УДК 303.7
Дьячков В.В.[1], Зайко Т. А.[2]
[1]студ. гр. КНТ-117 НУ «Запорізька Політехніка»
[2] канд. техн. наук, доц. НУ «Запорізька Політехніка»

БЕЗПЕКА ВЕБ-ДОДАТКІВ. МІЖСАЙТОВИЙ СКРИПТИНГ (XSS) ТА МЕТОДИ БОРОТЬБИ З НИМ

Сьогодні є нормальним, коли через Інтернет люди ведуть приватне листування, оформляють банківські перекази чи зберігають у “хмарі” персональні дані. Більшість користувачів вже звикли довіряти веб-сервісам. Тим не менш, в мережі Інтернет ще є багато сервісів, що є “потенційно небезпечними” для користувачів – за дослідженням компанії “Positive Technologies” [1]. Дві третини перевірених ними сайтів виявились такими, що, за їх словами, мають критичні уразливості.

Однією з найпоширеніших вразливостей веб-додатків є міжсайтовий скриптинг (англ. “Cross-Site Scripting” або XSS). XSS атаки направлені на впровадження чужорідного для системи коду з метою отримання доступу до персональних даних користувачів або конфіденційної інформації у системі [2].

XSS виникає, коли на сторінки, які були згенеровані сервером, з якоїсь причини потрапляють користувацькі скрипти (сценарії). Нижче наведено невеликий список основних методів боротьби з міжсайтовим скриптингом, рекомендовані “Open Web Application Security Project (OWASP)” [3,4]:

– валідація всіх вхідних даних, наданих клієнтською стороною, на стороні серверу. Розробник має програмно передбачати ймовірність потрапляння чужорідного коду у систему через надані текстові дані зі сторони клієнту. Передбачаючи, які дані необхідно ввести користувачу і зменшуючи область допустимих значень (використовуючи, як приклад, відповідні значення аргументу “type” HTML-тегу “input” для різних типів даних), можна вже на етапі валідації даних (як на стороні клієнту, так і на стороні серверу) захистити власний веб-застосунок від XSS атак;

– екранування всіх вхідних даних. Навіть якщо код зловмисників потрапив на сервер, екранування тексту при його вбудовуванні в HTML-сторінку захистить систему від виконання небажаного коду;

– використання HTTP заголовку “X-Content-Security-Policy” (з англ. “Політика захисту контенту”). За допомогою цього заголовку можна вказати браузеру, які джерела коду (а також стилей, шрифтів, медіафайлів тощо) є надійними, а які є небезпечними;

– вказання кодування для всіх полів вводу на сторінці, наприклад, ISO-8859-1 або UTF-8;

– забезпечення захисту передачі і збереження Cookies, що реалізується за допомогою обмеження допустимих доменів для передачі та прийому Cookies, використання параметру HttpOnly, TLS та інш.

– визначення HTTP заголовку “Access-Control-Allow-Origin”;

– використання названих вище порад може захистити розроблюваний вами веб-додаток від XSS атак, але все одно не дає стовідсоткової гарантії, оскільки результат залежить від того, наскільки вдалими є методи валідації, наскільки багато можливостей потрапляння небезпечних даних на сервер передбачає сам розробник.

Для найбільш ефективного захисту Вашого додатку, особисто я рекомендую використовувати найактуальніші методології і інструмент для розробки веб-застосунків: відмовитися від генерації сторінок на стороні серверу та реалізовувати взаємодії клієнтської та серверної частини через програмний інтерфейс (API), проводити екранування усіх текстових даних, що потрапляють на сервер, надавати користувачеві альтернативні варіанти розмітки та форматування тексту, відмінні від HTML (наприклад, BB-code, Markdown тощо), використовувати фреймворки для розробки, що мають захист від XSS атак за замовчуванням (наприклад, Django, Ruby on Rails, ASP.NET і інші), а також використовувати HTTP заголовок відповіді “X-Content-Security-Policy”, що наразі підтримується більшістю сучасних веб-браузерів.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Міжсайтовий скриптинг: Вікіпедія [Електронний режим] – Режим доступу: https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B6%D1%81%D0%B0%D0%B9%D1%82%D0%BE%D0%B2%D1%8B%D0%B9_%D1%81%D0%BA%D1%80%D0%B8%D0%BF%D1%82%D0%B8%D0%BD%D0%B3

2. Веб-вразливості. Неймовірне – очевидно: Хабр [Електронний ресурс] – Режим доступу: https://habr.com/ru/company/pt/blog/145329/

3. Безпека в PHP (частина 3): Хабр [Електронний ресурс] – Режим доступу: https://habr.com/ru/company/mailru/blog/352442/

4. A7-Cross-Site Scripting (XSS): OWASP [Електронний ресурс]. – Режим доступу: https://owasp.org/www-project-top-ten/OWASP_Top_Ten_2017/Top_10-2017_A7-Cross-Site_Scripting_(XSS)