Dependabot & Dependency Graph — 1 Page Cheat Sheet

Quick reference for GHAS dependency risk management

Core Concepts

Vulnerability. A weakness in software, hardware, or systems that attackers can exploit to gain access, steal data, or disrupt operations. Examples include buffer overflows that allow code injection.

Dependabot Alerts. Automatic findings when dependencies in your repo match known vulnerable versions. Alerts include metadata, severity, and links to remediation guidance.

Dependabot Security Updates vs Version Updates.

- **Security Updates:** When a vulnerable version is detected and a safe version exists, Dependabot opens a PR to patch it.
- **Version Updates:** Dependabot proactively opens PRs to keep dependencies current as new versions are released (not just during vulnerability scans).

Dependency Graph (SBOM). A dynamic inventory of your repository's dependencies and their relationships; integrates with Dependabot and supports SBOM export.

How It Works (At a Glance)

1. Repo manifests (e.g., `requirements.txt`, `package-lock.json`, `pom.xml`) are analyzed.
2. Dependabot builds the *dependency graph* and tracks versions.
3. Versions are compared against multiple sources (e.g., NVD, vendor advisories, package registries).
4. When a match with a known vulnerable version is found, Dependabot issues an alert and, if enabled, opens a PR.

Quick Start: `.github/dependabot.yml`

Place at repo root in `.github/`*. Compile this doc with* `-shell-escape` *to enable syntax highlighting.*

```yaml
version: 2
updates:
  - package-ecosystem: "pip"          # npm, maven, gradle, cargo, etc.
    directory: "/"                    # location of manifest (e.g., /app)
    schedule:
      interval: "weekly"              # daily | weekly | monthly
      day: "monday"
      time: "09:00"
    open-pull-requests-limit: 5
    reviewers:
      - "org/security-reviewers"
    labels: ["dependabot", "security"]
    ignore:
      - dependency-name: "pytest"
        versions: ["< 5.0.0"]
```

Notify Chat Platforms (Example)

Idea: post new alerts or open PRs to Teams via webhook (similar patterns work for Slack).

```
# In a workflow step, send a message with curl (Teams Incoming Webhook)
curl -X POST "$TEAMS_WEBHOOK_URL" \
  -H 'Content-Type: application/json' \
  -d '{
    "text": "Dependabot found a vulnerable dependency. See Security tab."
  }'
```

Operational Tips

- **Act early.** Treat alerts like code reviews; triage continuously to reduce risk and PR backlog.
- **Tune noise.** Limit open Dependabot PRs per repo; batch schedules to avoid alert floods.
- **Label & route.** Auto-label Dependabot PRs, assign reviewers, and wire notifications to Teams/Slack.
- **Dismiss responsibly.** Use consistent dismissal reasons (e.g., false positive, already remediated, will not fix with rationale).

Exam/Interview Recall

- Dependabot sources include public vulnerability feeds (e.g., NVD), vendor advisories, package registries, partner feeds, community reports, and GitHub research.
- The **dependency graph** powers SBOM export and links findings to manifests.
- **Security updates** vs **version updates**: both make PRs; security updates react to vulnerabilities, version updates keep you current.

Compile with: `latexmk -pdf -shell-escape -interaction=nonstopmode ghas-dependabot-cheatsheet-minted.tex`