Secure Code Review Methodology (Focused)

Our focused code review methodology focuses on identifying security weaknesses in the application source code to detect high-impact vulnerabilities — those introduced through design or logic flaws rather than syntactic errors. The focused code review service majorly relies on expert human intelligence with semi-automated tools just utilized for tracing. Below is the methodology that is used for a focused secure code review: -

Application Overview/Pre-Requisites

As a foundational step, we review the overall code structure, underlying framework, the programming languages, and any integration points like APIs or third-party dependencies. Gaining this context is critical to understanding the intended behaviour of the application, the data flows and potential areas where security vulnerabilities might arise.

Threat Modelling

Before initiating the actual assessment, a structured threat modelling exercise is conducted to map out potential attack vectors across the code landscape. This includes identifying security controls implemented in the code (authentication, authorization, integrations, etc.) and understanding how user inputs are processed. The goal is to pinpoint where insecure coding patterns might lead to exploitable conditions.

Security Control and Test Cases

1. **Authentication and Session Management** - This category focuses on verifying that authentication and session handling mechanisms including Single Sign-On (SSO) implementations and federated identity integrations are implemented correctly. It includes testing for hard-code storage of credentials, proper session validation and flags for session identifiers, proper authentication validation before CRUD operations etc.
2. **Access Control & Business Logic** – This category ensures that access checks are consistently enforced in the code. The review identifies missing authorization checks, improper role validation, and potential privilege escalation through direct object or function-level access along with potential bypasses to application business logic.
3. **API and Third-Party Integrations** – This category focuses on the security of integrated APIs, and third-party connections. It includes checking for lack of API validation around external service calls and if integrated components introduce vulnerabilities to the application.

The list provided is not limited to but represents core test cases, but the high level methodology varies based on the logic/business use case of the services being tested.

Vulnerability Assessment

Copyright © Blueinfy

On the basis of resource attributes and control categories, a thorough vulnerability assessment is performed using human intelligence with the help of supportive semi-automatic tools. This detects the vulnerabilities residing in the code, thus giving the actionable item list from application security standpoint.

Mitigation Strategies

Based on the identified vulnerabilities, weaknesses, and overall risk posture—along with the system architecture and industry best practices—a comprehensive mitigation plan is formulated. This includes a set of actionable, prioritized recommendations outlining the security measures required to effectively harden the environment.

Actionable Report with Zero False Positives

A key deliverable of the assessment is a highly actionable, well-structured report designed to drive immediate remediation. The report is curated to maintain zero false positives and includes the following critical components: -

- Executive Summary
- Description of Discovered Vulnerabilities
- Risk Rating (curated after business impact assessment and industry security standards like CVSS/CWE/CVE)
- Evidence of Vulnerabilities (screenshots, HTTP traffic, file path, vulnerable parameter, exploit vector, tool results, reproduction steps etc.)
- Exploit Evidence of Vulnerabilities (if required)
- Mitigation Strategies and Defence Approaches (catered to help Developers)
- Report Readout and Guidance

Tools

Blueinfy uses its own tools along with open source tools and products during the review process. Blueinfy has its own tools and utilities for performing manual source code review. Some of these tools are available at https://www.blueinfy.com/tools.html

Copyright © Blueinfy