SECURITY RISK ASSESSMENT REPORT

https://sit-chameleon-website-0bc2323.ts.r.appspot.com/

VERSION 0.0.1

05/12/2023

Miriam Azmy

TABLE OF CONTENTS

Introduction:

Background

While undertaking the security risk assessment for the web application, careful attention was directed towards the examination of two pivotal components: the responsive navigation bar and the newsletter signup form associated with Chameleon company. Both of these elements play integral roles in the overall user experience and functionality of the web application, making them prime candidates for scrutiny regarding security vulnerabilities. The responsive navigation bar serves as the user's gateway to navigate through the application efficiently, ensuring seamless interaction and providing access to various sections. Simultaneously, the newsletter signup form acts as a crucial entry point for users interested in receiving updates and information from Chameleon company and its user’s information.

The responsive navigation bar, being a fundamental element of the web application's user interface, is responsible for delivering a cohesive and user-friendly experience. It dictates how users interact with the application, affecting their navigation journey and overall satisfaction. Given its significance, any potential vulnerabilities within this component could have long term effects, impacting user trust and the application's reliability. Meanwhile, the newsletter signup form, being a gateway for user engagement, necessitates a robust security posture. Users sharing their email addresses for subscription purposes expect their data to be handled securely, evaluating potential vulnerabilities within this component is crucial for maintaining the integrity of user information.

Objectives

The overarching objectives of this security risk assessment is to address the unique challenges and responsibilities associated with securing the web application. The primary aim is to systematically identify, assess, and mitigate security risks inherent in the responsive navigation bar and the newsletter signup form. The multifaceted nature of these objectives includes various critical aspects to ensure a comprehensive approach to web application security.

Identification of Security Risks:

The initial phase of the assessment focuses on a thorough examination of the codebase, scrutinizing each component for potential vulnerabilities. This involved a detailed analysis of the logic, data flow, and external interactions associated with the responsive navigation bar and the newsletter signup form. The objective here was to unearth any vulnerabilities or weaknesses that could be exploited by malicious entities.

Assessment of Potential Impact:

Understanding the potential impact of identified security risks is a key vital to ensure the progression of this project remains on track. This involves evaluating the consequences of potential exploits, considering factors such as data integrity, user trust, and overall system reliability. By assessing the impact, the goal was to prioritize risks based on their severity and potential repercussions.

Actionable Recommendations:

The final objective centres around providing actionable and context-specific recommendations to enhance the security posture of the web application. These recommendations were tailored to address the identified vulnerabilities and mitigate the associated risks effectively. They encompassed improvements in coding practices, implementation of security best practices, and enhancements to user guidance and experience.

Ultimately, the objectives were meticulously designed to solidify and enhance the security foundations of the web application, ensuring that users can interact with confidence while safeguarding sensitive information. The assessment aims to uplift the development team with actionable insights to bolster security measures effectively.

Methodology

3.1 Responsive Navigation Bar Assessment

3.1.1 Overview

This assessment involves a comprehensive analysis of the underlying code governing the responsive navigation bar, which is a critical component shaping the user's interaction with the web application and dictating their navigation through various sections. The primary focus is on understanding the intricacies of the code governing this navigation bar, intending to fortify its security foundations.

Conducting this assessment, involved a thorough examination of the codebase, implementation of security best practices, and consideration of potential threat vectors. Various methodologies are employed, including code review, penetration testing, and security auditing. By reviewing code it allows for a detailed analysis of the source code, identifying potential vulnerabilities and areas for improvement. Penetration testing involves simulating real-world attack scenarios to assess the system's resilience against potential threats. Additionally, security auditing ensures adherence to established security standards and best practices.

By using both static and dynamic analysis methods, it enabled a comprehensive understanding of the responsive navigation bar's security posture. Static analysis involves reviewing the source code without executing the application, and identifying vulnerabilities that may be present in the codebase. Dynamic analysis, on the other hand, involves assessing the application during runtime and simulating user interactions to uncover potential security issues. This multifaceted approach ensures a thorough and effective evaluation of the responsive navigation bar's security, addressing both code-level vulnerabilities and runtime risks.

3.1.2 Findings

During the ongoing assessment of the responsive navigation bar, the evaluation extensively documents any vulnerabilities or issues discovered, covering a range of aspects such as key observations, security vulnerabilities, and potential compliance issues. One notable concern that surfaces is the absence of validation in image sources, particularly within the chameleonHeader variable. Image sources, if not adequately validated, present a considerable security risk, notably in the form of injection attacks.

Furthermore, the assessment identified another finding related to the button click handler (onClick={() => setNavbarOpen(!navbarOpen)}). The lack of detailed validation for secure state changes underscores the potential vulnerabilities associated with state manipulation in the context of inadequate validation.

While compliance is a crucial consideration for web applications, especially in regulated industries, the current assessment concentrates on identifying and addressing potential security risks.

3.1.3 Risk Assessment

The absence of validation in image sources introduces a notable risk, particularly concerning injection attacks. In this context, malicious code could exploit the lack of validation and be injected into the image source, posing a potential security threat. This risk is assessed considering factors such as the likelihood of occurrence, gauging how probable it is for an injection attack to take place. Additionally, the impact on the business is considered, considering the potential consequences of a successful attack on the image source. The overall risk rating, a comprehensive measure of the identified risk, provides a nuanced understanding of the severity of the security concern.

For the absence of validation in image sources, the likelihood of occurrence might be influenced by factors such as the visibility of the application to external users, the complexity of the image source validation process, and the security measures in place. Similarly, the impact on the business would depend on the criticality of the affected image source to the application's functionality and user experience.

Regarding the button click handler, the likelihood of unauthorized state changes could be influenced by the sensitivity of the state being manipulated, the complexity of the code controlling the state, and the potential avenues for exploitation. The impact on the business would hinge on how essential the navbarOpen state is to the application's responsiveness and the user's navigation experience.

3.1.4 Recommendations

To address the identified security issues, several recommendations are proposed. For the image source (<img src={chameleonHeader}), it is recommended to implement thorough validation to ensure the integrity and security of the image being displayed. This includes validating the source against a predefined set of acceptable values or patterns to prevent potential injection attacks. Regarding the button click handler, the recommendation is to incorporate additional checks to ensure secure state changes and prevent unauthorized manipulations. By implementing these changes, the security posture of the responsive navigation bar can be significantly enhanced.

3.2 Newsletter Sign Up Form

3.2.1 Overview

The ongoing assessment of the newsletter signup form, integral to Chameleon company's user engagement strategy, focuses on ensuring data integrity and enhancing the overall user experience. This evaluation falls under the category of a comprehensive security risk assessment, which encompasses an in-depth analysis of the code governing the newsletter signup form. The scope of this assessment extends to the entire codebase responsible for handling user subscriptions and interactions with the Chameleon newsletter service.

Various techniques are employed, including code review, vulnerability scanning, and analysis of potential user inputs to identify vulnerabilities that could compromise data integrity or impact the user experience. This ensures a thorough examination of the security foundations of the newsletter signup form, considering both technical vulnerabilities and user-centric aspects.

The scope extends to all components involved in the signup process, including user input validation, data transmission, and error handling. This includes analysing the codebase for common security pitfalls, employing automated tools to identify vulnerabilities, and simulating user interactions to evaluate how the form handles various inputs.

3.2.2 Findings

Notably, one critical area of concern lies in the absence of client-side email format validation. This vulnerability exposes the form to potential exploits, allowing users to submit incorrectly formatted or even malicious data. Without proper validation, the risk of poorly formatted email addresses being processed increases, posing a direct threat to data integrity.

Furthermore, the evaluation revealed flaws in the error handling method, particularly within the catch block. The generated error messages are not user-friendly, and thereofre might be exploited by hostile actors looking to exploit user confusion. The ambiguous and non-informative error messages may not only impair the user experience, but also provide chances for attackers to control the system.

3.2.3 Risk Assessment

In the context of the newsletter signup form assessment, the absence of client-side email format validation is recognized as a risk. This vulnerability increases the likelihood of poorly formatted or incorrect email addresses being submitted, which, in turn, heightens the risk of issues arising during the subscription process.

Considering the likelihood of occurrence, the risk is deemed moderate, as users may inadvertently input email addresses with incorrect formats. The impact on the business is assessed as moderate as well since improperly formatted email addresses can lead to failed subscriptions or complications in communication with subscribers. Combining these factors results in an overall risk rating that underscores the significance of implementing client-side email format validation to mitigate these potential issues effectively. Similarly, the user-unfriendly error messages identified in the catch block are subject to the same risk assessment criteria. The likelihood of occurrence is considered moderate, as errors during the subscription process can happen, leading to the display of error messages.

The impact on the business is also evaluated as moderate, given that unclear error messages may cause user confusion and frustration, potentially discouraging them from completing the subscription. The overall risk rating emphasizes the importance of refining error messages to enhance user understanding and, consequently, the overall user experience.

3.2.4 Recommendations

To mitigate the identified risks, recommendations are made to implement client-side email format validation within the newsletter signup form. This ensures that users provide correctly formatted email addresses, reducing the likelihood of data entry errors. Additionally, enhancing the error messages generated in the catch block to be more user-friendly and informative is recommended. Clear and concise error messages guide users effectively, improving the overall user experience and reducing potential frustrations. These recommendations are tailored to fortify the security and usability aspects of the newsletter signup form, aligning with best practices for web development and user interaction.

Findings and Recommendations:

The examination of the responsive navigation bar code exposed potential security concerns. One notable finding pertains to the need for rigorous validation in image sources, specifically in the `chameleonHeader` variable. While the current implementation doesn't immediately indicate a security risk, introducing input validation and sanitization mechanisms would add layer of protection against potential injection attacks or other malicious activities related to image sources. Another identified area for improvement involves the button click handler (`onClick={() => setNavbarOpen(!navbarOpen)}`), which is responsible for toggling the visibility of the navigation menu. While this is a common practice for responsive navigation, ensuring secure state changes is paramount. Additional validation checks and secure state manipulation practices should be incorporated to prevent any vulnerabilities associated with state handling.

To address the identified issues, proposed changes to the code have been suggested in the relevant code review reportd. Enhancing the image source validation for the `chameleonHeader` variable involves implementing stricter validation mechanisms to ensure that the image source is secure and not susceptible to injection attacks. This is particularly crucial if the image source is dynamically generated or fetched from user inputs or external APIs. Additionally, for the button click handler, ensuring secure state changes is imperative.

It is necessary to have both server-side validation while incorporating client-side validation to enhance the overall user experience by providing immediate feedback on the correct email format. Moreover, the error messages provided in the catch block lack user-friendliness. Improved, user-centric error messages would not only aid users in understanding the cause of subscription failures but also contribute to a more positive user experience.

To enhance the security and user experience of the newsletter signup form, recommendations include the implementation of client-side email format validation. This addition ensures that users receive prompt feedback on the correct email format, reducing the likelihood of submission errors. By addressing these recommendations, the Chameleon company can fortify the security foundations of its newsletter signup process and elevate user satisfaction.

Conclusion:

In conclusion, the comprehensive security risk assessment conducted on key components of the web application has provided invaluable insights into potential vulnerabilities and security best practices. The primary objectives of the assessment were met, with a focus on identifying security risks, assessing their potential impact, and delivering actionable recommendations to bolster the overall security posture.

The examination of the responsive navigation bar underscored the importance of stringent validation in image sources and secure state changes in the button click handler. By proposing changes to the code, including enhanced image source validation and improved state manipulation practices. The assessment of the newsletter signup form highlighted key areas for improvement, notably the absence of client-side email format validation and the need for user-friendly error messages. The recommendations emphasize the implementation of client-side validation for immediate user feedback and the refinement of error messages to enhance user understanding.

Regular security evaluations are necessary as the web application matures and integrates new features to identify and mitigate emerging threats. This iterative method guarantees that the application maintains a strong security posture, protecting user data and preserving user confidence. Chameleon organisation can improve the overall security of its web application and encourage a positive and safe user experience by adhering to security best practices and aggressively addressing potential vulnerabilities.