Received: 28 June 2021 | Revised: 07 August 2021 | Accepted: 10 August 2021
DOI: 10.37917/ijeee.17.2.8 Early View | December 2021

Iraqi Journal for Electrical and Electronic Engineering
Original Article

Open Access

Server Side Method to Detect and Prevent Stored XSS Attack

Iman F. Khazal*, Mohammed A. Hussain
Department of Computer Science, Education College for Pure Science, University of Basrah, Basrah, Iraq

Correspondence
* Iman F. Khazal
Department of Computer Science,
Education College for Pure Science,
University of Basrah, Basrah, Iraq
Email: pgs2183@uobasrah.edu.iq

Abstract
Cross-Site Scripting (XSS) is one of the most common and dangerous attacks. The user is the target of an XSS attack, but the attacker gains access to the user by exploiting an XSS vulnerability in a web application as Bridge. There are three types of XSS attacks: Reflected, Stored, and Dom-based. This paper focuses on the Stored-XSS attack, which is the most dangerous of the three. In Stored-XSS, the attacker injects a malicious script into the web application and saves it in the website repository. The proposed method in this paper has been suggested to detect and prevent the Stored-XSS. The prevent Stored-XSS Server (PSS) was proposed as a server to test and sanitize the input to web applications before saving it in the database. Any user input must be checked to see if it contains a malicious script, and if so, the input must be sanitized and saved in the database instead of the harmful input. The PSS is tested using a vulnerable open-source web application and succeeds in detection by determining the harmful script within the input and prevent the attack by sterilized the input with an average time of 0.3 seconds.
KEYWORDS: Cross Site Scripting (XSS), Stored-XSS (persistent), Web application, Detecting XSS, Preventing XSS.

I. INTRODUCTION

Web applications are currently the best way to represent data and provide various services to users via the web. Banking or financial services, educational and news websites, and social media channels are among the services offered. Furthermore, the web application has become the primary means of gathering information on any topic. As a result, the use of web applications has increased, and it has become more appealing to hackers, not just users. This vast amount of sensitive data stored in web applications can be stolen by hackers for a variety of reasons, including monetary gain or spying [1]. The security issues are one of the main dangers that information technology faces and their application, Indicate the Measures Put in the place to maintain them information system capabilities and services from illegitimate access. malicious attack explore a computer and technology-based system companies [2]. The XSS (Cross-Site Script) attack is one of the most common security issues in web applications.

The injection attack, XSS, is one of the most common web application attacks. As a consequence of this attack, sensitive data, cookies, and sessions have been stolen. The injection attack is used, in which malicious scripts are injected into the web application's source code. This type of attack may occur in any web application that does not use the encryption method or verification of the validity of the input. Therefore, the attacker exploits a vulnerability in the application to launch XSS attacks by storing malicious scripts on the website or deceiving the user with the URL injected by malicious script [3].

This attack is aimed at the user rather than the application (the user is the victim). XSS attacks are considered one of the most dangerous approaches that exploit weaknesses in web applications and are ranked second of the most dangerous vulnerability and are considered critical with a rate of approximately 38%. What is concerning, however, is the low rate of solutions or treatment for this type of attack. Furthermore, according to the Open Web Application Security Project (OWASP) report, XSS attacks were ranked seventh out of ten major security risks, while XSS was ranked third in 2013. The severity of this attack is confirmed by Imperva data, which is associated with XSS attacks that exploited the greatest number of Web application vulnerabilities in 2017. Indeed, the number of vulnerabilities exploited by XSS has more than doubled since 2016. According to Imperva prediction, it will be one of the most common attacks in 2018 [4].

Cross-site script (XSS) attacks are carried out by injecting malicious code into various types of interpreters in the user's browser, such as JavaScript, Flash, ActiveX, HTML, VBScript, or any other client-side language. XSS is defined as an attack on a specific website's customers' privacy that