Download Eccouncil 312-97 Exam Dumps Free

Shared by Good on 17-06-2026

For More Free Questions and Preparation Resources

Check the Links on Last Page

Question 1

Question Type: MultipleChoice

(Andrew Gerrard has recently joined an IT company located in Fairmont, California, as a DevSecOps engineer. Due to robust security and cost-effective service provided by AWS, his organization has migrated all the workloads from on-prem to AWS cloud in January of 2020. Andrew's team leader has asked him to integrate AWS Secret Manager with Jenkins. To do so, Andrew installed the ''AWS Secret Manager Credentials provider'' plugin in Jenkins and configured an IAM policy in AWS that allows Jenkins to take secrets from AWS Secret manager. Which option best file should Andrew edit to add access id and secret key parameters along with the region copied from AWS?.)

Options:

A- /etc/file/Jenkins.
B- /etc/sysconfig/Jenkins.
C- /etc/sysconfig file/Jenkins.
D- /etc/filebeat/filebeat.yml.

Answer:

B

Explanation:

On Linux systems, Jenkins environment variables such as AWS access key ID, secret access key, and default region are commonly configured in the /etc/sysconfig/Jenkins file. This file allows administrators to define environment variables that are loaded when the Jenkins service starts. By placing AWS credentials and region information in this file, Jenkins jobs and plugins---such as the AWS Secrets Manager Credentials Provider---can securely access AWS resources. The other options reference invalid paths or unrelated configuration files (such as Filebeat). Editing /etc/sysconfig/Jenkins ensures consistent credential availability across Jenkins jobs while supporting secure integration with AWS services during the Code stage.

Question 2

Question Type: MultipleChoice

(Curtis Morgan has been working as a software developer in an MNC company. His team has

developed a NodeJS application. While doing peer review of the NodeJS application, he observed that there are insecure libraries in the application. Therefore, he approached, Teresa Lisbon, who is working as a DevSecOps engineer, to detect the insecure libraries in the NodeJS application. Teresa used a SCA tool to find known vulnerabilities in JavaScript libraries for Node.JS applications and detected all the insecure libraries in the application. Which option best tools did Teresa use for detecting insecure libraries in the NodeJS application?)

Options:

A- Bandit.
B- Bundler-Audit.
C- Retire.js.
D- Tenable.io.

Answer:

C

Explanation:

Retire.js is a Software Composition Analysis (SCA) tool designed specifically to identify known vulnerabilities in JavaScript libraries used in web and NodeJS applications. It scans dependencies and compares detected versions against a vulnerability database to identify insecure libraries. Bandit is a static analysis tool for Python, Bundler-Audit is used for Ruby dependencies, and Tenable.io focuses on infrastructure and vulnerability management rather than JavaScript libraries. Using Retire.js during the Code stage allows DevSecOps teams to identify insecure third-party dependencies early, reducing the likelihood of vulnerable libraries being deployed into production. This supports shift-left security and strengthens the application's overall security posture.

Question 3

Question Type: MultipleChoice

(Dustin Hoffman has been working as a DevSecOps engineer in an IT company located in San Diego, Californi

a. For detecting new security vulnerabilities at the beginning of the source code development, he would like to integrate Checkmarx SCA tool with GitLab. The Checkmarx template has all the jobs defined for pipeline. Where should Dustin incorporate the Checkmarx template file 'https://raw.githubusercontent.com/checkmarx-ltd/cx-flow/develop/templates/gitlab/v3/Checkmar

x.gitlab-ci.yml'?)

Options:

A- gitlab-cd.yml root directory.
B- gitlab-ci/cd.yml root directory.
C- gitlab.yml root directory.
D- gitlab-ci.yml root directory.

Answer:

D

Explanation:

GitLab CI/CD pipelines are defined using a configuration file named gitlab-ci.yml, which must be placed in the root directory of the repository. This file controls pipeline stages, jobs, and template inclusions. To integrate Checkmarx SCA using a predefined template, the template reference must be included in the root-level gitlab-ci.yml file so GitLab can load and execute the defined jobs automatically. The other filenames listed in the options are not recognized by GitLab as valid pipeline configuration files. Integrating SCA at the Code stage allows early detection of vulnerable open-source dependencies, reducing remediation cost and preventing insecure components from progressing further in the DevSecOps pipeline.

Question 4

Question Type: MultipleChoice

(Dustin Hoffman is a DevSecOps engineer at SantSol Pvt. Ltd. His organization develops software products and web applications related to mobile apps. Using Gauntlt, Dustin would like to facilitate testing and communication between teams and create actionable tests that can be hooked in testing and deployment process. Which of the following commands should Dustin use to install Gauntlt?.)

Options:

A- $ gems install Gauntlt.
B- curl -i https://download.sqreen.com/php/install.sh > sqreen-install.sh \ && bash sqreen-install.sh [CHARLOTTE'S ORG TOKEN HERE] ''[CHARLOTTE'S APP NAME HERE]''.

C- $ gem install gauntlt.
D- curl -s https://download.sqreen.com/php/install.sh < sqreen-install.sh \ && bash sqreen-install.sh [CHARLOTTE'S ORG TOKEN HERE] "[CHARLOTTE'S APP NAME HERE]".

Answer:

C

Explanation:

Gauntlt is a security testing framework written in Ruby and distributed as a Ruby gem. The correct way to install a Ruby gem is using the gem install command followed by the lowercase gem name. RubyGems are case-sensitive and standardized to lowercase naming conventions, which makes gem install gauntlt the correct command. The gems command does not exist in Ruby's package management ecosystem, and using uppercase names such as Gauntlt can lead to installation failures. Installing Gauntlt allows DevSecOps teams to write human-readable security tests and integrate them into CI/CD pipelines, enabling automated and collaborative security validation during the Build and Test stage.

Question 5

Question Type: MultipleChoice

(Rockmond Dunbar is a senior DevSecOps engineer in a software development company. His organization develops customized software for retail industries. Rockmond would like to avoid setting mount propagation mode to share until it is required because when a volume is mounted in shared mode, it does not limit other containers to mount and modify that volume. If mounted volume is sensitive to changes, then it would be a serious security concern. Which of the following commands should Rockmond run to list out the propagation mode for mounted volumes?.)

Options:

A- docker ps -quiet -all | xargs docker inspect -format ': Propagation='.
B- docker ps --quiet --all | xargs docker inspect --format ': Propagation'.
C- docker ps --quiet --all | xargs docker inspect --format ': Propagation='.
D- docker ps -quiet -all | xargs docker inspect -format ': Propagation'.

Answer:

C

Explanation:

To inspect mount propagation modes for Docker containers, Rockmond needs to list all container IDs and then inspect their configuration. The docker ps --quiet --all command outputs container IDs only, which are then passed to docker inspect using xargs. The --format option allows extraction of specific fields, such as mount propagation settings. Option C correctly uses valid flags (--quiet --all) and proper formatting syntax. Options A and D incorrectly use single hyphens, and option B omits the equals sign, which is required to display the propagation value. Inspecting mount propagation during the Operate and Monitor stage helps prevent unintended privilege escalation or data modification by other containers, aligning with container hardening best practices.

Question 6

Question Type: MultipleChoice

(Thomas Gibson has been working as a DevSecOps engineer in an IT company that develops software products and web applications related to law enforcement. To automatically execute a scan against the web apps, he would like to integrate InsightAppSec plugin with Jenkins. Therefore, Thomas generated a new API Key in the Insight platform. Now, he wants to install the plugin manually. How can Thomas install the InsightAppSec plugin manually in Jenkins?)

Options:

A- By creating a .conf file and uploading to his Jenkins installation.
B- By creating a .war file and uploading to his Jenkins installation.
C- By creating a .zip file and uploading to his Jenkins installation.
D- By creating a .hpi file and uploading to his Jenkins installation.

Answer:

D

Explanation:

Jenkins plugins are distributed and installed as .hpi files. To manually install a plugin, administrators upload the .hpi file through the Jenkins Plugin Manager using the ''Upload Plugin'' option. This approach is commonly used in environments with restricted internet access or when

custom plugin versions are required. .war files are used for deploying the Jenkins application itself, not plugins, while .zip and .conf files are not recognized plugin formats. Installing the InsightAppSec plugin allows Jenkins pipelines to automatically trigger dynamic application security scans during the Build and Test stage. This integration ensures that web applications are continuously evaluated for vulnerabilities before deployment, supporting proactive security testing and risk reduction.

Question 7

Question Type: MultipleChoice

(Joyce Vincent has been working as a senior DevSecOps engineer at MazeSoft Solution Pvt. Ltd. She would like to integrate Trend Micro Cloud One RASP tool with Microsoft Azure to secure container-based application by inspecting the traffic, detecting vulnerabilities, and preventing threats. In Microsoft Azure PowerShell, Joyce created the Azure container instance in a resource group (ACI) (named ''aci-test-closh'') and loaded the container image to it. She then reviewed the deployment of the container instance. Which of the following commands should Joyce use to get the logging information from the container?.)

Options:

A- az container logs --resource-group ACI --name aci-test-closh.
B- az container logs -resource-group ACI -name aci-test-closh.
C- azure container logs --resource-group ACI --name aci-test-closh.
D- azure container logs -resource-group ACI -name aci-test-closh.

Answer:

A

Explanation:

Azure Container Instances (ACI) exposes container logs via the Azure CLI using the az container logs command. To retrieve logs, you must provide the resource group and the container group name using the long-form parameters --resource-group and --name. Option A matches the correct CLI structure and parameter format: az container logs --resource-group ACI --name aci-test-closh. Options B and D incorrectly use single-dash forms (-resource-group and -name), which are not valid for these long option names. Options C and D incorrectly use azure instead of az; the Azure CLI command group is invoked with az, not azure. Getting logs after deployment review is a critical Operate and Monitor activity: it helps confirm the container started correctly, diagnose runtime errors, and validate that runtime protection (such as a RASP/micro-agent) is

functioning. This visibility supports faster incident response and helps ensure the containerized workload remains secure and stable in its runtime environment.

Question 8

Question Type: MultipleChoice

(Evan Peters has been working as a DevSecOps engineer in an IT company located in Denver, Colorado. His organization has deployed various applications on Docker containers. Evan has been running SSH service inside the containers, and handling of SSH keys and access policies is a major security concern for him. What will be the solution for Evan security concern?)

Options:

A- Run SSH on the registry and utilize docker exec for interacting with the container.
B- Run SSH on the docker build and utilize docker exec for interacting with the container.
C- Run SSH on the client and utilize docker exec for interacting with the container.
D- Run SSH on the host and utilize docker exec for interacting with the container.

Answer:

D

Explanation:

Running an SSH service inside Docker containers is considered a security anti-pattern because it increases the attack surface and complicates key and access management. Containers are designed to run a single primary process and be managed externally rather than accessed via SSH. The recommended solution is to run SSH on the host system and use docker exec to interact with containers when administrative access is required. This approach eliminates the need to manage SSH keys inside containers, reduces exposure to brute-force attacks, and simplifies access control. The other options incorrectly suggest running SSH in inappropriate locations such as the registry, client, or build process, which do not address the core security concern. During the Operate and Monitor stage, minimizing unnecessary services within containers is critical to enforcing least privilege and maintaining a secure runtime environment.

Question 9

Question Type: MultipleChoice

(Allen Smith has been working as a senior DevSecOps engineer for the past 4 years in an IT company that develops software products and applications for retail companies. To detect common security issues in the source code, he would like to integrate Bandit SAST tool with Jenkins. Allen installed Bandit and created a Jenkins job. In the Source Code Management section, he provided repository URL, credentials, and the branch that he wants to analyze. As Bandit is installed on Jenkins' server, he selected Execute shell for the Build step and configure Bandit script. After successfully integrating Bandit SAST tool with Jenkins, in which of the following can Allen detect security issues?.)

Options:

A- Java code.
B- Ruby code.
C- Python code.
D- C++ code.

Answer:

C

Explanation:

Bandit is a Static Application Security Testing (SAST) tool developed specifically for analyzing Python source code. It scans Python scripts and applications to identify common security issues such as use of weak cryptography, hardcoded passwords, unsafe use of functions like eval, and insecure imports. Bandit works by parsing Python Abstract Syntax Trees (ASTs) and applying a set of security-focused rules. It does not support Java, Ruby, or C++ code, which require different static analysis tools tailored to their respective languages. By integrating Bandit with Jenkins during the Build and Test stage, Allen enables automated detection of Python-specific security flaws as soon as code changes are introduced. This shift-left approach reduces remediation costs, prevents vulnerable code from progressing further in the pipeline, and improves overall application security posture.

Question 10

Question Type: MultipleChoice

(William Scott has been working as a senior DevSecOps engineer at GlobalSec Pvt. Ltd. His organization develops software products related to mobile apps. William would like to exploit Jenkins using Metasploit framework; therefore, he downloaded Metasploit. He would like to

initiate an Nmap scan by specifying the target IP to find the version of Jenkins running on the machine. Which of the following commands should William use to find the version of Jenkins running on his machine using Nmap?.)

Options:

A- Nmap -sN -sJ ''Target IP''.
B- Nmap -sJ -sN ''Target IP''.
C- Nmap -sS -sV ''Target IP''.
D- Nmap -sV -sS ''Target IP''.

Answer:

D

Explanation:

To identify the version of a service running on a target system, Nmap uses the -sV option, which enables service version detection. The -sS flag specifies a TCP SYN scan, which is a common and efficient scanning method. Combining these two flags allows Nmap to discover open ports and accurately identify the service versions running on those ports, such as Jenkins. Options A and B reference invalid scan types (-sJ) and do not enable version detection. Option C includes the correct flags but places them in a less conventional order; however, the commonly accepted and documented usage is -sV -sS. Running this scan during the Operate and Monitor stage helps security teams understand exposed services and assess potential attack surfaces.

Question 11

Question Type: MultipleChoice

(SinCaire is a software development company that develops web applications for various clients. To measure the successful implementation of DevSecOps, the organization enforced U.S. General Service Administrator (GSA) high-value DevSecOps metrics. Which of the following metrics implemented by SinCaire can measure the time between the code commit and production, and tracks the bug fix and new features throughout the development, testing, and production phases?)

Options:

A- Mean time to recovery (for applications).
B- Change volume (for application).
C- Time to value.
D- Change lead time (for application).

Answer:

D

Explanation:

Change lead time measures the duration between a code commit and its successful deployment into production. This metric tracks how efficiently new features, bug fixes, and changes move through development, testing, and release stages. It is a key DevSecOps performance indicator used to assess pipeline efficiency and the effectiveness of automation and security integration. Mean time to recovery focuses on restoring service after incidents, change volume measures the number of changes rather than delivery speed, and time to value is a broader business metric. Change lead time directly reflects how well DevSecOps practices enable rapid yet secure delivery, making it the correct metric for measuring commit-to-production flow across all phases.