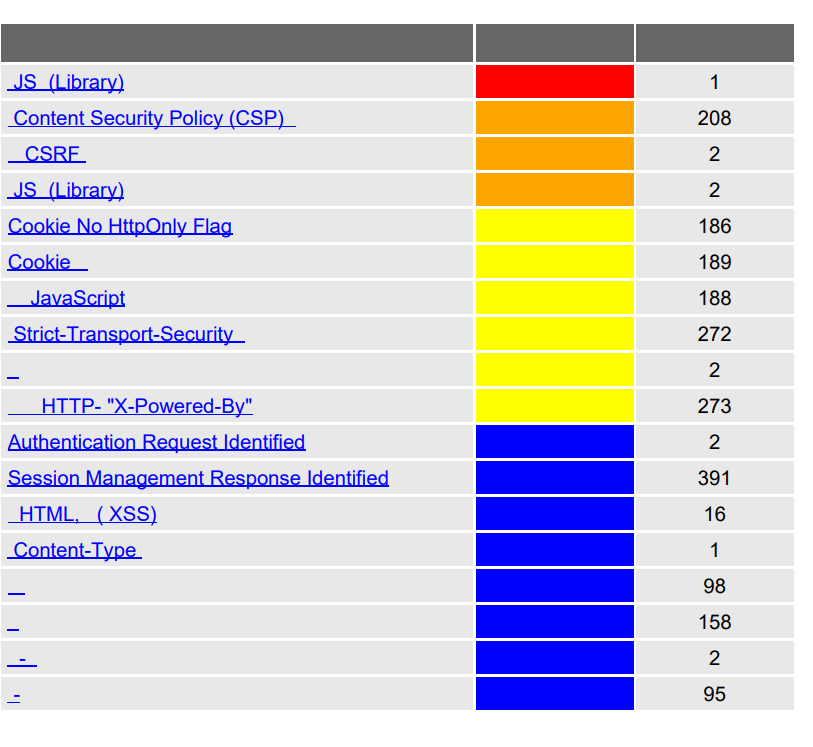
1. Під час перевірки було знайдено:

* відсутність або неправильне налаштування Content Security Policy (CSP) — це спеціальні правила, які керують тим, які скрипти можна запускати на сайті. Якщо цих правил немає, зловмисник може вставити свій шкідливий код (наприклад, JavaScript), який буде виконуватись у браузері користувача. Щоб захиститися, потрібно встановити чітку політику безпеки і заборонити підключення скриптів з ненадійних джерел.
* Cross-Site Request Forgery (CSRF). Зловмисник може відправити запит на сайт від імені користувача без його згоди. Щоб цього уникнути, потрібно додавати спеціальні захисні коди (токени) у всі форми, а також налаштувати файли cookie так, щоб вони не працювали з інших сайтів.
* Деякі JavaScript-бібліотеки на сайті є застарілими або можуть містити помилки безпеки. Такі бібліотеки використовуються для додавання функціональності на сайті. Щоб уникнути проблем, їх потрібно регулярно оновлювати і перевіряти на вразливості за допомогою спеціальних програм.
* Частина файлів cookie, які зберігаються у браузері, не мають прапора HttpOnly. Це означає, що до них може отримати доступ сторонній скрипт. Щоб цього не сталося, потрібно додавати цей прапор — тоді cookie будуть доступні тільки серверу, а не JavaScript-коду на сторінці.
* Не було знайдено правильного налаштування Strict-Transport-Security (HSTS) — це правило, яке змушує браузер використовувати тільки захищене з'єднання (HTTPS). Без цього з'єднання можуть бути перехоплені сторонніми особами. Щоб уникнути такого, потрібно вказати цей заголовок у відповіді сервера.
* Сервер показує зайву інформацію, зокрема заголовок X-Powered-By, який вказує, яка саме технологія використовується на сайті (наприклад, PHP чи Node.js). Ця інформація може допомогти зловмиснику обрати спосіб атаки. Рекомендується цей заголовок приховати.
* Слабке управління сесіями. Наприклад, сесії можуть не завершуватись після певного часу або не захищені від крадіжки. Щоб підвищити безпеку, потрібно встановити таймер на завершення сесії, використовувати складні й випадкові ідентифікатори сесій, а також позначати cookie як захищені.
* Було знайдено приклади XSS-атак (Cross-Site Scripting) — це коли зловмисник вставляє на сайт шкідливий код, який потім виконується у браузері користувача. Такі атаки можна попередити, якщо правильно обробляти (очищати) дані, які вводить користувач, і не виводити їх на сторінку "як є".
* Відсутність або неправильне встановлення заголовку Content-Type, який повідомляє браузеру, який тип інформації він отримує. Без цього браузер може неправильно розпізнати дані й допустити помилку або вразливість. Тому потрібно завжди вказувати правильний тип вмісту у відповіді сервера.



Загалом, ці вразливості показують, що сайт потребує кращого захисту. Для цього варто оновлювати сторонні бібліотеки, правильно налаштовувати безпеку сесій, cookie, заголовки, а також не дозволяти виконання стороннього коду на сторінках.

1. Під час перевірки заголовків безпеки сайту було виявлено:

* Відсутній заголовок Strict-Transport-Security (HSTS). Це спеціальний механізм, який змушує браузер завжди використовувати захищене з'єднання (HTTPS). Якщо цього заголовка немає, зловмисник може перехопити дані під час передачі. Щоб уникнути такого, варто додати цей заголовок на сервері з параметром, який забезпечує тривалу дію — наприклад, на один рік.
* Не налаштовано Content Security Policy (CSP) — політику, яка вказує браузеру, з яких джерел дозволено завантажувати скрипти, стилі, зображення тощо. Без цієї політики сайт може бути вразливим до XSS-атак, коли шкідливий код виконується в браузері користувача. Щоб зменшити ризик, потрібно чітко вказати дозволені джерела контенту.
* Відсутній заголовок Permissions-Policy, який дозволяє обмежити доступ до певних функцій браузера, таких як камера, мікрофон, геолокація та інші. Якщо не вказати обмеження, зловмисник може спробувати скористатися цими функціями без згоди користувача. Рекомендується встановити політику, яка забороняє або чітко контролює доступ до таких можливостей.
* Кукі, що зберігається у браузері (csrftoken), не має важливих прапорів безпеки — HttpOnly і Secure. Без цих прапорів кукі можуть бути прочитані сторонніми скриптами або передані через незахищене з’єднання. Щоб покращити захист, потрібно зробити ці кукі доступними лише серверу й передавати їх тільки через HTTPS.
* Заголовок X-Powered-By показує, що сайт працює на основі Phusion Passenger. Це технічна інформація, яка не потрібна звичайному користувачу, але може допомогти хакеру вибрати правильний спосіб атаки. Щоб цього уникнути, краще повністю видалити цей заголовок.

