

THE PROBLEM

The client's product was a comprehensive mobile electronic communication compliance solution and it ensured messaging compliance with various regulatory bodies. The target businesses of our client were medium and large business enterprises managed by veterans in the Compliance, Data Security and Information Technology industries.

We were required to perform a formal review of this .NET based application code in order to trace weaknesses and vulnerabilities based on security standards. The team at Kualitatem carried out R&D and training to gain in-depth understanding of all related standards which eventually helped to produce a clear-cut code review.

The client required a code review for this .NET based application using CWE/SANS (Common Weakness Enumeration) and OWASP (Open Web Application Security Project) standards. To effectively perform this code review the QA team required a reliable and cost effective Security Code Review Automation tool that could satisfy majority of the security measures implemented in the system.

THE APPROACH

The QA team at Kualitatem commenced the testing process through performing in-depth R&D around the OWASP and CWE/SANS standards. Once the team was well equipped with the required tools and concepts a detailed checklist was developed for testing the application.

The QA team developed a presentable layout for the draft report. After that detailed analysis was performed to finalize the layout and its contents.

Subsequent to the successful implementation of the platform, formal review was carried out and all the ambiguities and vulnerabilities were reported in more efficient manners. Security vulnerabilities, errors and suggestions were reported based on: Documentation rules, Layout rules, Maintainability rules, Naming rules, Ordering rules, Readability rules and Spacing rules etc.

To measure the Security vulnerabilities following CWE/SANS and OWASP standards were followed to develop a checklist as shown in the right hand side column:

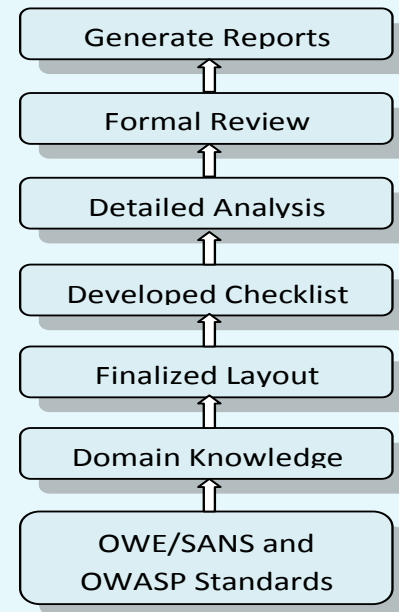


Figure 1

1. SQL Injection
2. Cross-Site Request Forgery (CSRF)
3. Improper Access Control (Authorization)
4. Reliance on Un-trusted Inputs in a Security Decision
5. Use of a Broken or Risky Cryptographic Algorithm
6. Unrestricted Upload of File with Dangerous Type
7. Incorrect Calculation of Buffer Size
8. Missing Authentication for Critical Function
9. Download of Code Without Integrity Check
10. Incorrect Permission Assignment for Critical Resource
11. Allocation of Resources Without Limits or Throttling
12. Improper Check for Unusual or Exceptional Conditions
13. Information Exposure Through an Error Message
14. Cross-site Scripting
15. Path Traversal
16. OS Command Injection
17. Missing Encryption of Sensitive Data
18. Use of Hard-coded Credentials
19. Buffer Access with Incorrect Length Value
20. .net File Inclusion
21. Improper Validation of Array Index
22. Integer Overflow or Wraparound
23. Open Redirect
24. Race Condition

SUMMARY

The team at Kualitatem believes in providing an unbiased view of any application/product to the concerned client. Our comprehensive, yet efficient bug reporting mechanism is designed to help each client with delivering a high quality and reliable product. For this particular application, suggestions were made in the code compliant to .NET best practices. The suggestions were purely based on checklist withdrawn from the standards. The purpose of the Code Review conducted by the team at Kualitatem was to find security vulnerabilities in the .NET application code.

Kualitatem provided a detailed yet clear-cut code review to the client within the defined time frame. Standards were the key track for the Test Engineers to review the code more efficiently and with complete coverage.

The approach followed involved attaining domain knowledge, layout design, identifying checklist based on standards, detailed analysis, conducting a formal review and generating report based on the review. Apart from the security vulnerabilities other suggestions were also reported that could be implemented to improve the coding technique.