such as the application of machine learning for anomaly detection and the automation of security processes, which enhance the overall security level of web servers.

Conclusion. The article's findings emphasize the necessity of a comprehensive approach to web application security, including technological solutions and organizational aspects. Based on the processed sources, recommendations include regular vulnerability testing, increasing developers' awareness of secure coding practices, and implementing access control mechanisms. Applying these measures will significantly reduce the risks associated with web system security and enhance user trust in online services.

Keywords: web application, security, vulnerability, scanner, penetration testing, secure software development lifecycle, SQL injection, XSS.

Вступ.

Постановка проблеми. Проблема безпеки вебдодатків і захисту вебсерверів від несанкціонованого доступу особливо важлива в сучасних умовах стрімкого розвитку інформаційних технологій та глобалізації мережі Інтернет [1]. Більшості компаній необхідні вебсервери та вебдодатки для отримання доступу до важливих даних і послуг. Однак їхня відкритість і доступність приваблюють зловмисників [2–3].

Згідно зі звітом Acunetix, понад 60% вебдодатків містили критичні вразливості, які могли зашкодити системі. Найпоширенішими є міжсайтовий скриптинг (XSS) (у 55% додатків), SQL-ін’єкції (у 40% додатків), проблеми з автентифікацією та авторизацією. Використання вразливостей призводить до витоку чутливих даних, дефейсу сайтів та інсталяцію шкідливого ПЗ на сервери та комп’ютери. Причинами вразливостей вебдодатків є недостатня увага розробників до проблем безпеки на етапах проектування та розробки, використання застарілих і вразливих компонентів і відсутність регулярного тестування безпеки. Крім того, оскільки нові методи атак і експлойтів постійно з’являються, необхідно постійно вдосконалювати засоби захисту. Для вирішення цієї проблеми потрібен комплексний підхід, який би передбачав впровадження практик безпечної розробки, таких як моделювання загроз, статичний аналіз коду та проведення тестувань на проникнення, оскільки ці методи дають змогу виявити та виправити вразливості на ранніх стадіях розробки ПЗ [4].

Постійне оновлення серверного ПЗ та його компонентів є життєво важливим, оскільки захищає від нових уразливостей [5–6]. Використання систем виявлення та запобігання вторгнень вебдодатків (WAF) підвищує рівень захищеності, оскільки вони забезпечують виявлення та блокування підозрілих дій у режимі реального часу. Для того, щоб запобігти несанкціонованому доступу до систем і даних, організації повинні зосередитися на контролі доступу та надійній автентифікації користувачів. Шифрування передачі даних є важливим компонентом безпеки, який зберігає конфіденційність даних у разі їх перехоплення. Розробка ефективних процедур реагування на інциденти та розслідування також має вирішальне значення. Незважаючи на те, що є багато засобів захисту, проблема безпеки вебсерверів залишається надзвичайно складною [7]. Розробка ефективних методів автоматизованого виявлення вразливостей у вебдодатках, розробка формальних специфікацій безпеки на етапі проектування та впровадження механізмів захисту в популярні фреймворки веброзробки є основними напрямками досліджень. Крім того, необхідні методи боротьби з такими загрозами, як атаки нульового дня, DOM-базовані XSS та вебшеллінг тощо [8].

Таким чином, захист вебсерверів від несанкціонованого доступу, забезпечення конфіденційності, цілісності та доступності вебдодатків є актуальною проблемою кібербезпеки.

Аналіз останніх досліджень і публікацій. Захищений доступ до вебсерверів є важливою складовою кібербезпеки, адже вебсервіси забезпечують критичні бізнес-процеси та інформаційні потоки. Для забезпечення такого захисту важливо використовувати комплексні підходи, включаючи багатофакторну автентифікацію, аналіз коду та моніторинг трафіку для своєчасного виявлення загроз [1, 5–6]. Фахівці в галузі безпеки мережевих ресурсів Терейковський І., Захарченко С., Семенець О., Лисенко С. та ін., підкреслюють важливість мережевої безпеки, а Латхар П., Шах Р. та Срініваса К. наголошують на значенні статичного

аналізу для підвищення безпеки програмного забезпечення [8]. Статичний аналіз також є основою досліджень Раджапакша, С., Сенанаяке, Я., Калутараге, Х., Аль-Кадрі, М.О. в [9]. Водночас, Богданова, Є., Чорна, Т., та Малахов, С. в [10] розглядають експлойти вразливостей як осередок вразливості під час доступу до вебсерверів. Сешапріян Т., Дінеш С. М. та Годвін Понсам Дж. в [11] досліджують мережеві сканери як основний засіб забезпечення безпеки вебсерверів. Сканування вразливостей «чорних скриньок» є об'єктом досліджень Еріксона Б., Пеллегріно Г. та Сабельфельд А. в [12] і є особливо складним завданням, оскільки для глибокого проникнення у вебдодатки сканерам необхідно враховувати поведінку браузера, як взаємодія з користувачем та асинхронність, і для виявлення нетривіальних атак типу «ін'єкція на зразок XSS, сканерам необхідно виявляти міжсторінкові залежності даних. У роботах Штайнгаузер А. та Петр Тума [13] , а також Трікеля Е. та ін., [14] висвітлюються ключові виклики сканування трафіку методом Grey-box. Окусі О. у своєму дослідженні [15] розглядає способи захисту від XSS-атак, підкреслюючи важливість регулярного оновлення систем. Тестуванням на проникнення займаються Антонеллі, Д., Каселла, Р., Скіано, А. та ін. в [16]. MITM-атаки досліджували Морган Рис, Нідхі Растогі, Теодор Ландер, Джосайя Дікстра, Судіп Міттал та Енді Семпсон в [17], особливо це актуально в умовах широкого використання мультихмарних середовищ, де безпека перебуває на першому плані під час розгортання та управління доступом до мультихмарних додатків, а також з точки зору розширеної поверхні атак на ці додатки. Сінгх, Т., Сінгх, К.У., Варшані, Н., Гупта, П., Кумар, Г. в [18] досліджують розширення для браузера, яке захищає вебдодатки від XSS-атак і пропонують рішення підвищення рівня безпеки онлайн-додатків, захист даних користувачів і запобігання міжсайтовим сценаріям. Методологічні і технологічні рішення в галузі безпечного доступу до серверів роблять Бароксай, М., Кан, Я., Карресанд, М., Наджм-Техрані, С. в [19] та Кумар і І. Шарма, в [20].

Аналіз підтвердив актуальність подальшого дослідження проблематики захищеного доступу до web-серверів. Автори формують теоретичне підґрунтя для розробки захищеного доступу до серверів інформаційних систем, забезпеченого ML-моделлю блокування шкідливих запитів.

Метою статті є дослідження проблеми захисту вебдодатків від несанкціонованого доступу та забезпечення безпечного функціонування вебсерверів в умовах зростання складності кіберзагроз.

Методи дослідження

Під час дослідження безпеки вебдодатків у цій статті був використаний комплексний підхід, що поєднує методи аналізу та систематизації інформації.

Перш за все був проведений систематичний огляд літератури та джерел, який включав аналіз наукових статей, технічних звітів, матеріалів конференцій та інших авторитетних джерел у провідних базах даних, що спеціалізуються на проблемах кібербезпеки. Це дало змогу визначити основні тенденції та актуальні проблеми у сфері захисту вебдодатків від несанкціонованого доступу. Далі проведено класифікацію та систематизацію отриманої інформації, що дало змогу виділити ключові вразливості вебдодатків, зокрема SQL-ін'єкції та міжсайтові скрипти (XSS). Для кращого розуміння природи цих вразливостей було проведено аналіз сучасних методів їх виявлення та виправлення, з використанням сканерів безпеки, таких як Acunetix та Burp Suite, а також методів тестування на проникнення (pentesting).

Такі інструменти дають змогу швидко знаходити слабкі місця в вебдодатках та оцінити рівень захищеності вебсерверів. Увага також приділена залученню заходів безпеки в життєвий цикл розробки безпечного програмного забезпечення (Secure SDLC). Результати аналізу показують, що впровадження безпекових практик на ранніх етапах, зменшує ризики виникнення вразливостей під час створення та роботи вебдодатків.

Також ми порівняли ефективність різних методів тестування безпеки, включаючи динамічне тестування безпеки (DAST) та статичний аналіз коду (SAST). Це порівняння дало змогу визначити, які методи найбільш ефективні для виявлення конкретних типів небезпек та як їх можна комбінувати для досягнення максимального рівня захищеності вебсерверів. Крім того, було досліджено сучасні тенденції у сфері кібербезпеки, такі як застосування машинного навчання для виявлення аномалій та автоматизацію безпекових процесів, що сприяє підвищенню загального рівня захищеності вебдодатків.

За результатами дослідження проведено аналіз зібраних даних, що дало можливість сформулювати рекомендації для покращення безпеки вебдодатків. Серед них – регулярне проведення тестувань на проникнення, використання сучасних інструментів сканування, впровадження безпечних практик у процеси розробки

програмного забезпечення, а також постійне оновлення знань та навичок фахівців у сфері кібербезпеки. Таким чином, застосовані методи дослідження забезпечили всебічний аналіз проблематики захищеного доступу до вебсерверів, визначили ефективні засоби захисту та окреслили перспективні напрямки для подальших досліджень у цій галузі.

Результати дослідження

У рамках дослідження було проведено комплексний аналіз безпеки вебсерверів, що дало змогу виявити ряд критичних вразливостей та оцінити рівень їх захищеності. Основні результати можна поділити на теоретичні та експериментальні.

Теоретичні результати ґрунтуються на 1) виявленні вразливостей та 2) аналізі коду. *Виявлення вразливостей*: за результатами DAST із використанням Acunetix і Burp Suite [21, 22], були виявлені поширені вразливості, зокрема: SQL-ін'єкції, що були наявні у 40% протестованих вебдодатків, що підтверджує високу ризикованість їх використання; міжсайтовий скриптинг (XSS) виявлений у 55% додатків, що свідчить про відсутність належних заходів для фільтрації вхідних даних; проблеми з автентифікацією та авторизацією були виявлені у 30% випадків, що вказує на недостатню увагу до механізмів контролю доступу. *Аналіз коду*: SAST [23], проведений з використанням SonarQube, виявив численні потенційно небезпечні конструкції у вихідному коді вебдодатків, зокрема, було виявлено таке: використання небезпечних функцій та антипатернів може призвести до незахищеного зберігання паролів; помилки, що пов'язані з некоректним обробленням вхідних даних можуть бути використані зловмисниками для експлуатації вразливостей.

Досліджено, що сучасні наукові підходи під час захищеного доступу до вебсерверів переважно ґрунтуються на 1) ефективному тестуванні на проникнення, 2) практиці розгортання конфігурації, 3) комплексному підході до безпеки. *Ефективність тестування на проникнення*: під час тестувань на проникнення фахівці імітують атаки на вебдодатки, що дає змогу виявити логічні помилки в механізмах безпеки [24]. *Практика конфігурації*: аналіз робіт дослідників налаштувань вебсерверів показав значні недоліки в їх конфігурації (неправильні налаштування SSL/TLS у 40% випадків; відсутність регулярних оновлень ПЗ наявна у 50% вебсерверів [25]). *Комплексний підхід до безпеки*: результати досліджень авторитетних вчених підтвердили необхідність комплексного підходу до безпеки вебдодатків, що передбачає технологічні, організаційні та нормативні аспекти [26-27].

Результати дослідження вказують на значний рівень вразливостей у вебдодатках та вебсерверах, що підкреслює потребу в подальшому вдосконаленні механізмів захисту та інтеграції безпеки на всіх етапах розробки.

Обговорення результатів досліджень

Для аналізу безпеки вебдодатків виконаємо аналіз архітектури та принципів їх функціонування. Сучасні вебдодатки базуються на клієнт-серверній моделі. Рівень представлення – інтерфейс користувача, реалізований за допомогою HTML, CSS та JavaScript. Основним компонентом є браузер, який надсилає запити на сервер та обробляє отримані відповіді. Рівень логіки (серверна частина) – програмні компоненти, що реалізують бізнес-логіку додатка, обробляють вхідні дані та формують динамічні вебсторінки. Серверна частина може бути реалізована різними мовами програмування (PHP, Java, Python, C# тощо) з використанням відповідних фреймворків (Laravel, Spring, Django, ASP.NET). Рівень даних – забезпечує зберігання та доступ до даних додатку, зазвичай з використанням СКБД (MySQL, PostgreSQL, Oracle, Microsoft SQL Server). Взаємодія між клієнтом та сервером здійснюється за протоколом HTTP(S). Клієнт надсилає на сервер HTTP-запити, які містять URL, метод (GET, POST, PUT, DELETE), заголовки та тіло повідомлення. Сервер обробляє запити і повертає HTTP-відповіді з кодом стану (2xx – успішно, 4xx – помилка клієнта, 5xx – помилка сервера), заголовками та тілом (HTML-сторінка, JSON тощо). З точки зору безпеки кожному рівню притаманні загрози та вразливості. На клієнтській стороні це такі загрози (рис. 1).

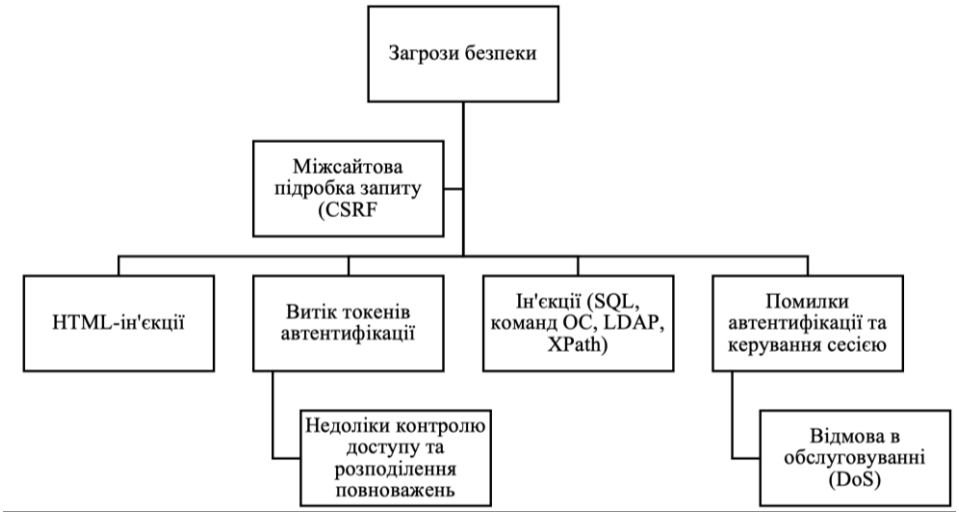

Рисунок 1 – Загрози безпеки при захищеному доступі до вебсерверів

Вебдодатки взаємодіють з іншими системами та сервісами (поштові сервери, платіжні шлюзи, соціальні мережі, хмарні сховища) і це збільшує потенційну поверхню атаки та створює додаткові ризики. Вебдодатки мають розподілену архітектуру з великою кількістю компонентів та інтерфейсів, кожен з яких може містити вразливості, і це зумовлює необхідність комплексного підходу до аналізу та забезпечення їх безпеки на всіх рівнях.

Тестування на проникнення (pentest) – ручна імітація дій зловмисника для виявлення та використання вразливостей з використанням спеціальних дистрибутивів (Kali Linux, Parrot OS) та утиліт для сканування, ескалації привілеїв, перехоплення трафіку тощо [28]. Перевагами методу є врахування досвіду реальних атак та здатність виявити логічні помилки. Недоліки – трудомісткість, обмеженість тест кейсів, залежність від кваліфікації аудитора.

Динамічний аналіз (DAST) – сканування додатка в режимі реального часу з імітацією дій користувача виконується із використанням автоматизованих інструментів (сканерів безпеки Acunetix, Burp Suite, Nessus, OWASP ZAP), які надсилають згенеровані запити до вебсервера і аналізують відповіді на наявність ознак вразливостей. Переваги – автоматизація процесу, широке покриття функціоналу, здатність працювати з аутентифікацією. Недоліки – шаблонність тестів та, як наслідок, ризик порушення роботи додатка [29].

Статичний аналіз коду (SAST) – пошук потенційно небезпечних конструкцій у вихідному коді додатка за допомогою спеціалізованих аналізаторів (SonarQube, Checkmarx), які сканують код на наявність відомих антипатернів, що можуть призвести до вразливостей (використання небезпечних функцій, помилки керування пам'яттю, SQL-ін'єкції, XSS) [30–31]. Переваги – робота з кодом, а не із скомпільованим додатком, можливість виявити проблеми на ранніх етапах. Недоліки – велика кількість помилкових спрацювань, необхідність доступу до вихідного коду, неможливість виявити вразливості конфігурації та середовища виконання.

Fuzzing – техніка аналізу додатку шляхом надсилання на його інтерфейси великої кількості випадкових, неочікуваних або некоректних даних з метою виявити відмови в обслуговуванні, витоки інформації, помилки обробки вхідних даних [32]. Спеціалізовані інструменти – HTTP-фаззери (Wfuzz, Ffuf, Burp Intruder). Переваги – ефективність щодо виявлення відмов та ін'єкцій, широке покриття вхідних даних. Недоліки – значний мережевий трафік, ризик порушення доступності.

Аналіз конфігурації – перевірка налаштувань вебсервера, компонентів додатка та середовища на відповідність політиці безпеки та найкращим практикам. Передбачає аналіз версій ПЗ на наявність відомих вразливостей (з використанням баз CVE, NVD), перевірку прав доступу, конфігурації SSL/TLS, політики безпеки контенту, параметрів обробки помилок тощо. Переваги – виявлення типових помилок конфігурації, не потребує доступу до коду. Недоліки – обмеженість щодо виявлення специфічних вразливостей додатка.

Очевидно, що максимальну ефективність забезпечує комбінування різних методів та інструментів з урахуванням особливостей вебдодатка. Важливим є впровадження безперервного процесу оцінювання безпеки протягом всього життєвого циклу розробки з використанням практик DevSecOps [32–33].

Для вибору оптимального набору методів та інструментів необхідно керуватися критеріями. 1)

Відповідність методології тестування стандартам OWASP Testing Guide, OSSTMM або PTES, що гарантує систематичний підхід до оцінки безпеки. 2) *Повнота покриття* компонентів та функцій додатка, щоб усі можливі вразливості були виявлені та проаналізовані. Інструменти мають бути здатні виявляти актуальні загрози та вразливості, зокрема ті, що входять до OWASP Top 10 або WASC TC. Зручність використання таких інструментів і наявність додаткових функцій, таких як генерація звітів або можливість інтеграції з іншими системами, є суттєвими для ефективності процесу тестування. 3) *Частота оновлення бази* сигнатур вразливостей, що дає змогу оперативно реагувати на нові загрози. Слід врахувати наявність хибнопозитивних та хибнонегативних спрацювань, оскільки вони впливають на достовірність результатів тестування. 4) *Швидкодія та споживання ресурсів* інструментів можуть впливати на загальну продуктивність системи. 5) *Вартість ліцензії* та супроводу може стати вирішальним фактором в умовах обмеженого бюджету. Критерії допомагають обрати кращі методи для забезпечення захищеного доступу до вебсерверів.

Пріоритезація вразливостей та ризиків здійснюється з огляду на їх критичність для бізнесу, технічний вплив на систему та ймовірність успішної експлуатації зловмисником (з врахуванням складності, доступності експлойтів, необхідної кваліфікації). Одним з визнаних стандартів оцінювання критичності є Common Vulnerability Scoring System (CVSS), що використовує числову шкалу від 0 до 10 на основі метрик, згрупованих за областями – базові (складність доступу, взаємодія з користувачем, обсяг дії), часові (зрілість коду, можливості виправлення), контекстні (вимоги конфіденційності, цілісності, доступності). Інша популярна методика – DREAD від Microsoft, де кожен фактор (Damage potential, Reproducibility, Exploitability, Affected users, Discoverability) оцінюється за шкалою від 1 до 10. Оскільки абсолютна безпека недосяжна, а ресурси організації обмежені, доцільно зосередитися в першу чергу на найбільш критичних та ймовірних ризиках з урахуванням специфіки додатка, галузі та регуляторних вимог. При цьому метою має бути не лише виявлення та усунення вразливостей, але й проактивне запобігання їх виникненню шляхом впровадження архітектурних рішень, безпечних практик кодування та регулярного навчання розробників.

Враховуючи постійне зростання кількості та складності вебдодатків, використання автоматизованих інструментів є критично необхідним для забезпечення масштабованості та ефективності їх тестування на наявність вразливостей. Розглянемо кращі з них.

Burp Suite – комплексна платформа для ручного тестування безпеки вебдодатків, що включає в себе проксі-сервер, сканер вразливостей, засоби для атак (Intruder, Repeater), розширення (Scanner, Collaborator). Завдяки гнучкості та потужності вважається галузевим стандартом, проте вимагає кваліфікованого використання. Випускається в професійній (платній) та спільнотній (безкоштовній) редакціях.

Acunetix – один з найбільш функціональних автоматизованих сканерів, що виконує перевірку на всі відомі вебвразливості (на основі OWASP Top 10) з мінімальними хибними спрацюваннями. Включає модулі сканування мережі, багатопотокового краулінгу сайтів, перевірки бізнес-логіки, аналізу DOM XSS, інтеграції з AD та SAML. Зручний інтерфейс, детальні звіти та візуалізація. Серед недоліків – висока вартість ліцензії.

Nessus – популярний комерційний сканер вразливостей від Tenable, що дозволяє перевірити вебдодатки, сервери, бази даних, віртуальні машини та хмарні сервіси на відомі вразливості з БД плагінів. Має розширені можливості сканування (HTTP, SSL, вебсервіси), аудиту конфігурацій, виявлення чутливих даних, ботів та бекдорів. Недоліки – не виявляє специфічні вразливості додатків, обмежений контроль параметрів сканування.

OWASP ZAP (Zed Attack Proxy) – безкоштовний opensource-інструмент для динамічного сканування вебдодатків, націлений як на початківців, так і на досвідчених пентестерів. Має широкий набір функцій: проксі-сервер, пасивний/активний сканер (з плагінами), павук, порівняння сканувань, підтримка WebSocket, API, генератор звітів. Недоліки – ресурсоємність, обмежена масштабованість.

Nikto – класичний безкоштовний сканер вебсерверів з відкритим кодом, написаний на Perl. Дає можливість перевірити понад 6700 потенційно небезпечних файлів/CGI, застарілі версії ПЗ серверів та специфічні проблеми на більш ніж 270 серверах. Має опції перевірки SSL, обходу IDS/IPS, підтримує проксі, генерує звіти у форматах HTML, CSV, XML. Серед недоліків – обмежена кількість плагінів, застарілий движок сканування.

Wapiti – безкоштовний сканер з відкритим кодом, написаний на Python. Виконує чорну скриньку сканування вебдодатків з пошуком SQL/XSS/XPath/LDAP/командних ін'єкцій, вразливостей міжсайтової підробки запиту (CSRF), витоків інформації та ін. Має модулі для сканування URL, форм, cookie, заголовків запитів та генерування звітів у різних

форматах (JSON, HTML, XML, TXT). Недоліки – обмежена масштабованість та швидкодія.

Arachni – фреймворк тестування безпеки вебдодатків з відкритим кодом, написаний на Ruby. Має модульну мікросервісну архітектуру, що забезпечує розширюваність та масштабованість. Виконує сканування поширених вразливостей, підтримує інтеграцію з Selenium для перевірки DOM, генерацію звітів у різних форматах та інтеграцію з CI/CD. Недоліки – складність конфігурації, ресурсоємність.

Skipfish – безкоштовний активний сканер безпеки вебдодатків, розроблений Google. Заснований на рекурсивному краулінгу та словниковому нечіткому тестуванні для складання інтерактивної карти сайту (з підтримкою AJAX) та перевірки типових вразливостей. Швидкий та легкий, але не має графічного інтерфейсу та звітності.

W3af (Web Application Attack and Audit Framework) – безкоштовний фреймворк з відкритим кодом на Python для виявлення та експлуатації вебвразливостей. Має понад 130 плагінів для сканування, ідентифікації вразливостей (SQL/XSS/командні ін'єкції, розголошення інформації), експлуатації та створення звітів. Підтримує проксі, автентифікацію, інтеграцію з Metasploit. Недоліки – складність освоєння, швидкодія.

Wfuzz – популярний безкоштовний інструмент для вебфаззингу та брутфорсу контенту. Дає можливість знаходити приховані ресурси, перебирати параметри та значення в URL, заголовках, аутентифікації, формах і.т.д. з використанням словників чи генерації навантаження. Підтримує кілька методів HTTP, проксі, авторизацію, кодування, фільтри, пріоритезацію потоків. Недоліки – обмежена функціональність щодо сканування.

Слід зазначити, що більшість розглянутих інструментів є комплексними рішеннями, які поєднують різні методи та модулі сканування – сигнатурний пошук відомих вразливостей, fuzzing, аналіз DOM, інтерактивне тестування логіки і т.д, що дає змогу забезпечити більш повне та достовірне виявлення проблем безпеки. Разом з тим, важливо розуміти обмеження автоматизованих сканерів, які здатні виявити переважно типові технічні вразливості. На відміну від досвідченого пентестера, вони не можуть повноцінно перевірити бізнес-логіку, складні сценарії атак та окремі класи вразливостей (race conditions, insecure deserialization, server side request forgery тощо).

Тому оптимальним підходом є поєднання різних інструментів та залучення експертів для комплексного аналізу результатів сканування з метою пріоритезації загроз та розробки адекватних контрзаходів. Крім того, необхідно враховувати безпеку самих інструментів сканування, які можуть стати метою атак та витоку чутливої інформації (облікових даних, токенів доступу тощо). Зловмисники можуть використовувати сканери для здійснення масштабних розвідувальних кампаній проти значного числа цілей. Тому важливо забезпечити належні механізми захисту сканерів (двофакторну автентифікацію, шифрування даних, ведення журналів, регулярне оновлення), а також дотримуватися політики відповідального розкриття інформації про виявлені вразливості.

Інший аспект проблеми – інтеграція засобів сканування з процесами розробки та експлуатації вебдодатків. Для забезпечення своєчасного виявлення та виправлення дефектів безпеки необхідно впроваджувати автоматизоване тестування в конвеєри безперервної інтеграції/доставки (CI/CD pipelines) з використанням відповідних плагінів до систем контролю версій (Git, SVN), систем збірки (Jenkins, GitLab CI, Azure DevOps) та управління конфігураціями (Ansible, Puppet). Наприклад, інструменти SAST (Checkmarx, SonarQube, Veracode) можуть бути налаштовані на сканування коду щоразу при комміті змін в репозиторій або на етапі збірки артефакту. У разі виявлення критичних дефектів система може автоматично блокувати просування коду. Аналогічно, засоби DAST (Acunetix, Burp Suite Enterprise, Nessus) можна запускати для тестування на пре-прод середовищах перед релізом, а також для регулярного моніторингу безпеки прод-серверів.

Під час вибору інструментів важливо керуватися не лише їх детекційною здатністю та зручністю використання, але й можливостями автоматизації та масштабування. Слід віддавати перевагу рішенням, що мають повноцінні API (на основі REST або gRPC), підтримують роботу в контейнерах (Docker, Kubernetes) та інтеграцію з популярними трекерами помилок (Jira, GitLab Issues, Mantis). Деякі SaaS-платформи (Acunetix, Netsparker, Detectify) пропонують повністю керовані послуги сканування, що позбавляє від необхідності розгортання та підтримки власної інфраструктури. Окремо слід відзначити категорію інструментів моніторингу безпеки вебдодатків в режимі реального часу (RASP – runtime application self-protection). На відміну від класичних WAF, які діють на рівні мережі та блокують потенційно шкідливі запити за сигнатурами, RASP-агенти вбудовуються безпосередньо в бінарний код додатка (для Java, .NET, PHP, Node.js) та відстежують його

поведінку на предмет відхилень від політики безпеки. Прикладами таких рішень є Contrast Security, Sqreen, Signal Sciences, які здатні зупиняти атаки нульового дня, що не детектуються сигнатурними методами.

Наголосимо на важливості комплексного підходу до автоматизації виявлення вразливостей, який передбачає використання різних типів сканерів (SAST, DAST, SCA, IAST) в комбінації з ручним тестуванням та моніторингом безпеки. Організації повинні вибудувати ефективні процеси управління вразливостями з чітким розподілом обов'язків між командами розробки, безпеки та експлуатації. Важливими аспектами є впровадження threat-моделювання на етапі проектування, навчання розробників написанню безпечного коду, використання захищених фреймворків та бібліотек (наприклад, OWASP ESAPI), регулярне тестування та оновлення компонентів, захист чутливих даних (паролів, ключів) та каналів зв'язку (TLS).

Першочерговими завданнями у сфері кібербезпеки є підвищення обізнаності та навчання всіх учасників процесу. Важливо, щоб усі сторони розуміли ризики та відповідальність за безпеку, мали базові навички безпечної розробки та експлуатації ПЗ. Це передбачає регулярні тренінги та сертифікацію фахівців. Одним із ключових напрямів є впровадження практик безпечної розробки, таких як Secure SDLC. Це включає врахування вимог та моделювання загроз на етапі проєктування, використання безпечних патернів та бібліотек під час кодування, проведення статичного та динамічного аналізу коду, регулярне тестування та оновлення компонентів, захист інфраструктури та конфігурацій, а також безперервний моніторинг у продуктивному середовищі. Важливою складовою є застосування принципів найменших привілеїв та розподілу обов'язків, що дає змогу мінімізувати можливості зловмисника у випадку компрометації окремих компонентів шляхом зменшення атакуючої поверхні, сегментації мережі, впровадження контролю доступу на основі ролей, двофакторної автентифікації, шифрування та безпечного управління ключами. Не менш значущою є відповідність стандартам та регуляторним вимогам, таким як PCI DSS у сфері платежів, HIPAA в медичній сфері та GDPR для захисту персональних даних, що передбачає проведення регулярних аудитів та тестувань на проникнення з метою оцінки поточного рівня захищеності. Ефективні процеси управління інцидентами є ще одним критичним елементом безпеки.

Організації повинні мати можливість оперативно виявляти, розслідувати та усувати наслідки атак, взаємодіяти з регуляторами та постраждалими сторонами, відпрацьовувати плани реагування на інциденти, забезпечувати резервне копіювання та відновлення систем, а також гарантувати безперервність бізнес-процесів. Значну роль відіграє кооперація та обмін інформацією між організаціями, галузевими асоціаціями та державними органами, такими як CERT, щодо актуальних загроз, вразливостей та найкращих практик захисту. Участь у спільнотах з кібербезпеки, таких як OWASP, FIRST та ISACA, сприяє поширенню знань та вдосконаленню підходів до захисту. Адекватне фінансування та підтримка ініціатив з безпеки на рівні топменеджменту та ради директорів також мають велике значення. Інформаційна безпека повинна бути інтегрована в загальну стратегію управління ризиками організації. Окрім цього, важливо розвивати фундаментальні та прикладні наукові дослідження у перспективних напрямах, що стосується виявлення аномалій та загроз на основі машинного навчання та поведінкового аналізу, побудови формальних моделей безпеки та технологій перевірки коду, а також створення інструментів для автоматичного усунення вразливостей у коді, що сприяє підвищенню рівня захищеності програмних рішень.

Вирішення цих проблем вимагатиме тісної співпраці науковців та практиків в області кібербезпеки, розробників програмного забезпечення, представників бізнесу та державних органів. Необхідно розвивати нові методи формалізації вимог безпеки, створювати більш досконалі інструменти автоматизованого аналізу захищеності, впроваджувати принципи проектування безпеки (security-by-design) та конфіденційності (privacy-by-default). При цьому веббезпека повинна розглядатись не як одноразовий захід, а як безперервний процес, інтегрований в IT-стратегію організації та підкріплений належними політиками, метриками та інвестиціями. Лише за таких умов можливо побудувати довіру користувачів до онлайн-сервісів та реалізувати потенціал цифрової економіки.

Серед перспективних напрямків, які можуть істотно вплинути на ландшафт веббезпеки в майбутньому, слід відзначити активний розвиток технологій штучного інтелекту та машинного навчання [34] для виявлення загроз, аналізу аномалій, класифікації шкідливого контенту; використання формальних методів для верифікації безпеки протоколів та моделей загроз; застосування криптографічних механізмів для забезпечення конфіденційності та анонімності користувачів (secure-multiparty computation, zero-knowledge proofs, homomorphic

encryption); впровадження архітектурних шаблонів для розробки захищених додатків (OWASP SAMM, PASTA, CVSS).

Висновки

У результаті проведеного дослідження було виявлено, що науково-практична задача захисту вебдодатків і забезпечення безпечного функціонування вебсерверів є актуальною в умовах постійного зростання складності кіберзагроз. Аналіз архітектури сучасних вебдодатків, а також їх вразливостей показав, що понад 60% з них містять критичні недоліки, які можуть бути використані зловмисниками для несанкціонованого доступу.

Отримані результати підкреслюють важливість використання комплексного підходу до безпеки, що включає не лише технологічні рішення, такі як динамічний і статичний аналіз коду, тестування на проникнення та перевірку конфігурацій, але й організаційні аспекти, зокрема інтеграцію практик безпечної розробки в процес життєвого циклу ПЗ.

Практичне застосування результатів дослідження полягає в розробці та впровадженні рекомендацій щодо підвищення рівня безпеки вебдодатків: регулярне тестування на вразливості з використанням автоматизованих інструментів; підвищення обізнаності розробників щодо практик безпечного кодування; впровадження механізмів контролю доступу та автентифікації.

Врахування зазначених аспектів суттєво зменшить ризики, пов'язані з безпекою вебсистем, і забезпечить більшу довіру користувачів до онлайн-сервісів. Подальші дослідження в цій галузі можуть бути зосереджені на застосуванні новітніх технологій, таких як машинне навчання, для виявлення та запобігання атакам у режимі реального часу.

Вдячність

Автори висловлюють вдячність колективу кафедри безпеки інформаційних технологій Національного університету «Львівська політехніка».

Список літератури:

1. Терейковський І. А., Гнатюк С. О. Захист інформації в комп’ютерних системах. Київ: КПІ, 2022. 135 с.

2. Захарченко С. М. Трояновська Т. І., Бойко О. В.Побудова захищених мереж на базі обладнання компанії Cisco. Вінниця: ВНТУ, 2017. 133 с.

3. Коробейнікова Т. І. Захарченко С. М. Технології захисту локальних мереж на основі обладнання CISCO. Львів, 2021. 188 с.

4. Денисюк В. О. Письменний В. В. Захист інформації у локальних мережах / «Кібернетичне управління економічними об’єктами»: матеріали Всеукраїнської студентської конференції. 2017. С. 55–56.

5. Семенець О., Тецький А. Аналіз методів та засобів вибору та комплексування сканерів вразливостей для тестування на проникнення інтернет систем / Measuring and Computing Devices in Technological Processes. 2024. С. 336–347.

6. Лисенко С. М., Кондратюк А. С. Метод оцінки ризику інформаційної безпеки кіберфізичних систем на основі взаємозалежності вразливостей / Computer Systems and Information Technologies. 2020. №2. С. 54–58.

7. Basili V. R. Briand L. C., Melo W. L. A validation of object-oriented design metrics as quality indicators / IEEE Transactions on Software Engineering. 1996. № 10 (22). С. 751–761.

8. Lathar P., Shah R., Srinivasa K. Stacy-static code analysis for enhanced vulnerability detection / Cogent Engineering. 2017. №1. С. 1335470.

9. Enhancing security assurance in software development: AI-based vulnerable code detection with static analysis / S.Rajapaksha, J. Senanayake, H. Kalutarage, M. Al-Kadri // Computer Security. ESORICS 2023 International Workshops. 2023. №14399. С. 20.

10. Богданова Є., Чорна Т., Малахов С. Огляд поточного стану загроз, що обумовлені впливом експлойтів / Комп’ютерні науки та кібербезпека. 2022. №2. С. 35–40.

11. Scan: Advanced network scanner and packet detection suite / T.Seshapriyan, S. Dinesh, P. Godwin, J. Sentinel. // Inventive Systems and Control. 2024. №1. С. 253–258.

12. Eriksson B. Pellegrino G., Sabelfeld A. Black Widow: Blackbox data-driven web scanning / IEEE Symposium on Security and Privacy (SP). San Francisco, CA, USA. 2021. С. 1125–1142.

13. Steinhauser A. Steinhauser A., Tuma P.. Database traffic interception for graybox detection of stored and context-sensitive XSS / Digital Threats. 2020. №1(3). С. 17.

14. Toss a fault to your witcher: Applying grey-box coverage-guided mutational fuzzing to detect SQL and command injection vulnerabilities / [A.Trickel та ін.] // IEEE Symposium on Security and Privacy (SP). San Francisco, CA, USA. 2023. С. 2658–2675.

15. Okusi O. Cybersecurity techniques for detecting and preventing cross-site scripting attacks / World Journal of Innovation and Modern Technology. 2024. №2(8). С. 71–89.

16. Antonelli D. Cascella R., Schiano A. Dirclustering: A semantic clustering approach to optimize website structure discovery during penetration testing / Journal of Computer Viruses and Malware. 2024. №20. С. 565–577.

17. Defending multi-cloud applications against man-in-the-middle attacks / [M. Reece, N. Rastogi, T. Lander та ін.] // Proceedings of the 29th ACM Symposium on Access Control Models and Technologies (SACMAT 2024). New York, NY, USA: Association for Computing Machinery. 2024. С. 47–52.

18. Enhancing web browser extensions: Preventing JavaScript code injection and vulnerabilities / [T. Singh, K. Singh, N. Varshney та ін.] // Innovative Computing and Communications. ICICC 2024. Singapore: Springer. 2024. №1020. С. 44.

19. Mapping and analysis of common vulnerabilities in popular web servers / M. Barocsai, J. Can, M. Karresand, S. Nadjm-Tehrani // Critical Information Infrastructures Security. Cham: Springer. 2024. С. 14599–14604.

20. Kumar I. S. Methodology for safeguarding cloud servers from web application attacks / International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering. Chennai, India. 2023. С. 1–6.

21. Analysis of web application vulnerabilities using dynamic application security testing / R.Singh, G. M. Kumar, D. R. Patil, S. M. Patil // IEEE 9th International Conference for Convergence in Technology (I2CT). Pune, India. 2024. С. 1–6.

22. Shahrivar P. Millar S., Shereen E.. Detecting web application DAST attacks with machine learning / IEEE Conference on Dependable and Secure Computing (DSC). Tampa, FL, USA. 2023. С. 1–8.

23. Kuszczyński K. Walkowski M. Comparative analysis of open-source tools for conducting static code analysis / Sensors. 2023. №18(23). С. 7978–7989.

24. Tudosi A. Research on Security Weakness Using Penetration Testing in a Distributed Firewall / Sensors. 2023. №5. С. 2683–2695.

25. Samba: Detecting SSL/TLS API misuses in IoT binary applications / [K. Liu, та ін.] // IEEE INFOCOM 2024 - IEEE Conference on Computer Communications. Vancouver, BC, Canada. 2024. С. 2029–2038.

26. Enhancing Monitoring Performance: A Microservices Approach to Monitoring with Spyware Techniques and Prediction Models / A.Rossetto, D. Noetzold, L. Silva, V. Leithardt // Sensors. 2024. №13. С. 4212–4237.

27. Alquwayzani A., Aldossri R., Frikha M. Mitigating Security Risks in Firewalls and Web Applications using Vulnerability Assessment and Penetration Testing (VAPT) / International Journal of Advanced Computer Science and Applications. 2024. №5. С. 1348–1364.

28. Softić J., Vejzović Z. Impact of vulnerability assessment and penetration testing (VAPT) on operating system security / 22nd International Symposium Infoteh-Jahorina (INFOTEH). East Sarajevo, Bosnia and Herzegovina. 2023. С. 1–6.

29. Dynamic Security Analysis on Android: A Systematic Literature Review / [T. Sutter, T. Kehrer, M. Rennhard та ін.] // IEEE Access. 2024. №12. С. 57261–57287.

30. Testability Tarpits: the Impact of Code Patterns on the Security Testing of Web Applications / [F. Al Kassar, G. Clerici, L. Compagna та ін.] // Annual Network and Distributed System Security Symposium. 2022. №29. С. 1–18.

31. Automatically Seed Corpus and Fuzzing Executables Generation Using Test Framework / J.Jeon, M. Ryu, D. Kim, H. Kim // IEEE Acces. 2022. №10. С. 90408–90428.

32. Коробейнікова Т. І., Кравчук Н. В. Організація захищеного доступу до web-серверів засобами машинного навчання. 2023. №1 (20). С. 52–59.

33. Коробейнікова Т. І., Кравчук Н. В. Огляд безпечного доступу до веб-ресурсу за допомогою методів машинного навчання / TInternational Scientific Integration, Seattle, Washington, USA: ProConference. 2023. С. 26–33.

34. Кравчук Н., Коробейнікова Т. Безпечний доступ до серверів інформаційних систем, забезпечений ML-моделлю для блокування шкідливих запитів / *Вісник Хмельницького національного університету*. 2024. №5. С. 327–333.

References:

1. Tereykovskyi, I. A., & Hnatyuk, S. O. (2022). *Zakhyst informatsiyi v komp'yuternykh systemakh* [Protection of information in computer systems]. Kyiv: KPI. pp. 135.

2. Zakharchenko, S. M., Troianovska, T. I., & Boyko, O. V. (2017). *Pobudova zakhyschenykh merezh na bazi obladnannya kompanii Cisco* [Building secure networks based on Cisco equipment]. Vinnytsia: VNTU. pp. 133.

3. Korobeynikova, T. I., & Zakharchenko, S. M. (2021). *Tekhnologii zakhystu lokal'nykh merezh na osnovi obladnannya CISCO* [Technologies for protecting local networks based on CISCO equipment]. Lviv: Vydavnytstvo Lvivskoi politekhniky.

4. Denysiuk, V. O., & Pysmennyi, V. V. (2017). Zakhyst informatsii u lokal'nykh merezhakh. In *Materialy Vseukrainskoi studentkoi konferentsii "Kibernetychne upravlinnia ekonomichnymy obiektyamy"* (pp. 55–56). Vinnytsia: VNAU [in Ukrainian].

5. Semenets, O., & Tetskyi, A. (2024). Analiz metodiv ta zasobiv vybory ta kompleksuvannia skaneriv vrazlyvostei dlia testuvannia na proniknennia internet system [Analysis of methods

and means for selecting and integrating vulnerability scanners for penetration testing of internet systems]. *Measuring and Computing Devices in Technological Processes*, (2), 336–347. [in Ukrainian].

6. Lysenko, S. M., & Kondratiuk, A. S. (2020). Metod otsinky ryzyku informatsiinoi bezpeky kibernetychnykh system na osnovi vzaemozalezhnosti vrazlyvostei [Method for assessing the risk of information security in cyber-physical systems based on the interdependence of vulnerabilities]. *Computer Systems and Information Technologies*, (2), 54–58. [in Ukrainian].

7. Basili, V. R., Briand, L. C., & Melo, W. L. (1996). A validation of object-oriented design metrics as quality indicators. *IEEE Transactions on Software Engineering*, 22(10), 751–761. doi:10.1109/32.545430

8. Lathar, P., Shah, R., & Srinivasa, K. (2017). Stacy-static code analysis for enhanced vulnerability detection. *Cogent Engineering*, 4(1), 1335470. doi:10.1080/23311916.2017.1335470

9. Rajapaksha, S., Senanayake, J., Kalutarage, H., & Al-Kadri, M. O. (2024). Enhancing security assurance in software development: AI-based vulnerable code detection with static analysis. In S. Katsikas et al. (Eds.), *Computer Security. ESORICS 2023 International Workshops* (Vol. 14399, p. 20). Cham: Springer. doi:10.1007/978-3-031-54129-2_20

10. Bohdanova, Ye., Chorna, T., & Malakhov, S. (2022). Ohyd potochnogo stanu zahroz, shcho obumovlenni vplivom eksplojtiv. *Kompyuterni nauky ta kiberbezpeka*, 2, 35–40.

11. Seshapriyan, T., Dinesh, S. M., & Godwin Ponsam, J. (2024). SentinelScan: Advanced network scanner and packet detection suite. In *2024 8th International Conference on Inventive Systems and Control (ICISC)* (pp. 253–258). Coimbatore, India. doi:10.1109/ICISC62624.2024.00050

12. Eriksson, B., Pellegrino, G., & Sabelfeld, A. (2021). Black Widow: Blackbox data-driven web scanning. In *2021 IEEE Symposium on Security and Privacy (SP)* (pp. 1125–1142). San Francisco, CA, USA. doi:10.1109/SP40001.2021.00022

13. Steinhauser, A., & Tuma, P. (2020). Database traffic interception for graybox detection of stored and context-sensitive XSS. *Digital Threats*, 1(3), 17. doi:10.1145/3399668

14. Trickel, E., et al. (2023). Toss a fault to your witcher: Applying grey-box coverage-guided mutational fuzzing to detect SQL and command injection vulnerabilities. In *2023 IEEE Symposium on Security and Privacy (SP)* (pp. 2658–2675). San Francisco, CA, USA.
doi:10.1109/SP46215.2023.10179317

15. Okusi, O. (2024). Cybersecurity techniques for detecting and preventing cross-site scripting attacks. *World Journal of Innovation and Modern Technology*, 8(2), 71–89.

16. Antonelli, D., Cascella, R., Schiano, A., et al. (2024). Dirclustering: A semantic clustering approach to optimize website structure discovery during penetration testing. *Journal of Computer Viruses and Malware*, 20, 565–577. doi:10.1007/s11416-024-00512-6

17. Reece, M., Rastogi, N., Lander, T., Dykstra, J., Mittal, S., & Sampson, A. (2024). Defending multi-cloud applications against man-in-the-middle attacks. In *Proceedings of the 29th ACM Symposium on Access Control Models and Technologies (SACMAT 2024)* (pp. 47–52). New York, NY, USA: Association for Computing Machinery. doi:10.1145/3649158.3657051

18. Singh, T., Singh, K. U., Varshney, N., Gupta, P., & Kumar, G. (2024). Enhancing web browser extensions: Preventing JavaScript code injection and vulnerabilities. In A. E. Hassanien, S. Anand, A. Jaiswal, & P. Kumar (Eds.), *Innovative Computing and Communications. ICICC 2024* (Vol. 1020, p. 44). Singapore: Springer. doi:10.1007/978-981-97-3588-4_44

19. Barocsai, M., Can, J., Karresand, M., & Nadjm-Tehrani, S. (2024). Mapping and analysis of common vulnerabilities in popular web servers. In S. Pickl, B. Hämmerli, P. Mattila, & A. Sevillano (Eds.), *Critical Information Infrastructures Security. CRITIS 2023* (Vol. 14599, p. 1). Cham: Springer. doi:10.1007/978-3-031-62139-0_1

20. Kumar, I., & Sharma. (2023). Methodology for safeguarding cloud servers from web application attacks. In *2023 International Conference on Research Methodologies in Knowledge Management, Artificial Intelligence and Telecommunication Engineering (RMKMATE)* (pp. 1–6). Chennai, India. doi:10.1109/RMKMATE59243.2023.10369725

21. Singh, R., Kumar Gupta, M., Patil, D. R., & Patil, S. M. (2024). Analysis of web application vulnerabilities using dynamic application security testing. In *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)* (pp. 1–6). Pune, India. doi:10.1109/I2CT61223.2024.10543484

22. Shahrivar, P., Millar, S., & Shereen, E. (2023). Detecting web application DAST attacks with machine learning. In *2023 IEEE Conference on Dependable and Secure Computing (DSC)* (pp. 1–8). Tampa, FL, USA. doi:10.1109/DSC61021.2023.10354106

23. Kuszczyński, K., & Walkowski, M. (2023). Comparative analysis of open-source tools for conducting static code analysis. *Sensors*, 23(18), 7978. doi:10.3390/s23187978

24. Tudosi, A., Graur, A., Balan, D. G., & Potorac, A. D. (2023). Research on Security Weakness Using Penetration Testing in a Distributed Firewall. *Sensors*, *23*(5),2683–2695.
https://doi.org/10.3390/s23052683

25. Liu, K., et al. (2024). Samba: Detecting SSL/TLS API misuses in IoT binary applications. In *IEEE INFOCOM 2024 - IEEE Conference on Computer Communications* (pp. 2029–2038). Vancouver, BC, Canada. doi:10.1109/INFOCOM52122.2024.10621138

26. De Moraes Rossetto, A. G., Noetzold, D., Silva, L. A., & Leithardt, V. R. Q. (2024). Enhancing Monitoring Performance: A Microservices Approach to Monitoring with Spyware Techniques and Prediction Models. Sensors, 24(13), 4212–4237. https://doi.org/10.3390/s24134212

27. Alquwayzani, A., Aldossri, R., & Frikha, M. (2024). Mitigating Security Risks in Firewalls and Web Applications using Vulnerability Assessment and Penetration Testing (VAPT). International Journal of Advanced Computer Science and Applications, 15(5).doi.org/10.14569/ijacsa.2024.01505136

28. Softić, J., & Vejzović, Z. (2023). Impact of vulnerability assessment and penetration testing (VAPT) on operating system security. In *2023 22nd International Symposium INFOTEH-JAHORINA (INFOTEH)* (pp. 1–6). East Sarajevo, Bosnia and Herzegovina. doi:10.1109/INFOTEH57020.2023.10094095

29. Sutter, T., Kehrer, T., Rennhard, M., Tellenbach, B., & Klein, J. (2024). Dynamic Security Analysis on Android: A Systematic Literature Review. IEEE Access, 12, 57261–57287. https://doi.org/10.1109/access.2024.3390612

30. Kassar, F. A., Clerici, G., Compagna, L., Balzarotti, D., & Yamaguchi, F. (2022). Testability Tarpits: the Impact of Code Patterns on the Security Testing of Web Applications. Annual Network and Distributed System Security Symposium, 1–18. https://doi.org/10.14722/ndss.2022.24150

31. Jeon, S., Ryu, M., Kim, D., & Kim, H. K. (2022). Automatically Seed Corpus and Fuzzing Executables Generation Using Test Framework. IEEE Access, 10, 90408–90428. https://doi.org/10.1109/access.2022.3202005.

32. Korobeynikova, T. I., & Kravchuk, N. V. (2023). Orhanizatsiya zakhyshchenoho dostupu do web-serveriv zasobamy mashynnoho navchannya [Organization of secure access to web servers using machine learning methods]. *International Periodical Scientific Journal «SWorldJournal»*, 20(1), 52–59. doi:10.30888/2663-5712.2023-20-01-041

33. Korobeynikova, T. I., & Kravchuk, N. V. (2023). Ohlyad bezpechnoho dostupu do veb-resursu za dopomohoyu metodiv mashynnoho navchannya [Overview of secure access to web resources using machine learning methods]. In *International Scientific Integration 2023: Mizhnarodna naukova* konferentsiia, 11 July 2023: tesi dopovidiei (pp. 26–33). Seattle, Washington, USA: ProConference. doi:10.30888/2709-2267.2023-19-01-012

34. Кравчук, Н., & Коробейнікова, Т. (2024). Безпечний доступ до серверів інформаційних систем, забезпечений ML-моделлю для блокування шкідливих запитів. Herald of Khmelnytskyi National University. Technical Sciences, 341(5), 327-333. doi.org/10.31891/2307-5732-2024-341-5-48

Оглядова стаття.
Надійшла до редакції 07.11.2024.
Прийнято до публікації 18.12.2024.