Секція 1

АНАЛІЗ СПОСОБІВ ЗАПОБІГАННЯ XSS-АТАК РЕАЛІЗОВАНИХ В ASP.NET CORE

Чучин В.В.
Національний аерокосмічний університет ім. М. Є. Жуковського «ХАІ»
Науковий керівник Пєвнєв В.Я.

Актуальність. Однією з найрозповсюдженіших атак, яка застосовувалася останнім часом для веб-застосунків є Cross-Site Scripting (XSS). Міжсайтові сценарії (XSS) — це вразливість безпеки, яка дає змогу зловмиснику розміщувати на веб-сторінках клієнтські скрипти (зазвичай JavaScript). Коли користувач завантажує уражені сторінки, запускаються сценарії зловмисника, що дозволяє зловмиснику вкрасти файли cookie та маркери сеансу, змінити вміст веб-сторінки за допомогою маніпуляції DOM або перенаправити браузер на іншу сторінку. Уразливості XSS зазвичай виникають, коли програма приймає дані користувача та виводить їх на сторінку без перевірки, кодування чи екранування. Недоліки, які дозволяють цим атакам досягати успіху, є досить поширеними і виникають скрізь, де веб-додаток використовує вхідні дані від користувача в межах результату, який він генерує, не перевіряючи чи не кодуючи його [1].

Мета. Провести аналіз можливих сценаріїв запобіганню XSS-атак.

Основні положення. Щоб можна було вважати успішною, зловмиснику необхідно вставити та виконати шкідливий вміст на веб-сторінці. Тому Для захисту від XSS атаки необхідно щоб кожна змінна у веб-додатку проходила перевірку, а потім була вилучені або очищена. Фреймворки дозволяють легко переконатися, що змінні правильно перевірені, екрановані або очищені. Вихідне кодування та очищення HTML допомагають усунути ці прогалини [2].

Движок Razor, що використовується в ASP.NET Core MVC, автоматично кодує весь вихід. Він використовує правила кодування атрибутів HTML щоразу, коли використовується директива @. Оскільки кодування атрибутів HTML є наднабором кодування HTML, це означає, що при розробці не потрібно турбуватися про те, яке саме кодування необхідно використовувати. Іноді користувачам потрібно створити свою власну HTML-розмітку, що в тому числі передбачає написання скриптів, і одним із способів задовільнити цю потреби є використання візуального редактору WYSIWYG. Цей спосіб допоможе запобігти появі XSS вразливостей, але

порушить передбачувану функціональність програми. У цих випадках слід використовувати HTML Sanitization [3]. За замовчуванням ASP.NET Core не має бібліотеки для роботи з HTML Sanitization, але існує декілька HTML-парсерів, таких як HtmlSanitizer та HtmlAgilityPack, написаних на мові програмування C#, які можна використовувати для очищення HTML від потенційно шкідливих конструкцій.

Окрім вставки скриптів у розмітку веб-сторінки, XSS атака може також бути реалізована шляхом вставки шкідливих скриптів у URL. Перевірка HTTP-запитів додає захист від розмітки або коду в рядку запиту URL-адреси, файли cookie або опубліковані значення форми, які могли бути додані зі зловмисними цілями. Ця перевірка допомагає запобігти XSS-атаці, викликаючи в браузері помилку «потенційно небезпечне значення виявлено» та зупиняючи обробку сторінки, якщо вона виявляє введення, яке може бути зловмисним, як-от розмітка або код у запиті. Але варто зазначити, що ASP.NET Web API не використовує функцію перевірки запитів для очищення введених користувачами даних за замовчуванням, як це є в ASP.NET MVC, тому розробник має додати цю перевірку самостійно [4].

Висновки. У доповіді наведені можливі підходи щодо протидії XSS атакам при використанні технології ASP.NET.

Список літератури

1. Prevent Cross-Site Scripting (XSS) in ASP.NET Core. *Microsoft*. URL – https://docs.microsoft.com/en-us/aspnet/core/security/cross-site-scripting?view=aspnetcore-6.0 (дата звернення: 24.06.2022);
2. Cross Site Scripting Prevention Cheat Sheet. *OWASP*. URL – https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html (дата звернення: 24.06.2022);
3. Cross Site Scripting Prevention Cheat Sheet. *OWASP*. URL – https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html (дата звернення: 24.06.2022);
4. ASP.NET Request Validation. *OWASP*. URL – https://owasp.org/www-community/ASP-NET_Request_Validation (дата звернення: 24.06.2022).

Відомості про авторів

Чучин Віталій Вадимович, студент кафедри комп'ютерних систем, мереж і кібербезпеки, м.т. 099-093-55-43, v.chuchin@student.csn.khai.edu
Пєвнєв Володимир Яковлевич, професор кафедри комп'ютерних систем, мереж і кібербезпеки, д.т.н., доцент, v.pevnev@csn.khai.edu