УДК 621.384.3

Д. О. Жаворонок
Н. В. Сачанюк-Кавецька

УДОСКОНАЛЕННЯ МЕТОДУ СКАНУВАННЯ, ЯК МОЖЛИВОСТІ УНИКНЕННЯ XSS АТАК

Вінницький національний технічний університет

Анотація

В даній доповіді розглянуто можливість удосконалення методу сканування на предмет вразливості web-сайтів до атак типу XSS. Наведено схему архітектури сканеру на тип такої атаки та модифікований алгоритм його роботи

Ключові слова: сканер, атака, Web - сайт, несанкціонований доступ

Abstract

In this report discusses the possibility of improving the method of crawling web-sites for XSS attacks. The scheme of the scanner architecture for the type of such attack and the modifications algorithm of its operation are given

Keywords: scanner, attack, Web - site, unauthorized access

Web – сервери постійно піддаються безлічі різноманітних небезпек. Причому найсерйознішу загрозу становлять для них зловмисники і віруси. Зловмисники можуть отримати доступ до конфіденційної інформації, розміщеної на сервері, зламати сайти і підмінити їх вміст, а також вивести з ладу сервер за допомогою DDoS–атаки [1, 2]. Багато вірусів, особливо інтернет – хробаки, використовують для поширення вразливості в програмному забезпеченні. Переважна частина вразливих місць з'являється через програми – сервери, такі як Apache, Microsoft Internet Information Server та у комплексному поєднанні з базами даних.

Сьогодні на сайтах, присвячених інформаційній безпеці, постійно з'являються повідомлення про виявлення нових вразливостей в програмному забезпеченні Web – серверів. Виявити ці вразливості в скриптах можна за допомогою сканерів безпеки. Оскільки програмний продукт постійно змінюється, а види атак модифікуються та ускладнюються, то для безпеки додатку та користувачів кожен власник Web – сервера, повинен проводити періодичне сканування свого додатку [3, 4]. За версією звіту «Nine years of bugs & coordinated vulnerability disclosure: Trends, observations & recommendations for the future» («Дев'ять років помилок та координоване розкриття вразливостей: тенденції, спостереження та рекомендації на майбутнє» англ.), що був представлений від NCC Group (рис. 1), яка аналізує найпоширеніші чинні web – вразливості у додатках, саме різновиди XSS атак займають найбільш критичні позиції.

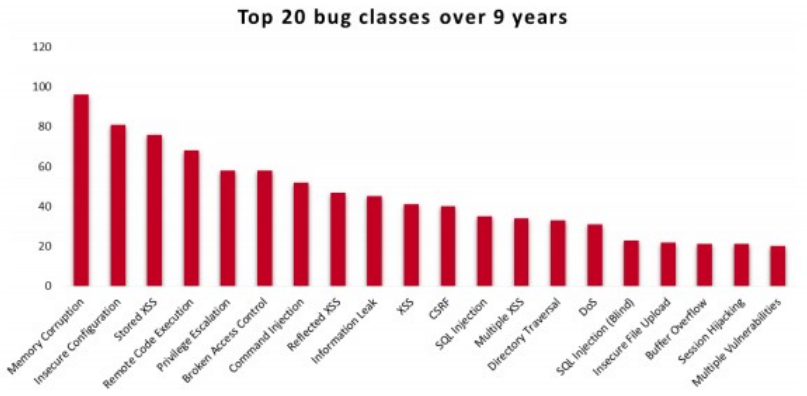

Рисунок 1 – Вигляд рейтингу вразливостей від NNC Group

XSS – це вразливість безпеки у Web – середовищі, яка дозволяє зловмиснику компрометувати взаємодію користувачів з додатком. Ці атаки можна здійснити за допомогою HTML, JavaScript, VBScript, ActiveX, Flash та інших мов клієнтської сторони. Можливість здійснення XSS атаки виникає, коли додаток приймає ненадійні дані і відправляє їх в браузер без належної перевірки або екранування. XSS дозволяє зловмисникам виконувати скрипти в браузері жертви, захоплювати сесії користувачів, спотворювати Web – сторінки або перенаправляти користувача на шкідливі сайти.

Загальний алгоритм роботи сканера можна розбити на такі кроки:

Крок 1. Введення URL – адреси.

Крок 2. Вибір типу вразливості, на наявність якої буде відбуватися сканування.

Крок 3. Запуск сканування.

Крок 4. Перевірка даних, які були виявленні під час сканування на відповідність умові.

Крок 5. Підготовка звіту відносно аналізу, описаному у кроці 4.

На базі загального алгоритму було розроблено алгоритм для виявлення певних вразливостей, таких як SQL ін'єкція, XSS. Предметом покращення та удосконалення було обрано метод знаходження вразливих місць до атак типу XSS (рис. 2) у Web – додатках.

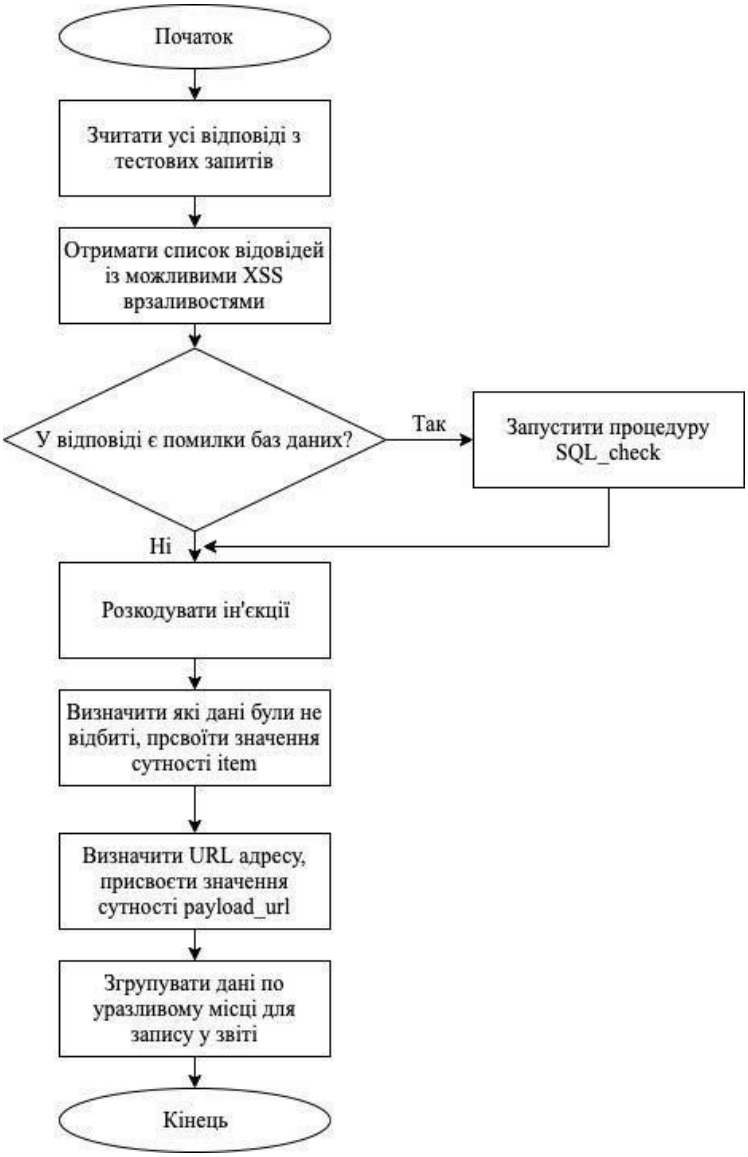

Рисунок 2 – Вигляд блок – схеми модифікацій методу сканування

Автоматизовані інструменти можуть знайти деякі вразливості XSS автоматично. Однак, кожен додаток будує сторінки по – різному і використовує різні інтерпретатори на стороні браузера, такі як JavaScript, ActiveX, Flash і Silverlight, що ускладнює автоматичне виявлення вразливостей. Найкращий варіант – комбінація декількох перевірок даних, які відбуваються у тілі відповіді запиту POST.

Для знаходження вразливостей для кожного посилання потрібно модифікувати запит та перевірити, чи присутній запит у чистому вигляді, чи в структурі додатку. Для кожного посилання буде додаватись payload. Він у свою чергу генеруватиметься наступним чином: payload = delim_str + self.test_str + delim_str + ';9'; де delim_str – це рандомна послідовність символів, а self.test_str = '\'"(){}<x>:/'.

Існує залежність між небезпечними символами XSS та небезпечними символами введення SQL ін'єкції, а саме одинарними та подвійними лапками. Завдяки виявленню помилок SQL у відповіді вірогідність знаходження вразливостей значно збільшується. При генерації "корисних даних" буде надсилатись додатковий запит із значенням "d'z"0 " у "тілі", що має викликати SQL помилку у відповідь. Також варто додати параметр, за яким можна буде визначити кількість одночасних з'єднань із сайтом в одиницю часу. Оскільки часто брандмауери та системи IDS можуть визначити сканування як атаку, та блокувати усі запити з IP-адресами сканера. Зменшуючи значення цього параметра, ймовірність забезпечення стабільної роботи сканеру буде збільшуватись.

Якщо сканер може знайти помилки і видати попередження, програміст повинен перевірити безпеку для цього сайту та звернути увагу на слабкі місця, які можуть призвести до несанкціонованого доступу до інформації злочинною особою. Модульну архітектуру сканеру на тип XSS атаки подано на рис. 3.

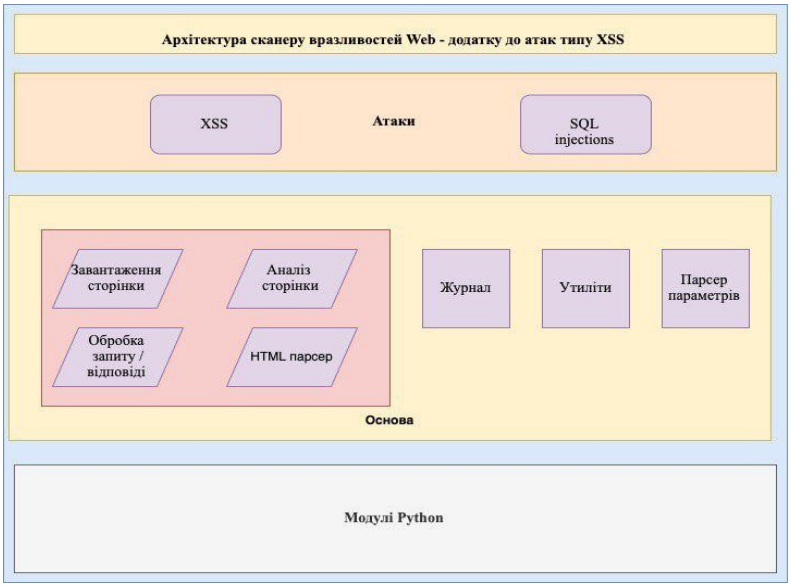

Рисунок 3 – Вигляд схеми архітектури сканеру

Робота сканеру буде розпочинатись із виділення випадкового агенту для кожного запиту. Рядок User–Agent (UA) міститься в заголовках HTTP і призначений для ідентифікації пристроїв, що запитують онлайн – вміст. Користувач-агент повідомляє серверу, який пристрій відвідує сторінку і ця інформація може бути використана для визначення вмісту для повернення. Кожен тип пристрою, включаючи телефони, планшети і ін., може мати свій власний UA, що дозволяє виявити цей пристрій з будь – якою метою. Для запобігання блокування запитів з одного пристрою, сканер буде надсилати їх кожен раз з нового агента. Після старту сканера та зчитуванні параметрів, сканер зчитуватиме адресу пошуку та спробує зчитати файл robots.txt переданого ресурсу. Robots.txt – це текстовий файл, який Web – майстри створюють, щоб спонукати Web – роботів (як правило, пошукових роботів), сканувати сторінки на своєму Web–сайті. На практиці файли robots.txt вказують, чи можуть певні користувацькі агенти (програмне забезпечення для Web – сканування) сканувати частини Web –сайту чи ні. Ці інструкції щодо сканування визначаються "забороною" або

"дозволом" поведінки певних (або всіх) користувачів–агентів. Такі пошукові системи, як Google, Bing, Yahoo тощо, мають періодичні розробки Web–сайтів, щоб збирати наявну та/або нову інформацію, наприклад Web–сторінки, статті блогу, зображення тощо. Після публікації цих ресурсів на Web – сайті, саме пошукові системи визначають, що буде індексуватися. Після цього система сканера намагатиметься зробити тестовий запит до сайту та перевірити чи наявний файл robots.txt. У разі успіху, алгоритм проіндексує robots.txt та створить запити для заборонених доменів. На кожен запит, який буде приходити від сервера, алгоритм буде аналізувати його на наявність XSS вразливостей у заголовку запиту або у адресі ресурсу. Після успішного сканування сайту створюється мапа усіх можливих посилань та місць введення даних, таких як поля пошуку, форми коментарів тощо. Через те, що пошук відбувається у реальному потоці інформації, є можливість, що одні і ті самі сторінки можуть повторюватись. Для фільтрації схожих випадків буде використана фільтрація на основі фільтрів Блума. Далі відбуватиметься процес створення запитів, які містять небезпечні елементи для вразливих місць у коді. Для цього генеруватиметься запит зі шкідливим змістом для кожного параметру введення окремо, а саме для адреси, iframe атрибутів, і т. д. Для створення корисного навантаження алгоритм буде використовувати унікальний параметр delim.

Delim – це послідовність одного або декількох символів для визначення межі між окремими незалежними регіонами у простому тексті чи інших потоках даних. Після чого генерується остаточний вигляд посилання, за яким буде надсилатись запит.

Наприклад, посилання example.com/page1.php?param=val перетворюється на example.com/page1.php/FUZZCHARS/?param=val. Як результат, сканер отримує відповіді на всі відправлені запити та після індексації починає перевірку на наявність XSS елементів у структурі відповіді або додатку. Якщо, після перевірки було знайдено вразливі місця, усі дані записуються у .txt файл, з повним описом виявленого місця.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Галатенко В. А. под ред. академика РАН В. Б. Бетелина Основы информационной безопасности: учебное пособие, 4-е изд. Москва: Интернет-Университет Информационных технологий; Бином. Лаборатория знаний, 2008. 205 с.
2. Німченко Т. В. Аналіз загроз персональним даним та засобів їх захисту // *Technology audit production reserves.* 2015 №2/5 (22). С. 63-67.
3. Ахрамович В. М. Ідентифікація й аутентифікація, керування доступом // *Сучас. захист інформації.* 2016. №4. С. 47-51.
4. Сканер уязвимостей XSpider7 [Електронний ресурс] Режим доступу: http://www.ixbt.com/soft/Xspider7.shtml#int.

Жаворонок Дарина Олексіївна – студент групи УБ-14б, факультет менеджменту та інформаційної безпеки, Вінницький національний технічний університет, м.Вінниця

Науковий керівник: ***Сачанюк-Кавецька Наталія Василівна –*** к.т.н., доцент, доцент каф. ВМ Вінницького національного технічного університету, м.Вінниця, e-mail: skn1901@gmail.com

Zhavoronok Daryna O. – student of UB-14b group, Department of Management and Information Security, Vinnytsia National Technical University, Vinnytsia

Supervisor: ***Sachaniuk-Kavets'ka Natalia V.*** – Candidate of Technical Sciences, Associate Professor, Associate Professor the department of Higher mathematics Vinnytsia National Technical University, Vinnytsia, e-mail: skn1901@gmail.com