SDK Audit Validation Report
of
Neptune Mutual Application
for

CyRAACS™
Your Trusted Security Partner

CYRAAC Services Private Limited

(No.7/24, Second Floor |11th Main Rd |4th T Block East | Jayanagar | Bengaluru | Karnataka 560011)

SDK Audit Report – Neptune Mutual Application

Report Created By
CYRAAC Services Private Limited

Report Created For
Chain Commit Limited

Confidential Information

The following report contains company confidential information. Do not distribute, email, fax, or transfer via any electronic mechanism unless it has been approved by the recipient company's security policy. All copies and backups of this document should be saved on protected storage always. Do not share any of the information contained within this report with anyone unless they are authorized to view the information. The specific IP addresses / Domain were identified by Client. Our subsequent test work, study of issues in detail and developing action plans are directed towards the issues identified. Consequently, this report may not necessarily comment on all the weaknesses perceived as important by the Client and / or Client management.

Contents

1. Introduction

1.1 Context

Security of the applications has become a major concern for the organisations and its users. More and more applications are becoming an integral part of everyone's lives and used for both Admin and leisure purposes. Information managed by the applications has become the most important asset that needs to be protected as it may contain sensitive personal, Admin, and intellectual properties.

From the information security perspective, this brings in new challenges that need to be addressed for the protection and safeguarding of the information processed by such applications and its users. The confidentiality and integrity of information is of paramount importance because it could be misused if the security gets compromised.

In today's world most of the applications are exposed and available on the internet exponentially increasing the threat scope and making them a prime target because of the valuable information that it handles.

SDK audit helps the organisation to identify vulnerabilities and security issues in the applications code and fix them before releasing them for general use. This helps to achieve enhanced security and a safe user experience.

1.2 Objective

The objective of this document is to provide the results of the SDK Audit of the **Neptune Mutual Application**. The objective of this audit is to examine the software and identity security issues with the application design and implementation, the risks it carries to its users, integrity and confidentiality of the processed information and the long-term application maintainability from an application security standpoint.

The SDK Audit was performed at a controlled environment and CyRAACS has ensured that findings presented in this report are within the provided access control and environment restrictions.

2. Engagement Scope

The project scope covers SDK Audit of **Neptune Mutual Application** during the Validation Assessment. The following are the details of the reviewed application for code / implementation level security issues.

2.1 Application Details – Neptune Mutual Application

Application Name	Neptune Mutual Application
SDK Audit Review Owner	Broken Tusk (Setu)
Review Start	21-Jul-2022
Review End	10-Aug-2022
Review Start (Validation Assessment)	22-Aug-2022
Review End (Validation Assessment)	25-Aug-2022
Objective	SDK Audit
Num. Of. Lines	Frontend - 86,940 Backend - 12,945
Version	Mock
Programming Language(s)	JavaScript, TypeScript, License, Markdown, TypeScript Typings
SDK Audit Methodologies	Validation - Automated and Manual
Automated	Yes
Manual	N/A
3rd Party Libraries	N/A
Extensions / Plugins	N/A
Input management / Data Handling	No
Authentication Controls	Yes
Session Management	Yes
Authorization Management	Yes
Cryptography	Yes
Error Handling / Information Leakage	Yes
Secure communications	Yes
Logging / Auditing implementations	Yes
Secure Design	Yes
Security Controls	Yes
Platform Specific Controls	Yes (according to platforms and best practices)
Database security	N/A
Secrets Management	Yes

3. Review Summary

The issues identified and proposed action plans in this report are based on SDK Audit conducted by CyRAACS professionals. CyRAACS has made specific efforts to verify the accuracy and authenticity of the information gathered only in those cases where it was felt necessary.

The identification of the issues in the report is primarily based on the reviews carried out during the timelines for conducting such an exercise. The vulnerabilities reported in this report are valid as of date **25-Aug-22**. Any vulnerability, which may have been introduced due to the code changes after this or any issue, been made available after the above stated date, does not come under the purview of this report.

Any code updates / developments changes or configuration updates made on the code of the applications covered in this test after the date mentioned herein may impact the security posture either positively or negatively and hence invalidates the claims & observations in this report. Whenever there is an update to the applications code, enhancements, introduction of new application modules, changes, and updates to the development platform, we recommend that a SDK Audit is performed to ensure the security posture of the application is compliant with the organisation's security policies and security requirements.

CyRAACS recommends best practices in secure design and architecture of the applications, secure coding practices and periodic reviews of the application code to ensure a safe user experience and to maintain the security posture of the applications.

CyRAACS has identified **NO** issues in the validation SDK Audit of the Neptune Mutual Application as defined by the scope.

3.1 Static Application Security Testing (SAST) – SDK Audit

The provided recommendations adhere to the industry best practices and addressing them shall result in the following improvements:

- secure software life cycle process
- secure design and architecture implementation and review
- secure coding strategies
- secure operations
- addressing in-house / internal threats factors
- addressing perennial, silent and ever evolving threat categories
- continuous security improvements for – integrity, confidentiality and availability of application and organization, customer data
- protecting intellectual property

3.2 Static Application Security Testing – (SAST) – SDK Audit summary

SDK Audit was performed on the code and has resulted in the following categories of issues.

3.2.1 Frontend

TYPE	INITIAL COUNT	VALIDATION COUNT
CRITICAL	0	0
HIGH	0	0
MEDIUM	0	0
LOW	0	0

3.2.2 Backend

TYPE	INITIAL COUNT	VALIDATION COUNT
CRITICAL	0	0
HIGH	0	0
MEDIUM	0	0
LOW	1	0

The above-listed issue categories were identified by the SDK Audit process.

4. SDK Audit Results – Details

The following details are categorized as per the code organisation / modules across the reviewed applications.

4.1 Frontend

No issues were identified during the validation assessment.

4.2 Backend

No issues were identified during the validation assessment.

5. Conclusion

Though the SDK Audit identified **NO** issues in the provided application files, the following key areas of improvement are suggested as generic guidelines.

It helps the organization to improve its SSDLC maturity (Secure Software Development Life Cycle) methodology and apply necessary processes, controls and reviews for the on-going design and development operations for its application development.

The recommendations are broadly categorized into the following topics covering the secure application design and architecture, development, validation (QA), deployment and maintenance operations.

Coding recommendations

- Create and employ a proper securing code coding guidelines, commenting guidelines for managing the application code. Add adequate code comments to describe the application feature, functionality, and flow of data.
- Employ proper secrets management for managing application configuration and secrets.
- Never hardcode or store any sensitive details like usernames, passwords, customer details, application URLs, endpoints, API keys, license keys etc., in the application code.

Quality Assurance and Internal Code Review recommendations:

- Ensure security testing / code auditing is performed on both static and dynamic (runtime) versions of the application.

-= THANK YOU =-