**УДУ імені Михайла Драгоманова**

Факультету математики, інформатики та фізики

*Кафедра комп’ютерної та програмної інженерії*

ЛАБОРАТОРНА РОБОТА №10

з курсу

***«Тестування програмного забезпечення»***

**Тема:" Тестування безпеки"**

Назаров Дмитро

Група   31 КН

Факультет математики, інформатики та фізики

Викладач: Кархут В.Я.

Київ 2025

**1. . Завантажте ZAP сканнер https://www.zaproxy.org/download/ та роскануйте з його допомогою** [**https://automationexercise.com/**](https://automationexercise.com/)**. Згенеруйте репорт та опишіть основні типи знайдених вразливостей та потенційні способи їх запобігання.**

В ході перевірки було виявлено низку критичних недоліків, серед яких особливу увагу привернуло використання застарілих і вразливих JavaScript-бібліотек. Зокрема, йдеться про застарілі версії файлів jquery.prettyPhoto.js та bootstrap.min.js. Наявність відомих вразливостей у цих компонентах створює потенційну небезпеку для вебзастосунку. Щоб мінімізувати ризики, необхідно регулярно оновлювати сторонні бібліотеки, орієнтуючись на офіційні релізи та довірені джерела.

Ще однією суттєвою проблемою є відсутність заголовка Content Security Policy (CSP) у відповідях сервера. Невпровадження цієї політики значно підвищує ймовірність атак типу Cross-Site Scripting (XSS), що може призвести до крадіжки особистих даних користувачів або компрометації їхніх облікових записів. Рекомендовано налаштувати CSP таким чином, щоб дозволити завантаження ресурсів лише із перевірених джерел та заблокувати небезпечний контент.

Крім того, виявлено відсутність захисту форм автентифікації від CSRF-атак (Cross-Site Request Forgery). Це дозволяє зловмисникам здійснювати небажані дії від імені користувача. Щоб уникнути подібних загроз, слід впроваджувати унікальні CSRF-токени у всі важливі форми та перевіряти їх на сервері.

Також виявлено відсутність заголовка HTTP Strict Transport Security (HSTS), який забезпечує використання виключно захищених з’єднань HTTPS, запобігаючи атакам типу "людина посередині" (MITM). Для посилення безпеки слід налаштувати HSTS з відповідним терміном дії.

Окремо зазначено проблему розкриття зайвої технічної інформації через заголовок X-Powered-By у відповідях сервера. Наявність таких даних може спростити зловмисникам підготовку атак. Рекомендується або видалити цей заголовок, або налаштувати сервер так, щоб він не передавав його.

Сканування також виявило, що деякі cookie-файли не мають атрибутів HttpOnly та Secure. Відсутність цих атрибутів підвищує ризики викрадення сесій через XSS-атаки та передачу конфіденційних даних через незахищені з'єднання. Всі важливі cookie мають бути захищені цими параметрами.

Ще одним виявленим недоліком стала некоректна конфігурація кешування конфіденційного контенту. Відсутність директив Cache-Control сприяє можливості небажаного зберігання чутливої інформації у проміжних кешах. Рекомендується налаштувати політику кешування для уникнення подібних ризиків.

Також у процесі аналізу виявлено потенційні XSS-вразливості через неправильну обробку даних у HTML-атрибутах. Це створює можливості для виконання шкідливого коду в браузерах користувачів. Щоб цього уникнути, слід ретельно перевіряти та очищати всі вхідні дані.

На завершення, сканування виявило наявність внутрішніх повідомлень про помилки та коментарів у коді JavaScript-файлів. Вони можуть містити важливу інформацію про внутрішню логіку роботи застосунку, тому їх необхідно видаляти перед випуском продукту в експлуатацію.

У підсумку, результати перевірки свідчать про необхідність всебічного удосконалення безпеки вебресурсу. Впровадження зазначених рекомендацій допоможе суттєво знизити ризик атак та забезпечити надійний захист даних користувачів.

**2. З допомогою https://securityheaders.com/ проскануйте https://automationexercise.com/ та визначте які загаловки доцільно змінити\додати щоб покращити безпеку додатку.**

Під час перевірки безпеки сайту <https://automationexercise.com/> було зафіксовано рейтинг "C" за даними сервісу SecurityHeaders. Така оцінка свідчить про те, що основні механізми захисту частково впроваджені, однак багато критично важливих елементів або відсутні, або налаштовані недостатньо ефективно.

Одним із найбільших недоліків є відсутність заголовка Strict-Transport-Security. Його роль полягає в примусовому використанні HTTPS-з'єднання навіть у випадках ручного введення адреси без префіксу https. Налаштування цього заголовка дозволяє мінімізувати ризик атак "людина посередині" (MITM) шляхом інформування браузера про необхідність використовувати лише захищене з'єднання протягом тривалого періоду для всього домену та його піддоменів.

Ще одна серйозна прогалина — відсутність політики безпеки контенту (Content-Security-Policy, CSP). Без неї браузер не обмежує джерела завантаження скриптів, стилів та інших ресурсів, що значно підвищує ризик XSS-атак. Правильна конфігурація CSP дозволяє чітко визначити довірені джерела і забезпечити більш високий рівень захисту.

Також варто звернути увагу на відсутність заголовка Permissions-Policy. Цей заголовок визначає доступність функцій браузера, таких як використання камери, мікрофона або геолокації. За замовчуванням ці функції можуть бути доступні без обмежень, що становить загрозу приватності користувачів. Впровадження Permissions-Policy дозволить керувати цими ризиками.

Крім того, для додаткового зміцнення захисту слід впровадити заголовки Cross-Origin-Embedder-Policy, Cross-Origin-Opener-Policy та Cross-Origin-Resource-Policy. Вони сприяють більш надійній ізоляції ресурсів і допомагають запобігти загрозам, пов'язаним із обробкою контенту з інших доменів.

У відповідях сервера присутній заголовок X-Powered-By, який розкриває використання технології Phusion Passenger. Така інформація може бути використана зловмисниками для виявлення вразливостей. Для підвищення безпеки рекомендується приховати або повністю видалити цей заголовок.

Окрім цього, виявлено, що кукі-файл csrftoken не має атрибутів Secure та HttpOnly. Через це існує ризик, що cookie стане доступним для скриптів або буде передано через незахищений канал, що може призвести до викрадення сесій чи CSRF-атак. Для усунення цієї загрози необхідно забезпечити встановлення обох атрибутів.

Отже, для зміцнення безпеки сайту необхідно першочергово впровадити належні заголовки HTTP, підвищити захист cookie-файлів та обмежити доступ до технічної інформації про сервер. Ці заходи допоможуть мінімізувати можливості для атак і підвищити довіру користувачів та автоматизованих систем оцінки безпеки.