GitHub Advanced Security (GHAS) Best Practices

Quick Reference for Engineering Teams

Purpose. Practical, auditable practices to get sustained value from GHAS across repos and teams.

Speak the language of vulnerabilities

- **CVE** = specific known vulnerability; **CWE** = broader weakness class; **CVSS** informs urgency.
- Link GHAS alerts to CVE/CWE, read remediation notes, act by severity and exploitability.

Turn on the full GHAS stack (right-sized)

- Enable and tune **Dependabot**, **Secret scanning**, and **CodeQL**.
- Prefer repo-specific config files over defaults to reduce noise.

Close vs. Dismiss: decision guardrails

- **Close** when fixed and scanners rerun clean; document "what changed" for auditability.
- **Dismiss** (with reason) only for accepted risk, false positives, or unreachable code paths; review periodically.

Clear roles and collaboration loop

- **Developers:** secure coding, unit tests, resolve code scanning alerts, collaborate on threat modeling.
- **Security:** define policy, maintain rules, run audits/pen-tests, coach developers.

Cadence that matches risk

- High-risk or fast-moving apps: **weekly** reviews; lower-risk apps: **monthly**.
- Adapt cadence for new threats, major releases, and exposure changes.

Put policy in the repo (and surface it)

- Add `SECURITY.md` (root or `.github/`) describing requirements and reporting process.
- Reinforce via short enablement sessions or "lunch and learns."

Code 1: `SECURITY.md` starter skeleton

```
# Security Policy

## Reporting a Vulnerability
Please email security@[yourorg].com with details and steps to reproduce.
Do not create public issues for suspected vulnerabilities.

## Supported Versions
- main: actively maintained
- previous: critical fixes only

## Requirements
- 2FA required for committers
- All PRs: CodeQL + secrets + dependency checks must pass

## Disclosure
We follow coordinated disclosure. We acknowledge within 48h and provide
status updates until resolution.
```

Make scanning policy-driven

- Align **CodeQL** queries and thresholds to written policy; keep a lean baseline and add targeted custom rules.
- Treat alerts as actionable work items with owners and SLAs, not informational noise.

Gate merges with branch protection

- Protect `main` and other critical branches:
  - Require PRs and at least one approval.
  - Require status checks to pass: `codeql`, `dependabot`, `secret-scanning`.
  - Dismiss stale reviews on new commits.
  - Restrict who can push; block force-pushes and branch deletion.

Notifications that matter

- Route **critical** security events to the right responders immediately (avoid over-alerting).

Stay plugged into the community

- Track advisories and research from sources like the GitHub Security Lab; fold insights into rules and runbooks.

Build note: This document uses the `minted` package. Compile with: `pdflatex -shell-escape ghas-best-practices.tex` (or via `latexmk -pdf -shell-escape`).