КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2(22), 2023
ISSN 2663 - 4023
DOI: 10.28925/2663-4023.2023.22.96112
УДК 004.49

Петрів Петро Петрович
асистент кафедри захисту інформації
Національний Університет «Львівська Політехніка», Львів, Україна
ORCID 0000-0002-3423-5655
petro.p.petriv@lpnu.ua

Опірський Іван Романович
Доктор технічних наук, професор, завідувач кафедри захисту інформації
Національний Університет «Львівська Політехніка», Львів, Україна
ORCID 0000-0002-8461-8996
ivan.r.opirskyi@lpnu.ua

АНАЛІЗ ПРОБЛЕМАТИКИ ВИКОРИСТАННЯ ІСНУЮЧИХ СТАНДАРТІВ З ВЕБ-ВРАЗЛИВОСТЕЙ

Анотація. У сучасному цифровому середовищі безпека веб-ресурсів набуває первинної важливості через постійне зростання кількості веб-вразливостей. Це створює потенційні ризики для користувачів та бізнесу. В цьому контексті, стандарти та методології для виявлення веб-вразливостей служать ключовим інструментом у їх ідентифікації та усуненні. Два провідних стандарти у цій області, OWASP Top 10 та CWE (Common Weakness Enumeration), надають докладні рекомендації та огляди поширених вразливостей. Однак, вони мають різниці у підходах до класифікації та оцінки вразливостей.
Ця стаття зосереджена на глибокому аналізі та порівнянні цих стандартів, виявленні їх переваг та обмежень. Основна мета полягає у розробці рекомендацій для оптимізації використання цих стандартів, адаптованих до специфічних потреб організацій, щоб забезпечити вищий рівень безпеки веб-ресурсів.

Ключові слова: веб-вразливості; стандарти безпеки; OWASP; CWE; адаптація стандартів; оцінка вразливостей.

ВСТУП

У сучасному цифровому віці безпека веб-додатків є однією з найбільших проблем, з якими стикаються організації та користувачі. З ростом залежності від веб-технологій, потенційний ризик для користувачів та організацій через веб-вразливості стає все більш очевидним [1], [2]. Зокрема, за даними Homeland Security Systems Engineering and Development Institute, в 2023 році було опубліковано список 25 найнебезпечніших програмних вразливостей (CWE) [3].

OWASP (Open Web Application Security Project) та CWE (Common Weakness Enumeration) є двома провідними стандартами у галузі безпеки веб-додатків. OWASP акцентує увагу на 10 найбільш критичних вразливостях веб-додатків [4], тоді як CWE надає докладний огляд поширених вразливостей у веб-додатках.

Обидва ці стандарти відіграють ключову роль у виявленні та усуненні веб-вразливостей, але мають різні підходи до класифікації та оцінки ризиків. Враховуючи вищезазначене, дослідження та порівняння цих стандартів є вкрай актуальним для забезпечення ефективної безпеки веб-ресурсів.

Постановка проблеми. Незважаючи на наявність відомих стандартів безпеки, таких як OWASP Top 10 та MITRE CWE Top 25, існує проблема їхньої неповної взаємної

відповідності. Це може призвести до ситуацій, коли деякі вразливості виявлені одним стандартом, але проігноровані іншим [5], [6].

Обидва списки мають спільні ризики, але їх підходи до класифікації та оцінки цих ризиків можуть відрізнятися. Така невідповідність може призвести до проблем у реальних сценаріях безпеки, коли організація, слідуючи лише одному стандарту, може пропустити вразливості, виявлені іншим.

Враховуючи це, ключовим є розуміння різниці між цими стандартами та розробка рекомендацій для їх комбінованого використання, щоб забезпечити найвищий рівень безпеки веб-застосунків.

Аналіз останніх досліджень і публікацій. У сучасному цифровому віці безпека веб-додатків є однією з ключових проблем, яка вимагає постійного дослідження та аналізу. Дві провідні ініціативи у цій сфері — це OWASP (Open Web Application Security Project) та Common Weakness Enumeration (CWE).

OWASP підтримує список з 10 найнебезпечніших вразливостей веб-додатків, який регулярно оновлюється та служить важливим джерелом інформації для спеціалістів у галузі IT [3]. З іншого боку, CWE — це формальний список слабких місць програмного та апаратного забезпечення, зокрема, CWE Top 25 представляє найбільш поширені та впливові проблеми [7].

Хоча обидва ресурси активно використовуються спільнотою безпеки, їх підходи до класифікації та оцінки ризиків можуть відрізнятися. Аналіз та порівняння цих стандартів є вкрай актуальним для забезпечення ефективної безпеки веб-ресурсів.

Мета статті. Мета цієї статті полягає не лише в аналізі та порівнянні OWASP Top 10 та CWE, але й у вивченні потенціалу їх комбінованого використання для підвищення ефективності ідентифікації та мітигації вразливостей веб-додатків. Основна увага приділяється виявленню переваг та недоліків кожного стандарту, а також їх підходам до класифікації вразливостей. Додатково, ми розглядаємо можливості їх взаємодоповнення та надаємо рекомендації організаціям щодо ефективного застосування та адаптації цих стандартів для забезпечення вищого рівня безпеки веб-ресурсів.

РЕЗУЛЬТАТИ ДОСЛІДЖЕНЬ

У рамках цього дослідження ми провели глибокий аналіз вразливостей веб-застосунків, визначених OWASP Top 10, та їх відповідників у базі CWE. Метою цього аналізу є виявлення спільних та унікальних характеристик кожної вразливості, а також розробка рекомендацій щодо їх усунення та запобігання.

Для кожної вразливості з OWASP Top 10 ми визначили відповідні підкатегорії в базі CWE, провели детальний аналіз кожної з них та надали конкретні приклади та рекомендації. Результати цього аналізу представлені у вигляді таблиці нижче, а детальний огляд кожної вразливості наведено у відповідних підрозділах.

Таблиця 1

Порівняльний аналіз вразливостей OWASP та CWE

OWASP	Опис OWASP	Підкатегорії CWE	Опис CWE	Приклади
Injection	Вразливість, яка виникає, коли ненадійний вхідний даний інтерпретується як частина команди або запиту. Це може дозволити зловмисникам виконувати небажані команди або отримувати доступ до даних.	CWE-77: Command Injection	Дозволяє зловмисникам виконувати команди на системі через ненадійні вхідні дані.	Використання команди ; rm -rf / в полі вводу.
		CWE-89: SQL Injection	Зловмисник може вставляти або «ін'єктувати» SQL код в запит. Це може призвести до несанкціонованого доступу до бази даних.	Введення '; DROP TABLE users; -- в поле вводу.
		CWE-564: Hibernate Injection	Вразливість, яка дозволяє зловмисникам маніпулювати HQL (Hibernate Query Language) запитами.	Введення специфічних HQL запитів в поле вводу.
		CWE-917: Expression Language Injection	Зловмисник може ін’єктувати вирази, які будуть виконані на сервері.	Використання ${...} в даних форми.

Injection [4]:

- **CWE-77:** Command Injection [8]. Ця вразливість виникає, коли ненадійний вхід дозволяє зловмиснику формувати та виконувати команди на операційній системі. Це може призвести до неконтрольованого виконання коду, доступу до конфіденційної інформації та інших небезпечних наслідків. Використовуйте параметризовані запити, валідуйте та санітізуйте вхідні дані, обмежуйте права користувача, який виконує команди.
- **CWE-89:** SQL Injection [8]. Зловмисники можуть вставляти або «ін’єктувати» ненадійний SQL код в запит, що призводить до виконання небажаних SQL команд. Це може дозволити зловмисникам читати, модифікувати або видаляти дані в базі даних. Використовуйте параметризовані запити, валідуйте вхідні дані та використовуйте принцип найменших привілеїв при наданні доступу до бази даних.
- **CWE-564:** Hibernate Injection [8]. Ця вразливість відноситься до ненадійного введення, яке може бути використано для формування та виконання HQL (Hibernate Query Language) запитів. Це може призвести до неконтрольованого доступу до даних або їх модифікації. Використовуйте параметризовані HQL запити, валідуйте вхідні дані та обмежуйте доступ до даних за допомогою ролевої моделі.

- **CWE-917:** Expression Language Injection [8] Зловмисники можуть вставляти вирази мови виразів, які будуть виконані на сервері. Це може призвести до витоку інформації, виконання небажаного коду або інших небезпечних наслідків. Валідуйте та санітізуйте вхідні дані, використовуйте безпечні конфігурації серверів та обмежуйте використання мови виразів там, де це можливо.

Аутентифікація є важливим аспектом безпеки веб-застосунків. Вразливості, пов'язані з аутентифікацією, можуть дозволити зловмисникам отримувати несанкціонований доступ до системи. У цьому розділі ми детально розглядаємо вразливість «Broken Authentication» з OWASP Top 10 та аналізуємо її відповідники в базі CWE.

Таблиця 2

Аналіз вразливості «Broken Authentication»

OWASP	Опис OWASP	Підкатегорії CWE	Опис CWE	Приклади
Broken Authentication	Вразливості в процесах аутентифікації та сесійного управління, які можуть бути використані зловмисниками для отримання несанкціонованого доступу.	CWE-287: Improper Authentication	Неналежна аутентифікація відбувається, коли система неправильно перевіряє особу користувача.	Неправильна перевірка облікових даних користувача.
		CWE-306: Missing Authentication for Critical Function	Відсутня аутентифікація для критичної функції, що дозволяє зловмисникам отримати доступ до функцій без перевірки особи.	Доступ до адміністративної панелі без необхідності входу.
		CWE-384: Session Fixation	Зловмисник може фіксувати сесійний ідентифікатор користувача, що дозволяє йому використовувати цю сесію.	Зловмисник відправляє жертві посилання з фіксованим сесійним ідентифікатором.
		CWE-613: Insufficient Session Expiration	Сесія не закінчується або закінчується занадто пізно, що створює вікно можливостей для атак.	Користувач залишає свій аккаунт, але сесія залишається активною протягом тривалого часу.

Broken Authentication [6]:

- **CWE-287:** Improper Authentication [8]. Ця вразливість виникає, коли система не вдається правильно або повністю перевірити особу користувача. Це може бути результатом використання слабких алгоритмів

перевірки, відсутності перевірки або неправильної реалізації процесу аутентифікації. Використовуйте сильні алгоритми аутентифікації, переконайтеся, що всі точки входу системи вимагають належної аутентифікації, та регулярно перевіряйте та тестуйте свої системи на наявність вразливостей.

- **CWE-306**: Missing Authentication for Critical Function [8]. Ця вразливість відноситься до ситуацій, коли критичні функції системи доступні без необхідної аутентифікації. Це може дозволити зловмисникам виконувати важливі операції без будь-яких обмежень. Всі критичні функції системи повинні вимагати належної аутентифікації. Перевірте конфігурації та права доступу, щоб уникнути ненавмисних витоків доступу.
- **CWE-384:** Session Fixation [8]. Зловмисник може спробувати використовувати вже існуючий сесійний ідентифікатор, змусивши жертву використовувати цей ідентифікатор. Це може дозволити зловмиснику «зафіксувати» сесію користувача та отримати доступ до його облікового запису. Використовуйте методи, які гарантують унікальність сесійних ідентифікаторів після кожного входу користувача. Також регулярно оновлюйте сесійні ідентифікатори під час сесії.
- **CWE-613:** Insufficient Session Expiration [8]. Якщо сесія користувача не закінчується або закінчується занадто пізно, зловмисник може отримати доступ до активної сесії, особливо якщо жертва залишила свій пристрій без нагляду. Встановіть короткі та обґрунтовані терміни дії сесій. Забезпечте автоматичне закінчення сесії після певного періоду бездіяльності користувача.

Витік чутливих даних не лише може призвести до втрати конфіденційності користувацької інформації або корпоративних даних, але й створює ризик фінансових втрат, репутаційних ризиків та інших юридичних наслідків для організацій. У цьому розділі ми детально розглядаємо вразливість «Sensitive Data Exposure» з OWASP Top 10 та аналізуємо її відповідники в базі CWE.

Таблиця 3

Аналіз вразливості «Sensitive Data Exposure»

OWASP	Опис OWASP	Підкатегорії CWE	Опис CWE	Приклади
Sensitive Data Exposure	Витік чутливих даних може статися через ряд причин, включаючи слабкі алгоритми шифрування, вразливості в процесах аутентифікації та сесійного управління, а також неналежне зберігання даних.	CWE-287: Improper Authentication	Неналежна аутентифікація відбувається, коли система неправильно перевіряє особу користувача.	Неправильна перевірка облікових даних користувача.
		CWE-306: Missing Authentication for Critical Function	Відсутня аутентифікація для критичної функції, що дозволяє зловмисникам отримати доступ до функцій без перевірки особи.	Доступ до адміністративної панелі без необхідності входу.

		CWE-384: Session Fixation	Зловмисник може фіксувати сесійний ідентифікатор користувача, що дозволяє йому використовувати цю сесію.	Зловмисник відправляє жертві посилання з фіксованим сесійним ідентифікатором.
		CWE-613: Insufficient Session Expiration	Сесія не закінчується або закінчується занадто пізно, що створює вікно можливостей для атак.	Користувач залишає свій аккаунт, але сесія залишається активною протягом тривалого часу.

Sensitive Data Exposure [6]:

- **CWE-287:** Improper Authentication [8]. Неналежна аутентифікація може призвести до витоку чутливих даних користувача. Якщо система не перевіряє особу користувача належним чином, це може дозволити несанкціонований доступ до чутливої інформації, такої як особисті дані, фінансова інформація та інше. Використовуйте сильні методи аутентифікації, переконайтеся, що всі точки входу системи вимагають належної аутентифікації, та регулярно перевіряйте свої системи на наявність вразливостей.
- **CWE-306:** Missing Authentication for Critical Function [8]. Відсутність аутентифікації для критичних функцій може призвести до витоку чутливих даних. Зловмисники можуть отримати доступ до цих функцій без будь-якої перевірки особи, що може призвести до витоку, модифікації або видалення даних. Забезпечте, що всі критичні функції системи вимагають належної аутентифікації. Перевірте конфігурації та права доступу, щоб уникнути ненавмисних витоків доступу.
- **CWE-384:** Session Fixation [8]. Якщо зловмисник може «зафіксувати» сесійний ідентифікатор, він може отримати доступ до чутливої інформації, що зберігається в сесії користувача. Це може включати особисті дані, інформацію про покупки та інше. Використовуйте методи, які гарантують унікальність сесійних ідентифікаторів після кожного входу користувача. Також регулярно оновлюйте сесійні ідентифікатори під час сесії.
- **CWE-613:** Insufficient Session Expiration [8]. Якщо сесія користувача не закінчується або закінчується занадто пізно, зловмисник може отримати доступ до активної сесії і, відповідно, до чутливої інформації, що зберігається в ній. Встановіть короткі та обґрунтовані терміни дії сесій. Забезпечте автоматичне закінчення сесії після певного періоду бездіяльності користувача.

Використання XML як формату для передачі даних, хоча й є популярним та зручним, може призвести до специфічних вразливостей. Однією з найбільш небезпечних є атаки на зовнішні сутності XML (XXE). Ці атаки можуть не лише призвести до витоку внутрішніх файлів, але й дозволити зловмисникам виконувати віддалений код, робити запити до внутрішніх систем, а також викликати інші небезпечні наслідки, що можуть

підривати конфіденційність, цілісність та доступність системи. У цьому розділі ми детально розглядаємо вразливість «XML External Entities (XXE)» з OWASP Top 10 та аналізуємо її відповідник в базі CWE.

Таблиця 4

Аналіз вразливості «XML External Entities (XXE)»

OWASP	Опис OWASP	Підкатегорії CWE	Опис CWE	Приклади
XML External Entities (XXE)	Атаки XXE виникають, коли зовнішні сутності XML використовуються для завантаження змісту зовнішнього джерела. Це може призвести до витоку внутрішніх файлів, виконання віддаленого коду та інших небезпечних наслідків.	CWE-611: Improper Restriction of XML External Entity Reference ('XXE')	Вразливість виникає, коли програма обробляє вхідні XML документи, не обмежуючи використання зовнішніх сутностей.	Атака, яка використовує зовнішню сутність для завантаження локального файлу з сервера.

XML External Entities (XXE) [6]:

- **CWE-611:** Improper Restriction of XML External Entity Reference ('XXE') [7]. Атаки XXE виникають, коли програма обробляє вхідні XML документи без належного обмеження використання зовнішніх сутностей. Зловмисники можуть використовувати це для завантаження змісту зовнішнього джерела, що може призвести до витоку внутрішніх файлів, виконання віддаленого коду, запитів до внутрішніх систем та інших небезпечних наслідків. Вимкніть обробку зовнішніх сутностей у вашому XML парсері. Використовуйте менш вразливі формати обміну даними, такі як JSON, якщо це можливо. Перевіряйте та валідуйте вхідні дані перед їх обробкою. Використовуйте безпечні конфігурації та бібліотеки, які спеціально розроблені для запобігання атакам XXE.

Контроль доступу є ключовим елементом безпеки будь-якої інформаційної системи. Він визначає, хто може робити що в системі, і коли правильно налаштований, служить потужним інструментом для захисту від несанкціонованого доступу. Однак, якщо контроль доступу не належним чином налаштований або реалізований, це може створити великі діри в безпеці, дозволяючи зловмисникам отримати доступ до конфіденційних даних, виконувати несанкціоновані дії або навіть отримати повний контроль над системою. У цьому розділі ми детально розглядаємо вразливість «Broken Access Control» з OWASP Top 10 та аналізуємо її відповідники в базі CWE.

Таблиця 5

Аналіз вразливості «Broken Access Control»

OWASP	Опис OWASP	Підкатегорії CWE	Опис CWE	Приклади
Broken Access Control	Неналежний контроль доступу може дозволити зловмисникам обходити обмеження доступу, виконувати дії від імені інших користувачів або отримувати доступ до	CWE-284: Improper Access Control	Відсутність або неналежна реалізація механізмів контролю доступу.	Зловмисник отримує доступ до адміністративної панелі без належних прав.

	конфіденційної інформації.			
		CWE-285: Improper Authorization	Неналежна авторизація може дозволити користувачам виконувати дії, для яких вони не мають прав.	Користувач без прав адміністратора може видаляти облікові записи інших користувачів.
		CWE-639: Authorization Bypass Through User-Controlled Key	Зловмисник може обійти авторизацію, контролюючи ключ, який використовується для перевірки доступу.	Зловмисник змінює ідентифікатор користувача в URL, щоб отримати доступ до облікового запису іншого користувача.

Broken Access Control [6]:

- **CWE-284:** Improper Access Control [8]. Неналежний контроль доступу відбувається, коли система дозволяє користувачам виконувати дії, для яких вони не мають належних прав. Це може призвести до несанкціонованого доступу до даних, зміни конфігурації системи або виконання дій від імені інших користувачів. Впроваджуйте принцип найменших привілеїв, надаючи користувачам лише ті права, які їм дійсно потрібні. Регулярно перевіряйте та аудитуйте права доступу користувачів. Використовуйте системи авторизації, які можна легко масштабувати та адаптувати до змінюваних вимог безпеки.
- **CWE-285:** Improper Authorization [8]. Неналежна авторизація відбувається, коли система не перевіряє, чи має користувач право виконувати певну дію. Це може призвести до виконання дій, які повинні бути обмежені або заборонені для даного користувача. Використовуйте централізовану систему авторизації, яка перевіряє права користувача перед виконанням будь-якої дії. Переконайтеся, що всі точки входу в систему вимагають належної авторизації. Регулярно перевіряйте логи системи на наявність спроб несанкціонованого доступу.
- **CWE-639:** Authorization Bypass Through User-Controlled Key [6]. Ця вразливість виникає, коли зловмисник може контролювати ключ, який використовується системою для перевірки доступу. Це може призвести до обходу авторизації та отримання доступу до ресурсів, які повинні бути обмежені. Не довіряйте ключам, які контролюються користувачами. Використовуйте криптографічно безпечні методи генерації ключів. Перевіряйте автентичність та цілісність ключів перед їх використанням.

Безпека конфігурації є важливим аспектом захисту будь-якої інформаційної системи. Вона включає в себе правильне налаштування параметрів, служб та функцій системи, щоб запобігти потенційним загрозам. Неправильна або недостатня конфігурація може створити слабкі місця, які можуть бути використані зловмисниками для витоку чутливої інформації, отримання неналежного доступу або проведення інших зловмисних дій. У цьому розділі ми детально розглядаємо вразливість «Security Misconfiguration» з OWASP Top 10 та аналізуємо її відповідники в базі CWE.

Таблиця 6

Аналіз вразливості «Security Misconfiguration»

OWASP	Опис OWASP	Підкатегорії CWE	Опис CWE	Приклади
Security Misconfiguration	Неправильна конфігурація безпеки може призвести до неналежного доступу, витоку даних або інших потенційних загроз.	CWE-16: Configuration	Проблеми, пов'язані з неправильною конфігурацією системи.	Відсутність пароля для адміністративного доступу до бази даних.
		CWE-2: Environment	Проблеми, пов'язані з неправильною конфігурацією середовища.	Відсутність оновлень безпеки в операційній системі.
		CWE-215: Information Exposure Through Debug Information	Витік інформації через відомості для налагодження.	Відображення стеку викликів при помилці на веб-сайті.
		CWE-548: Exposure of Information Through Directory Listing	Витік інформації через перелік директорій.	Доступ до списку файлів на веб-сервері через неналежну конфігурацію.

Security Misconfiguration [6]:

- **CWE-16:** Configuration [8]. Неправильна конфігурація системи може призвести до ряду проблем з безпекою, включаючи неналежний доступ, витік даних та інші потенційні загрози. Це може включати все, від неналежно налаштованих серверів до відсутності оновлень безпеки. Регулярно перевіряйте та оновлюйте конфігурації системи. Використовуйте автоматизовані інструменти для виявлення та усунення неправильних конфігурацій. Забезпечте, щоб усі застосунки та системи були належно патчені та оновлені.
- **CWE-2:** Environment [8]. Неправильна конфігурація середовища може створити додаткові вектори атаки для зловмисників. Це може включати в себе використання застарілих бібліотек, відсутність оновлень безпеки або неналежну конфігурацію мережі. Впроваджуйте строгий контроль над конфігурацією середовища. Регулярно проводьте аудит середовища на предмет потенційних проблем з безпекою. Забезпечте, щоб усі компоненти системи були належно налаштовані та оновлені.
- **CWE-215:** Information Exposure Through Debug Information [8]. Відображення інформації для налагодження може призвести до витоку чутливої інформації, яка може бути використана зловмисниками для подальших атак. Вимкніть відображення інформації для налагодження в продакшн-середовищі. Використовуйте механізми журналювання для зберігання інформації для налагодження, забезпечуючи її належний захист.
- **CWE-548:** Exposure of Information Through Directory Listing [8]. Якщо сервер налаштований неналежним чином, зловмисники можуть отримати доступ до

списка директорій та файлів, що може призвести до витоку чутливої інформації. Вимкніть функцію переліку директорій на веб-серверах. Забезпечте, щоб усі директорії та файли були належно захищені від несанкціонованого доступу.

Cross-Site Scripting (XSS) є однією з найпоширеніших веб-вразливостей, яка дозволяє зловмисникам вставляти та виконувати зловмисний код в браузері жертви, часто без її відома. Це може призвести не лише до крадіжки сесійних кукісів, але й до виконання дій від імені жертви, спотворення веб-сторінки або навіть до розповсюдження шкідливого ПЗ. Ці атаки можуть мати довгострокові наслідки для користувачів та організацій. У цьому розділі ми детально розглядаємо вразливість «Cross-Site Scripting (XSS)» з OWASP Top 10 та аналізуємо її відповідники в базі CWE.

Таблиця 7

Аналіз вразливості «Cross-Site Scripting (XSS)»

OWASP	Опис OWASP	Підкатегорії CWE	Опис CWE	Приклади
Cross-Site Scripting (XSS)	Атака, при якій зловмисники можуть вставляти та виконувати зловмисний код в браузері жертви.	CWE-79: Improper Neutralization of Input During Web Page Generation	Вставка зловмисного коду через неналежну обробку вхідних даних при генерації веб-сторінки.	Вставка <script>alert('XSS')</script> в поле коментарів на веб-сайті.
		CWE-80: Improper Neutralization of Script-Related HTML Tags	Неналежна обробка тегів, пов’язаних із скриптами, що призводить до виконання зловмисного коду.	Вставка <img src="x" onerror="alert('XSS')"> в блозі.
		CWE-81: Improper Neutralization of Script in an Error Message	Вставка зловмисного коду через неналежну обробку вхідних даних в повідомленнях про помилки.	Введення неправильного значення, яке викликає помилку із вставленим кодом XSS.
		CWE-82: Improper Neutralization of Script in Attributes of IMG Tags	Вставка зловмисного коду через атрибути тегів IMG, що призводить до виконання зловмисного коду.	Вставка <img src="x" onmouseover="alert('XSS')"> на форумі.

Cross-Site Scripting (XSS) [6]:

- **CWE-79:** Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting') [8]. Ця вразливість виникає, коли веб-додаток не належним чином обробляє вхідні дані користувача при генерації веб-сторінки, дозволяючи зловмисникам вставляти зловмисний код, який буде виконуватися в браузері жертви. Використовуйте безпечні методи обробки

вхідних даних, такі як використання списків дозволених символів або екранування спеціальних символів. Використовуйте Content Security Policy (CSP) для обмеження виконання зловмисного коду.

- **CWE-80:** Improper Neutralization of Script-Related HTML Tags in a Web Page (Basic XSS) [8]. Ця вразливість виникає, коли веб-додаток не належним чином обробляє теги, пов'язані із скриптами, дозволяючи зловмисникам вставляти та виконувати зловмисний код. Завжди екрануйте спеціальні символи та теги перед вставкою вхідних даних на веб-сторінку. Використовуйте безпечні бібліотеки, які автоматично обробляють вхідні дані для запобігання XSS.
- **CWE-81:** Improper Neutralization of Script in an Error Message Web Page [8]. Ця вразливість виникає, коли веб-додаток вставляє вхідні дані користувача в повідомлення про помилку без належної обробки, що може призвести до виконання зловмисного коду. Не вставляйте вхідні дані користувача безпосередньо в повідомлення про помилки. Використовуйте параметризовані повідомлення про помилки та екрануйте вхідні дані.
- **CWE-82:** Improper Neutralization of Script in Attributes of IMG Tags in a Web Page [8]. Ця вразливість виникає, коли веб-додаток дозволяє користувачам вставляти зловмисний код в атрибути тегів IMG, що може призвести до виконання зловмисного коду при завантаженні зображення. Обмежте можливість користувачів вставляти код в атрибути тегів IMG. Використовуйте безпечні методи обробки вхідних даних та екрануйте спеціальні символи.

Небезпечна десеріалізація є вразливістю, при якій зловмисники можуть використовувати десеріалізовані об'єкти для виконання шкідливих дій в системі. Це може призвести до виконання довільного коду, обходу механізмів аутентифікації та авторизації, а також до втручання в логіку роботи застосунку. Така вразливість може дати зловмисникам можливість отримати несанкціонований доступ до системних ресурсів або даних. У цьому розділі ми детально розглядаємо вразливість «Insecure Deserialization» з OWASP Top 10 та аналізуємо її відповідники в базі CWE.

Таблиця 8

Аналіз вразливості «Insecure Deserialization»

OWASP	Опис OWASP	Підкатегорії CWE	Опис CWE	Приклади
Insecure Deserialization	Атака, при якій зловмисники можуть використовувати десеріалізовані об'єкти для виконання шкідливих дій в системі.	CWE-502: Deserialization of Untrusted Data	Вразливість, яка виникає, коли додаток десеріалізує дані, які не були належно перевірені або санітізовані.	Використання десеріалізованого об'єкта для виконання довільного коду на сервері.

Insecure Deserialization [6]:

- **CWE-502:** Deserialization of Untrusted Data [8]. Десеріалізація ненадійних даних відбувається, коли додаток приймає вхідні дані, які можуть бути десеріалізовані в об'єкт. Якщо ці дані були змінені або підроблені зловмисниками, це може призвести до виконання довільного коду, витоку даних або обходу механізмів безпеки. Ніколи не десеріалізуйте дані,

отримані від ненадійних джерел. Використовуйте безпечні механізми десеріалізації, які обмежують типи об'єктів, які можна десеріалізувати. Застосовуйте принцип найменших привілеїв, обмежуючи доступ до функцій десеріалізації лише для довірених користувачів або компонентів системи.

Небезпечна десеріалізація є серйозною загрозою для безпеки веб-додатків, оскільки вона може дати зловмисникам можливість виконувати код на сервері, отримувати доступ до чутливої інформації або обходити механізми аутентифікації та авторизації. Важливо розуміти механізми десеріалізації та вживати необхідних заходів для запобігання цій вразливості.

Використання компонентів із відомими вразливостями є поширеною проблемою в індустрії кібербезпеки. Це відбувається, коли розробники, намагаючись спростити свою роботу та прискорити розробку, використовують сторонні бібліотеки, плагіни або інші компоненти, які мають відомі вразливості. Такий підхід може призвести до компрометації системи, навіть якщо основний код застосунку є безпечним. У цьому розділі ми детально розглядаємо проблему «Використання компонентів із відомими вразливостями» з OWASP Top 10 та аналізуємо її відповідники в базі CWE.

Таблиця 9

Аналіз «Використання компонентів із відомими вразливостями»

OWASP	Опис OWASP	Підкатегорії CWE	Опис CWE	Приклади
Using Components with Known Vulnerabilities	Проблема виникає, коли в системі використовуються компоненти, які мають відомі вразливості, що може призвести до атак або компрометації системи.	CWE-937: Using Components with Known Vulnerabilities	Використання компонентів у програмному забезпеченні, які мають відомі вразливості.	Використання застарілої версії бібліотеки jQuery, яка має відомі вразливості.
		CWE-200: Information Exposure	Ненавмисне викриття інформації, яка не повинна бути доступна зовні.	Відображення повідомлень про помилки, які містять чутливу інформацію про систему.

Using Components with Known Vulnerabilities [6]:

- **CWE-937:** Using Components with Known Vulnerabilities [8]. Ця вразливість виникає, коли в системі використовуються компоненти, такі як бібліотеки, плагіни або інші модулі, які мають відомі вразливості. Зловмисники можуть використовувати ці вразливості для атак на систему, отримання несанкціонованого доступу або виконання довільного коду. Регулярно перевіряйте використовувані компоненти на наявність відомих вразливостей. Оновлюйте компоненти до останніх безпечних версій. Використовуйте інструменти для автоматичного виявлення вразливостей в залежностях.
- **CWE-200:** Information Exposure [8]. Ця вразливість відноситься до ненавмисного викриття інформації, яка може бути використана зловмисниками для подальших атак. Це може включати в себе відображення чутливої інформації в повідомленнях про помилки, логах або інших

відкритих джерелах. Обмежуйте відображення чутливої інформації в повідомленнях про помилки або логах. Використовуйте принцип найменших привілеїв для обмеження доступу до чутливої інформації. Застосовуйте шифрування та інші механізми захисту для зберігання та передачі чутливої інформації.

Використання компонентів із відомими вразливостями може призвести до серйозних наслідків для безпеки системи. Важливо розуміти ризики, пов'язані з використанням застарілих або небезпечних компонентів, та вживати необхідних заходів для їх усунення.

Недостатнє ведення журналів та моніторинг є критичним аспектом безпеки, який, на жаль, часто ігнорується або недооцінюється. Відсутність ефективного моніторингу та ведення журналів може не тільки приховати поточні атаки, але й ускладнити виявлення та розслідування минулих інцидентів. Це може призвести до того, що зловмисники зможуть вторгнутися в систему, виконувати шкідливі дії та залишитися невиявленими протягом тривалого часу. У цьому розділі ми детально розглядаємо проблему «Недостатнє ведення журналів та моніторинг» з OWASP Top 10 та аналізуємо її відповідники в базі CWE.

Таблиця 10

Аналіз «Недостатнє ведення журналів та моніторинг»

OWASP	Опис OWASP	Підкатегорії CWE	Опис CWE	Приклади
Insufficient Logging & Monitoring	Проблема, коли система не веде достатньо детальних журналів або не має ефективних механізмів моніторингу, що може призвести до відсутності виявлення атак або порушень безпеки.	CWE-778: Insufficient Logging	Відсутність або недостатність ведення журналів, що може ускладнити виявлення або аналіз інцидентів безпеки.	Відсутність записів про невдалі спроби входу в систему.
		CWE-223: Omission of Security-relevant Information	Відсутність важливої інформації в журналах, яка могла б допомогти в аналізі інцидентів безпеки.	Журнали не фіксують IP-адреси користувачів, які намагаються отримати доступ.
		CWE-285: Improper Authorization	Неправильна реалізація механізмів авторизації, що може дозволити зловмисникам отримати доступ до ресурсів, до яких вони не мають прав доступу.	Користувач без адміністративних прав може отримати доступ до панелі адміністратора.

Insufficient Logging & Monitoring [6]:

- **CWE-778:** Insufficient Logging [8]. Відсутність або недостатнє ведення журналів може призвести до того, що інциденти безпеки або порушення не будуть виявлені або аналізовані належним чином. Журнали можуть не фіксувати важливі події, такі як невдалі спроби входу, доступ до чутливих ресурсів або зміни конфігурації. Забезпечте ведення детальних журналів для всіх критичних подій системи. Регулярно перевіряйте та аналізуйте журнали на наявність підозрілих дій. Застосовуйте автоматичні інструменти для моніторингу журналів та виявлення аномалій.
- **CWE-223:** Omission of Security-relevant Information [8]. Журнали можуть не містити важливої інформації, необхідної для аналізу інцидентів безпеки. Це може ускладнити виявлення джерела атаки, методів вторгнення або інших деталей, які можуть бути корисними для розслідування. Включіть в журнали всю важливу інформацію, таку як IP-адреси, часові мітки, типи запитів та ін. Регулярно перевіряйте журнали на наявність повної інформації про події.
- **CWE-285:** Improper Authorization [8]. Неправильна реалізація механізмів авторизації може дозволити зловмисникам отримати доступ до ресурсів, до яких вони не мають прав доступу. Це може призвести до витоку даних, зміни конфігурації або інших небажаних дій. Реалізуйте строгі механізми авторизації для всіх ресурсів системи. Регулярно перевіряйте права доступу користувачів та груп, щоб запобігти неправильному доступу.

Недостатнє ведення журналів та моніторинг може призвести до того, що атаки або порушення безпеки не будуть виявлені вчасно. Важливо розуміти значущість правильного ведення журналів та ефективного моніторингу для забезпечення безпеки системи.

У ході нашого дослідження ми провели детальний аналіз двох ключових стандартів у сфері веб-безпеки: OWASP Top 10 та CWE. Ці стандарти відіграють важливу роль у виявленні та усуненні веб-вразливостей.

OWASP Top 10 надає загальний огляд десяти найбільш критичних ризиків безпеки для веб-застосунків, заснований на досвіді спільноти. З іншого боку, CWE надає більш деталізований список загальних слабких місць програмного забезпечення.

Хоча обидва стандарти мають багато спільного, існують і відмінності у їх підходах. Наприклад, деякі вразливості можуть бути високо оцінені в одному списку, але менш акцентовані в іншому. Це може призвести до ситуацій, коли організації, що слідують лише одному з цих стандартів, можуть пропустити важливі аспекти безпеки.

Також варто зазначити, що існує потреба в комбінованому підході до використання цих стандартів. Інтеграція рекомендацій з обох джерел є не тільки можливою, але й вкрай ефективною. Комбінація методологій OWASP і CWE дозволяє детальніше аналізувати та мітігувати вразливості, використовуючи сильні сторони обох стандартів. Це сприяє розробці більш голістичних стратегій безпеки, адаптованих до специфічних потреб кожного веб-додатку.

ВИСНОВКИ

У цьому дослідженні ми не лише зосередилися на аналізі OWASP Top 10 та CWE, але й виявили великий потенціал для їх взаємодоповнення. Інтегрований підхід, який

об'єднує переваги обох стандартів, сприяє глибшому розумінню вразливостей, їх причин, потенційних наслідків та способів мітігації.

Вразливості, які були розглянуті, хоча й класифіковані за OWASP та CWE, мають різний ступінь ризику. Однак кожна вразливість важлива і вимагає специфічних заходів для її усунення та запобігання.

Через детальний аналіз цих вразливостей ми прийшли до висновку, що комбінований підхід, який включає елементи як OWASP Top 10, так і CWE, надає більш цілісний і глибокий погляд на аспекти безпеки веб-застосунків. Наш аналіз підкреслює важливість комбінованого використання OWASP та CWE для оптимізації стратегій веб-безпеки. Рекомендується розробити специфічні процедури для інтеграції цих методологій, адаптовані під конкретні веб-додатки та їх архітектурні особливості. Використання автоматизованих інструментів та навчання персоналу є ключовими елементами для реалізації цього підходу ефективно. Постійний моніторинг та адаптація стратегій безпеки забезпечать їх актуальність та ефективність у відповідь на еволюцію загроз.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Yevseiev, S., et al. (2022). Modeling of security systems for critical infrastructure facilities. *PC Technology Center*.
2. Kurii, Y., & Opirskyy, I. (2021). Analysis and Comparison of the NIST SP 800-53 and ISO/IEC 27001:2013. In *Cybersecurity Providing in Information and Telecommunication Systems, 3288*, 21–32.
3. *2023 CWE Top 25 Most Dangerous Software Weaknesses.* (2023). Cybersecurity & Infrastructure Security Agency. https://www.cisa.gov/news-events/alerts/2023/06/29/2023-cwe-top-25-most-dangerous-software-weaknesses
4. Nadeau, J. (2023). *The top 10 API security risks OWASP list for 2023*. Security Intelligence. https://securityintelligence.com/articles/the-top-10-api-security-risks-owasp-list-for-2023
5. *Common Weakness Enumeration (CWE) Top 25*. (2023). Common Weakness Enumeration https://cwe.mitre.org/top25/archive/2023/2023_top25_list.html
6. *Navigating API Security: The OWASP API Security Top 10 2023.* (2023). APTORI. https://aptori.dev/blog/navigating-api-security-the-owasp-api-security-top-10-2023
7. *Frequently Asked Questions (FAQ).* Common Weakness Enumeration. https://cwe.mitre.org/about/faq.html
8. *Common Weakness Enumeration (CWE) — database*. Cybersecurity Help. https://www.cybersecurity-help.cz/vdb/cwe/
9. *OWASP — wiki*. Wikipedia. https://en.wikipedia.org/wiki/OWASP
10. *CWE — wiki*. Wikipedia. https://en.wikipedia.org/wiki/Common_Weakness_Enumeration
11. *CWE* Definitions. CVE Details. https://www.cvedetails.com/cwe-definitions/
12. *Difference between CWE, CVE, and OWASP*. Crashtest Security. https://crashtest-security.com/common-weakness-enumeration/
13. National Vulnerability Database. https://nvd.nist.gov/vuln/categories
14. *CWE (Common Weakness Enumeration) and the CWE Top 25 Explained.* HackerOne. https://www.hackerone.com/vulnerability-management/cwe-common-weakness-enumeration-and-cwe-top-25-explained
15. *CWE — database*. Security Database. https://www.security-database.com/cwe.php

КІБЕРБЕЗПЕКА: освіта, наука, техніка
CYBERSECURITY: EDUCATION, SCIENCE, TECHNIQUE
№ 2 (22), 2023
ISSN 2663 - 4023

Petro Petriv
Assistant of the Department of Information Protection
Lviv Polytechnic National University, Lviv, Ukraine
ORCID 0000-0002-3423-5655
petro.p.petriv@lpnu.ua

Ivan Opirskyi
Doctor of Science, professor, Head of the Department of Information Protection
Lviv Polytechnic National University, Lviv, Ukraine
ORCID 0000-0002-8461-8996
ivan.r.opirskyi@lpnu.ua

ANALYSIS OF PROBLEMS OF USING EXISTING STANDARDS WITH WEB VULNERABILITIES

Abstract. In today's digital environment, the security of web resources is of primary importance due to the constant increase in the number of web vulnerabilities. This creates potential risks for users and businesses. In this context, standards and methodologies for detecting web vulnerabilities serve as a key tool in their identification and elimination.
The two leading standards in this area, OWASP Top 10 and CWE (Common Weakness Enumeration), provide detailed recommendations and overviews of common vulnerabilities. However, they differ in their approaches to vulnerability classification and assessment.
This article focuses on an in-depth analysis and comparison of these standards, identifying their advantages and limitations. The main goal is to develop recommendations to optimize the use of these standards, adapted to the specific needs of organizations, to ensure a higher level of security of web resources.

Keywords: web vulnerabilities; safety standards; OWASP; CWE; adaptation of standards; vulnerability assessment.

REFERENCES (TRANSLATED AND TRANSLITERATED)

1. Yevseiev, S., et al. (2022). Modeling of security systems for critical infrastructure facilities. *PC Technology Center*.
2. Kurii, Y., & Opirskyy, I. (2021). Analysis and Comparison of the NIST SP 800-53 and ISO/IEC 27001:2013. In *Cybersecurity Providing in Information and Telecommunication Systems, 3288*, 21–32.
3. *2023 CWE Top 25 Most Dangerous Software Weaknesses.* (2023). Cybersecurity & Infrastructure Security Agency. https://www.cisa.gov/news-events/alerts/2023/06/29/2023-cwe-top-25-most-dangerous-software-weaknesses
4. Nadeau, J. (2023). *The top 10 API security risks OWASP list for 2023*. Security Intelligence. https://securityintelligence.com/articles/the-top-10-api-security-risks-owasp-list-for-2023
5. *Common Weakness Enumeration (CWE) Top 25*. (2023). Common Weakness Enumeration https://cwe.mitre.org/top25/archive/2023/2023_top25_list.html
6. *Navigating API Security: The OWASP API Security Top 10 2023.* (2023). APTORI. https://aptori.dev/blog/navigating-api-security-the-owasp-api-security-top-10-2023
7. *Frequently Asked Questions (FAQ).* Common Weakness Enumeration. https://cwe.mitre.org/about/faq.html
8. *Common Weakness Enumeration (CWE) — database*. Cybersecurity Help. https://www.cybersecurity-help.cz/vdb/cwe/
9. *OWASP — wiki*. Wikipedia. https://en.wikipedia.org/wiki/OWASP
10. *CWE — wiki*. Wikipedia. https://en.wikipedia.org/wiki/Common_Weakness_Enumeration
11. *CWE* Definitions. CVE Details. https://www.cvedetails.com/cwe-definitions/
12. *Difference between CWE, CVE, and OWASP*. Crashtest Security. https://crashtest-security.com/common-weakness-enumeration/
13. National Vulnerability Database. https://nvd.nist.gov/vuln/categories

14. *CWE (Common Weakness Enumeration) and the CWE Top 25 Explained.* HackerOne. https://www.hackerone.com/vulnerability-management/cwe-common-weakness-enumeration-and-cwe-top-25-explained
15. *CWE — database*. Security Database. https://www.security-database.com/cwe.php