Download Microsoft GH-500 Exam Dumps Free

Shared by Solis on 17-06-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

-- [Describe GitHub Advanced Security Best Practices]

As a contributor, you discovered a vulnerability in a repository. Where should you look for the instructions on how to report the vulnerability?

Options:

A- support.md
B- readme.md
C- contributing.md
D- security.md

Answer:

D

Explanation:

The correct place to look is the SECURITY.md file. This file provides contributors and security researchers with instructions on how to responsibly report vulnerabilities. It may include contact methods, preferred communication channels (e.g., security team email), and disclosure guidelines.

This file is considered a GitHub best practice and, when present, activates a ''Report a vulnerability'' button in the repository's Security tab.

Question 2

Question Type: MultipleChoice

-- [Configure and Use Secret Scanning]

Secret scanning will scan:

Options:

A- A continuous integration system.
B- Any Git repository.
C- The GitHub repository.
D- External services.

Answer:

C

Explanation:

Secret scanning is a feature provided by GitHub that scans the contents of your GitHub repositories for known types of secrets, such as API keys and tokens. It operates within the GitHub environment and does not scan external systems, services, or repositories outside of GitHub. Its primary function is to prevent the accidental exposure of sensitive information within your GitHub-hosted code.

Question 3

Question Type: MultipleChoice

-- [Configure and Use Dependency Management]

A dependency has a known vulnerability. What does the warning message include?

Options:

A- The security impact of these changes
B- An easily understandable visualization of dependency change
C- How many projects use these components
D- A brief description of the vulnerability

Answer:

D

Explanation:

When a vulnerability is detected, GitHub shows a warning that includes a brief description of the vulnerability. This typically covers the name of the CVE (if available), a short summary of the

issue, severity level, and potential impact. The message also links to additional advisory data from the GitHub Advisory Database.

This helps developers understand the context and urgency of the vulnerability before applying the fix.

Question 4

Question Type: MultipleChoice

-- [Describe the GHAS Security Features and Functionality]

What is a security policy?

Options:

A- An automatic detection of security vulnerabilities and coding errors in new or modified code
B- A security alert issued to a community in response to a vulnerability
C- A file in a GitHub repository that provides instructions to users about how to report a security vulnerability
D- An alert about dependencies that are known to contain security vulnerabilities

Answer:

C

Explanation:

A security policy is defined by a SECURITY.md file in the root of your repository or .github/ directory. This file informs contributors and security researchers about how to responsibly report vulnerabilities. It improves your project's transparency and ensures timely communication and mitigation of any reported issues.

Adding this file also enables a ''Report a vulnerability'' button in the repository's Security tab.

Question 5

Question Type: MultipleChoice

-- [Configure and Use Dependency Management]

In the pull request, how can developers avoid adding new dependencies with known vulnerabilities?

Options:

A- Enable Dependabot alerts.
B- Add Dependabot rules.
C- Add a workflow with the dependency review action.
D- Enable Dependabot security updates.

Answer:

C

Explanation:

To detect and block vulnerable dependencies before merge, developers should use the Dependency Review GitHub Action in their pull request workflows. It scans all proposed dependency changes and flags any packages with known vulnerabilities.

This is a preventative measure during development, unlike Dependabot, which reacts after the fact.

Question 6

Question Type: MultipleChoice

-- [Use Code Scanning with CodeQL]

When using CodeQL, what extension stores query suite definitions?

Options:

A- .yml
B- .ql
C- .qll
D- .qls

Answer:

D

Explanation:

Query suite definitions in CodeQL are stored using the .qls file extension. A query suite defines a collection of queries to be run during an analysis and allows for grouping them based on categories like language, security relevance, or custom filters.

In contrast:

.ql files are individual queries.

.qll files are libraries used by .ql queries.

.yml is used for workflows, not query suites.

Question 7

Question Type: MultipleChoice

-- [Assessing Code Scanning Alerts]

You are managing code scanning alerts for your repository. You receive an alert highlighting a problem with data flow. What do you click for additional context on the alert?

Options:

A- Show paths
B- Security
C- Code scanning alerts

Answer:

A

Explanation:

When dealing with a data flow issue in a code scanning alert, clicking on 'Show paths' provides a detailed view of the data's journey through the code. This includes the source of the data, the path it takes, and where it ends up (the sink). This information is crucial for understanding how

untrusted data might reach sensitive parts of your application and helps in identifying where to implement proper validation or sanitization.

Question 8

Question Type: MultipleChoice

-- [Use Code Scanning with CodeQL]

Which of the following options are code scanning application programming interface (API) endpoints? (Each answer presents part of the solution. Choose two.)

Options:

A- List all open code scanning alerts for the default branch
B- Modify the severity of an open code scanning alert
C- Get a single code scanning alert
D- Delete all open code scanning alerts

Answer:

A, C

Explanation:

The GitHub Code Scanning API includes endpoints that allow you to:

List alerts for a repository (filtered by branch, state, or tool) --- useful for monitoring security over time.

Get a single alert by its ID to inspect its metadata, status, and locations in the code.

However, GitHub does not support modifying the severity of alerts via API --- severity is defined by the scanning tool (e.g., CodeQL). Likewise, alerts cannot be deleted via the API; they are resolved by fixing the code or dismissing them manually.

Question 9

Question Type: MultipleChoice

-- [Use Code Scanning with CodeQL]

Which CodeQL query suite provides queries of lower severity than the default query suite?

Options:

A- github/codeql-go/ql/src@main
B- github/codeql/cpp/ql/src@main
C- security-extended

Answer:

C

Explanation:

The security-extended query suite includes additional CodeQL queries that detect lower severity issues than those in the default security-and-quality suite.

It's often used when projects want broader visibility into code hygiene and potential weak spots beyond critical vulnerabilities.

The other options listed are paths to language packs, not query suites themselves.

Question 10

Question Type: MultipleChoice

-- [Configure and Use Dependency Management]

Which security feature shows a vulnerable dependency in a pull request?

Options:

A- Dependency graph
B- Dependency review
C- Dependabot alert
D- The repository's Security tab

Answer:

B

Explanation:

Dependency review runs as part of a pull request and shows which dependencies are being added, removed, or changed --- and highlights vulnerabilities associated with any added packages.

It works in real-time and is specifically designed for use during pull request workflows.

The dependency graph is an overview, Dependabot alerts notify post-merge, and the Security tab shows the aggregated alert list.

Question 11

Question Type: MultipleChoice

-- [Configure and Use Secret Scanning]

Which option best secret scanning features can verify whether a secret is still active?

Options:

A- Push protection
B- Validity checks
C- Branch protection
D- Custom patterns

Answer:

B

Explanation:

Validity checks, also called secret validation, allow GitHub to check if a detected secret is still active. If verified as live, the alert is marked as 'valid', allowing security teams to prioritize the most critical leaks.

Push protection blocks secrets but does not check their validity. Custom patterns are user-defined and do not include live checks.

Question 12

Question Type: MultipleChoice

-- [Configure and Use Secret Scanning]

Which option best is the best way to prevent developers from adding secrets to the repository?

Options:

A- Create a CODEOWNERS file
B- Make the repository public
C- Configure a security manager
D- Enable push protection

Answer:

D

Explanation:

The best proactive control is push protection. It scans for secrets during a git push and blocks the commit before it enters the repository.

Other options (like CODEOWNERS or security managers) help with oversight but do not prevent secret leaks.

Making a repo public would increase the risk, not reduce it.