УДК 004.056.55:004.738.5

А.А. Мурсалова

АНАЛІЗ ПІДХОДІВ ДО ЗАХИСТУ ВЕБ-ДОДАТКІВ НА ОСНОВІ ASP.NET MVC

Вінницький національний технічний університет

Анотація

У роботі розглянуто питання забезпечення безпеки веб-додатків, створених на платформі ASP.NET MVC. Описано основні загрози, з якими стикаються сучасні веб-системи, та проаналізовано підходи до їх усунення. Показано, що ефективний захист можливий лише за умови комплексного використання різних механізмів безпеки.

Ключові слова: веб-додатки, захист інформації, ASP.NET MVC, автентифікація, вразливості.

Abstract

The paper considers security issues of web applications developed using ASP.NET MVC. The main threats to modern web systems are described and approaches to their mitigation are analyzed. It is shown that effective protection requires a comprehensive use of different security mechanisms.

Keywords: web applications, security, ASP.NET MVC, authentication, vulnerabilities.

Вступ

Сьогодні практично кожен цифровий сервіс починаючи освітніми платформами та закінчуючи державними та бізнес-системами функціонують як веб-додаток. Оскільки їхні можливості постійно розширюються, пропорційно зростає й кількість загроз для даних користувачів та стабільності сервісів. Як показує досвід, навіть незначні прорахунки на етапі написання коду здатні створити критичні вразливості.

Фреймворк ASP.NET MVC є одним із найпопулярніших інструментів для створення таких систем завдяки зручній архітектурі. Розділення логіки, інтерфейсу та даних суттєво спрощує підтримку проекту в майбутньому. Проте сама по собі архітектура не гарантує захищеність – реальний рівень безпеки повністю залежить від рішень, які приймає розробник при написанні коду.

Метою цієї роботи є аналіз наявних підходів до захисту веб-додатків на базі ASP.NET MVC та пошук ефективних рішень для запобігання кібератакам. Акцент зроблено на практичних інструментах безпеки, які доцільно впроваджувати безпосередньо під час розробки сучасних систем.

Результати дослідження

Сьогодні веб-додатки використовуються практично в усіх сферах: від навчання до банківських сервісів. Через це вони стають привабливою ціллю для зловмисників. Найчастіше проблеми виникають не через складні атаки, а через базові помилки в розробці, наприклад, недостатню перевірку введених даних або неправильну роботу з доступом до системи.

Платформа ASP.NET MVC є зручною для створення веб-додатків, оскільки дозволяє розділити логіку, інтерфейс і обробку даних. Проте сама по собі вона не гарантує безпеку, якщо розробник не враховує основні принципи захисту [2]. На практиці часто зустрічаються такі вразливості, як SQL-ін’єкції, XSS та CSRF-атаки. Вони входять до числа найпоширеніших проблем веб-безпеки [1].

Для зменшення ризиків необхідно підйти до вирішення комплексно. Таким чином захист має відбуватися не в одному місці, а на всіх рівнях додатку. Наприклад, на стороні клієнта варто обмежувати неправильне введення даних, а на сервері – обов’язково перевіряти всі отримані значення. Особливо важливо не довіряти даним, які приходять від користувача, навіть якщо вони виглядають правильними.

Окрему увагу слід приділити роботі з базою даних. Використання параметризованих запитів або ORM-засобів дозволяє уникнути SQL-ін’єкцій, які можуть призвести до витоку інформації або її зміни [3]. Також важливо правильно налаштовувати права доступу до бази даних, щоб навіть у разі атаки шкода була мінімальною.

Ще одним важливим елементом є автентифікація та авторизація користувачів. У сучасних веб-додатках часто застосовуються токени доступу, які дозволяють безпечно передавати інформацію між

клієнтом і сервером. Крім того, варто обмежувати доступ до різних функцій залежно від ролі користувача. Це допомагає уникнути ситуацій, коли користувач отримує більше прав, ніж повинен [2].

Не менш важливою є загальна "гігієна" безпеки: використання HTTPS, оновлення бібліотек, перевірка коду та тестування додатку на наявність вразливостей. Багато рекомендацій щодо цього наведено у спеціалізованих джерелах, які описують типові помилки та способи їх виправлення [4].

Таблиця 1 – Аналіз підходів до захисту веб-додатків

Механізм / Підхід	**Об'єкт захисту**	**Спосіб реалізації в ASP.NET MVC**	**Переваги**	**Недоліки / Обмеження**
Параметризація запитів (LINQ / Entity Framework)	База даних (захист від SQL-ін'єкцій)	Використання ORM замість динамічного SQL	Автоматичне екранування символів, висока швидкість розробки	Потребує контролю при написанні сирих SQL-запитів
Валідація даних (Клієнт + Сервер)	Вразливості типу XSS, некоректне введення	Атрибути [Required], [StringLength], AntiXssEncoder	Фільтрація загроз на ранньому етапі, зручність через DataAnnotations	Клієнтську валідацію легко обійти, серверна є обов'язковою
Антифальсифікаційні токени	Захист від CSRF-атак	Спільне використання атрибуту [ValidateAntiForgeryToken] та @Html.AntiForgeryToken()	Надійне блокування підробки запитів з інших ресурсів	Працює лише для запитів типу POST/PUT/DELETE
Рольова авторизація (Identity)	Контроль доступу та прав користувачів	Атрибут [Authorize(Roles = "...")]	Чітке розмежування прав, інтеграція «з коробки»	Ускладнюється при гнучких (динамічних) системах прав
Шифрування трафіку (HTTPS)	Дані в каналі зв'язку (перехоплення, MITM)	Глобальний фільтр RequireHttpsAttribute, налаштування HSTS	Повний захист даних під час передачі між клієнтом і сервером	Потребує правильного керування SSL/TLS сертифікатами

Таким чином, захист веб-додатку – це не окрема функція, а постійний процес, який супроводжує розробку. Чим раніше враховуються питання безпеки, тим менше проблем виникає в майбутньому.

Висновки

Проаналізовано основні механізми безпеки та особливості їх застосування в ASP.NET MVC. У ході роботи було виділено ключові загрози для вебсистем, серед яких SQL-ін'єкції, XSS та CSRF-атакию. Вони найчастіше з'являються через прорахунки при написанні коду. Надійний захист можна побудувати тільки комплексно — поєднуючи валідацію даних, налаштування автентифікації та актуальні інструменти безпеки. Саме такий підхід допомагає суттєво підвищити захищеність веб-додатків.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. OWASP Top 10 – Web Application Security Risks. OWASP Foundation. 2026. URL: https://owasp.org (дата звернення: 10.06.2026).
2. ASP.NET MVC Security. Microsoft Docs. 2026. URL: https://learn.microsoft.com (дата звернення: 10.06.2026).
3. Halfond W. G. J. SQL Injection Attacks and Defense. Elsevier, 2010. 410 p.
4. Stuttard D., Pinto M. The Web Application Hacker's Handbook. Wiley, 2011. 912 p.

Мурсалова Анастасія Андріївна – студентка групи 1КІТС-24б кафедри менеджменту і безпеки інформаційних систем факультет менеджмент і інформаційна безпека, Вінницький національний технічний університет, Вінниця, anastasia.mursalovaa@gmail.com

Науковий керівник**: *Зоря Ірина Сергіївна* –** асистент кафедри менеджменту та безпеки інформаційних систем, Вінницький національний технічний університет, м. Вінниця, Україна, e-mail: ira.zoria@vntu.edu.ua

Anastasiia A. Mursalova – Faculty of management and information security, Vinnytsia National Technical University, Vinnytsia, anastasia.mursalovaa@gmail.com

Supervisor. ***Iryna S. Zoria*** – assistant of the Department of Management and Information Systems Security, Vinnytsia National Technical University, Vinnytsia, Ukraine, e-mail: ira.zoria@vntu.edu.ua