Cheat sheet: 8 security code review best practices

snyk

1. Sanitize and validate all input

Consider all input potentially malicious and sanitize it accordingly using a vetted library. Next to direct user input think of:

- data streams
- events
- files
- cookies
- system properties
- command line parameters

2. Never store credentials as code/config

Some good practices:

- Block sensitive data being pushed to GitHub by adding git-secrets, or its likes, as a git pre-commit hook.
- Break the build using the same tools.
- Audit for slipped secrets with GitRob or truffleHog.
- Use ENV variables for secrets in CI/CD and secret managers like Vault in production.

3. Enforce the least privilege principle

People and automated processes should only have access to the data they actually need. Test if a module can perform operations they're not entitled to perform.

4. Enforce secure authentication

- Assume they're not who they say they are.
- Enforce TLS and TLS client authentication.
- Re-authenticate before sensitive operations.
- Enforce password complexity.

5. Test for new vulnerabilities in your OS app dependencies

Check your dependencies for known issues and don't introduce new vulnerabilities — implement tests on your local machine and connect your git repository. Snyk helps you with this by providing tooling to scan throughout every stage of your SDLC.

6. Handle sensitive data with care

- Only store the data you need.
- Encrypt data that is sensitive, for example, PII and financial data.
- Use the correct strong encryption algorithm.
- Transport sensitive data only over TLS.
- Check your cookies and session data for sensitive information.

7. Protect against well-known attacks

Know how common attacks — like SQL injections and Cross-site scripting (XSS) — work and take that with you in a review. Review the OWASP Top 10 vulnerabilities and learn how to spot them.

8. Statically test your source code

Use a Static Application Security Testing (SAST) tool — like Snyk Code — to find security issues in your code. It is recommended to automate these checks as part of your pipeline or build process.

Brian Vermeer
@BrianVerm
Developer Advocate at Snyk

Trisha Gee
@trisha_gee
Java Champion & Developer Advocate at JetBrains

www.snyk.io