Detecting XSS Based Web Application Vulnerabilities

Abstract-

A Web Applications are developed using various technologies like HTML, JavaScript, XML, AJAX, etc. and are accessed by millions of users for various services. The vulnerabilities at the design level in these technologies lead to security breach, resulting in theft of the user’s credentials. It is the type of hacking techniques to attack the web application is Cross-site scripting (XSS). XSS is a computer security vulnerability found in web applications. XSS vulnerabilities exists denial of service, stealing of cookies, session tokens, and other user sensitive data. A Cross-site scripting (XSS) targets web applications by embedding scripts in a web page that will get executed at client side or server-side and the attacker will manipulate the information in desired manner. Cross-site scripting (XSS) vulnerabilities in current browser’s session in web applications are discussed in this paper. The websites may contain malicious script codes that can be detected using detection tool at the client side. In proposed model, XSS-Check add-on will detect the XSS which is Persistent based on single /Response cycle. It determines the user input for the webpage returned with the result, and check for the WebPages with logon functionality, information that got encoded with the http headers and includes DOM parameter. Once identified, the validation is done across dynamic WebPages in both server and client side.

Keywords: Cross-site scripting (XSS), Web Application Vulnerabilities, Persistent XSS, XSS-Check add-on
I. INTRODUCTION

Cross-site Scripting (XSS) attacks is when an attacker by sending malicious code, in the form of a browser side script, to a different end users in a web applications. A web application are designed and coded for dynamic web pages to provide online services for organizations. Cross-site Scripting (XSS) attack involves three parties – the attacker, a client and the website. To obtain user information are associated with the website, namely username and password and other information. Cross-site scripting attacks use known vulnerabilities in web-based applications, servers, or the plug-in systems on which they rely. A hacker able to inject malicious codes in the dynamic websites and the code is executed in the web browser, it changes the web pages. The goal of XSS attack is to steal the client cookies or any other sensitive information which is used to authenticate the client to the website.

The web application vulnerability is harmful to application owner, application users and other entities. A website can prevent from XSS vulnerability by some service providers. The special characters include (“<”,”>”,”/”, etc) are identified and encoded by the output and need to be filtered as input in the web applications.

The websites may be vulnerable or secured, which is unaware of users to identify XSS attacks. Before accessing the websites, user can scan and detect the web application which can be used to identify secure or unsecure web application by the detection tool.

The various XSS Payloads are used to scan and detecting the website vulnerabilities lively. The payloads are included in the code and easy to detect some vulnerable websites. The vulnerable website contains web pages forms which can contain malicious JavaScript code. Using JavaScript code alert, the detection tool are used to detect the XSS vulnerabilities in the websites.

A web page is loaded in the browser by a client-side JavaScript code. Cross-site Scripting is a JavaScript based attacks because it includes JavaScript code on a user’s browser from a malicious web server which is executable.

The Problem of XSS vulnerabilities are User’s cookie stealing, web page modification etc. A stored or persistent XSS attacks is malicious JavaScript or code permanently stored in vulnerable database server.

A stored XSS attack in which a victim requests the stored information from the vulnerable server, injects the requested malicious script into the victim’s browser. The browser then executes the code or script because the vulnerable server is usually a known or trusted site. For instance, an attacker can post a message containing the malicious script to the message board, which stores and subsequently displays it to other users, causing the attacks.
II. PROBLEM STATEMENT

To scan and detecting the web application vulnerabilities and protect the website from web-based attacks, "XSS-Check" add-on are developed.

III. Objectives of XSS-Check add-on

The objective of this paper is to study the detection of XSS vulnerabilities in web applications and the solutions to prevent such attacks. The development of XSS-Check add-on is used for detecting Cross-site scripting (XSS) vulnerabilities to protect the website from web-based attacks. XSS-Check add-on is used to detect the types of XSS which include Non-persistent and Persistent based on single/Response cycle. It is used to determine if user input for the webpage returned with the result, and check for the WebPages with logon functionality, information that got encoded with the http headers and includes DOM parameter. Once identified, the validation is done across dynamic WebPages in both server and client side.

XSS-Check add-on User Interface is used to scan, detect and providing solutions to such type of XSS attacks in the current browser’s session. It counts the number of attacks in the website.

A “XSS-Check add-on” which opens as User Interface in the browser to detect the given websites. It can be used as security detection tool which detects malicious JavaScript code from the retrieved web pages. It uses various payloads as input for the website to form web pages.

The user input is URL to find all the links in the given website. It crawls all the web pages that are available in website. In this add-on, all the links are stored as report. It extracts all links present in the website and scans for web pages form field. When clicking on “Live Detect” in the user interface, it retrieves all the links associated with the given website.

In the next page of the user interface, it shows the number of XSS counts detected in the website. The ‘solution’ link provides the prevention for detected payloads for XSS vulnerabilities.
IV. Cross-site Scripting

The 2 types of XSS

- Persistent XSS
- Non-Persistent XSS

Persistent XSS

Persistent XSS attack is also known as a stored XSS or Type-I XSS attack. Persistent XSS attack involves injecting malicious script into a website, and those scripts are stored in vulnerable database.

Persistent XSS can be difficult to detect and considered more harmful than the other two attack types. A malicious script is rendered automatically, there is no need to target individual victims or lure to a third party website. An attacker can easily hide their activity; for example, in a blog could embed the script in a seemingly innocuous comment. The sensitive data are stored at risk by the visitor of the websites.

A stored XSS attack in which a victim requests the stored information from the vulnerable server, injects the requested malicious script into the victim’s browser. The browser then executes the code or script because the vulnerable server is usually a known or trusted site. For instance, an attacker can post a message containing the malicious script to the message board, which stores and subsequently displays it to other users, causing the attacks.

A stored XSS are stored in a vulnerable database and may be in a resource like file system. The victim gets the online message board as part of JavaScript code in the website.

Persistent XSS vector

```<SCRIPT>
</SCRIPT>```

The Persistent XSS vector starts with `<script>` and ends with `</script>`

In the evil.com, the file images.jpg cookie is theft is performed.

Non-Persistent XSS

A Reflected XSS is a type of XSS in which a page containing a malicious code that is reflected by the browser as a search result. The attack will target the website vulnerability that deals with dynamic property of web application. A user to visit a manipulated URL with embedded malicious code using social engineering by an attacker. A modified code in the URL is
to be executed by the web browser when user clicks on the malicious links.

V. IMPLEMENTATION

Flow Diagram

Flow Diagram of XSS-Check add-on User Interface

This diagram shows the user input is given as URL.

‘Live Detect’ is used to scan and detect the web application vulnerabilities in current browser’s sessions.

The label named ‘Number of Links’ displays number of XSS vulnerabilities present in current browser’s session for the given website.

The solutions for such attacks to prevent will be provided.

The label ‘Search any website’ is used to enter any website for scanning the links for the given website.

After giving website as input for instance http://www.google.com,

It checks all the links and sub-links for the given website.

Before searching, a variable check whether the given url has been passed or not as input.

Report is generated which contains extracted links for the given website.

The “solution” link which gives the prevention for respective XSS attacks.

User Interface

XSS-Check User interface for XSS is used to search and scan any given website, check for specific XSS and click the ‘Live Detect’ button to function the links in website for web application vulnerabilities in current browser’s session.
The number of links related to XSS attacks present in websites are counted, display the links and provides the solutions for such attacks to prevent.

The Report is generated as extraction of links for the given websites.

The label named ‘Number of Counts’ displays number of XSS vulnerabilities present in current browser’s session for the given website.

The “solution” link provides prevention of respective XSS vulnerabilities.

If the url has given correctly, the website checks the url of the links lively.

The process of short traversal is used to finding all the links that are associated with the given url.

The process of comprehensive traversal is used to finding all the sub-links for each link. It scans the given website and search all the links that are associated with url. The function ‘Live Detect’ is clicked to finding XSS for the website. It starts finding the XSS inside the url and the url starts with http://www.

Inside the website, it opens the link and find the forms that having payload. The page which is having payload able to read and scan the url related to vulnerable.

Otherwise, the page could not read in the url website.

The second payload is the alert to the page that visited by users.

If there is no alert, the requested page cannot be open to read in the url website.

For all the payloads, in the url website, the instance ‘could not able to read the page’ and it returns ‘No link found’ and it will exit the page.

For all the payloads, in the url website, the instance ‘could not able to read the page’ and it returns ‘No link found’ and it will exit the page.

The url starts with ‘http://www.’ is the given input to search and finding cross-site scripting.

It searches all the links and sub-links for the given website as input.
VI. CONCLUSIONS

In XSS Check User Interface is used to search, scan the websites for finding all the links and detecting web application vulnerabilities.

It extracts all the links and sub-links present in the website as Report. It counts the number of XSS links present in the website. It provides prevention for such attacks in “solution” link.

VII. REFERENCES


