Web Security Vulnerabilities: Classification according to their way of cause

Makhmudjonov Adkhamjon, Ga-Hyun Lee, and Young Sil Lee*
Department of Computer Engineering, International College, Dongseo University
Busan, South Korea
Department of Information Security, Dongseo University
Busan, South Korea
[e-mail: adkhamjon101@gmail.com, igh1116@naver.com, youngsil.lee0113@gmail.com]
*Corresponding author: Young Sil Lee

Abstract

Vulnerabilities exist in many forms within modern web applications which can be easily mitigated with an investment of time and research. This paper tries to classify the OWASP top 10 most common web vulnerabilities according to their way of cause in the web application. Overall, the paper tries to give additional information about the causes of vulnerabilities and enables researchers and developers to secure the applications more easily.

***Keywords*:** Web Application, Vulnerability, OWASP Top 10, Classification, Security threats

1. Introduction

Web applications have become the conventional way to provide services on the Internet. They are used for indispensable tasks and frequently handle sensitive user data. Quite the reverse, web application security is one of the relevant issues in the security field due to the nonstop growth in the number of web-related attacks. Unfortunately, this situation may happen due to problems with time-to-market pressure and financial constraints. To prevent the threats, several robust research papers exist in the literature. Each research paper has its own strong points and barriers. This paper aims to present a classification of OWASP top 10 [1]. most common vulnerabilities according to their ways of cause. In the end, we give some conclusions to the research paper.

2. Classification

This section will provide information on a Classification of vulnerabilities according to Table 1.

2.1 Broken Access Control

In [2] research paper, the BAC vulnerability is described as a critical security vulnerability, which allows attackers to bypass authorization safeguards and perform tasks as if they were privileged users. It mainly occurs because of Design Errors and Misconfigurations, here is one example of this [1]:

- Privilege escalation, where the attacker gains unauthorized access to the user's account without logging in to the system or even worst gaining the access to the admin account when logged in as a user.

"This research was supported by the MIST(Ministry of Science, ICT), Korea, under the National Program for Excellence in SW), supervised by the IITP(Institute of Information & communications Technology Planing & Evaluation) in 2022"(2019-0-01817)

Table 1. Main Causes of the vulnerabilities

Vulnerabilities		Classification						
		Policy flaws	Design errors	Protocol Weaknesses	Misconfiguration	Software Vulnerability	Malicious software	Hardware vulnerability
1	Broken Access Control	☐	☑	☐	☑	☐	☐	☐
2	Cryptographic Failures	☐	☐	☑	☑	☑	☐	☐
3	Injection	☐	☐	☑	☐	☑	☐	☐
4	Insecure Design	☐	☑	☐	☐	☐	☐	☐
5	Security Misconfiguration	☐	☑	☐	☑	☑	☑	☑
6	Vulnerable and outdated components	☑	☐	☐	☑	☑	☑	☑
7	Identification and Authentication failures	☐	☑	☑	☑	☑	☑	☐
8	Software and data integrity failures	☐	☑	☐	☑	☑	☑	☐
9	Security Logging and Monitoring Failures	☑	☑	☐	☑	☑	☐	☐
10	Server Side Request Forgery (SSRF)	☐	☑	☑	☑	☑	☐	☐

2.2 Cryptographic Failures

Data in transit and at rest – such as health records, personal information, and confidential business information, all require additional protection because of the probability of Cryptographic Failures (Sensitive Data Exposure). This especially needs to be taken seriously if it falls under the GDPR, CCPA, or other regulations. We can see that the causes of Cryptographic Failures are Protocol Weakness, Misconfiguration, and Software Vulnerability. An example of this vulnerability is [1]:

- When the data is transmitted in clear text. This relates to the HTTP, SMTP, FTP, and TLS protocols

2.3 Injection

In [3], the injection attack refers to a broad class of attack vectors, in which the attacker supplies malicious input to the application. Then, the input gets processed by an interpreter as a part of a command/query, ending with executing that malicious code in the website's servers. Protocol Weaknesses and Software vulnerabilities are the main reasons to occur Injection vulnerability. An example of that is [1]:

- The application doesn't validate, filter, or sanitize the user input data properly.

2.4 Insecure Design

This vulnerability type occurs when the lack of security controls is integrated into the application throughout the development cycle. It may cause wide-range and deep-rooted security consequences as the application itself isn't designed with security in mind [4]. The main cause of this vulnerability falls under Design errors. Below is one example of this [5]:

- When development teams often ignore secret management and access control best practices, which later leads to security breaches.

2.5 Security Misconfiguration

When it is failed to implement the proper Security Controls for application, container, or any other software component, Security Misconfiguration vulnerability can occur. These types of misconfigurations can allow unauthorized access to system data and functionality, or even, complete system failure. [6][1]. Security Misconfiguration vulnerability mainly occurs because of Design Errors, Misconfiguration, Software vulnerability, Malicious software, and Hardware vulnerability, here is an example of this:

- There are default accounts, and their passwords are enabled and unchanged in the production.

2.6 Vulnerable and Outdated Components

Previously known as Using Components with Known Vulnerabilities, It usually occurs when the existing system uses outdated components such as libraries, frameworks, and other software frameworks at the same privilege level as the web application [1]. Policy flaws, Misconfiguration, Software Vulnerability, Malicious Software, and Hardware vulnerability are the causes of this attack. An example is [1]:

- The development team doesn't secure components' configurations properly.

2.7 Identification and Authentication Failures

Previously known as Broken Authentication Vulnerability, it is related to authentication schemes of applications. If a web application has a such vulnerability, it may cause serious and damaging data breaches [7]. It falls under Design errors. Protocol Weaknesses, Misconfiguration, Software Vulnerability, and Malicious software criteria following example is one of the occasions [1]:

- The web application uses default, weak, or well-known passwords.

2.8 Software and Data Integrity Failures

This vulnerability category is quite new. Basically, it relates to code and infrastructure that isn't protected against integrity violations [8]. The table shows that the causes of Software and Data Integrity Failures are Design Errors, Misconfiguration, Software Vulnerability, and Malicious Software. One of the reasons why it falls under these criteria is [1]:

- System uses insecure CI/CD pipeline.

2.9 Security Logging and Monitoring Failures

This vulnerability happens, when the system fails to log, monitor, or fails to report security events, such as login attempts, making suspicious behavior hard to detect and the probability of taking control over the application rises [9]. Security Logging and Monitoring Failures mainly occur because of Policy Flaws, Design Errors, Misconfiguration, and Software Vulnerability, here is an example of this [1]:

- Application logs and APIs are not monitored for malicious activity.

2.10 Server-Side Request Forgery

SSRF is a web application vulnerability, which allows the attacker can misuse the application functionality to read/modify internal recourse [10]. Design Errors, Protocol Weaknesses, Misconfiguration, and Software Vulnerability are the causes of this attack. One of the reasons, why it falls under these criteria, is [1][11]:

- Broken trust relationships between systems.

3. Conclusions

In this paper, we present the causes of the OWASP Top 10 most common vulnerabilities. We tried to collect all the information related to them and give clear causes. By doing this we hope to help security researchers and developers get brief information about each vulnerability type and try to make their applications more secure.

References

[1] OWASP. "OWASP Top 10 – 2021", 2021.
[2] M.M. Hassan, M. Ali, T. Bhuiyan, M.H. Sharif, S.Biswas, "Quantitative Assessment on Broken Access Control Vulnerability in Web Applications", 2018.
[3] IBM. "IBM Security Network Intrusion Prevention System", 2021.
[4] HackerOne. "OWASP Top 10 Web App Security Risks", 2021.
[5] CrashTest Security. "Insecure Design", 2022.
[6] B. Eshete, A. Villafiorita, K. Weldemariam, "Early Detection of Security Misconfiguration Vulnerabilities in Web Applications", 2011.
[7] O. B. Fredj, O. Cheikhrouhou, M. Krichen, H. Haman, A. Derhab, "An OWASP Top Ten Driven Survey on Web Applications protection Methods", 2020.
[8] AskF5, "K50295355: Software and data integrity failures (A8) | Secure against OWASP Top 10 for 2021", 2022.
[9] AskF5, "K94068935: Security logging and monitoring failures (A9) | Secure against the OWASP Top 10 for 2021", 2022.
[10] J. Bahruz, K. Amin, M. Omid, K. Engin, "Preventing Server-Side Request Forgery Attacks", 2021.
[11] AskF5, "K36263043: Server-side request forgery (SSRF)(A10) | Secure against OWASP Top 10 for 2021", 2022.