Microsoft.GH-500.v2026-10-23.q57

Exam Code:	GH-500
Exam Name:	GitHub Advanced Security
Certification Provider:	Microsoft
Free Question Number:	57
Version:	v2026-10-23
# of views:	103
# of Questions views:	570
https://www.freecram.net/torrent/Microsoft.GH-500.v2026-10-23.q57.html	

NEW QUESTION: 1
Which alerts do you see in the repository's Security tab? (Each answer presents part of the solution. Choose three.)
A. Repository permissions
B. Secret scanning alerts
C. Dependabot alerts
D. Security status alerts
E. Code scanning alerts
Answer: (SHOW ANSWER)
In a repository's Security tab, you can view:
* Secret scanning alerts : Exposed credentials or tokens
* Dependabot alerts : Vulnerable dependencies from the advisory database
* Code scanning alerts : Vulnerabilities in code detected via static analysis (e.g., CodeQL) You won't see general "security status alerts" (not a formal category) or permission-related alerts here.
: GitHub Docs - Understanding the Security Tab

NEW QUESTION: 2
Which of the following options are code scanning application programming interface (API) endpoints? (Each answer presents part of the solution. Choose two.)
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: (SHOW ANSWER)
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.
: GitHub Docs - Code Scanning REST API

NEW QUESTION: 3
You are configuring a CodeQL workflow for compiled languages. What happens if your workflow uses a language matrix?
A. Analysis of other languages in your repository will fail unless you supply explicit build commands.
B. Autobuild attempts to build the supported language that has the most source files in the repository.
C. You may need to install additional software to use the autobuild process.
D. Autobuild attempts to build each of the languages listed in the matrix.
Answer: (SHOW ANSWER)
When a CodeQL workflow uses a language matrix, the autobuild step attempts to build each compiled language listed in that matrix. This differs from a workflow without a language matrix, where autobuild attempts to build the supported compiled language with the greatest number of source files in the repository.
A matrix is therefore useful when a repository contains multiple languages that require separate CodeQL analysis jobs. GitHub's current CodeQL build documentation explicitly describes this matrix behavior. The analysis may use newer build modes such as none for some compiled languages, depending on language and configuration, but when the workflow specifically uses autobuild with a language matrix, each compiled language represented by that matrix is processed independently. Therefore, option D correctly describes the behavior.

NEW QUESTION: 4
Which of the following benefits do code scanning, secret scanning, and dependency review provide?
A. Search for potential security vulnerabilities, detect secrets, and show the full impact of changes to dependencies
B. Confidentially report security vulnerabilities and privately discuss and fix security vulnerabilities in your repository's code
C. View alerts about dependencies that are known to contain security vulnerabilities
D. Automatically raise pull requests, which reduces your exposure to older versions of dependencies
Answer: (SHOW ANSWER)
These three features provide a complete layer of defense:
* Code scanning identifies security flaws in your source code
* Secret scanning detects exposed credentials

* Dependency review shows the impact of package changes during a pull request Together, they give developers actionable insight into risk and coverage throughout the SDLC.
: GitHub Docs - About GitHub Advanced Security Features

NEW QUESTION: 5
You want to enforce an enterprise policy that allows repository administrators within all organizations to enable GitHub Advanced Security for their repositories. Which option should you choose for this policy?
A. No policy
B. Allow for all organizations
C. Never allow
D. Allow for selected organizations
Answer: (SHOW ANSWER)
The Allow for all organizations policy permits repository administrators throughout organizations owned by the enterprise to enable applicable Advanced Security capabilities on their repositories. GitHub enterprise policy can determine whether repository administrators are allowed to enable GitHub Secret Protection, GitHub Code Security, or legacy GitHub Advanced Security features. Selecting an option limited to chosen organizations would restrict availability rather than permit it enterprise-wide. A prohibition such as Never allow would prevent repository administrators from enabling the feature, while No policy does not explicitly establish the enterprise-wide permission requested. GitHub's current documentation describes this policy as controlling availability across all organizations or specified organizations, with organization owners and security managers retaining broader security-management capabilities. Therefore, B matches the stated requirement to allow repository administrators throughout all enterprise organizations.

NEW QUESTION: 6
As a repository owner, you do not want to run a GitHub Actions workflow when changes are made to any .txt or markdown files. How would you adjust the event trigger for a pull request that targets the main branch?
(Each answer presents part of the solution. Choose three.)
* on:
* pull_request:
* branches: [main]
A. - '/*.md'
B. - '/*.txt'
C. paths:
D. paths-ignore:
E. - 'docs/*.md'
Answer: A,B,D (LEAVE A REPLY)
To exclude .txt and .md files from triggering workflows on pull requests to the main branch:
* on: defines the event (e.g., pull_request)

* pull_request: is the trigger
* paths-ignore: is the key used to ignore file patterns
Example YAML:
yaml
CopyEdit
on:
pull_request:
branches:
- main
paths-ignore:
- '*.md'
- '*.txt'
Using paths: would include only specific files instead - not exclude. paths-ignore: is correct here.
: GitHub Docs - Workflow Syntax for GitHub Actions

NEW QUESTION: 7
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. Draft a pull request to update the open-source query.
B. Ignore the alert.
C. Open an issue in the CodeQL repository.
D. Dismiss the alert with the reason "false positive."
Answer: (SHOW ANSWER)
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.

NEW QUESTION: 8
What YAML syntax do you use to exclude certain files from secret scanning?
A. decrypt_secret.sh
B. paths-ignore:
C. branches-ignore:
D. secret scanning.yml
Answer: (SHOW ANSWER)

To exclude specific files or directories from being scanned by secret scanning in GitHub Actions, you can use the paths-ignore: key within your YAML workflow file.
This tells GitHub to ignore specified paths when scanning for secrets, which can be useful for excluding test data or non-sensitive mock content.
Other options listed are invalid:
* branches-ignore: excludes branches, not files.
* decrypt_secret.sh is not a YAML key.
* secret scanning.yml is not a recognized filename for configuration.
: GitHub Docs - Ignoring Files in GitHub Actions for Secret Scanning

NEW QUESTION: 9
Which of the following statements best describes secret scanning push protection?
A. Commits that contain secrets are blocked before code is added to the repository.
B. Secret scanning alerts must be closed before a branch can be merged into the repository.
C. Buttons for sensitive actions in the GitHub UI are disabled.
D. Users need to reply to a 2FA challenge before any push events.
Answer: (SHOW ANSWER)
Comprehensive and Detailed Explanation:
Secret scanning push protection is a proactive feature that scans for secrets in your code during the push process. If a secret is detected, the push is blocked, preventing the secret from being added to the repository.
This helps prevent accidental exposure of sensitive information.
GitHub Docs
References: GitHub Docs - About push protection

NEW QUESTION: 10
Assuming that default security and analysis settings have not been changed at the repository, organization, or enterprise level, which scenario would generate a dependency graph for the repository?
Freecram.net
A. When a public repository has a new dependency added to its manifest file
B. When a private repository is forked to be used as a dependent
C. When a public repository is forked to be used as a dependent
D. When a private repository has a new dependency added to its manifest file
Answer: (SHOW ANSWER)
GitHub enables the dependency graph by default for public repositories. The graph analyzes supported manifest and lock files and identifies dependencies declared in those files. When a dependency is added or changed in a supported manifest on the repository's default branch, GitHub updates the dependency graph automatically. Private repositories can also use the dependency graph, but the feature must be enabled unless an organization or enterprise configuration enables it automatically. Forking a repository does not itself constitute the dependency declaration that this question describes. Therefore, under default settings, adding a

new dependency to a manifest in a public repository is the scenario that causes GitHub's dependency graph to identify and represent that dependency.

NEW QUESTION: 11
When does Dependabot alert you of a vulnerability in your software development process?
A. When a pull request adding a vulnerable dependency is opened
B. As soon as a vulnerable dependency is detected
C. As soon as a pull request is opened by a contributor
D. When Dependabot opens a pull request to update a vulnerable dependency
Answer: (SHOW ANSWER)
Dependabot alerts are generated as soon as GitHub detects a known vulnerability in one of your dependencies. GitHub does this by analyzing your repository's dependency graph and matching it against vulnerabilities listed in the GitHub Advisory Database. Once a match is found, the system raises an alert automatically without waiting for a PR or manual action.
This allows organizations to proactively mitigate vulnerabilities as early as possible, based on real-time detection.
: GitHub Docs - About Dependabot alerts; Managing alerts in GitHub Dependabot

NEW QUESTION: 12
Where can you find a deleted line of code that contained a secret value?
A. Insights
B. Issues
C. Commits
D. Dependency graph
Answer: (SHOW ANSWER)
Secrets committed and then deleted are still accessible in the repository's Git history . To locate them, navigate to the Commits tab. GitHub's secret scanning can detect secrets in both current and historical commits, which is why remediation should also include revoking the secret , not just removing it from the latest code.
: GitHub Docs - Remediating Secret Scanning Alerts

NEW QUESTION: 13
Which of the following options would close a Dependabot alert?
A. Creating a pull request to resolve the vulnerability that will be approved and merged
B. Viewing the Dependabot alert on the Dependabot alerts tab of your repository
C. Viewing the dependency graph
D. Leaving the repository in its current state
Answer: (SHOW ANSWER)
A Dependabot alert is only marked as resolved when the related vulnerability is no longer present in your code - specifically after you merge a pull request that updates the vulnerable dependency.

Simply viewing alerts or graphs does not affect their status. Ignoring the alert by leaving the repo unchanged keeps the vulnerability active and unresolved.
: GitHub Docs - Managing Dependabot Security Updates

NEW QUESTION: 14
Which of the following features can be used to enforce passing status checks for code scanning and dependency review workflows?
A. Security GuardRails
B. Status enforcement
C. Repository rulesets
D. Insights
Answer: (SHOW ANSWER)
Repository rulesets provide centralized enforcement of repository policies before changes can be merged.
Rulesets can require workflows or status checks to succeed, and GitHub specifically supports security- oriented enforcement such as dependency review and code scanning merge protection. For dependency review, an organization can require the dependency review workflow through a ruleset so that vulnerable dependency changes prevent merging. Code scanning rulesets can likewise require acceptable code scanning results and define alert thresholds. "Security GuardRails" and "status enforcement" are not GitHub repository features with the functionality described here, while Insights is primarily used for repository statistics and reporting. Therefore, repository rulesets provide the policy mechanism needed to enforce these security gates before code reaches protected branches.

NEW QUESTION: 15
Where in the repository can you give additional users access to secret scanning alerts?
A. Security
B. Settings
C. Secrets
D. Insights
Answer: (SHOW ANSWER)
To grant specific users access to view and manage secret scanning alerts , you do this via the Settings tab of the repository. From there, under the "Code security and analysis" section, you can add individuals or teams with roles such as security manager .
The Security tab only displays alerts; access control is handled in Settings.
: GitHub Docs - Granting Access to Secret Scanning Alerts

NEW QUESTION: 16
If notification and alert recipients are not customized, which users receive notifications about new Dependabot alerts in an affected repository?
A. Users with Write permissions to the repository

B. Users with Admin privileges to the repository
C. Users with Maintain privileges to the repository
D. Users with Read permissions to the repository
Answer: (SHOW ANSWER)
By default, users with Write, Maintain, or Admin permissions will receive notifications for new Dependabot alerts . However, Write permission is the minimum level needed to be automatically notified.
Users with only Read access do not receive alerts unless added explicitly.: GitHub Docs - Dependabot Alerts Notification Scope

Valid GH-500 Dumps shared by ExamDiscuss.com for Helping Passing GH-500 Exam! ExamDiscuss.com now offer the **newest GH-500 exam dumps**, the ExamDiscuss.com GH-500 exam **questions have been updated** and **answers have been corrected** get the **newest** ExamDiscuss.com GH-500 dumps with Test Engine here: https://www.examdiscuss.com/Microsoft/exam/GH-500/premium/ (**127** Q&As Dumps, **35%OFF** **Special Discount Code: freecram**)

NEW QUESTION: 17
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
Answer: (SHOW ANSWER)
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 .
: GitHub Docs - About Dependabot Alerts

NEW QUESTION: 18
A colleague ignores a code scanning alert. What are the implications of the colleague's action? (Each answer presents part of the solution. Choose three.)
A. A dangerous argument could be passed to functions.
B. Data could be used insecurely.
C. GitHub removes the alert after sixty days.
D. Webhooks and the code scanning API remove the alert.

E. Sensitive information could be leaked.

Answer: A,B,E (LEAVE A REPLY)

Ignoring an unresolved code scanning alert leaves the underlying security issue present in the application.

GitHub identifies several risks through CodeQL data-flow analysis, including data being used insecurely, dangerous arguments reaching sensitive functions, and sensitive information being leaked. These are precisely the kinds of vulnerabilities for which code scanning traces data from a source to a potentially unsafe sink.

GitHub does not automatically remove an unresolved alert merely because sixty days have elapsed. Likewise, webhooks and the code scanning API can expose alert events and enable integrations, but they do not automatically eliminate the vulnerability. Proper remediation requires fixing the affected code or intentionally dismissing an alert with an appropriate reason after security review. Therefore, A, B, and E describe the relevant security implications.

NEW QUESTION: 19

What happens when you disable secret scanning for a GitHub organization?

A. Changes affect public repositories for your organization where GitHub Advanced Security is enabled.

B. All GitHub Advanced Security features are disabled.

C. Changes affect private repositories where GitHub Advanced Security is also enabled.

D. Changes affect all repositories with GitHub Advanced Security disabled.

Answer: (SHOW ANSWER)

The organization-level secret-scanning control represented by this question applies to the organization's private repositories that use the applicable Advanced Security or Secret Protection capability. Public repositories are treated differently because GitHub runs secret scanning for public repositories automatically and provides that protection independently of a paid Advanced Security license. Disabling secret scanning also does not disable every other Advanced Security capability; code scanning, dependency features, and related security controls are managed separately or through security configurations. Current GitHub administration increasingly manages these features through Secret Protection and organization-level security configurations, but the principle remains that disabling the licensed secret-scanning capability affects repositories to which that configuration applies rather than universally switching off every security feature. Therefore, C is the intended answer.

NEW QUESTION: 20

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

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: (SHOW ANSWER)

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.
: GitHub Docs - About Dependabot alerts; Reviewing and managing alerts

NEW QUESTION: 21
Where can you view code scanning results from CodeQL analysis?
A. The repository's code scanning alerts
B. A CodeQL database
C. A CodeQL query pack
D. At Security advisories
Answer: (SHOW ANSWER)
All results from CodeQL analysis appear under the repository's code scanning alerts tab. This section is part of the Security tab and provides a list of all current, fixed, and dismissed alerts found by CodeQL.
A CodeQL database is used internally during scanning but does not display results. Query packs contain rules, not results. Security advisories are for published vulnerabilities, not per-repo findings.
: GitHub Docs - Viewing code scanning alerts

NEW QUESTION: 22
Which of the following workflow events would trigger a dependency review? (Each answer presents a complete solution. Choose two.)
A. pull_request
B. workflow_dispatch
C. trigger
D. commit
Answer: (SHOW ANSWER)
Comprehensive and Detailed Explanation:
Dependency review is triggered by specific events in GitHub workflows:
pull_request: When a pull request is opened, synchronized, or reopened, GitHub can analyze the changes in dependencies and provide a dependency review.
workflow_dispatch: This manual trigger allows users to initiate workflows, including those that perform dependency reviews.
The trigger and commit options are not recognized GitHub Actions events and would not initiate a dependency review.
References: GitHub Docs - Events that trigger workflows

NEW QUESTION: 23
What is a security policy?
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: (SHOW ANSWER)
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.
: GitHub Docs - Adding a security policy to your repository

NEW QUESTION: 24
Assuming there is no custom Dependabot behavior configured, where possible, what does Dependabot do after sending an alert about a vulnerable dependency in a repository?
A. Creates a pull request to upgrade the vulnerable dependency to the minimum possible secure version
B. Scans repositories for vulnerable dependencies on a schedule and adds those files to a manifest
C. Constructs a graph of all the repository's dependencies and public dependents for the default branch
D. Scans any push to all branches and generates an alert for each vulnerable repository
Answer: (SHOW ANSWER)
After generating an alert for a vulnerable dependency, Dependabot automatically attempts to create a pull request to upgrade that dependency to the minimum required secure version -if a fix is available and compatible with your project.
This automated PR helps teams fix vulnerabilities quickly with minimal manual intervention. You can also configure update behaviors using dependabot.yml, but in the default state, PR creation is automatic.
: GitHub Docs - About Dependabot alerts; About Dependabot security updates

NEW QUESTION: 25
What step is required to run a SARIF-compatible (Static Analysis Results Interchange Format) tool on GitHub Actions?
A. Update the workflow to include a final step that uploads the results.
B. By default, the CodeQL runner automatically uploads results to GitHub on completion.
C. The CodeQL action uploads the SARIF file automatically when it completes analysis.
D. Use the CLI to upload results to GitHub.

Answer: (SHOW ANSWER)
When using a SARIF-compatible tool within GitHub Actions, it's necessary to explicitly add a step in your workflow to upload the analysis results. This is typically done using the upload-sarif action, which takes the SARIF file generated by your tool and uploads it to GitHub for processing and display in the Security tab.
Without this step, the results won't be available in GitHub's code scanning interface.
: GitHub Docs - Uploading a SARIF file to GitHub

NEW QUESTION: 26
When secret scanning detects a set of credentials on a public repository, what does GitHub do?
A. It notifies the service provider who issued the secret.
B. It displays a public alert in the Security tab of the repository.
C. It scans the contents of the commits for additional secrets.
D. It sends a notification to repository members.
Answer: (SHOW ANSWER)
When a public repository contains credentials that match known secret formats, GitHub will automatically notify the service provider that issued the secret. This process is known as "secret scanning partner notification" . The provider may then revoke the secret or contact the user directly.
GitHub does not publicly display the alert and does not send internal repository notifications for public detections.
: GitHub Docs - Secret Scanning for Public Repositories

NEW QUESTION: 27
Dependabot has created a pull request. Which of the following commands should you use in a comment to prevent Dependabot from re-creating the same pull request?
A. @dependabot recreate
B. @dependabot ignore this dependency
C. @dependabot close
D. @dependabot rebase
Answer: (SHOW ANSWER)
The command @dependabot ignore this dependency closes the current Dependabot pull request and creates an ignore condition that prevents Dependabot from opening additional pull requests for that dependency. The restriction remains until the pull request is reopened or the dependency is manually upgraded to the suggested version. @dependabot recreate performs the opposite action: it recreates the pull request and overwrites edits made to it. @dependabot rebase rebases the Dependabot pull request against the current target branch. GitHub does not document @dependabot close as the command for permanently suppressing the dependency update. GitHub also provides more granular commands to ignore only a major, minor, or patch version when permanent dependency-level suppression would be too broad.

NEW QUESTION: 28
Why should you dismiss a code scanning alert?
A. If you fix the code that triggered the alert
B. To prevent developers from introducing new problems
C. If it includes an error in code that is used only for testing
D. If there is a production error in your code
Answer: (SHOW ANSWER)
You should dismiss a code scanning alert if the flagged code is not a true security concern , such as:
* Code in test files
* Code paths that are unreachable or safe by design
* False positives from the scanner
Fixing the code would automatically resolve the alert - not dismiss it. Dismissing is for valid exceptions or noise reduction.
: GitHub Docs - Dismissing Code Scanning Alerts

NEW QUESTION: 29
In the pull request, how can developers avoid adding new dependencies with known vulnerabilities?
A. Enable Dependabot alerts.
B. Add Dependabot rules.
C. Add a workflow with the dependency review action.
D. Enable Dependabot security updates.
Answer: (SHOW ANSWER)
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 .
: GitHub Docs - Dependency Review Action

NEW QUESTION: 30
Which of the following conditions must be met to enable secret scanning for private repositories?
A. An Advanced Security license must be uploaded to your repository.
B. An Advanced Security license must be applied to your user account.
C. The repository must be owned by an organization.
D. The repository must be owned by a user account.
Answer: (SHOW ANSWER)
For the organization-owned private repository scenario tested by this question, C is the intended answer.

GitHub Secret Protection supports secret scanning for private and internal repositories owned by organizations using an eligible GitHub Team or GitHub Enterprise plan with Secret Protection enabled.
Licensing is managed at the organization or enterprise level; an individual license is not uploaded to a repository or attached directly to a user's account. Current GitHub documentation has expanded availability in some enterprise scenarios, including certain user-owned repositories under Enterprise Managed Users and GitHub Enterprise Server. However, for the standard GitHub Advanced Security organization model represented by the question, an organization-owned repository is the applicable prerequisite among the supplied choices.

NEW QUESTION: 31
Where can you remove access to GitHub Advanced Security features for an individual repository in an organization? (Each answer presents part of the solution. Choose two.)
A. The enterprise's Settings tab
B. The organization's Settings tab
C. The repository's Settings tab
D. The organization's Repository permissions
Answer: (SHOW ANSWER)
GitHub Advanced Security features can be managed at both the organization and repository levels.
Organization owners can control security configurations and Advanced Security access across repositories from the organization's Settings area. At the repository level, administrators can manage eligible security and analysis features through the repository's Settings, subject to any policies or security configurations enforced by the organization or enterprise. Current GitHub terminology separates Advanced Security into products such as GitHub Secret Protection and GitHub Code Security, but the same management hierarchy applies.
The organization's Repository permissions area manages general repository access permissions rather than feature licensing or enablement, while enterprise settings primarily establish higher-level policies rather than directly serving as the normal per-repository control described here.

Valid GH-500 Dumps shared by ExamDiscuss.com for Helping Passing GH-500 Exam! ExamDiscuss.com now offer the **newest GH-500 exam dumps**, the ExamDiscuss.com GH-500 exam **questions have been updated** and **answers have been corrected** get the **newest** ExamDiscuss.com GH-500 dumps with Test Engine here: https://www.examdiscuss.com/Microsoft/exam/GH-500/premium/ (**127** Q&As Dumps, **35%OFF** **Special Discount Code: freecram**)

NEW QUESTION: 32
Which CodeQL query suite provides queries of lower severity than the default query suite?

A. github/codeql-go/ql/src@main
B. github/codeql/cpp/ql/src@main
C. security-extended
Answer: (SHOW ANSWER)
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.
: GitHub Docs - CodeQL Query Suite Types

NEW QUESTION: 33
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. Ignore
B. Participating and @mentions
C. All Activity
D. Custom
Answer: (SHOW ANSWER)
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.
: GitHub Docs - Configuring notifications; Managing security alerts

NEW QUESTION: 34
Assuming dependabot.yml does not exist in the repository, how many reviewers would Dependabot automatically assign to an open pull request?
A. 0
B. 1
C. 2
D. 3
Answer: (SHOW ANSWER)
By default, Dependabot creates pull requests without automatically assigning any reviewers, so the correct number is zero. Reviewers can be configured explicitly through Dependabot configuration or assigned through repository mechanisms such as CODEOWNERS. GitHub recommends using appropriate ownership and review processes when particular teams should review dependency updates. This default differs from Dependabot's ability to create pull requests automatically: creating the update does not imply that a reviewer is selected automatically. A

repository can customize Dependabot behavior through .github/dependabot.yml, including configuration related to update handling, but in the scenario specified there is no such configuration file. Consequently, no individual reviewer is automatically attached to the Dependabot pull request merely because Dependabot created it.

NEW QUESTION: 35
Which syntax in a query suite tells CodeQL to look for one or more specified .ql files?
A. qls
B. query
C. qlpack
Answer: (SHOW ANSWER)

NEW QUESTION: 36
How would you build your code within the CodeQL analysis workflow? (Each answer presents a complete solution. Choose two.)
A. Upload compiled binaries.
B. Use CodeQL's init action.
C. Ignore paths.
D. Implement custom build steps.
E. Use jobs.analyze.runs-on.
F. Use CodeQL's autobuild action.
Answer: (SHOW ANSWER)
Comprehensive and Detailed Explanation:
When setting up CodeQL analysis for compiled languages, there are two primary methods to build your code:
GitHub Docs
Autobuild: CodeQL attempts to automatically build your codebase using the most likely build method. This is suitable for standard build processes.
GitHub Docs
Custom Build Steps: For complex or non-standard build processes, you can implement custom build steps by specifying explicit build commands in your workflow. This provides greater control over the build process.
GitHub Docs
The init action initializes the CodeQL analysis but does not build the code. The jobs.analyze.runs-on specifies the operating system for the runner but is not directly related to building the code.
Uploading compiled binaries is not a method supported by CodeQL for analysis.
References: GitHub Docs - CodeQL code scanning for compiled languages

NEW QUESTION: 37
When configuring code scanning with CodeQL, what are your options for specifying additional queries?

(Each answer presents part of the solution. Choose two.)
A. Packs
B. github/codeql
C. Scope
D. Queries
Answer: (SHOW ANSWER)
You can customize CodeQL scanning by including additional query packs or by specifying individual queries :
* Packs : These are reusable collections of CodeQL queries bundled into a single package.
* Queries : You can point to specific files or directories containing .ql queries to include in the analysis.
github/codeql refers to a pack by name but is not a method or field. Scope is not a valid field used for configuration in this context.
: GitHub Docs - Customizing CodeQL Workflows

NEW QUESTION: 38
What is the first step in CodeQL analysis?
A. Converting results produced during query execution
B. Running CodeQL queries against the database
C. Preparing the code by creating a CodeQL database
D. Interpreting the query results
Answer: (SHOW ANSWER)
Before CodeQL can execute security queries, the source code must first be represented in a CodeQL database.
Database creation extracts structured, relational information from the source code and, where applicable, observes or performs the build process required for the language being analyzed.
Once the database exists, CodeQL can execute queries against it. The resulting query output is then interpreted using query metadata and can ultimately be converted to a format such as SARIF for presentation as code scanning alerts.
Therefore, running queries, interpreting results, and converting results are later phases of analysis. GitHub's CodeQL documentation explicitly identifies creation of the CodeQL database as a prerequisite for analyzing source code with CodeQL queries, making preparation/database creation the correct first step.

NEW QUESTION: 39
Which features are part of GitHub Advanced Security in the context of GitHub Enterprise? (Each correct answer presents part of the solution. Choose two.)
A. Dependency review
B. Dependency graph
C. Security policy
D. Secret scanning

Answer: (SHOW ANSWER)
Dependency review and secret scanning are capabilities associated with GitHub Advanced Security. In GitHub's current product structure, dependency review is included with GitHub Code Security, while secret scanning is a central capability of GitHub Secret Protection. Both products are components of the broader GitHub Advanced Security offering, and legacy GHAS licenses can include the combined capabilities. The dependency graph is an important supply-chain feature and provides the underlying dependency inventory, but it is broadly available beyond the licensed GHAS feature set and is enabled by default for public repositories. A repository security policy, normally provided through SECURITY.md, is also a general repository capability rather than one of the licensed Advanced Security features being tested. Therefore, dependency review and secret scanning are the correct selections.

NEW QUESTION: 40
To be compatible with code scanning, what data format must third-party code scanning tools use for output?
Freecram.net
A. Static Analysis Results Interchange Format (SARIF)
B. YAML
C. ESLint
D. ECMAScript
Answer: (SHOW ANSWER)
GitHub code scanning integrates with third-party static analysis tools through SARIF, the Static Analysis Results Interchange Format. A compatible analysis tool generates its findings as a SARIF file, which GitHub can process and display as code scanning alerts within the repository. GitHub supports SARIF output generated outside GitHub, including results from external CI/CD systems or security-analysis products. The results can then be uploaded to GitHub using an appropriate workflow action, API, or supported integration.
YAML is commonly used for GitHub Actions workflow configuration but is not the interchange format for code scanning findings. ESLint is an analysis tool rather than a results-exchange specification, and ECMAScript defines JavaScript language standards. Therefore, SARIF is the required output format.

NEW QUESTION: 41
By default, where will secret scanning look in a repository in order to execute its job? (Each correct answer presents part of the solution. Choose three.)
A. Dependencies
B. Selected files in the repository
C. All branches
D. Full commit history
E. All files in the repository
Answer: C,D,E (LEAVE A REPLY)

GitHub secret scanning examines the repository comprehensively rather than limiting analysis to selected files or dependency metadata. GitHub states that secret scanning scans the entire Git history across all branches for supported secrets. Consequently, both current files and historical versions of files can be inspected, which is why simply deleting a leaked credential in a later commit does not eliminate the security risk. The feature searches repository content for supported provider, generic, and other enabled secret patterns. Dependencies themselves are handled primarily through dependency graph and Dependabot functionality and are not a secret-scanning scope category. Therefore, the correct selections are all branches, the full commit history, and all files in the repository.

NEW QUESTION: 42
Which of the following secret scanning features can verify whether a secret is still active?
A. Push protection
B. Validity checks
C. Branch protection
D. Custom patterns
Answer: (SHOW ANSWER)
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.
: GitHub Docs - Secret Scanning Validity

NEW QUESTION: 43
What is required to trigger code scanning on a specified branch?
A. Developers must actively maintain the repository.
B. Secret scanning must be enabled on the repository.
C. The repository must be private.
D. The workflow file must exist in that branch.
Answer: (SHOW ANSWER)

NEW QUESTION: 44
Assuming security and analysis features are not configured at the repository, organization, or enterprise level, secret scanning is enabled on:
A. Public repositories
B. All new repositories within your organization
C. User-owned private repositories
D. Private repositories
Answer: (SHOW ANSWER)

By default, secret scanning is enabled automatically for all public repositories . For private or internal repositories, secret scanning must be enabled manually unless configured at the organization or enterprise level.
This default behavior helps protect open-source projects without requiring additional configuration.
: GitHub Docs - Secret Scanning Defaults and Scope

NEW QUESTION: 45
As a developer with write access, you navigate to a code scanning alert in your repository. When will GitHub close this alert?
A. After you triage the pull request containing the alert
B. When you use data-flow analysis to find potential security issues in code
C. After you find the code and click the alert within the pull request
D. After you fix the code by committing within the pull request
Answer: (SHOW ANSWER)
GitHub automatically closes a code scanning alert when the vulnerable code is fixed in the same branch where the alert was generated , usually via a commit inside a pull request . Simply clicking or triaging an alert does not resolve it. The alert is re-evaluated after each push to the branch, and if the issue no longer exists, it is marked as resolved.
: GitHub Docs - Code Scanning Alerts Lifecycle

NEW QUESTION: 46
A repository's dependency graph includes:
A. Dependencies parsed from a repository's manifest and lock files.
B. Annotated code scanning alerts from your repository's dependencies.
C. A summary of the dependencies used in your organization's repositories.
D. Dependencies from all your repositories.
Answer: (SHOW ANSWER)
The dependency graph in a repository is built by parsing manifest and lock files (like package.json, pom.
xml, requirements.txt). It helps GitHub detect dependencies and cross-reference them with known vulnerability databases for alerting.
It is specific to each repository and does not show org-wide or cross-repo summaries.
: GitHub Docs - Understanding the Dependency Graph

Valid GH-500 Dumps shared by ExamDiscuss.com for Helping Passing GH-500 Exam! ExamDiscuss.com now offer the **newest GH-500 exam dumps**, the ExamDiscuss.com GH-500 exam **questions have been updated** and **answers have been corrected** get the **newest** ExamDiscuss.com GH-500 dumps with Test Engine here:

https://www.examdiscuss.com/Microsoft/exam/GH-500/premium/ (**127** Q&As Dumps, **35%OFF**
Special Discount Code: freecram)

NEW QUESTION: 47
What permission level is required to manage code scanning alerts in a repository?
A. Read
B. Maintain
C. View
D. Triage
Answer: (SHOW ANSWER)
Current GitHub documentation requires Write access or higher to list, dismiss, and delete code scanning alerts. The PDF marks Triage, but that is not correct under the current repository-role permissions documented by GitHub. A user with Triage access can view code scanning alerts associated with pull requests, but Triage does not grant the broader permission required to manage the repository's code scanning alert list. Because Write is not one of the supplied choices, Maintain is the lowest listed role that exceeds Write and therefore has the required management capability. Read access does not permit alert management, and "View" is not one of GitHub's standard organization repository roles in this context. Accordingly, based on current official documentation, the technically correct option from the choices provided is B. Maintain.

NEW QUESTION: 48
When using CodeQL, how does extraction for compiled languages work?
A. By generating one language at a time
B. By resolving dependencies to give an accurate representation of the codebase
C. By monitoring the normal build process
D. By running directly on the source code
Answer: (SHOW ANSWER)
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.
: GitHub Docs - CodeQL for compiled languages

NEW QUESTION: 49
Assuming that notification and alert recipients are not customized, what does GitHub do when it identifies a vulnerable dependency in a repository where Dependabot alerts are enabled? (Each answer presents part of the solution. Choose two.)
A. It generates a Dependabot alert and displays it on the Security tab for the repository.

B. It notifies the repository administrators about the new alert.
C. It generates Dependabot alerts by default for all private repositories.
D. It consults with a security service and conducts a thorough vulnerability review.
Answer: (SHOW ANSWER)
Comprehensive and Detailed Explanation:
When GitHub identifies a vulnerable dependency in a repository with Dependabot alerts enabled, it performs the following actions:
Generates a Dependabot alert: The alert is displayed on the repository's Security tab, providing details about the vulnerability and affected dependency.
Notifies repository maintainers: By default, GitHub notifies users with write, maintain, or admin permissions about new Dependabot alerts.
GitHub Docs
These actions ensure that responsible parties are informed promptly to address the vulnerability.
References: GitHub Docs - About Dependabot alerts; Configuring notifications for Dependabot alerts

NEW QUESTION: 50
When using CodeQL, what extension stores query suite definitions?
A. .yml
B. .ql
C. .qll
D. .qls
Answer: D (LEAVE A REPLY)
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.
: GitHub Docs - CodeQL Query Suite Files

NEW QUESTION: 51
Which of the following tasks can be performed by a security team as a proactive measure to help address secret scanning alerts? (Each answer presents a complete solution. Choose two.)
A. Dismiss alerts that are older than 90 days.
B. Configure a webhook to monitor for secret scanning alert events.
C. Enable system for cross-domain identity management (SCIM) provisioning for the enterprise.
D. Document alternatives to storing secrets in the source code.
Answer: (SHOW ANSWER)
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.
: GitHub Docs - Managing and Responding to Secret Scanning Alerts

NEW QUESTION: 52
Secret scanning will ignore a secret_scanning.yml file that:
A. Is 100 KB or larger.
B. Is 1 MB or larger.
C. Contains 1,000 or more entries.
D. Has 1,000 or more directories.
Answer: (SHOW ANSWER)
GitHub documents a size limit for the .github/secret_scanning.yml configuration file. If the file is larger than
1 MB, secret scanning ignores the entire configuration file. This file is used primarily to define paths-ignore entries for directories or files where matching secrets should automatically be excluded or alerts closed as ignored by configuration. There is also a separate limit on path exclusions: if paths-ignore contains more than
1,000 entries, GitHub processes only the first 1,000 directories rather than ignoring the complete configuration file. Therefore, the 1,000-entry condition has different behavior from exceeding the file-size limit. A 100 KB file remains below GitHub's documented threshold. Consequently, option B accurately identifies the condition that causes the entire secret_scanning.yml file to be ignored.

NEW QUESTION: 53
What filter or sort settings can be used to prioritize the secret scanning alerts that present the most risk?
A. Sort to display the oldest first
B. Sort to display the newest first
C. Filter to display active secrets
D. Select only the custom patterns
Answer: (SHOW ANSWER)
The best way to prioritize secret scanning alerts is to filter by active secrets - these are secrets GitHub has confirmed are still valid and could be exploited. This allows security teams to focus on high-risk exposures that require immediate attention.
Sorting by time or filtering by custom patterns won't help with risk prioritization directly.
: GitHub Docs - Filtering Secret Scanning Alerts

NEW QUESTION: 54

What does a CodeQL database of your repository contain?
A. A build for Go projects to set up the project
B. A build of the code and extracted data
C. Build commands for C/C++, C#, and Java
D. A representation of all of the source code
GitHub
Agentic AI for AppSec Teams
Answer: (SHOW ANSWER)
Comprehensive and Detailed Explanation:
A CodeQL database contains a representation of your codebase, including the build of the code and extracted data. This database is used to run CodeQL queries to analyze your code for potential vulnerabilities and errors.
GitHub Docs
References: GitHub Docs - Preparing your code for CodeQL analysis

NEW QUESTION: 55
What is the format of the GitHub security advisory form?
A. A CVE Numbering Authority (CNA) description format
B. A Dependabot alert sent to the affected repositories
C. A form matching the MITRE database security advisory format
D. A form matching the Common Vulnerabilities and Exposures (CVE) description format
Answer: (SHOW ANSWER)
GitHub states that its repository security advisory form is a standardized form that matches the Common Vulnerabilities and Exposures (CVE) description format. GitHub Security Advisories build on the CVE ecosystem, and GitHub itself is a CVE Numbering Authority capable of assigning CVE identification numbers for eligible vulnerabilities. The advisory form captures structured information such as the vulnerability description, affected products, vulnerable and patched versions, severity, CWEs, references, and CVE information. A CNA is an organization authorized to assign CVEs rather than a separate advisory- description format. Dependabot alerts are generated for vulnerable dependencies and are unrelated to the structure of the security advisory form. Therefore, the CVE description format is the correct choice.

NEW QUESTION: 56
How many alerts are created when two instances of the same secret value are in the same repository?
A. 1
B. 2
C. 3
D. 4
Answer: (SHOW ANSWER)

When multiple instances of the same secret value appear in a repository, only one alert is generated. Secret scanning works by identifying exposed credentials and token patterns, and it groups identical matches into a single alert to reduce noise and avoid duplication.
This makes triaging easier and helps teams focus on remediating the actual exposed credential rather than reviewing multiple redundant alerts.
: GitHub Docs - About secret scanning alerts

NEW QUESTION: 57
Which of the following would raise secret scanning alerts?
A. GitHub personal access token
B. Structured Query Language (SQL) injection
C. Cross-site scripting (XSS)
D. Server-side request forgery
Answer: A (LEAVE A REPLY)
A GitHub personal access token (PAT) is a credential and is therefore a type of value that secret scanning is designed to detect. GitHub supports patterns for GitHub-issued tokens and can generate secret scanning alerts when supported PAT formats are committed to repositories. GitHub also performs validity checks for certain GitHub tokens and can distinguish active from inactive credentials. SQL injection, cross-site scripting, and server-side request forgery are application vulnerabilities rather than exposed credentials; those issues are detected through code scanning tools such as CodeQL. The distinction is important: secret scanning looks for credentials, API keys, passwords, tokens, and similar sensitive values, whereas code scanning analyzes source code for insecure programming patterns. Therefore, only the GitHub PAT belongs to secret scanning.

Valid GH-500 Dumps shared by ExamDiscuss.com for Helping Passing GH-500 Exam! ExamDiscuss.com now offer the **newest GH-500 exam dumps**, the ExamDiscuss.com GH-500 exam **questions have been updated** and **answers have been corrected** get the **newest** ExamDiscuss.com GH-500 dumps with Test Engine here: https://www.examdiscuss.com/Microsoft/exam/GH-500/premium/ (**127** Q&As Dumps, **35%OFF** **Special Discount Code: freecram**)