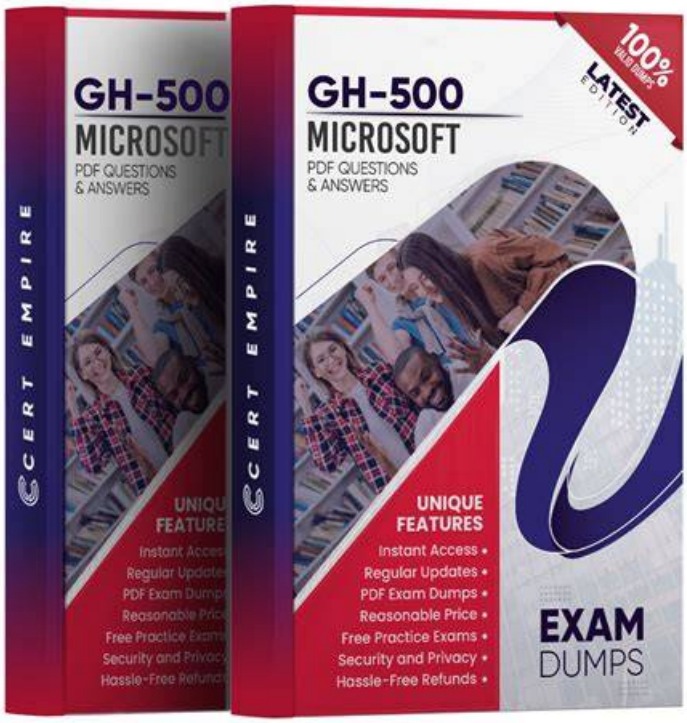
Valid free GH-500 exam dumps collection - Microsoft GH-500 exam tests



P.S. Free 2026 Microsoft GH-500 dumps are available on Google Drive shared by PassTorrent: https://drive.google.com/open?id=1XsgB9_cYfQMNfQ_HquGoaVrvMCT1XS6w

By using our GH-500 exam braindumps, it will be your habitual act to learn something with efficiency. With the cumulative effort over the past years, our GH-500 study guide has made great progress with passing rate up to 98 to 100 percent among the market. A lot of professional experts concentrate to making our GH-500 Preparation materials by compiling the content so they have gained reputation in the market for their proficiency and dedication.

Microsoft GH-500 Exam Syllabus Topics:

Topic	Details
Topic 1	• 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 2	• 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 3	• 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 4	• 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 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 Exam Questions And Answers & New GH-500 Exam Book

PassTorrent presents its GitHub Advanced Security (GH-500) exam product at an affordable price as we know that applicants desire to save money. To gain all these benefits you need to enroll in the GitHub Advanced Security EXAM and put all your efforts to pass the challenging GitHub Advanced Security (GH-500) exam easily. In addition, you can test specs of the GitHub Advanced Security practice material before buying by trying a free demo. These incredible features make PassTorrent prep material the best option to succeed in the Microsoft GH-500 examination. Therefore, don't wait. Order Now !!!

Microsoft GitHub Advanced Security Sample Questions (Q93-Q98):

NEW QUESTION # 93
In a private repository, what minimum requirements does GitHub need to generate a dependency graph? (Each answer presents part of the solution. Choose two.)

- A. Dependency graph enabled at the organization level for all new private repositories
- B. Write access to the dependency manifest and lock files for an enterprise
- C. Read-only access to the dependency manifest and lock files for a repository
- D. Read-only access to all the repository's files

Answer: A,C

Explanation:
Comprehensive and Detailed Explanation:
To generate a dependency graph for a private repository, GitHub requires:
Dependency graph enabled: The repository must have the dependency graph feature enabled. This can be configured at the organization level to apply to all new private repositories.
Access to manifest and lock files: GitHub needs read-only access to the repository's dependency manifest and lock files (e.g., package.json, requirements.txt) to identify and map dependencies.

NEW QUESTION # 94
What is a benefit of using a custom CodeQL configuration file?

- A. It specifies a token that has access to the private repository.
- B. It allows configuration options for multiple repositories in a single place.
- C. It disables packs from running the default query suite.
- D. It allows you to schedule the scan.

Answer: B

Explanation:
Using a custom configuration file
A custom configuration file is an alternative way to specify additional packs and queries to run.
You can also use the file to disable the default queries, exclude or include specific queries, and to specify which directories to scan during analysis.
The configuration file can be located within the repository you are analyzing, or in an external repository. Using an external repository allows you to specify configuration options for multiple repositories in a single place.

NEW QUESTION # 95
Which of the following is the best way to prevent developers from adding secrets to the repository?

- A. Create a CODEOWNERS file
- B. Configure a security manager
- C. Make the repository public
- 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.

NEW QUESTION # 96
You have a GitHub organization on GitHub Team that has GitHub Secret Protection enabled. The organization includes a team named Group1 and a repository named Repo1. Repo1 has secret scanning push protection enabled and an organization-level custom security configuration applied.
Developers who are NOT members of Group1 can request a delegated bypass when push protection blocks a commit.
You need to ensure that only the Group1 members can approve or deny push protection bypass requests for Repo1.
What should you do in Repo1?

- A. Set Bypass privileges to Specific actors, and then choose only Group1.
- B. Configure delegated bypass and specify Group1 as the reviewers.
- C. Create a custom security configuration and enable Prevent direct alert dismissals for code scanning.
- D. In the organization-level custom security configuration, set Bypass privileges to Repository administrators.

Answer: D

Explanation:

To restrict push protection bypass approvals to a specific team, you must configure the delegated bypass list within the organization-level custom security configuration. Because an organization- level custom security configuration is applied to the repository, the local repository settings for delegated bypass are automatically locked and overridden.
Reference:
https://docs.github.com/en/enterprise-server@3.20/code-security/how-tos/secure-your-secrets/manage-bypass-requests/enabling-delegated-bypass-for-push-protection

NEW QUESTION # 97
In the pull request, how can developers avoid adding new dependencies with known vulnerabilities?

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

Answer: B

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.

NEW QUESTION # 98
.....

The most notable feature of our GH-500 learning quiz is that they provide you with the most practical solutions to help you learn the exam points of effortlessly and easily, then mastering the core information of the certification course outline. Their quality of our GH-500 Study Guide is much higher than the quality of any other materials, and questions and answers of GH-500 training materials contain information from the best available sources.