Eccouncil 312-97 Practice Questions

Shared by Woodward on 17-08-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

(Michael Rady recently joined an IT company as a DevSecOps engineer. His organization develops software products and web applications related to online marketing. Michael deployed a web application on Apache server. He would like to safeguard the deployed application from diverse types of web attacks by deploying ModSecurity WAF on Apache server. Which of the following command should Michael run to install ModSecurity WAF?)

Options:

A- sudo apt install libapache2-mod-security2 --y.
B- sudo apt install libapache2-mod-security2 --x.
C- sudo apt install libapache2-mod-security2 --w.
D- sudo apt install libapache2-mod-security2 --z.

Answer:

A

Explanation:

On Debian- and Ubuntu-based systems, ModSecurity for Apache is installed using the package libapache2-mod-security2. The correct command to install this package is sudo apt install libapache2-mod-security2 -y, where the -y flag automatically confirms installation prompts. The other options include invalid flags that are not recognized by the package manager and would result in command failure. Installing ModSecurity during the Operate and Monitor stage provides an additional layer of defense by inspecting incoming HTTP requests and blocking malicious traffic such as SQL injection, cross-site scripting, and protocol violations. A Web Application Firewall helps protect deployed applications from common attack vectors and supports defense-in-depth strategies in production environments.

Question 2

Question Type: MultipleChoice

(Patrick Fisher is a DevSecOps engineer in an IT company that develops software products and web applications. He is using IAST to analyze code for security vulnerabilities and to view real-time reports of the security issues. Patrick is using IAST in development, QA, and production stages to detect the vulnerabilities from the early stage of development, reduce the remediation

cost, and keep the application secure. How can IAST perform SAST on every line of code and DAST on every request and response?.)

Options:

A- Because IAST has access to server and local machine.
B- Because IAST has access to the code and HTTP traffic.
C- Because IAST has access to offline and runtime environment.
D- Because IAST has access to internal and external agents.

Answer:

B

Explanation:

Interactive Application Security Testing (IAST) works by instrumenting the application at runtime, allowing it to observe both the source code execution paths and the HTTP requests and responses flowing through the application. Because of this dual visibility, IAST can analyze every executed line of code (similar to SAST) while also monitoring real-time application behavior (similar to DAST). This unique capability enables highly accurate vulnerability detection with fewer false positives. The other options do not correctly explain how IAST achieves this hybrid analysis. Access to both code and HTTP traffic is what allows IAST to bridge static and dynamic testing techniques, making it highly effective across development, QA, and production environments.

Question 3

Question Type: MultipleChoice

(Lara Grice has been working as a DevSecOps engineer in an IT company located in Denver, Colorado. Her team leader has told her to save all the container images in the centos repository to centos-all.tar. Which option best is a STDOUT command that Lara can use to save all the container images in the centos repository to centos-all.tar?.)

Options:

A- # docker save centos > centos all.tar.
B- # docker save centos > centos-all.tar.
C- # docker save centos < centos all.tar.
D- # docker save centos < centos-all.tar.

Answer:

B

Explanation:

The docker save command exports one or more Docker images to a tar archive by writing the image data to standard output (STDOUT). To redirect this output into a file, the > redirection operator is used. The correct syntax is docker save <image> > <filename>.tar. In this scenario, the image repository name is centos, and the desired archive file is centos-all.tar, making option B correct. Options C and D incorrectly use input redirection (<) instead of output redirection. Option A includes a space in the filename (centos all.tar), which would be interpreted as two separate arguments and cause an error unless quoted. Saving images to a tar archive is a common operational task used for backups, transfers between environments, or offline analysis during the Operate and Monitor stage.

Question 4

Question Type: MultipleChoice

(SNF Pvt. Ltd. is a software development company located in Denver, Colorado. The organization is using pytm, which is a Pythonic Framework for threat modeling, to detect security issues and mitigate them in advance. James Harden has been working as a DevSecOps engineer at SNF Pvt. Ltd. for the past 3 years. He has created a tm.py file that describes an application in which the user logs the app and posts the comments on the applications. These comments are stored by the application server in the database and AWS lambda cleans the database. Which of the following command James can use to generate a sequence diagram?)

Options:

A- tm.py --seq | java -Djava.awt.headless=true -jar plantuml.jar -tpng -pipe > seq.png.
B- The integrated PHPStan into the AWS pipeline will invoke AWS CloudFormation to parse and send result to the security hub.
C- tm.py --seq | java -Djava.awt.headless=true -jar plantum.jar -tpng -pipe > seq.png.
D- The integrated PHPStan into the AWS pipeline will invoke AWS Elastic BeanStalk to parse and send result to the security hub.

Answer:

A

Explanation:

The pytm framework generates threat models that can be visualized using PlantUML diagrams. To create a sequence diagram, the --seq option is used with the model file, and the output is piped to the PlantUML processor. The correct command must reference the correct Java system property -Djava.awt.headless=true, which allows diagram rendering in environments without a graphical interface, such as CI/CD pipelines. Additionally, the correct jar file name is plantuml.jar. Options using lowercase -d instead of uppercase -D are invalid, and commands referencing plantum.jar are incorrect due to a misspelled jar name. Generating sequence diagrams during the Plan stage helps DevSecOps teams visualize data flows, understand attacker paths, and identify security threats early in the application design phase.

Question 5

Question Type: MultipleChoice

(Cheryl Hines has been working as a senior DevSecOps engineer over the past 5 years in an IT company. Due to the robust features offered by Keywhiz secret management tool such as compatibility with all software, untraceable secrets, no impact of power cut or server outage, etc., Cheryl's organization is using it for managing and distributing secrets. To add a secret using Keywhiz CLI, which of the following commands should Cheryl use?)

Options:

A- $ keywhiz.cli --devTrustStore --user keywhizAdmin login
$ keywhiz.cli add secret --name mySecretName < mySecretFile.
B- $ keywhiz.cli --devsecTrustStore --admin keywhizAdmin login
$ keywhiz.cli add secret --name mySecretName < mySecretFile.
C- $ keywhiz.cli --devTrustStore --admin keywhizAdmin login
$ keywhiz.cli add secret --name mySecretName < mySecretFile.
D- $ keywhiz.cli --DevSecTrustStore --user keywhizAdmin login
$ keywhiz.cli add secret --name mySecretName < mySecretFile.

Answer:

C

Explanation:

Keywhiz CLI requires authentication before secrets can be added. The correct process involves logging in using the --devTrustStore option and authenticating as an administrator using the --admin flag. Once authenticated, the add secret command is used with input redirection to

securely store the secret. Options that use incorrect flag names, incorrect casing, or invalid trust store identifiers do not follow Keywhiz CLI syntax. Adding secrets through Keywhiz instead of embedding them in code supports secure secret distribution and management, which is a fundamental aspect of DevSecOps culture. This approach ensures secrets remain protected, auditable, and available even during outages.

Question 6

Question Type: MultipleChoice

(Kevin Williamson has been working as a DevSecOps engineer in an MNC company for the past 5 years. In January of 2017, his organization migrated all the applications and data from on-prem to AWS cloud due to the robust security feature and cost-effective services provided by Amazon. His organization is using Amazon DevOps services to develop software products securely and quickly. To detect errors in the code and to catch bugs in the application code, Kevin integrated PHPStan into the AWS pipeline for static code analysis. What will happen if security issues are detected in the application code?.)

Options:

A- The integrated PHPStan into the AWS pipeline will invoke AWS CloudFormation to parse and send result to the security hub.
B- The integrated PHPStan into the AWS pipeline will invoke AWS Config to parse and send result to the security hub.
C- The integrated PHPStan into the AWS pipeline will invoke AWS Elastic BeanStalk to parse and send result to the security hub.
D- The integrated PHPStan into the AWS pipeline will invoke the AWS Lambda function to parse and send result to the security hub.

Answer:

D

Explanation:

In AWS-based DevSecOps pipelines, static analysis tools such as PHPStan commonly send their results to AWS services through event-driven processing. When PHPStan detects security issues, the results are typically parsed and processed by an AWS Lambda function, which can transform findings and forward them to AWS Security Hub. CloudFormation is used for infrastructure provisioning, AWS Config evaluates configuration compliance, and Elastic Beanstalk is an application deployment service---none of these are suited for parsing and relaying scan results. Lambda functions provide a scalable and serverless way to handle scan outputs automatically. This integration ensures that security findings are centralized, visible,

and actionable, aligning with secure automation practices during the Code stage.

Question 7

Question Type: MultipleChoice

(Christopher Brown has been working as a DevSecOps engineer in an IT company that develops software and web applications for an ecommerce company. To automatically detect common security issues and coding error in the C++ code, she performed code scanning using CodeQL in GitHub. Which of the following entries will Christopher find for CodeQL analysis of C++ code?)

Options:

A- CodeQL/Analyze (cp) (pull-request).
B- CodeQL/Analyze (cp) (push-request).
C- CodeQL/Analyze (cpp) (push-request).
D- CodeQL/Analyze (cpp) (pull-request).

Answer:

D

Explanation:

When GitHub Code Scanning is enabled using CodeQL, each supported programming language is identified by a specific language key. For C++ code, CodeQL uses the identifier cpp, not ''cp.'' CodeQL workflows are commonly configured to run during pull request events so that security issues and coding errors can be detected and reviewed before code is merged into the main branch. As a result, the CodeQL analysis entry displayed in GitHub Actions and the Security tab for C++ pull request analysis appears as CodeQL/Analyze (cpp) (pull-request). Options A and B are incorrect because ''cp'' is not a valid CodeQL language identifier. Option C uses the correct language identifier but references an incorrect event format. Identifying the correct CodeQL analysis entry helps DevSecOps engineers confirm that scans are executing correctly for the intended language during the Code stage and that security feedback is available early in the development lifecycle.

Question 8

Question Type: MultipleChoice

(Steven Gerrard has been working as a DevSecOps engineer at an IT company that develops software products and applications related to the healthcare industry. His organization has been using Azure DevOps services to securely and quickly develop software products. To ensure that the deployed infrastructure is in accordance with the architecture and industrial standards and the security policies are appropriately implemented, she would like to integrate InSpec with Azure. Therefore, after installation and configuration of InSpec, she created InSpec profile file and upgraded it with personal metadata and Azure resource pack information; then she wrote the InSpec tests. Which of the following commands should Steven use to run InSpec tests to check the compliance of Azure infrastructure?)

Options:

A- inspec exe inspec-tests/integration/ -t azure://.
B- inspec exec inspec-tests/integration/ -it azure://.
C- inspec exec inspec-tests/integration/ -t azure://.
D- inspec exe inspec-tests/integration/ -it azure://.

Answer:

C

Explanation:

Chef InSpec executes compliance tests using the inspec exec command. When testing Azure infrastructure, InSpec requires a target specification using the -t flag with the Azure transport identifier azure://. The correct command is inspec exec inspec-tests/integration/ -t azure://. Options using exe instead of exec are invalid due to incorrect command spelling. Options that use the -it flag misuse command-line parameters that are not intended for target selection. Running InSpec tests in this way allows DevSecOps teams to validate that Azure resources comply with architectural, security, and regulatory requirements. Integrating these checks into the Build and Test stage ensures continuous compliance and reduces the risk of insecure infrastructure reaching production environments.

Question 9

Question Type: MultipleChoice

(Gabriel Bateman has been working as a DevSecOps engineer in an IT company that develops virtual classroom software for online teaching. He would like to clone the BDD security framework on his local machine using the following URL, https://github.com/continuumsecurity/bdd-security.git. Which option best command should Gabriel use to clone the BBD security framework?)

Options:

A- github clone https://github.com/continumsecurity/bdd-security.git.
B- git clone https://github.com/continuumsecurity/bdd-security.git.
C- git clone https://github.com/continumsecurity/bdd-security.git.
D- github clone https://github.com/continuumsecurity/bdd-security.git.

Answer:

B

Explanation:

To clone a repository from GitHub, the correct command is git clone followed by the accurate repository URL. The organization name continuumsecurity and repository name bdd-security must be spelled correctly for the command to succeed. Options using github clone are invalid because github is not a standard Git command-line utility. Options with misspelled organization names will result in errors. Cloning security testing frameworks during the Code stage enables DevSecOps engineers to evaluate, customize, and integrate security automation tools into development workflows, supporting secure application development and testing practices.

Question 10

Question Type: MultipleChoice

(Jason Wylie has been working as a DevSecOps engineer in an IT company located in Sacramento, Californi

a. He would like to use Jenkins for CI and Azure Pipelines for CD to deploy a Spring Boot app to an Azure Container Service (AKS) Kubernetes cluster. He created a namespace for deploying the Jenkins in AKS, and then deployed the Jenkins app to the Pod. Which of the following commands should Jason run to see the pods that have been spun up and running?)

Options:

A- kubectl get pods -k Jenkins.
B- kubectl get pods -s jenkins.
C- kubectl get pods -n jenkins.
D- kubectl get pods -p jenkins.

Answer:

C

Explanation:

Kubernetes uses namespaces to logically isolate resources such as pods, services, and deployments. When an application like Jenkins is deployed into a specific namespace, the correct way to view the pods running in that namespace is by using the -n (or --namespace) flag with the kubectl get pods command. The command kubectl get pods -n jenkins instructs Kubernetes to list all pods in the ''jenkins'' namespace. The other options use invalid or unrelated flags that are not supported for namespace selection. Verifying pod status during the Release and Deploy stage is essential to ensure that applications have been deployed successfully and are running as expected before exposing services or proceeding to monitoring. This step supports deployment validation and operational readiness in Kubernetes-based DevSecOps environments.

Question 11

Question Type: MultipleChoice

(Teresa Wheeler is a DevSecOps engineer at Altschutz Solution Pvt. Ltd. She would like to test the web applications and API's from outside without accessing the source code using BDD security framework. The framework is a collection of Cucumber-JVM features that are pre-configured with OWASP ZAP, Nessus scanner, SSLyze, and Selenium. Hence, she downloaded and ran the jar application, and then cloned the BDD security framework. Next, she utilized a command for executing the authentication feature. Which of the following commands allows Teresa to execute all the features of BDD security framework, including the OWASP ZAP?.)

Options:

A- ./gardlew.
B- /gardlev.
C- /gardlew.
D- ./gardlev.

Answer:

A

Explanation:

The Gradle wrapper script used to execute all features in the BDD Security framework on Unix-like systems is ./gradlew. The dot-slash prefix indicates execution from the current directory, which is required when running scripts locally. Options using /gardlew or /gardlev imply incorrect paths or misspelled wrapper names. Executing ./gradlew without additional parameters runs the default task, which includes all configured features such as OWASP ZAP, Nessus, SSLyze, and Selenium tests. Running all features during the Build and Test stage provides comprehensive external security testing coverage, helping identify vulnerabilities without needing access to source code.

Question 12

Question Type: MultipleChoice

(Kevin Ryan has been working as a DevSecOps engineer in an MNC company that develops various software products and web applications. For easy management of secret credentials in CI/CD pipeline, he would like to integrate Azure Key Vault with Jenkins. Therefore, he created an Azure Key Vault, noted down the credentials displayed on the screen, and created a secret in Azure Key Vault. Then, he used the secret key from the credentials obtained from creating the vault. Kevin went back to Jenkins and installed Azure Key Vault plugin. Then, he navigated to Configure System under Manage Jenkins and added the URL for Azure Key Vault. How can Kevin complete the integration of Azure Key Vault with Jenkins?.)

Options:

A- By modifying old credentials in Global Credentials (unrestricted).
B- By creating new credentials in Global Credentials (unrestricted).
C- By creating new credentials in Global Credentials (restricted).
D- By modifying old credentials in Global Credentials (restricted).

Answer:

B

Explanation:

To complete Azure Key Vault integration with Jenkins, Kevin must create new credentials in Jenkins under Global Credentials (unrestricted). These credentials store the Azure client ID, client secret, tenant ID, and subscription details required by the Azure Key Vault plugin to authenticate securely. Modifying old credentials can lead to misconfiguration or credential reuse risks, while restricted credentials may prevent the plugin from accessing secrets across pipelines. Creating new unrestricted credentials ensures proper authentication and controlled access to secrets during the Code stage, supporting secure secret management across CI/CD workflows.