Dependabot Alerts & Workflows — Hands-On Cheatsheet

Heads-up: This document uses `minted` for syntax highlighting. If you want actual GraphQL highlighting, install the Pygments lexer plugin `pygments-graphql` and switch the block back to `graphql`. Compile with: `pdflatex -shell-escape dependabot-cheatsheet.tex` (or `xelatex` with `-shell-escape`).

1 Core Concepts

- **Dependency types:** Direct dependencies are packages your code imports/uses. *Transitive* dependencies are packages required by your direct dependencies.
- **Where alerts appear:** Security tab (repository overview), Pull Requests (as comments/checks), and via API/CLI.
- **Security updates:** Dependabot can automatically open PRs to bump vulnerable versions.

2 Enable Security Updates

Add or adjust `dependabot.yml` at repo/org level to scan ecosystems and open PRs automatically.

Minimal dependabot.yml

```yaml
version: 2
updates:
  - package-ecosystem: "pip"        # npm, maven, gradle, gomod, etc.
    directory: "/"                  # location of manifest
    schedule:
      interval: "daily"             # daily, weekly, monthly
    open-pull-requests-limit: 5
    rebase-strategy: auto
    reviewers: ["team/security"]
    labels: ["dependencies","security"]
```

3 Triage & Remediation Workflow

1. Review new alerts in **Security** → **Alerts** or in the PR that raised them.
2. Validate severity, affected manifests, and suggested upgrade.
3. Run tests locally and/or in CI (unit, integration, regression).
4. Merge the Dependabot PR (or create one via *Update now*) and monitor post-merge checks.

4 Query Alerts Programmatically

4.1 GraphQL API (curl)

Query via GraphQL (replace TOKEN/owner/repo)

```
curl -s -H "Authorization: bearer TOKEN" \
     -X POST https://api.github.com/graphql \
     -d '{"query":"
query {
  repository(owner:\"OWNER\", name:\"REPO\") {
    vulnerabilityAlerts(first: 20) {
      nodes {
        securityVulnerability { severity package { name } }
        vulnerableManifestFilename
        vulnerableRequirements
        createdAt
        state
      }
    }
  }
}"}' | jq .
```

4.2 GraphQL Shape

GraphQL selection set sketch

```
query {
  repository(owner: "OWNER", name: "REPO") {
    vulnerabilityAlerts(first: 20) {
      nodes {
        securityVulnerability { severity package { name } }
        vulnerableManifestFilename
        vulnerableRequirements
        createdAt state
      }
    }
  }
}
```

4.3 REST via GitHub CLI

Fetch Dependabot alerts with gh api

```
# Authenticate once
gh auth login --web --hostname github.com

# List alerts (REST): /repos/{owner}/{repo}/dependabot/alerts
gh api repos/OWNER/REPO/dependabot/alerts --paginate > alerts.json

# Convert to CSV (example jq shape)
jq -r '.[] | [
  .number,
  .state,
  .dependency.package.name,
  .security_advisory.severity,
  .security_vulnerability.severity,
  .created_at
] | @csv' alerts.json > alerts.csv
```

5 Working From the Security Tab or PR

- **Security tab:** Review alert details and use *Update now* to spawn a security update PR.
- **Pull Request:** Dependabot comments highlight the vulnerable dependency and the safe version.
- **Change control:** PR discussion and reviews record decisions and provide traceability.

6 Advisory Intelligence

- Browse the **GitHub Advisory Database** (filter by ecosystem, severity, CWE, newest).
- Use it to proactively check dependencies and plan upgrade windows.

7 Dependabot CLI and Core

- **Dependabot Core:** Core logic is open source; backend services are proprietary to GitHub.
- **CLI:** Useful for local testing/debugging of update jobs.

8 Automation Tips

- Ensure unit/integration tests run on Dependabot PRs.
- Consider grouping updates to reduce PR noise when appropriate.
- Label, auto-assign reviewers, and protect main branches with required checks.

9 Quick Commands

```
# Authenticate GitHub CLI (web flow)
gh auth login --web --hostname github.com

# List Dependabot alerts (REST)
gh api repos/OWNER/REPO/dependabot/alerts

# Example: narrow by severity (jq)
gh api repos/OWNER/REPO/dependabot/alerts \
 | jq '.[] | select(.security_advisory.severity=="critical")'
```