Microsoft

GH-500

GitHub Advanced Security Exam

For More Information – Visit link below:

https://www.examsempire.com/

Product Version

1. **Up to Date products, reliable and verified.**
2. **Questions and Answers in PDF Format.**



Latest Version: 6.2

Question: 1

Assuming that no custom patterns are configured, what type of secret is detected by secret scanning?

A. Usernames
B. Personally Identifiable Information (PII)
C. Private keys
D. Sealed boxes

Answer: C

Explanation:
GitHub secret scanning is designed to identify credentials and other sensitive values that match supported secret patterns. Among the options listed, private keys are the supported secret type. GitHub classifies private keys such as RSA, OpenSSH, EC, and PGP private keys as generic secret patterns. These are distinct from organization-specific custom patterns, which administrators create by defining their own regular expressions. Usernames and general PII are not treated as standard secret-scanning credential patterns, while “sealed boxes” are not a supported secret category. In current GitHub terminology, generic-pattern detection is separately configurable, so it may need to be enabled for the repository. Nevertheless, of the choices presented, private keys are the valid secret-scanning type.

Question: 2

To be compatible with code scanning, what data format must third-party code scanning tools use for output?

A. Static Analysis Results Interchange Format (SARIF)
B. YAML
C. ESLint
D. ECMAScript

Answer: A

Explanation:
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.

Question: 3

Which GitHub Advanced Security options are available under the Security section of the GitHub Enterprise Server Management Console? (Each answer presents part of the solution. Choose two.)

A. Secret scanning
B. Code scanning
C. Dependency review
D. Dependabot version updates

Answer: A, B

Explanation:
For GitHub Enterprise Server, secret scanning and code scanning are Advanced Security capabilities that require instance-level availability and configuration before organizations and repositories can use them. GitHub's GHES administration documentation identifies code scanning and secret scanning as security products that site administrators enable and configure for the appliance. Dependency review operates through repository dependency data and GitHub Actions rather than being one of the corresponding appliance-level Management Console options described by this question. Dependabot version updates are also a separate Dependabot capability for automatically keeping dependencies current and are not the requested GHAS Management Console security option. GitHub's newer licensing terminology may refer to these capabilities through GitHub Secret Protection and GitHub Code Security, but the functional distinction tested by the question remains the same.

Question: 4

Which organization policy lets organizations choose whether to allow members to view dependency insights?

A. Enable all
B. Enabled
C. Disabled
D. No policy

Answer: D

Explanation:
The correct selection is No policy. At the enterprise level, GitHub uses policy states to determine whether a setting is imposed on organizations or left under organization-level control. GitHub's enterprise administration schema defines NO_POLICY as meaning that no enterprise policy has been set for organizations. Consequently, the enterprise does not force the feature to be enabled or disabled, allowing organization owners to make the applicable decision within their own organization. By contrast, an Enabled or Disabled enterprise policy explicitly determines the state for organizations and therefore removes that discretion. “Enable all” is not the relevant policy state represented by this setting. This distinction is important in hierarchical GitHub governance because enterprise policies can either enforce behavior globally or deliberately delegate configuration to individual organizations.

Question: 5

You are tasked with filtering queries in a CodeQL query suite. Which metadata tag matches on the last path component?

A. query path
B. tags contain all
C. query filename
D. tags contain

Answer: C

Explanation:
In a CodeQL query suite, the query filename constraint matches the final path component—that is, the filename of the query file itself. GitHub distinguishes this from query path, which matches the path to the query relative to the enclosing CodeQL pack. The tags contain and tags contain all constraints operate on values from the query's @tags metadata rather than its filesystem path. tags contain requires one of the specified strings to match a tag component, while tags contain all requires all specified strings to be represented. Therefore, when the objective is specifically to filter queries according to the last component of their file path, query filename is the appropriate CodeQL query-suite constraint.

Thank You for Trying Our Product

Special 16 USD Discount Coupon: NSZUBG3X

Email: support@examsempire.com

Check our Customer Testimonials and ratings available on every product page.

Visit our website.

https://examsempire.com/