# GITHUB ADVANCED SECURITY CERTIFICATION PRACICE TEST (SAMPLE)

## STUDY GUIDE

## SAMPLE STUDY GUIDE WITH LIMITED QUESTIONS

**VISIT US ONLINE FOR THE FULL VERSION STUDY GUIDE**

https://githubadvsec.examzify.com

# Table of Contents

# Introduction

Preparing for a certification exam can feel overwhelming, but with the right tools, it becomes an opportunity to build confidence, sharpen your skills, and move one step closer to your goals. At Examzify, we believe that effective exam preparation isn't just about memorization, it's about understanding the material, identifying knowledge gaps, and building the test-taking strategies that lead to success.

This guide was designed to help you do exactly that.

Whether you're preparing for a licensing exam, professional certification, or entry-level qualification, this book offers structured practice to reinforce key concepts. You'll find a wide range of multiple-choice questions, each followed by clear explanations to help you understand not just the right answer, but why it's correct.

The content in this guide is based on real-world exam objectives and aligned with the types of questions and topics commonly found on official tests. It's ideal for learners who want to:

- Practice answering questions under realistic conditions,
- Improve accuracy and speed,
- Review explanations to strengthen weak areas, and
- Approach the exam with greater confidence.

We recommend using this book not as a stand-alone study tool, but alongside other resources like flashcards, textbooks, or hands-on training. For best results, we recommend working through each question, reflecting on the explanation provided, and revisiting the topics that challenge you most.

Remember: successful test preparation isn't about getting every question right the first time, it's about learning from your mistakes and improving over time. Stay focused, trust the process, and know that every page you turn brings you closer to success.

Let's begin.

# How to Use This Guide

This guide is designed to help you study more effectively and approach your exam with confidence. Whether you're reviewing for the first time or doing a final refresh, here's how to get the most out of your Examzify study guide:

### 1. Start with a Diagnostic Review

Skim through the questions to get a sense of what you know and what you need to focus on. Your goal is to identify knowledge gaps early.

### 2. Study in Short, Focused Sessions

Break your study time into manageable blocks (e.g. 30 – 45 minutes). Review a handful of questions, reflect on the explanations.

### 3. Learn from the Explanations

After answering a question, always read the explanation, even if you got it right. It reinforces key points, corrects misunderstandings, and teaches subtle distinctions between similar answers.

### 4. Track Your Progress

Use bookmarks or notes (if reading digitally) to mark difficult questions. Revisit these regularly and track improvements over time.

### 5. Simulate the Real Exam

Once you're comfortable, try taking a full set of questions without pausing. Set a timer and simulate test-day conditions to build confidence and time management skills.

### 6. Repeat and Review

Don't just study once, repetition builds retention. Re-attempt questions after a few days and revisit explanations to reinforce learning. Pair this guide with other Examzify tools like flashcards, and digital practice tests to strengthen your preparation across formats.

There's no single right way to study, but consistent, thoughtful effort always wins. Use this guide flexibly, adapt the tips above to fit your pace and learning style. You've got this!

# Questions

1. **Which statement best describes how Code Scanning findings relate to remediation?**

   A. Findings are local to the repo and point to issues to fix

   B. Findings replace vulnerability advisories

   C. Findings automatically fix issues

   D. Findings do not indicate remediation steps

2. **Which statement about Dependabot alerts and Dependabot security updates is accurate?**

   A. Alerts notify about known vulnerabilities; updates automatically creates PRs to fix them.

   B. Alerts update the dependencies directly.

   C. Alerts replace the need for manual code review.

   D. Alerts notify about known vulnerabilities; Security updates automatically creates PRs to fix them.

3. **How can you assign advisories to a team for remediation?**

   A. Email everyone individually after the release.

   B. Use the Security tab to assign to a team or maintainers; include relevant parties in the advisory.

   C. Create an issue and assign it to no one.

   D. Keep it unassigned to avoid accountability.

4. **What does Security Overview provide?**

   A. A centralized view of code/secret scanning and dependabot alerts across your codebases

   B. A list of all vulnerabilities

   C. A manual code review tool

   D. A history of commits

5. **How does GAS support compliance across a software supply chain?**

   A. By providing SBOMs, dependency risk insights, and an auditable security process including advisories and vulnerability management.

   B. By printing compliance certificates for each release.

   C. By restricting all third-party libraries to a single vendor.

   D. By minimizing reporting and hiding vulnerabilities.

**6. After addressing a high-severity Code Scanning alert, what is the recommended final step?**

A. Dismiss the alert and keep it open

B. Remove the file from the repository

C. Disable Code Scanning for the repository

D. Remediate, then close the alert

**7. Push Protection for Secrets is primarily concerned with preventing commits to which branch when a secret is identified?**

A. Main branch

B. Feature branch

C. Develop branch

D. Release branch

**8. How can you differentiate Code Scanning alerts from Secret Scanning alerts in the UI?**

A. They originate from the same scanning process and are shown under the same section.

B. Code Scanning alerts originate from CodeQL queries; Secret Scanning alerts come from secret detection in code or history; they appear under different sections in Security.

C. They are both shown under the 'Code Scanning' tab only.

D. Secret Scanning alerts never appear in the UI.

**9. Dependabot Alerts are designed to do what?**

A. Notify you about malware

B. Notify you about vulnerable dependencies in public repositories

C. Notify you about license changes

D. Notify you about new issues

**10. What are the recommended language coverage and scanning frequency for CodeQL in a typical GitHub repository using GAS?**

A. Cover a subset of languages and run scans monthly

B. Only scan on PRs

C. Cover all languages used; run scans on push and PRs (and nightly if possible)

D. Disable nightly scans

# Answers

1. A
2. D
3. B
4. A
5. A
6. D
7. A
8. B
9. B
10. C

# Explanations

## 1. Which statement best describes how Code Scanning findings relate to remediation?

A. Findings are local to the repo and point to issues to fix

B. Findings replace vulnerability advisories

C. Findings automatically fix issues

D. Findings do not indicate remediation steps

*Code Scanning findings center on issues inside your own repository, showing exactly where something needs attention and what to fix. They surface alerts from code analysis tools and point to the files and lines involved, often including guidance on how to remediate or mitigate the issue. Because these findings live in your repo, they help developers prioritize and track fixes within the project's own codebase, making remediation a team and workflow concern rather than something external. They aren't meant to replace vulnerability advisories, which are external notices about known issues in libraries or ecosystems. Findings also don't automatically fix problems; remediation requires someone to implement the suggested changes or apply fixes through a CI/CD pipeline. Importantly, these findings typically provide remediation guidance, so saying they don't indicate remediation steps isn't accurate.*

## 2. Which statement about Dependabot alerts and Dependabot security updates is accurate?

A. Alerts notify about known vulnerabilities; updates automatically creates PRs to fix them.

B. Alerts update the dependencies directly.

C. Alerts replace the need for manual code review.

D. Alerts notify about known vulnerabilities; Security updates automatically creates PRs to fix them.

*Dependabot splits its functionality into alerts and security updates. Alerts warn you when a dependency has a known vulnerability, but they don't change your code by themselves. Security updates are the automated part that creates pull requests to bump the vulnerable dependency to a patched version. So the statement that alerts notify about known vulnerabilities and security updates automatically create PRs to fix them accurately describes how the two features work together. The other ideas aren't correct because alerts don't update dependencies directly, and alerts don't replace the need for review—the generated PRs (from security updates) still go through review before merging.*

## 3. How can you assign advisories to a team for remediation?

A. Email everyone individually after the release.

B. Use the Security tab to assign to a team or maintainers; include relevant parties in the advisory.

C. Create an issue and assign it to no one.

D. Keep it unassigned to avoid accountability.

*Assigning advisories to the right people is essential for a coordinated remediation. In GitHub, you manage who handles an advisory by using the Security tab to assign it to a team or to specific maintainers, and by including all relevant parties in the advisory. This creates clear ownership, ensures notifications reach the right people, and keeps the remediation process tracked in one place with status updates and discussion. Doing it by emailing individuals one by one is error-prone and hard to track, and creating an issue without an assignee or leaving the advisory unassigned creates ambiguity about responsibility and can delay fixes.*

## 4. What does Security Overview provide?

A. A centralized view of code/secret scanning and dependabot alerts across your codebases

B. A list of all vulnerabilities

C. A manual code review tool

D. A history of commits

*Security Overview provides a centralized view of code/secret scanning and Dependabot alerts across your codebases. This dashboard collects security signals from multiple sources into one place, so you can quickly see overall risk, triage issues, and track remediation across all repositories. It isn't just a list of vulnerabilities, nor a manual code review tool, nor a history of commits.*

## 5. How does GAS support compliance across a software supply chain?

A. By providing SBOMs, dependency risk insights, and an auditable security process including advisories and vulnerability management.

B. By printing compliance certificates for each release.

C. By restricting all third-party libraries to a single vendor.

D. By minimizing reporting and hiding vulnerabilities.

*GAS supports compliance across the software supply chain by providing visibility into what components are used, how risky those components are, and a traceable security workflow. It generates Software Bill of Materials (SBOMs), which list every component, library, and license in a release, enabling you to verify exactly what's included and to demonstrate compliance during audits. It also offers dependency risk insights that flag known vulnerabilities, outdated components, and exposure levels, helping teams prioritize remediation and reduce supply-chain risk. In addition, GAS establishes an auditable security process with advisories and vulnerability management, creating records of identified issues, remediation steps, and approval workflows that auditors can review. Together, these elements give the transparency, risk visibility, and traceability needed to maintain compliance across complex software supply chains and across multiple teams and vendors. Printing compliance certificates, restricting to a single vendor, or hiding vulnerabilities do not provide the same level of visibility, governance, or trust essential for compliance in a software supply chain.*

## 6. After addressing a high-severity Code Scanning alert, what is the recommended final step?

A. Dismiss the alert and keep it open

B. Remove the file from the repository

C. Disable Code Scanning for the repository

D. Remediate, then close the alert

*After addressing a high-severity Code Scanning alert, the final step is to fix the underlying issue and then close the alert once the fix is verified. This keeps the alert status accurate, signals to the team that the risk has been mitigated, and provides a traceable record of what was changed and why. It's important to re-run the scan or review the changes in a pull request to confirm the remediation actually resolves the vulnerability before closing. Dismissing the alert and keeping it open, removing the file, or disabling Code Scanning all undermine security visibility and protection, whereas remediating and then closing ensures proper triage and accountability.*

## 7. Push Protection for Secrets is primarily concerned with preventing commits to which branch when a secret is identified?

A. Main branch

B. Feature branch

C. Develop branch

D. Release branch

*Push Protection for Secrets targets stopping sensitive credentials from entering the production code path. The main branch is typically the default, most protected, and the branch that represents release-ready code. If a secret sneaks into main, it can end up in release artifacts, CI logs, and distributed code, exposing credentials to a wide audience. By blocking commits that contain secrets to the main branch, the system enforces removing the secret before it can contaminate the stable codebase. This encourages rotating the secret and pushing a clean update, while development can continue on other branches without immediate blockage. In short, the main branch is the primary guardrail because it's the version of the code most likely to be deployed and shared, so preventing secrets there minimizes exposure risk.*

## 8. How can you differentiate Code Scanning alerts from Secret Scanning alerts in the UI?

A. They originate from the same scanning process and are shown under the same section.

B. Code Scanning alerts originate from CodeQL queries; Secret Scanning alerts come from secret detection in code or history; they appear under different sections in Security.

C. They are both shown under the 'Code Scanning' tab only.

D. Secret Scanning alerts never appear in the UI.

*Code scanning and secret scanning come from different scanning processes and are surfaced in separate parts of the Security UI. Code scanning alerts come from CodeQL analysis runs, while secret scanning alerts come from detecting secrets in code or in history. Because they originate from distinct checks, GitHub places them in different sections under Security—code scanning alerts in the Code scanning area and secret scanning alerts in the Secrets/Secret scanning area. This separation helps you triage by type and know which tool flagged the issue. The other options imply they share the same section, or that secret alerts never appear, which isn't accurate.*

## 9. Dependabot Alerts are designed to do what?

A. Notify you about malware

B. Notify you about vulnerable dependencies in public repositories

C. Notify you about license changes

D. Notify you about new issues

*Dependabot Alerts are about keeping your project secure by watching the dependencies it uses. They scan the dependency manifests and lockfiles in public repositories and compare the versions you rely on against a database of known security advisories. When a vulnerability is found in one of those dependencies, an alert is raised with details and recommended fixed versions, so you can update promptly and reduce risk. This is different from warnings about malware, license changes, or general new issues, which are not specifically about vulnerabilities in dependencies.*

**10. What are the recommended language coverage and scanning frequency for CodeQL in a typical GitHub repository using GAS?**

A. Cover a subset of languages and run scans monthly

B. Only scan on PRs

C. Cover all languages used; run scans on push and PRs (and nightly if possible)

D. Disable nightly scans

*Cover all languages used in the repository so CodeQL can analyze every part of the codebase, not just a subset. Running scans on push ensures each new commit is examined for vulnerabilities right away, while scanning on pull requests checks the changes being proposed before they're merged, catching issues early in the review process. If possible, enabling nightly scans adds a valuable safety net, catching problems that can arise from dependency updates or files that change indirectly, even when there isn't a new push or PR that day. This approach minimizes gaps in detection and keeps security visibility high across the entire codebase.*

# Next Steps

Congratulations on reaching the final section of this guide. You've taken a meaningful step toward passing your certification exam and advancing your career.

As you continue preparing, remember that consistent practice, review, and self-reflection are key to success. Make time to revisit difficult topics, simulate exam conditions, and track your progress along the way.

If you need help, have suggestions, or want to share feedback, we'd love to hear from you. Reach out to our team at hello@examzify.com.

Or visit your dedicated course page for more study tools and resources:

https://githubadvsec.examzify.com

We wish you the very best on your exam journey. You've got this!