GH-500 Reliable Test Forum | GH-500 Detailed Study Dumps

P.S. Free & New GH-500 dumps are available on Google Drive shared by NewPassLeader: https://drive.google.com/open?id=1aVlMPx5FBKqKey0Vh_LpAYmp6JqseWc3

We are here to lead you on a right way to the success in the Microsoft certification exam and save you from unnecessary hassle. Our GH-500 braindumps torrent are developed to facilitate our candidates and to validate their skills and expertise for the GH-500 Practice Test. We are determined to make your success certain in GH-500 real exams and stand out from other candidates in the IT field.

Microsoft GH-500 Exam Syllabus Topics:

Topic	Details
Topic 1	• Configure and use secret scanning: This domain targets DevOps Engineers and Security Analysts with the skills to configure and manage secret scanning. It includes understanding what secret scanning is and its push protection capability to prevent secret leaks. Candidates differentiate secret scanning availability in public versus private repositories, enable scanning in private repos, and learn how to respond appropriately to alerts. The domain covers alert generation criteria for secrets, user role-based alert visibility and notification, customizing default scanning behavior, assigning alert recipients beyond admins, excluding files from scans, and enabling custom secret scanning within repositories.
Topic 2	• Configure and use Dependabot and Dependency Review: Focused on Software Engineers and Vulnerability Management Specialists, this section describes tools for managing vulnerabilities in dependencies. Candidates learn about the dependency graph and how it is generated, the concept and format of the Software Bill of Materials (SBOM), definitions of dependency vulnerabilities, Dependabot alerts and security updates, and Dependency Review functionality. It covers how alerts are generated based on the dependency graph and GitHub Advisory Database, differences between Dependabot and Dependency Review, enabling and configuring these tools in private repositories and organizations, default alert settings, required permissions, creating Dependabot configuration files and rules to auto-dismiss alerts, setting up Dependency Review workflows including license checks and severity thresholds, configuring notifications, identifying vulnerabilities from alerts and pull requests, enabling security updates, and taking remediation actions including testing and merging pull requests.

Topic 3	• Describe GitHub Advanced Security best practices, results, and how to take corrective measures: This section evaluates skills of Security Managers and Development Team Leads in effectively handling GHAS results and applying best practices. It includes using Common Vulnerabilities and Exposures (CVE) and Common Weakness Enumeration (CWE) identifiers to describe alerts and suggest remediation, decision-making processes for closing or dismissing alerts including documentation and data-based decisions, understanding default CodeQL query suites, how CodeQL analyzes compiled versus interpreted languages, the roles and responsibilities of development and security teams in workflows, adjusting severity thresholds for code scanning pull request status checks, prioritizing secret scanning remediation with filters, enforcing CodeQL and Dependency Review workflows via repository rulesets, and configuring code scanning, secret scanning, and dependency analysis to detect and remediate vulnerabilities earlier in the development lifecycle, such as during pull requests or by enabling push protection.
Topic 4	• Configure and use Code Scanning with CodeQL: This domain measures skills of Application Security Analysts and DevSecOps Engineers in code scanning using both CodeQL and third-party tools. It covers enabling code scanning, the role of code scanning in the development lifecycle, differences between enabling CodeQL versus third-party analysis, implementing CodeQL in GitHub Actions workflows versus other CI tools, uploading SARIF results, configuring workflow frequency and triggering events, editing workflow templates for active repositories, viewing CodeQL scan results, troubleshooting workflow failures and customizing configurations, analyzing data flows through code, interpreting code scanning alerts with linked documentation, deciding when to dismiss alerts, understanding CodeQL limitations related to compilation and language support, and defining SARIF categories.
Topic 5	• Describe the GHAS security features and functionality: This section of the exam measures skills of Security Engineers and Software Developers and covers understanding the role of GitHub Advanced Security (GHAS) features within the overall security ecosystem. Candidates learn to differentiate security features available automatically for open source projects versus those unlocked when GHAS is paired with GitHub Enterprise Cloud (GHEC) or GitHub Enterprise Server (GHES). The domain includes knowledge of Security Overview dashboards, the distinctions between secret scanning and code scanning, and how secret scanning, code scanning, and Dependabot work together to secure the software development lifecycle. It also covers scenarios contrasting isolated security reviews with integrated security throughout the development lifecycle, how vulnerable dependencies are detected using manifests and vulnerability databases, appropriate responses to alerts, the risks of ignoring alerts, developer responsibilities for alerts, access management for viewing alerts, and the placement of Dependabot alerts in the development process.

GH-500 Detailed Study Dumps | GH-500 Pass Test Guide

As you may find on our website, we will never merely display information in our GH-500 praparation guide. Our team of experts has extensive experience. They will design scientifically and arrange for GH-500 actual exam that are most suitable for users. In the study plan, we will also create a customized plan for you based on your specific situation. And our professional experts have developed three versions of our GH-500 Exam Questions for you: the PDF, Software and APP online.

Microsoft GitHub Advanced Security Sample Questions (Q53-Q58):

NEW QUESTION # 53
Which CodeQL query suite provides queries of lower severity than the default query suite?

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

Answer: B

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.

NEW QUESTION # 54
How do I configure a webhook to monitor key scan alert events? What are the steps of this operation?

- A. Document alternatives to storing secrets in the source code.
- B. Dismiss alerts that are older than 90 days.
- C. Enable system for cross-domain identity management (SCIM) provisioning for the enterprise.
- D. Configure a webhook to monitor for secret scanning alert events.

Answer: A,D

Explanation:
To proactively address secret scanning:
Webhooks can be configured to listen for secret scanning events. This allows automation, logging, or alerting in real-time when secrets are detected.
Documenting secure development practices (like using environment variables or secret managers) helps reduce the likelihood of developers committing secrets in the first place.
Dismissal based on age is not a best practice without triage. SCIM deals with user provisioning, not scanning alerts.

NEW QUESTION # 55
Which CodeQL query suite provides queries of lower severity than the default query suite?

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

Answer: B

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.

NEW QUESTION # 56
When using CodeQL, how does extraction for compiled languages work?

- A. By monitoring the normal build process
- B. By generating one language at a time
- C. By resolving dependencies to give an accurate representation of the codebase
- D. By running directly on the source code

Answer: A

Explanation:
For compiled languages, CodeQL performs extraction by monitoring the normal build process. This means it watches your usual build commands (like make, javac, or dotnet build) and extracts the relevant data from the actual build steps being executed.
CodeQL uses this information to construct a semantic database of the application.
This approach ensures that CodeQL captures a precise, real-world representation of the code and its behavior as it is compiled, including platform-specific configurations or conditional logic used during build.

NEW QUESTION # 57
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?

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

Answer: C

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.

NEW QUESTION # 58
.....