УДК 004.056

Л. М. Куперштейн
О. С. Печенюк

ЗАХИСТ ВЕБ-ДОДАТКУ ВІД XSS АТАК

Вінницький національний технічний університет

Анотація
Розглянуто основні види XSS атак та проведено дослідження можливих шляхів захисту веб-додатку від них.
Ключові слова: *шкідливий код, веб-додаток, JavaScript, міжсайтовий скриптинг, JQuery.*

Abstract
Main types of XSS attacks are considered and researches of possible ways of web-application defense against them are conducted.
Keywords: *malicious code, web-application, JavaScript, Cross Site Scripting, JQuery.*

Сьогодні веб-додатки потребують захисту від різноманітних атак. Більшість користувачів використовують мови програмування клієнтської сторони, такі як JavaScript, але розповсюдження такого підходу також викликає проблему вразливості додатку до атак типу XSS (Cross Site Scripting).

Метою міжсайтового скриптингу може бути крадіжка cookies користувача за допомогою вбудованого на сервери скрипта з подальшою вибіркою необхідних даних та використання їх для подальших атак та злому, вилучення даних із форм для передачі персональних даних користувача зловмиснику, або ж для проведення DDoS-атаки . Зловмисник здійснює атаку не на пряму, а за допомогою вразливостей веб-сайту впроваджуючи свій спеціальний Java Script код. У користувачів цей код відображається як частина сайту.

Чіткої класифікації для міжсайтового скриптингу не існує, однак експертами по всьому світу прийнято виділяти три основних типи [1]:

- Постійний XSS. Один із найбільш небезпечних типів вразливостей, так як дозволяє зловмиснику отримати доступ до сервера і вже з нього керувати шкідливим кодом (видаляти, модифікувати). Кожного разу при зверненні до сайту виконуватиметься заздалегідь завантажений код, працюючий в автоматичному режимі. В основному таким вразливостям піддаються форуми, портали, блоги де присутня можливість коментування в HTML без обмежень. Сам шкідливий код може бути вбудований як в текст так і в картинки/рисунки.
- Непостійний XSS. В цьому випадку шкідливий скрипт виступає в ролі запиту жертви до зараженого веб-сайту. Цей принцип працює по наступній схемі:
    1) Зловмисник заздалегідь створює URL-посилання, яке буде містити шкідливий код та відправляє його жертві.
    2) Вона направляє цей URL-запит на сайт (переходячи по посиланню)
    3) Сайт автоматично бере дані з шкідливого скрипта та підставляє у вигляді модифікованої URL-відповіді жертві.
    4) У результаті в браузері у жертви виконується даний код, а зловмисник отримує всі cookies користувача.
- на основі DOM-моделі. В цьому варіанті можливе використання як постійного так і непостійного XSS.

Для виявлення XSS вразливостей можна використовувати різноманітні спеціалізовані сервіси, які в автоматичному режимі проведуть сканування сторінки [2]. Хоча даний метод не дає повної гарантії успіху, тому рекомендовано перевіряти сторінки в ручному режимі та обов'язково вилучити всі вразливі до даного типу атак спецсимволи, в яких прописуються всі зарезервовані мовою html-запити та теги.

Серед методів боротьби з XSS можна виділити декілька основних [3]:

1) Якщо на сайті присутній користувацький вхід, то необхідно виконувати шифрування.
2) Якщо шифрування виконати не можливо по певним причинам, варто використовувати перевірку введення (валідацію). Вона зазвичай використовує білі списки, а не чорні. Наприклад для того, щоб скласти список усіх шкідливих протоколів необхідно просто внести у список усі безпечні протоколи і заборонити усе що відсутнє у ньому. Це забезпечить захист навіть при появі нових шкідливих протоколів.
3) Зашифрування HTML на стороні клієнта за допомогою JavaScript. Оскільки JavaScript не має в наявності API для зашифрування HTML необхідно створити його власноруч.
4) Безпечна обробка даних повинна виконуватися не лише на стороні веб-сервера а й на стороні клієнта
5) Використання JQuery. Найпоширеніша форма XSS в JQuery - це коли ви передаєте введення користувача селектору JQuery. Веб-розробники часто використовують location.hash і передають його селектору, що спричинить XSS, оскільки JQuery буде представляти HTML. jQuery розпізнав цю проблему і випрацював їх логіку вибору, щоб перевірити, чи починається введення з хеша. Тепер jQuery візуалізує HTML лише у тому випадку, якщо перший символ є <. Якщо ви передаєте недовірені дані селектору JQuery, переконайтесь, що ви правильно вимкніть значення за допомогою функції jsEscape, наведеної вище.
6) Використання PHP. У PHP є вбудована функція для шифрування сутностей, відомих як htmlentities. Необхідно викликати цю функцію, щоб уникнути введення даних у контексті HTML.
7) Використання CSP (Content Security Policy). CSP є останньою лінією захисту від міжсайтового скриптингу. Якщо захист від xss не спрацював по певним причинам використовується політика безпеки для пом’якшення xss, обмеживши можливості зловмисника. CSP дозволяє керувати різними речами, наприклад, чи можна завантажувати зовнішні сценарії та чи виконуватимуться вбудовані сценарії. Для розгортання CSP потрібно включити заголовок відповіді HTTP під назвою Content-Security-Policy зі значенням, що містить вашу політику.

На завершення варто зауважити що запобігання xss атакам є надзвичайно важливим для забезпечення цілісності даних веб-додатку. Основними з методів боротьби з даним типом атак є використання JQuery, PHP, CSP, шифрування HTML, перевірка введення. А отже варто використати дані методи при захисті власного веб-додатку.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Open web Application security project, XSS(cross site scripting).prevention cheat sheet,2011; http://www.owasp.org/index.php/Xss_(Cross_site_scripting))_preventation _cheat_Sheet
2. Защита от XSS [Електронний ресурс] – Режим доступу: URL: http://www.spy-soft.net/zashhita-ot-xss/ Назва з екрану.
3. How to prevent XSS [Електронний ресурс] – Режим доступу: URL: https://portswigger.net/web-security/cross-site-scripting/preventing Назва з екрану.

Печенюк Олександр Сергійович — студент групи 1БС-16б, факультет інформаційних технологій, Вінницький національний технічний університет, Вінниця, e-mail: pechenyuk_oleksandr@gmail.com.

Куперштейн Леонід Михайлович — доцент кафедри захисту інформації, Вінницький національний технічний університет.

Oleksandr S. Pecheniuk- student of 1BS-16b group, Faculty of Information Technologies, Vinnytsya National Technical University, Vinnytsia

Leonid M. Kuperstein - Associate Professor of Information Security Department, Vinnytsia National Technical University