5 Tips to Supercharge App Security from Code to Cloud

Code/IDE

Code development
- Explanation: This step involves writing and developing code
- Recommended tooling:
  - Snyk IDE: Helps with real-time vulnerability scanning of code, OS libraries, containers and cloud infrastructure (https://snyk.io/platform/ide-plugins/).
- Pre-commit hooks:
  - GitGuardian CLI: run ggshield to detect hardcoded secrets and policy breaks.

Sample commands
- Install Snyk CLI: `npm install snyk -g`
- Install GitGuardian ggshield: `brew install gitguardian/tap/ggshield`
- Run Snyk CLI: `snyk test`, `snyk code test`, `snyk container test`, `snyk iac test`
- Run GitGuardian ggshield: `ggshield secret scan repo .`

Secrets management
- Encrypt your secrets using SOPS.
- Use a secrets manager (or a vault).
- Honeytoken in source code (How to Secure Your SCM Repositories with GitGuardian Honeytokens)

CI/CD

Continuous integration/continuous deployment
- Explanation: Automate the building, testing, and deployment of code changes.
- Recommended tooling:
  - SnykCI/CD integration to scan for vulnerabilities during the build process.
  - GitGuardian's ggshield gating the CI to ensure 0 secrets exposed in production.
- Scan for vulnerabilities and secrets in container images:
  - Snyk Open Source for hardcoded secrets, Snyk Code and Snyk Container for most secure images/packages
  - `ggshield secret scan docker`

Deploy

Cloud deployment
- Explanation: Deploy the code to a cloud environment via infrastructure as code

Cloud security monitoring
- Explanation: Monitor cloud environments for security risks, misconfigurations, and exposed secrets.
- Recommended tooling:
  - Snyk IaC unifies security visibility and governance from IDE to running cloud environments using a single policy engine and rule set.
  - Logging, threat detection
  - Secrets best practices: IAM, rotation, short-lived secrets
- Cloud Security Essentials (https://snyk.io/series/cloud-security)

Merge/Git

Code Integration
- Explanation: Integrate code changes with the main branch of the repository
- Recommended Tooling:
  - GitGuardian Check Run: Scans and ensures secrets are not inadvertently merged into the main or feature branch during integration

Automated Code Review
- Explanation. Review the code for quality, security, and adherence to coding standards
- Recommended Tooling:
  - Snyk Open Source: Helps find and fix security vulnerabilities and license issues in OS dependencies
  - Snyk Code: real-time SAST
  - Snyk Automatic Pull Requests for Snyk Open Source
  - GitGuardian: Detects secrets and sensitive information in code during the review process

Tips to Supercharge App Security from Code to Cloud

Sign up for GitGuardian

Sign up for Snyk