Northumbria University
NEWCASTLE

Security Report

Part-B

Word Count – 894 words

Name: Kurre Sai Lakshma Reddy

Student ID: W21037098

Subject: Web Development and Deployment

Code: KF7013

Tutor: Emma Anderson

Introduction

In the modern internet age, data is crucial for predicting user behavior and creating institution-specific models. Alternatively, the main difficulty for corporations is safeguarding user data from hackers. Website security's goal is to stop these (or any) threats, according to developer.mozilla.org (Website security - Learn web development | MDN, 2022). Website security is more formally defined as "the act or practice of safeguarding websites against illegal access, use, alteration, destruction, or interruption."

The safeguarding of servers and websites against online dangers is the subject of web security, often known as cyber security. By restricting, detecting, and responding to threats, it seeks to secure sensitive data. Utilizing website security tools to scan the URL for malware and vulnerabilities is part of the website security evaluation.

Attacks on computers have become more frequent. Securing internet devices and accounts has become crucial for both organizations and people. Websites, YouTube channels, and personal blogs must all be secured. There are several strategies for safeguarding and securing oneself. Because of these key motives, it is crucial to offer website security (Agnes Talalaev, 2022).

- **Targeting our clients are hacked websites.**
- **Website hacks are becoming more commonplace.**
- **Drop in income and damage to the company's reputation**
- **Internet domain is banned.**
- **The cost of website clean-up is higher than that of protection, we have learned from our errors.**

The main vulnerabilities on websites are Injection Flaws, Broken Authentication, Cross Site Scripting, Insecure Direct Object References, Security Misconfiguration, Sensitive Data Exposure, Missing Function Level Access Control, Cross-Site Request Forgery, Using Components Known Vulnerabilities, and Unvalidated Redirects and Forwards. In this study, we primarily talk about two issues:

1. **Cross-Site Scripting(XSS)**
2. **Injection Flaws**

1. Cross Site Scripting(XSS):

Attackers can insert client-side scripts into web pages using XSS by taking advantage of flaws in dynamically generated websites. Data theft, session hijacking, web page redirection, and other issues can all be brought on by an attacker inserting malicious code into a trustworthy website or online application.

Stored XSS:

The database, for instance, a blog, will save the attacker's input. This script will be active on every visitor's computer, having an impact on everyone who accesses the page.

What is XSS | Stored Cross Site Scripting Example | Imperva, 2022). To carry out a stored XSS attack successfully, a perpetrator must first identify a weakness in a website and then inject a malicious script into its server (for example, via a comment field). To prevent cross-site scripting, I have set most of the input fields in this created web application are secured with patterns and minimum and maximum lengths. All sessions are configured to end and be unset when the user logs out.

2. Injection Flaws:

Malicious code can be sent to a different device via an application thanks to injection flaws. Injection vulnerabilities can be of three different types:

2.1. LDAP Injection
2.2. SQL Injection
2.3. Command Injection

2.1. LDAP Injection:

An IP-based active directory called LDAP uses user characteristics to hierarchically arrange the information. Similar to how SQL injection operates, LDAP injection involves an attacker attempting to introduce arbitrary data to create malicious queries that the LDAP server will execute.

In order to prevent undesired intrusions into the program, I utilized the input data clean method to remove HTML tags, special characters, and strings in this built **Happy Travels** . Similar to this, I have only ever used the GET technique when absolutely necessary when it comes to URLs; no sensitive data is transmitted over it. To safeguard user accounts if a hacker gains access to the backend database, all passwords are hashed using the standard function to hash passwords.

2.2. SQL Injection:

As dynamic inputs that are used to generate requests for online applications, user login screens, URLs, and search boxes are of interest to hackers. When an attacker successfully tricks the server into

generating a malicious query and directing the back-end database to execute it, SQL injection happens. The database could undergo a denial-of-service attack, deletion, or even modification.

2.3. Command Injection:

A user's ability to insert operating system commands into any user input area opens the door for an attacker to insert malicious commands into the web server and steal sensitive data.

Conclusion

In order to conclude, the necessary security defensive mechanisms were developed, such as sanitizing all user input fields to remove undesirable inputs like scripts and SQL queries, making sure that sensitive information like usernames and passwords was never included in URLs, and ensuring that the majority of data was sent via the POST method only. Finally, sessions are carefully controlled, scheduled to expire when the user logs out, and well managed.

References

iPage Blog. 2022. Why Web Security is Important – iPage Blog. [online] Available at: [https://ipage.com/blog/why-web-security-is-important/].

cWatch Blog. 2022. What is Web Security? | Web Application Security Tools 2022. [online] Available at: [https://cwatch.comodo.com/blog/website-security/what-is-web-security/].

Total Engineering Blog. 2022. 10 Most Common Web Security Vulnerabilities. [online] Available at: [https://www.toptal.com/security/10-most-common-web-security-issues/].

Learning Center. 2022. What is XSS | Stored Cross Site Scripting Example | Imperva. [online] Available at: [https://www.imperva.com/learn/application-security/cross-site-scripting-xss-attacks/#:~ :text=Stored%20XSS%2C%20also%20known%20as, application%2C%20onto%20a%20user's%20browser.].

Patchstack. 2022. 5 Reasons Why Website Security Is Important - Patchstack. [online] Available at: [https://patchstack.com/reasons-why-website-security-important/].

Developer.mozilla.org. 2022. Website security - Learn web development | MDN. [online] Available at: [https://developer.mozilla.org/en-US/docs/Learn/Server-side/First_steps/Website_security].