GH-500赤本勉強 & GH-500出題内容

ちなみに、JPTestKing GH-500の一部をクラウドストレージからダウンロードできます：https://drive.google.com/open?id=1UtW2nr6_eOFig642fRD7MdkqwML8qTf2

IT認証資料を提供したほかのサイトより、JPTestKingのプロかつ高品質の製品は最高のものです。JPTestKingを選んだら成功を選んだということです。JPTestKingのMicrosoftのGH-500試験トレーニング資料はあなたが成功への保証です。JPTestKingを利用したら、あなたはきっと高い点数を取ることができ、あなたの理想なところへと進むことができます。

我々社のMicrosoft GH-500認定試験問題集の合格率は高いのでほとんどの受験生はGH-500認定試験に合格するのを保証します。もしあなたはMicrosoft GH-500試験問題集に十分な注意を払って、GH-500試験の解答を覚えていれば、GH-500認定試験の成功は明らかになりました。Microsoft GH-500模擬問題集で実際の質問と正確の解答に疑問があれば、無料の練習問題集サンプルをダウンロードし、チェックしてください。

>> GH-500赤本勉強 <<

GH-500出題内容、GH-500学習関連題

誰でも給料が高いのを希望します。でも、給料が高いかどうかはあなたの価値次第です。GH-500認証試験に合格したら、自分の価値を高めることができます。我々JPTestKingの問題集は全面的で質量の高いですから、受験生としてのあなたに一番ふさわしいです。我々の資料を利用したら、あなたはGH-500試験に合格することができます。

Microsoft GH-500 認定試験の出題範囲:

トピック	出題範囲
トピック 1	• 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.

トピック 2	• 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.
トピック 3	• 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.
トピック 4	• 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.
トピック 5	• 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.

Microsoft GitHub Advanced Security 認定 GH-500 試験問題 (Q53-Q58):

質問 # 53
What is the purpose of the SECURITY.md file in a GitHub repository?

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

正解：B

解説：
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.

質問 # 54
As a repository owner, you want to receive specific notifications, including security alerts, for an individual repository. Which repository notification setting should you use?

- A. Custom
- B. Participating and @mentions
- C. All Activity
- D. Ignore

正解：A

解説：
Using the Custom setting allows you to subscribe to specific event types, such as Dependabot alerts or vulnerability notifications, without being overwhelmed by all repository activity. This is essential for repository maintainers who need fine-grained control over what kinds of events trigger notifications.
This setting is configurable per repository and allows users to stay aware of critical issues while minimizing notification noise.

質問 # 55
When using CodeQL, how does extraction for compiled languages work?

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

正解：B

解説：
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.

質問 # 56
After investigating a code scanning alert related to injection, you determine that the input is properly sanitized using custom logic. What should be your next step?

- A. Dismiss the alert with the reason "false positive."
- B. Draft a pull request to update the open-source query.
- C. Open an issue in the CodeQL repository.
- D. Ignore the alert.

正解：A

解説：
When you identify that a code scanning alert is a false positive-such as when your code uses a custom sanitization method not recognized by the analysis-you should dismiss the alert with the reason "false positive." This action helps improve the accuracy of future analyses and maintains the relevance of your security alerts.
As per GitHub's documentation:
"If you dismiss a CodeQL alert as a false positive result, for example because the code uses a sanitization library that isn't supported, consider contributing to the CodeQL repository and improving the analysis." By dismissing the alert appropriately, you ensure that your codebase's security alerts remain actionable and relevant.

質問 # 57
If default code security settings have not been changed at the repository, organization, or enterprise level, which repositories receive Dependabot alerts?

- A. None
- B. Repositories owned by an enterprise account
- C. Repositories owned by an organization
- D. Private repositories

正解：A

解説：
By default, no repositories receive Dependabot alerts unless configuration is explicitly enabled. GitHub does not enable Dependabot alerts automatically for any repositories unless:
The feature is turned on manually
It's configured at the organization or enterprise level via security policies This includes public, private, and enterprise-owned repositories - manual activation is required.

質問 # 58
.....

BONUS！！！ JPTestKing GH-500ダンプの一部を無料でダウンロード：https://drive.google.com/open?id=1UtW2nr6_eOFig642fRD7MdkqwML8qTf2