

ECCouncil.312-97.v2026-10-02.q65

Exam Code:	312-97
Exam Name:	EC-Council Certified DevSecOps Engineer (ECDE)
Certification Provider:	ECCouncil
Free Question Number:	65
Version:	v2026-10-02
# of views:	108
# of Questions views:	650
https://www.freepdfdumps.com/ECCouncil.312-97.v2026-10-02.q65.html	

NEW QUESTION: 1

Kenji Watanabe, a DevSecOps engineer at a Tokyo gaming studio, needs a testing tool that combines code instrumentation with live traffic analysis, so it can pinpoint the exact line of vulnerable code triggered when a QA tester clicks through the application during functional testing. Which approach should Kenji choose?

- A. IAST
- B. Penetration testing
- C. Threat modeling
- D. Chaos engineering

Answer: A (LEAVE A REPLY)

IAST works by instrumenting the application with agents that monitor code execution from within while the application is exercised through normal functional or QA testing, allowing it to correlate detected vulnerabilities directly back to specific lines of source code in real time -- precisely what Kenji needs. Penetration testing is typically a manual or semi-manual black-box/gray-box assessment performed by security testers simulating real-world attacks, and does not inherently tie findings to exact source lines through instrumentation during routine QA clicks. Threat modeling is a design-time planning activity performed before code execution. Chaos engineering intentionally injects failures into production or production-like systems to test resilience, not to detect code-level vulnerabilities during functional testing. Because Kenji wants instrumented, line-level detection during live QA interaction, IAST is correct.

NEW QUESTION: 2

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?

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

Answer: C (LEAVE A REPLY)

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.

NEW QUESTION: 3

Oliver Bennett, a DevSecOps engineer at a London insurance firm, discovers that a base container image his team relies on has an outdated OpenSSL package with a known critical CVE.

He wants an automated scanner integrated into the Build stage to flag such OS-level package vulnerabilities in container images before they are pushed to the registry. Which tool category should Oliver use?

- A. Container image vulnerability scanner (e.g., Trivy, Clair)
- B. Web Application Firewall
- C. Network Intrusion Detection System
- D. Git secret scanner

Answer: A (LEAVE A REPLY)

Container image vulnerability scanners such as Trivy, Clair, or Anchore inspect the layers of a container image, including the base OS packages and installed libraries, against known CVE databases, and can be integrated directly into the Build stage of a CI/CD pipeline to block or flag images before they reach the registry. This precisely matches Oliver's need to catch an outdated OpenSSL package with a known CVE pre-push. A Web Application Firewall protects a running web application at the network edge during Operate, not during image build. A Network Intrusion Detection System monitors network traffic for malicious activity in a live environment, not static image contents. A Git secret

scanner detects hardcoded credentials in repository history, not OS package vulnerabilities. Because Oliver needs pre-push detection of vulnerable OS packages inside a container image, a container image vulnerability scanner is correct.

NEW QUESTION: 4

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 of the following is a STDOUT command that Lara can use to save all the container images in the centos repository to centos-all.tar?

- 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: (SHOW ANSWER)

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.

NEW QUESTION: 5

William Friedkin has been working as a DevSecOps engineer in an IT company for the past 3 years. His team leader has asked him to validate the host configuration that runs the Docker containers and perform security checks at the container level by implementing Docker's CIS Benchmark Recommendations. Therefore, William would like to integrate Docker Bench with Jenkins to incorporate security testing in DevOps workflow and secure the Docker Container.

Before starting the procedure, he would like to install openssh on Ubuntu. Which of the following command should William run to install openssh on Ubuntu?

- A. sudo apt-get -s install openssh-server.
- B. sudo apt.get install openssh-server.
- C. sudo apt-get install openssh-server.
- D. sudo apt.get -s install openssh-server.

Answer: C (LEAVE A REPLY)

Ubuntu systems use the Advanced Package Tool (APT) for installing and managing software packages. The correct syntax for installing a package is sudo apt-get install

<package-name>. In this case, the OpenSSH server package required for secure remote access is named `openssh-server`, making `sudo apt-get install openssh-server` the correct command. Options using `apt.get` are invalid because the command syntax is incorrect. Options that include the `-s` flag only simulate the installation process and do not actually install the package. Installing OpenSSH is often required when integrating Docker Bench with Jenkins, as it enables secure communication and remote execution of compliance checks. Performing this setup during the Build and Test stage ensures that container hosts comply with Docker CIS Benchmark recommendations, helping organizations identify misconfigurations and security weaknesses early in the pipeline.

NEW QUESTION: 6

Terry Crews has been working as a DevSecOps engineer at an IT company that develops software products and web applications related to IoT devices. She integrated Sscreen RASP tool with Slack for sending notifications related to security issues to her team. How can Sscreen send notification alerts to Slack?

- A. By creating a cookbook, defining a trigger, security response, and notification.
- B. By creating a cookbook, defining a trigger, Alert a response, and notification.
- C. By creating a playbook, defining a trigger, Alert a response, and notification.
- D. By creating a playbook, defining a trigger, security response, and notification.

Answer: D (LEAVE A REPLY)

Sscreen provides runtime application self-protection (RASP) capabilities that allow teams to detect and respond to security threats in real time. Sscreen uses a structured automation mechanism called a playbook to define how security events are handled. A playbook consists of three key components: a trigger that detects suspicious or malicious behavior, a security response that defines what action Sscreen should take (such as blocking a request or flagging an attack), and a notification that sends alerts to external systems like Slack. The term "cookbook" is not used in Sscreen's alerting and response model, making options A and B incorrect. Option C incorrectly uses the phrase "Alert a response" instead of "security response," which does not accurately describe Sscreen's configuration model. By using playbooks, Sscreen enables automated detection, response, and team notification during the Operate and Monitor stage, ensuring rapid awareness and collaboration when security incidents occur.

NEW QUESTION: 7

Katie Holmes is working as a DevSecOps engineer at SeCSafe Anti-virus. The DevOps team of her organization has developed a distributed application with multiple microservices. Katie deployed all the microservices to the Kubernetes nodes successfully. The DevOps team approached Katie and informed her that the application is not working. Katie wants to check whether the Kubernetes cluster is working or not. Which of the following commands should Katie run step by step to verify that the Kubernetes is working?

- A. kube-etcd version kube-etcd cluster-info.
- B. kubernetes version kubernetes cluster-info.
- C. kube version kube cluster-info.
- D. kubectl version kubectl cluster-info.

Answer: D ([LEAVE A REPLY](#))

Kubernetes clusters are managed and inspected using the kubectl command-line tool. To verify whether a Kubernetes cluster is functioning correctly, administrators commonly run kubectl version to confirm that both the client and server components are reachable and operational. This is followed by kubectl cluster-info, which displays information about the cluster's control plane and core services. These commands together confirm API server availability, cluster connectivity, and basic health status. The other options list invalid command names such as kube, kubernetes, or kube- etcd, which are not used for standard cluster validation. Performing these checks during the Operate and Monitor stage helps quickly identify whether application issues stem from cluster-level problems or application-level misconfigurations. This supports faster troubleshooting and more reliable production operations.

NEW QUESTION: 8

Camila Duarte, a DevSecOps engineer at a Sao Paulo media company, is setting up her Jenkins pipeline to store database credentials, API keys, and TLS certificates outside of source code and configuration files, with automatic rotation and fine-grained access policies. Which type of tool should she integrate?

- A. Static code analyzer
- B. Secrets management tool (e.g., HashiCorp Vault)
- C. Container registry
- D. Load balancer

Answer: (SHOW ANSWER)

Secrets management tools such as HashiCorp Vault, AWS Secrets Manager, or Azure Key Vault are purpose-built to securely store, dynamically generate, rotate, and control access to sensitive credentials like API keys, database passwords, and certificates, removing the need to hardcode them into source code or pipeline configuration files. This directly matches Camila's requirement for automatic rotation and fine-grained access policies. A static code analyzer scans source code for vulnerabilities but does not manage or store secrets. A container registry stores and distributes container images, unrelated to credential management. A load balancer distributes network traffic across servers and has no role in secrets handling. Since Camila needs centralized, rotatable, access-controlled secret storage, a secrets management tool is correct.

NEW QUESTION: 9

Paul McCartney has been working as a senior DevSecOps engineer in an IT company over the past 5 years. He would like to integrate Conjur secret management tool into the

CI/CD pipeline to secure the secret credentials in various phases of development. To integrate Conjur with Jenkins, Paul downloaded Conjur.hpi file and uploaded it to the Upload Plugin section of Jenkins. Paul declared a policy branch using a code and saved it as a .yml file. Which of the following commands should Paul use to load this policy in Conjur root?

- A. `$ conjur policy load -f root -p < file-name >.`
- B. `$ conjur policy load -f root -b < file-name >.`
- C. `$ conjur policy load -p root -f < file-name >.`
- D. `$ conjur policy load -b root -f < file-name >.`

Answer: [\(SHOW ANSWER\)](#)

Conjur policies define access controls, authentication rules, and secret variables, and they must be loaded into the correct policy branch. The conjur policy load command uses the -b flag to specify the policy branch and the -f flag to specify the policy file. To load a policy into the root branch, the correct command is `conjur policy load -b root -f <file-name>.`

Options that reverse or misuse these flags are invalid and would either fail or load the policy incorrectly. Loading policies correctly during the Build and Test stage ensures that Jenkins pipelines can securely access secrets at runtime, enforcing centralized secret management, least-privilege access, and compliance with security requirements.

NEW QUESTION: 10

Hassan Al-Rashid, a build engineer at a Dubai fintech, wants to ensure that build artifacts cannot be tampered with between the CI pipeline and the artifact repository, and that consumers can cryptographically verify an artifact's origin and build process. Which framework/practice should Hassan adopt?

- A. SLSA (Supply-chain Levels for Software Artifacts)
- B. OWASP Top 10
- C. CIS Benchmarks
- D. MITRE ATT&CK

Answer: A [\(LEAVE A REPLY\)](#)

SLSA is a security framework specifically designed to protect the software supply chain by defining a set of increasing levels of build integrity requirements, including provenance generation, tamper-resistant build pipelines, and cryptographic attestations that let consumers verify how, where, and from what source an artifact was built. This directly addresses Hassan's need for tamper-evidence and origin verification between CI and the artifact repository. The OWASP Top 10 catalogs common web application vulnerability categories but does not define supply-chain build integrity controls. CIS Benchmarks provide configuration hardening guidance for operating systems and platforms, not build provenance. MITRE ATT&CK is a knowledge base of adversary tactics and techniques used for threat intelligence and detection engineering, not artifact integrity. Because Hassan specifically needs tamper resistance and provenance verification for build artifacts, SLSA is correct.

NEW QUESTION: 11

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 of the following tools did Teresa use for detecting insecure libraries in the NodeJS application?

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

Answer: C (LEAVE A REPLY)

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.

NEW QUESTION: 12

Sarah Wheeler is an experienced DevSecOps engineer. She recently joined an IT company that develops software products for customers stretched across the globe. Sarah would like to use a security testing tool that protects the application from false positives, network sniffing, tampering with code, etc. The tool should monitor the incoming traffic to the server and APIs for suspicious activities and help her team in remediating them during runtime. Which of the following tools should Sarah select that will help her team in precisely detecting and remediating the security issues in the application code during runtime?

- A. IAST.
- B. SAST.
- C. RASP.
- D. DAST.

Answer: C (LEAVE A REPLY)

Runtime Application Self-Protection (RASP) operates from within the application runtime environment, monitoring incoming traffic, API calls, and execution behavior in real time.

Because it has deep visibility into application logic and execution context, RASP can accurately detect attacks such as injection, tampering, and abnormal behavior while minimizing false positives.

SAST analyzes source code statically, DAST tests running applications externally, and IAST combines some runtime insight with testing but does not actively block threats.

RASP's ability to detect and remediate attacks during runtime makes it ideal for protecting applications in production environments, aligning with the Operate and Monitor stage of the DevSecOps pipeline.

NEW QUESTION: 13

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?

- A. `$ gems install Gauntlt.`
- B. `$ gems install gauntlt.`
- C. `$ gem install gauntlt.`
- D. `$ gem install Gauntlt.`

Answer: C (LEAVE A REPLY)

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.

NEW QUESTION: 14

Tobias Hartmann, a DevSecOps engineer at a Munich manufacturing firm, wants to simulate a full real-world attack chain - from initial reconnaissance through exploitation to lateral movement and data exfiltration - against a pre-production environment using both automated tools and skilled human testers, to validate defenses holistically. Which activity is Tobias planning?

- A. SAST scan
- B. Penetration testing / Red team exercise
- C. SBOM generation
- D. Static Dockerfile linting

Answer: B (LEAVE A REPLY)

Penetration testing, especially when extended into a red team exercise, involves skilled human testers combined with automated tooling simulating the full lifecycle of a real-world attack -- reconnaissance, exploitation, lateral movement, and exfiltration -- against systems to holistically validate the effectiveness of defenses, which precisely matches Tobias's plan. A SAST scan is a narrow, automated, code-level analysis technique that does not simulate a full multi-stage attack chain or involve human adversarial testers. SBOM generation produces a component inventory and has no offensive testing component whatsoever. Static Dockerfile linting checks build-file syntax against best practices and is unrelated to simulating an end-to-end attack. Because Tobias needs a holistic, human-plus-automated simulation of a full attack chain, penetration testing/red teaming is correct.

NEW QUESTION: 15

Marcus Alberts, a DevSecOps engineer at a Berlin automotive supplier, must ensure that every artifact promoted from the Build stage to the Release stage is digitally signed, and that the deployment pipeline refuses to deploy any artifact whose signature cannot be verified. What security principle is Marcus enforcing?

- A. Least privilege
- B. Artifact/code signing and verification
- C. Network segmentation
- D. Data masking

Answer: B (LEAVE A REPLY)

Artifact or code signing applies a cryptographic signature to a build artifact so that its authenticity and integrity can be verified before deployment; enforcing signature verification as a pipeline gate prevents unsigned or tampered artifacts from being deployed, which is exactly what Marcus is implementing. Least privilege refers to limiting user or service permissions to the minimum necessary for their function, not artifact integrity verification. Network segmentation divides a network into isolated zones to limit lateral movement, unrelated to artifact signing. Data masking obscures sensitive data values (e.g., in non-production environments) and has nothing to do with verifying artifact authenticity. Because Marcus's control specifically concerns signing and verifying artifacts before deployment, artifact/code signing and verification is correct.

NEW QUESTION: 16

Alex Hales recently joined TAVR Software Solution Pvt. Ltd. As a DevSecOps engineer. To automatically detect security loopholes in the web applications while building and testing them, he integrated OWASP ZAP DAST Plugin with Jenkins. How can Alex uniquely identify every build in the project?

- A. By specifying a file name followed by \${Profile_ID} in Post-build Actions tab.
- B. By specifying a file name followed by \${zap_scan} in Post-build Actions tab.
- C. By specifying a file name followed by \${ZAPROXY_HOME} in Post-build Actions tab.
- D. By specifying a file name followed by \${Build_ID} in Post-build Actions tab.

Answer: D (LEAVE A REPLY)

Jenkins automatically assigns a unique identifier to each build using the environment variable BUILD_ID. When integrating OWASP ZAP with Jenkins, appending \${BUILD_ID} to output filenames or reports ensures that every scan result corresponds to a specific build execution. This avoids overwriting previous reports and allows traceability between build artifacts and security findings. Variables such as \${ZAPROXY_HOME} refer to installation paths, not build uniqueness, while \${Profile_ID} and \${zap_scan} are not standard Jenkins variables for uniquely identifying builds. Using \${BUILD_ID} supports better auditing, historical analysis, and correlation between detected vulnerabilities and the exact build in which they were found, which is critical during the Build and Test stage of a DevSecOps pipeline.

Valid 312-97 Dumps shared by Actual4test.com for Helping Passing 312-97 Exam! Actual4test.com now offer the **newest 312-97 exam dumps**, the Actual4test.com 312-97 exam **questions have been updated** and **answers have been corrected** get the **newest** Actual4test.com 312-97 dumps with Test Engine here:

https://www.actual4test.com/312-97_examcollection.html (**150 Q&As Dumps, 30%OFF**

Special Discount: Freepdfdumps)

NEW QUESTION: 17

Farah Haddad, a QA-turned-security engineer at a Beirut telecom company, wants to test a running staging instance of a customer portal for vulnerabilities such as reflected XSS and broken authentication, from the perspective of an external attacker with no access to source code. Which testing approach fits her requirement?

- A. SAST
- B. SCA
- C. DAST
- D. Secret scanning

Answer: C (LEAVE A REPLY)

DAST tools operate as black-box testers against a running application, simulating external attacker behavior by sending crafted HTTP requests and observing responses to detect vulnerabilities such as reflected cross-site scripting, broken authentication, and injection flaws -- exactly matching Farah's scenario of testing a live staging instance without source code access.

SAST requires access to source code or binaries and analyzes them statically, which contradicts Farah's black-box, no-source-access constraint. SCA specifically targets known vulnerabilities in third-party and open-source dependencies rather than application logic flaws like XSS. Secret scanning searches repositories or file systems for exposed

credentials, unrelated to runtime attacker-perspective testing. Since Farah needs external, black-box testing of a live application, DAST is the correct answer.

NEW QUESTION: 18

Bruno Nascimento, a DevSecOps engineer at a Sao Paulo bank, wants to ensure that microservices communicating within his Kubernetes cluster mutually authenticate each other and encrypt all traffic between them, without modifying application code. Which technology should Bruno deploy?

- A. Service mesh with mutual TLS (e.g., Istio)
- B. WAF at the ingress
- C. VPN between developer laptops
- D. Static application firewall rules

Answer: [\(SHOW ANSWER\)](#)

A service mesh such as Istio or Linkerd injects sidecar proxies alongside each microservice to transparently handle mutual TLS (mTLS) authentication and encryption between services, providing zero-trust, code-free security for service-to-service communication -- exactly what Bruno needs. A WAF at the ingress protects against attacks arriving from outside the cluster targeting the application layer (e.g., HTTP-based attacks), but it does not provide mutual authentication and encryption between internal microservices. A VPN between developer laptops secures remote access for individual engineers, unrelated to intra-cluster service communication.

Static application firewall rules are generally coarse network-layer controls that do not provide identity-based mutual authentication or automatic encryption without code changes. Since Bruno needs code-transparent mTLS between microservices, a service mesh is correct.

NEW QUESTION: 19

(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?)

- 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 [\(LEAVE A REPLY\)](#)

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.

NEW QUESTION: 20

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?

- 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 ([LEAVE A REPLY](#))

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.

NEW QUESTION: 21

Henrik Larsson, a DevSecOps engineer at a Gothenburg automotive manufacturer, wants his CI pipeline to fail the build if any Dockerfile violates best practices, such as running as root or using the "latest" tag for a base image. Which type of tool should Henrik integrate?

- A.** Dockerfile linter (e.g., Hadolint)
- B.** Load balancer health check
- C.** SIEM correlation rule
- D.** Kubernetes NetworkPolicy

Answer: ([SHOW ANSWER](#))

A Dockerfile linter such as Hadolint statically analyzes Dockerfile syntax and instructions against established best practices, flagging issues like running containers as the root user, using mutable

"latest" image tags, or including unnecessary packages, and can be configured to fail CI builds when violations are found -- exactly matching Henrik's requirement. A load balancer health check monitors the availability of running application instances and has nothing to do with Dockerfile content analysis. A SIEM correlation rule analyzes security event data from running systems, not static Dockerfile definitions. A Kubernetes NetworkPolicy controls pod-to-pod network traffic at runtime and does not evaluate Dockerfile build instructions. Because Henrik needs static analysis of Dockerfile best practices integrated into CI, a Dockerfile linter is correct.

NEW QUESTION: 22

Brett Ryan has been working as a senior DevSecOps engineer in an IT company in Charleston, South Carolina. He is using git-multimail tool to send email notification for every push to git repository. By default, the tool will send one output email providing details about the reference change and one output email for every new commit due to a reference change. How can Brett ensure that git-multimail is set up appropriately?

- A.** Running the environmental variable GITHUB_MULTIMAIL_CHECK_SETUP by setting it to non- empty string.
- B.** Running the environmental variable GIT_MULTIMAIL_CHECK_SETUP by setting it to empty string.
- C.** Running the environmental variable GIT_MULTIMAIL_CHECK_SETUP by setting it to non-empty string.
- D.** Running the environmental variable GITHUB_MULTIMAIL_CHECK_SETUP by setting it to empty string.

Answer: C (LEAVE A REPLY)

The git-multimail tool provides a mechanism to verify whether it has been installed and configured correctly before being relied upon for production notifications. This verification is done using an environment variable named GIT_MULTIMAIL_CHECK_SETUP. When this variable is set to a non-empty string, git-multimail performs a setup validation and outputs diagnostic information to confirm that configuration values, hooks, and parameters are correctly defined. This helps prevent silent failures where commits occur but email notifications are not sent. Options that reference GITHUB_MULTIMAIL_CHECK_SETUP are incorrect because git-multimail is not limited to GitHub and does not use that variable name. Additionally, setting the variable to an empty string does not trigger the setup check. Ensuring proper configuration during the Code stage is important because it supports auditability, traceability, and timely communication among development and security teams. Therefore, Brett must run the environment variable GIT_MULTIMAIL_CHECK_SETUP with a non-empty value to ensure the tool is set up appropriately.

NEW QUESTION: 23

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?

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

Answer: D (LEAVE A REPLY)

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.

NEW QUESTION: 24

Victor Garber is a DevSecOps team leader in SanSec Pvt. Ltd. His organization develops various types of software products and web applications. Currently, his team is working on security of Java-based web application product. How can Victor identify vulnerabilities that are missed in pre-production testing activities?

- A. By performing deploy-time checks.
- B. By performing test-time checks.
- C. By performing commit-time checks.
- D. By performing build-time checks.

Answer: A (LEAVE A REPLY)

Deploy-time checks are designed to identify vulnerabilities that may not surface during earlier stages such as commit-time, build-time, or test-time checks. These checks analyze applications in environments that closely resemble or are part of production, making it possible to detect configuration issues, runtime vulnerabilities, and environment-specific weaknesses. Pre- production testing often cannot fully replicate production conditions, so

deploy-time checks act as an additional safety net. Commit-time and build-time checks focus on code quality and static analysis, while test-time checks validate application behavior in controlled environments. Deploy-time checks therefore help Victor uncover vulnerabilities missed earlier, improving overall security assurance before or during deployment.

NEW QUESTION: 25

Peter Dinklage has been working as a senior DevSecOps engineer at SacramentoSoft Solution Pvt. Ltd. He has deployed applications in docker containers. His team leader asked him to check the exposure of unnecessary ports. Which of the following commands should Peter use to check all the containers and the exposed ports?

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

Answer: (SHOW ANSWER)

To inspect exposed ports for running Docker containers, the recommended approach is to first retrieve container IDs using `docker ps --quiet` and then pass them to `docker inspect`. The `--format` option allows selective output of container configuration details, including port mappings. The command `docker ps --quiet | xargs docker inspect --format ': Ports='` correctly extracts port information for each container. Options that include the `--all` flag or incorrect formatting are not valid for this inspection use case. Checking exposed ports is an important activity in the Operate and Monitor stage because unnecessary open ports increase the attack surface and may violate container security best practices. Regular inspection helps ensure that only required ports are exposed, supporting secure runtime operations.

NEW QUESTION: 26

Trevor Noah has been working as a DevSecOps engineer in an IT company located in Detroit, Michigan. His team leader asked him to perform continuous threat modeling using ThreatSpec.

To do so, Trevor installed and initialized ThreatSpec in the source code repository; he then started annotating the source code with security issues, actions, or concept. Trevor ran ThreatSpec against the application code and he wants to generate the threat model report. Which of the following command Trevor should use to generate the threat model report using ThreatSpec?

- A. `$ ThreatSpec report.`
- B. `$ ThreatSpec Report.`
- C. `$ Threatspec Report.`
- D. `$ threatspec report.`

Answer: D (LEAVE A REPLY)

ThreatSpec is a command-line tool that follows standard Unix-style conventions, where commands are lowercase. To generate a threat model report after annotating source code, the correct command is `threatspec report`. Commands using incorrect casing or capitalization will fail because the CLI is case-sensitive. Options A, B, and C incorrectly capitalize either the command or the subcommand. Generating threat model reports during the Plan stage allows DevSecOps teams to continuously identify, document, and visualize security threats as the code evolves. This practice embeds threat modeling directly into the development lifecycle, enabling early risk identification and more secure system design decisions.

NEW QUESTION: 27

William O'Neil has been working as a senior DevSecOps engineer in an IT company that develops software products related to ecommerce. At this point in time, his team is working on securing a python-based application. Using GitGraber, William would like to detect sensitive information in real-time in his organizational GitHub repository. Therefore, he downloaded GitGraber and installed the dependencies. Which of the following commands should William use to find secrets using a keyword (assume the keyword is yahoo)?

- A. `python3 gitGraber.py -w wordlist/keywordsfile.txt -q "\yahoo" -s.`
- B. `python3 gitGraber.py -g wordlist/keywordsfile.txt -q "\yahoo" -s.`
- C. `python3 gitGraber.py -p wordlist/keywordsfile.txt -q "\yahoo" -s.`
- D. `python3 gitGraber.py -k wordlist/keywordsfile.txt -q "\yahoo" -s.`

Answer: D (LEAVE A REPLY)

NEW QUESTION: 28

James Harden has been working as a senior DevSecOps engineer in an IT company located in Oakland, California. To detect vulnerabilities and to evaluate attack vectors compromising web applications, he would like to integrate Burp Suite with Jenkins. He downloaded the Burp Suite Jenkins plugins and then uploaded the plugin and successfully integrated Burp Suite with Jenkins. After integration, he would like to scan web application using Burp Suite; therefore, he navigated to Jenkins' dashboard, opened an existing project, and clicked on Configure. Then, he navigated to the Build tab and selected Execute shell from Add build step. Which of the following commands should James enter under the Execute shell?

- A. `sudo BURP_SCAN_URL = http://target-website.com.`
- B. `grep BURP_SCAN_URL = http://target-website.com.`
- C. `cat BURP_SCAN_URL = http://target-website.com.`
- D. `echo BURP_SCAN_URL = http://target-website.com.`

Answer: D (LEAVE A REPLY)

When configuring Burp Suite scans in Jenkins using an Execute shell build step, environment variables are often set or echoed so that subsequent scan steps can consume them. The echo command is used to output or define values in the shell context.

In this case, `echo BURP_SCAN_URL = http://target-website.com` correctly defines the target URL for Burp Suite scanning. Commands like `grep` and `cat` are used for searching or displaying file contents and are not appropriate for setting scan parameters. The `sudo` command is unnecessary and incorrect in this context. Using the correct shell command ensures that Burp Suite receives the proper target information during the Build and Test stage, enabling accurate dynamic application security testing.

NEW QUESTION: 29

George Lennon is working as at InfoWorld Pvt. Solution as a DevSecOps engineer. His colleague, Sarah Mitchell, is a senior software developer. George told her to participate in a bug bounty program conducted by AWS for python and Java code developers. He informed Sarah that the challenge is a fun-based solution for bashing bugs, encouraging team building, and bringing friendly competition to enhance the quality of the code and application performance.

Acting on George's advice, Sarah participated in the bug bounty program and scored the highest points in the challenge, and she received a reward of \$10,000. Based on the given information, which of the following bug bounty programs did Sarah participate?

- A. AWS BugFixer.
- B. AWS BugFinder.
- C. AWS BugHunt.
- D. AWS BugBust.

Answer: (SHOW ANSWER)

The description matches AWS BugBust, which AWS positions as a gamified, team-based bug fixing challenge rather than a classic external "bug bounty" for finding vulnerabilities in AWS itself.

The key hints are "fun-based solution for bashing bugs," "encouraging team building," and "friendly competition," along with scoring points and awarding prizes. BugBust focuses on improving code quality by motivating developers to find and fix issues (often via static analysis findings) in languages like Java and Python. Participants earn points for remediations and compete on leaderboards, which aligns directly with Sarah "scored the highest points" and received a cash reward. The other names (BugFixer, BugFinder, BugHunt) are plausible- sounding but do not match the commonly referenced AWS gamified program described. In a DevSecOps context, this type of program supports culture by incentivizing secure coding habits, encouraging shared ownership of quality, and making remediation visible and rewarding across the engineering team.

NEW QUESTION: 30

Aiko Tanaka, a cloud security engineer at an Osaka retail company, is defining infrastructure using Terraform and wants to scan the Terraform configuration files for insecure settings - such as publicly exposed storage buckets or overly permissive security

groups - before the infrastructure is ever provisioned. Which practice is Aiko implementing?

- A. Infrastructure as Code (IaC) security scanning
- B. Dynamic Application Security Testing
- C. Penetration testing of live cloud resources
- D. Log aggregation

Answer: A (LEAVE A REPLY)

IaC security scanning tools (such as Checkov, tfsec, or Terrascan) statically analyze Infrastructure as Code definitions like Terraform, CloudFormation, or ARM templates to identify misconfigurations -- such as publicly accessible storage buckets or overly permissive security group rules -- before any actual cloud resources are provisioned, matching Aiko's exact goal of pre-provisioning detection. DAST tests running applications, not declarative infrastructure definitions, and requires a live target. Penetration testing of live cloud resources occurs after resources already exist and are deployed, which contradicts Aiko's "before provisioned" requirement. Log aggregation collects and centralizes log data from running systems for analysis, unrelated to pre-deployment configuration scanning. Since Aiko wants to catch misconfigurations in Terraform files prior to provisioning, IaC security scanning is correct.

NEW QUESTION: 31

Jason Barry has been working as a DevSecOps engineer in an IT company that develops software products and applications for ecommerce companies. During the build-time check, Jason discovered SQL injection and XSS security issues in the application code. What action does the build-time check perform on the application code?

- A. It will ignore the security issue and continue the build process.
- B. It will send a message to issue and project management tool and continue with deployment check.
- C. It will send an alert to SIEM and continue with test-time check.
- D. It will stop the build process.

Answer: D (LEAVE A REPLY)

Build-time checks are designed to enforce security gates within the CI/CD pipeline. When critical vulnerabilities such as SQL injection and cross-site scripting (XSS) are detected during this stage, the correct and expected behavior is to fail the build. Stopping the build process prevents insecure code from progressing to later stages such as testing, deployment, or production.

Ignoring issues or merely sending alerts while continuing the pipeline undermines the purpose of shift-left security. Alerts to SIEM systems and issue trackers are typically supplementary actions, but the primary enforcement mechanism at build time is to block the pipeline when severity thresholds are exceeded. This approach reduces remediation costs, limits exposure, and ensures that only secure artifacts move forward in the DevSecOps lifecycle.

Valid 312-97 Dumps shared by Actual4test.com for Helping Passing 312-97 Exam! Actual4test.com now offer the **newest 312-97 exam dumps**, the Actual4test.com 312-97 exam **questions have been updated** and **answers have been corrected** get the **newest** Actual4test.com 312-97 dumps with Test Engine here:
https://www.actual4test.com/312-97_examcollection.html (**150** Q&As Dumps, **30%OFF** Special Discount: **Freepdfdumps**)

NEW QUESTION: 32

Ana Beatriz Silva, a DevSecOps engineer at a Lisbon travel-booking company, wants to prevent developers from accidentally committing AWS access keys or API tokens into the Git repository, catching such secrets before they are even pushed to the remote. Which control should Ana implement?

- A. Post-deployment log review
- B. Pre-commit hook with secret scanning
- C. WAF rule update
- D. Load testing

Answer: (SHOW ANSWER)

A pre-commit hook integrated with a secret-scanning tool (such as git-secrets, TruffleHog, or Gitleaks) runs locally on the developer's machine before a commit is finalized, scanning staged changes for patterns matching credentials like AWS access keys or API tokens, and blocking the commit if secrets are detected -- this stops leaks at the earliest possible point, before code is even pushed, exactly matching Ana's goal. Post-deployment log review happens far too late in the pipeline, after code has already been committed, pushed, built, and possibly deployed. A WAF rule update addresses runtime web application attack traffic, unrelated to preventing secret leakage in source control. Load testing evaluates application performance under load and has no bearing on credential leakage prevention. Since Ana needs to catch secrets before they're pushed to the remote repository, a pre-commit hook with secret scanning is correct.

NEW QUESTION: 33

Jordon Garrett has recently joined a startup IT company located in Chicago, Illinois, as a DevSecOps engineer. His team leader asked him to find a SAST tool that can secure the organization Azure environment. Which of the following is a SAST tool that Jordon can select to secure his organization's Azure environment?

- A. Coverity.
- B. Accurics.
- C. Tenable.io.
- D. DevSkim.

Answer: A (LEAVE A REPLY)

Coverity is a well-known Static Application Security Testing (SAST) tool used to analyze source code for security vulnerabilities, coding errors, and quality issues. It integrates with CI/CD pipelines and supports enterprise-scale environments, including cloud-based development on platforms such as Azure. Accurics focuses on Infrastructure as Code security, Tenable.io is a vulnerability management platform for infrastructure and assets, and DevSkim is a lightweight code scanning extension rather than a full SAST platform. Selecting Coverity enables deep static analysis of application code during the Code stage, helping teams detect vulnerabilities early and reduce remediation costs.

NEW QUESTION: 34

(Amy Ryan is a DevSecOps engineer in an IT company that develops software products and web applications related to cyber security. She is using Anchore tool for container vulnerability scanning and Software Bill of Materials (SBOM) generation. It helped her to perform quick scanning and generating a list of known vulnerabilities from an SBOM, container image, or project directory. Which of the following commands should Amy run to include software from all the image layers in the SBOM?.)

- A. `syft packages < image > scope all_layers SBOM.`
- B. `syft packages < image > --scope all-layers Anchore.`
- C. `syft packages < image > scope all_layers.`
- D. `syft packages < image > --scope all-layers.`

Answer: (SHOW ANSWER)

Syft is used by Anchore to generate Software Bill of Materials (SBOMs) from container images and directories. By default, Syft may only analyze the squashed image view. Using the `--scope all-layers` flag instructs Syft to include software components from all image layers, ensuring comprehensive visibility into dependencies introduced at every stage of image creation. The other options use invalid syntax or unsupported flags. Including all layers during SBOM generation improves vulnerability detection accuracy and supports compliance requirements, making it a critical practice during the Build and Test stage.

NEW QUESTION: 35

Sara Lindqvist, a DevSecOps engineer at a Helsinki healthtech company, must ensure that patient data used in the staging environment for testing does not expose real patient identities, while still preserving realistic data formats and referential integrity for QA testing. Which technique should Sara apply?

- A. Full production data replication with no changes
- B. Data masking / anonymization of sensitive fields
- C. Disabling the staging environment entirely
- D. Encrypting the entire database with a single shared key visible to all developers

Answer: B (LEAVE A REPLY)

Data masking (or anonymization/pseudonymization) techniques replace or obscure sensitive fields such as names, national ID numbers, or medical record identifiers with realistic but non-identifying substitute values, while preserving data format, structure, and referential integrity needed for effective QA testing -- exactly satisfying Sara's dual requirement of privacy protection and testing realism. Fully replicating production data with no changes would expose real patient identities in a lower-security environment, violating privacy regulations like HIPAA or GDPR.

Disabling the staging environment entirely would prevent QA testing altogether, which is not a viable solution to the data privacy problem. Encrypting the entire database with a single shared key visible to all developers does not actually anonymize the data -- anyone with that key could still view real patient identities, and broad key visibility itself creates a serious exposure risk.

Since Sara needs privacy-preserving yet realistic test data, data masking/anonymization is correct.

NEW QUESTION: 36

Andrew Gerrard has recently joined an IT company that develops software products and applications as a DevSecOps engineer. His team leader asked him to download a jar application from the organization GitHub repository and run the BDD security framework. Andrew successfully downloaded the jar application from the repository and executed the jar application; then, he cloned the BDD security framework. Which of the following commands should Andrew use to execute the authentication feature?

- A. `./gradlev -Dcucumber.options="--tags @authentication -tags @skip"`.
- B. `./gradlew -Dcucumber.options="--tags @authentication -tags @skip"`.
- C. `./gradlev -Dcucumber.options="--tags @authentication -tags ~@skip"`.
- D. `./gradlew -Dcucumber.options="--tags @authentication -tags ~@skip"`.

Answer: (SHOW ANSWER)

The BDD Security framework is executed through Gradle wrapper commands, and the correct wrapper script on Unix-like systems is `./gradlew` (dot-slash indicates "run the wrapper from the current directory"). Options using `./gradlew` or `./gradlev` imply an absolute path at filesystem root and are typically incorrect for a cloned project. Also, the wrapper name is `gradlew`, not `gradlev`.

For executing only the authentication feature (or scenarios tagged for authentication), Cucumber tag expressions are used through the `-Dcucumber.options` system property. The command must include `--tags @authentication` to select authentication-tagged scenarios. To skip scenarios tagged "skip," the exclusion operator is used as `--tags ~@skip` (meaning "exclude @skip").

Options A and B incorrectly include `--tags @skip` which would include skipped tests rather than exclude them. Therefore, `./gradlew -Dcucumber.options="--tags @authentication --tags ~@skip"` is the correct choice to run authentication scenarios while excluding anything marked to skip.

NEW QUESTION: 37

Chidinma Eze, a DevSecOps engineer at a Lagos e-commerce company, needs to define measurable targets - such as "95% of critical vulnerabilities remediated within 7 days" - that her security and engineering teams are jointly accountable for meeting. What are these targets called?

- A. Service Level Agreements only
- B. Security Service Level Objectives (Security SLOs)
- C. Sprint velocity metrics
- D. Network uptime guarantees

Answer: [\(SHOW ANSWER\)](#)

Security Service Level Objectives (Security SLOs) are specific, measurable security-related targets -- such as vulnerability remediation timeframes based on severity -- that define shared accountability between security and engineering teams, embedding security expectations into the operational culture of DevSecOps, which matches exactly what Chidinma is defining. Service Level Agreements (SLAs) are typically formal, often externally facing contractual commitments (often with penalties) between a provider and a customer, which is broader and less specifically tied to internal security remediation accountability than an SLO. Sprint velocity metrics measure a team's rate of completing story points during sprints and have nothing to do with vulnerability remediation timelines. Network uptime guarantees pertain to availability commitments, not vulnerability management targets. Since Chidinma is defining internal, measurable vulnerability-remediation targets for shared team accountability, Security SLOs is the correct answer.

NEW QUESTION: 38

Grace Odhiambo, a DevSecOps engineer at a Nairobi-based agritech firm, needs to scan her organization's Java-based application source code for insecure coding patterns such as SQL injection and hardcoded credentials without executing the application. Which category of testing should she use?

- A. Dynamic Application Security Testing (DAST)
- B. Interactive Application Security Testing (IAST)
- C. Static Application Security Testing (SAST)
- D. Runtime Application Self-Protection (RASP)

Answer: [C \(LEAVE A REPLY\)](#)

SAST analyzes source code, bytecode, or binaries "white-box" style without executing the application, making it ideal for detecting insecure coding patterns like SQL injection-prone queries or hardcoded secrets directly in the codebase during the Code stage. DAST, by contrast, requires a running application and tests it externally as a black box, so it cannot be used purely on source code. IAST combines elements of SAST and DAST by instrumenting a running application to observe code execution paths in real time, which also requires execution. RASP is not a testing methodology at all but a runtime defense

mechanism embedded in the application to detect and block attacks as they occur in production. Since Grace wants to scan source code without running the application, SAST is correct.

NEW QUESTION: 39

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?

- 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 (LEAVE A REPLY)

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.

NEW QUESTION: 40

Richard Branson has been working as a DevSecOps engineer in an IT company that develops apps for Android mobiles. To manage the secret information of an application in various phases of development lifecycle and to provide fine-grained access to each secret, he would like to integrate HashiCorp Vault with Jenkins. To access the vault from Jenkins, Richard installed `hashicorp-vault-` plugin and ran a vault instance; he then selected the `AppRole` authentication method, which allows apps to access vault with a predefined role. Which of the following commands should Richard use to enable `AppRole` authentication?

- A. `enable vault auth approle`.
- B. `auth vault enable approle`.
- C. `vault auth enable approle`.
- D. `enable auth vault approle`.

Answer: C (LEAVE A REPLY)

HashiCorp Vault enables authentication mechanisms using the `vault auth enable` command followed by the name of the authentication method. To enable AppRole authentication, the correct command is `vault auth enable approle`. AppRole is specifically designed for machine-to-machine authentication, making it ideal for CI/CD tools like Jenkins. It allows applications to authenticate securely using role IDs and secret IDs instead of static credentials. The other options do not follow Vault CLI syntax and would result in command errors. Enabling AppRole during the Build and Test stage ensures that secrets are accessed securely and dynamically, supporting least-privilege access control and reducing the risk of credential leakage across the DevSecOps pipeline.

NEW QUESTION: 41

Kevin Williamson is working as a DevSecOps engineer in an IT company located in Los Angeles, California. His team has integrated Jira with Jenkins to view every issue on Jira, including the status of the latest build or successful deployment of the work to an environment. Which of the following can Kevin use to search issues on Jira?

- A. Java query language.
- B. Structured query language.
- C. Atlassian query language.
- D. Jira query language.

Answer: C (LEAVE A REPLY)

Jira uses Atlassian Query Language, commonly referred to as JQL, to search, filter, and manage issues. This query language allows users to create advanced searches using fields such as project, status, assignee, priority, and custom attributes. Although often informally called Jira Query Language, the official name among the given options is Atlassian Query Language. SQL and Java query language are unrelated and not used for issue searching in Jira. Using JQL during the Code stage improves traceability between source code commits, builds, and tracked issues, enabling teams to monitor progress, validate deployment status, and maintain alignment between development and delivery activities.

NEW QUESTION: 42

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?

- 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 (LEAVE A REPLY)

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.

NEW QUESTION: 43

Cindy Williams has recently joined an IT company as a DevSecOps engineer. She configured Bundle-Audit in Travis CI. Cindy detected vulnerability in Gemfile dependencies and resolved it by adding some line of codes. How does Bundler scan Gemfile.lock for insecure versions of gems?

A. By taking the information from the travis.yml file and comparing it with the known vulnerabilities.

B. By taking the information from the Gemfile and comparing it with the unknown vulnerabilities.

C. By taking the information from the Gemfile and comparing it with the known vulnerabilities.

D. By taking the information from the travis.yml and comparing it with the unknown vulnerabilities.

Answer: C (LEAVE A REPLY)

Bundler-Audit is a Software Composition Analysis (SCA) tool designed specifically for Ruby applications. It scans the Gemfile and Gemfile.lock to identify all declared dependencies and their resolved versions. The Gemfile specifies which gems the application depends on, while the Gemfile.lock ensures consistent dependency versions across environments. Bundler-Audit compares this dependency information against a database of known vulnerabilities to identify insecure or outdated gems. It does not rely on the Travis CI configuration file for vulnerability detection, nor does it compare against unknown vulnerabilities. Integrating Bundler-Audit into the Build and Test stage ensures that vulnerable third-party libraries are detected early, allowing developers to remediate issues before the application progresses further in the pipeline. This practice supports shift-left security and reduces the risk of introducing known vulnerabilities into production systems.

NEW QUESTION: 44

Peter McCarthy is working in TetraVerse Soft Solution Pvt. Ltd. as a DevSecOps engineer. His organization develops customized software products and web applications. To develop software products quickly and securely, his organization has been using AWS cloud-based services, including AWS DevOps services. Peter would like to use CloudMapper to examine the AWS cloud environment and perform auditing for security issues. Which of the following privileges should Peter possess in order to collect information about the AWS account?

A. `arn:aws:iam::aws:policy/SecurityAudit` `arn:aws:iam::aws:policy/job-function/ViewOnlyAccess`.

B. `arn:aws:iam::aws:policy/SecurityCheck` `arn:aws:iam::aws:policy/job-function/ViewOnlyAccess::EditOnlyAccess`.

C. `arn:aws:iam::aws:policy/SecurityAudit::SecurityCheck` `arn:aws:iam::aws:policy/job-role/ViewOnlyAccess::EditOnlyAccess`.

D. `arn:aws:iam::aws:policy/AWSLambdaFullAccess` `arn:aws:iam::aws:policy/job-role/ViewOnlyAccess`.

Answer: (SHOW ANSWER)

CloudMapper requires read-only access to AWS resources in order to collect metadata, visualize architectures, and perform security analysis without modifying infrastructure. The AWS-managed policy `SecurityAudit` provides permissions to view security-related configuration across services, while `ViewOnlyAccess` allows read-only access to AWS resources more broadly. Together, these policies enable CloudMapper to gather comprehensive information about the AWS environment without granting write privileges. The other options either reference invalid policy names, incorrect formatting, or excessive permissions such as `AWSLambdaFullAccess`, which are unnecessary and violate least-privilege principles. Granting `SecurityAudit` and `ViewOnlyAccess` aligns with secure auditing practices during the Operate and Monitor stage.

NEW QUESTION: 45

Amara Nwosu, a security operations engineer at a Lagos telecom, wants to aggregate logs from applications, containers, and cloud infrastructure into a centralized platform where she can correlate events and detect anomalous patterns indicating a potential breach. Which category of tool should Amara deploy?

A. SIEM (Security Information and Event Management)

B. SAST scanner

C. Secrets vault

D. Load testing tool

Answer: A (LEAVE A REPLY)

A SIEM platform (such as Splunk, IBM QRadar, or Microsoft Sentinel) centralizes log and event data from diverse sources -- applications, containers, network devices, and cloud infrastructure -- and applies correlation rules, analytics, and often machine learning to

detect anomalous patterns that may indicate a security incident, matching exactly what Amara needs for the Monitor stage. A SAST scanner analyzes static source code for vulnerabilities and has no log correlation or anomaly detection capability. A secrets vault securely stores and manages credentials, unrelated to log aggregation or event correlation. A load testing tool measures application performance under simulated user load and does not perform security event correlation. Since Amara needs centralized log correlation and anomaly detection across multiple sources, a SIEM is correct.

NEW QUESTION: 46

Dustin Hoffman has been working as a DevSecOps engineer in an IT company located in San Diego, California. 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/Checkmarx.gitlab-ci.yml``?

- 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 (LEAVE A REPLY)

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.

Valid 312-97 Dumps shared by Actual4test.com for Helping Passing 312-97 Exam! Actual4test.com now offer the **newest 312-97 exam dumps**, the Actual4test.com 312-97 exam **questions have been updated** and **answers have been corrected** get the **newest** Actual4test.com 312-97 dumps with Test Engine here:
https://www.actual4test.com/312-97_examcollection.html (**150 Q&As Dumps, 30%OFF**
Special Discount: Freepdfdumps)

NEW QUESTION: 47

(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?.)

- 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 (LEAVE A REPLY)

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.

NEW QUESTION: 48

Sarah Wright has recently joined a multinational company as a DevSecOps engineer. She has created a container and deployed a web application in it. Sarah would like to stop this container.

Which of the following commands stop the running container created by Sarah Wright?

- A. `[root@574bac18f89d /]# kill.`
- B. `[root@574bac18f89d /]# exit.`
- C. `[root@574bac18f89d /]# clear.`
- D. `[root@574bac18f89d /]# stop.`

Answer: (SHOW ANSWER)

When working inside an interactive Docker container session, the container continues running as long as its primary foreground process is active. Executing the exit command terminates the shell session, which in turn stops the container if no other foreground processes are running. The kill command requires a process identifier and is not used in this context, while clear simply clears the terminal screen and does not affect container execution. The stop command is not a valid shell command inside a container. Properly stopping containers during the Operate and Monitor stage helps free system resources, prevent unintended service exposure, and maintain a clean runtime environment. This

practice aligns with container lifecycle management best practices and reduces operational risk.

NEW QUESTION: 49

Joe Adler has recently been offered a job as a DevSecOps engineer in an IT company that develops software products and web applications for the healthcare industry. He would like to implement DevSec Hardening Framework to add a layer into the automation framework that configures operating systems and services and takes care of difficult settings, compliance guidelines, cryptography recommendations, and secure defaults. To apply DevSec Hardening Framework to the machine, he scanned the machine using Nessus scanning tool; he then checked the compliance results before using DevSec Hardening Framework. Which of the following commands should Joe use to run DevSec Hardening Framework?

- A. Chef-solo -c solo.rb -j solo.json.
- B. Chef-solo -m solo.rb -h solo.json.
- C. Chef-solo -j solo.rb -c solo.json.
- D. Chef-solo -h solo.rb -m solo.json.

Answer: A (LEAVE A REPLY)

The DevSec Hardening Framework is commonly implemented using Chef, and it is executed locally using the chef-solo command. The -c flag specifies the configuration file (solo.rb), and the

-j flag specifies the JSON attributes file (solo.json). Option A correctly uses both required parameters in the proper format. The other options incorrectly swap or misuse flags that are not supported by Chef- solo. Running this command applies secure configurations, compliance controls, and cryptographic standards to the target system. Executing DevSec Hardening Framework during the Operate and Monitor stage ensures that systems remain secure, compliant, and resilient against misconfiguration- based attacks.

NEW QUESTION: 50

Erica Mena has been working as a DevSecOps engineer in an IT company that provides customize software solutions to various clients across United States. To protect serverless and container applications with RASP, she would like to create an Azure container instance using Azure CLI in Microsoft PowerShell. She created the Azure container instance and loaded the container image to it. She then reviewed the deployment of the container instance. Which of the following commands should Erica run to get the logging information from the Azure container instance? (Assume the resource group name as ACI and container name as aci-test-closh.)

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

Answer: D (LEAVE A REPLY)

Azure Container Instances provide built-in logging capabilities that can be accessed using the Azure CLI. To retrieve logs from a deployed container instance, the correct command is `az container logs` followed by the resource group and container name. The proper syntax requires double-dash parameters: `--resource-group` and `--name`. In Erica's case, the correct command is `az container logs -- resource-group ACI --name aci-test-closh`. Options that use "az get container logs" are invalid because "get" is not a supported verb in this context. Option C uses incorrect single-dash flags, which do not match Azure CLI standards. Accessing container logs during the Code stage helps engineers validate application behavior, identify runtime errors, and ensure that security instrumentation such as RASP agents are functioning correctly before progressing further in the pipeline.

NEW QUESTION: 51

(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 of the following command should Gabriel use to clone the BDD security framework?)

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

Answer: B (LEAVE A REPLY)

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.

NEW QUESTION: 52

Brady Coleman is a senior DevSecOps engineer at CloudVac Security Private Ltd. He has created a new container named "eccbrad" from the `centos:7` image using the command `docker run -i -t --name geeklab centos:7 /bin/bash`. Now, Brady wants to install the `httpd` package inside the `eccbrad` container. Which of the following commands should Brady use to install the `httpd` package inside the container?

- A. `sudo install-httpd`.
- B. `sudo install httpd`.
- C. `yum install-httpd`.

D. yum install httpd.

Answer: D (LEAVE A REPLY)

The CentOS operating system uses the yum package manager to install software packages.

Inside a CentOS-based Docker container, the correct way to install the Apache HTTP Server is by running yum install httpd. Containers typically run as the root user by default, making the use of sudo unnecessary. Commands such as install-httpd are invalid because yum requires the install keyword followed by the package name separated by a space. Installing required packages inside containers should be done carefully to avoid bloating images and increasing the attack surface. During the Operate and Monitor stage, DevSecOps teams must balance functionality with container hardening best practices by installing only necessary components.

NEW QUESTION: 53

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?

- 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 (LEAVE A REPLY)

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.

NEW QUESTION: 54

Timothy Dalton has been working as a senior DevSecOps engineer in an IT company located in Auburn, New York. He would like to use Jenkins for CI and Azure Pipelines for CD to deploy a Java-based app to an Azure Container Service (AKS) Kubernetes cluster. Before deploying Azure Kubernetes Service (AKS) Cluster, Timothy wants to create a Resource group named Jenkins in southindia location. Which of the following commands should Timothy run?

- A. `az group create --name Jenkins --location southindia.`
- B. `az grp create --n Jenkins --loc southindia.`
- C. `azure group create --name Jenkins --location southindia.`
- D. `azure group create --n Jenkins --loc southindia.`

Answer: [\(SHOW ANSWER\)](#)

Azure resource groups are created using the Azure CLI command `az group create`. The `--name` parameter specifies the resource group name, and `--location` defines the Azure region. Option A uses the correct CLI prefix (`az`), command group (`group create`), and valid parameters. Options B, C, and D are incorrect due to invalid command abbreviations or incorrect CLI prefixes (`azure` instead of `az`). Creating a resource group is a foundational step in the Release and Deploy stage, as it provides a logical container for AKS clusters, networking components, and related resources, enabling organized, secure, and manageable deployments.

NEW QUESTION: 55

Jason Wylie has been working as a DevSecOps engineer in an IT company located in Sacramento, California. 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?

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

Answer: [C \(LEAVE A REPLY\)](#)

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 `kubectrl get pods` command. The command `kubectrl 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.

NEW QUESTION: 56

Frances Fisher joined TerraWolt Pvt. Ltd. as a DevSecOps engineer in 2020. On February 1, 2022, his organization became a victim of cyber security attack. The attacker targeted the network and application vulnerabilities and compromised some important functionality of the application. To secure the organization against similar types of attacks, Franches used a flexible, accurate, low maintenance vulnerability management and assessment solution that continuously scans the network and application vulnerabilities and provides daily updates and specialized testing methodologies to catch maximum detectable vulnerabilities. Based on the above- mentioned information, which of the following tools is Frances using?

- A.** SonarQube.
- B.** Black Duck.
- C.** BeSECURE.
- D.** Shadow Daemon.

Answer: C (LEAVE A REPLY)

BeSECURE is a vulnerability management and assessment solution designed for continuous scanning of both network and application vulnerabilities. It emphasizes flexibility, accuracy, low maintenance overhead, and frequent updates to vulnerability detection mechanisms. These characteristics align directly with the scenario described, where the organization requires continuous scanning, daily updates, and specialized testing methodologies to detect a wide range of vulnerabilities. SonarQube focuses on static code quality and security analysis during development, Black Duck is primarily used for open-source software composition analysis, and Shadow Daemon is a web application firewall rather than a comprehensive vulnerability management solution. Using BeSECURE during the Operate and Monitor stage allows organizations to maintain ongoing visibility into their security posture, detect new vulnerabilities as they emerge, and reduce the likelihood of repeat attacks by addressing weaknesses proactively.

NEW QUESTION: 57

Lisa Kramer carries an experience of 4 years as a DevSecOps engineer in an IT company. The software development team of her organization has developed a Ruby on Rails web application and would like to find vulnerabilities in Ruby dependencies. Therefore, the team leader of the software development team approached Lisa for help in this regard. Which of the following SCA tool should Lisa use to detect vulnerabilities in Ruby dependencies?

- A.** Bandit.

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

Answer: (SHOW ANSWER)

Bundler-Audit is an SCA tool designed specifically for Ruby applications. It analyzes the Gemfile and Gemfile.lock to identify dependencies and checks them against known vulnerability databases. Bandit is intended for Python code analysis, Retire.js targets JavaScript libraries, and Tenable.io focuses on infrastructure-level vulnerabilities. By using Bundler-Audit during the Code stage, DevSecOps teams can detect vulnerable Ruby gems early and ensure that only secure dependencies are used. This reduces the risk of exploiting known vulnerabilities in third-party libraries and supports secure dependency management throughout the development lifecycle.

NEW QUESTION: 58

William O'Neil has been working as a senior DevSecOps engineer in an IT company that develops software products related to ecommerce. At this point in time, his team is working on securing a python-based application. Using GitGraber, William would like to detect sensitive information in real-time in his organizational GitHub repository. Therefore, he downloaded GitGraber and installed the dependencies. Which of the following commands should William use to find secrets using a keyword (assume the keyword is yahoo)?

- A. `python3 gitGraber.py -w wordlist/keywordsfile.txt -q "\yahoo" -s.`
- B. `python3 gitGraber.py -g wordlist/keywordsfile.txt -q "\yahoo" -s.`
- C. `python3 gitGraber.py -p wordlist/keywordsfile.txt -q "\yahoo" -s.`
- D. `python3 gitGraber.py -k wordlist/keywordsfile.txt -q "\yahoo" -s.`

Answer: (SHOW ANSWER)

GitGraber uses specific command-line flags to define how secret detection is performed. The -k flag is used to specify a keyword file that contains search terms for identifying sensitive data in repositories. In this case, William wants to search for secrets using the keyword "yahoo," which is passed using the -q flag. Options -w, -g, and -p are not valid flags for keyword-based scanning in GitGraber. By using -k, GitGraber scans repositories for matches against the defined keywords and reports potential secret exposures in real time. This capability is especially valuable during the Code stage, helping teams prevent credential leakage and maintain secure repositories.

NEW QUESTION: 59

Priya Nair, a DevSecOps engineer at a Bengaluru SaaS company, configures her CI pipeline so that any commit introducing a critical-severity vulnerability in an open-source dependency automatically fails the build and blocks the merge. Which control is Priya implementing?

- A. Manual code review gate
- B. Automated Software Composition Analysis (SCA) gate

C. Canary release gate

D. Rollback trigger

Answer: B (LEAVE A REPLY)

An automated SCA gate integrates dependency-scanning tools directly into the CI pipeline so that builds are automatically blocked when scanned open-source components contain vulnerabilities above a defined severity threshold, enforcing a "fail fast" security policy without requiring human intervention. A manual code review gate depends on a human reviewer approving changes and does not automatically evaluate dependency vulnerability data. A canary release gate is a Deploy-stage technique that gradually routes a small percentage of production traffic to a new version to observe behavior before full rollout, unrelated to dependency scanning. A rollback trigger is an Operate-stage mechanism that reverts a deployment after failure is detected in production. Since Priya's pipeline automatically fails builds based on dependency vulnerability severity, an automated SCA gate is correct.

NEW QUESTION: 60

Craig Kelly has been working as a software development team leader in an IT company over the past 8 years. His team is working on the development of an Android application product. Sandra Oliver, a DevSecOps engineer, used DAST tools and fuzz testing to perform advanced checks on the Android application product and detected critical and high severity issues. She provided the information about the security issues and the recommendations to mitigate them to Craig's team.

Which type of security checks performed by Sandra involve detection of critical and high severity issues using DAST tools and fuzz testing?

A. Commit-time checks.

B. Build-time checks.

C. Deploy-time checks.

D. Test-time checks.

Answer: D (LEAVE A REPLY)

Dynamic Application Security Testing (DAST) and fuzz testing require a running application in order to actively probe for vulnerabilities such as injection flaws, authentication bypasses, and improper input handling. These techniques are therefore performed after the application has been built and deployed to a testing environment, categorizing them as test-time checks. Commit-time and build-time checks rely primarily on static analysis and dependency scanning and do not exercise application behavior at runtime. Deploy-time checks focus on configuration validation rather than aggressive attack simulation. Test-time checks are specifically designed to uncover critical and high-severity vulnerabilities by mimicking real-world attack scenarios. Performing DAST and fuzz testing during this stage allows teams to detect exploitable flaws before production release, significantly strengthening application security.

NEW QUESTION: 61

Tomasz Wieczorek is a DevSecOps lead at a logistics company in Krakow. He wants his team to define security requirements collaboratively with product owners at the very start of each sprint, ensuring acceptance criteria include abuse cases alongside normal use cases. Which practice is Tomasz implementing?

- A. Security Champions Program
- B. Secure Sprint Planning
- C. Chaos Engineering
- D. Blue-Green Deployment

Answer: (SHOW ANSWER)

Secure sprint planning integrates security requirements, misuse/abuse cases, and risk-based acceptance criteria directly into sprint backlogs during the Plan stage, ensuring security is "shifted left" and treated as a first-class requirement rather than an afterthought. A Security Champions Program instead designates specific engineers within teams to advocate for security practices and knowledge-sharing, which is a cultural/organizational structure rather than a sprint-level planning activity. Chaos engineering is an Operate-stage resilience testing practice, and blue- green deployment is a Release/Deploy strategy for minimizing downtime. Because Tomasz is embedding abuse cases into sprint acceptance criteria at the beginning of each sprint, secure sprint planning is the correct answer.

Valid 312-97 Dumps shared by Actual4test.com for Helping Passing 312-97 Exam! Actual4test.com now offer the **newest 312-97 exam dumps**, the Actual4test.com 312-97 exam **questions have been updated** and **answers have been corrected** get the **newest** Actual4test.com 312-97 dumps with Test Engine here:
https://www.actual4test.com/312-97_examcollection.html (**150 Q&As Dumps, 30%OFF**
Special Discount: Freepdfdumps)

NEW QUESTION: 62

BVR Pvt. Ltd. is an IT company that develops software products and applications related to IoT devices. The software development team of the organization is using Bitbucket repository to plan projects, collaborate on code, test, and deploy. The repository provides teams a single place for projects planning and collaboration on coding, testing, and deploying the software application.

Which of the following is offered by Bitbucket to BVR Pvt. Ltd.?

- A. Free limited public repositories.
- B. Free unlimited private repositories.
- C. Free limited private repositories.
- D. Free unlimited public repositories.

Answer: B (LEAVE A REPLY)

Bitbucket provides a cloud-based source code management platform that supports collaboration, CI/CD integration, and secure code hosting. One of the key features offered by Bitbucket is free unlimited private repositories, particularly beneficial for organizations developing proprietary software such as IoT applications. This allows teams to store source code securely without exposing it publicly while still enabling collaboration features like pull requests, issue tracking, and pipeline automation. The term "limited private repositories" is inaccurate because Bitbucket does not restrict the number of private repositories under its free offering; rather, user count limits apply. While Bitbucket also supports public repositories, the option that best represents its value to enterprise and product-based teams is unlimited private repositories. This capability aligns with DevSecOps practices by ensuring confidentiality of source code while enabling integrated planning, testing, and deployment workflows within a single platform.

NEW QUESTION: 63

Charlotte Flair is a DevSecOps engineer at Egma Soft Solution Pvt. Ltd. Her organization develops software and applications related to supply chain management. Charlotte would like to integrate Sscreen RASP tool with Slack to monitor the application at runtime for malicious activities and block them before they can damage the application. Therefore, she created a Sscreen account and installed Sscreen Microagent. Now, she would like to install the PHP microagent. To do so, she reviewed the PHP microagent's compatibility, then she signed in to Sscreen account and noted the token in Notepad. Which of the following commands should Charlotte run in the terminal to install the PHP extension and the Sscreen daemon?

A. `curl -s https://download.sscreen.com/php/install.sh > sscreen-install.sh \ \&& bash sscreen-install.sh`

`[CHARLOTTE'S ORG TOKEN HERE] "[ CHARLOTTE'S APP NAME HERE]"`.

B. `curl -s https://download.sscreen.com/php/install.sh < sscreen-install.sh \ \&& bash sscreen-install.sh`

`[CHARLOTTE'S ORG TOKEN HERE] "[ CHARLOTTE'S APP NAME HERE]"`.

C. `curl -i https://download.sscreen.com/php/install.sh > sscreen-install.sh \ \&& bash sscreen-install.sh`

`[CHARLOTTE'S ORG TOKEN HERE] "[ CHARLOTTE'S APP NAME HERE]"`.

D. `curl -i https://download.sscreen.com/php/install.sh < sscreen-install.sh \ \&& bash sscreen-install.sh`

`[CHARLOTTE'S ORG TOKEN HERE] "[ CHARLOTTE'S APP NAME HERE]"`.

Answer: A (LEAVE A REPLY)

The correct installation procedure for the Sscreen PHP microagent involves downloading the installer script and executing it with the organization token and application name. The `curl -s` option downloads the script silently, while the `>` redirection operator saves it locally as `sscreen-install.sh`. The script is then executed using `bash`, passing the required token

and app name as parameters. Options using input redirection (<) are incorrect because they do not save the downloaded script to a file. The -i option includes HTTP headers in the output, which is unnecessary and could corrupt the script. Installing the microagent correctly enables runtime monitoring, attack detection, and automatic blocking, supporting strong runtime security during the Operate and Monitor stage.

NEW QUESTION: 64

Scott Adkins has recently joined an IT company located in New Orleans, Louisiana, as a DevSecOps engineer. He would like to build docker infrastructure using Terraform; therefore, he has created a directory named terraform-docker-container. He then changed into the directory using the command: `cd terraform-docker-container`. Now, Scott wants to create a file to define the infrastructure. Which of the following commands should Scott use to create a file to define the infrastructure?

- A. `cat main.tf`.
- B. `echo main.tf`.
- C. `touch main.tf`.
- D. `sudo main.tf`.

Answer: C (LEAVE A REPLY)

Terraform infrastructure definitions are written in files with the .tf extension, commonly named main.tf. To create a new, empty file where infrastructure code can be added, the correct command is `touch main.tf`. This command creates the file without adding any content, allowing Scott to begin defining Docker infrastructure using Terraform syntax. The `cat` command is used to display file contents, not create files. The `echo` command prints text to standard output and does not create files unless output redirection is used. The command `sudo main.tf` is invalid and does not create files. Creating Terraform configuration files during the Release and Deploy stage supports Infrastructure as Code practices, enabling version control, repeatability, and security validation of infrastructure deployments. This approach allows DevSecOps teams to define, review, and deploy infrastructure in a consistent and auditable manner.

NEW QUESTION: 65

GainInsights is an IT company that develops mobile applications software. On February 11, 2022, the organization became a victim of a cyber-attack. The attacker targeted the organization's application and compromised some important functionality. After the incident, the DevSecOps team of GainInsights identified the cause of the security issue, resolved it, and noted it for future reference. Based on this information, which of the following set of tests was conducted by GainInsights?

- A. Blameless post-mortem.
- B. Security acceptance tests.
- C. Security smoke tests.
- D. White box testing.

Answer: A ([LEAVE A REPLY](#))

A blameless post-mortem is conducted after a security incident to analyze what happened, why it happened, and how similar incidents can be prevented in the future--without assigning individual blame. The key indicators in the scenario are that the team identified the cause, resolved the issue, and documented lessons learned for future reference. Security acceptance tests and smoke tests are pre-release validation activities, while white-box testing focuses on code-level analysis rather than incident review. Blameless post-mortems are a cornerstone of DevSecOps culture, encouraging transparency, continuous learning, and systemic improvement during the Operate and Monitor stage.

Valid 312-97 Dumps shared by Actual4test.com for Helping Passing 312-97 Exam!

Actual4test.com now offer the **newest 312-97 exam dumps**, the Actual4test.com 312-97 exam **questions have been updated** and **answers have been corrected** get the **newest** Actual4test.com 312-97 dumps with Test Engine here:

https://www.actual4test.com/312-97_examcollection.html (**150** Q&As Dumps, **30%OFF**

Special Discount: [Freepdfdumps](#))