Code Review Security Checklist

SafeStack

Please send all feedback to hello@safestack.io

Before pushing code to the team repository

Have all secrets been removed from the committed code?
- ❑ Yes

Before completing the code review

Have unresolved risks been raised and documented?
- ❑ Yes

During a review of the code (with author, reviewers, tester)

Have the right people been engaged to review the code?
- ❑ Yes

Is the purpose of the change stated and understood by the reviewers?
- ❑ Yes

Is there debug functionality in the code?
- ❑ No
- ❑ Yes, and it can only run in test environments.

Is user-supplied data:
- ❑ Validated before it is used or stored?
- ❑ Escaped when it is passed to an interpreter?

Do log entries:
- ❑ Cover all key events and states?
- ❑ Include enough information to uniquely identify the event?
- ❑ Exclude secrets and customers' PII?

For frameworks, libraries, tools and other dependencies:
- ❑ Are they being used effectively?
- ❑ Have new dependencies been vetted?
- ❑ Are they up-to-date?

Do response messages:
- ❑ Make use of appropriate status codes?
- ❑ Exclude information that should remain internal to the system?
- ❑ Limit information to the correct level of authorization?

To testers:
- ❑ Is the test coverage sufficient?
- ❑ Are misuse cases represented?

Revised: **2020-02-20**

This checklist is not intended to be comprehensive. Additions and modifications to fit local practice are encouraged.

This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License.
To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/4.0/.

SafeStack

Please send all feedback to hello@safestack.io

Implementing the checklist

Purpose

Improve code review culture in consistently applying secure coding practices. The Checklist is not intended as a standalone teaching tool, an accountability mechanism, or as a complete guide to secure development.

How to run the checklist

The Checklist has three phases - before code is pushed, during the code review, and before the code review is marked complete. The Checklist itself can be included as a template in a code review request and the review tools configured to require its completion. It may still be helpful to have physical copies visible around teams' workstations.

The first phase takes place *before* the original author shares their code with the team and consists of the author verifying they haven't included any real passwords, keys, tokens, or other secrets in their code.

The next phase happens during review and each item may be completed by any of the reviewers besides the original author. The reviewers confirm the right people have been tagged in and that they all understand the intended change.

They then check for the presence of debug code, the handling of untrusted data and response information, the correct use of tools, and that there is sufficient log and test coverage.

Finally, if the code review has raised risks beyond the scope of the review to fix, the reviewers raise the risk to their team and ensure it is logged somewhere it will be reviewed. This can also be completed by any of the reviewers.

Modifying the checklist

Teams should modify the Checklist to suit their needs. They shouldn't remove safety steps because they are unable or unwilling to perform them. The entire team should be involved in decisions to modify the Checklist, and the modified Checklist tested on a single system to ensure it works as intended. Changes should result in a checklist that is focused, brief, actionable, collaborative, tested, and integrated.

Introducing the checklist

Identify a core group of people who are enthusiastic about improving their code review culture. Start with a single system and expand incrementally. Identify key quality metrics and measure them. Integrate the checklist directly into your build-test-release workflow.

For more detail see the Code Review Security Checklist Implementation Manual.

This work is licensed under the Creative Commons Attribution-ShareAlike 4.0 International License.
To view a copy of this license, visit http://creativecommons.org/licenses/by-sa/4.0/.