**4. Security and compliance**
└─ **4.4 Security and compliance scanning**
└─ **4.4.5 Dependabot for OSS scanning**

1. What is Dependabot and how does it help with open-source security?
2. How do you enable and configure Dependabot in a GitHub repository?
3. What file types and ecosystems does Dependabot support?
4. How does Dependabot detect and report vulnerabilities?
5. What are Dependabot security updates vs version updates?
6. How are vulnerabilities prioritized in Dependabot alerts?
7. How can PRs from Dependabot be managed and approved?
8. How do you control update frequency and behavior in Dependabot?
9. Where are Dependabot alerts shown and how are they resolved?
10. What are best practices for integrating Dependabot in CI/CD?

**1. What is Dependabot and how does it help with open-source security?**
Dependabot scans dependencies for known vulnerabilities and automatically creates pull requests to upgrade to secure versions.

**2. How do you enable and configure Dependabot in a GitHub repository?**
Create a .github/dependabot.yml file defining package ecosystems, directories, and update frequency. Security updates are enabled by default in repo settings.

**3. What file types and ecosystems does Dependabot support?**
It supports npm, Maven, pip, RubyGems, NuGet, Cargo, Go modules, Docker, and more—based on lockfiles or manifest files in the repo.

**4. How does Dependabot detect and report vulnerabilities?**
It matches dependency versions in the repo against the GitHub Advisory Database. If a match exists, it creates alerts and optionally PRs.

**5. What are Dependabot security updates vs version updates?**

- Security updates target known vulnerabilities.
- Version updates are routine upgrades (e.g., minor/patch bumps).

Both are configured via dependabot.yml.

**6. How are vulnerabilities prioritized in Dependabot alerts?**
Alerts are ranked by CVSS severity (low to critical) and show impact details, affected version ranges, and fixed versions.

**7. How can PRs from Dependabot be managed and approved?**
PRs can be auto-approved using *GitHub Actions* or require manual review. Apply branch protection rules to enforce checks before merge.

**8. How do you control update frequency and behavior in Dependabot?**
In dependabot.yml, set schedule.interval (daily/weekly/monthly), and open-pull-requests-limit to manage PR volume.

**9. Where are Dependabot alerts shown and how are they resolved?**
Alerts appear in the *Security > Dependabot* tab. Resolve them by merging the auto-generated PR or manually updating dependencies.

---

**10. What are best practices for integrating Dependabot in CI/CD?**
Enable alerts, review PRs promptly, restrict PRs to safe branches, run tests on PRs, and auto-merge low-risk updates after passing checks.