GH-500완벽한덤프자료 - GH-500높은통과율공부문제

참고: DumpTOP에서 Google Drive로 공유하는 무료 2026 Microsoft GH-500 시험 문제집이 있습니다: https://drive.google.com/open?id=13RBMmlo29vxhoPjHgIhI_sSZZjIRH-rB

IT업계에서 자신만의 위치를 찾으려면 자격증을 많이 취득하는것이 큰 도움이 될것입니다. Microsoft 인증 GH-500 시험은 아주 유용한 시험입니다. Microsoft 인증GH-500시험출제경향을 퍼펙트하게 연구하여DumpTOP에서는 Microsoft 인증GH-500시험대비덤프를 출시하였습니다. DumpTOP에서 제공해드리는Microsoft 인증GH-500시험덤프는 시장에서 판매하고 있는Microsoft 인증GH-500덤프중 가장 최신버전덤프로서 덤프에 있는 문제만 공부하시면 시험통과가 쉬워집니다.

It 업계 중 많은 분들이 인증시험에 관심이 많은 인사들이 많습니다.it산업 중 더 큰 발전을 위하여 많은 분들이 Microsoft GH-500를 선택하였습니다.인증시험은 패스를 하여야 자격증취득이 가능합니다.그리고 무엇보다도 통행증을 받을 수 잇습니다.Microsoft GH-500은 그만큼 아주 어려운 시험입니다. 그래도Microsoft GH-500인증을 신청하여야 좋은 선택입니다.우리는 매일매일 자신을 업그레이드 하여야만 이 경쟁이 치열한 사회에서 살아남을 수 있기 때문입니다.

>> GH-500완벽한 덤프자료 <<

GH-500높은 통과율 공부문제, GH-500시험유형

GH-500인증시험은 IT업계에 종사하고 계신 분이시라면 최근 많은 인기를 누리고 있다는 것을 알고 계실것입니다. GH-500인증시험을 패스하여 자격증을 취득하는데 가장 쉬운 방법은 DumpTOP에서 제공해드리는 GH-500덤프를 공부하는 것입니다. Microsoft GH-500덤프에 있는 문제와 답만 기억하시면 GH-500시험을 패스하는데 많은 도움이 됩니다.덤프구매후 최신버전으로 업데이트되면 업데이트버전을 시스템 자동으로 구매시 사용한 메일주소로 발송해드려 덤프유효기간을 최대한 길게 연장해드립니다.

Microsoft GH-500 시험요강:

주제	소개
주제 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.

주제 2	• 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.
주제 3	• 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.
주제 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.
주제 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.

최신 GitHub Administrator GH-500 무료샘플문제 (Q74-Q79):

질문 # 74
If default code security settings have not been changed at the repository, organization, or enterprise level, which repositories receive Dependabot alerts?

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

정답：C

설명：
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.

질문 # 75
When using CodeQL, what extension stores query suite definitions?

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

정답： A

설명:
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.

질문 # 76
In the pull request, how can developers avoid adding new dependencies with known vulnerabilities?

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

정답： C

설명:
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.

질문 # 77
A secret scanning alert should be closed as "used in tests" when a secret is:

- A. In a test file.
- B. Not a secret in the production environment.
- C. Solely used for tests.
- D. In the readme.md file.

정답： C

설명:
If a secret is intentionally used in a test environment and poses no real-world security risk, you may close the alert with the reason "used in tests". This helps reduce noise and clarify that the alert was reviewed and accepted as non-critical.
Just being in a test file isn't enough unless its purpose is purely for testing.

질문 # 78
You are a maintainer of a repository and Dependabot notifies you of a vulnerability. Where could the vulnerability have been disclosed? (Each answer presents part of the solution. Choose two.)

- A. In the National Vulnerability Database
- B. In manifest and lock files
- C. In the dependency graph
- D. In security advisories reported on GitHub

정답：A,D

설명：
Comprehensive and Detailed Explanation:
Dependabot alerts are generated based on data from various sources:
National Vulnerability Database (NVD): A comprehensive repository of known vulnerabilities, which GitHub integrates into its advisory database.
GitHub Docs
Security Advisories Reported on GitHub: GitHub allows maintainers and security researchers to report and discuss vulnerabilities, which are then included in the advisory database.
The dependency graph and manifest/lock files are tools used by GitHub to determine which dependencies are present in a repository but are not sources of vulnerability disclosures themselves.

질문 # 79
.....