KICSP Conference of Korea Institute of Convergence Signal Processing

A study on OWASP Top Ten’ s constantly remaining vulnerabilities on Web Applications and their protection methods

Adkhamboy Makhmudjonov[1], Young Sil Lee[2*]
[1]Dept. of of Computer Engineering, Dongseo University
[2]Dept. of Computer Engineering English track, International College, Dongseo University

Abstract The concept of “Web Application” was introduced to the public in 1999. Since then, the concept of “Web Application” has changed dramatically and today, Web Applications are deployed on a large scale. Unfortunately, these web applications continue to face different types of attacks. The biggest challenge the companies face is how to build a Web Application that will satisfy their requirements with the respect to sensitive data exchange, and business and security workflows. In this paper, we identify these remaining web vulnerabilities according to OWASP Top Ten, their corresponding attacks, and their countermeasures.

• Key Words : Web application, Security, Cyber attack, SQLI, Broken Access Control

Ⅰ. Introduction

In the past few years, Web Applications are constantly deployed at a broad scale. With the increasing use of the Internet as a commercial channel, there are a growing number of websites utilized to offer online services, selling all sorts of goods, sharing information, distributing news and articles, etc. On the other hand, with the rise of cyber-attacks in the past decade, websites have been vastly targeted and are being exploited leaving businesses with substantial losses and at risk. This paper will aim to provide readers with information about specifically 2 vulnerabilities from the OWASP Top 10 reports [1], that is remaining the issue from the invention time of Web Application and will try to give some solutions to those vulnerabilities and the impact on businesses who suffer from an attack and concluding whether the security of this sections should be taken seriously.

Ⅱ. CYBER-ATTACKS AGAINST WEB APPLICATIONS

In accordance with OWASP Top Ten, between 2003 and 2021 there are still constantly remaining vulnerabilities in Web Applications like “Code Injection” and “Broken Access Control”. In this section, we describe the four remaining attacks from the invention time of the Web Application.

2.1 Injection attack

An injection attack [1] is a type of attack, where the attacker is able to put a malicious code into a program or query or even able to upload the malicious program into a web application to execute the commands remotely, which allows the attacker to modify the database, or change, delete, put data on a website. This type of attack is usually made possible due to a lack of proper input/output validation. By providing malicious information, the attacker can mislead the interpreter and cause unintended commands [2]. In general, SQL injection[3], Code Injection[4], and XSS[1] are considered critical Injection attacks.

2.2 Broken Access Control

When the regulations do not allow a user to act out of their permitted actions is called Access Control. Failure of this access control leads to unauthorized information disclosure, modifications on the server-side, or even leads to the destruction of all data, which will lead later on for the organizations to not be able to service the clients of the website and this vulnerability is called Broken Access Control. Broken Access Control consists of multiple attack vectors such as bypassing access control checks, editing other accounts, the elevation of privilege, metadata manipulation through access control tokens like JWT (JSON Web Tokens), unauthorized API access through misconfigurations of CORS or access to unauthorized web pages as the underprivileged user which can lead to attackers control business functions or the possibility of attackers obtaining all data [5].

Ⅲ. COUNTERMEASURES AGAINST THESE ATTACKS

3.1 Countermeasures against Injection attacks

As one of the widely spread attack types, the SQL injection (SQLI), there were given many kinds of prevention methods to it. In [6], the authors propose a Web Application Firewalls (WAF) using Artificial Neural Network (ANN) to avoid SQL injections, which consists of pair of steps: The training step and the Working Step. During the training step, a set of normal and malicious data is used to train the ANN with MATLAB. The Training step is integrated into a WAF to protect the web application during the Working step. Three open-source applications written in PHP were used to test and verify the framework that is presented in the [7] paper. First, the framework computes the entropy of each query in a program accessed before program deployment. As part of the program execution process, they compute the entropy again when an SQL query is invoked to see if there has been any change in the entropy measure since the previous query. The approach then relies on the assumption that dynamically queries with benign inputs do not result in any change of entropy value. In [8], the researchers proposed a CCSD (Cloud Computing SQLIA Detection) to detect SQLI attacks. CCSD does not have any access to the application’s source code but, it can directly apply to existing cloud environments. The main idea of CCDS is to compare the structure of the parse tree to the user’s request. During the first deployment, cloud administrators need to send various requests to this application to build a repository which later stores the parse trees.

XSS is another type of Injection attack which is widely spread in Web Applications. Researchers of [9] propose three server-side techniques to prevent session hijacking attacks. Each of these techniques removes one of the identified prerequisites of the attack classes, and this combination of techniques is called “Session Safe”. It Protects the Web Application by removing the fundamental requirements session hijacking and disabling the reliability of attacks.

3.2 Countermeasures against Broken Access Control

In [10], the associative relation of factors has been identified for Broken Access Control venerability and the possibility of preferences among the Web Designers and Developers about different factors. From the result of the research, they proposed that web designers and developers should keep an eye on the latest web problems and their solutions to be secure from intruders. They analyzed over 330 web applications that were developed with PHP, Java, and .Net platform and used in various sectors. Of those Web Applications, 129 were vulnerable to the Broken Access Control and 201 were non-vulnerable. And the hosting servers of those sample applications were found operating with UNIX, Windows, and Cent-OS respectively. These results can be used for them to take countermeasures before the site is hosted on the production.

Ⅳ. CONCLUSION

This paper summarized critical vulnerabilities in web applications and introduced protection methods on the server-side. Knowing the basic knowledge of these types of attacks can help businesses to actively secure against these attacks and keep their assets and data safely protected.

ACKNOWLEDGMENTS

본 논문은 2022년 해양수산부 재원으로 해양수산과학기술진흥원의 지원을 받아 수행된 연구임 (해양 디지털 항로표지 정보협력시스템 개발(2/5) (20210650))

REFERENCES

[1] OWASP. “OWASP Top 10 vulnerabilities” Retrieved from https://owasp.org/www-project-top-ten/

[2] Artur S. Choudhary and M.L. Dhore “CIDT: Detection of Malicious Code Injection Attacks on Web Application” Vishwakarma Institute of Technology, Pune, India.

[3] PortSwigger. “Web Security Academy” Retrieved from https://portswigger.net/web-security/sql-injection

[4] Ouissem B. F., Omar CH., Moez K., Habib H., Abdelouahid D. “An OWASP Top Ten Driven Survey on Web Application Protection Methods.” Univ. Of Sousse, Tunisia & Taif Univ., KSA & Univ. Of Monastir , Tunisia & Al-Baha Univ., KSA & Univ. of Sfax, Tunisia & Univ. of Moncton, Camada & King Saud Univ., KSA

[5] OWASP. “Broken Access Control” Retrieved from https://owasp.org/Top10/A01_2021-Broken_Access_Control/

[6] Moosa A. “Artificial Neural Network based Web Application Firewall for SQL Injection”, World Acad. of Science, Engineering Technology

[7] Shahriar, H., Zulkirnine, M., “Information-theoretic detection of SQL injection attacks. In High-Assurance Systems Engineering (HASE)”, School of Computing , Queen’s University, Kingston, Canada

[8] Tsu-Yang Wu, Chien-Ming Chen, Xiuyag Sun, Shuai Liu, Jerry Chun-Wei Lin, “A Counter measure to SQL Injection Attack for Cloud Environment”, Wireless Personal Communications

[9] Martin Johns, “Session Safe: Implementing XSS immune Session Handling”, University of Hamburg, Hamburg.

[10] M. M. Hassan, M. A. Ali, T. Bhuiyan, M. H. Sharif, S. Biswas, “Quantitative Assessment on Broken Access Control Vulnerability in Web Applications” Daffodil International University, Dhaka, Bangladesh.