RPACT Secure Coding & Code Review Policy

RPACT GbR

POL-RPACT-003

Version 1.0.0

Contents

RPACT GbR

Am Rodenkathen 11
23611 Sereetz
Germany
www.rpact.com

Policy ID	POL-RPACT-003
Title	RPACT Secure Coding & Code Review Policy
Description	This policy defines the code review and secure coding standards followed by RPACT GbR to ensure that application code is free of vulnerabilities, malicious code, and other issues prior to production release.
Author	Friedrich Pahlke
Reviewer	Gernot Wassmer, Daniel Sabanés Bové
Creation date	2025-03-26
Version	1.0.0
Date of modification	2025-03-31
Effective date	2025-03-31

1 Purpose

This policy defines the code review and secure coding standards followed by RPACT GbR to ensure that application code is free of vulnerabilities, malicious code, and other issues prior to production release.

The level and scope of these practices are tailored to the type and risk level of the application being developed. Not all secure coding controls are relevant for all projects; for example, a computational R package like *rpact* does not process or store external data and therefore does not require measures like anti-automation or input sanitization in the same way as web applications do.

2 Scope

This policy applies to all software development projects managed by RPACT GbR, including both internal and client-related systems. The application of secure coding practices is **risk-based** and varies depending on the system architecture, data handling, and exposure to users or networks.

3 Responsibilities

- The RPACT partners are responsible for enforcing and conducting code reviews.
- All contributors must adhere to secure coding practices and are responsible for submitting code via approved workflows (e.g., Pull Requests).

4 Secure Coding Practices

Secure coding standards are applied in proportion to the type and risk profile of each project. The following guidance outlines applicable controls for typical system types:

4.1 Analytical Libraries (e.g., *rpact*)

These systems are computational packages without GUI or database/network connections. They are run inside the R environment (R Project) and are therefore not directly installed into or interfacing with the operating system.

- All public APIs are unit tested.
- Numerical stability and reproducibility are ensured.
- Input validation is applied only as needed for consistency and to prevent user errors.
- Vulnerability to injection or automation is not applicable due to local, controlled use.

4.2 GUI Applications without Database (e.g., *RPACT Cloud*)

These systems may receive user input but do not store or process persistent data.

- User input is validated and sanitized before being used in calculations.
- Output validation is applied where output is displayed in dynamic elements.
- Security considerations focus on user session control, input throttling, and frontend integrity.

4.3 Applications with Data Storage or Network Integration (no current product, but future potential)

These systems would require full secure coding coverage, including:

- Input/output validation, encryption, authentication, and authorization controls
- Session handling, secure API use, and protection against automation or misuse
- Continuous review of third-party dependencies and updates
- Security-relevant code clearly documented and reviewed
- Logging and monitoring of access to critical components

5 Code Review Process

5.1 Review Prior to Production Release

- No code is released to production without a formal review.
- Code is submitted via Pull Requests and reviewed by at least one other qualified team member (typically one of the partners).
- The review includes functionality, readability, maintainability, and security.

5.2 Tool Support and Traceability

- All code is version-controlled (e.g. via GitHub).
- The commit history as well as the GitHub Pull Request history provide a transparent review trail.
- Reviews are documented via GitHub's built-in review tools, including approvals, comments, and commit tracking.

5.3 Review for Vulnerabilities and Malicious Code

- Code reviews specifically include checks for potentially vulnerable constructs or patterns.
- Static analysis tools and peer discussion are used to catch potential issues.
- No third-party contributions are merged without full internal review.
- All internal code is written by trusted contributors working under NDA.

6 Proportionality

- The depth of code review and documentation is adjusted to the risk level of the application component.
- All GxP-relevant code is subject to full validation and traceability.

7 Review of Policy

This policy will be reviewed annually or following any major change to development or review practices.

Approved by:

Dr. Friedrich Pahlke
Prof. Dr. Gernot Wassmer

Date: *2025-03-31*
