**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"**

Крайнов Дмитро Олексійович

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.**

Однією з найбільш критичних проблем виявилася наявність застарілих та вразливих JavaScript-бібліотек. У репорті зафіксовано використання вразливих версій jquery.prettyPhoto.js та bootstrap.min.js. Така ситуація створює загрозу через те, що відомі уразливості в цих бібліотеках можуть бути використані зловмисниками для атак на вебдодаток. Для зниження ризику необхідно регулярно оновлювати сторонні бібліотеки до їх актуальних версій, стежити за офіційними релізами та використовувати тільки перевірені джерела.

Ще однією серйозною проблемою є відсутність налаштованої політики безпеки вмісту (Content Security Policy, CSP). Відсутність цього заголовка в HTTP-відповідях сайту значно підвищує ризик атак типу Cross-Site Scripting (XSS), які можуть призводити до крадіжки даних користувачів або компрометації сесій. Для покращення безпеки рекомендується впровадити правильну CSP, яка дозволяє завантажувати контент тільки з довірених джерел і забороняє небезпечні типи вмісту.

Крім того, сканування виявило відсутність токенів захисту від CSRF-атак у формах автентифікації. CSRF (Cross-Site Request Forgery) дозволяє зловмиснику здійснювати дії від імені користувача без його відома, що може мати серйозні наслідки для безпеки облікових записів. Для запобігання таким атакам необхідно впроваджувати у всі критичні форми унікальні CSRF-токени та перевіряти їх на серверній стороні.

Ще одним виявленим недоліком є відсутність заголовка Strict-Transport-Security (HSTS). Цей заголовок примушує браузер користувача використовувати лише захищене з’єднання (HTTPS), що допомагає запобігти атакам типу "людина посередині" (MITM). Для підвищення захищеності слід налаштувати HSTS на сервері, задавши відповідний заголовок з тривалим терміном дії.

Сканування також виявило розголошення зайвої технічної інформації через заголовок X-Powered-By у відповідях сервера. Ця інформація може бути використана зловмисниками для планування цільових атак. Рекомендується видалити цей заголовок або налаштувати сервер таким чином, щоб він його не відправляв.

Крім того, були виявлені проблеми з безпекою cookie-файлів: деякі з них не мають атрибутів HttpOnly і Secure. Відсутність HttpOnly дозволяє скриптам на сторінці отримувати доступ до cookie, що підвищує ризик крадіжки сесій при XSS-атаках. Відсутність атрибуту Secure дозволяє передавати cookie через незахищене HTTP-з'єднання. Обидва атрибути повинні бути встановлені для чутливих cookie-файлів.

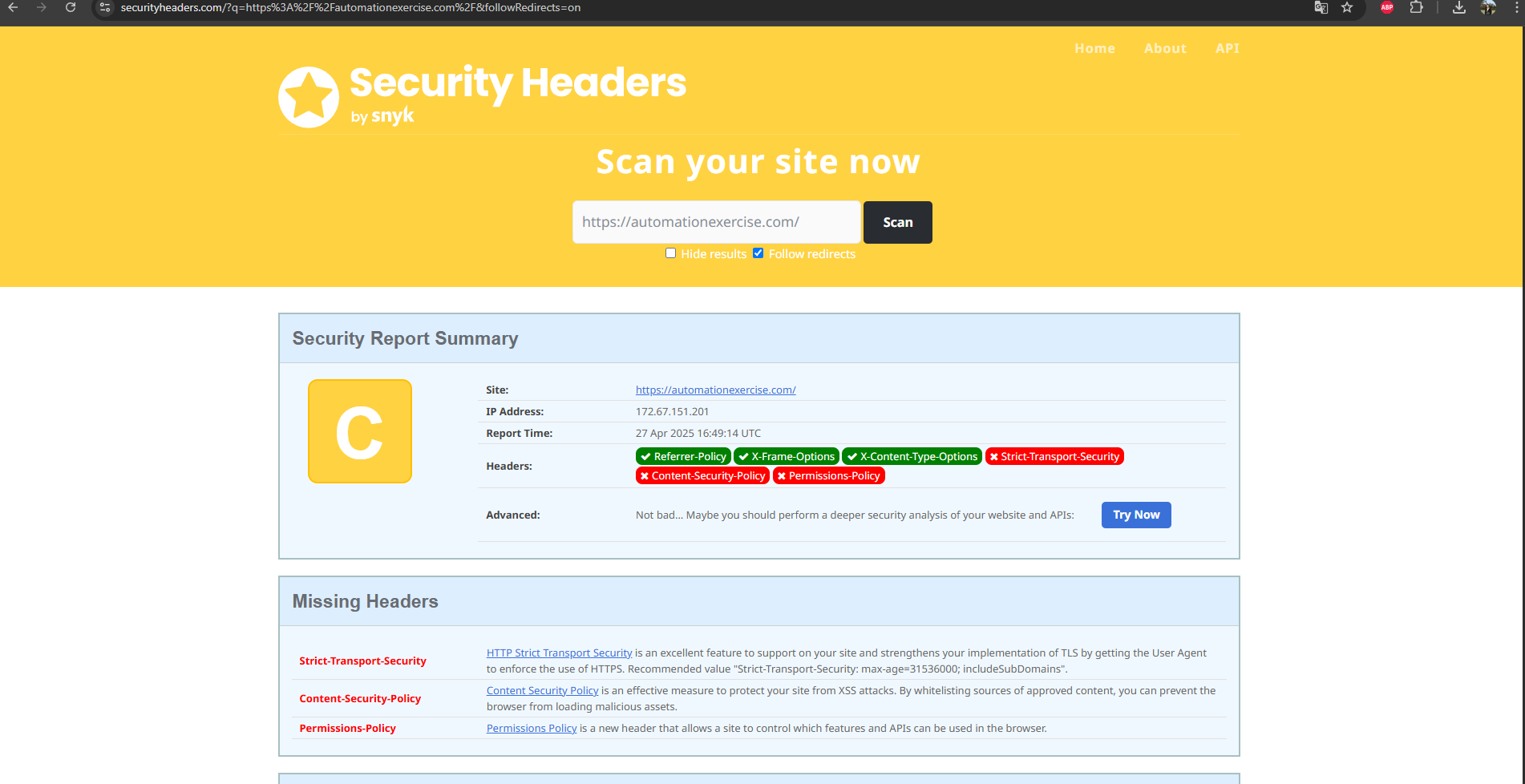
Ще однією проблемою є відсутність правильних налаштувань кешування для конфіденційного контенту. Відсутність директив Cache-Control може призводити до збереження конфіденційної інформації у проміжних кешах, що є небажаним явищем. Для уникнення подібних ризиків рекомендується налаштувати політику кешування так, щоб конфіденційна інформація не зберігалася на стороні клієнта або у проміжних проксі-серверах.

Окремо слід відзначити виявлені потенційні вразливості типу XSS через некоректну обробку даних у атрибутах HTML-елементів. Це може дозволити зловмиснику вставити шкідливий код, який буде виконаний у браузері користувача. Для запобігання таким вразливостям необхідно виконувати сувору валідацію та санітизацію всіх даних, які надходять від користувача.

На завершення варто зауважити, що під час сканування також було виявлено розкриття внутрішніх повідомлень про помилки та підозрілі коментарі у вихідному коді JavaScript-файлів. Вони можуть містити підказки про логіку роботи застосунку, тому їх необхідно прибирати перед розгортанням у продакшн-середовище.

Підсумовуючи, результати сканування вказують на потребу в комплексній роботі над підвищенням безпеки вебдодатку. Застосування вищезазначених заходів допоможе істотно знизити ризики атак і 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.**

****

Сайт <https://automationexercise.com/> на момент перевірки отримав оцінку безпеки "C" за версією сервісу SecurityHeaders. Це означає, що базові заголовки безпеки частково впроваджені, але багато важливих елементів залишаються або відсутніми, або налаштовані недостатньо повно.

У першу чергу виявлено відсутність заголовка Strict-Transport-Security. Він необхідний для того, щоб примусити браузери користувачів автоматично використовувати захищене HTTPS-з'єднання навіть у випадках, коли користувач вручну набирає адресу без https. Наявність цього заголовка дозволяє зменшити ризик атак типу "людина посередині", тому його варто додати з параметрами, які вказують браузеру зберігати цю політику протягом тривалого часу для всього домену і всіх піддоменів.

Також відсутній заголовок Content-Security-Policy, який є ключовим для запобігання атакам через впровадження стороннього коду, зокрема XSS-атак. Без цього заголовка браузер дозволяє завантажувати скрипти, стилі та інші ресурси з будь-яких джерел, що відкриває можливості для атак. Впровадження правильної політики CSP дозволяє обмежити завантаження контенту лише довіреними джерелами і значно підвищити безпеку сайту.

Ще однією важливою недоробкою є відсутність заголовка Permissions-Policy. Він дозволяє адміністратору сайту визначати, які функції браузера, такі як доступ до камери, мікрофона або геолокації, будуть доступні сторінці. Без цього заголовка браузер може надавати доступ до цих функцій за умовчанням, що може бути небажаним з міркувань безпеки та конфіденційності користувачів.

Крім трьох основних заголовків, які наразі відсутні, варто також розглянути додавання ще трьох сучасних заголовків для захисту взаємодії між доменами. Мова йде про Cross-Origin-Embedder-Policy, Cross-Origin-Opener-Policy та Cross-Origin-Resource-Policy. Вони допомагають ізоляції ресурсів і зменшують ризики атак через перехресне завантаження ресурсів.

У заголовках HTTP-відповідей також видно присутність заголовка X-Powered-By з вказанням технології серверу — Phusion Passenger. Наявність цього заголовка розкриває зайву технічну інформацію про технології, що використовуються на сервері, і може бути використана зловмисниками для пошуку відомих вразливостей. Щоб зменшити площу потенційної атаки, рекомендується видалити цей заголовок або замаскувати його.

Додатково було виявлено, що кукі csrftoken встановлено без прапорців Secure та HttpOnly. Це означає, що даний кукі може бути доступним для скриптів на сторінці та може бути переданий через незахищене HTTP-з'єднання. Відсутність цих прапорців створює ризики викрадення сесій або CSRF-атак. Щоб підвищити захищеність, необхідно забезпечити правильне встановлення атрибутів Secure та HttpOnly для всіх чутливих кукі-файлів.

Таким чином, для істотного підвищення рівня безпеки сайту необхідно в першу чергу впровадити або скоригувати налаштування заголовків HTTP, посилити захист кукі-файлів та обмежити розкриття інформації про сервер. Це допоможе зменшити поверхню для атак і підвищить довіру до ресурсу з боку користувачів та автоматичних систем перевірки безпеки.