A Survey on Securing Web Applications against Cross Site Scripting, Injection Attacks & Authentication Attacks

Susithra D.[1] Dr. T. Parameswaran[2]
[1]M.E. Student
[1,2]Department of Computer Science & Engineering
[1,2]Anna University Regional Campus, Coimbatore, Tamilnadu, India

Abstract— This paper focus on defending against web application attacks. Web applications have become one of the standard platforms for service releases and representing information and data over the World Wide Web. In this paper mainly focus on the specific problem of cross site scripting attacks against web applications. In this paper presents the various popular and common web applications attacks found over the internet such as Injection attacks, and session management attacks and cross site scripting attacks. It also proposes the future possibilities and feasible countermeasures against these attacks.

Key words: Web Application Attacks, Malicious Injection Attacks, Script Injection Attacks, Cross-Site Scripting, Web Application Security

I. INTRODUCTION

In today's world the use of web paradigm is becoming an emerging strategy for applications software companies. It allows the design of pervasive applications which can be potentially used by thousands of customers from simple web clients in this paper mainly focused on the specific case of cross-site scripting attack against the web application. Now-a-days almost each and every one in contact with the computer technology Web application openly present an interface through which clients can interact. In actual world it is very crucial to achieve entire security as some security flaws continually exist which can attack the application in distinct ways. XSS offers an opening to the invader or hacker to enter the web server database, multilate websites, seize the compel him/her to take an unfamiliar routes. We focus in this paper to specific cross-site-scripting attacks against the security web applications.

We survey in this paper the two most representative XSS attacks that can actually affect current web applications. Here some alternative categorizations, XSS attacks and of the prevention mechanisms. code injection attack is the most familiar and fatal attacks among the top ten web application vul nerabilities followed by broken authentication and session management and cross-site scriptingattacks Web applications get input from the end users by way of textboxes in the form of name, passwords, feedback etc. These input values are stored in database. Malevolent users insert SQL (Structured Query Language) query or script to do injection attacks.

II. LITERATURE SURVEY

A. Non-Persistent XSS Attack

Non-Persistent is also known as Reflected XSS. In general XSS attacks are based on the victim's browser trust in a legitimate, but vulnerable website or web application. The reflected XSS condition is met when a website or web application employs user input in HTML pages returned to the user's browser, without validating the input first.

B. Preventing Cross Site Attacks

CSRF is an attack that tricks the victim into submitting a malicious request. Therefore, if the user is currently authenticated to the site, the site will have no way to distinguish between the forged request sent by the victim and a legitimate request sent by the victim. CSRF attacks target a state change on the server, such as changing the email address or password. Forcing the victim to retrieve data doesn't benefit an attacker because the attacker doesn't receive the response.

C. Prevention of Stored XSS & Dom

The authors reviewed and concluded that there is very less possibility to detect the XSS, so the author recommended prevention technique using hierarchical approach. This technique applies the security testing using black box and white box in the initial level, the second level in the hierarchy utilized the XSS testing tool Burp. The attack is prevented when the author used the first level. However, the prevention is not fully performed in this paper.

D. Detection of Fine Grained Randomization Approach

A fine grained randomization based approach to mitigate a code injection based attack, which is named as return oriented programming (ROP), this ROP changes the sequence of instructions. The authors mitigated the code reuse attacks using random shuffling of function blocks in the target binary. This approach could effectively deny the attacker. The detection and prevention of such attacks has different types of characteristics. However, the Marlin randomized prototype is difficult and unable to perform certain binary rewriting.

E. Stealthy False Data Injection Attack

A detection scheme for stealthy false data injection attack is proposed. Unlike earlier works, the authors concentrated on the intelligent attackers, where they can design a sequence of data injection into the sensors. This process was performed by the attacker to be undetected. So, authors developed a coding matrix to convert the real sensors outputs to improve the performance under code and data injection attacks. Authors proposed a heuristic algorithm to decide the time interval. Based on the time interval, the system updates the matrix. This work is the recent one which also performed in the sensor networks. But the authors failed to explore the coding scheme for structural constraints.

F. SQL Injection Attack Based On the Type of User Input

SQL injection attack and its classification was provided by Halfond et al., This work classified SQL injection attack based on the type of user input, namely Injection through cookies, injection through server variables and second order injections. The attack was also categorized based on the goal of the attacker, namely identifying injectable parameters, performing database determining database schema, extracting data, adding data, performing denial of service, evading detection, by passing authentication, executing remote commands, performing privilege escalation.

In another work implemented a system named, Webs SARI that detects input-validation-related errors using information flow analysis. In this approach, static analysis was used to check taint flows against preconditions for sensitive functions. One technique to detect when tainted input has been used to construct an SQL query has been proposed by Livs hits.

G. Information Flow Technique

It gets the vulnerability specifications from the user and uses this as static analyzer point. This technique detects SQLIA, XSS and Hyper Text Transfer Protocol (HTTP) splitting attacks.

An approach called SQLR and based on instruction-set randomization was developed. It allows developers to create SQL queries using randomized step by step instruction instead of normal SQL keywords. This approach specifically designed for Common Graphical Interface (CGI) application.

Technique	Description	Features	Advantages	Drawbacks
PRIVILEGE ESCALATION	Higher Level of permission on a system or network.	Protected from an Application or user.	User account with Administrator like access can also be used.	Adversaries Can enter a system with unprivileged access and must take advantage of a system weakness to obtain system privileges.
INPUT VALIDATION MECHANISM	Inputs are checked and validated	Vulnerabilities that would be solved by input validation.	Easier implement validation.	It will take longer or the programmer to script the validation code.
OWASP	Improving the security of software.	Automatically find Security vulnerability in your web application.	Using known vulnerable components.	Cannot guarantee that all possible security threats will be uncovered.
CSRF TOKENS	Each sensitive HTTP request contain the correct token.	Attacker determine the right values for all the form of URL inputs.	Accepting request From trusted sources only.	Wide spread vulnerability.
SPONTSNEOUS RETRIVAL MECHANISM	Monitoring and spontaneous retrieval.	Dependent on prefrontal cortex and working memory capacity.	Should not be equated with automatized prospective memory responding.	Participants may instead selectively engage monitoring when they Enter into context.

Table 1:

III. PROPOSED SYSTEM

As technology development increases, equally web vulnerabilities also increases. Hence the more the research focuses on web security, the greater the emphasis laid on the area of detection and prevention of related vulnerabilities. This also applies to the field of mail and other dynamic web applications and its security. Many works introduced and provided vulnerability detection mechanisms, while some others provide both detection and prevention mechanisms. However, the application has several issues like performance oriented, more vulnerabilities are not handled together etc.,

In the proposed system a security scheme is proposed to protect the web application from cross site scripting and injection attacks, with the aim of regulating the existing techniques into detailed analysis that promotes future investigation. The proposed system specifically considered 4 types of attack such as Cross Site scripting, Code, data and SQL injection attacks. In addition, the existing research works on the three attack categories.

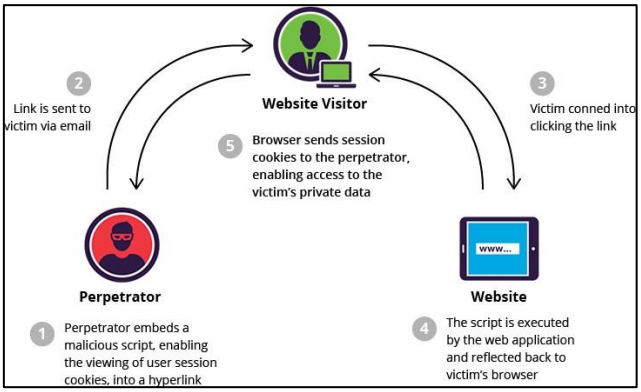

Fig. 2:

ACKNOWLEDGEMENT

I would like to express my sense of gratitude and regards to my guide Dr.T.Parameshwaran., M.E.,Ph.D., for guiding me properly in my project work and for helping to solve the project work difficulties. I would like to thanks all the staff members of computer science and engineering department for supporting me and guiding me in my project work whenever required.

REFERENCES

[1] Shrivastava, Ankit, Santosh Choudhary, and Ashish Kumar. "XSS vulnerability assessment and prevention in web application." Next Generation Computing Technologies (NGCT).
[2] Gupta, Aditi, JavidHabibi, Michael S. Kirkpatrick, and Elisa Bertino. "Marlin: Mitigating code reuse attacks using coder and randomization
[3] Yang,Xinyu, jie Lin, Wei Yu,Paul-Marie moulema,Xinwen Fu,and Wei Zhao Xinwen Fu."A novel en-route filtering scheme against false data injection attacks in cyber-physical networked systems. "IEEE" Transactions on Computers 64, no. 1 (2015): 4-18.
[4] Miao, Fei, Quanyan Zhu, MiroslayPajic, and George J. Pappas. "Coding schemes for securing cyber-physical systems against stealthy data injection attacks.

