Palo Alto Networks XSIAM-Engineer関連合格問題 & XSIAM-Engineerテスト対策書

P.S. ShikenPASSがGoogle Driveで共有している無料かつ新しいXSIAM-Engineerダンプ：https://drive.google.com/open?id=1ys0qhS8tjnrJlSmmk7OwXdq26TIC8RlY

なぜ我々はあなたが購入した前にやってみることを許しますか。なぜ我々はあなたが利用してからPalo Alto NetworksのXSIAM-Engineer試験に失敗したら、全額で返金するのを承諾しますか。我々は弊社の商品があなたに試験に合格させるのを信じでいます。Palo Alto NetworksのXSIAM-Engineer試験が更新するとともに我々の作成するソフトは更新しています。

Palo Alto Networks XSIAM-Engineer 認定試験の出題範囲：

トピック	出題範囲
トピック 1	• 計画とインストール：このセクションでは、XSIAMエンジニアのスキルを評価し、Palo Alto Networks Cortex XSIAMコンポーネントの計画、評価、インストールについて学習します。既存のITインフラストラクチャの評価、ハードウェア、ソフトウェア、および統合に関する導入要件の定義、そしてXSIAMアーキテクチャの通信ニーズの確立に重点を置いています。受験者は、エージェント、ブローカーVM、エンジンの設定に加え、ユーザーロール、権限、アクセス制御の管理も行う必要があります。

トピック 2	• コンテンツ最適化：この試験セクションでは、検知エンジニアのスキルを評価し、XSIAMコンテンツと検知ロジックの改良に焦点を当てます。正規化のための解析およびデータモデリングルールの導入、相関関係、IOC、BIOC、攻撃対象領域管理に基づく検知ルールの管理、インシデントおよびアラートレイアウトの最適化などが含まれます。受験者は、運用の可視性を高めるためのカスタムダッシュボードとレポートテンプレートの作成能力も証明する必要があります。
トピック 3	• メンテナンスとトラブルシューティング：このセクションでは、セキュリティ運用エンジニアのスキルを評価し、XSIAMコンポーネントの導入後のメンテナンスとトラブルシューティングを網羅します。例外設定の管理、XDRエージェントやBroker VMなどのソフトウェアコンポーネントの更新、データの取り込み、正規化、解析に関する問題の診断などが含まれます。受験者は、運用の信頼性を確保するために、統合、自動化プレイブック、システムパフォーマンスのトラブルシューティングも実施する必要があります。
トピック 4	• 統合と自動化：この試験セクションでは、SIEMエンジニアのスキルを評価し、XSIAMにおけるデータのオンボーディングと自動化の設定に焦点を当てます。エンドポイント、ネットワーク、クラウド、IDなどの多様なデータソースの統合、メッセージング、認証、脅威インテリジェンスなどの自動化フィードの設定、マーケットプレイスコンテンツパックの実装などを網羅します。また、効率的なワークフロー自動化のためのプレイブックの計画、作成、カスタマイズ、デバッグ能力も評価されます。

試験XSIAM-Engineer関連合格問題 & 便利なXSIAM-Engineerテスト対策書 | 大人気XSIAM-Engineer受験資格

あなたはこのような人々の一人ですか。さまざまな資料とトレーニング授業を前にして、どれを選ぶか本当に困っているのです。もしそうだったら、これ以上困ることはないです。ShikenPASSはあなたにとって最も正確な選択ですから。我々はあなたに試験問題と解答に含まれている全面的な試験資料を提供することができます。ShikenPASSの解答は最も正確な解釈ですから、あなたがより良い知識を身につけることに助けになれます。ShikenPASSを利用したら、Palo Alto NetworksのXSIAM-Engineer認定試験に受かることを信じています。それも我々が全てのお客様に対する約束です。

Palo Alto Networks XSIAM Engineer 認定 XSIAM-Engineer 試験問題 (Q77-Q82):

質問 # 77
An XSIAM engineer is tasked with optimizing a correlation rule that triggers on 'Multiple Failed Login Attempts followed by Successful Login from a New Device'. This rule is generating an excessive number of alerts for legitimate user password resets. Which of the following modifications to the XSIAM correlation rule logic would best optimize its performance and accuracy while minimizing false positives for this specific scenario?

- A. Modify the rule to only trigger if the successful login originates from an IP address categorized as 'malicious' by external threat intelligence feeds.
- B. Change the aggregation function for failed login attempts from 'count' to 'sum' and introduce a 'group by' clause for 'application_name'.
- C. Create a separate suppression rule that silences alerts for 24 hours if a user performs a password reset via the organization's self-service portal.
- D. Increase the number of required failed login attempts to 20 within a 5-minute window and decrease the time window for successful login to 30 seconds.
- E. Add an exclusion filter to the rule that ignores events where the 'device_id' for the successful login is present in a 'known_user_devices' lookup list and the 'user_agent' matches common browser strings.

正解：E

解説：
Option B directly addresses the false positive scenario of legitimate password resets. By excluding known devices or common

browser agents, the rule can distinguish between a user resetting their password on a new but legitimate device (like a new laptop) and an attacker. Option A might reduce some false positives but could also miss legitimate attacks. Option C is too restrictive and would miss insider threats or attacks from previously unknown IPs. Option D is irrelevant to the problem. Option E is a reactive suppression, not a proactive optimization of the detection logic itself.

質問 # 78
Consider an XSIAM environment where the XDR Collectors are deployed as Docker containers orchestrated by Kubernetes. A new XDR Collector image version (2.5.0) has been released, and you need to perform a controlled update across your cluster. Your current deployment uses a Helm chart. Which of the following kubectl commands, when used in conjunction with a modified Helm chart value for the image tag, would facilitate a rolling update with zero downtime, assuming the Helm chart is correctly configured for rolling updates?

- A. `kubectl rollout restart deployment/xdr-collector-deployment`
- B. `kubectl apply -f collectors-deployment.yaml --record`
- C. `helm upgrade xdr-collectors ./xdr-collector-chart --set image.tag=2.5.0`
- D. `kubectl scale deployment/xdr-collector-deployment --replicas=0 && kubectl scale deployment/xdr-collector-deployment --replicas=3`
- E. `kubectl delete pod -l app=xdr-collector && kubectl apply -f collectors-deployment.yaml`

正解：C

解説：
When managing applications deployed via Helm, the standard and most effective way to perform updates, especially rolling updates, is through 'helm upgrade'. By modifying the 'image.tag' value in the Helm chart (either directly in 'values.yamr or via '-set' as shown), Helm will intelligently detect the change and perform a rolling update on the Kubernetes deployment, ensuring zero downtime if the deployment strategy is set to 'RollingUpdate' (which is the default for most Helm charts). Options A, B, D, and E are either less appropriate for a Helm-managed application, cause downtime, or are not the primary mechanism for an image update through Helm.

質問 # 79
A security engineer is tasked with integrating a custom-built internal application's security audit logs into XSIAM. The application generates JSON formatted logs directly to a dedicated S3 bucket in AWS. The logs contain critical information like user actions, access attempts, and configuration changes. The requirement is to ingest these logs efficiently and ensure they are properly parsed for XSIAM's analytics and correlation engines, while minimizing custom development within XSIAM. Which XSIAM integration approach is most suitable?

- A. Configure an AWS S3 trigger to invoke an AWS Lambda function that pushes the JSON logs to an XSIAM Broker via syslog, then create a custom parser in XSIAM.
- B. Configure the S3 bucket to directly send notifications to an SNS topic, which then triggers an HTTPS endpoint on an XSIAM Data Broker to ingest the raw JSON.
- C. Use an XSIAM Playbook to periodically query the S3 bucket via the AWS S3 API, then parse the JSON within the playbook and push the data using the XSIAM Event Ingest API.
- D. Set up an XSIAM Data Collector on an EC2 instance within the AWS VPC, which pulls logs from the S3 bucket using the AWS SDK, then forwards them to XSIAM's Data Lake. XSIAM's auto-parsing for JSON can be leveraged, or a minimal custom parser defined if needed.
- E. Manually download the JSON logs from S3 daily and upload them to XSIAM's Data Lake via the XSIAM UI for batch processing.

正解：D

解説：
Setting up an XSIAM Data Collector (Broker) within the AWS VPC to pull logs directly from S3 is an efficient and scalable approach. XSIAM Brokers are designed for data collection from various sources, including cloud storage. XSIAM has strong capabilities for parsing JSON, often requiring only minimal configuration or custom parsing. This avoids the complexity of Lambda functions for simple ingestion and provides a robust, resilient ingestion pipeline. Using playbooks for direct ingestion might be less efficient for high volumes of raw log data compared to a dedicated data collector.

質問 # 80
An XSIAM engineer needs to create a custom content pack that includes a new integration for a proprietary internal vulnerability scanner. This integration will define several commands, one of which is `get_scan_results`, which accepts a `scan_id` and returns a JSON object containing scan findings. Another command, `trigger_scan`, initiates a scan and returns a `scan_id`. Which of the following components are absolutely essential for defining and making these commands usable within XSIAM playbooks, and what consideration is crucial for `get_scan_results`?

- An Integration YAML file, a Python script implementing the commands, and a Mapper for `trigger_scan` output.
  Crucial consideration for `get_scan_results`: Ensure the output adheres to schema is strictly adhered to for XSIAM's UI rendering.
- An Integration YAML file, a Python script implementing the commands, and a Parser for `get_scan_results`.
  Crucial consideration for `get_scan_results`: Implement polling logic within the command if the vulnerability scanner's API is asynchronous.
- An Automation Rule, a Playbook that calls the commands, and a Dashboard Widget to display results.
  Crucial consideration for `get_scan_results`: Optimize API calls to prevent rate limiting on the scanner.
- A Data Connector for continuous ingestion of scan results, and Correlation Rules to identify vulnerabilities.
  Crucial consideration for `get_scan_results`: Define specific data types for all returned fields in the XSIAM schema.
- Only a Python script with the commands is sufficient; XSIAM automatically detects and registers them.
  Crucial consideration for `get_scan_results`: Manage pagination if the scan results are large.

- A. Option C
- B. Option B
- C. Option A
- D. Option D
- E. Option E

正解：B

解説：
To define custom integrations and their commands in XSIAM, you absolutely need an Integration YAML file (which describes the integration, its parameters, and the commands it supports) and a Python script that implements the actual logic for each command. A Parser is essential for `get_scan_results` to transform the raw JSON output from the vulnerability scanner into structured XSIAM data (e.g., incidents, artifacts, or indicators) that can be easily consumed by playbooks, search, and the UI. Crucially, for `get_scan_results`, if `trigger_scan` is asynchronous (which is common for long-running scans), the `get_scan_results` command's implementation in the Python script must often include polling logic . This means it repeatedly queries the scanner's API for the status of the scan using the `scan_id` until the results are ready, or a timeout is reached. This is a common design pattern for integrating with asynchronous external systems. Options A, C, D, E miss these fundamental requirements or considerations.

質問 # 81
An organization is migrating its cloud infrastructure from AWS to Azure, while simultaneously planning for XSIAM adoption. They heavily utilize serverless functions (AWS Lambda, Azure Functions) and containerized applications (EKS, AKS). What challenges might arise in collecting comprehensive telemetry from these ephemeral and dynamic cloud-native components, and how does XSIAM address these?

- A. Challenge: Inability to deploy traditional network-based sensors. XSIAM addresses this by performing agentless network scanning of the cloud environment.
- B. Challenge: Dynamic scaling and short lifespans make consistent monitoring difficult. XSIAM addresses this by integrating directly with cloud provider APIs (e.g., CloudWatch, Azure Monitor, Activity Logs) and leveraging specialized collectors for container runtime security (e.g., Cortex XDR for Containers).
- C. Challenge: Lack of persistent file systems for log storage. XSIAM addresses this by automatically deploying dedicated persistent storage volumes for each serverless function and container.
- D. Challenge: Ephemeral nature makes traditional agent deployment difficult. XSIAM addresses this by requiring agents to be baked into container images and serverless runtimes.
- E. Challenge: Increased network egress costs due to telemetry forwarding. XSIAM addresses this by compressing all telemetry data by 95% before ingestion.

正解：B

解説：
Ephemeral and dynamic cloud-native components (serverless, containers) present significant challenges for traditional monitoring. Their short lifespans and frequent scaling make persistent agent deployment or manual log configuration impractical. XSIAM tackles this by leveraging direct API integrations with cloud providers' native logging and monitoring services (e.g., AWS CloudWatch,

Azure Monitor, Azure Activity Logs) and specialized collectors for container environments (Cortex XDR for Containers). This allows XSIAM to ingest logs, metrics, and runtime activity from these dynamic workloads without requiring a persistent agent on every ephemeral instance.

質問 # 82

.....

当社ShikenPASSのすべての専門家および教授の唯一の目標は、すべての人々に最適で適切なXSIAM-Engineer学習教材を設計することです。多くの顧客のさまざまな要求に応じて、彼らはすべての顧客向けに3つの異なるバージョンのXSIAM-Engineer認定試験ガイド資料を設計しました：PDF、ソフト、およびAPPバージョン。弊社のXSIAM-Engineer試験問題を使用するすべての人がXSIAM-Engineer試験に合格し、関連する認定資格を取得できることを心から願っています。そして、XSIAM-Engineer試験問題の合格率は98％以上です。

XSIAM-Engineerテスト対策書: https://www.shikenpass.com/XSIAM-Engineer-shiken.html

- XSIAM-Engineer試験勉強過去問 □ XSIAM-Engineer問題サンプル □ XSIAM-Engineer過去問無料 □ □ www.goshiken.com □に移動し、[XSIAM-Engineer]を検索して、無料でダウンロード可能な試験資料を探しますXSIAM-Engineer試験合格攻略
- XSIAM-Engineer関連資料 □ XSIAM-Engineer模擬モード □ XSIAM-Engineerクラムメディア □ [www.goshiken.com]は、□ XSIAM-Engineer □を無料でダウンロードするのに最適なサイトですXSIAM-Engineer専門知識訓練
- XSIAM-Engineer試験の準備方法｜実際的なXSIAM-Engineer関連合格問題試験｜正確的なPalo Alto Networks XSIAM Engineerテスト対策書 □ ➠ www.passtest.jp □は、▶ XSIAM-Engineer ◀を無料でダウンロードするのに最適なサイトですXSIAM-Engineer関連資料
- XSIAM-Engineer資格試験 □ XSIAM-Engineer過去問無料 □ XSIAM-Engineer日本語問題集 □ □ www.goshiken.com □から簡単に□ XSIAM-Engineer □を無料でダウンロードできますXSIAM-Engineer試験合格攻略
- 権威のあるXSIAM-Engineer関連合格問題一回合格-実用的なXSIAM-Engineerテスト対策書 □ [www.passtest.jp]サイトにて最新▷ XSIAM-Engineer ◁問題集をダウンロードXSIAM-Engineerクラムメディア
- 一番優秀なXSIAM-Engineer関連合格問題試験-試験の準備方法-素晴らしいXSIAM-Engineerテスト対策書 □ 《 XSIAM-Engineer 》の試験問題は □ www.goshiken.com □で無料配信中XSIAM-Engineer試験問題集
- XSIAM-Engineer専門知識訓練 □ XSIAM-Engineerトレーリングサンプル □ XSIAM-Engineer勉強の資料 □ □ www.mogiexam.com □に移動し、□ XSIAM-Engineer □を検索して、無料でダウンロード可能な試験資料を探しますXSIAM-Engineer日本語問題集
- XSIAM-Engineer専門知識訓練 □ XSIAM-Engineer試験合格攻略 □ XSIAM-Engineer資格試験 □ □ www.goshiken.com □サイトにて最新「 XSIAM-Engineer 」問題集をダウンロードXSIAM-Engineer試験問題集
- XSIAM-Engineer試験の準備方法｜実際的なXSIAM-Engineer関連合格問題試験｜正確的なPalo Alto Networks XSIAM Engineerテスト対策書 □ ▶ www.xhs1991.com ◀で使える無料オンライン版▷ XSIAM-Engineer ◁の試験問題XSIAM-Engineer問題サンプル
- XSIAM-Engineer試験の準備方法｜実際的なXSIAM-Engineer関連合格問題試験｜正確的なPalo Alto Networks XSIAM Engineerテスト対策書 □ ウェブサイト[www.goshiken.com]を開き、《 XSIAM-Engineer 》を検索して無料でダウンロードしてくださいXSIAM-Engineer模擬モード
- XSIAM-Engineer問題サンプル □ XSIAM-Engineer専門知識訓練 □ XSIAM-Engineer認証試験 □ URL 《 www.passtest.jp 》をコピーして開き、□ XSIAM-Engineer □を検索して無料でダウンロードしてくださいXSIAM-Engineer試験問題集
- d2.ilc.edu.tw, www.stes.tyc.edu.tw, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, myportal.utt.edu.tt, aaa.yyiii.com, www.kickstarter.com, www.sg588.tw, 39.98.44.44, www.stes.tyc.edu.tw, forum2.isky.hk, tutorial.mentork.in, Disposable vapes

ちなみに、ShikenPASS XSIAM-Engineerの一部をクラウドストレージからダウンロードできます：https://drive.google.com/open?id=1ys0qhS8tjnrJlSmmk7OwXdq26TIC8RlY