SECURE SOURCE CODE REVIEW FOR A BIOTECHNOLOGY APPLICATION DEVELOPED USING R LANGUAGE

BACKGROUND

A global biotechnology company, in its pursuit to acquire a cutting-edge application originally developed by a biotechnology research group, recognized the importance of ensuring the security and integrity of the software before integrating it into their existing ecosystem. The application, primarily developed using the R programming language, was a critical asset that required a thorough and secure source code review as part of the formal acquisition process. The primary goal was to verify that the application's code was free from security vulnerabilities that could lead to any compromise of the existing data and systems of the company.

CHALLENGE

The integration of a newly acquired application into an established software ecosystem presents inherent risks, particularly when the application is developed using a specialized language like R. The biotechnology company's existing Static Application Security Testing (SAST) program and scanners were not equipped to fully assess the application, as they lacked the capability to effectively scan and analyze code written in R. This limitation posed a significant challenge in ensuring that the application adhered to strict security standards without compromising its functionality or introducing vulnerabilities into the secure environment.

SOLUTION

To meet these challenges, the biotechnology company engaged Blueinfy. Blueinfy's team embarked on a multi-step comprehensive review process designed to meticulously assess the application's source code and ensure its readiness for integration: -

Gathering Background Information:
Blueinfy began by obtaining detailed background information on the application, including its purpose, key features, targeted audience, and deployment environment. This foundational understanding was critical for tailoring the security assessment to the specific needs of the application and its user base.

Code Analysis:
The team performed an exhaustive examination of the source code, focusing on crucial aspects such as user input handling, data file import/export processes, configuration management, data processing workflows, external and third-party calls, and the libraries/packages utilized. Additionally, the review extended to the generation of the user interface, ensuring that each component was scrutinized for potential security vulnerabilities. This comprehensive code analysis provided a deep insight into the application's architecture and its potential weak points.

R Language Best Practices:
Leveraging the expertise of subject matter experts in R, Blueinfy ensured that the application adhered to best practices specific to the R programming language. This included the correct implementation of built-in security features, such as memory management, data type handling, and error checking mechanisms, all of which were crucial for enhancing the overall security posture of the software.

KEY SECURITY CHECKS:

Blueinfy conducted several critical security assessments to ensure comprehensive coverage of potential vulnerabilities. Some of the key security checks are:

1. **User Input Sanitization:**
The team meticulously traced user inputs received from the interface, ensuring that all input data was validated, escaped, and sanitized using appropriate blacklisting or whitelisting techniques. For file imports, Blueinfy verified that the data was properly sanitized before being processed by the program logic, preventing potential injection attacks.

2. **Secure Password and Secret Storage:**
Blueinfy assessed the mechanisms for storing sensitive information, such as passwords and API keys, ensuring compliance with best practices for secure storage. This involved evaluating encryption methods and access controls to prevent unauthorized access.

3. **Secure Communication:**
The application's communication protocols were examined to ensure that all data transmission was encrypted and secure. Blueinfy also validated the interaction with external resources, ensuring that these connections did not introduce vulnerabilities or leak sensitive data to third parties.

4. **Data Anonymization:**
The team verified that sensitive data was appropriately anonymized before processing, protecting user privacy and ensuring compliance with data protection regulations.

5. **Vulnerability in Packages:**
Blueinfy checked for the use of vulnerable packages within the application code, ensuring that no outdated or insecure libraries were in use.

SOFTWARE COMPOSITION ANALYSIS (SCA):

In addition to the manual code review, Blueinfy conducted a Software Composition Analysis (SCA) to evaluate the third-party libraries and dependencies used within the application. This step was crucial for identifying known vulnerabilities in the external components that could compromise the overall security of the application.

OUTCOME

The secure source code review conducted by Blueinfy provided the biotechnology company with significant benefits:

Enhanced Security Assurance: The review confirmed that the application did not contain vulnerabilities that could lead to sensitive information leakage, and all user inputs were properly validated and sanitized.

Compliance with Security Standards: The findings ensured that the application met necessary security standards, thus mitigating potential risks associated with data breaches and facilitating its integration into the company's secure environment.

Integration Confidence: With the application deemed secure, the biotechnology company proceeded with the acquisition and integration of the software, confident that it would not compromise their existing security posture.

This thorough review not only facilitated the safe integration of the application into the company's software ecosystem but also helped mitigate potential risks associated with data breaches. As a result, the biotechnology company was able to proceed with the acquisition and deployment of the application, assured of its security and compliance.

Article by Maunik Shah & Krishna Choksi