# PaloAltoNetworks.SecOps-Pro.v2026-05-14.q105

| | |
|---|---|
| 試験コード: | SecOps-Pro |
| 試験名称: | Palo Alto Networks Security Operations Professional |
| 認定資格: | Palo Alto Networks |
| 無料問題数: | 105 |
| バージョン: | v2026-05-14 |
| アクセス数: | 137 |
| ページビュー数: | 1050 |
| https://www.jpnpdf.com/PaloAltoNetworks.SecOps-Pro.v2026-05-14.q105-mondaishu.html | |

最新問題: **1**

セキュリティアナリストは、反射型DLLインジェクションを利用し、その後C2通信用の名前付きパイプを作成することが知られている特定のマルウェアファミリの脅威ハンティングを行っています。アナリストは、このハンティングにCortex XDRのログスティッチングを活用したいと考えています。必要なデータソースが取り込まれていると仮定した場合、このような複雑なイベントチェーンを特定するために、基盤となるスティッチングされたログデータを最も効果的に活用できるAQL（XDRクエリ言語）クエリはどれでしょうか？

**A.**

```
dataset = xdr_data | filter event_type = PROCESS_CREATION and process_name = 'rundll32.exe' and command_line contains 'malicious.dll'
```

**B.**

```
dataset = xdr_data | filter event_type = NETWORK_CONNECTION and destination_port = 80 and action = 'allow'
```

**C.**

```
dataset = xdr_data | filter event_type = PROCESS_CREATION and action_process_image_path contains 'rundll32.exe' and command_line contains '.dll' | join process_instance_id as parent_process_id with dataset = xdr_data | filter event_type = NAMED_PIPE_CREATION and action_process_image_path = previous.action_process_image_path | dedup agent_id | limit 100
```

**D.**

```
dataset = xdr_data | filter event_type = FILE_WRITE and file_path contains 'temp' and file_extension = 'exe'
```

**E.**

```
dataset = xdr_data | filter event_type = LOGIN and user_name = 'admin' and country = 'Russia'
```

**Answer: C** (メーッセージを残す)

説明: この質問では、AQL と、複雑な動作パターンのためにステッチ データを活用する方法の理解が必要です。リフレクション DLL インジェクションでは、多くの場合、rund1132.exe または類似のプロセスが、ディスク上に存在しない DLL をロードしますが、これは単純なシグネチャでは検出が困難です。後続の名前付きパイプの作成は、CZ のプロセス間通信を意味します。オプション A は範囲が広すぎて、DLL インジェクションと名前付きパイプを結び付けていません。オプション B と E は一般的すぎて、説明されている攻撃に固有ではありません。オプション D は、攻撃の一部である可能性のあるファイル書き込みに焦点を当てていますが、リフレクション DLL インジェクションや名前付きパイプを捕捉していません。オプション C は、AQL を使用して次の操作を正しく実行しています。1. rund1132.exe および DLL が関係する PROCESS_CREATION イ

ベントをフィルター処理します。2. process_instance_id (ログ ステッチングによって維持される親子関係を表す) に基づく結合操作を使用して、同じプロセスまたは子孫から発生した後続の NAMED_PIPE_CREATION イベントを検索します。これにより、因果関係で結びついた 2 つの異なる動作 (DLL インジェクションの前駆動作と C2 の名前付きパイプ) が 1 つのクエリに効果的にまとめられ、脅威ハンティングにおけるログ ステッチングの実際的な応用が実証されます。

最新問題: **2**
組織は、高度な多段階攻撃に見舞われています。最初の侵害によって認証情報が窃取され、その後PowerShellを使用したラテラルムーブメントが行われます。攻撃者は、エンコードされたPowerShellコマンドを利用して、従来のシグネチャベースの検出を回避しています。Cortex XSIAMセキュリティ運用プロフェッショナルとして、疑わしいエンコードされたPowerShell実行を高い信頼度で識別し、誤検知を最小限に抑え、通常のアクティビティのベースラインを侵害した際にアラートをトリガーするカスタム検出ルールを作成する必要があります。XQL、ルールタイプ、集計ロジックのどの組み合わせが最も適切でしょうか？

**A.** ルールタイプ: 動作。XQL:
```
dataset = xdr_data | filter event_type = 'process' and action_process_image_name = 'powershell.exe' and command_line contains '-EncodedCommand' |
group count() as encoded_count by host_name, user_name | where encoded_count > 5 in 10m | limit 100
```
**B.** ルールタイプ: 相関。XQL:
```
dataset = xdr_data | filter event_type = 'process' and action_process_image_name = 'powershell.exe' and command_line contains '-EncodedCommand' |
eval decoded_cmd = base64_decode(substring(command_line, indexof(command_line, '-EncodedCommand') + strlen('-EncodedCommand') + 1)) |
where decoded_cmd contains ('Invoke-Mimikatz' or 'Invoke-CredentialStealing') | limit 100
```
**C.** ルールタイプ: 異常。XQL:
```
dataset = xdr_data | filter event_type = 'process' and action_process_image_name = 'powershell.exe' and command_line contains '-EncodedCommand' |
track by host_name, user_name | time series by 1h | detect anomalies with baseline(metric=count(), interval='1d', threshold=3)
```
**D.** ルールタイプ: 動作。XQL:
```
dataset = xdr_data | filter event_type = 'process' and action_process_image_name = 'powershell.exe' and command_line contains '-EncodedCommand' and
(parent_process_image_name not in ('explorer.exe', 'cmd.exe', 'powershell.exe') or parent_process_image_name = 'powershell.exe'
and parent_process_command_line contains '-NonInteractive') | limit 100
```
**E.** ルールタイプ: 異常。XQL:
```
dataset = xdr_data | filter event_type = 'process' and action_process_image_name = 'powershell.exe' and command_line contains '-EncodedCommand' |
eval is_suspicious_length = (strlen(command_line) > 500) | eval entropy_score = entropy(command_line) | track by host_name, user_name |
time_series by 1h | detect anomalies with baseline(metric=count(where is_suspicious_length and entropy_score > 0.7), interval='1d', threshold=2)
```
**Answer: (**解答を表示する**)**
オプションEは、高度なエンコードされたPowerShellを検出するための最も堅牢なソリューションを提供します。「異常」ルールタイプは、通常のアクティビティのベースライン設定と逸脱の検出に重要です。「EncodedCommand」オプションA、C）を単に検索するだけでは、正規ツールもこのルールを使用しているため、多くの誤検知が発生します。オプションBは強力なデコードを試みますが、特定の悪意のある文字列をハードコーディングすることはポリモーフィック攻撃に対して拡張性に欠け、また「異常」ルールではなく「相関」ルールです。オプションDは親プロセス分析を使用します。これは優れたフィルターですが、ベースライン設定は活用されません。オプションEは、「（長いエンコードされたコマンドは悪意のあるものであることが多い）」と「entropy_score」（高いエントロピーはエンコード/難読化を示す）を追加することで検出機能を強化します。これらの計算フィールドと、「host_name、user_name」ごとの疑わしいコマンドの数に基づく異常検知を組み合わせることで、通常の動作を学習することで誤検知を最小限に抑える、高精度で適応的なルールが実現します。これは、XSIAM の高度な脅威の探索と検出と一致しています。

最新問題: **3**
あるグローバル企業はCortex XSIAMを実装し、エンドポイントセンサー（XDRエージェント）、ネットワークファイアウォール、クラウドインフラストラクチャ（AWS CloudTrail、Azure Activity Logs）、アイデンティティプロバイダー（Okta）など、さまざまなソースからログを取り込み始めています。セキュリティチームは、基本的なイベント相関は機能しているものの、クラウドとオンプレミスのインタラクションを含むインシデントのステッチ精度が期待よりも低いことに気づきました。具体的には、XSIAMの「インシデントビュー」には、統一された攻撃ストーリーではなく、関連するアクティビティ（例：ユーザーがOktaにログインし、次にEC2インスタンスにアクセスし、次にオンプレミスサーバーで疑わしいファイルにアクセス）に対して個別のアラートが表示されることがよくあります。ステッチ精度の低下の原因として最も可能性が高いのは次のどれですか？また、XSIAM内でどのような設定調整を行うことで、この問題を解決できますか？
**A.** XSIAM データレイクが適切にシャーディングされていないため、クエリパフォーマンスが低下し、リアルタイムのステッチングが妨げられています。「データレイクのシャーディング設定」を調整してください。
**B.** データ保持ポリシーが不十分なため、ステッチングが行われる前に関連する履歴ログが消去されています。重要なデータソースの「ログ保持期間」を長くしてください。
**C.** クラウドとオンプレミスのログソース間で共通識別子（ユーザー名形式、セッションID、資産タグなど）が不一致または欠落しているため、ステッチングエンジンによる効果的な相関分析が妨げられています。異なるソースに対して「データ正規化ルール」と「属性マッピング」を実装してください。
**D.** XSIAM「行動分析エンジン」の調整が誤っているため、包括的な行動パターンではなく、個々のアラートしきい値に過度に依存する傾向があります。「UEBA異常検出プロファイル」を再調整してください。
**E.** XSIAM の「自動対応プレイブック」がインシデントを早期に終了することで、ステッチングプロセスを妨害しています。特定の「プレイブックトリガー」を無効にするか、変更してください。
**Answer: C (**メーッセージを残す**)**
これは、大規模なハイブリッド環境においてよくある課題です。ログスティッチングは、イベントをリンクするために共通属性の識別に大きく依存しています。ユーザーのIDがOktaでは「john.doe@company.com」、AWSでは「doe」、オンプレミスサーバーでは「JOHN_D」である場合、これらの異なる形式が単一の正規IDに正規化またはマッピングされていない限り、XSIAMのスティッチングエンジンはこれらのアクティビティを同じエンティティに属するものとして関連付けることができません。同様に、環境間で一貫したアセットタグや一意のセッションIDが不可欠です。したがって、最も可能性の高い根本原因は、共通識別子の不一致または欠落です。XSIAMのデータ取り込みパイプライン内に堅牢な「データ正規化ルール」（データの解析とフォーマットを一貫して行う）と「属性マッピング」（同一エンティティの異なる表現を正規形式にマッピングする）を実装することは、このようなクロス環境シナリオにおけるスティッチングの忠実度を向上させるための重要な構成調整です。

最新問題: **4**

あるグローバル金融機関が、高度な多段階攻撃に見舞われています。最初の偵察活動ではフィッシングが使用され、エンドポイントへの侵入に至りました。その後、攻撃者は正規の管理ツール（LOLBin）を使用してラテラルムーブメント（水平展開）を行い、機密データを盗み出しました。既存のEDRソリューションは、いくつかの疑わしいプロセスに対してアラートを発しましたが、これらの個別のイベントを一貫した攻撃の記録として関連付けるのが困難で、アラート疲れと対応の遅延につながっていました。スタンドアロンのEDRと比較して、このシナリオに最も効果的に対処できるCortex XDR機能は、次のうちどれですか？

**A.** 高度な行動分析と機械学習により、攻撃対象領域全体にわたって通常のユーザーおよびシステム行動からの逸脱を識別します。

**B.** 従来のウイルス対策と同様に、シグネチャベースの検出を通じて悪意のある実行可能ファイルをリアルタイムでブロックする機能。

**C.** ネットワーク内のすべてのエンドポイントに対する自動パッチ管理と脆弱性スキャン。

**D.** 集中ログ収集のみを目的としたセキュリティ情報およびイベント管理 (SIEM) システムとの統合。

**E.** ネットワーク境界でディープ パケット インスペクションを提供し、既知の悪意のある IP アドレスをブロックします。

**Answer: (**解答を表示する**)**

Cortex XDRは、行動分析と機械学習を用いて様々なソース（エンドポイント、ネットワーク、クラウド、アイデンティティ）からのアラートを相関させ、完全な攻撃ストーリー（インシデントビュー）を構築することに優れています。これにより、アラート疲れが大幅に軽減され、セキュリティチームは実際の脅威に集中できるようになります。これは、個別のアラートを提供することが多いEDRの大きな制約です。EDRは疑わしいプロセス（LOLBinなど）をフラグ付けすることはできますが、これらの低忠実度のアラートを高忠実度のインシデントに結び付けるためのクロスドメイン可視性とAI駆動型の相関分析機能は、Cortex XDRの拡張された検出および対応機能によって実現されます。

最新問題**: 5**

ある組織は、セキュリティオーケストレーション、自動化、そしてレスポンスのために、Palo Alto Networks Cortex XSOARに大きく依存しています。ランサムウェアが関与する大規模なインシデントにより、複数の部門にまたがる重要データが暗号化されました。駆除フェーズにおいて、インシデント対応チームは、ランサムウェアが残した永続化メカニズムを削除し、復号ツールを配布するためのカスタムスクリプトを展開する必要があります。このスクリプトは、影響を受けた数百のエンドポイントで実行する必要があります。このタスクに最も適した効率的なXSOARプレイブックコマンドまたは統合はどれでしょうか？

**A.**

```
!send-email to=soc@example.com subject='Ransomware Eradication Status' body='Decryption script executed on all systems.'
```

**B.**

```
!create-incident incidentType='Post-Ransomware Cleanup' name='Eradication Script Deployment'
```

**C.**

```
!demisto-api-call command='endpoint.execute_script' args='{ "script_id": "ransomware_cleanup", "target_systems": "all_affected" }'
```

**D.**

```
!exec-remote-command command='powershell.exe -file C:\temp\cleanup.ps1' on_endpoints='affected_group'
```

**E.** 影響を受ける各エンドポイントに手動でログインし、クリーンアップ スクリプトを実行します。

**Answer: D (**メーッセージを残す**)**

オプションDが最も適切かつ効率的です。XSOARは、多数のエンドポイントにわたるタスクの自動化に優れています。「exec-remote-command」(またはエンドポイント統合に応じて同様のエンドポイント管理統合コマンド)を使用すると、指定されたシステム上でスクリプトをリモート実行できます。これはまさに駆除に必要な機能です。オプションAは通信用です。オプションBはインシデント作成用であり、実行用ではありません。オプションCは汎用API呼び出しを示していますが、「endpoint.execute_script」を処理する特定の統合がないため、「exec-remote-command」ほど直接的ではありません。オプションEは、数百のエンドポイントを扱う場合、非常に非効率的で実用的ではありません。

最新問題**: 6**

広く利用されているWebアプリケーションにゼロデイ脆弱性が積極的に悪用されており、組織のインターネット接続サーバーに深刻な懸念が生じています。ベンダーからのパッチはまだ提供されていませんが、Palo Alto Networks NGFWが導入されています。NGFWの機能を活用した一時的な代替制御で、正規のトラフィックを妨害したりカスタムシグネチャを必要とせずに、このゼロデイ脆弱性に対する最適な即時保護を提供できるものはどれでしょうか？

**A.** 影響を受ける Web アプリケーション サーバーへのすべての受信 HTTP/HTTPS トラフィックをブロックします。

**B.** App-ID を使用して 「厳密な」アプリケーション レベルのセキュリティ ポリシーを有効にし、Web サーバーへの既知の正当なアプリケーション トラフィックのみを許可し、それ以外はブロックします。

**C.** 関連するセキュリティ ポリシーに適用された、ゼロデイ CVE に固有のシグネチャ (脅威インテリジェンスから入手可能な場合) を使用して、 「脆弱性保護」ルールを含むカスタム 「脅威防止」プロファイルを構成します。

**D.** Palo Alto Networks GlobalProtect を使用してホスト情報プロファイル (HIP) チェックを強制し、パッチを適用したクライアントのみが Web アプリケーションにアクセスできるようにします。

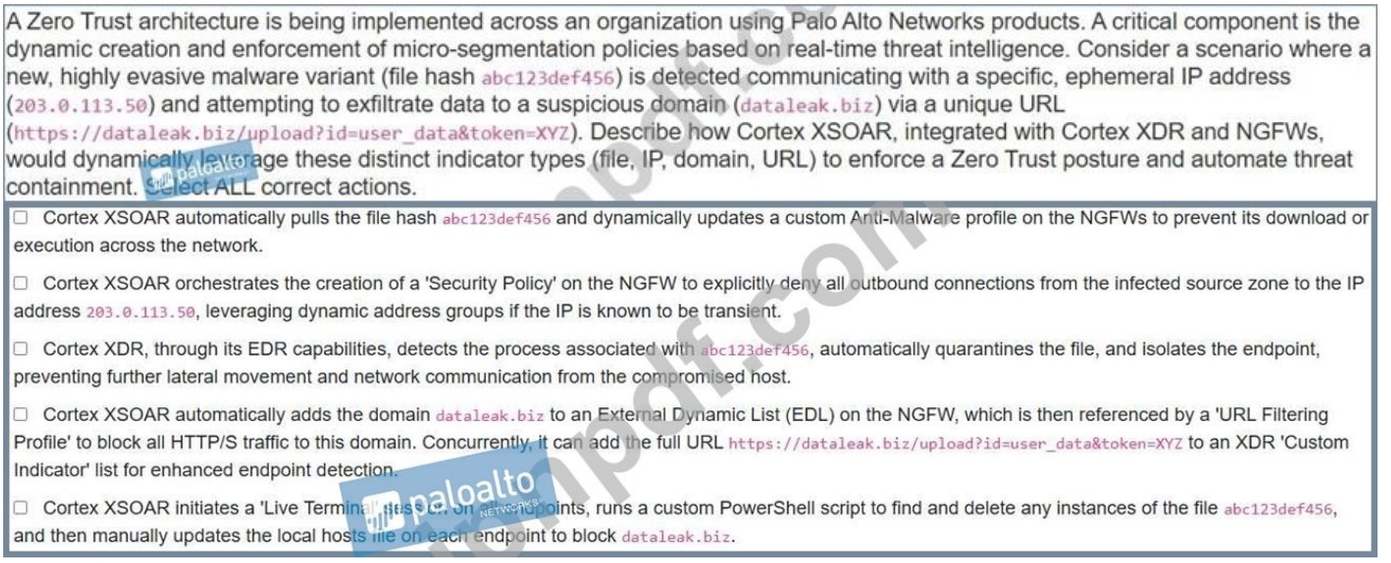
**E.** Web サーバーへの接続をレート制限するために、 「サービス拒否 (DoS) 保護」ポリシーを展開します。

**Answer: B (**メーッセージを残す**)**

課題は、利用可能なパッチや特定のシグネチャがないゼロデイ攻撃です。すべてのHTTP/HTTPS (A)をブロックすると、正当なトラフィックが妨害されます。カスタムシグネチャ(C)は理想的ですが、外部のインテリジェンスが迅速に提供しない限り、ゼロデイ攻撃には利用できません。GlobalProtect (D)はクライアントアクセス用であり、サーバー保護用ではありません。DoS防御 (E)はDoSを軽減するものであり、エクスプロイトを軽減するものではありません。

最も効果的な即時補償制御はApp-ID（B）です。正当なアプリケーショントラフィック（「Webブラウジング」や特定のサブアプリケーションなど）のみを厳密に定義して許可し、それ以外をブロックすることで、NGFWは、特定の脆弱性シグネチャがなくても、ゼロデイ攻撃に悪用される可能性のある悪意のあるコードや異常なプロトコルの実行を阻止できます。これは、「ポジティブセキュリティモデル」の適用にとって強力な機能です。

最新問題: 7


A Zero Trust architecture is being implemented across an organization using Palo Alto Networks products. A critical component is the dynamic creation and enforcement of micro-segmentation policies based on real-time threat intelligence. Consider a scenario where a new, highly evasive malware variant (file hash abc123def456) is detected communicating with a specific, ephemeral IP address (203.0.113.50) and attempting to exfiltrate data to a suspicious domain (dataleak.biz) via a unique URL (https://dataleak.biz/upload?id=user_data&token=XYZ). Describe how Cortex XSOAR, integrated with Cortex XDR and NGFWs, would dynamically leverage these distinct indicator types (file, IP, domain, URL) to enforce a Zero Trust posture and automate threat containment. Select ALL correct actions.

- ☐ Cortex XSOAR automatically pulls the file hash abc123def456 and dynamically updates a custom Anti-Malware profile on the NGFWs to prevent its download or execution across the network.
- ☐ Cortex XSOAR orchestrates the creation of a 'Security Policy' on the NGFW to explicitly deny all outbound connections from the infected source zone to the IP address 203.0.113.50, leveraging dynamic address groups if the IP is known to be transient.
- ☐ Cortex XDR, through its EDR capabilities, detects the process associated with abc123def456, automatically quarantines the file, and isolates the endpoint, preventing further lateral movement and network communication from the compromised host.
- ☐ Cortex XSOAR automatically adds the domain dataleak.biz to an External Dynamic List (EDL) on the NGFW, which is then referenced by a 'URL Filtering Profile' to block all HTTP/S traffic to this domain. Concurrently, it can add the full URL https://dataleak.biz/upload?id=user_data&token=XYZ to an XDR 'Custom Indicator' list for enhanced endpoint detection.
- ☐ Cortex XSOAR initiates a 'Live Terminal' session on affected endpoints, runs a custom PowerShell script to find and delete any instances of the file abc123def456, and then manually updates the local hosts file on each endpoint to block dataleak.biz.

**A.** オプションA
**B.** オプションB
**C.** オプションC
**D.** オプションD
**E.** オプションE
**Answer: B,C,D** (メーッセージを残す)
この質問は、ゼロトラストの適用のために、Cortex製品間で複数のインジケータータイプを動的に統合する能力を評価します。A（誤）XSOARはNGFWと統合できますが、特定のファイルハッシュでアンチマルウェアプロファイルを更新することは、NGFWの一般的な動的またはリアルタイムのアクションではありません。NGFWは主にファイルベースの防御にWildFireを使用し、Palo Alto Networksから動的な更新を受け取ります。XDRはエンドポイントのファイルブロッキングに適しています。B（正解）これは動的マイクロセグメンテーションの好例です。XSOARはNGFWセキュリティポリシーを自動的に作成または更新できます。エフェメラルIPに動的アドレスグループを使用すると、IPが変更されても柔軟にブロックできます。これにより、脅威インテリジェンス（IPインジケーター）に基づいてネットワークアクセスを制限することで、ゼロトラストを直接適用できます。C（正解）これはCortex XDRの中核機能です。悪意のあるファイル（ファイルハッシュインジケーター）が検出されると、XDRのEDR機能によってファイルが自動的に隔離され、エンドポイントが分離されます。これは、ラテラルムーブメントを防止し、ホストレベルで脅威を封じ込めるために不可欠であり、「決して信じず、常に検証する」というゼロトラスト原則に合致しています。D（正解）XSOARは、ドメインおよびURLインジケーターを効果的に運用化できます。NGFWのURLフィルタリングプロファイルで使用されるEDLにドメインを自動的に追加することで、疑

わしいドメインへの通信をネットワーク全体で即座にブロックできます。さらに、XDRの「カスタムインジケーター」リストにURL全体を追加することで、エンドポイント固有の検出機能が強化され、ドメインが他の目的で部分的に許可されている場合でも、XDRは特定のURLパターンへのアクセスを警告またはブロックできます。この包括的なアプローチは、URL/ドメインインジケーターのネットワーク層とエンドポイント層の両方をカバーします。E（不正解）：修復は「ライブターミナル」を使用できますが、手動のPowerShellスクリプトとローカルホストファイルの更新に依存することは、拡張性、自動化、および企業向けの動的なゼロトラスト適用との整合性に欠けます。XDRに組み込まれた防止ポリシーとXSOARのオーケストレーションが適切なツールです。

最新問題: **8**
組織は新しいクラウドネイティブアプリケーションを導入したばかりですが、脅威インテリジェンスによると、類似のクラウド環境における設定ミス、特にIAMロールとAPIキーの漏洩を狙った攻撃が急増していることが示唆されています。Palo Alto Networks Prisma Cloudを導入済みです。インシデント対応チームは、Prisma Cloud内のこの脅威インテリジェンスを積極的に活用し、基本的なポスチャ管理からアクティブな脅威検知・対応へと移行することで、潜在的なセキュリティインシデントを未然に防ぐことができますか？
**A.** 明示的な除外なしに「AdministratorAccess」を付与するすべての IAM ロールを自動的に修復し、公開されたすべての API キーを無効にするように Prisma Cloud を構成します。
**B.** Prisma Cloud でカスタム RQL (リソース クエリ言語) ルールを開発して、過度に許可されたポリシーを持つ IAM ロールを識別し、一般的な誤った構成に関する脅威インテリジェンスと相互参照し、自動化されたセキュリティ チェックのために CI/CD パイプラインと統合します。
**C.** Prisma Cloud で新しい IAM ロールの作成に関するアラートを設定し、脅威インテリジェンスの調査結果と照らし合わせて手動で確認します。
**D.** Prisma Cloud のネットワーク保護を使用して、脅威インテリジェンス フィードで識別された外部 IP アドレスから発信される異常な API 呼び出しをブロックします。
**E.** 公開された API キーと IAM の誤った構成を自動的に検出する Prisma Cloud 脅威インテリジェンス フィードにサブスクライブします。
**Answer: B (**メーッセージを残す**)**
この質問は、Prisma Cloud のようなクラウド セキュリティ ポスチャ管理 (CSPM) とクラウド ワークロード保護プラットフォーム (CWPP) 内で脅威インテリジェンスを積極的に活用し、単純な検出から予防的かつ自動化された対策に移行することに焦点を当てています。
オプション B (カスタム RQL ルール + CIICD 統合): これは最も効果的なプロアクティブなアプローチです。
カスタム RQL ルール: RQL は、特定のリソース構成と関係を識別するための Prisma Cloud の強力なクエリ言語です。
脅威インテリジェンス (一般的な構成ミス、過度に許容的なポリシーのパターンなど) を活用して正確な RQL ルールを記述することで、組織はクラウド環境を積極的にスキャンして、これらの脆弱性を正確に検出できます。

CIICDパイプライン統合 :これらのRQLチェックをCI/CDパイプライン（Prisma CloudのIacセキュリティ機能など）に統合することで、設定ミスのあるIAMロールや公開されたAPIキーをデプロイ前に検出し、本番環境でのインシデント発生を効果的に防止できます。これは、攻撃者のTTP（戦術・技術・プロセス）に関するインテリジェンスによって直接推進される「シフトレフト・セキュリティ」の実践例です。
他のオプションが最適ではない理由を分析してみましょう。
A: 「AdministratorAccess」の自動修復は（原則的には有効ですが）、きめ細かな制御や特定の脅威インテリジェンスからのコンテキストがなければ、範囲が広すぎて混乱を招く可能性があります。公開されているAPIキーを無効にするのは、事後対応的な対応です。
C: 動的クラウド環境において、手動によるレビューは、プロアクティブな予防策を講じるには拡張性や迅速性が不十分です。自動化が鍵となります。
D: Prisma Cloud のネットワーク保護は、ネットワーク レベルのトラフィック検査を目的としています。これは有用ですが、脅威インテリジェンスによって明らかにされた最初の攻撃ベクトルである IAM ロールと API キーの誤った構成に直接対処するものではありません。
E フィードを購読するのは良いことですが、インシデント対応チームがこのインテリジェンスをどのように積極的に活用して予防につなげるかが問われています。一般的なフィード購読では、そのインテリジェンスをカスタムRQLルールやCI/CD統合といったプロアクティブなセキュリティ制御に反映させるための具体的なアクションは説明されていません。

最新問題: **9**
ある大企業が、レガシーSOARプラットフォームをCortex XSOARに移行しています。既存のセキュリティツール向けにPythonで開発された多数のカスタムプレイブックと統合機能がありますが、これらはマーケットプレイスパックとして直接利用できません。移行中に、セキュリティアーキテクトは、カスタムロジックへの投資を維持しながらXSOARのマーケットプレイスを活用する戦略を提案しました。既存のカスタムコードとXSOARのマーケットプレイス機能を最も適切に統合できるアプローチはどれですか？また、スケーラビリティと保守性に関するアーキテクチャ上の考慮事項は何ですか？
**A.** すべてのカスタムPythonスクリプトをXSOARネイティブの自動化とコマンドに書き換え、プライベートマーケットプレイスパックとして公開します。これにより完全な互換性と一元管理が確保されますが、大幅なリファクタリング作業が必要になります。
**B.** 既存のPythonスクリプトをDockerでコンテナ化し、XSOAR内のカスタム統合としてデプロイします。必要に応じて、既存または新規に作成されたMarketplaceコンテンツにリンクします。これにより分離性と移植性が確保されますが、コンテナオーケストレーションのオーバーヘッドが増加します。
**C.** XSOAR の組み込み Python インタープリターを利用して、レガシースクリプトを自動化として直接実行し、新しい Marketplace プレイブックにラップします。これは最も速いアプローチですが、依存関係の競合やカスタムスクリプトのバージョン管理の欠如につながる可能性があります。

**D.** カスタムPythonスクリプトを、XSOARのHTTP統合を介してアクセス可能な外部サービスとして統合し、Marketplaceプレイブックからトリガーします。これによりロジックは分離されますが、ネットワークレイテンシと外部サービス管理が必要になります。
**E.** XSOARの「Bridge」統合を活用して、レガシースクリプトをホストしている別のサーバーに接続し、Marketplaceのプレイブックからこれらのスクリプトを呼び出します。これにより元の環境は維持されますが、複雑さが増し、単一障害点が発生する可能性があります。
**Answer: B (**メーッセージを残す**)**
オプション B は、既存のカスタム Python スクリプトを XSOAR に統合するための最も堅牢でアーキテクチャ的に健全なアプローチであり、投資を保護し、スケーラビリティと保守性を考慮します。コンテナ化 (例: Docker) により、カスタムコードとその依存関係をパッケージ化して、一貫した実行環境を確保できます。これらのコンテナは、XSOAR 内にカスタム統合としてデプロイでき、マーケットプレイス パックのプレイブックを含むプレイブックから呼び出すことができます。このアプローチは、カスタムコードに優れた分離性、移植性、およびバージョン管理を提供し、スケーラブルで保守性の高いコードを実現します。コンテナ オーケストレーションのオーバーヘッドは増加しますが、XSOAR のエンジンはこれらのコンテナを効果的に管理できます。オプション A は、大幅なリファクタリング作業であり、「投資の保護」を無効にします。オプション C には、依存関係とバージョン管理の問題があります。オプション D と E は、外部依存関係と潜在的なパフォーマンス/信頼性の問題をもたらします。

最新問題**: 10**
高度なランサムウェア攻撃による重大なインシデント対応において、セキュリティアナリストはCortex XSOARのWar Roomを使用しています。アナリストは、重要な発見事項、具体的にはマルウェアによってドロップされた固有のレジストリキーを文書化し、すべてのインシデント対応者がこの情報に即座にアクセスできるようにしたいと考えており、同時に、将来のフォレンジック分析のためにインシデントの証拠ロッカーに自動的に追加したいと考えています。アナリストはWar Roomのどの機能を活用すべきでしょうか。また、この包括的な文書化と証拠収集を最も効率的に実現する方法は何でしょうか。
**A.** アナリストは、ウォールームの「メモを追加」機能を使用し、レジストリキーを手動で貼り付け、メモを手動で証拠ロッカーに添付する必要があります。また、発見しやすくするために、メモに適切なタグを付けることも忘れないでください。
**B.** アナリストは「エントリの追加」機能を利用し、エントリタイプとして「証拠」を選択してください。レジストリキーを入力すると、XSOARが自動的にインシデントにリンクし、証拠ロッカーに記録します。これにより、ウォールームおよびインシデントのコンテキスト内で検索可能になります。
**C.** アナリストは、key=HKEY_LOCAL_MACHINE\SOFTWARE\MalwareDrop のようなカスタム War Room コマンドを実行する必要があります。このコマンドは、War Room エントリとして追加されるだけでなく、自動的に証拠として分類され、将来の検索のためにタグ付けされます。このコマンドにより、すべての共同作業者が即座に確認できるようになります。

**D.** アナリストは 「ジャーナル」タブを使用して発見事項を記録し、タイムスタンプが付与されていることを確認する必要があります。証拠収集のためには、 「証拠」タブに移動し、ジャーナルエントリを参照する新しい証拠項目を手動で追加する必要があります。
**E.** アナリストは、ウォールーム内の 「コマンドラインインターフェース」を活用して、関連する 「エビデンス」出力を持つプレイブックタスクを実行する必要があります。このタスクは、レジストリキーをウォールームとエビデンスロッカーに同時に直接記録することで、自動化と一貫性を確保します。
**Answer: (**解答を表示する**)**
オプションCは、最も効率的で堅牢な方法です。Cortex XSOARs War Roomは、カスタムコマンドや統合コマンドなど、さまざまなコマンドをサポートしており、特定のタイプの証拠、メモ、エントリを直接追加できます。のようなコマンド （または同様の事前構成済みコマンド/スクリプト）を使用すると、単一のアクションで複数の目的を達成できます。つまり、構造化されたWar Roomエントリの追加、証拠としての分類、検索用のタグ付け、そしてすべての共同作業者がすぐに閲覧できるようにすることです。オプションBとEも妥当ですが、オプションCは、構造化されたデータエントリと自動化された証拠処理のための直接コマンド実行の威力を特に強調しています。これは、効率的なインシデント対応を実現するWar Roomの主要な強みです。オプションBはエントリの追加について説明していますが、 「証拠」エントリタイプは、多くの場合、特定の証拠収集コマンドまたは出力に関連付けられています。オプションEは、プレイブックタスクの出力に関するものであり、必ずしもWar Room CLI内でアナリストが直接アクションを実行して証拠を即座に記録することを意味するものではありません。

最新問題**: 11**
ある大企業が、Cortex XDRの行動分析を用いた高度な内部脅威検知能力を評価しています。具体的なシナリオの一つとして、不満を抱えた従業員が正規のクラウドストレージサービスを利用して数週間かけて段階的に知的財産を盗み出そうとするケースが挙げられます。この活動は、通常の業務と混在しています。目標は、過度の誤検知を発生させることなく、この 「じっくりと時間をかけて」の盗み出しを検知することです。このシナリオにおいて、Cortex XDRの行動分析要素のどの組み合わせが最も強力な検知力を発揮し、導入においてどのような課題に対処する必要があるでしょうか。
**A.** 許可されていない宛先への大容量ファイル転送を検出するために、ネットワークベースの IPS シグネチャのみに依存します。
**B.** ユーザーおよびエンティティの行動分析 (UEBA) を活用して、クラウド サービスへの個々のユーザー データ転送パターンをベースライン化し、エンドポイント DLP と組み合わせて機密ファイルへのアクセスとアップロード イベントを監視します。
**C.** すべてのエンドポイントに静的ファイル分析エンジンを導入し、既知の悪意のあるファイルの種類をアップロード前に識別します。
**D.** 主に脅威インテリジェンス フィードを使用して、既知の悪意のあるクラウド ストレージ URL をブラックリストに登録します。

**E.** クラウド サービスへの異常なピーク トラフィック時間を検出するために、ネットワーク フロー分析 (NetFlow/IPFIX) のみを実装します。
**Answer: (**解答を表示する**)**
正当なクラウド サービスへの 「低速でゆっくりとした」内部者のデータ流出を検出することは、高度な行動分析、具体的には UEBA と DLP を組み合わせた典型的な使用例です。オプション B は、この最適な組み合わせを正しく識別します。ユーザーおよびエンティティ行動分析 (UEBA): これは非常に重要です。UEBA は、典型的なデータ転送量、送信先、クラウド サービスの頻度など、各ユーザーの 「通常の」行動のプロファイルを作成します。増分的で低速なデータ流出は、正当なサービスを使用している場合でも、最終的にこの確立されたベースラインから逸脱し、ユーザーの異常スコアの増加を引き起こします。UEBA は、静的ルールを回避するこれらの微妙で持続的な時間経過による変化を検出することに優れています。エンドポイント DLP (データ損失防止): UEBA が異常なデータ移動の 「方法」と 「場所」を検出するのに対し、DLP はファイルのコンテンツを検査することで 「内容」を追加します。DLPポリシーによって知的財産が機密情報と分類されている場合、クラウドサービス （ユーザーが通常使用している正当なサービスであっても）へのアップロード試行はすべてフラグ付けされ、ブロックまたは監査される可能性があります。課題 :オプションBで述べた課題は確かに存在します。ユーザーアクティビティの正当な変動により、クラウド使用量の正確なベースラインを確立することは確かに複雑になる可能性があります。同様に、DLPでは誤検知を最小限に抑えるために、慎重な構成とコンテンツ分類が必要です。ポリシーが強すぎると、正当なビジネス文書が誤ってフラグ付けされる可能性があります。しかし、これらは適切なチューニングとポリシーの改良によって克服できる運用上の課題であり、これが最も堅牢なアプローチとなります。

最新問題**: 12**
高度な持続的脅威 （APT）グループが、重要なサーバー上のプロプライエタリアプリケーション （C AppX.exe）のゼロデイ脆弱性を悪用することに成功し、権限昇格と永続化のためのスケジュールタスクの作成に成功しました。Cortex XDRはXDRストーリーを生成し、Causality Viewは熟練したセキュリティ運用担当者によって活用されています。侵害の全容を特定し、根絶に向けて準備する上で、Causality Viewで観察された以下の要素のうち、その後の脅威ハンティングとインシデント対応に最も重要な情報を提供するのはどれですか？また、その理由は何ですか？
**A.** Cortex XDR によってアラートがトリガーされた正確な時刻。これがインシデントの明確な開始時刻であり、レポート作成が簡単になります。
**B.** 宛先に関係なく、'AppX.exe' によって行われたすべてのネットワーク接続の完全なリスト。これは、ネットワーク アクティビティを広く示します。
**C.** 'AppX.exe' とその直接/間接の子プロセスで使用される特定のプロセス引数とコマンド ライン、ドロップされた新しい実行可能ファイルの完全なパス、永続性のためのレジストリの変更 (実行キー、サービスなど)、およびスケジュールされたタスクまたはサービスを作成するために使用される正確なコマンド。これらは、攻撃者の TTP、C2、および永続性メカニズムを明らかにするためです。

**D.** 侵害されたサーバーのオペレーティング システムのバージョンとパッチ レベル。これは、悪用された脆弱性を直接示します。
**E.** 過去 24 時間以内に同じエンドポイントで生成された他のアラートの数。これは、エンドポイントの全体的なセキュリティ態勢を示します。
**Answer: C (**メーッセージを残す**)**
APTレベルの侵害の場合、効果的なインシデント対応と将来の予防には、攻撃者の技術、戦術、手順（TTP）を理解することが不可欠です。オプションCは、Causality Viewが提供する最も重要なインテリジェンスを網羅しています。具体的なプロセス引数、コマンドライン、ドロップされた実行ファイル（およびそのパス）、永続化のためのレジストリ変更、スケジュールされたタスクの正確なコマンドなどから、以下の情報が直接的に明らかになります。1. 具体的なエクスプロイト手法（コマンドライン引数経由）。2. 永続化が確立された場所とその削除方法。3. 新しいプロセスによって確立されたコマンドラインまたはネットワーク接続から得られるファイルハッシュやC2ドメイン/IPなどの侵害指標（IOC）。このレベルの詳細は、標的型脅威ハンティングの策定、検出ルールの開発、そして脅威の完全な根絶を確実にするために不可欠です。他のオプションでもある程度の背景情報は提供されますが、高度な攻撃への対応策に直接役立つ、オプションCのような実用的で詳細なインテリジェンスは提供されません。

最新問題**: 13**
インシデント対応中に、SOCは重要なアプリケーションサーバーが異常な動作を示していることを発見しました。異常な動作には、CPU使用率の上昇や、既知のボットネットC2へのアウトバウンド接続などが含まれます。このサーバーはEDRソリューションによって管理されていません。この管理されていないサーバーにおける迅速なフォレンジック分析と駆除に最も効果的なPalo Alto Networksツールはどれですか？また、どのような重要なデータが得られますか？
**A.** Cortex XDR Pro (マネージド EDR ソリューション)。エンドポイント エージェントからプロセスの因果関係、ファイル アクティビティ、ネットワーク接続を直接提供します。
**B.** Palo Alto Networks NGFW (次世代ファイアウォール)。送信接続に対する詳細なパケット検査ログとアプリケーション レベルの可視性を提供します。
**C.** Cortex XDR (Lite/Unmanaged)。ライブフォレンジック収集、メモリダンプの収集、実行中のプロセス、ネットワークアーティファクトのためにオンデマンドで展開できます。
**D.** WildFire (クラウドベースの脅威分析サービス)。疑わしいファイルを分析してマルウェアを検出しますが、サーバーからライブフォレンジックデータを直接提供することはありません。
**E.** Prisma Cloud (クラウド セキュリティ プラットフォーム)。クラウド ワークロードを保護しますが、これはオンプレミスのサーバー シナリオです。
**Answer: C (**メーッセージを残す**)**
サーバーはEDRによって管理されていないため、Cortex XDRの「Lite」またはオンデマンド導入機能は、エージェントをフルインストールすることなく、迅速なフォレンジック情報収集に最適です。これにより、メモリダンプ、実行中のプロセス、ネットワークアーティファクトなどの重要なライブデータを収集できます。Cortex XDR Pro (A) は事前の導入が必要です。NGFW (B) はネッ

トワークレベルの可視性を提供しますが、エンドポイントでのフォレンジックは直接提供しません。WildFire (D) はファイル分析用です。Prisma Cloud (E) はクラウド環境用です。

最新問題: **14**

Linuxサーバー上で実行されているカスタムアプリケーションが侵害された疑いがあります。脅威アクターは、アプリケーションのゼロデイ脆弱性を悪用して任意のコードを実行し、リバースシェルを確立しようとしていると考えられています。このLinuxサーバーにはCortex XDRエージェントが導入されています。SOCアナリストであるあなたは、リバースシェルを開始したプロセス、その親プロセス、そして疑わしい外部IPへのアウトバウンドネットワーク接続を正確に特定する必要があります。リバースシェルは通常、権限のないユーザーに代わって通常とは異なる場所からポート443に接続すると仮定した場合、この特定の調査において、Cortex Data Lakeに対するどのXDRクエリ言語（XQL）クエリが最も効果的でしょうか？

**A.**

```
dataset = xdr_data | filter event_type = PROCESS_EXECUTION and action_process_image_name = 'bash' or action_process_image_name = 'sh' |
join (dataset = xdr_data | filter event_type = NETWORK_CONNECTION and action_network_connection_destination_port = 443 and
action_network_connection_direction = 'outbound' ) as network_events on network_events.action_process_id = action_process_id |
fields action_process_image_name, action_process_command_line, network_events.action_network_connection_destination_ip, network_events.
action_network_connection_destination_port, actor_process_image_name, actor_process_command_line | sort by _time desc
```

**B.**

```
dataset = xdr_data | filter event_type = NETWORK_CONNECTION and action_network_connection_destination_port = 443 and
action_network_connection_direction = 'outbound' | join (dataset = xdr_data | filter event_type = PROCESS_EXECUTION ) as process_events
on process_events.action_process_id = action_process_id | fields action_process_image_name, action_process_command_line, process_events.
actor_process_command_line, action_network_connection_destination_ip | sort by _time desc
```

**C.**

```
dataset = xdr_data | filter event_type = FILE_WRITE and action_file_path contains '/tmp' | fields action_file_name, action_file_path,
actor_process_image_name | sort by _time desc
```

**D.**

```
dataset = xdr_data | filter event_type = AUTHENTICATION and action_authentication_status = 'failed' | fields actor_user_name,
action_authentication_method | sort by _time desc
```

**E.**

```
dataset = xdr_data | filter event_type = DNS_QUERY and action_dns_query_status = 'NXDOMAIN' | fields action_dns_query_name,
actor_process_image_name | sort by _time desc
```

**Answer: (**解答を表示する**)**

リバースシェルのプロセス、その親、および送信接続を識別するには、ネットワーク接続イベントとプロセス実行イベントを相関させる必要があります。オプション B は、関連するネットワーク接続 (ポート 443 の送信) をフィルタリングすることから開始し、プロセス ID を使用してこれをプロセス実行データと結合します。これにより、ネットワーク接続を担当するプロセスとその親 (process_events.actor_process_command_line)、および宛先 IP を識別できます。オプション A には誤った結合条件があります。最初に bash/sh をフィルタリングし、次に process_id に基づいて結合しようとするため、他のリバースシェルバイナリが見逃される可能性があります。オプション C、D、および E は、リバースシェルのプロセスとネットワークアクティビティをトレースするという特定の目的とは無関係です。

最新問題: **15**

貴社は、フィッシングによる初期アクセス、権限昇格、ラテラルムーブメント、そしてデータ窃取を含む、高度な多段階攻撃キャンペーンに見舞われています。Cortex XSIAMは、エンドポイント、ネットワーク、クラウドなど、様々なセキュリティドメインにわたって多数のアラートを生成し

ています。攻撃者の戦術、手法、手順（TTP）を完全に把握し、同期した防御体制を構築するには、この複雑な攻撃の記録を集約、相関分析、可視化するためのXSIAMのどの機能が不可欠ですか？

**A.** 個々の重大度の高いアラートにのみ焦点を当て、外部のスプレッドシートを使用して手動で相関させます。

**B.** XSIAM のインシデント グラフ (攻撃ストーリーライン) を使用して攻撃チェーン全体を視覚化し、各 TTP に XSIAM の MITRE ATT&CK マッピングを活用し、コンテキストの脅威インテリジェンス フィードを強化します。

**C.** 攻撃が主にネットワークに限定されていると想定して、分析をネットワーク関連のアラートのみに制限します。

**D.** ノイズを減らすために重大でないアラートをすべて無効にすると、全体的な攻撃ストーリーに寄与する重要な低重大度の指標が見逃される可能性があります。

**E.** より広範な攻撃手法を理解せずに、1 つの分離された IOC に基づいて新しいカスタム防止ルールのセットを実装します。

**Answer: B (メーッセージを残す)**

Cortex XSIAMのインシデントグラフ（攻撃ストーリーライン）は、まさにこのようなシナリオを想定して設計されています。様々なソースから収集された関連するアラートやイベントを、MITRE ATT&CKにマッピングされた一貫したタイムラインに自動的にまとめます。これにより、攻撃を包括的かつ視覚的に把握できるようになり、TTP（攻撃のプロセス）の特定と多面的な対応の調整が容易になります。脅威インテリジェンスを追加することで、コンテキスト情報もさらに強化されます。

最新問題: **16**

APT（Advanced Persistent Threat）攻撃グループは、コマンドアンドコントロール（C2）通信にカスタム難読化PowerShellスクリプトを使用することが知られています。SOCは、Cortex XSIAMのデータ取り込み機能を活用し、PowerShellのコマンドライン引数とネットワーク接続を分析することで、これらのC2アクティビティを検出したいと考えています。XDRエージェントがエンドポイントに導入され、ネットワークログがネットワークデータコレクターを介して取り込まれる場合、プロセス実行用のデータセット「endpoint_exec」とネットワークデータ用のデータセット「network connection」を想定し、取り込まれたデータを最も効果的に活用して疑わしいPowerShell C2を特定するには、次のXQLクエリのうちどれが適切でしょうか？

**A.**

```
dataset = endpoint_exec
| filter process_name = "powershell.exe" and command_line contains "-nop" and command_line contains "-encodedcommand"
| join (dataset = network_connections | filter action = "allowed") as network_data on src_ip = actor_ip
| filter network_data.port = 443 and network_data.dvc_direction = "outbound"
| project _time, actor_process_id, actor_process_name, command_line, network_data.dvc_ip, network_data.dest_ip, network_data.port
```

**B.**

```
dataset = network_connections
| filter dvc_direction = "outbound" and port = 80 or port = 443
| join (dataset = endpoint_exec | filter process_name = "powershell.exe") as endpoint_data on dvc_ip = actor_ip
| filter endpoint_data.command_line contains "-enc" and endpoint_data.command_line contains "http"
| project _time, dvc_ip, dest_ip, port, endpoint_data.actor_process_name, endpoint_data.command_line
```

**C.**

```
dataset = endpoint_exec
| filter process_name = "powershell.exe" and (command_line contains "-w hidden" or command_line contains "-enc" or command_line contains "-nop")
| lookup external_ip_reputation_list on dest_ip
| filter reputation = "malicious"
| project _time, actor_process_name, command_line, dest_ip
```

**D.**

```
dataset = endpoint_exec
| filter process_name = "powershell.exe" and (command_line contains "-nop" and command_line contains "-w hidden")
| alter encoded_command = extract(command_line, '(?i)-encodedcommand\s+([^\s]+)', 1)
| filter encoded_command != null
| join type=left (dataset = network_connections | filter dvc_direction = "outbound" and (port = 80 or port = 443)) as network_data on actor_ip = network_data.src_ip and _time between network_data._time - 1h and network_data._time + 1h
| filter network_data.dest_ip != null
| project _time, actor_process_name, command_line, encoded_command, network_data.dest_ip, network_data.port
```

**E.**

```
dataset = endpoint_exec
| filter process_name = "powershell.exe" and (command_line contains "-file" or command_line contains "-c")
| union (dataset = network_connections | filter dvc_direction = "outbound")
| project _time, process_name, command_line, dest_ip, port
```

**Answer: (解答を表示する)**

カスタム難読化PowerShell C2を検出するには、疑わしいPowerShellプロセスの実行とアウトバウンドネットワーク接続を相関させる必要があります。オプションAは良い出発点ですが、「join」条件「src_ip = actor_ip」は、「endpoint_exec」に「actor_ip」が存在することを前提としています。これは通常「dvc_ip」または「」です。相関付けにおいて重要な「join」条件の時間枠も欠落しています。また、「network_data.port = 443」と「dvc_direction = "outbound"」もフィルタリングしますが、これは優れたヒューリスティックですが、「port =」はC2でもよく使用されます。オプションBはネットワーク接続から開始しますが、これは範囲が広すぎる可能性があります。「join」条件「dvc_ip = actor_ip」の方が適していますが、「filter endpoint_data.command_line contains "http"」は難読化されたスクリプトに対して十分に具体的ではなく、「enc」は「encodedcommand」と部分一致です。オプションCはIPレピュテーションに焦点を当てています。これは有用ですが、未知のC2の取り込まれたPowerShellとネットワークアクティビティに直接結びついていません。プロセス実行とネットワークアクティビティを相関させません。オプションDは最も包括的かつ正確です。「powershell.exe」と一般的な難読化引数（「nop」、「whidden」）を正しくフィルタリングします。「extract」を使用して、「encodedcommanff」部分を具体的に抽出します。これは難読化されたC2の強力な指標です。「actor_ip =」を使用して「network_connections」との「left join」を実行し、同時に発生しない可能性のあるイベントを相関させるために、適切な時間枠（1時間）を設定した「time between」句を含めます。一般的なC2ポート（80または443）での送信接続を明示的にフィルタリングします。結合後に「」が存在することを確認し、相関が成功したことを確認します。このクエリは、特定のコマンドライン引数を検索し、それらを送信ネットワーク接続と相関させることで、難読化されたPowerShellを検出するというニーズに直接対応します。オプションEは「ユニオン」を使用するため、結果を相関させるのではなく結合するため、特定のプロセス実行と特定のネットワーク接続を関連付ける効果は低くなります。また、PowerShellのフィルターはC2検出には範囲が広すぎます。

最新問題: **17**
セキュリティアナリストは、Cortex XSIAM における潜在的な内部脅威シナリオを調査しています。あるユーザーが、許可されていないクラウドストレージサービスを介してデータを流出させているのではないかと疑っています。SOC は、エンドポイントアクティビティ、プロキシサーバー、ファイアウォールログなど、様々なソースからログを受信しています。これを効果的に検知するために、必要なデータポイントの取り込みと相関分析を行う上で、Cortex XSIAM のどの機能が重要であり、その理由は？
**A.** 長期保存用のデータ レイクとデバイス コンテキスト用の資産インベントリ。
**B.** クラウド サービス ログのクラウド フィード統合と異常検出のためのユーザー ビヘイビア分析 (UBA)。
**C.** 高度な相関ルールと組み合わせた、Common Event Format (CEF) と Syslog のネイティブ サポート。
**D.** 詳細なプロセスおよびファイル アクティビティ用のエンドポイント データ コレクター、ネットワーク フローおよびプロキシ ログ用のネットワーク データ コレクター、および非標準ソースのカスタム「データセット」スキーマを定義する機能。
**E.** 既知の悪意のある IP に対するインシデント対応および脅威インテリジェンス フィードのための自動プレイブック実行。
**Answer: D** (メーッセージを残す)
データ流出を検知するには、エンドポイントアクティビティ（どのプロセスがどのファイルにアクセスしたか）、ネットワークトラフィック（クラウドサービスへの接続、データ量）、プロキシログ（アクセスされたURL）の詳細な可視性が不可欠です。Cortex XSIAMのエンドポイントデータコレクターは、きめ細かなエンドポイントテレメトリを提供し、ネットワークデータコレクターはネットワークフローとプロキシログに不可欠です。カスタム「データセット」スキーマを定義できるため、脅威に関連する非標準または独自のログであっても、取り込み、分析のために適切に構造化できます。XSIAMの機能には他にもありますが、Dほど包括的に、この特定の調査に必要な基礎的なデータの取り込みと構造化に直接対応していません。

最新問題: **18**
ゼロデイ攻撃は、組織内で広く使用されているアプリケーションを標的とし、初期の侵入に成功しました。セキュリティチームは、Palo Alto Networksの次世代ファイアウォール（NGFW）を介して異常なネットワークトラフィックパターンを検知し、侵入されたホストを特定しました。NIST

インシデント対応計画の 封じ込め」フェーズにおいて、ゼロデイ攻撃の性質を考慮し、影響範囲を限定しつつ重要な脅威インテリジェンスを収集するために、どの戦略的および戦術的アクションを優先すべきでしょうか？
（該当するものをすべて選択してください）
**A.** NGFW にカスタム URL フィルタリング プロファイルを直ちに適用し、指定されたフォレンジック サーバーを除き、侵害されたホストからのすべての送信接続をブロックします。
**B.** Cortex XDR を利用して侵害を受けたホストをネットワークから分離し、横方向の移動を防止するとともに、詳細なテレメトリ キャプチャのための強化されたログ記録を有効にします。
**C.** 脅威インテリジェンスから特定された疑わしい C2 ドメインに対して、NGFW に一時的な シンクホール」構成を展開し、悪意のあるトラフィックを制御された環境にリダイレクトしてさらに分析します。
**D.** パッチがまだベータ版であっても、脆弱なアプリケーションに対するグローバル緊急パッチをすべてのエンタープライズ エンドポイントにプッシュします。
**E.** 影響を受けるすべてのユーザーにインシデントについて電子メールで通知し、パスワードを直ちに変更するように指示します。
**Answer: A,B,C (**メーッセージを残す**)**
封じ込め」フェーズは、インシデントの範囲を限定するために極めて重要です。ゼロデイ攻撃の場合、拡散の抑制とインテリジェンス収集を同時に行うことが重要です。- A: 侵害を受けたホストに対するカスタムURLフィルタリング またはセキュリティポリシー）は、ネットワークレベルの正確な封じ込めを実現しながらも、制御対象システムへのフォレンジックデータの流出を可能にします。- B: Cortex XDRの分離はエンドポイント封じ込めに不可欠であり、ラテラルムーブメントを防止し、拡張ログ機能を有効にすると、インシデント後の分析と新しいIOC生成のための詳細なテレメトリを確保できます。- C: シンクホール構成は、C2トラフィックに対する高度な封じ込めおよびインテリジェンス収集手法であり、SOCは攻撃者の能力をさらなる侵害なしに把握できます。- D: ベータパッチをグローバルにプッシュすることは非常にリスクが高く、標準的な変更管理に違反し、さらなる混乱を引き起こす可能性があります。- E: ユーザーに直ちに通知し、パスワードの変更を指示することは、復旧やコミュニケーションの一部となる可能性がありますが、ゼロデイ攻撃自体に対する主要な技術的封じ込め手順ではありません。

最新問題**: 19**
Palo Alto Networks XSOAR をインシデント管理に使用しているセキュリティオペレーションセンター SOC）は、毎日大量のアラートを受信します。アナリストは、潜在的なデータ漏洩に関連するインシデントの優先順位付けを任されています。以下のインシデント分類基準のうち、技術的な指標とビジネスへの影響の両方を考慮し、データ漏洩インシデントの正確な優先順位付けを最も効果的に促進できるのはどれですか？
**A.** 送信元IPの位置情報と宛先ポート。これらは有用ですが、それだけではデータ流出の全体像を把握できない可能性があります。

**B.** 脅威インテリジェンスフィードの一致（例Unit 42のC2 IP）と影響を受ける資産の重要度（例：クラウンジュエル資産）。これにより、技術的な指標とビジネスへの影響度が組み合わされ、効果的な優先順位付けが可能になります。
**C.** 時間帯とユーザーの部門。これらは主に状況に基づくものであり、差し迫った脅威の深刻度を示すものではありません。
**D.** 特定のセンサーからのアラート量と使用プロトコル。アラート量は誤解を招く可能性があり、プロトコルだけではデータの流出を示唆しない可能性があります。
**E.** ファイルハッシュレピュテーション（WildFire）とエンドポイントOSバージョン。ファイルハッシュはマルウェア検出に役立ちますが、OSバージョンは主要な情報漏洩指標ではありません。
**Answer: (**解答を表示する**)**
データ窃盗におけるインシデントの効果的な優先順位付けには、強力な技術的指標とビジネスへの影響の理解を組み合わせる必要があります。Unit 42（Palo Alto Networksの脅威調査チーム）のような信頼できる脅威インテリジェンスソースから、IPアドレスと既知のコマンドアンドコントロール（C2）サーバーを照合することで、潜在的な侵害の高精度な技術的指標が得られます。これに、影響を受ける資産の重要度（例えば、「クラウンジュエル」に分類される機密性の高い顧客データをホストするサーバー）を組み合わせることで、ビジネスリスクが直接的に把握され、正確な優先順位付けが可能になります。他の選択肢は、窃盗に関する技術的な詳細が不十分であるか、ビジネスへの影響を適切に考慮していません。

最新問題**: 20**
高度な電子メールベースの攻撃が初期防御を回避し、悪意のあるペイロードを送り込みます。インシデントはCortex XSOARでトリガーされます。プレイブックは、次の処理を実行するように設計されています。1. すべての電子メールヘッダーと本文の内容を抽出する。2. サンドボックスで疑わしい添付ファイルをデトネーションする。3. 電子メールとサンドボックスの結果から、すべてのURLとファイルハッシュを抽出する。4. これらのIOC（侵入の痕跡）について、複数の外部脅威インテリジェンスフィード（VirusTotal、AlienVault OTXなど）を照会する。5. いずれかのIOCが悪意のあるものであると確認された場合、電子メールセキュリティゲートウェイで送信者のメールアドレスをブロックし、プロキシで悪意のあるURLをブロックし、すべての受信トレイから元のメールを隔離する。プレイブックがサンドボックス統合でサポートされていない添付ファイルの種類に遭遇し、その結果、その特定の添付ファイルのハッシュが抽出されないものの、電子メールの他の部分と他の添付ファイルは正常に処理され、実行される場合、この特定の部分的な障害を適切に処理して調査を続行するために望ましいXSOARプレイブックの設計を最もよく表しているのは次のうちどれですか。
**A.** サンドボックスが添付ファイルに対してエラーを返す場合、プレイブックは実行を完全に停止し、プロセス全体を再起動するために手動介入が必要になります。
**B.** サンドボックスデトネーションタスク専用の「エラー処理」パスを実装します。「ファイルがサポートされていません」というエラーが返された場合は、警告をログに記録し、アナリストがサポートされていないファイルタイプを外部で確認するための手動タスクを作成します。ただし、プレイブックは他の添付ファイル、メールの内容、および残りのIOCの処理を続行できます。

**C.** サポートされていないファイルの種類を黙って無視するようにサンドボックス統合を構成します。これにより、エラーが防止されるだけでなく、添付ファイルから悪意のあるコンテンツが失われる可能性もあります。
**D.** プレイブックは、すべての添付ファイルが正常に実行されたと想定する必要があります。そうでない場合、サンドボックスの結果に依存する後続のタスクは、特別な処理なしで失敗します。
**E.** プレイブックには、サンドボックスに送信する前にファイル拡張子をチェックする条件付きタスクが必要です。サポートされていないタイプの場合、そのファイルのサンドボックス ステップはスキップされますが、電子メールの残りの部分から IOC は引き続き抽出されます。
**Answer: B,E (**メーッセージを残す**)**
BとEはどちらも、部分的な障害を適切に処理し、調査を継続するための強力なソリューションです。オプションBは堅牢なエラー処理を示しています。サンドボックスタスク専用の 「エラー処理」パスを実装することで、プレイブックは次のことが可能になります。1. 「ファイルがサポートされていません」などの特定のエラーをキャッチする。2. 警告をログに記録する（インシデント全体でプレイブックを停止させるような完全なエラーではない）。3. アナリストが特定のサポートされていないファイルに対処できるようにするための手動タスクを作成し、データの欠落を防ぐ。4. 重要なのは、プレイブックの残りの部分が他の添付ファイルやメールコンテンツの処理を続行できるようにすることです。これにより、サポートされていない単一のファイルによってインシデント対応全体が停滞することがなくなります。オプションEは、条件付きロジックを用いたプロアクティブな設計アプローチです。サンドボックスに送信する前にファイル拡張子をチェックすることで、既知のサポートされていないファイルタイプでエラーが発生するのを事前に防ぎます。これにより、プレイブックは残りの処理を続行できます。これによりエラーは回避されますが、すべてのエッジケースや、更新なしでは新たにサポートされなくなったタイプをキャッチできない可能性があります。しかし、これは既知の制限に対処するための有効かつ効率的な方法です。オプションA、C、Dは望ましくありません。Aは調査全体を停止します。Cはサイレントデータ損失につながります。Dは、正常な回復が不可能な後続の障害につながります。

最新問題**: 21**
ある金融機関は、Cortex XSOARを使用して脅威インテリジェンスを管理しています。外部フィードから新たに取り込まれたすべての指標は、適用ポイント（ファイアウォール、EDRなど）にプッシュされる前に、必ず人間によるレビュープロセスを経るという厳格な要件があります。ただし、「重大」なレピュテーションを持つ指標（信頼性の高いプライベートフィードなど）は、このレビューをバイパスして即座にブロックする必要があります。さらに、「高」レピュテーション指標のレビュープロセスには特定のチームが関与する必要がありますが、「中」レピュテーション指標は、別の大規模なチームでレビューできます。指標プレイブックとレピュテーション管理を活用し、これらの複雑なワークフローを効率的に管理するには、Cortex XSOARをどのように構成すればよいでしょうか？
**A.** インジケーターの評判に基づいて条件付きタスクを記述した単一の「インジケータープレイブック」を設定します。人間によるレビューには「手動タスク」を使用し、「条件付きブランチ」では

評判に基づいて「タスク担当者」を使用して異なるチームにタスクを割り当てます。重要な評判インジケーターは、手動タスクをバイパスするブランチに従います。
**B.** 3つの「インジケータープレイブック」を作成します。1つは「重大」、「高」、「中」のレピュテーション用です。各インジケーターの取り込み後、適切なプレイブックを手動でトリガーします。重大インジケーターのプレイブックにはレビューは含まれませんが、その他のプレイブックには特定のユーザーグループに割り当てられた手動レビュータスクが含まれます。
**C.** それぞれのレピュテーションレベル（重大、高、中）ごとに異なる「脅威インテリジェンスフィード」を設定します。各フィードには、特定のレビュー要件と適用アクションを処理するための「取り込みプレイブック」が設定されます。重大フィードの取り込みプレイブックは直接適用にプッシュされ、その他のフィードの取り込みプレイブックにはレビュータスクが含まれます。
**D.** 「インジケータータグ」を使用して、異なるレビューチームのインジケーターをマークします。特定のタグが付けられたインジケーターを定期的にクエリし、対応するレビューキューに自動的に割り当てる「スケジュールジョブ」を実装します。重要なインジケーターにはレビュー用のタグが付けられません。
**E.** これを実現する唯一の方法は、各指標の取り込み後に手動でレピュテーションを調整し、事前定義された自動化をトリガーしてブロックまたはレビューを実行することです。重要な指標は、レビューをバイパスするために手動で「重要」に設定します。
**Answer: A,C (**メーッセージを残す**)**
AとCはどちらも、この複雑なシナリオに対して実行可能かつ堅牢なソリューションであり、高度なXSOAR機能を発揮します。オプションA（条件付き単一インジケータープレイブック）これは、単一のプレイブック内で多様なワークフローを管理する非常に効率的な方法です。インジケーターの取り込み（任意のフィードから取得可能）が行われると、単一のインジケータープレイブックがトリガーされます。このプレイブックの内容：「条件分岐」（例：indicator.reputation「Critical」）は、重大な指標を、手動レビュータスクをバイパスして、直ちに適用へとプッシュするパスに誘導します。他の分岐（「elif indicator.reputation「High」および「elif indicator.reputation「Medium」）には、「手動タスク」ステップが含まれます。これらの手動タスクの「タスク担当者」は、指標のレピュテーションに基づいて異なるユーザーグループまたはロールに動的に設定できるため、チーム固有のレビューを実現できます。オプションC（専用の取り込みプレイブックを使用した複数のフィード）このアプローチは、フィード固有の取り込みプレイブックの柔軟性を活用します。ソースフィード自体がレピュテーションを確実に分類する場合：「Critical」、「High」、「Medium」のレピュテーション指標を提供することが知られているソースごとに、個別の「脅威インテリジェンスフィード」を設定できます（または、フィード自体を単純に分類することもできます）。各フィードには、異なる「取り込み」ステップが設定されます。「プレイブック」です。「重要フィードの取り込みプレイブック」は直ちに適用されます。「高レベルフィードの取り込みプレイブック」には、「高レベルチーム」に割り当てられた「手動タスク」が含まれます。「中レベルフィードの取り込みプレイブック」には、「中レベルチーム」に割り当てられた「手動タスク」が含まれます。どちらのアプローチも有効であり、脅威インテリジェンスが上流でどのように受信され、分類されるかによって選択が異なります。オプションBは手動でトリガーする必要があるため非効率的です。オプショ

ンDは事後対応型で、即時性に欠けます。オプションEは完全に手動であり、自動化の目的に反します。

最新問題: **22**
セキュリティ運用担当者が複雑なXDRストーリーを分析しています。このストーリーでは、攻撃者が正規の「notepad.exe」プロセスに対してプロセスホロウイングを用いて従来のウイルス対策を回避し、悪意のあるコードを実行しました。その後、このコードは改変された「procdump.exe」を使用して資格情報のダンプを実行し、イベントログを消去しようとしました。Cortex XDRのCausality Viewはここで非常に重要になります。「notepad.exe」と「procdump.exe」は正規のバイナリであることを踏まえると、Causality Viewはどのような重要な動作異常とプロセス間関係を浮き彫りにすることで、この巧妙な攻撃を明らかにすることができるでしょうか。また、なぜこの種の分析がCortex XDRで特に効果的なのでしょうか。
**A.** 因果関係ビューでは、「notepad.exe」が「不明」なデジタル署名を持っていると表示され、変更されたことが示されます。
**B.** 'notepad.exe' の元の親プロセス、続いて、改ざんされた notepad.exe のプロセス ID から予期せぬ子プロセス ('procdump.exe') が作成された様子、LSA をターゲットとした 'procdump.exe' のコマンドライン引数、そして関連プロセスによるイベントログの消去試行が明確に表示されます。複数の正規プロセス間の動作の逸脱をグラフィカルに相関関係として表示できるこの機能は、高度な脅威の検出における Cortex XDR の Causality View の核となる強みです。
**C.** Causality View は、'notepad.exe' プロセスに対してメモリフォレンジックを自動的に実行し、挿入された悪意のあるコードを抽出して署名分析を行います。
**D.** 実行コンテキストに関係なく、エンドポイントに 'procdump.exe' バイナリが存在する場合に特に警告します。
**E.** 因果関係ビューでは、特定のイベントを表示せずに、「プロセス ホローイング」および「資格情報ダンプ」の MITRE ATT&CK フレームワークへの直接リンクが提供されます。
**Answer: (**解答を表示する**)**
正規バイナリを用いたプロセスホローイングやクレデンシャルダンプといった高度な手法を検出するには、詳細な行動分析が必要です。Cortex XDRのCausality Viewは、まさにこの点に優れています。オプションBは、Causality Viewが強調表示する重要な要素を正しく特定しています。1. 'notepad.exe'の親プロセス：最初の'notepad.exe'がどのように起動されたかを観察する。2. 正当な親プロセスからの予期せぬ子プロセスの作成: 重要なのは、「procdump.exe」が、通常の親プロセスではなく、ホローイングされた「notepad.exe」のPIDによって生成されたことです。通常の「notepad.exe」の動作からのこの逸脱は、侵害の強力な指標です。3. 「procdump.exe」コマンドライン: 特定の引数(C-accepteula'、ma'、lsass.exe)は、資格情報のダンプの直接的な指標です。4. イベントログのクリア: イベントログのクリア(Cwevtutil.exe cl System'、'wevtutil.exe cl Security')などの後続アクションは、痕跡を隠すためのエクスプロイト後の一般的なアクティビティです。Cortex XDRのCausality Viewの強みは、正当なプロセスからのこれらの一見異なるイベントを、単一の一貫性があり視覚的に理解可能な攻撃チェーンに関連付け、シグネチャのみに頼るのではなく、動作の異常性を強調表示できることです。バイナリ自体を分析することで、アナリストは従来のシ

グネチャベースの検出を回避する高度な攻撃を迅速に特定できます。オプションA、C、D、Eは、このような複雑なシナリオに対して、誤った機能または不完全な分析アプローチを示しています。

最新問題: **23**
ある組織がセキュリティ運用をCortex XSIAMに移行しようとしています。現在、既存のSIEMには、独自のクエリ言語で定義された数千ものカスタム相関ルールが存在します。セキュリティ運用プロフェッショナルとして、これらのルールをXSIAM向けに翻訳・最適化する役割を担っています。特に、XSIAMの自動相関機能を活用し、純粋な「アラート中心」から「インシデント中心」の検知へと移行することに重点を置きます。この移行において、どのような主要な課題に直面するでしょうか。また、XSIAMの機能はどのように役立つでしょうか。特に、IOCと高忠実度BIOCの違いについて教えてください。
**A.** 主な課題は、単純に構文の変換です。XSIAM の相関関係は従来の SIEM と同一です。IOC と BIOC は、名前が異なるだけで概念は同じです。
**B.** 主な課題としては、XSIAM の統合データモデルへの適応（データスキーマの包括的な理解が必要）、レガシーロジックの XQL への変換、そして単純な「イベント発生時アラート」ルール（多くの場合 IOC に類似）をより複雑な「行動インシデント」ルール（BIOC）にリファクタリングすることが挙げられます。XSIAM は、強力なコンテキスト検索を実現する XQL クエリ言語、関連するアラートを包括的な「インシデント」に自動的にまとめる相関エンジン、そして複雑な攻撃のナラティブを表す高精度 BIOC を定義する機能を提供することで、アラート数を削減し、真の脅威に集中できるようにします。IOC は単一の静的な指標ですが、BIOC は侵害を示すイベントと行動の複合体であり、より高精度なインシデントにつながります。
**C.** 主な問題はデータ量です。XSIAM は従来の SIEM ほど大量のデータを処理できません。XSIAM は事前に構築されたルールのみを使用するため、カスタムルールの変換はできません。IOC は動的ですが、BIOC は静的です。
**D.** XSIAM には、すべてのレガシー SIEM ルールを直接インポートできるツールが用意されているため、問題はありません。XSIAM のインシデント概念は、個々のアラートを再パッケージ化したものにすぎません。IOC と BIOC はどちらもハッシュ値に基づいています。
**E.** 移行には、XSIAM のファイアウォールモジュール内のすべてのルールを防御ポリシーとして書き換える必要があります。XSIAM は検出用に設計されていません。IOC はネットワーク用、BIOC はエンドポイント用です。
**Answer: B (**メーッセージを残す**)**
この質問は、従来のSIEMからXSIAMへの移行における実際的な課題を取り上げ、コアとなるアーキテクチャと概念の違いを強調しています。主な課題は、XSIAMの統合データモデル（従来のSIEMとは異なる、より包括的なデータ構造）への適応、独自のクエリ言語をXQLに変換すること、そして個別のアラート（多くの場合IOC主導）への対応から、包括的なインシデントのプロアクティブな特定（BIOCと自動相関分析主導）への根本的な転換です。XSIAMがこの点で優れているのは、相関分析エンジンが、異なるドメイン（エンドポイント、ネットワーク、クラウド、アイデンティティ）にまたがる関連するセキュリティイベントを、単一の高精度な「インシデント」に自動的にリンクするためです。これにより、アラート疲れが大幅に軽減され、攻撃の全体像がより明確に

なります。高精度なBIOCは、単なる個別の悪意のある兆候ではなく、真の脅威を示す複雑で多段階的な動作を記述するため、この文脈において極めて重要です。IOCは低コンテキストで静的な指標（例：既知の悪意のあるIP）ですが、BIOCは高コンテキストで豊富な行動パターン（例：不審なプロセスの生成、それに続くネットワークビーコン、そしてデータアクセス。これらはすべて、通常とは異なるログイン時間を持つユーザーによるものです）です。目標は、多数の低忠実度のIOCアラートから、より少数の高忠実度のBIOC主導のインシデントへと移行することです。

最新問題**: 24**
SOCではアラート疲れが著しく増加しており、Tier 1アナリストは信頼性の低いアラートの調査に過度の時間を費やし、疲弊や高優先度インシデントの見逃しにつながっています。現在のSIEMはシグネチャベースのルールのみを使用しています。SOCマネージャーは、悪意のある行動や異常なアクティビティに焦点を当てることでアラートノイズを特に削減し、Tier 1アナリストを真の脅威に集中させるソリューションを導入したいと考えています。SOCワークフローに効果的に統合することで、この目的を達成するのに最適なコンポーネントまたは機能はどれですか？また、Palo Alto Networksのセキュリティエコシステムを前提とした理想的なシナリオにおいて、SOCの各階層における正当かつ信頼性の高いアラートの典型的な流れはどのようなものですか？
**A.** コンポーネント/機能: ネットワーク アクセス制御 (NAC); アラートの進行状況: NAC -> Tier 1 -> Tier 2 -> SOC マネージャー。
**B.** コンポーネント/機能: データ損失防止 (DLP); アラートの進行: DLP -> コンプライアンス アナリスト -> 法務。
**C.** コンポーネント/機能: XDR/SIEM プラットフォーム内のユーザーおよびエンティティの動作分析 (UEBA) (例: Cortex XSIAM); アラートの進行: XSIAM (AI/ML 相関) -> Tier 2 (初期検証/調査) Tier 3 (詳細な調査/封じ込め) -> インシデント対応リーダー (全体的な管理)。
**D.** コンポーネント/機能: 脆弱性管理プラットフォーム; アラートの進行状況: 脆弱性スキャン、脆弱性アナリスト -> パッチ適用チーム。
**E.** コンポーネント/機能: 従来のウイルス対策 (AV); アラートの進行状況: AV -> Tier 1 (手動レビュー) -> ユーザー (修復)。
**Answer: (**解答を表示する**)**
問題の説明では、「忠実度の低いアラート」による「アラート疲れ」と、「シグネチャベースのルール」を超えて「悪意のある行動や異常なアクティビティ」に重点を置く必要性が明示的に言及されています。コンポーネント/機能: ユーザーおよびエンティティの行動分析 (UEBA) は、異常なユーザーおよびエンティティの行動を検出するために特別に構築されており、シグネチャを超えて、内部脅威、侵害されたアカウント、またはラテラルムーブメントなどの高度な脅威を特定することで、アラートノイズを大幅に削減し、忠実度を向上させます。UEBA は、相関関係の分析に AI/ML を活用する Palo Alto Networks Cortex XSIAM などの最新の XDR/SIEM プラットフォームの中核機能です。アラートの進行: 理想的で忠実度の高いアラート (多くの場合、UEBA/XSIAM などの高度な分析によって生成) は、その固有の高い信頼性のため、通常、単純な Tier 1 トリアージをバイパスします。複雑または広範囲に及ぶインシデントの場合は、Tier 3にエスカレーションさ

れ、詳細な調査、マルウェア分析、高度な封じ込め戦略が実施されます。その後、インシデント対応責任者（インシデント全体の場合はSOCマネージャー）がインシデントのライフサイクル全体を管理し、修復を調整し、関係者と連絡を取ります。このプロセスにより、適切なスキルを持つ担当者が高忠実度のアラートを効率的に処理できるようになります。他の選択肢があまり正確でない理由：AとBは、行動異常による一般的なアラート疲労を主に解決するものではない特定のセキュリティ技術です。また、アラートの進行も単純すぎるか、方向性が間違っています。Dは、事後的なインシデント対応アラート処理ではなく、プロアクティブな脆弱性管理に関するものです。Eは非常に基本的で、多くの場合非常にノイズの多いAVアラートフローを表していますが、アラート疲労を解決するものではなく、むしろ悪化させることさえあります。

最新問題: **25**
ランサムウェア攻撃の事後調査において、インシデント対応チームは、初期アラートが生成されたにもかかわらず、重大度分類が「情報」であったため優先度が下げられていたことを突き止めました。分析の結果、これらのアラートは個々には信頼性が低いものの、全体としては偵察フェーズとそれに続く重要サーバーへの認証情報アクセスを指し示していたことが明らかになりました。同様の見落としを防ぐには、インシデントの分類と優先度付けのフレームワークをどのように調整するのが最も効果的でしょうか？
**A.** コンテキストに関係なく、24 時間後に「情報」レベルのアラートを「低」の重大度にエスカレートする自動システムを実装します。
**B.** 生成後 1 時間以内に、Tier 1 SOC アナリストによるすべての「情報」重大度アラートの手動レビューを義務付けます。
**C.** SIEM (Splunk、QRadar など) または SOAR (XSOAR など) で相関ルールを開発し、高価値資産を対象とする関連する低重大度イベントのシーケンスに基づいてインシデントの重大度を高めます。
**D.** すべてのネットワークベースのアラートのしきい値を 50% 増やして、誤検知を減らし、重大度の高いアラートのみに焦点を当てます。
**E.** 初期検出の信頼度レベルに関係なく、重要なサーバーに関連するすべてのアラートをデフォルトで重大度「高」に分類します。
**Answer: C (**メーッセージを残す**)**
ここで説明されている根本的な問題は、個々のイベントが低忠実度で構成されている、ローアンドスローな攻撃チェーンを認識できないことです。SIEMまたはSOARに相関ルール（オプションC）を実装することが最も効果的な解決策です。これにより、システムは一見無害に見える複数のイベントを順番に分析し、攻撃を示唆するパターン（例：偵察活動続いて重要な資産への認証情報アクセス）を特定し、集約されたインシデントの重大度と優先度を自動的に引き上げることができます。オプションAとBは非効率的、または事後対応的です。オプションDは、真の脅威を見逃すリスクがあります。オプションEは、深刻なアラート疲労と誤検知につながり、アナリストの負担を増大させる可能性があります。

最新問題: **26**

広く使用されているウェブアプリケーションの重大な脆弱性を狙ったゼロデイエクスプロイトが発表されました。プレミアム脅威インテリジェンスフィードは、特定のURLパターン、カスタムHTTPヘッダー値、およびエクスプロイトの試行に関連付けられた固有のユーザーエージェント文字列を含む、侵害指標（IOC）を即座に提供します。貴社では、Palo Alto NetworksのWildFireとThreat Preventionを使用しています。WildFireまたはThreat Preventionのシグネチャが完全に展開される前に、このエクスプロイトをプロアクティブに防止・検出するには、カスタム脅威インテリジェンスを活用したPalo Alto Networksのファイアウォール構成のどの組み合わせが最も効果的でしょうか？

**A.** 特定の URL パターンをブロックするようにカスタム URL フィルタリング プロファイルを構成し、それを適用するためのセキュリティ ポリシーを作成します。

**B.** カスタム HTTP ヘッダー用のカスタム スパイウェア対策シグネチャと、ユーザー エージェント文字列用のカスタム脆弱性保護シグネチャを作成します。

**C.** URL パターンと HTTP ヘッダーを照合する正規表現と、ユーザー エージェント文字列のカスタム アプリケーション オーバーライドを使用して、カスタム脅威防止シグネチャ (IPS) を実装します。

**D.** URL パターンのカスタム外部動的リスト (EDL) を開発し、ユーザーエージェント文字列のカスタム IPS 署名を展開します。

**E.** データ フィルタリング プロファイルを使用してカスタム HTTP ヘッダーをブロックし、ファイル ブロッキング プロファイルを使用して悪意のある URL からのダウンロードを防止します。

**Answer: (**解答を表示する**)**

このシナリオでは、カスタム脅威インテリジェンスを用いたゼロデイ攻撃に対するプロアクティブな防御を重視しています。オプションCは、Palo Alto Networksにとって最も包括的かつ効果的なアプローチです。

正規表現を使用したカスタム脅威対策シグネチャ（IPS）これは、ベンダーのシグネチャでカバーされていないトラフィックパターン（URLパターンやHTTPヘッダーなど）をプロアクティブに検出してブロックするための最も強力な方法です。正規表現は、複雑なパターンを柔軟にマッチングします。

ユーザーエージェントのカスタム アプリケーション オーバーライド: 防止には直接的ではありませんが、他の方法が適用できない場合や追加レイヤーとして使用できない場合に、特定の悪意のあるユーザーエージェントによるトラフィックを分類してブロックするのに役立ちます。

他の人がなぜそれほど効果的でないのかを分析してみましょう。

' A (カスタム URL フィルタリング): URL には適していますが、カスタム HTTP ヘッダーまたはユーザーエージェントを包括的に処理しません。

B（カスタムスパイウェア対策/脆弱性保護）：一般的な HTTP 要素またはユーザーエージェントに対して特定のスパイウェア対策または脆弱性保護シグネチャを作成することは可能ですが、エクスプロイトパターン自体のカスタム IPS シグネチャよりも精度や効率が低くなる可能性があります。IPS はエクスプロイトの検出を目的として設計されています。

（URL用EDL、ユーザーエージェント用カスタムIPS）EDLはIP/ドメインのブロッキングには適していますが、URLパターンの細分性は低くなります。ユーザーエージェント用のカスタムIPSは可能ですが、すべてのIOCを単一のIPSシグネチャに統合する方が効率的です。
E (データフィルタリング/ファイルブロッキング): データフィルタリングは、HTTPヘッダーを介したエクスプロイトの試みではなく、機密データの流出を対象とします。ファイルブロッキングは、エクスプロイトのパターンではなく、ファイルの種類を対象とします。

最新問題: **27**
Cortex XDR の次の XQL クエリについて考えてみましょう。WildFire のコンテキストにおけるこのクエリの主な目的は何ですか。また、その結果からどのような種類の脅威インテリジェンスを導き出すことができますか。（該当するものをすべて選択してください。）
**A.** Cortex XDR エージェントによって WildFire に送信され、最終的に 「悪意のある」または 「フィッシング」と判断されたすべてのファイルを識別します。これは、WildFire のクラウド分析による初期検出が成功したことを示します。
**B.** WildFire の判定を特定のエンドポイント アクション (プロセス実行、ネットワーク接続など) と相関させて、検出された脅威の攻撃チェーン全体を把握します。
**C.** WildFire のクラウド判定に依存せずに、ローカル シグネチャの一致に基づいて Cortex XDR のマルウェア対策エンジンによってブロックされたすべてのファイルを一覧表示します。
**D.** WildFire が当初 「グレーウェア」として分類したが、その後の動的分析やコミュニティからのフィードバックによって悪意のある動作を示したポリモーフィック マルウェアの亜種を検出します。
**E.** 環境から WildFire に送信される特定のファイル タイプの普及状況を追跡し、プロアクティブなポリシー調整や標的を絞った脅威ハンティングを可能にします。
**Answer: A,B (**メーッセージを残す**)**
この質問では、XQL が WildFire データとどのように統合されるかを理解する必要があります。WildFire を含む一般的な XQL クエリは、ファイルなどのテーブルを、WildFire の送信や判定に関連する情報と結合します。オプション A: プロセス内の WildFire の判定に重点を置いたクエリは、この目的に直接役立ち、成功した WildFire 検出を識別します。オプション B: WildFire の判定データを ( 「悪意のある」 「フィッシング」) プロセス実行、ネットワーク接続、またはファイル書き込みイベント (XQL で一般的) と結合することにより、アナリストはキル チェーンを再構築し、悪意のあるファイルの実行内容を理解し、影響を受けたエンドポイントを識別できます。これは、インシデント対応と脅威ハンティングにとって重要です。オプション C: このクエリはローカルのマルウェア対策に関するものであり、WildFire の判定とは直接関係ありません。オプション D: WildFire は再分類できますが、この特定のクエリ タイプは、グレーウェアとして始まり後で変化した 「ポリモーフィック型亜種」を識別するのには直接的ではありません。オプション E: XQL でも可能ですが、送信タイプと件数のクエリが必要になります。これは、WildFire のコア機能によって暗示される 「悪意のある」または 「フィッシング」判定の焦点に直接関連する主な目的ではなく、XQL 分析のより広範な使用例です。

最新問題: **28**
ある組織はWildFireと連携したCortex XDRを導入しており、厳格なデータレジデンシー要件を遵守しています。そのため、一部の機密ファイルはオンプレミスネットワークからクラウド分析に持ち出すことができません。しかし、これらのファイルにはWildFireの高度な脅威分析機能が必要です。WildFireとCortex XDRを併用することで、この要件をどのように満たすことができるでしょうか。また、拡張性とメンテナンスにはどのような影響があるでしょうか。
**A.** オンプレミスに専用のWildFireアプライアンス（WF-500）を導入します。このアプライアンスはローカルで動的分析を実行し、データの保存場所を確保します。拡張性はアプライアンスの容量によって制限され、メンテナンスには組織による定期的なソフトウェア更新とハードウェア管理が必要です。
**B.** Cortex XDRエージェントをローカル分析のみ実行するように設定し、機密性の高いエンドポイントに対するWildFireの送信を無効にします。これによりデータレジデンシーは確保されますが、これらのファイルに対するWildFireの高度な分析機能が犠牲になり、新規および未知の脅威に対する脅威検出能力が大幅に低下します。
**C.** WildFireのクラウドサービスを利用しますが、送信前に機密ファイルに対してカスタムデータ暗号化スキームを実装してください。このアプローチはWildFireではサポートされておらず、カスタム暗号化ファイルを復号できないため、分析機能が損なわれる可能性があります。
**D.** 機密ファイルがWildFireに送信されないようにネットワークDLPソリューションを導入し、従来のウイルス対策のみに頼る。これによりWildFireの高度な分析が回避され、重大なセキュリティギャップが残ってしまう。
**E.** 組織の管理された環境内でホストされるWildFireのプライベートクラウドインスタンスを活用します。これにより、データの保存場所を維持しながらWildFireの完全な分析機能が提供され、スケーラビリティとメンテナンスはPalo Alto Networksのマネージドサービスによって提供されます。
**Answer: A (**メーッセージを残す**)**
オプションAは適切かつ実用的なソリューションです。ファイル分析において厳格なデータレジデンシー要件を持つ組織では、オンプレミスのWildFireアプライアンス（WF-500など）の導入が不可欠です。このアプライアンスは動的分析をローカルで実行し、機密ファイルが組織のネットワーク外に漏洩することを防ぎます。つまり、拡張性はアプライアンスのハードウェア容量に左右され、ソフトウェアアップデート、パッチ適用、ハードウェアヘルスチェックなどのメンテナンスは組織が責任を負うことになります。オプションEは、オンプレミス導入シナリオ向けにPalo Alto Networksが取り扱う「WildFireのプライベートクラウドインスタンス」として一般には提供されていない、将来的に提供される可能性のある、または特殊なサービスを指します。通常、WildFireクラウドサービスが主要なモデルとなります。

最新問題: **29**
インシデント対応演習中、セキュリティアナリストは、悪意のある添付ファイルを含むフィッシングメールがユーザーの受信トレイに配信されたことを発見しました。ユーザーはまだ添付ファイルを開いていません。NISTインシデント対応計画の「封じ込め、根絶、復旧」フェーズにおい

て、Palo Alto Networksのセキュリティ機能を具体的に活用した、最も効果的かつ適切な一連のアクションはどれでしょうか？
**A.** Cortex XDR のライブ ターミナルを使用してユーザーのエンドポイントを分離し、ネットワーク全体のウイルス対策スキャンを実行し、最後にユーザーに電子メールを削除するように通知します。
**B.** 電子メール ゲートウェイで送信者の電子メール アドレスをブロックし、ユーザーの受信トレイから電子メールを削除し (電子メール セキュリティ ソリューション経由で可能な場合)、添付ファイルの WildFire 分析を開始して脅威インテリジェンスを更新します。
**C.** ユーザーのネットワーク アクセスを無効にし、マシンのイメージを再作成してから、ユーザー認識トレーニング セッションを実施します。
**D.** ユーザーのハードドライブの完全なフォレンジック分析を実行し、攻撃者の IP を特定して、境界ファイアウォールでその IP をブロックします。
**E.** 法執行機関に事件を報告し、何らかの措置を講じる前に指示を待ってください。
**Answer: B (**メーッセージを残す**)**
「封じ込め、根絶、復旧」フェーズの目的は、拡散を阻止し、根本原因を取り除き、サービスを復旧することです。未開封の悪意のあるメールに対しては、送信者をブロックし、メールを削除すること（B）が、即時の封じ込めおよび根絶のステップとなります。WildFire分析を開始することは、脅威インテリジェンスを更新し、同様の将来の攻撃を防止し、根絶と将来的な予防と整合させるために不可欠です。エンドポイントの隔離（A）は封じ込めステップですが、この段階では侵害が確認されていないため、ネットワーク全体のスキャンは範囲が広すぎる可能性があり、ユーザーに削除を通知するよりも強制削除の方が効果的です。添付ファイルが開かれていない場合、再イメージ化（C）は過剰です。フォレンジック分析（D）は通常、差し迫った脅威が封じ込められた後の根絶/インシデント後分析の一部です。法執行機関への報告（E）はインシデント後の活動であり、即時の封じ込めステップではありません。

最新問題**: 30**
ある組織は、高度な脅威インテリジェンス管理のためにCortex XSOARを導入しています。社内APIエンドポイントから特定の脅威インテリジェンスを集約するカスタムインジケーターフィードを作成する必要があります。このAPIは独自のXML形式でデータを返します。組織はこのXMLを解析し、特定のインジケータータイプ（SHA256ハッシュ、C2ドメインなど）を抽出し、XSOARの内部インジケーターフィールドにマッピングし、XML属性に基づいて動的な信頼度スコアを割り当てて取り込む必要があります。この複雑なカスタムフィード統合を実現するには、どのようなXSOAR構成と手順が必要ですか？
**A.** 新しい「Generic API Feed」インスタンスを構成し、XML 解析用の XPath 式を含む組み込み XSOAR「Mapper」を使用し、フィード構成内で静的な信頼スコアを設定します。
**B.** 新しい「カスタムフィード」統合を作成します。「Fetch Indicators」コマンド用のカスタムPythonスクリプトを実装し、API呼び出し、XML解析、指標の抽出、マッピング、動的な信頼度スコアリングを処理します。スクリプト内で指標の種類を定義し、スクリプトが期待されるXSOAR形式で指標を返すようにします。

**C.** 既存の「脅威インテリジェンスフィード」タイプを使用し、XSOAR I-JI 経由で XML ファイルを手動でアップロードします。次に、アップロードしたファイルに対して「データ変換」プレイブックを実行し、インジケーターを抽出してマッピングします。
**D.** XMLデータを受信する「Web Hook」を設定し、XMLを解析してフィールドをマッピングする「Incoming Mapper」を作成します。「Incident Type」を使用して、受信データを脅威インテリジェンスとして分類します。
**E.** XMLを解析し、XSOAR APIを使用してデータをXSOARにプッシュするスタンドアロンの外部スクリプトを開発します。このスクリプトは「インジケータープレイブック」をトリガーし、新しいインジケーターを処理します。
**Answer: B (**メーッセージを残す**)**
オプションBは、独自のXML解析と動的な信頼度スコアリングを備えた複雑なカスタムフィードに最適な、最も適切で強力なソリューションです。「カスタムフィード」統合：これにより、フェッチロジックを完全に制御できます。「フェッチインジケーター」用のカスタムPythonスクリプト：このスクリプトには、内部エンドポイントへのAPI呼び出しを行うロジックが含まれます。固有のXML形式を解析します（例：Pythonの「xml.etree.ElementTree」を使用）。特定のインジケータータイプ（SHA256、C2ドメイン）を抽出します。それらをXSOARの「値」、「タイプ」、「有効期限」、「評判」にマッピングし、さらに重要な点として、XML属性に基づいて「スコア（信頼度）」を動的に計算して割り当てます。このレベルの動的スコアリングと解析は、通常、標準的なマッパーでは対応できません。XSOARがインジケーターに期待する形式でデータを返します。オプションAの組み込みマッパーは、動的スコアリングや非常に固有のXML構造に対応できない可能性があります。オプションCは手動取り込み用で、自動化機能がありません。オプションDは、APIエンドポイントからデータを積極的に取得するのではなく、受信用であり、インシデント作成に重点を置いています。オプションEは、XSOARのネイティブフィード管理機能をバイパスする外部ソリューションであるため、XSOAR自体との統合性が低く、管理が困難です。

最新問題**: 31**
複数のエンドポイントが侵害された大規模なセキュリティインシデントが検出されました。XSOARのインシデント対応プレイブックでは、次の作業を行う必要があります。1) EDRソリューションを使用して影響を受けたエンドポイントを隔離する。2) Jiraでアナリスト割り当て用の高優先度チケットを作成する。3) 隔離されたエンドポイントからフォレンジックアーティファクトを収集する。4) 分析中に特定された新しいIOC（侵入の痕跡）を使用して、脅威インテリジェンスプラットフォーム（TIP）を更新する。この複雑な複数システムへの自動対応を実行するために不可欠なXSOARの機能と統合機能は、次のうちどれですか？また、どのような課題が生じる可能性がありますか？
**A.** 必須機能：XSOAR組み込みEDR統合、Jira統合、脅威インテリジェンス「Push Indicators」コマンド。課題：カスタムフォレンジックアーティファクト収集タイプのサポートが限定的。
**B.** 必須：EDR用の汎用REST API統合、Jira用のメール統合、アーティファクト収集用のSFTP、TIPへの手動アップロード。課題：リアルタイム応答の欠如と手動オーバーヘッドの増大。

**C.** 必須：XSOARは、EDR（CrowdStrike、SentinelOneなど）、Jira、TIPS（Anomali、MISPなど）との連携をすぐに利用できます。フォレンジック情報収集には、EDRのAPIまたは別のフォレンジックツールのAPIを活用したカスタムPython統合が必要です。課題：APIレート制限の超過防止、統合全体での認証情報の安全な管理、部分的な障害への適切な対応。
**D.** 必須: XSOARリモートエグゼキューターからすべてのシステムへのCLIアクセス、およびすべてのアクションに対するBashスクリプト。課題: スケーラビリティの問題とスクリプトのメンテナンスの難しさ。
**E.** 必須：既存のスクリプトを埋め込むためのXSOARの「外部統合」モジュール、Jira用の「チケット管理」モジュール、TIP用の「インジケーター管理」モジュール。課題：ネットワークを分割することなく、すべての外部システムがXSOARサーバーから直接アクセスできるようにする。
**Answer: (**解答を表示する**)**
オプションCは包括的なアプローチを正確に表現しています。XSOARは、EDR、Jira、TIPSといった一般的なセキュリティツールとの豊富な連携機能を標準装備しており、迅速なアクション（隔離、チケット発行、インジケーターの共有）を可能にします。高度なフォレンジックアーティファクト収集など、標準的なEDRコマンドでは対応しきれないような非常に特殊なタスクには、EDRのAPIまたは専用のフォレンジックツールのAPIを使用したカスタムPython統合が堅牢なソリューションです。挙げられている課題（APIレート制限、認証情報管理、適切な障害処理）は、複数のシステムと連携する、回復力の高いエンタープライズグレードのXSOARプレイブックを構築する上で、非常に重要な考慮事項です。



最新問題**: 32**
インシデント対応活動中、フォレンジック調査員は、ポート53（DNS）経由でカスタムコマンドアンドコントロール（C2）プロトコルを使用する執拗な脅威アクターを発見しました。既存のSIEMログには、一般的なDNSクエリしか記録されていません。C2インフラ、エクスプロイトの開発、動機など、攻撃者のTTP（戦術 手法、手順）を包括的に把握し、将来の攻撃を積極的に阻止するためには、どのようなリソースの組み合わせが最も効果的でしょうか？
**A.** ファイル ハッシュの検索には VirusTotal、一般的な脅威の傾向についてはオープン ソース インテリジェンス ブログを参照してください。
**B.** マルウェアのデトネーションとリアルタイムのシグネチャ生成のための WildFire と、Unit 42 の広範な調査レポートおよび敵対者のプレイブックを組み合わせたもの。

**C.** C2 ドメインのパッシブ DNS 偵察と WHOIS 検索。
**D.** 脅威インテリジェンス フィードを統合せずに、商用のエンドポイント検出および対応 (EDR) ソリューションを採用する。
**E.** すべてのネットワーク トラフィックの詳細なパケット検査と、疑わしいすべてのバイナリの手動リバース エンジニアリング。
**Answer: (**解答を表示する**)**
WildFireは、C2通信を含むマルウェアの技術的側面を理解する上で非常に優れています。しかし、攻撃者のTTP（戦術 技術、手順）、動機、そしてより広範なキャンペーンを包括的に把握するには、Unit 42の詳細な脅威リサーチ、攻撃者プレイブック、そしてインテリジェンスレポートが不可欠です。Unit 42は、脅威アクター、そのキャンペーン、そしてより広範な脅威ランドスケープの詳細な分析に重点を置き、WildFireの技術的成果を補完する戦略的および戦術的インテリジェンスを提供します。この組み合わせにより、攻撃の技術的理解と攻撃者に関する戦略的インテリジェンスの両方が可能になります。

最新問題**: 33**
SOCマネージャーは、脅威検出機能の現状を見直しています。SIEMは「ポートスキャン」イベントのアラートを頻繁に生成していますが、その多くはIT運用ツールによる無害なネットワークスキャンであり、高い誤検知率につながっていることに気づきました。そこで、Palo Alto Networks SIEM（例Palo Alto Networksアドオンを搭載したSplunk）とCortex XDRを組み合わせてこれらの検出精度を向上させ、真に悪意のあるポートスキャンと関連アクティビティを特定するための行動ベースのアプローチに移行したいと考えています。
特定の機能を活用した以下の戦略のうち、どれが最も効果的でしょうか?
**A.** SIEM のすべてのデフォルトの「ポート スキャン」アラートを無効にし、Cortex XDR の「脅威防止」モジュールのみを使用して既知の悪意のあるポート スキャンをブロックします。
**B.** スキャンを実行する IT 運用ツールの IP アドレスの許可リストを NGFW の「セキュリティ ポリシー」に作成し、これらの特定の IP を無視するように SIEM を構成します。
**C.** NGFWに「User-ID」と「App-ID」を実装し、トラフィックの送信元とアプリケーションを識別します。SIEMでは、ポートスキャンイベントにUser-IDとApp-Ldのコンテキストを追加します。さらに、Cortex XDRでは「Behavioral Threat Protection」（BTP）を活用し、スキャン自体だけでなく、疑わしいネットワークイベントのシーケンス（例ポートスキャンに続いて疑わしいプロセス実行やデータアクセスパターンが続くなど）を検出します。既知の無害なITスキャナーについては、プロセスハッシュまたはデジタル署名に基づいてXDRの「除外ポリシー」を作成します。
**D.** SIEM を構成して、内部スキャンを完全に無視し、外部 IP アドレスからのポート スキャンのみを警告します。
**E.** NGFW の「脆弱性保護」プロファイルの感度を上げて、より多くの種類のポート スキャン攻撃を検出し、WildFire を使用して関連する疑わしいファイルを分析します。
**Answer: C (**メーッセージを残す**)**
このシナリオでは、シグネチャベースから動作ベースに移行して、ポートスキャンの真陽性検出を改善しながら誤検知を減らすための、洗練された多層アプローチが必要です。

1. NGFWにおけるUser-IDとApp-ID およびSIEMエンリッチメント）これはコンテキストの取得に不可欠です。User-IDはネットワークアクティビティを特定のユーザーに紐付け、App-Ldは実際のアプリケーションを識別します。これにより、SIEMは正規のITスキャンツール（例App-IDで識別され、ITユーザーがUser-ID経由で実行するNessus）と悪意のあるスキャンを区別できます。このコンテキストでSIEMアラートをエンリッチメントすることは、分析にとって不可欠です。
2. Cortex XDR 行動ベースの脅威防御 (BTP): これは行動ベースのアプローチの中核です。ポートスキャンを単にフラグ付けするのではなく、BTPは一連のイベントに注目します。単独のポートスキャンは無害である可能性がありますが、ポートスキャンの後に疑わしいログイン、プロセス実行、またはデータアクセスパターンが続く場合は、悪意のある意図が強く示唆されます。これは、 環境寄生型」攻撃の特定に役立ちます。
3. XDR 除外ポリシー: 既知の正当な IT 運用ツール (脆弱性スキャナー、ネットワーク インベントリ ツールなど) の場合、信頼性の高い識別子 (プロセス ハッシュ、デジタル署名) に基づいて Cortex XDR で特定の除外を作成すると、これらのツールによる BTP アラートのトリガーが防止され、誤検知が大幅に減少します。
他のオプションを分析してみましょう:
A: すべてのアラートを無効にするのは無謀です。 脅威防止」だけに頼るのは、行動検知としては単純すぎるからです。
B: 許可リストの作成はノイズを減らすための一般的な方法ですが、静的IPアドレスに依存しており、高度な脅威の行動的側面には対応していません。これは良いステップではありますが、包括的な行動ベースのアプローチとしては最も効果的ではありません。
D: 内部での横方向の移動は一般的な攻撃ベクトルであるため、すべての内部スキャンを無視すると、重大なセキュリティギャップが生じます。
E: 脆弱性保護」の感度を上げると、誤検知が増える可能性があります。WildFireはファイル分析を目的としており、ポートスキャン検出の精度向上やネットワークアクティビティの挙動分析を直接目的としたものではありません。

最新問題**: 34**
インシデント対応活動において、セキュリティチームは侵害を受けたエンドポイントがDNSトンネリングを介してデータを盗み出そうとしていることを特定しました。この手法は、従来のシグネチャでは検出が困難な場合が多くあります。Cortex XSIAMの機能、特にデータの取り込み、処理、ルール適用といったアプローチが、この高度な攻撃の検出と調査をどのように促進するのか、そしてスタンドアロンのDNSファイアウォールよりも効果的な理由についてご説明ください。
**A.** XSIAM は境界ネットワークトラフィックのみを監視し、既知の DNS トンネリングツールに対してシグネチャベースの IOC を適用します。スタンドアロンの DNS ファイアウォールは、内部 DNS 異常の検出により優れています。
**B.** XSIAM は DNS トンネリングの検出に脅威インテリジェンスフィードのみを使用し、ブラックリストに登録された IP の IOC を作成します。スタンドアロンの DNS ファイアウォールも、最新の脅威フィードがあれば同様に効果的です。

**C.** XSIAM はファイアウォールから DNS クエリログのみを取り込み、既知の悪意のあるドメインに対して基本的な IOC ルールを適用します。スタンドアロンの DNS ファイアウォールは、ネットワークエッジでトラフィックをブロックできるため、より優れています。
**D.** XSIAMの主な機能は、疑わしいクエリに対するDNS解決をプロアクティブに阻止し、ルール適用を不要にすることです。スタンドアロンのDNSファイアウォールも同様のプロアクティブブロック機能を提供します。
**E.** XSIAMは、DNSクエリデータ、エンドポイントプロセスアクティビティ（DNSリクエストを発行するプロセスなど）、およびネットワークフローデータを統合します。BIOC（バイオメトリックオシレーション）を用いて、疑わしいプロセス挙動と相関する異常なDNSクエリパターン（高ボリューム、異常なクエリ長、特定のドメイン構造など）を特定します。この統合ビューにより、スタンドアロンのDNSファイアウォールとは異なり、XSIAMは攻撃チェーン全体を検出し、調査のための包括的なコンテキストを提供できます。
**Answer: E (**メーッセージを残す**)**
DNSトンネリングの検出には、DNSクエリを個別に検査するだけでは不十分です。Cortex XSIAMの強みは、複数のソース（エンドポイント、ネットワーク、アイデンティティ、クラウド、DNSログ）からデータを取り込み、正規化する機能にあります。DNSトンネリングの場合、XSIAMは異常なDNSクエリパターン（DNSログのBIOCで検出）と、それらのクエリを実行したエンドポイント上の特定のプロセス（EDRデータから取得）を相関させます。スタンドアロンのDNSファイアウォールは、既知の不正ドメインをブロックしたり、基本的なレート制限を適用したりすることはできますが、エンドポイントのプロセスとユーザーアクティビティのコンテキストを把握することができません。XSIAMの相関エンジンは、これらの異なるイベントを単一のインシデントに結び付け、プロセス実行からデータ漏洩までの攻撃チェーン全体を表示することで、調査と対応のためのより豊富なコンテキストを提供します。この包括的なアプローチは、SIEMの代替としてXSIAMが持つ重要な差別化要因です。

最新問題**: 35**
脅威インテリジェンスチームは、新たな難読化手法を用いて特定の業界セクターを標的とする新たなAPTグループに関するレポートを作成しました。このレポートには、IOC（侵入の痕跡）とTTP（戦術 手法、手順）が含まれています。このインテリジェンスを組織のインシデント分類および優先順位付けプロセスにどのように統合すれば、その効果を最大化できるでしょうか？
**A.** IOC はファイアウォールで直ちにブロックし、TTP を静的なインシデント分類マトリックスに追加する必要があります。
**B.** IOC を使用して、「重大」な重大度を持つ新しい検出ルールを作成する必要があります。また、TTP は、関連する動作の異常を識別し、これらの TTP に一致するインシデントに高い優先度を動的に割り当てるためのプレイブックとアナリストのトレーニングに通知する必要があります。
**C.** レポートはすべての IT スタッフに配布して認識してもらう必要があり、IOC に一致するアラートは毎日手動で確認する必要があります。

**D.** IOC のみをウォッチリストとして SIEM に取り込む必要があり、TTP は直接優先順位付けするには抽象的すぎるため無視する必要があります。
**E.** このインテリジェンスは主に遡及的なハンティング演習に使用する必要があり、リアルタイムの分類に直接統合されるべきではありません。
**Answer: (**解答を表示する**)**
脅威インテリジェンスを効果的に統合するには、IOC（Independent Operation Control）とTTP（Touch Control）の両方を活用する必要があります。IOC（ハッシュ、IP、ドメインなど）は、特定の高精度な検出ルール（オプションB）を作成するのに最適です。これらのルールには、既知の脅威アクターに基づいて高い重大度が自動的に割り当てられます。TTPは行動パターンであり、IOCの一致だけでなく、インシデントの分類と優先順位付けを通知および改善するために不可欠です。APTグループのTTPを理解することで、セキュリティチームは次のことが可能になります。1) SIEM/EDRでより洗練された検出ロジックを作成する、2) これらのTTPに一致するイベントの組み合わせを探すためにXSOARプレイブックを開発または修正する、3) これらの行動を認識できるようにアナリストをトレーニングし、明示的なIOCが存在しない場合でも、これらの特性を示すインシデントに動的に高い優先度を割り当てることができるようにすること。この包括的なアプローチにより、検出および対応能力が大幅に向上します。

最新問題**: 36**
セキュリティオペレーションセンター（SOC）のアナリストは、Cortex XSIAMによって検出された高度な多段階攻撃を調査しています。この攻撃は、フィッシングメールによる初期アクセス、PowerShellスクリプトを用いたラテラルムーブメント、そして最終的には異常なネットワークプロトコルによるデータ窃取で構成されています。Cortex XSIAMのログスティッチング機能が、アナリストが攻撃の全容を把握する上で主にどのような役割を果たしているかを最もよく説明しているのは次のうちどれですか？
**A.** 個々のログ エントリに対してリアルタイムの脅威インテリジェンス検索を実行し、既知の悪意のある指標にフラグを立てます。
**B.** さまざまなソースからの同一のログ エントリを集約して、ノイズを削減し、ストレージ効率を向上させます。
**C.** 共通の識別子または時間的な近接性を使用して、さまざまなデータ ソース (エンドポイント、ネットワーク、クラウドなど) にわたるさまざまなログ イベントを相関させ、アクションのシーケンスを再構築します。
**D.** 指定された期間内に観測された最も頻繁に発生するログの種類に基づいて、カスタム ダッシュボードとレポートを自動的に生成します。
**E.** 個々のログ ソースに割り当てられた定義済みの重大度レベルに基づいてアラートを優先順位付けします。
**Answer: C (**メーッセージを残す**)**
Cortex XSIAM のログスティッチングは、複雑な攻撃を理解する上で極めて重要です。様々なソース（例PowerShell 実行を示すエンドポイントログ、異常な出力を示すネットワークログ、フィッシングメールの配信を示すメールログ）からの異なるログエントリを取得し、共通の識別子（プロ

セス ID、ホスト名、IP アドレス、ユーザーアカウントなど）または時間的な近接性に基づいてそれらを相関させます。この相関関係により、XSIAM は一貫性のあるナラティブをつなぎ合わせ、個別のイベントを一連のアクションに変換します。これは、多段階攻撃の全容とタイムラインを理解する上で不可欠です。

最新問題: **37**
広く使用されているWebサーバーに影響を与える重大なゼロデイ脆弱性が公開されました。パッチが公開される前に、CISOはCortex XSIAMであらゆるエクスプロイトの試みを積極的に調査することを義務付けています。このエクスプロイトには特定のHTTPリクエストヘッダーと特定のユーザーエージェント文字列が関係していることが分かっています。Webトラフィックログが大量に生成されるため、効率的なクエリが不可欠です。Webサーバーのアクセスログが取り込まれ、「http」データセットにマッピングされていると仮定した場合、このシナリオにおいてCortex XSIAMで最も高度でパフォーマンスの高いハンティング手法を示すXQLクエリとアプローチはどれですか？

**A.**
```
dataset = http | filter http_method = 'POST' and http_uri contains '/vulnerable/endpoint' and http_user_agent = 'MaliciousBot/1.0' | limit 1000
```

**B.**
```
dataset = http | filter http_user_agent = 'MaliciousBot/1.0' and (http_uri contains '/vulnerable/endpoint' or http_headers contains 'X-Exploit-Header
: Payload') | join (dataset = endpoint_processes | filter process_name = 'cmd.exe') on src_ip | sort by _time desc
```

**C.**
```
dataset = http | filter http_user_agent = 'MaliciousBot/1.0' and http_uri = '/vulnerable/endpoint' and (http_headers = 'X-Custom-Header: value' or
http_headers = 'Another-Header: exploit') | project _time, src_ip, http_method, http_uri, http_user_agent, http_headers
```

**D.**
```
dataset = http | filter _time > now() - duration('1h') | filter http_user_agent = 'MaliciousBot/1.0' and http_method = 'POST' and http_uri_path =
'/vulnerable/endpoint' and http_headers like '%X-Exploit-Header: %' and http_headers like '%Content-Type: application/x-vuln-exploit%' |
map remote_ip = src_ip, request_uri = http_uri | sort by _time desc
```

**E.**
```
dataset = http | filter _time > now() - duration('30m') | filter http_user_agent = 'MaliciousBot/1.0' and http_method = 'POST' and http_uri_path =
'/vulnerable/endpoint' and http_request_headers_raw contains 'X-Exploit-Header: ' and http_request_headers_raw contains 'Content-Type:
application/x-vuln-exploit' | group count() by src_ip, dest_ip, http_user_agent | sort by count() desc
```

**Answer: (**解答を表示する**)**
オプションDは、最もパフォーマンスが高く、精度の高い探索手法です。クエリの先頭で「time > now() -」を使用すると、強力なプレフィルターとして機能し、後続のフィルターで処理されるデータセットを大幅に削減できます。「http_uri_path」を使用する方が、「http_uri contains」を使用するよりも具体的です。重要なのは、「like」を特定のヘッダーコンテンツに使用する方が、「http_headers contains 'string'」を使用するよりも堅牢であるということです。これは、「http_headers」は多くの場合、すべてのヘッダーを連結した単一の文字列であり、「like」は部分文字列の一致に最適化されているためです。「map」演算子を使用すると、基になるデータを変更することなく、結果を明確にするためにフィールド名を変更できます。オプション E も同様のフィルタリングを試みますが、「http_request_headers_raw」は取り込まれたすべての Web サーバー ログの標準フィールド名ではない可能性があり、「contains」は潜在的に大きな文字列の部分一致において「like」よりもパフォーマンスが低くなる可能性があります。オプション A、B、C は、フィルタリング ロジック、フィールド名、またはパフォーマンス上の考慮事項（事前フィルタリングの時間が不足している、または「join」を不必要に使用しているなど）に関して、それほど洗練されていません。

最新問題: **38**
ランサムウェアの新しい亜種が、クライアントのエンドポイントで従来のシグネチャベースのウイルス対策を回避しました。しかし、Cortex XDRは重要なファイルの暗号化を阻止し、エンドポイントを隔離することに成功しました。調査の結果、ランサムウェアはシャドウコピーの列挙、ボリュームシャドウコピーの削除、そして特定の拡張子を持つファイルの暗号化を試みていたことが判明しました。このゼロデイランサムウェア攻撃を特定し、阻止する上で、Cortex XDRの主要な行動分析機能のうち、どの2つが最も重要でしたか？
**A.** 脅威インテリジェンスクラウドとWildFire分析
**B.** 行動脅威保護（BTP）およびランサムウェア保護モジュール
**C.** IOCマッチングとカスタム検出ルール
**D.** エンドポイントデータ損失防止 (DLP) とファイルアクセス制御
**E.** ネットワークパケットキャプチャとディープパケットインスペクション
**Answer: B (**メーッセージを残す**)**
Cortex XDRのBehavioral Threat Protection（BTP）は、一連のアクションを分析することで、悪意のある行動を検知・防止するように設計されています。ここで説明するアクション（シャドウコピーの列挙、ボリュームシャドウコピーの削除、ファイルの暗号化）は、BTPが脅威チェーンとして識別するランサムウェアの特徴的な行動です。Cortex XDRのRansomware Protection Moduleは、ファイルシステムのアクティビティとプロセスの動作を監視し、ランサムウェアに類似したパターンを検出することで、こうしたタイプの暗号化ベースの攻撃を特に標的として防御します。Threat IntelligenceとWildFireは、一般的な脅威分析とサンドボックス化には重要ですが、BTPやRansomware Protection Moduleのようなリアルタイムの行動ベースの攻撃に対する、主要な直接的な防御メカニズムではありません。

最新問題: **39**
組織では、セキュリティ運用にCortex XSIAMを使用しています。新たなゼロデイエクスプロイトが出現し、緊急パッチがリリースされました。パッチを適用する前に、SOCチームは、脆弱なプロセスを実行している可能性のあるシステムや、エクスプロイトを示唆する不審な動作を示すシステムを特定し、すべてのLinuxサーバーに対する差し迫ったリスクを迅速に評価する必要があります。このリスクは重大であるため、誤検知を最小限に抑え、高い効率で評価を行う必要があります。このタスクには、以下のXSIAMプロセスと機能のうちどれを活用すべきですか？また、その理由も教えてください。
**A.** 開始する
**B.** XSIAMの
**C.** 各 Linux サーバーに手動でログインし、ネイティブ Linux コマンドを使用して実行中のプロセスとネットワーク接続を確認します。
**D.** XSIAMの
**E.** 確認する
**Answer: B (**メーッセージを残す**)**

このシナリオでは、ゼロデイ攻撃を迅速かつ的確に、かつ正確に評価する必要があります。オプションBは、XSIAMの高度な機能を活用した最も効果的なソリューションです。リアルタイムデータレイクとターゲットを絞ったXQLクエリを組み合わせることで、特定の指標や行動について、過去および現在のテレメトリを即座に検索できます。カスタムのBehavioral Threat Protectionルールを導入することで、正確なエクスプロイトが不明な場合でも、エクスプロイト後の影響を監視できます。これにより、広範囲のスキャンと比較して誤検知が最小限に抑えられ、大規模環境では非常に効率的です。オプションAでは、従来のAVエンジンでゼロデイ攻撃を検出できる可能性は低いです。オプションCはスケールアウトの面で非現実的です。オプションDは、UBAがプロセスやネットワークの異常ではなく、ユーザーに焦点を当てているため、対象範囲が狭すぎます。オプションEは、アクティブなエクスプロイトの検出ではなく、クラウドの設定ミスを対象としています。

最新問題**: 40**

SOCマネージャーは、過去30日間の特定のマルウェアファミリーにおける「ブロック」イベントと「アラートは発動したがブロックはされなかった」イベントの数を追跡することで、Cortex XDRのEDRポリシーの有効性を監視したいと考えています。また、「アラートは発動したがブロックはされなかった」イベントの数が最も多い上位5つのエンドポイントを特定する必要があります。この目的に最適なXDRクエリ言語（XQL）とダッシュボード可視化技術の組み合わせはどれでしょうか？

**A.** ブロックされたイベントの XQL: 'dataset = xdr_data | filter event_type = ENUM.MALWARE and action_status = ENUM.BLOCKED | group by malware_name, endpoint_name | アラートされたイベントの XQL: 'dataset = xdr_data | filter event_type = ENUM.MALWARE and action_status = ENUM.ALERTED | group by malware_name, endpoint_name | count()'

**B.** XQL:

```
dataset = xdr_data
| filter event_type = ENUM.MALWARE and (action_status = ENUM.BLOCKED or action_status = ENUM.ALERTED)
| alter status = case(
    action_status = ENUM.BLOCKED, 'Blocked',
    action_status = ENUM.ALERTED, 'Alerted but Not Blocked',
    'Other'
  )
| stats count() as event_count by malware_name, status
| sort event_count d
| join (dataset = xdr_data | filter event_type = ENUM.MALWARE and action_status = ENUM.ALERTED | stats count() as alerted_count by endpoint_name | sort alerted_count desc | limit 5) on 1=1
```

**C.** XQL:

```
dataset = xdr_data
 filter event_type = ENUM.MALWARE and (action_status = ENUM.BLOCKED or action_status = ENUM.ALERTED)
 alter status = case(
   action_status = ENUM.BLOCKED, 'Blocked',
   action_status = ENUM.ALERTED, 'Alerted but Not Blocked',
   'Other'
 )
 stats count() as event_count by malware_name, status
 pivot event_count by malware_name, status
 union (dataset = xdr_data | filter event_type = ENUM.MALWARE and action_status = ENUM.ALERTED | stats count() as alerted_count by endpoint_name | sort alerted_count desc | limit 5
```

**D.** XQL:

```
dataset = xdr_data
 filter event_type = ENUM.MALWARE and (action_status = ENUM.BLOCKED or action_status = ENUM.ALERTED
 alter status = case(
   action_status = ENUM.BLOCKED, 'Blocked',
   action_status = ENUM.ALERTED, 'Alerted but Not Blocked',
   'Other'
 )
 stats count() as event_count by malware_name, status
 as_list(endpoint_name) as endpoints_affected
 union (
   dataset = xdr_data
   | filter event_type = ENUM.MALWARE and action_status = ENUM.ALERTED
   | stats count() as alerted_count by endpoint_name
   | sort alerted_count desc
   | limit 5
 )
```

**E.** XQL:

```
dataset = xdr_data
| filter event_type = ENUM.MALWARE and (action_status = ENUM.BLOCKED or action_status = ENUM.ALERTED)
| alter classification = case(
    action_status = ENUM.BLOCKED, 'Blocked Events',
    action_status = ENUM.ALERTED, 'Alerted but Not Blocked Events',
    'Other'
  )
| stats count() as total_events by classification, malware_name
| sort total_events desc
| join type=left (dataset = xdr_data
    | filter event_type = ENUM.MALWARE and action_status = ENUM.ALERTED
    | stats count() as alerted_events_count by endpoint_name
    | sort alerted_events_count
    | limit 5) on 1=1
```



**Answer:** (解答を表示する)

オプションEは、要件の両方の部分に対して、最も包括的で正しく構造化されたXQLと適切な視覚化を提供します。「alter classifications」ステートメントはイベントを正しく分類します。「stats count() as total_events by classification, malware_name」は積み上げ棒グラフのデータを生成します。「join type=leff」と上位5つのアラートエンドポイントのサブクエリは、プライマリイベント数をマージすることなくエンドポイントデータを取り込む最も効率的な方法です。積み上げ棒グラフは、マルウェアファミリーごとにブロックされた数とアラートされた数を表示するのに最適で、テーブルウィジェットは上位5つのエンドポイントとそれぞれのアラートされたイベント数を一覧表示するのに最適です。

最新問題: **41**

レッドチームの演習中、ペネトレーションテスターは、Living-Off-The-Landバイナリ（LoLBin）とポリモーフィック型マルウェアを用いて初期検知を回避することに成功しました。アクティビティには、rund1132.exeによる悪意のあるDLLの実行、certutil.exeによるデータダウンロード、そしてschtasks.exeによる永続性確立が含まれます。いずれのアクティビティも、高重大度アラート

をトリガーしません。Cortex XDRにおけるログスティッチングと分析の原則のうち、この攻撃チェーンを統合インシデントとして特定する上で最も役立つものはどれでしょうか？
**A.** 既知の悪意のあるハッシュに対する厳密な署名照合と即時ブロック。
**B.** 個々のプロセスの異常のみに焦点を当てた、分離されたエンドポイント動作分析。
**C.** エンドポイントとネットワーク全体で通常のベースライン動作からの逸脱を識別する動作分析および機械学習モデル。これらのモデルは、ログ ステッチングにフィードされ、親子関係、コマンド ライン引数、共有ホスト/ユーザー コンテキストに基づいて、これらの異常だが個別には重大度の低いイベントを接続し、包括的なインシデントを作成します。
**D.** ネットワーク トラフィックのディープ パケット インスペクションにより、ネットワーク上のポリモーフィック型マルウェアを識別します。
**E.** さまざまなセキュリティ ツールからの個々のログを確認した後、人間のアナリストがイベントを手動で相関させます。
**Answer: C (**メーッセージを残す**)**
LoLBin とポリモーフィック型マルウェアは、シグネチャベースの検出 (A) を回避するように設計されており、単独で見ると通常のシステム アクティビティのように見えることがよくあります (B)。手動による相関関係の特定 (E) は非効率的で、大規模な場合には人的エラーが発生しやすくなります。ディープ パケット インスペクション (D) は有用ですが、エンドポイント レベルの実行チェーン全体を捕捉することはできません。このような高度な攻撃に対抗できる Cortex XDR のログ ステッチングの強みは、高度な行動分析および ML (C) との統合にあります。これらのエンジンは、微妙で異常な行動 (例: rund1132.exe の異常な動作、certutil.exe が疑わしい URL からダウンロードしている、schtasks.exe が異常なタスクを作成しているなど) を特定します。ログ ステッチングは、これらの 「点」を因果関係 (例: rund1132 が certutil につながり、さらに schtasks につながる)、共通のホスト/ユーザー コンテキスト、および時間的な近接性に基づいて接続し、攻撃の全容を明らかにする単一の高精度なインシデントを生成します。これは、 「環境寄生型」攻撃を検出するための基本です。

最新問題**: 42**
Cortex XDRのユーザーおよびエンティティ行動分析（UEBA）によって、ある著名な役員のワークステーションで不審なアクティビティが検出されました。アクティビティには、1) ユーザーにとって通常とは異なる位置情報からのログイン、2) ユーザーがほとんどアクセスしないSharePointサイト上の機密ファイルへのアクセス、3) 個人用クラウドストレージサービスへの大量データのダウンロード試行が含まれます。直接的なマルウェアアラートはトリガーされませんでした。Cortex XDRのUEBAコンポーネントがこれらの異なる 「関心イベント」を統合して高精度アラートを生成する仕組みを正確に説明している記述は次のうちどれですか？また、その基本原理は何ですか？
**A.** UEBA は、事前定義されたルール エンジンを使用して、結合されたアクティビティが 「侵害されたアカウント」シグネチャと一致するかどうかを確認します。

**B.** UEBA は、教師なし機械学習を使用して、さまざまなデータ ソースにわたるユーザーの通常の動作のベースラインを確立し、学習したベースラインからの逸脱を異常としてフラグ付けし、コンテキストと重大度に基づいてリスク スコアをエスカレートします。
**C.** UEBA は主に脅威インテリジェンス フィードに依存して、地理的位置または SharePoint サイトの URL が既知の悪意のある指標であるかどうかを識別します。
**D.** UEBA は、すべてのネットワーク トラフィックに対してディープ パケット インスペクションを実行し、データの流出に関連する暗号化されたコマンドと制御チャネルを識別します。
**E.** UEBA では、価値の高いユーザーの「ウォッチリスト」を手動で構成する必要があり、これらのアクティビティはウォッチリストの基準と照合されます。
**Answer: B (**メーッセージを残す**)**
Cortex XDRのUEBA機能は、基本的に機械学習、特に教師なし学習によって駆動され、ユーザーとエンティティの行動の動的なベースラインを構築します。特定のユーザーにとって「正常」な行動（ログインパターン、アクセスしたリソース、データ転送習慣など）をプロファイリングします。観測されたアクティビティ（通常とは異なる位置情報、ほとんど使用されない機密ファイルへのアクセス、パーソナルクラウドへのデータの持ち出しなど）がこの確立されたベースラインから大きく逸脱した場合、それらは異常として識別されます。その後、システムはこれらの個々の異常を相関させ、リスクスコアを集計し、コンテキスト化することで、潜在的なアカウント侵害や内部脅威に関する高精度なアラートを生成します。このアプローチは、動的な環境に適応し、特定の攻撃パターンに関する事前知識なしに新たな脅威を検出するため、静的ルールや脅威インテリジェンスのみを使用するよりも優れています。

最新問題**: 43**
レッドチーム演習中、攻撃者は一般的なブラウザのゼロデイ脆弱性を悪用して組織のEDRを迂回することに成功し、その後、未公開の手法を用いてプロセスホロウイングを実行し、正規のシステムプロセスにシェルコードを挿入しました。EDRは既知のシグネチャと一般的な行動パターンに依存していたため、この非常に回避性の高い攻撃を見逃してしまいました。「Prevention First」アプローチの一環として、Cortex XDRの検出エンジンのどの特性が、事前のシグネチャがなくても、このような高度で回避性の高い脅威を検知・防御できた可能性が高いでしょうか？
**A.** 既知の悪意のあるファイル ハッシュの、常に更新される脅威インテリジェンス フィードに依存します。
**B.** 行動に基づく脅威保護、機械学習、静的分析などの AI 駆動型分析の複数レイヤーを活用し、その固有の特性と異常な動作に基づいて、これまでに見たことのない脅威を検出します。
**C.** 実行前にすべての疑わしいファイルを隔離し、クラウド サンドボックスに送信して分析する機能。
**D.** セキュリティ チームによって作成された事前定義された YARA ルールに一致する脅威のみを検出します。
**E.** コンプライアンス目的で、すべてのユーザーのログインとログアウトの詳細なログ監査を提供します。
**Answer: B (**メーッセージを残す**)**

このシナリオは、一般的なEDRを回避するように設計された、高度に回避性の高いゼロデイ攻撃について説明しています。Cortex XDRの「Prevention First（予防優先）」アプローチは、シグネチャや一般的な行動パターンにとどまりません。オプションBは、その多層的なAI駆動型検出エンジンを正確に説明しています。Behavioral Threat Protection（BTP）は、特定のマルウェアが不明な場合でも、異常なプロセス動作（プロセスホロウイングやインジェクションなど）を特定します。機械学習は、ファイルの特性（静的分析）と実行動作を分析し、シグネチャに依存せずにポリモーフィック型マルウェアやカスタムマルウェアを検出します。この組み合わせは、既知の指標に依存することが多い標準的なEDRでは見逃されがちな、高度で回避的な脅威を検出するように設計されています。

最新問題: **44**
ある大企業では、Cortex XSIAM と直接連携する API がない、カスタムビルドの特権アクセス管理 (PAM) ソリューションを使用しています。セキュリティチームは、XSIAM が侵害されたアカウントからの不審なログイン試行を検知した場合、特権認証情報の一時的な失効を自動化したいと考えています。そのためには、PAM システムの Web UI と連携するための Python スクリプトが必要です。直接連携する API がないことを考慮すると、Cortex XSIAM 内でこの自動化をどのように設計すればよいでしょうか？
**A.** PAM ソリューションがオプションとして自動的に表示されることを前提として、PAM 統合用に事前に構築された XSIAM アクションを利用します。
**B.** XSIAM でアラートを手動で作成し、別のサーバーで Python スクリプトを手動で実行して、PAM UI と対話します。
**C.** プレイブックの一部としてカスタムXSIAMアクションを開発します。
**D.** XSIAM インシデント データを CSV にエクスポートし、手動で処理してから、別の自動化ツールを使用して PAM システムと対話します。
**E.** カスタム PAM による自動修復を試みずに、XSIAM のネイティブ検出機能のみに依存します。
**Answer: (**解答を表示する**)**
オプションCは、この複雑なシナリオに対する最も洗練された適切なアプローチです。直接APIが利用できない場合、Cortex XSIAMのPlaybookフレームワーク内の「コンテナ化されたアプリ/パック」を使用すると、制御された環境でカスタムコード（Pythonスクリプトなど）を実行できます。このスクリプトは、ブラウザ自動化ライブラリ（Seleniumなど）を活用して、従来のPAMシステムのWeb UIとやり取りすることで、統合ギャップを効果的に埋めることができます。自動化ルールは、疑わしいログインを検出すると、このPlaybookとそのカスタムアクションをトリガーします。オプションA、B、D、Eは、誤った仮定、手動、または問題を回避するためのものです。

最新問題: **45**
SOCアナリストは、カスタムマルウェア亜種が関与する複雑な攻撃を調査しています。EDRは複数の疑わしいプロセスインジェクションとネットワーク接続を検出しましたが、マルウェアの起源、関連するユーザーアカウント、ネットワークを介したラテラルムーブメントに関する完全なコンテキストを提供できませんでした。アナリストは、詳細なフォレンジック分析を実施し、脅威

を迅速に封じ込める必要があります。EDRが提供する可能性のある次のKQLクエリについて考えてみましょう。

```
DeviceProcessEvents | where FileName == 'powershell.exe' and ProcessCommandLine contains '-EncodedCommand'
```

この EDR レベルのクエリを超えて、Cortex XDR の次の機能のうちどれが、この調査と対応において SOC アナリストに大きく役立ちますか? (該当するものをすべて選択してください)

**A.** 自動インシデント作成とストーリーライン相関: Cortex XDR は、エンドポイント、ネットワーク、クラウド、および ID からの関連アラートを、グラフィカルな攻撃ストーリーラインを持つ単一の 「インシデント」に自動的にまとめ、キル チェーン全体を明らかにします。

**B.** ネイティブ ネットワーク トラフィック分析: Cortex XDR のネットワーク センサー (ファイアウォールや専用の NTA など) は詳細なネットワーク セッション ログを提供するため、アナリストは EDR エージェントが確認できない横方向の移動や C2 通信を追跡できます。

**C.** 統合ユーザー行動分析 (UBA): 資格情報が盗まれた場合でも、通常とは異なる場所からログインするユーザー アカウントや、通常とは異なるリソースにアクセスするユーザー アカウントなどの異常なユーザー行動を検出します。

**D.** 自動修復プレイブック: Cortex XDR コンソールから直接、複数のセキュリティ レイヤーにわたる自動応答アクション (エンドポイントの分離、ファイアウォールでの IP のブロック、ユーザー アカウントの無効化など) をトリガーする機能。

**E.** エンドポイントにダウンロードされたすべてのファイルに対して、シグネチャベースのリアルタイムウイルス対策スキャンを実行します。

**Answer: A,B,C,D (メーッセージを残す)**

この質問は、XDRの 「」とCortex XDRの統合性に特に焦点を当てています。EDRクエリはエンドポイントのコンテキストを提供しますが、断片化されています。A: Cortex XDRのインシデントストーリーラインは、EDRだけでは実現できない攻撃の全体像を提供するという重要なメリットです。B: ネイティブネットワークトラフィック分析は、EDRの可視性が限られているラテラルムーブメントとC2を理解するために不可欠です。Cortex XDRは、ネットワークファイアウォールまたは専用のNTAのデータを活用します。C: UBAは、エンドポイントプロセス分析にとどまらず、侵害されたアカウントや内部脅威の検出に不可欠です。D: 複数のセキュリティドメインにまたがる自動修復は、迅速な対応のための重要なXDR機能ですが、EDRは通常、エンドポイント固有の分離を提供します。E: Cortex XDRには高度なエンドポイント保護が含まれていますが、リアルタイムのシグネチャベースのAVスキャンは基本的なEDR/EPP機能であり、この複雑な調査における「EDRを超える」機能を表すものではありません。

最新問題**: 46**

新しいコンプライアンス規制では、エンドポイントにおけるすべてのPII（個人識別情報）アクセスイベントをログに記録し、7年間保存し、容易に監査できるようにすることが義務付けられています。Cortex XDRの固有の機能は、ログ管理とコンプライアンスに関するこの特定の要件の遵守をどのように促進するのでしょうか？

**A.** Cortex XDR のデータ保護モジュールは、保存中のすべての PII データを自動的に暗号化するため、規制に従って詳細なアクセス ログを記録する必要がなくなります。

**B.** Cortex XDRはエンドポイントアクティビティログ（ファイルアクセスイベントを含む）を収集します。これらのログはフィルタリングされ、Cortex Data Lakeに長期間保存されるため、監査要件に対応できます。その後、コンプライアンスダッシュボードを設定できます。
**C.** Cortex XDR は、PII ログの収集と保持を担当するサードパーティの SIEM ソリューションと統合され、Cortex XDR の役割は純粋にインシデント検出のみになります。
**D.** Cortex XDR では、ユーザーに PII へのアクセスを制限する特定のロールが割り当てられるため、生成されるログの量が削減され、コンプライアンスが簡素化されます。
**E.** Cortex XDR には、PII アクセス専用のコンプライアンス レポート テンプレートが組み込まれており、検出時にログが不変のアーカイブに自動的にエクスポートされます。
**Answer: B (**メーッセージを残す**)**
Cortex XDRは、ファイルアクセスイベントを含む豊富なエンドポイントテレメトリを収集し、Cortex Data Lakeに保存できます。このデータレイクは長期保存向けに設計されており、強力なクエリ（XQL）とレポート作成機能を備えており、PIIへのログ記録と監査可能なアクセスに関するコンプライアンス要件を直接サポートします。このデータに基づいてコンプライアンスダッシュボードを構築できます。



最新問題**: 47**
セキュリティアナリストは、Cortex XSIAMを使用して、内部脅威の疑いのある事象を調査しています。「Alice」というユーザーが最近、営業時間外に機密性の高い金融文書にアクセスし、未知の外部IPアドレスへの大規模なデータ転送を開始したことが特定されました。この疑わしいアクティビティの検出と相関分析に最も効果的なXSIAMの機能とルールタイプはどれですか？また、この文脈におけるIOCとBIOCの主な違いは何ですか？
**A.** 異常なアクセス パターンとデータ流出に対する BIOC ルールを活用するユーザー ビヘイビア アナリティクス (UBA)。IOC は静的なインジケーターであり、BIOC は動作の異常です。
**B.** 既知の悪意のあるファイル ハッシュとネットワーク接続に対する IOC ルールを備えたエンドポイント検出および応答 (EDR)。IOC は動的インジケーターであり、BIOC は静的インジケーターです。
**C.** 未知の IP に対して従来のシグネチャベースの IOC ルールを使用するネットワーク検出および応答 (NDR)。ここで、IOC はデータの原子部分であり、BIOC は複雑な一連のイベントです。

**D.** クラウド リソースの誤った構成に対するコンプライアンス ベースの BIOC ルールを備えたクラウド セキュリティ ポスチャ管理 (CSPM)。IOC は脅威アクター側の TTP であり、BIOC は侵害されたホストです。
**E.** タイプに関係なく、あらゆるアラートによってトリガーされるセキュリティ オーケストレーション、自動化、および対応 (SOAR) プレイブック。IOC は独立したイベントであり、BIOC は高度にコンテキスト化されたアラートです。
**Answer: A (**メーッセージを残す**)**
ユーザー行動分析（UBA）は、通常のユーザー行動をベースライン化し、逸脱をフラグ付けすることで、内部脅威を検知するために不可欠です。行動侵害指標（BIOC）はまさにこのために設計されており、異常なイベントや行動のシーケンスを表すもので、これらを組み合わせることで悪意を示唆します。IOC（侵害指標は通常、過去または現在の侵害を示す静的でアトミックなデータ（ハッシュ、IP、ドメインなど）です。一方、BIOCは、個々のイベントは悪意がないかもしれないものの、その組み合わせが非常に疑わしく、侵害または脅威アクターの活動を示唆する、活動パターンまたはイベントシーケンスを表します。このシナリオでは、アリスの異常なアクセス時間とデータ転送は、BIOCによって最もよく検出される行動異常です。オプションB、C、D、Eは、主要な機能、またはIOCとBIOCの違いを誤って表現しています。

最新問題**: 48**
SOCアナリストは、組織が通常アクセスしない、通常とは異なるトップレベルドメイン（TLD）への内部ホストからのアウトバウンドDNSクエリが急増していることを確認しました。ホストはパッチ未適用のレガシーサーバーです。このアクティビティを検知し、対応を開始する主な責任を負っているSOC機能は次のうちどれですか？また、それらの機能が推奨すべき、最も緊急かつ優先度の高い対応策は何ですか？
**A.** 脅威インテリジェンス。TLD に既知の悪意のある関連付けがないか調査します。
**B.** セキュリティ監視とアラート; 侵害されたホストをネットワークから隔離します。
**C.** インシデント対応。EDR ソリューションをホストに直ちに展開します。
**D.** 脆弱性管理。レガシー サーバーにパッチを適用することを推奨します。
**E.** フォレンジック; 影響を受けるサーバーの完全なディスク イメージを開始します。
**Answer: B (**メーッセージを残す**)**
このような異常をリアルタイムで検知する主な役割は、セキュリティ監視とアラートです。侵害の疑いがある場合、特にC2のような異常なアウトバウンドトラフィックが発生した場合、最も緊急かつ最優先の対応は、ホストを隔離し、さらなる拡散やデータ流出を防ぐことです。他の選択肢も有効なSOC機能ですが、この緊急シナリオでは優先度は低くなります。脅威インテリジェンスは初期検知後に実施し、インシデント対応は隔離とその後の対応を網羅します。脆弱性管理は根本原因に対処しますが、差し迫った脅威には対処しません。フォレンジックは封じ込め後に実施します。

最新問題**: 49**

高度なXSOARプレイブックは、脆弱性管理を自動化するように設計されています。新しい脆弱性が発見された場合（例スキャナー統合によるもの）、プレイブックは以下の処理を実行する必要があります。
1. 脆弱性の詳細に基づいて影響を受ける資産を特定します。
2. 重要度に基づいて資産に優先順位を付けます (CMDB から取得)。
3. 優先度の高い資産については、パッチ適用のために ServiceNow で変更リクエストを自動的に作成します。
4. 中程度の優先度の資産については、資産所有者に手動レビュー タスクを割り当てます。
5. 未解決の脆弱性とその修復ステータスの概要レポートを毎週生成します。
XSOAR インシデント フィールド ( 影響を受けるホスト名」、 脆弱性 ID」など) と外部システム フィールド (ServiceNow の 構成項目」、 変更要求の説明」など) 間のデータの一貫性と動的なマッピングを確保するために、この双方向のデータ フローと変換に最も重要な XSOAR 機能はどれですか。
**A.** リアルタイムコラボレーションを実現する War Room と ChatOps 機能。
**B.** データを視覚的に表現するための XSOAR レイアウトとカスタム ダッシュボード。
**C.** 統合構成およびプレイブック タスク内の Mapper および Transformer 機能。
**D.** セキュリティとコンプライアンスのためのロールベースのアクセス制御 (RBAC) と監査ログ。
**E.** プレイブックを開始するためのジョブ スケジューリングとトリガー メカニズム。
**Answer: C (**メーッセージを残す**)**
「マッパー」と「トランスフォーマー」機能は、異なるシステム間のデータ整合性と動的マッピングを処理する上で極めて重要です。マッパーは、ServiceNow、CMDBなどの統合構成内で使用され、外部からの入力データをXSOARインシデントフィールドにどのようにマッピングし、XSOARインシデントデータを外部システムフィールドにどのようにマッピングするかを定義します。トランスフォーマー（多くの場合、JINJA2テンプレートまたはカスタム自動化スクリプトによって実装されます）は、外部システムとの間でデータを送受信する前に、複雑なデータ操作、フォーマット、およびエンリッチメントを可能にし、データが各システムの期待値に準拠していることを保証します。これは、双方向のデータフローと一貫性の維持にとって非常に重要です。オプションA、B、D、およびEは重要なXSOAR機能ですが、異種システム間のデータマッピングと変換の課題に直接対処するものではありません。

最新問題**: 50**
Palo Alto Networks NGFWが、高度な検知能力を持つカスタムマルウェアがデータ窃取を試みるシナリオを考えてみましょう。このマルウェアはDNS over HTTPS (DOH)を使用して従来のDNSフィルタリングを回避し、C2通信を確立します。SOCのNGFWに関する現在のポリシーは、既知の悪意のあるDOHドメインをブロックすることです。SOCがCortex XDRとWildFireも導入している場合、このような高度な窃取を検知・防止するために、どのようなNGFWセキュリティプロファイル、あるいはそれらの組み合わせを有効化・調整すべきでしょうか？
**A.** マルウェア署名を検出するためのウイルス対策およびスパイウェア対策プロファイル。
**B.** DOH サーバー IP をブロックする URL フィルタリング プロファイル。

**C.** DOH C2 トラフィック用のカスタム シグネチャと、機密データ タイプの流出を防ぐためのデータ フィルタリング プロファイルを備えた脅威防止 (IPS) プロファイル。
**D.** SSL/TLS 検査の復号化プロファイルと、DOH ペイロードを分析するための送信 HTTP/S トラフィック上の WildFire 分析プロファイル、および DNS トラフィックの動作分析のための Advanced Threat Prevention (ATP) サブスクリプションを組み合わせたもの。
**E.** DOH トラフィック量を軽減する DoS 保護プロファイルと、ファイル転送を防止するファイル ブロッキング プロファイル。
**Answer: D (**メーッセージを残す**)**
回避的な DOH の流出を検出して防止するには、複数の高度な機能が必要です。
1. 復号プロファイル（SSL/TLSインスペクション）：DOHトラフィックは暗号化されています。復号を行わないと、NGFWはDOHリクエストの内部内容を検査してC2通信や流出したデータを特定できません。
2. WildFire 分析プロファイル: 復号化された後、NGFW は復号化された DOH ペイロード (カスタム マルウェアの C2 トラフィックまたはデータ フラグメントが含まれている可能性があります) を WildFire に転送し、動的分析とゼロデイ検出を行うことができます。
3. Advanced Threat Prevention (ATP) サブスクリプション: DNS トラフィックを含む、より高度な動作分析が可能になり、C2 を示す異常な DOH パターンの識別に役立ちます。
A（ウイルス対策／スパイウェア対策）は既知のシグネチャに依存しており、カスタムマルウェアはこれを回避します。B（URLフィルタリング）は、DOHサーバーが既知の悪意のあるIPアドレスであれば有効かもしれませんが、回避型マルウェアは動的IPアドレスや新規IPアドレスを使用することが多いです。C（カスタムIPS／データフィルタリング）は有効ですが、復号化を行わないとIPSシグネチャはトラフィックを認識できず、データフィルタリングは暗号化されたデータを認識できません。E（DoS／ファイルブロッキング）は範囲が広すぎて、回避型DOHデータの抽出検出に特化していません。

最新問題**: 51**
セキュリティ監査担当者は、Cortex XSIAMの脅威検知機能が、新型マルウェアやポリモーフィック型マルウェアに対してどれほど有効であるか疑問視しています。監査担当者は特に、事前シグネチャのない脅威の検知において、XSIAMが従来のSIEMやEDRとどのように差別化できるかを尋ねています。監査担当者の懸念に対処する上で重要なXSIAMの機能は、次のうちどれですか？
**A.** Cortex XSIAM の強みは、事前定義されたシグネチャと IOC の豊富なライブラリにあり、これらは 1 時間ごとに更新されます。
**B.** XSIAMは
**C.** XSIAM の主な利点は、従来の SIEM と比較して、より幅広いサードパーティのセキュリティツールと統合できることです。
**D.** XSIAMは高度な
**E.** XSIAMは、
**Answer: (**解答を表示する**)**

この質問は、XSIAM が新規およびポリモーフィック型の脅威を検知する際の中核的な差別化要因に直接的に言及しています。選択肢 B は、XSIAM の高度な検知機能を正確に説明しています。統合データレイク全体で機械学習と AI を活用することで、既知のシグネチャのない脅威（ポリモーフィック型マルウェアやゼロデイなど）にとって極めて重要な、行動異常の検知を可能にします。行動脅威保護、ネットワーク脅威検知、UBA はすべて、エンドポイント、ネットワーク、ユーザー全体のアクティビティを分析する、この機能に貢献する主要コンポーネントです。選択肢 A は、従来のシグネチャベースの検知について説明しています。選択肢 C は機能ではありますが、新規脅威検知における主要な差別化要因ではありません。選択肢 D と E は、予防的または間接的な対策について説明しており、新規脅威の検知メカニズムの中核的なものではありません。

最新問題: **52**
APT（Advanced Persistent Threat）攻撃グループが、標的組織から機密性の高いデータを盗み出すことに成功しました。侵入後の分析により、攻撃者は高度に難読化されたカスタムPowerShellスクリプトを使用してデータを圧縮し、DNSクエリ（DNSトンネリング）を介して、一見正当なドメインに見せかけたドメインに徐々にデータを盗み出していたことが明らかになりました。エンドポイントプロセスログにはPowerShellによるCPU使用率の高さが記録されていましたが、Cortex XDRの行動分析機能は、この盗み出しフェーズで重大度の高いアラートをトリガーしませんでした。SOCチームは、今後の再発防止のため、行動分析の設定を見直しています。行動分析機能がこの特定の盗み出しを検知できなかった理由として最も考えられるのはどれですか？また、どのような調整を行えば検知率を大幅に向上させることができますか？（該当するものをすべて選択してください）
**A.** 「DNS 流出」または「異常な DNS クエリ」の動作モデルは、正当な DNS トラフィックを装った微妙で低速なデータ量を識別できるほど十分に調整またはトレーニングされていませんでした。
**B.** C2 または流出によく使用される、新しく登録されたドメインや疑わしいドメインを識別するために特別に設計された外部脅威インテリジェンス フィードとの統合が不足しています。
**C.** 「データ窃盗を伴う PowerShell 実行」を検出するために特別に設計された Behavioral Threat Protection (BTP) ルールは範囲が広すぎたり狭すぎたりしたため、誤検知が過剰になったり、この特定の難読化されたメソッドが検出されなかったりしました。
**D.** Cortex XDR の機械学習モデルは、PowerShell の CPU 使用率とネットワーク トラフィックの「通常の」ベースラインを十分に設定しなかったため、わずかに高い CPU 使用率と非常に異常な DNS トラフィックが異常検出しきい値を下回りました。
**E.** 組織のネットワーク アーキテクチャにより、Cortex XDR は DNS クエリの完全な内容を監視できず、宛先 IP とポートのみを確認できました。
**Answer: (**解答を表示する**)**
これは高度な回避策を伴う複雑なシナリオです。選択された各オプションが、考えられる理由と有効な調整方法である理由を分析してみましょう。A：「DNSデータ流出」または「異常なDNSクエリ」の動作モデルが十分に調整されていませんでした。DNSトンネリングは検知が難しいものです。動作モデルがDNSトンネリングの特性（異常に長いクエリ長、単一ドメインに対するA/TXTレコードの高頻度、非標準のサブドメインなど）に合わせて特別にトレーニングまたは調整されてい

ない場合、特にデータがゆっくりと流出する場合、検出を見逃す可能性があります。高度なDNS分析がここでは不可欠です。C：「データ流出を伴うPowerShell実行」を検出するために特別に設計されたBehavioral Threat Protection（BTP）ルールが広すぎたり狭すぎたりしていました。BTPは動作シーケンスの認識に依存しています。難読化されたPowerShellスクリプトや、DNSトンネリングのような非常に異常な流出方法は、一般的なBTPルールをバイパスする可能性があります。PowerShellアクティビティとDNS異常のこの特定の組み合わせを探すためにBTPをカスタマイズするか、新しい動作指標（BI）を作成することは、直接的な改善につながるでしょう。D: Cortex XDRの機械学習モデルは、「正常な」PowerShellのCPU使用率とネットワークトラフィックのベースラインを十分に設定していませんでした…「ゆっくりとデータを抜き出す」という表現は、ベースラインの粒度が十分でなければ、純粋に統計的な観点から見ると、「高いCPU使用率」がPowerShellの「正常な」偏差内であった可能性を示唆しています。さらに重要なのは、ネットワークトラフィック（DNSトンネリング）の性質が非常に異常であるにもかかわらず、モデルがこれを具体的に探していなかった場合、または異常しきい値が高すぎた場合、見逃される可能性があるということです。ベースライン設定と感度調整の改善が鍵となります。E: 組織のネットワークアーキテクチャが原因で、Cortex XDRはDNSクエリの完全な内容を観察できませんでした…これはDNSトンネリングの検出に極めて重要です。Cortex XDR（またはその基盤となるセンサー）が送信元/宛先のIPアドレスとポート番号のみしか認識できない場合、DNSクエリの内容（サブドメイン内の窃取されたデータなど）を分析することはできません。DNSクエリログの完全な可視性は不可欠です。B：外部の脅威インテリジェンスフィードとの統合不足…脅威インテリジェンス（TI）は常に有益ですが、ゼロデイ攻撃やカスタム開発されたC2/窃取ドメインを即座に見逃す主な理由となる可能性は低いです。APTグループは、攻撃時点では既存のTIフィードには存在しない、新規登録または侵害された正規のドメインを使用することが多いです。TIは事後分析と将来の予防に役立ちますが、行動分析は未知の脅威を捕捉することを目的としています。したがって、TIは有益ではありますが、他のオプションほど行動自体を見逃す直接的な原因にはなりません。

最新問題: **53**
高度な内部脅威アクターは、正規のホワイトリストに登録されたチャネルを介して、暗号化された少量のデータを段階的に送信することで、検出を回避し、機密データを盗み出しています。アクターは、エンドポイント、クラウドストレージ同期クライアント、正規のSaaSアプリケーション上のPowerShellスクリプトを組み合わせて使用しています。Cortex XSIAMが導入されていますが、「ログスティッチング」では、一見無害で少量のイベントを、データ窃盗を示す信頼性の高いインシデントとして統合できないことがよくあります。このような段階的なデータ窃盗を検出する上で、XSIAMの高度なログスティッチング機能またはサポート機能のうち、最も重要なものはどれでしょうか？
**A.** 実行前にすべての PowerShell スクリプトを静的に分析し、既知の悪意のあるシグネチャを探します。
**B.** 同一のネットワーク フロー ログを集約して、ボリュームを削減し、分析を簡素化します。
**C.** ユーザー/エンドポイントごとの「通常の」データ転送量と送信先のベースラインを確立し、時間の経過に伴う偏差をまとめるユーザーおよびエンティティ行動分析 (UEBA) モデル。

**D.** すべての実行可能ファイルをリアルタイムでサンドボックス化し、ゼロデイマルウェアを識別します。
**E.** 取り込まれたログと特定の IOC を照合するためのルールベースの相関エンジン。
**Answer: (**解答を表示する**)**
このシナリオは、従来のシグネチャベースやルールベースの方法では捕捉が非常に困難な「低速でゆっくりとした」データ流出を表しています。個々のイベント（正規のチャネルを介した小規模なデータ転送）は一見無害に見えるかもしれません。ここで、ログスティッチングと統合されたUEBAの威力が極めて重要になります。「C」（UEBAモデル）は最も重要な機能です。XSIAMのUEBAは、ユーザーとエンティティの「正常な」行動のベースライン（典型的なデータ転送量、一般的な送信先、データ同期の通常のタイミングなど）を構築します。内部脅威アクターが徐々にデータを流出させ始めると、たとえ個々のチャンクが小さくても、累積的な影響、または頻度、送信先、または総量の観点からベースラインからのわずかな逸脱は、UEBAによって異常としてフラグ付けされます。XSIAMのログスティッチングは、これらの個々の異常イベント（複数のログソースや時間に分散している可能性があります）を統合し、段階的なデータ流出のパターンを示す信頼性の高いインシデントを生成します。これは、人間のアナリストや単純なルールではノイズの中から見つけるのが難しいものです。他のオプションは、この特定の「低速でゆっくりとした」かつ「正当なチャネル」による流出手法には効果がありません。

最新問題**: 54**
一般的なブラウザを標的とした新たなゼロデイエクスプロイトが公開されました。SOCチームは、ベンダーからのパッチが公開される前に、潜在的なエクスプロイトの試みを特定するために、Cortex XSIAMにカスタム検出ルールを迅速に導入する必要があります。このエクスプロイトには、正規のブラウザアクティビティでは一般的ではない特定のAPI呼び出しシーケンスとメモリアクセスパターンが含まれています。効果的で誤検知率の低い検出を行うために、XSIAM内で以下のルールタイプと考慮事項のうち、最も適切なものはどれでしょうか？
**A.** エクスプロイトの動作面を無視して、ファイル名内の特定の文字列を検索する、単純な静的シグネチャベースのルール。
**B.** XQL (Cortex Query Language) を活用した「動作」ルールは、プロセス アクティビティ、ネットワーク接続、およびメモリ割り当ての複雑なシーケンスを定義し、特に既知のエクスプロイト パターンをターゲットにし、正当なベースライン アクティビティのアラート抑制と組み合わせます。
**C.** 既知の悪意のある実行可能ファイルをブロックする「ファイル ハッシュ」ルールですが、ハッシュが最初は不明なゼロデイ攻撃には効果がありません。
**D.** ブラウザの実行可能ファイルへのすべてのトラフィックをブロックし、重大なサービス中断を引き起こす「ネットワーク」ルール。
**E.** カスタム ルール エンジニアリングを使用せずに、XSIAM の機械学習モデルのみを使用してゼロデイを検出します。カスタム ルール エンジニアリングは、即時のターゲット検出には遅すぎたり、汎用的すぎる可能性があります。
**Answer: B (**メーッセージを残す**)**

特定の行動パターンを持つゼロデイエクスプロイトには、XQLを用いた高度な行動ルールが最適です。XQLは、様々なテレメトリポイント（プロセス、ネットワーク、メモリ）を相関させる複雑なクエリを実行し、エクスプロイト固有の特性を特定できます。これを既知の正当なアクティビティに対するアラート抑制と組み合わせることで、誤検知の削減に役立ちます。静的シグネチャ（A）は未知の脅威には効果がなく、ハッシュベースのルール（C）は事前の知識が必要であり、広範なネットワークブロック（D）は混乱を招きます。機械学習（E）は強力ですが、ターゲットを絞ったカスタムルールを使用することで、新たに発見されたゼロデイエクスプロイトを迅速かつ正確に検出できます。

最新問題: **55**
ある組織は最近、インフラストラクチャの大部分をマルチクラウド環境（AWS、Azure）に移行しました。Cortex XDRからの重要なアラートは、AWSのEC2インスタンスから発生した「不正なAPIキーの使用」を示しており、その後、Azureサブスクリプションで異常なアクティビティが発生しています。SOCチームは、高度な技術を持つ攻撃者が認証情報を侵害し、複数のクラウド環境を切り替えて利用しているのではないかと疑っています。調査担当者として、侵害されたAPIキーを正確に特定し、AWSとAzureの両方でその使用状況を追跡し、特定のクラウド資産への影響を判断するために、Cortex XDRの機能をどのように活用すればよいでしょうか？
**A.** Cortex XDRのクラウドセキュリティモジュール統合を利用して、AWS CloudTrailログで「不正なAPIキーの使用」イベントを分析します。具体的には、UserIdentity.accessKeyIdを探します。次に、この「accessKeyId」をXDR経由で取得したAzureアクティビティログと相関させ、一致するアクティビティを検索します。特に「CallerIpAddress」と「OperationName」に注目し、実行された具体的なアクションと「ResourceGroup」や「SubscriptionId」などのAzureリソースへの影響を特定します。最後に、「インシデントグラフ」を使用して、クロスクラウドのキルチェーンを可視化します。
**B.** 侵害を受けたEC2インスタンスを直ちに隔離してください。Live Responseを実行してEC2インスタンスのディスクフォレンジック情報を収集し、構成ファイル内のAPIキーを見つけてください。Azure ADのサインインログで、EC2インスタンスと同じIPアドレスを手動で検索してください。
**C.** AWS IAMで侵害されたAPIキーをブロックし、それに関連付けられたユーザーアカウントを無効化します。AWSとAzureの両方のネットワークセキュリティグループに焦点を当て、送信トラフィックを制限します。さらなる侵害を示す新たなアラートを待ちます。
**D.** AWSとAzureの両方のすべてのクラウド資産に対して脆弱性スキャンを実行し、パッチが適用されていないサービスを特定します。攻撃者が既知の脆弱性を悪用したと想定します。両方のクラウド環境におけるユーザーロールと権限を確認し、過剰な権限が付与されていないか確認します。
**E.** WildFire を活用し、EC2 インスタンス上で見つかった疑わしいスクリプトやバイナリを静的および動的に分析します。その後、Autofocus を使用してクロスクラウド攻撃に関連する脅威インテリジェンスを検索し、観測された侵害指標に基づいてグローバルブロックを適用します。
**Answer: (**解答を表示する**)**

このシナリオは、マルチクラウド環境における XDR の重要性を強調しています。オプション A は、最も効果的で統合されたアプローチを提供します。クラウド セキュリティ モジュールの統合: Cortex XDR は、クラウド プロバイダーのログ (AWS の場合は CloudTrail、Azure の場合はアクティビティ ログ) と統合されます。これは、クラウドネイティブ攻撃の検出と調査に不可欠です。API キーの識別: CloudTrail ログには、API 呼び出しの「UserIdentity.accessKeyId」が正確に記録されるため、侵害されたキーを直接識別できます。クラウド間の相関: Cortex XDR (Cortex Data Lake 経由など) 内で AWS と Azure の両方のログを取り込んで相関させる機能により、調査担当者は侵害された「accessKeyId」または関連する「CallerIpAddressS」を両方の環境で追跡し、ピボットを特定できます。影響評価: クラウド ログの「OperationName」、「ResourceGroup」、および「SubscriptionId」に焦点を当てることで、どのようなアクションが実行され、どのクラウド資産が影響を受けたかを判断するのに役立ちます。インシデントグラフ :インシデントグラフで複雑で多段階的なクロスクラウド攻撃を可視化することで、キルチェーン、タイムライン、そして異なるクラウド環境におけるイベント間の関係性を把握しやすくなります。オプションB、C、D、Eは、事後対応的、手動対応が多すぎる、クラウド間の相関関係を考慮していない、あるいは特定のAPIキーの侵害とピボットを対象とする調査ではなく、一般的なセキュリティ対策に重点を置くといった問題があります。

最新問題**: 56**
セキュリティオペレーションセンター（SOC）がCortex XSIAMを導入しています。大規模エンタープライズネットワークへの初期センサー導入フェーズにおいて、チームは地理的に分散した複数のWindows Server 2019インスタンスからのデータ取得、特にDNSクエリログとプロセス実行の詳細に関する問題に遭遇しました。ネットワークトポロジには、複数のファイアウォール、プロキシ、そして最終的にXSIAMからエンリッチされたデータを受信する中央SIEMが含まれています。以下のCortex XSIAMセンサータイプのうち、この種の詳細なホストレベルテレメトリの収集を主に担っているのはどれですか？また、このシナリオにおいて、データ取得の失敗につながる可能性のある一般的な構成上の課題は何ですか？
**A.** ネットワーク センサー (ネットワーク パケット、NetFlow など) が主な選択肢となりますが、一般的な課題としては、ファイアウォール ポートのブロック (NetFlow の場合は UDP/4739) や、NetFlow エクスポート構成の誤りなどが挙げられます。
**B.** ホスト センサー (エンドポイント エージェントなど) はこのデータにとって非常に重要であり、一般的な課題としては、グループ ポリシー オブジェクト (GPO) によるエージェントのインストールの妨げ、ウイルス対策/EDR の競合、Cortex XSIAM ブローカーへのネットワーク接続の不足などがあります。
**C.** このデータにはクラウド センサー (AWS CloudTrail、Azure アクティビティ ログなど) が不可欠であり、よくある課題としては、IAM ロール/サービス プリンシパルの構成ミスや、ログ ストリームにアクセスするための API 権限の欠如などがあります。
**D.** ID センサー (Active Directory、Okta など) が原因であり、一般的な課題としては、LDAP/SCIM 接続の問題や、ディレクトリ同期のためのサービス アカウント権限の不足などがあります。

**E.** データ収集にはオーケストレーション センサー (SOAR プレイブックなど) が使用されますが、よくある課題としては、API キーのローテーションが正しくなかったり、Webhook エンドポイントが誤って構成されていたりして、自動データ プルが妨げられることが挙げられます。
**Answer: B (**メーッセージを残す**)**
ホストセンサー、特にエンドポイントエージェント（Cortex XDRエージェントなど）は、DNSクエリログ、プロセス実行の詳細、ファイルアクティビティ、ネットワーク接続といった詳細なホストレベルのテレメトリをエンドポイントやサーバーから直接収集するように設計されています。ホストセンサーの導入とデータ取り込みにおける一般的な課題は、GPOによるインストールのブロック、既存のセキュリティソフトウェア（アンチウイルス/EDR）との競合、ネットワーク接続の問題によりエージェントがXSIAMブローカーやXSIAMクラウドに直接アクセスできないといったエンタープライズレベルの設定に起因することがよくあります。オプションA、C、D、Eは、指定されたデータ収集シナリオとは異なるセンサータイプ、または無関係な課題を示しています。

最新問題**: 57**
ある組織は、内部脅威と潜在的なデータ流出を懸念しています。脅威ハンティングチームは、不満を抱えた従業員が正規のクラウドストレージサービス（Dropbox、Google Driveなど）を不正なデータ転送に使用し、特に大容量ファイルを狙っているのではないかと疑っています。Palo Alto Networksのファイアウォールには、App-ID、URLフィルタリング、データフィルタリングが設定されており、すべてのログはCortex Data Lakeに送信されます。特定のユーザーによる、認可されたクラウドストレージサービスへの不審な大容量ファイル転送を特定するには、Palo Alto Networksの機能とハンティング手法のどの組み合わせが最も効果的でしょうか？
**A.** クラウドストレージアプリケーションへのファイル転送をすべてブロックするセキュリティポリシーを作成してください。ブロックログを監視してください。これは予防策であり、ハンティング手法ではありません。また、重大な業務中断を引き起こす可能性があります。
**B.** データフィルタリングプロファイルを設定して、機密性の高いファイルタイプ（例：「財務文書」、「ソースコード」）を検出し、承認されたクラウドストレージアプリケーションを許可するセキュリティポリシーに適用します。データフィルタリングログでヒットを監視し、特に「app」が「dropbox-base」、「google-drive-base」などに一致するものや、内部ユーザーIPからの大容量転送を示す「bytes」に注目してください。これにより、ファイルの内容に関する詳細な情報が得られます。
**C.** Sapp カテゴリ「cloud-storage」の URL ログを分析します。1 GB を超える値を探します。ユーザー ID と相関関係を調べます。これにより大規模な転送を特定できますが、データの機密性やユーザー認証コンテキストは確認できません。
**D.** 「dropbox-upload」、「google-drive-upload」などのアプリケーションのApp-IDログを確認してください。「bytes_sent」値が高いセッションをフィルタリングしてください。エンドポイントログを介して、これらのセッションを内部ファイル共有上の既知の機密データの場所と相互参照してください。これには外部相関分析が必要であり、汎用的な「ベース」アプリ経由のアップロードが見逃される可能性があります。

**E.** 従業員を識別するためのUser-IDを実装します。そのユーザーに対して特定のセキュリティポリシールールを設定し、「ウェブブラウジング」と「SSL」アプリケーションのみを許可します。このユーザーによる標準外のアプリケーションアクティビティがないか、脅威ログを監視します。これは過度に制限的で事後対応的な封じ込め策であり、大容量ファイル転送に対するハンティング戦略としては適切ではありません。

**Answer: (**解答を表示する**)**

ここで重要なのは、「不正なデータ転送」、「大容量ファイル」、「機密コンテンツ」を特定することです。オプションBは最も包括的かつ効果的です。データフィルタリング（Palo Alto Networksのデータ損失防止機能の一部）は、機密情報を検出するために明示的に設計されています。このプロファイルをクラウドストレージを許可するポリシーに適用することで、ファイアウォールは転送されるファイルの実際の内容を検査できます。これを、高い「バイト数」値と特定の「アプリ」カテゴリ（アップロードを含むDropboxの一般的なアクティビティをカバーする「dropbox-base」など）の監視と組み合わせることで、認可されたクラウドサービスへの大規模で機密性の高いデータの流出を正確に検出できます。これは、「機密データ」と「大容量ファイル」の基準に直接対応します。オプションAは予防的であり、検出ではありません。オプションCは大容量転送を特定しますが、機密コンテンツは特定しません。オプションDはエンドポイントログとの外部相関分析を必要としますが、これはデータ流出に対するファイアウォールの検出手法とは直接関係ありません。オプションEは事後対応型の封じ込め対策です。

最新問題**: 58**

ある大企業が、従来のSIEMからCortex XSIAMへの移行を進めています。既存のSplunkクエリとカスタム相関ルールの膨大なリポジトリがあり、それらは自社の環境で非常に高い効果を発揮しています。セキュリティアーキテクトは、これらの既存のセキュリティロジックをXSIAMのネイティブ検出機能に移植するための労力を最小限に抑えたいと考えています。この目標を効率的かつ効果的に、場合によっては自動化も活用しながら達成するために、以下のコンテンツパックコンポーネントのうち、最も適切なものはどれでしょうか？

**A.** インシデント レイアウトと応答プレイブック。検出後のワークフローを規定します。

**B.** データ モデルとパーサー。特に、Splunk データを XSIAM の Common Information Model (CIM) に正規化することに重点を置いています。

**C.** 検出ルール (相関ルール、行動バイアス) とダッシュボード。ロジックを直接変換して可視性を提供します。

**D.** 同じ脅威インテリジェンスを取得するための外部統合とインジケーター フィード構成。

**E.** インシデントの量を管理するためのアラートのグループ化と抑制ポリシー。

**Answer: C (**メーッセージを残す**)**

Splunk クエリとカスタム相関ルールの変換の核心は、XSIAM 内での検出ロジックの複製にあります。これは、相関ルールと行動バイアスを含む XSIAM の検出ルールに直接マッピングされます。これらは、Splunk の相関検索と同様に、セキュリティインシデントを識別するための条件とロジックが定義されているコンポーネントです。ダッシュボードも、Splunk ダッシュボードが提供するのと同じ可視性と洞察を提供するために不可欠です。データモデルとパーサー (オプショ

ン B) はデータの取り込みと正規化に不可欠ですが、検出ルールの前提条件であり、ロジックを直接変換するものではありません。インシデントレイアウトと対応プレイブック (オプション A) は検出の後に使用されます。外部統合 (オプション D) はデータソースに関するものであり、ロジックに関するものではありません。アラートのグループ化 (オプション E) はインシデント管理に関するものであり、ルールの変換に関するものではありません。

最新問題: **59**
SOCは、エンドポイントで実行された疑わしいPowerShellコマンドがランサムウェア攻撃の既知のTTPに一致することを示すアラートをCortex XDRから受信しました。NISTインシデント対応計画の 準備」フェーズは、効果的な対応に不可欠です。このシナリオを考慮すると、 準備」フェーズのどの側面が、迅速かつ効果的な 検知・分析」および 封じ込め」対応を可能にする上で最も直接的に有益であることが示されていますか？
**A.** Cortex XDR 自動化機能を活用して、ランサムウェアに対する具体的な手順を含む包括的なインシデント対応プレイブックを開発し、定期的に更新します。
**B.** Cortex XDR を含むすべてのセキュリティ ツールが完全に統合され、WildFire および AutoFocus と双方向に脅威インテリジェンスを共有するように構成されていることを確認します。
**C.** 毎年、組織全体でフィッシングシミュレーションと全従業員を対象としたセキュリティ意識向上トレーニングを実施します。
**D.** インシデント対応チームと外部の利害関係者 (法務、PR など) 内で明確なコミュニケーションチャネルと役割/責任を確立します。
**E.** 重要な資産の識別と分類とともに、最新のハードウェアおよびソフトウェア インベントリを維持します。
**Answer: A,B,D,E (**メーッセージを残す**)**
準備」フェーズは、効率的なインシデント対応の基盤を構築します。すべてのオプションは準備の一部ですが、この特定のシナリオでは、検出／分析と封じ込めに直接影響を与えるオプションがいくつかあります。- A: Cortex XDR自動化機能を備えた適切に開発されたプレイブック（例ランサムウェア封じ込めのプレイブック）は、対応アクションを直接ガイドし、迅速化することで、検出、分析、封じ込めの両方に影響を与えます。- B: セキュリティツール（Cortex XDR、WildFire、AutoFocus）の統合により、脅威の相関分析の迅速化、疑わしいファイルの自動分析、新しい保護対策の迅速な導入が可能になり、検出と分析を直接サポートし、共有された脅威インテリジェンスを活用することで効果的な封じ込めを実現します。- C: フィッシングシミュレーションと意識向上トレーニングは予防策であり、準備の一部ですが、インシデント発生後の技術的な検出、分析、封じ込めを直接促進するものではありません。- D: 明確なコミュニケーションチャネルと明確な役割・責任（誰が何をするか、誰に報告するか）は、迅速かつ効果的な対応を調整するための基本であり、迅速な意思決定を確保することで、特に封じ込めを含むすべてのフェーズに影響を与えます。- E: 最新のインベントリと資産分類は、影響（検知／分析）を理解し、封じ込め活動の優先順位を決定し、適切な資産を優先的に保護するために不可欠です。保有資産を把握することで、異常を検知し、効果的に封じ込めることができます。

最新問題: **60**
高度なAPTグループは、正規のシステムユーティリティを装ったPowerShellスクリプトを実行するスケジュールタスクを作成することで、初期のネットワーク防御を回避し、Windowsドメインコントローラに永続性を確立します。Cortex XDRは、異常なプロセス作成とラテラルムーブメントの試みを特定します。Palo Alto Networksのセキュリティ運用担当者として、NISTインシデント対応計画の「根絶」サブフェーズにおいて、PowerShellスクリプトの悪意のある性質と永続性メカニズムを確認し、かつ業務への影響を最小限に抑えながら、どのような効果的で高度なアクションを優先しますか？
**A.** 影響を受けるドメイン コントローラーのネットワーク インターフェイスを直ちに無効にし、サーバーの完全な再イメージ化を続行します。
**B.** Cortex XDR の Live Response を使用して、悪意のある PowerShell プロセスをリモートで終了し、スケジュールされたタスクを削除してから、特定されたスクリプト ハッシュに対してカスタム IOC 除外ルールを展開します。
**C.** ファイアウォール セキュリティ ポリシーを変更して、すべてのドメイン コントローラー上のすべての PowerShell トラフィックをブロックし、ドメイン コントローラーの以前の正常なバックアップにロールバックします。
**D.** ドメイン コントローラーの完全なメモリ ダンプを開始し、詳細な分析のために外部のフォレンジック ラボに送信して、結果が返されるまで駆除を延期します。
**E.** 組織全体に汎用エンドポイント セキュリティ更新をプッシュして、すべての潜在的な脆弱性を修正します。
**Answer: B (**メーッセージを残す**)**
「根絶」フェーズでは、インシデントの根本原因の除去に重点を置きます。オプションBは最も正確かつ効果的です。Cortex XDRのLive Responseを使用すると、重要なドメインコントローラをオフラインにすることなく、悪意のあるプロセスと永続化メカニズム（スケジュールされたタスク）を外科的に削除できるため、業務への影響を最小限に抑えることができます。カスタムIOC除外ルールを導入することで、スクリプトが再び出現した場合（例えば、別の侵害されたホストから）でも、即座に特定されブロックされます。ドメインコントローラの無効化（A）または再イメージ化（C）は大きな混乱を引き起こすため、永続化の正確な原因が判明し、削除済みであれば不要となる可能性があります。メモリダンプの送信（D）は根絶を遅らせ、汎用アップデート（E）は事後対応的であり、特定された脅威に固有のものではありません。

最新問題: **61**
サイバーセキュリティチームは、新たな脅威ハンティングワークフローを構築しています。SIEMに定期的に（例えば1時間ごとに）クエリを実行し、疑わしいアクティビティを検出し、EDRのデータで結果を拡充し、高忠実度アラートが生成された場合はXSOARで新しいインシデントを作成する必要があります。高忠実度アラートが見つからない場合でも、サマリーログを記録する必要があります。XSOARコンポーネントのどの組み合わせが最も効率的で保守性の高いソリューションとなるでしょうか？

**A.** SIEM をクエリし、EDR データで強化し、条件に応じてインシデントまたはログの概要を作成する、スケジュールされた Python スクリプト。
**B.** cronスケジュールで設定されたジョブ。プレイブックを実行します。このプレイブックには、SIEMへのクエリ、EDRエンリッチメント用のサブプレイブックの呼び出し、条件付きでインシデントの作成またはサマリーのログ記録を行うタスクが含まれています。
**C.** 1 時間ごとにトリガーされ、複雑な JavaScript スクリプトを実行してすべてのステップを実行し、インシデントを作成する自動化ルール。
**D.** SIEM コネクタとして機能し、アラートを継続的にポーリングして、別のインシデント作成プレイブックをトリガーするカスタム統合。
**E.** クエリ、エンリッチメント、インシデント作成のために、それぞれ個別のジョブによってスケジュールされた複数のスタンドアロン Python スクリプト。
**Answer: B (**メーッセージを残す**)**
このシナリオには、複数のステップと条件付きロジックを含む、スケジュールされた定期的なプロセスが含まれます。ジョブは、スケジュールの側面に最適です。プレイブックは、クエリ統合（SIEM、EDR）、データのエンリッチメント、条件付きインシデント作成などの複雑なワークフローをオーケストレーションするために設計されています。EDRエンリッチメント用のサブプレイブックは、モジュール性と再利用性を促進します。オプションAは、1つのスクリプトにロジックを詰め込みすぎて、視覚的に見にくくなり、保守が困難になります。オプションCは、複雑なワークフローに対する堅牢性が低くなります。オプションDは、一般的なプルベースの統合について説明していますが、エンリッチメントと条件付きインシデント作成のオーケストレーションは、統合によってトリガーされるプレイブック、またはこの場合は、データをプルするスケジュールされたジョブによって処理するのが最適です。オプションEは、単一のオーケストレーションされたワークフローではなく、複数のジョブとスクリプトによって不要な複雑さを生み出します。



最新問題**: 62**
高度な技術を持つ攻撃者が企業ネットワークへの初期アクセスを獲得し、永続的な攻撃を試みています。彼らは、正規のスケジュールタスクを改変してログオン時に悪意のあるスクリプトを実行するという、あまり一般的ではない手法を用いています。ただし、新しいタスクを作成したり、タスク名を大幅に変更したりすることは避けています。Cortex XDRのデフォルトの行動分析機能は、これを効果的に検知し、防御します。 「対象イベント」という概念と 「イベントシーケンス」に基

づく行動分析機能の中で、特に効果的なのはどれでしょうか。また、従来のシグネチャベースの手法よりも優れているのはなぜでしょうか。
**A.** 静的 AI 分析: スケジュールされたタスクが実行される前に、ディスク上のファイルに悪意のあるコードがないか検査します。
**B.** ハッシュベースの検出: 正当なスケジュールされたタスク ファイルの変更されたハッシュを識別します。
**C.** 行動ベースの脅威保護 (BTP): スケジュールされたタスクを変更し、その後、異常なスクリプトまたは署名されていないスクリプトを既知の悪意のあるパターンとして実行するアクション プロセスのシーケンスを識別します。
**D.** WildFire サンドボックス: 悪意のあるスクリプトを仮想環境で実行し、その悪意のある動作を観察します。
**E.** IP レピュテーション分析: 攻撃者がスケジュールされたタスクを変更した IP アドレスをブラックリストに登録します。
**Answer: (**解答を表示する**)**
このシナリオは、Cortex XDRのBehavioral Threat Protection （BTP）の強みを的確に表しています。BTPは、一連のイベント（例えば、プロセスがスケジュールされたタスクAPIにアクセスし、その後、認識されていない、あるいは疑わしいスクリプトを実行するなど）を監視し、それらを相関分析することで悪意のあるキルチェーンを特定します。ここで重要なのは、「正当なスケジュールされたタスクの変更」と「悪意のあるスクリプトの実行」の組み合わせです。従来のシグネチャベースの手法では、新しい悪意のある実行可能ファイルのシグネチャが存在せず、タスク名も正当なため、これを見逃してしまう可能性が高くなります。静的AI （A）とWildFire （D）は通常、ファイル分析を目的としており、正当なシステムコンポーネントの挙動変更には適用されません。ハッシュベースの検出 （B）は、ファイル自体が大幅に変更された場合でも機能しますが、多くの場合、変更されるのはタスクに関連するコマンドライン引数やレジストリエントリのみであり、バイナリは変更されません。IPレピュテーション （E）はネットワークに焦点を当てており、エンドポイントのパーシステンスメカニズムとは無関係です。

最新問題**: 63**
定期的なセキュリティ監査中に、数週間前にAPT （Advanced Persistent Threat） 攻撃グループによって重要なサーバーが侵害されたことが判明しました。この侵害には高度なラテラルムーブメントとデータ窃取が含まれていましたが、Palo Alto Networks Cortex XDRエンドポイント保護プラットフォームとWildFireクラウドベースの脅威分析サービスを含む既存のセキュリティインフラストラクチャではアラートが生成されていませんでした。セキュリティ管理の観点から、このシナリオをどのように分類しますか？また、SOCにとっての主な課題は何でしょうか？
**A.** 真陽性。コントロールは脅威を特定しましたが、SOCは対応できませんでした。課題はインシデント対応の実行です。
**B.** 誤検知。制御システムが過剰に警戒を発令したため、SOCは実際の脅威に対して鈍感になってしまった。問題は警戒疲れである。

**C.** False Negative; セキュリティ対策では実際の侵害を検出できませんでした。課題は、検出機能と脅威インテリジェンスの統合を改善することです。
**D.** 真陰性; 統制によって脅威は存在しないと正しく判定されました。課題は監査結果の検証です。
**E.** これは未知の状態であり、分類するにはさらなる調査が必要です。問題は可視性の欠如です。
**Answer: C (**メーッセージを残す**)**
これは典型的なFalse Negativeです。セキュリティ対策（Cortex XDR、WildFire）は、実際の悪意のあるイベント（侵害を検知できませんでした。主な課題は検知機能の強化です。これには、より包括的な脅威インテリジェンスの統合、既存の検知ルールの調整、追加の監視ツールの導入、あるいはシグネチャベースや基本的な異常検知を回避する高度でステルス性の高い攻撃を特定するための行動分析の改善などが含まれることがよくあります。

最新問題**: 64**
DNSトンネリングを悪用してデータ窃盗を行うことで知られる高度な攻撃グループが、貴社を標的としています。脅威インテリジェンスフィードには、特定のDNSクエリパターン（例：異常長いサブドメイン名、特定の文字セット、高いエントロピー）と、彼らが窃盗によく使用するリゾルバIPのリストが含まれています。この脅威インテリジェンスに合わせて正確に調整されたPalo Alto Networksファイアウォール機能の組み合わせは、どのようなものがこの高度な窃盗手法の検出と防御に最も効果的でしょうか？
**A.** リゾルバ IP に対して DNS シンクホールを有効にし、高エントロピー ドメインをブロックするようにカスタム URL フィルタリング プロファイルを構成します。
**B.** PCRE を使用してカスタム脅威防止 (IPS) シグネチャを実装し、DNS クエリ内の長くてエントロピーの高いサブドメイン パターンを検出し、DNS セキュリティの DGA 検出を利用するセキュリティ プロファイルを適用します。
**C.** リゾルバ IP 用のカスタム DNS 署名を使用してスパイウェア対策プロファイルを作成し、特定の長さを超える DNS クエリをブロックするカスタム データ フィルタリング プロファイルを展開します。
**D.** セキュリティ ポリシー内のリゾルバ IP に外部動的リスト (EDL) を使用し、すべての DNS トラフィックで疑わしいパターンを検査するように WildFire を構成します。
**E.** DNS トンネリング用のカスタム アプリケーション オーバーライドを展開し、大量の DNS トラフィックの優先順位を下げる QOS ポリシーを設定します。
**Answer: B (**メーッセージを残す**)**
この質問には、Palo Alto Networks の機能と、正確な脅威インテリジェンスを使用して特定の高度な脅威 (DNS トンネリング) に対してそれらの機能を効果的に組み合わせる方法についての深い理解が必要です。
オプション B は最も直接的かつ効果的な組み合わせを提供します。
PCRE を使用したカスタム脅威防止 (IPS) シグネチャ :これは、DNS クエリ内のトンネリングを示唆する特定のパターン（長いサブドメイン名、特定の文字セット、高いエントロピー）を検出するために不可欠です。PCRE は、DNS パケットのペイロード（流出したデータや C2 コマンドが存在する場所）との非常にきめ細かな照合を可能にします。

DNS SecurityのDGA検出（セキュリティプロファイルの一部）DGAは通常C2を指しますが、DNSトンネリングは動的に生成されたドメインに関係することがよくあります。Palo AltoのDNS Securityサービス（DGA検出機能を含む）は、通常のパターンから逸脱した不審なDNSクエリを識別し、Palo Altoの高度な分析機能を活用してカスタムIPSシグネチャを補完します。
この特定の脅威に対して他のオプションが最適ではない理由を分析してみましょう。
A (DNS シンクホール + URL フィルタリング): シンクホールは既知の悪意のあるドメイン/IP を対象としていますが、トンネリング パターンは検出しません。URL フィルタリングは HTTP/HTTPS に適用され、コンテンツ分析のための生の DNS クエリは直接適用されません。
C（カスタムアンチスパイウェアDNSシグネチャ + データフィルタリング）アンチスパイウェアDNSシグネチャは、既知の悪意のあるドメインをブロックすることが主な目的であり、クエリ自体のパターンマッチングには使用されません。データフィルタリングは、ネットワークから流出する機密データを対象としており、クエリ構造を解析して流出方法（DNSトンネリング）を検出するものではありません。長さによるブロックは、あまりにも曖昧で、誤検知が発生しやすくなります。
D（リゾルバIPのEDL + DNSトラフィックのWildFire）EDLは既知の不正IPをブロックするのに効果的ですが、DNSトンネリングでは多くのリゾルバが使用される可能性があります。WildFireは通常、トンネリングのためのDNSクエリ構造のディープパケットインスペクションではなく、ファイル分析とドメインレピュテーションに重点を置いています。
E (カスタムアプリケーションオーバーライド + QOS): アプリケーションオーバーライドは、プロトコル内の悪意のあるコンテンツを検出するのではなく、不明なアプリを分類するためのものです。QOS はトラフィックの優先順位を下げますが、トンネリングを防止または検出するものではありません。

最新問題: **65**
組織内で広く使用されている重要なWebサーバーアプリケーションに影響を与える新たなゼロデイ脆弱性が発表されました。CISOは、影響を受ける資産の特定、仮想パッチの適用、パッチの検証を含む、迅速かつ協調的な対応を求めています。Cortex XSIAM Playbookをどのように活用すれば、特にフローや他のコンポーネントとの相互作用に焦点を当てて、この課題を解決できるでしょうか？
**A.** 最初にパブリック フィードから脆弱性アドバイザリを取得し、次にネットワークをスキャンして特定の脆弱性を検出し、最後に SCCM 経由で自動パッチ展開をトリガーするプレイブックを設計します。
**B.** Cortex XDR に対してエンドポイント インベントリと脆弱性評価データを照会し、「仮想パッチ」アクションを使用して関連する Palo Alto Networks ファイアウォールに緩和ポリシーを適用し、その後「セキュリティ ポリシー検証」タスクを実行するプレイブックを作成します。
**C.** 主に XSIAM コンソールで優先度の高いアラートを作成することに重点を置いたプレイブックを実装し、IT 運用による手動パッチ展開用に Jira でチケットを生成します。
**D.** すべての Web サーバーで完全な侵入テストを実行し、結果を手動で確認してパッチ適用戦略を決定するプレイブック。

**E.** 分析と相関関係の確立のためにすべてのサーバー ログを外部 SIEM にエクスポートするプレイブックを構成し、アナリストが影響を受けるシステムを手動で特定するまで待機します。
**Answer: B (**メーッセージを残す**)**
オプションBは、Cortex XSIAMの統合機能を最大限に活用し、ゼロデイ攻撃への迅速な対応を実現します。XDRを用いて資産および脆弱性データを取得し、XSIAMのオーケストレーション機能を用いてPalo Alto Networksファイアウォール（一般的な仮想パッチ適用メカニズム）経由で仮想パッチを適用し、検証手順も備えています。オプションAは実現可能ですが、ゼロデイ攻撃においてしばしば重要となる、即時の仮想パッチ適用という側面を見逃してしまう可能性があります。オプションCは手動介入に依存しており、ゼロデイ攻撃には時間がかかりすぎます。オプションDは広範なテストプロセスであり、即時対応ではありません。オプションEはログ記録と分析手順であり、プロアクティブな修復ではありません。

最新問題**: 66**
ある大規模企業が、オンプレミスの仮想マシン（VM）インフラストラクチャの大部分をパブリッククラウドプロバイダー（AWS EC2、Azure VMなど）に移行しています。現在、オンプレミスのエンドポイント保護にはCortex XDRを使用しており、この保護範囲をクラウドVMにもシームレスに拡張したいと考えています。この企業は「クラウドファースト」のセキュリティ体制を敷いており、自動化されたスケーラブルな導入を目指しています。エージェントのインストールだけでなく、この動的なクラウド環境において、特にライフサイクル管理とコスト最適化の観点から、Cortex XDRエージェントを最適に管理・導入するためには、どのような高度な考慮事項や手法が重要でしょうか？
**A.** Cortex XDRエージェントを、新しいVMのデプロイメントに使用するゴールデンAMI（AWS）またはカスタムイメージ（Azure）に組み込み、エージェントが事前にインストールされていることを確認します。デプロイメント後のスクリプトを実装し、ワンタイム登録キーを使用してエージェントをCortex XDRに登録します。
**B.** クラウドネイティブのオーケストレーションツール (AWS Systems Manager、Azure Automation など) を利用して、インスタンスのブートストラッププロセスの一部として Cortex XDR エージェントをデプロイし、S3 バケットまたは Blob ストレージから最新のインストーラーを自動的に取得します。
**C.** Cortex XDR 内でタグベースの自動グループ割り当てを実装し、クラウド リソース タグ (例: 'Environment:Production'、'CostCenter:Finance') を XDR エンドポイント グループにマッピングして、ポリシーの適用と可視性を実現します。
**D.** Cortex XDR の「休止エンドポイントの自動削除」機能を活用し、短い休止期間を設定して、頻繁に終了される一時的なクラウド インスタンスからエージェントを自動的に登録解除し、ライセンスの過剰消費を防ぎます。
**E.** クラウド イベント (EC2 インスタンスの起動、VM の終了など) によってトリガーされるサーバーレス関数 (AWS Lambda、Azure Functions など) を開発し、XDR API を介してプログラムで Cortex XDR エージェントをインストール/アンインストールし、インスタンスの実行中にのみエージェントがアクティブになるようにします。

**Answer: (**解答を表示する**)**
この質問は、クラウド導入における高度で重要な考慮事項について問いかけています。A: ゴールデンイメージへの組み込み :これはクラウド導入における基本的かつ非常に効率的な方法です。エージェントを事前にインストールすることで、バージョンの一貫性が確保され、起動後のオーバーヘッドが削減されます。導入後のスクリプト（cloud-init、user data など）が特定のテナント登録を処理します。B: クラウドネイティブオーケストレーション :エージェント導入にはAWS Systems ManagerまたはAzure Automationを使用するのがベストプラクティスです。クラウド環境において、一元管理、パッチコンプライアンス、スケーラブルな導入機能を提供します。C: タグベースのグループ割り当て :クラウド環境では、リソース管理、コスト配分、セキュリティのためにタグ付けに大きく依存しています。これらのタグをCortex XDRグループにマッピングすることで、動的なポリシー適用と可視性の向上が実現し、クラウドファーストのセキュリティ体制に適合します。D: 休止エンドポイントの自動削除：一時的なクラウドインスタンスは、エージェントベースのライセンス管理における一般的な課題です。この機能は、終了したインスタンスからエージェントを自動的に登録解除し、ライセンスの「漏洩」を防ぐことで、ライセンスを効果的に管理するために不可欠です。E: API駆動型ライフサイクルのためのサーバーレス関数：技術的には可能ですが、エージェントのインストール/アンインストールイベントごとにカスタムサーバーレス関数を構築・維持するのは非常に複雑であり、標準的なXDRエージェントライフサイクル管理では一般的に不要です。ネイティブのクラウドオーケストレーションツールとXDRの組み込み機能（休止エンドポイントの削除など）で通常は十分です。XDRエージェントはインスタンスの終了を適切に処理するように設計されています。これは通常、高度にカスタマイズされた要件やニッチな要件に対応する高度なユースケースであり、最適な管理のための「重要な」一般的な考慮事項ではありません。

最新問題**: 67**
セキュリティアナリストは、Cortex XDRによって管理されているエンドポイント上の不審なプロセスを調査しています。プロセスsvchost.exeは、既知の悪意のあるC2サーバーへの接続を試みており、異常なネットワーク動作を示しています。このネットワークアクティビティの検出と報告を主に担っているCortex XDRの主要なセンサー要素はどれですか？また、別途ネットワークタップを必要とせずに、どのようにしてこれを実現しているのでしょうか？
**A.** プロセス メモリに挿入されたシェルコードを分析する、Behavioral Threat Protection (BTP) エンジン。
**B.** ローカル分析エンジンは、svchost.exe バイナリの PE ヘッダーに対して静的分析を実行します。
**C.** エンドポイント センサーのネットワーク監視モジュール。オペレーティング システムのネットワーク スタック (Windows の場合は Winsock LSP、macOS の場合は kext など) に接続して、カーネル レベルでネットワーク接続を監視および報告します。
**D.** 疑わしいネットワーク トラフィック パケットをサンドボックスに送信することによる WildFire 統合。

**E.** ファイアウォールとプロキシからのログ データを相関させることによって生成されるデータレイク。
**Answer: C (**メッセージを残す**)**
エンドポイントセンサーのネットワーク監視機能は、不審なネットワークアクティビティの検出に不可欠です。この機能は、オペレーティングシステムのネットワークスタックと緊密に統合することで実現され、外部ネットワークタップを必要とせずに、エンドポイントからネットワーク接続、DNSクエリ、その他のネットワーク関連イベントを直接監視・報告できます。オプションAとBは他のセンサー機能（動作分析、静的分析）に関連し、オプションDとEはクラウドベースのサービスとデータ集約に関するものであり、エンドポイントでのライブネットワーク監視を担う主要なセンサー要素ではありません。

最新問題: **68**
SOCは、カスタムビルドのRPC（リモートプロシージャコール）ベースのXMLAPI経由でのみデータを提供する独自の社内資産管理データベース（AMDB）を統合する必要があります。Cortex XSOARは、インシデントエンリッチメント中にこのAMDBに対して資産の詳細を照会し、資産ステータスを更新する必要があります。この独自のAPIを考慮すると、この統合を構築するのに最適なXSOARアプローチはどれでしょうか？また、主な技術的課題は何でしょうか？
**A.** XSOAR の Generic Webhook 統合を使用します。課題: RPC ベースの API は Webhook と互換性がありません。
**B.** 次のようなライブラリを使用してカスタムPython統合を開発します。
**C.** XSOARの汎用HTTP統合を設定し、コンテンツタイプを「application/xml」に設定します。課題：HTTP統合はRESTful API向けに設計されており、RPCベースのXML構造では扱いにくい場合があります。
**D.** RPC XMLをRESTに変換できるサードパーティ製APIゲートウェイをインストールし、XSOARをAPIゲートウェイに統合します。課題：インフラストラクチャのオーバーヘッドが大幅に増加し、新たな障害点が発生します。
**E.** AMDBからCSVファイルにデータを手動でエクスポートし、フィードとしてXSOARに取り込みます。課題：データが古くなり、リアルタイムのクエリや更新が実行できない。
**Answer: B (**メッセージを残す**)**
オプション B は正しく、最も効果的なアプローチです。RPC ベースの XML API は非常に特殊であり、汎用的な REST または SOAP 統合ではカバーされていないため、カスタム Python 統合が必要です。Python の柔軟性により、xmlrpc.client (XML-RPC の場合) などのライブラリ、または完全に独自の RPC の場合は raw ソケット プログラミングを使用して、このような API と直接対話できます。開発者は、特定の XML 要求を構築して送信し、XML 応答を解析し、カスタム認証やセッション管理を処理するためのコードを記述します。リストされている課題は、高度にカスタマイズされた非標準 API との統合に固有のものです。オプション A、C、および E は、RPC XML API とは互換性がないか、効果がありません。オプション D は有効な技術的ソリューションですが、外部の複雑さが生じます。一方、XSOAR の拡張性は、可能な限りこのようなロジックをプラットフォーム内に保持することを目的としています。

最新問題: **69**
Palo Alto Networks（PAN-OS）次世代ファイアウォールを採用したセキュリティオペレーションセンター（SOC）は、新たに取り込んだ脅威情報フィードによってコマンドアンドコントロール（C2）サーバーとして特定された不審なIPアドレス（192.0.2.10）への内部ホストからのアウトバウンド接続を検出しました。また、フィードには、このC2が「Cobalt Strike」攻撃グループと関連していることも示されています。インシデント対応の初期封じ込めフェーズにおいて、主に脅威情報に基づいて実施すべき、以下の緊急対応策のうち最も重要なものはどれですか？
**A.** 192.0.2.10 との間のすべてのトラフィックに対して、ファイアウォール上で完全なパケットキャプチャを開始します。
**B.** ファイアウォール ポリシーを変更して内部ホストをネットワークから分離し、すべての送信接続をブロックします。
**C.** ファイアウォールの WildFire サブスクリプションを更新して、最新のマルウェア シグネチャが適用されていることを確認します。
**D.** 侵害された内部ホストに対して脆弱性スキャンを実行し、潜在的なエントリ ポイントを特定します。
**E.** 検出された C2 通信について直ちに法執行機関に通知します。
**Answer: B (**メーッセージを残す**)**
封じ込めフェーズでは、インシデントの範囲を限定することを最優先します。C2 IPアドレスと、Cobalt Strikeのような高度な攻撃者との関連性に関する脅威インテリジェンスを調査するには、さらなる侵害やデータ流出を防ぐために、侵害を受けたホストを直ちに隔離する必要があります。その他の選択肢も重要ですが、それらは異なるフェーズ（例：根絶分析、インシデント後）に属し、封じ込めにはそれほど即時性がありません。完全なパケットキャプチャ（A）は分析用、WildFireアップデート（C）はプロアクティブ、脆弱性スキャン（D）は根絶用、法執行機関への通知（E）は後続のステップです。

最新問題: **70**
SOCは、従来のSIEMからクラウドネイティブのセキュリティ運用プラットフォームへの移行を進めており、特にPalo Alto Networks Cortex XSOARの統合機能を評価しています。主な目的は、脅威インテリジェンスによるアラートの拡充、侵害されたエンドポイントの封じ込め、インシデントレポートの生成といった、反復的なインシデント対応タスクを自動化することです。XSOAR機能のグローバルオブジェクトを「demisto」、現在のインシデントオブジェクトを「incident」と仮定した場合、Cortex XSOARのカスタムプレイブックに統合した際に、外部APIからの脅威インテリジェンスによるアラートの拡充を自動化する例として、次のPythonコードスニペットはどれですか？

```
import requests

def enrich_ip_address(ip_address):
    response = requests.get(f'https://api.threatintel.com/v1/ip/{ip_address}')
    return response.json()
```

**A.**

```
incident.add note(f'IP enrichment: {enrich ip address(incident.get("ip address"))}')
```

**B.**

```
import subprocess

def block_ip_on_firewall(ip_address):
    subprocess.run(['firewall-cmd', '--permanent', '--add-rich-rule="rule family=\"ipv4\" source address=\"{ip_address}\" drop"'], check=True)
    subprocess.run(['firewall-cmd', '--reload'], check=True)

block_ip_on_firewall(incident.get('source_ip'))
```

**C.**

```
port demisto_client.demisto_api

f get_threat_intelligence(indicator_value, indicator_type):
  # Assume a pre-configured 'VirusTotal' integration instance
  results = demisto.executeCommand('vt-get-indicator', {'indicator': indicator_value, 'indicator_type': indicator_type})
  if demisto.results():
      return demisto.results()
  return None
Get an IP address from the incident context
ncident_ip = demisto.incident()['labels'][0]['value'] if 'labels' in demisto.incident() and demisto.incident()['labels'] else No

f incident_ip:
   ti_data = get_threat_intelligence(incident_ip, 'IP_ADDRESS')
   if ti_data:
       demisto.results({'Type': 1, 'ContentsFormat': 'json', 'Contents': ti_data})
```

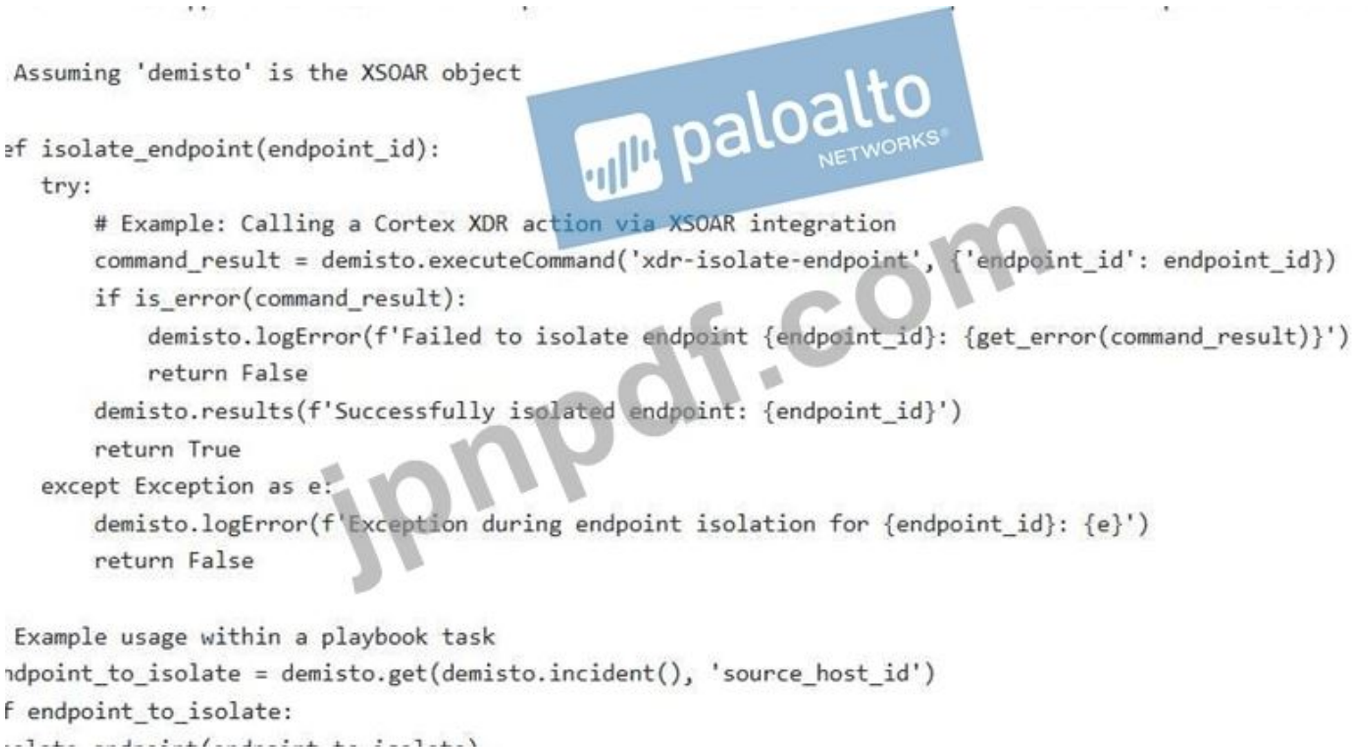
**D.**

```
from datetime import datetime

def generate_report_header(incident_id):
    return f'Incident Report - Incident ID: {incident_id}\nDate: {datetime.now().strftime("%Y-%m-%d %H:%M:%S")}'

print(generate_report_header(incident.get('id')))
```

**E.**



```
Assuming 'demisto' is the XSOAR object

ef isolate_endpoint(endpoint_id):
   try:
       # Example: Calling a Cortex XDR action via XSOAR integration
       command_result = demisto.executeCommand('xdr-isolate-endpoint', {'endpoint_id': endpoint_id})
       if is_error(command_result):
           demisto.logError(f'Failed to isolate endpoint {endpoint_id}: {get_error(command_result)}')
           return False
       demisto.results(f'Successfully isolated endpoint: {endpoint_id}')
       return True
   except Exception as e:
       demisto.logError(f'Exception during endpoint isolation for {endpoint_id}: {e}')
       return False

Example usage within a playbook task
ndpoint_to_isolate = demisto.get(demisto.incident(), 'source_host_id')
f endpoint_to_isolate:
```



**Answer: C,E (メーッセージを残す)**

これは、SOAR 自動化と Palo Alto Networks XSOAR の詳細に関する知識を必要とする複数回答の質問です。オプション C (正解): このスニペットは、Cortex XSOAR 内の Python スクリプト( 「demisto.executeCommand」を使用) が、事前設定された統合 (VirusTotal など) を呼び出してインジケーターを強化し、次に 「demisto.resultS」と 「demisto.setContext」を呼び出してインシデント内でデータを利用できるようにする方法を正しく示しています。これは、質問の 「脅威インテリジェンスによるアラートの強化」の部分に直接対応しています。オプション E (正解): このスニペットは、XSOAR を使用して、統合 EDR ソリューション (Cortex XDR など) からアクションを呼び出すことにより、「侵害されたエンドポイントの封じ込め」タスクを自動化する方法を正しく示しています。これは、SOAR のコア機能です。オプション A: これは 「リクエスト」を直接使用しますが、これは通常、XSOAR の組み込み統合やプレイブックが外部 API と対話する方法ではありません。XSOARは、統合インタラクションに 「demisto.executeCommand」を推奨します。オプションB :これは 「subprocess.run」を使用してシェルコマンドを実行しますが、これはシステムに大きく依存し、SOARプラットフォームを介してネットワークデバイスとインタラクトするための標準的、安全、かつ移植性の高い方法ではありません。XSOARは、このために特定のファイアウォール統合を使用します。オプションD :これはレポートヘッダーのみを生成し、完全なレポートは生成しません。また、エンリッチメントやコンテインメントの自動化は含まれません。レポート生成はSOARの機能ですが、このコードスニペットは単純すぎて、自動化の主要な目的に対応していません。質問は、エンリッチメントやコンテインメントといった反復的なインシデント対応タスクの自動化、そしてインシデントレポート (ヘッダーだけでなく)の生成を求めています。

**最新問題: 71**

セキュリティオペレーションセンター（SOC）のアナリストが、ラテラルムーブメントの疑いのあるインシデントを調査しています。Cortex XDRは、侵害されたエンドポイントから発信された不審なPowerShellアクティビティを示すアラートを発令しました。アナリストは、侵害の範囲を迅速に把握し、特に攻撃者が窃取した認証情報を使用してアクセスした可能性のある他のシステムを特定する必要があります。攻撃者の経路を効率的に追跡し、影響を受けた資産を特定するには、Cortex XDRのどの主要要素を組み合わせることが最も重要でしょうか？
**A.** エンドポイント エージェント (プロセス、ネットワーク接続) からのテレメトリ データとユーザー 行動分析 (UBA) データ。
**B.** ネットワーク接続ログ (NetFlow)、ファイアウォール ログ、脅威インテリジェンス フィード。
**C.** ユーザー アクティビティ ログ (ログオン、グループの変更)、資産インベントリ、脆弱性スキャンの結果。
**D.** ファイル アクティビティ ログ、DNS クエリ、および電子メール ゲートウェイ ログ。
**E.** クラウド アクセス ログ、SaaS アプリケーション ログ、エンドポイント フォレンジック イメージ。
**Answer: (**解答を表示する**)**
SOCアナリストは、ラテラルムーブメントを追跡し、影響を受ける資産を特定するために、エンドポイントのアクティビティとユーザーの行動の両方について詳細な洞察を得る必要があります。Cortex XDRエージェント（プロセス、ネットワーク接続、ファイルアクセス）からのテレメトリデータは、侵害されたエンドポイントで何が起こったか、他のシステムとどのように通信したかに関する基本的な可視性を提供します。Cortex XDRの分析エンジンを搭載したユーザー行動分析（UBA）データは、盗まれた資格情報を使用したラテラルムーブメントの主要な指標である、異常なユーザーログオン、資格情報の使用パターン（対話型ログオンでのサービスアカウントの使用など）、および通常とは異なるリソースへのアクセスを明らかにできます。オプションB、C、D、およびEは貴重なデータを提供しますが、特にCortex XDRの統合機能を考慮すると、資格情報の再利用による攻撃者の経路の追跡や、ラテラルムーブメントのコンテキストにおける侵害されたシステムの特定という、当面のタスクに直接的に焦点を当てているとは言えません。

最新問題**: 72**
広く普及している仮想化プラットフォームに重大なゼロデイ脆弱性が発見され、活発な悪用が確認されています。Palo Alto Networksのお客様である貴社は、この脆弱性の具体的なメモリ破損パターンとネットワークビーコンの特徴を詳述した緊急の脅威インテリジェンス速報を受け取りました。貴社は、カスタム検出メカニズムを迅速に導入する必要があります。Palo Alto Networksの機能を活用し、この特定のゼロデイ脆弱性に対するPalo Alto Networksの公式シグネチャ更新への依存を最小限に抑えながら、最も迅速かつ効果的な保護を提供できるアプローチはどれでしょうか？
**A.** ネットワーク ビーコン特性に基づいたカスタム スパイウェア対策シグネチャと、メモリ破損パターン用のカスタム脆弱性保護シグネチャを開発します。

**B.** PCRE (Perl 互換正規表現) を使用してカスタム脅威防止 (IPS) シグネチャを構成し、ネットワーク トラフィック内のメモリ破損パターンを検出し、ビーコン C2 IP のカスタム外部動的リスト (EDL) を作成します。
**C.** Cortex XDR の Behavioral Threat Protection を活用して、エクスプロイト後のアクティビティを検出し、エクスプロイト ペイロードに対して WildFire にカスタム YARA ルールを展開します。
**D.** エクスプロイト トラフィックを識別するためのカスタム アプリケーション オーバーライドと、既知の C2 ドメインをブロックするためのカスタム URL フィルタリング プロファイルを作成します。
**E.** エクスプロイトのサンプルを WildFire に送信して分析し、新しいシグネチャが利用可能になったら脅威防止プロファイルを更新します。
**Answer: B (**メーッセージを残す**)**
このシナリオでは、公式ベンダー署名がリリースされる前に、ゼロデイ攻撃に対する即時のカスタム保護に重点を置いています。
*オプション B (カスタム IPS シグネチャ + EDL): これは最も効果的で即時的なアプローチです。o PCREによるカスタム脅威防止（IPS）シグネチャ：PCREは、ネットワークトラフィック内で非常にきめ細かく複雑なパターンマッチングを可能にするため、特定の脆弱性シグネチャがなくても、ネットワーク上で発生する特定のメモリ破損パターンを検出するのに最適です。これにより、「仮想パッチ適用」が可能になります。o C2 IP用のカスタム外部動的リスト（EDL）EDLは、脅威インテリジェンスによって特定された悪意のある新しいIPおよびドメインを迅速かつ動的にブロックできるため、既知のC2インフラストラクチャへのビーコン送信を効果的に防止します。
他のものを調べてみましょう:
*A (カスタムアンチスパイウェア/脆弱性保護): 技術的には可能ですが、ベンダー提供のフォーマットを使用せずにゼロデイ攻撃用の特定のシグネチャタイプをゼロから作成することは、カスタムIPSシグネチャよりも複雑で柔軟性に欠ける可能性があります。IPSはエクスプロイト検出を目的として設計されています。
*C (Cortex XDR Behavioral + WildFire YARA): Cortex XDRのBehavioral Protectionはエクスプロイト後の保護には優れていますが、質問ではエクスプロイトの防止について尋ねられています。WildFire YARAルールはファイルベースの分析を目的としており、ネットワークレベルのエクスプロイトパターンを直接ブロックするものではありません。
*D（カスタムアプリケーションオーバーライド + URLフィルタリング）アプリケーションオーバーライドは未知のアプリケーションを分類するためのものであり、エクスプロイトパターンを検出するためのものではありません。URLフィルタリングはドメイン/URLをブロックするためのものであり、トラフィック内のメモリ破損パターンを検出するためのものではありません。2026/1/152026/1/152026/1/15*E (WildFireへのサンプル提出): 長期的な保護には不可欠ですが、これは事後対応的なステップです。質問では、正式な署名の前に即時の保護を求めています。

最新問題**: 73**
貴社ではCortex XSIAMを使用しており、機密データに影響を与える重大度の高いインシデント（特定のタグ「sensitive_data_impact」で分類されるもの）が発生した場合、直ちに堅牢なデータ漏

洩防止（DLP）ワークフローをトリガーするという厳格なポリシーを定めています。このワークフローには、1) 専任の「データインシデント対応」チームへのインシデントのエスカレーション、2) 関連するすべての証拠の安全で変更不可能なストレージへのアーカイブ、3) 監査用の特定フィールドを含むコンプライアンスレポートの生成、4) 影響を受けるユーザーアカウントに対する法的保留の開始が含まれます。この多面的で信頼性の高いDLPワークフローを効果的に実装するために不可欠な、Cortex XSIAM Playbookのコンポーネントと設計原則をすべて選択してください。

**A.** プレイブックの先頭で「条件付き」タスクを使用して「sensitive_data_impact」タグをチェックし、必要な場合にのみ DLP ワークフローが実行されるようにします。

**B.** 「並列」タスクを使用して、「データ インシデント対応」チームへのエスカレーション (チケットシステムとの統合など) を同時にトリガーし、証拠のアーカイブ プロセス (安全なクラウド ストレージ API との統合など) を開始します。

**C.** プレイブック タスク内にカスタム JavaScript 自動化スクリプトを実装し、インシデント データを取得して定義済みテンプレートを入力し、それを SharePoint サイトにアップロードすることで、コンプライアンス レポートを動的に構築します。

**D.** プレイブック タスク内に組み込まれた「Active Directory」または「HR システム」統合を活用して、法的保留通知のユーザーのマネージャーを識別し、「ServiceNow」統合を使用して法的保留要求チケットを開始します。

**E.** データの機密性を考慮し、人間による監視と承認を確実に行うために、DLP ワークフローの各ステップで「手動タスク」のみに依存します。

**Answer: (**解答を表示する**)**

オプション A、B、C、D はすべて、Cortex XSIAM でこのような堅牢で保証度の高い DLP ワークフローを実装するために不可欠であり、高度なプレイブック機能を示しています。A (条件付きタスク): 絶対に重要です。これにより、複雑な DLP ワークフローが 'sensitive_data_impact' 基準を本当に満たすインシデントに対してのみトリガーされ、不要な実行や誤報が防止されます。B (並列タスク): 効率に不可欠です。エスカレーション、アーカイブ、コンプライアンス レポートはほぼ同時に実行できるため、重大度の高いインシデントへの対応時間が大幅に短縮されます。ここで鍵となるのは、XSIAM の並列タスク機能です。C (コンプライアンス レポートのカスタム スクリプト): 動的なデータと特定の書式設定要件を伴う非常に特殊なコンプライアンス レポートの場合、標準の統合では提供されないデータの取得、処理、書式設定を行うために、カスタム スクリプト (JavaScript など) が必要になることがよくあります。D（リーガルホールドのための組み込み統合）：既存の統合（マネージャー向けのAD/HR、リーガルホールドリクエスト向けのServiceNow）を活用することで、リーガルホールドプロセスの重要な部分を自動化し、既存のIT/法務ワークフローに連携させることができます。E（手動タスク）：このオプションは正しくありません。手動タスクのみに依存すると、重大度の高いポリシー主導の要件に対する自動インシデント対応の目的が達成されず、遅延や人的ミスが発生する可能性があります。一部のレビュー手順は手動で行う場合もありますが、コアとなるトリガーと実行は自動化する必要があります。

最新問題**: 74**

脅威ハンターは、潜在的なLiving Off The Land（LOTL）攻撃を調査しています。この攻撃では、攻撃者が正規のシステムツールを悪意のある目的で使用し、具体的にはPowerShellスクリプトを実行して永続性を確立している疑いがあります。Palo Alto Networksファイアウォールは、Cortex XDRを介してエンドポイントからのプロセス情報を記録するように設定されており、これらのログはSIEM（Splunk）に取り込まれます。ハンターは、「cmd.exe」が疑わしいコマンドライン引数を使用して「powershell.exe」を生成し、悪意のあるスクリプトをエンコードしている可能性のあるインスタンスを特定したいと考えています。Cortex XDRエンドポイントデータを利用する以下のSplunkクエリのうち、これらの隠蔽またはエンコードされた悪意のあるアクティビティを明らかにするのに最も効果的なものはどれですか？

**A.**

```
sourcetype=cortex_xdr_events event_type=process | where process_name="powershell.exe" AND parent_process_name="cmd.exe" | search command_line IN
(" -EncodedCommand ", " -Decode ", " IEX ", " b64 ", " FromBase64String ") | stats count by host_name, command_line | sort -count
```

**B.**

```
sourcetype=cortex xdr events event type=process AND process name="cmd.exe" AND child process name="powershell.exe" | regex command_line=".
(?:-e|-en|-enc|-enco|-encod|-encode|FromBase64String|IEX). " | stats count by host_name, user, command_line | sort -count
```

**C.**

```
sourcetype=cortex xdr events event type=process AND parent process name="cmd.exe" AND process name="powershell.exe" | eval suspicious command =
case(like(command_line, "%-EncodedCommand%") OR like(command_line, "%-e%") OR like(command_line, "%FromBase64String%") OR like(command_line,
"%IEX%"), "true", 1=1, "false") | where suspicious_command="true" | table _time, host_name, user, command_line | sort _time desc
```

**D.**

```
sourcetype=cortex_xdr_events event_type=process | eval parent_process_name = lower(parent_process_name), process_name = lower(process_name),
command_line = lower(command_line) | where parent process name="cmd.exe" AND process name="powershell.exe" AND (match(command line,
"-encodedcommand") OR match(command_line, "frombase64string") OR match(command_line, "iex")) | stats count by host_name, user, command_line |sort -count
```

**E.**

```
sourcetype=cortex_xdr_events event_type=process AND process_name="powershell.exe" AND parent_process_name="cmd.exe" |
`detect_encoded_powershell_macro` | stats count by host_name, command_line | sort -count
```

**Answer: (解答を表示する)**

この質問は、一般的な LOTL 手法である、エンコードされた PowerShell コマンドの検出を対象としています。C と D はどちらも非常に効果的です。オプション C は、柔軟なパターン マッチングのために 'eval' を 'case' および 'like' と共に使用し、特に難読化の一般的な指標 (C-EncodedCommancf、FromBase64String、'IEX') を探します。これは、疑わしいアクティビティに対してブール フラグを作成してフィルタリングする堅牢な方法です。オプション D は、大文字と小文字を区別しないことを保証するために 'lower()' を使用し、疑わしいキーワードに対して OR 条件で 'match()' を使用します。これも非常に効率的で堅牢なアプローチです。オプション A は、ワイルドカードで 'SIN' を使用しますが、精度が低く、バリエーションを見逃す可能性があります。オプション B は強力な 'regex' を使用しますが、正規表現は '-e' などに対して精度が低く、正当な短いフラグと一致する可能性があります。オプション E は未定義のマクロに依存しています。

最新問題: **75**

インシデント対応において、プレイブックはサードパーティの脅威インテリジェンスプラットフォーム（TIP）から複数の指標のレピュテーションスコアを動的に取得する必要があります。指標の数はインシデントごとに異なります。プレイブックはこれらのスコアに基づいて次のアクションを決定する必要があります。単一のインシデントフロー内でレピュテーションの取得、結果の処理、条件付き意思決定を行うのに最適なXSOARコンポーネントはどれでしょうか？

**A.** 5 分ごとに実行され、TIP で新しいインジケーター データをポーリングするように構成されたスタンドアロン ジョブ。
**B.** プレイブック内のタスクとして実行される Python スクリプト。TIP 統合を活用してデータを取得し、意思決定のための条件付きロジックが含まれています。
**C.** TIP と直接対話し、評判に基づいてインシデント フィールドを更新するスケジュールされたスクリプト。
**D.** 自動化ルール。各インジケーターの評判を取得するために個別のプレイブックをトリガーします。
**E.** この動的なレピュテーション検索専用に構築されたカスタム統合。バックグラウンド サービスとして実行されます。
**Answer: B (**メーッセージを残す**)**
プレイブック内でタスクとして実行されるPythonスクリプトが最適です。スクリプトは、特定のロジックをカプセル化し、統合（TIP統合など）と連携し、データを処理し、プレイブックの実行コンテキスト内で結果を返すように設計されています。これにより、インシデント固有のデータに基づいて、動的な取得、処理、条件分岐をすべてインシデントのワークフロー内で実行できます。

最新問題**: 76**
ある大手製造会社は、ITネットワークから分離された重要なOT（オペレーショナルテクノロジー）ネットワークを運用しています。OTデバイスはインターネットへの直接アクセスが制限されている一方で、サプライチェーン攻撃やIT-OTコンバージェンスは重大なリスクをもたらします。既存のEDRはITエンドポイントに導入されていますが、独自のOTプロトコルや特殊な産業用制御システム内のイベントを監視・対応することはできません。このシナリオにおいて、Cortex XDRのどのような独自の機能が、他のPalo Alto Networks製品と組み合わせる際に重要となるでしょうか？
**A.** 一般的な IT ソフトウェアのゼロデイ脆弱性を予測するためのディープラーニング機能。
**B.** ネットワーク トラフィック分析 (NTA) および IoT/OT セキュリティ ソリューション (Zingbox/IoT Security など) と統合して、IT および OT ドメイン全体にわたって統合された可視性と脅威検出を提供する機能。
**C.** IT ネットワーク内の従来の Windows および Linux サーバーのエンドポイント保護にのみ焦点を当てます。
**D.** ランサムウェア攻撃が発生した場合に、すべての OT デバイスの包括的なバックアップ イメージを自動的に作成します。
**E.** 従来の PLC (プログラマブル ロジック コントローラー) デバイスに直接インストールできる EDR エージェントを提供します。
**Answer: B (**メーッセージを残す**)**
この質問は、特にOTのような特殊な環境におけるXDRの「拡張」側面を浮き彫りにしています。EDRは従来のITエンドポイントに限定されていますが、Palo Alto Networksエコシステムの一部であるCortex XDRは、ネットワークトラフィック分析（NTA）や専用のIoT/OTセキュリティソリューション（買収したZingboxなど、現在はIoT Securityに統合されています）と統合できます。この統合により、Cortex XDRはITネットワークとOTネットワークからデータを取り込み、相関分析

することで、両ドメインにわたる包括的な脅威検知と対応が可能になります。これは、OTプロトコルの理解とセンサー機能を備えていないスタンドアロンのEDRでは不可能です。



最新問題: **77**
Cortex XSIAMでセキュリティインシデント「MalwareDetectedOnEndpoint」がトリガーされました。関連するプレイブック「P -malware-Response」
が開始されます。アナリストは、プレイブックによってエンドポイントの隔離は成功したものの、影響を受けたエンドポイントからのネットワーク接続の問題により、後続の「脅威インテリジェンスのファイルハッシュの取得」タスクが失敗したことを確認しました。次のタスク「脅威インテリジェンスプラットフォームの確認」は依存タスクです。このようなシナリオにおいて、回復力と効果的な進行を確保するために、最も適切なプレイブックの設計または運用上の考慮事項は何でしょうか？
**A.** 以前の失敗に関係なく、後続のすべてのステップが確実に試行されるように、すべてのタスクに対して「エラー時に続行」するようにプレイブックを設計する必要があります。
**B.** 「Threat Intel のファイル ハッシュの取得」タスクには再試行メカニズムを設定する必要があり、「Threat Intelligence プラットフォームの確認」タスクはハッシュ取得タスクが成功した場合にのみ実行される「条件付き」タスクとして設計する必要があります。
**C.** タスクが失敗するとプレイブックは直ちに終了し、SOC アナリストに手動で介入するよう警告する必要があります。
**D.** ネットワークの問題がよく発生し、自動化を妨げる可能性があるため、「Threat Intel のファイル ハッシュの取得」タスクはプレイブックから削除する必要があります。
**E.** プレイブック内のすべてのタスクは「オプション」としてマークし、重要なデータ収集手順が失敗した場合でもプレイブックを完了できるようにする必要があります。
**Answer: B (**メーッセージを残す**)**
オプションBは、レジリエンス（回復力）のための堅牢なプレイブック設計を示しています。再試行メカニズムは、ネットワーク接続などの一時的な問題に対処します。「脅威インテリジェンスプラットフォームの確認」をハッシュの取得成功に依存する「条件付き」タスクにすることで、プレイブックが不完全なデータで続行されることを防ぎ、他の独立した成功したアクション（隔離など）は有効にすることができます。オプションAは、不完全または不正確な情報で続行される可能性があります。オプションCは過度に積極的であり、自動化のメリットを低下させます。オプションD

は重要なステップを削除します。オプションEは、インシデント処理が不完全になる可能性があります。

最新問題: **78**
セキュリティエンジニアは、Windows Management Instrumentation (WMI) を悪用して PowerShell スクリプトをリモートで実行し、イベントログを即時に削除するという、新たなラテラルムーブメント手法を検出するためのカスタム Cortex XSIAM BIOC ルールを作成するという任務を負っています。BIOC は、WMI プロセス (wmiprvse.exe) が PowerShell プロセスを生成し、その PowerShell プロセスが 60 秒以内に同一ホスト上で「Clear-WinEventLog」を含むコマンドを実行した場合にインシデントをトリガーする必要があります。このような BIOC 定義の一部となる適切な XQL (Cortex Query Language) スニペットを選択してください。

**A.**
```
dataset = xdr_data | filter event_type = PROCESS and (action_process_image_name = 'wmiprvse.exe' or action_process_image_name = 'powershell.exe') | join kind=inner (dataset = xdr_data | filter event_type = PROCESS and action_process_image_name = 'powershell.exe' and action_process_command_line contains 'Clear-WinEventLog') on (action_host_name, action_process_start_time in (60s, -60s)) | limit 1
```

**B.**
```
dataset = xdr_data | filter event_type = PROCESS | alter wmi_event = (action_process_image_name = 'wmiprvse.exe'), powershell_clear_event = (action_process_image_name = 'powershell.exe' and action_process_command_line contains 'Clear-WinEventLog') | event_sequence by action_host_name maxspan=60s | sequence 'wmi process_start' = wmi_event, 'powershell_clear' = powershell_clear_event | where 'wmi_process_start' and 'powershell_clear' | limit 1
```

**C.**
```
dataset = xdr_data | filter event_type = NETWORK and action_network_protocol = 'WMI' and action_network_target_port = 5985 | group by action_host_name | aggregate count() | where count_ > 5
```

**D.**
```
dataset = xdr_data | filter event_type = FILE and action_file_name = 'wmiprvse.exe' and action_file_operation = 'delete'
```

**E.** BIOC は GUI のみで定義され、XQL は使用されないため、上記のいずれにも該当しません。

**Answer: B (**メーッセージを残す**)**
この質問は、動作検出のための BIOC と XQL の理解をテストします。オプション B は、指定された時間枠 ( ) 内でのイベント シーケンスの順序 ( ) を正しく使用し、同じホスト ( ) に ( ) 関連付けられています。これは、説明されている横方向の移動と回避手法を正確に捉えています。オプション A は、結合である を使用していますが、オプション C はプロセス イベント シーケンスではなくネットワーク アクティビティ用であり、シーケンスをキャプチャしません。オプション D は「wmiprvse.exe」のファイル削除を検索しますが、これはシナリオには当てはまりません。オプション E は、XSIAM でのカスタム ルール作成に XQL が不可欠であるため、正しくありません。

最新問題: **79**
Cortex XSIAM Playbook の重要な機能は、インシデントやインジケーターからコンテキストを活用できることです。インシデントは、「新しい地域からの稀なログイン」アラートに基づいてトリガーされます。関連するプレイブックでは、次の処理を実行する必要があります。1) ユーザーの人事データ（部門、マネージャーなど）でインシデントを拡充する。2) ユーザーが現在、その地域への承認済み出張中であるかどうかを確認する。3) そうでない場合は、多要素認証（MFA）チャレンジを開始する。適切な統合が構成されていることを前提として、Cortex XSIAM Playbook 内でエンリッチメントと条件付き MFA チャレンジを実現する方法を正しく示しているコードスニペットと概念的アプローチはどれですか？

**A.**

```
Task: Enrich User Data
ActiveDirectory-GetUser email='{{incident.user.email}}'

Task: Check Travel Status
TravelManagementSystem-IsUserOnTravel user='{{incident.user.username}}' geo='{{incident.source_ip_geo}}

Task: Initiate MFA Challenge (always runs)
Duo-InitiateMFA user='{{incident.user.username}}'
```

**B.**

```
# Task: Enrich User Data (HTTP Call to HR API)
!http-request url='https://hr.example.com/api/users?email={{incident.user.email}}' method='GET'

# Task: Conditional Branching
# Condition: Check if incident.user.hr_data.on_travel == 'false' AND incident.user.ip_geo != incident.user.travel_geo

# Task: Initiate MFA Challenge (inside conditional branch)
!Duo-InitiateMFA user='{{incident.user.username}}'
```

**C.**

```
# Task: Enrich User Data
!CortexXDR-GetEndpointDetails endpoint_id='{{incident.endpoint_id}}'

# Task: Initiate MFA Challenge (always runs)
!Okta-SendPushNotification user='{{incident.user.email}}'
```

**D.**

```
# Task: Enrich User Data (Manual Task)
- description: 'Analyst to manually lookup HR data for {{incident.user.email}}'

# Task: Check Travel Status (Manual Task)
- description: 'Analyst to manually verify travel plans'

# Task: Initiate MFA Challenge (Manual Task)
- description: 'Analyst to manually trigger MFA'
```

**E.**

```
# Task: Check Threat Intel for IP
!VirusTotal-IPReport ip='{{incident.source_ip}}'

# Task: Initiate MFA Challenge (always runs)
!Okta-InitiateMFA user='{{incident.user.username}}'
```

**Answer: (**解答を表示する**)**

オプションBは、アプローチを正しく概念化しています。エンリッチメントには、多くの場合、内部システム（HR APIなど）へのHTTPリクエストや専用の統合が含まれます。重要なのは、MFAチャレンジを開始する前に、ユーザーが承認された移動中ではないかどうかを（エンリッチメントされたデータに基づいて）評価するための「条件分岐」または「条件付きタスク」が必要であることです。これにより、疑わしいアクティビティが検出された場合にのみMFAチャレンジが送信され、不要な中断を防止できます。オプションAは、MFAの条件付きという側面を見落としています。オプションCは、ユーザーの移動ではなく、エンドポイントの詳細に焦点を当てています。オプション

Dは完全に手動で、自動化を妨げています。オプションEは、ユーザーの移動状況ではなく、IP脅威インテリジェンスに焦点を当てています。

最新問題: **80**
SIEMによって重大な脆弱性の悪用試行が検出され、XSOARインシデントがトリガーされました。インシデントには、攻撃者のIPアドレス、脆弱なサービス、および影響を受けるホストが含まれています。プレイブックでは、以下の処理を実行する必要があります。
1. サードパーティの脅威インテリジェンス プラットフォーム (TIP) を使用して、攻撃者の IP の評判を検証します。
2. IP が悪意のあるものである場合は、境界ファイアウォールでブロックします。
3. 影響を受けるホストでエンドポイントフォレンジック収集を開始します。
4. IT サービス管理 (ITSM) システムで優先度の高いチケットを開きます。
5. XSOAR インシデント ウォー ルームへの直接リンクを含め、PagerDuty 経由でインシデント対応チームに通知します。
これらの要件を考慮すると、PagerDuty 通知にライブ XSOAR インシデント War Room リンクが含まれるようにするために最も重要な XSOAR プレイブック設計要素はどれですか。また、プレイブック タスク内でプログラム的にそれを実現するにはどうすればよいでしょうか。
**A.** 「インシデントフィールド」機能は非常に重要です。War Roomリンクはインシデントフィールド（例${incident.warRoomURL}）として自動的に利用可能になり、PagerDuty統合タスクで直接使用できます。
**B.**
The 'Context Data' feature is crucial. A custom script task would be needed to construct the War Room URL using the incident ID and store it in context, e.g., `demisto.setContext('warRoomLink', f'https://your_xsoar_instance/incidents/{incident.id}/warroom')`, then referenced in PagerDuty.
**C.** 「統合」自体が重要です。PagerDutyとの統合により、明示的なプレイブック設定なしで、XSOARからWar Roomリンクが自動的に直接取得されます。
**D.** 「プレイブック入力」機能は非常に重要です。プレイブックをトリガーする際に、War Room リンクを手動で入力するか、カスタム統合コマンドで取得する必要があります。
**E.** 「レイアウト」機能は非常に重要です。War Roomリンクを表示するためのカスタムレイアウトを設計する必要があります。これにより、通知で使用できるようになります。
**Answer: A (**メーッセージを残す**)**
「インシデントフィールド」は非常に重要です。XSOARは、War Room URLを含む、システムレベルのインシデントフィールドをいくつか自動的に入力します。インシデントのWar Room URLは、インシデントオブジェクトの固有のプロパティであり、インシデントコンテキストから直接アクセスできます。そのため、PagerDuty統合タスクなど、通知を送信するタスク内で、JINJA2テンプレートまたはDemisto共通言語（DCL）を使用して直接参照できます。オプションBは不正解です。URLはすぐに利用可能であり、通常はカスタムスクリプトを作成する必要はありません。オプションCは不正解です。統合では、送信するデータを明示的に設定する必要があります。オプションDは自動化には実用的ではなく、オプションEはUIの表示に関するものであり、自動化のためのデータアクセスに関するものではありません。

最新問題: **81**
サイバーセキュリティインシデント対応チームは、ポリモーフィック型RAT（リモートアクセス型トロイの木馬）を用いた高度に洗練された攻撃を調査しています。この攻撃は、セキュリティ製品のメモリ内のサービスやプロセスを直接操作することで、それらの機能を無効化しようとします。このRATは高度な難読化技術を用いており、従来のシグネチャベースの手法では検出が困難です。Cortex XDRセンサーのどのような機能がこのような攻撃に対抗するために設計されており、なぜ効果的なのでしょうか？
**A.** ポリモーフィック RAT をサンドボックス内で実行し、その悪意のある動作を特定できるため、WildFire クラウド分析のみが有効です。
**B.** ローカル分析エンジンは、ファイル属性と PE ヘッダーの特性に基づいて RAT を識別します。
**C.** Behavioral Threat Protection (BTP) エンジンは、メモリ操作やコード インジェクションの試みを特に防止する Exploit Protection と、センサー自体が無効化されるのを防ぐ Anti-Tampering と組み合わせて、RAT の異常なプロセス動作 (予期しないネットワーク接続、プロセス インジェクションの試み、異常なファイルの変更など) を検出します。
**D.** ネットワーク保護モジュールは、事前に定義されたブラックリストに基づいて、RAT から C2 サーバーへのすべての通信をブロックします。
**E.** Cortex XDR のセンサーは、外部の脅威インテリジェンス フィードを利用して、RAT の C2 インフラストラクチャを識別します。
**Answer: C (**メーッセージを残す**)**
この質問は、多層的なセンサー保護を必要とする高度な攻撃について説明しています。WildFire（A）は効果的ですが、実際の攻撃に対しては事後対応的です。ローカル分析（B）は、ポリモーフィック型やファイルレス型の亜種を見逃す可能性があります。ネットワーク保護（D）は事後対応型で、既知のC2を想定しています。外部脅威インテリジェンス（E）も事後対応型で、事前の知識に依存します。このシナリオで最も効果的なセンサー機能の組み合わせは次のとおりです。1. 行動ベースの脅威保護（BTP）：RATの実行とそれに続く異常なアクティビティ（プロセスインジェクション、ネットワーク通信、システム変更など）を検出します。2. エクスプロイト保護：RATが使用するメモリ操作やコードインジェクション手法をプロアクティブに防止します。3. 改ざん防止：Cortex XDRセンサー自体が動作を継続し、マルウェアによって無効化されないようにします。エンドポイントセンサーからのこの包括的なアプローチは、検出を回避し、セキュリティ制御を無効化しようとする高度なポリモーフィック型攻撃を検出し、防止するために不可欠です。

最新問題: **82**
ある組織はセキュリティ運用をCortex XSOARに移行しており、インシデント対応中に実施されたすべてのアクション（実行者、実行日時、正確な結果を含む）を文書化するという厳格なコンプライアンス要件を負っています。これは、自動化されたプレイブックのアクションとアナリストによる手動のインタラクションの両方に適用されます。特に複雑なプレイブックが複数のサブプレイブックや統合を含む場合、インシデント調査においてこのレベルの詳細な監査可能性とレポートを総合的に保証できるXSOARの機能はどれでしょうか？

**A.** リアルタイムのコラボレーション ログ用の「War Room」と、高レベルのインシデント ステータス更新用の「Incident Summary」。
**B.** すべてのユーザー アクションとシステムの変更を記録する「監査証跡」機能と、ステップごとの実行の可視性を実現する「プレイブック デバッガー」、および各インシデント レコード内の「インシデント ログ」を組み合わせて、サブプレイブックの実行を含むすべてのコマンド出力とプレイブックのアクティビティをキャプチャします。
**C.** インシデント メトリックを視覚化するための「ダッシュボードとレポート」と、IOC を追跡するための「インジケーター」モジュール。
**D.** インシデントの進行状況を追跡するための「ケース管理」ビューと、標準操作手順 (SOP) を保存するための「ナレッジ ベース」。
**E.** 外部監査の目的で、調査の終了時にインシデント データを手動で CSV ファイルにエクスポートします。
**Answer: B (メーッセージを残す)**
オプションBは、詳細な監査とレポート作成のための最も包括的なソリューションを提供します。「監査証跡」は、XSOAR内のすべてのユーザーアクション（誰が何をいつ行ったか）とシステム変更を追跡するための基盤となります。「プレイブックデバッガー」は、開発段階や、ネストされたサブプレイブックを含む複雑なプレイブック実行パスを理解する上で不可欠であり、各ステップを可視化します。最も重要なのは、各インシデントレコード内の「インシデントログ」が、アナリストまたはプレイブックによって実行されたすべてのコマンド、その入力、および出力（統合およびサブプレイブックからの出力を含む）の詳細な時系列ログを取得することです。この組み合わせにより、自動または手動のすべてのアクションがプラットフォーム内で綿密に記録され、厳格なコンプライアンスおよび監査要件を満たすことができます。オプションA、C、D、およびEは、XSOARの貴重な機能をカバーしていますが、オプションBほど詳細で監査可能なすべてのアクションのログ記録は提供していません。

最新問題: **83**
高度なランサムウェア攻撃の事後フォレンジック分析において、チームは高度にカスタマイズされたパッカーと、C2通信に使用される通常とは異なるDGA（ドメイン生成アルゴリズム）を特定しました。Palo Alto Networks WildFireとThreat Preventionは、当初これらの新奇性のために検出できませんでしたが、後に詳細な脅威インテリジェンスレポートによって、パッカーとDGAのシード値の具体的なバイトパターンが提示されました。この後期段階の詳細な脅威インテリジェンスをPalo Alto Networksエコシステム内で最も効果的に活用し、将来の類似攻撃の検知と防御、特に初期の侵害の防止を向上させるにはどうすればよいでしょうか？
**A.** ファイアウォール上でカスタム アプリケーション オーバーライドを開発し、DGA によって生成されたトラフィックを識別し、カスタム判定のためにパッカーを WildFire に送信します。
**B.** パッカーのバイト パターン用のカスタム脅威防止 (IPS) シグネチャを作成し、DGA によって生成されたドメインを URL フィルタリング用の外部動的リスト (EDL) に統合します。

**C.** Cortex XDR の Behavioral Threat Protection を構成して、DGA のようなネットワーク アクティビティを監視し、パッカーの WildFire にカスタム YARA ルールを展開します。
**D.** ファイアウォールのスパイウェア対策プロファイルを DGA ドメインで更新し、パッカーのファイル タイプ用のカスタム ファイル ブロッキング プロファイルを作成します。
**E.** DGA シード値をネットワーク トラフィック アナライザーに入力してパッシブ検出を行い、ファイアウォールの脅威防止プロファイルでパッカーのカスタム脆弱性シグネチャを作成します。
**Answer: (**解答を表示する**)**
この設問は、インシデント発生後の詳細な脅威インテリジェンスを将来の予防に活用するための最も効果的な方法を特定し、パロアルトネットワークスのエコシステムにおける複数の効果的な戦略に焦点を当てています。BとCはどちらも強力で補完的なソリューションを提供しています。
オプション B (カスタム IPS + EDL): これは、初期の侵害防止に最適なネットワーク中心のアプローチです。
カスタム脅威防止 (IPS) シグネチャ: ネットワーク トラフィック内で直接パッカーの新しいバイト パターンを検出するのに最適で (たとえば、悪意のあるダウンロードやエクスプロイトのペイロードの一部として)、「仮想パッチ」または早期検出を提供します。
DGA ドメインの外部動的リスト (EDL): DGA によって生成された C2 ドメインを動的かつ継続的にブロックし、送信通信を防止します。
オプション C (Cortex XDR Behavioral + WildFire YARA): 強力なエンドポイントおよびファイルベースの検出を提供し、ネットワーク レベルの制御を補完します。
Cortex XDRのBehavioral Threat Protection（BTP）DGAに特徴的な異常なネットワークアクティビティ（ランダムドメインへのDNSルックアップの頻繁な失敗、通常とは異なるポートへの接続、特定のトラフィックパターンなど）やエクスプロイト後の挙動の検出に優れています。DGAシードを直接使用することはありませんが、DGAシードが引き起こす挙動を検出できます。
WildFireへのカスタムYARAルール :YARAはファイル内のパターンマッチングに特化して設計されています。パッカーのバイトパターンから構築されたカスタムYARAルールをWildFireにアップロードすることで、送信されたすべてのファイルに対してこのカスタマイズされたパッカーを検出・ブロックし、実行を阻止できます。
他のオプションが最適ではない理由:
A: アプリケーションオーバーライドは未知のアプリケーションを分類するためのものであり、悪意のあるパターンを検出するものではありません。WildFireにカスタム判定を送信することは良いステップですが、カスタムYARAルールやIPSほど直接的にプロアクティブな防御を行うことはできません。
D: アンチスパイウェアプロファイルは、既知のスパイウェアに対するシグネチャを主に使用します。DGAドメインを追加することも可能ですが、EDLはより動的です。ファイルブロッキングはファイルタイプ全般に適用され、カスタムパッカー固有の特性に限定されるものではありません。

E: DGAシードをネットワークアナライザーに送るのは手動または外部的な手順であり、Palo Altoの防御メカニズムに直接統合されていません。パッカーの「カスタム脆弱性シグネチャ」は一般的に誤った用語です。エクスプロイト/マルウェアパターンにはIPS（脅威防御が使用されます。

最新問題: **84**
公開されている脅威インテリジェンス ソースと非公開の独自のソースを照会するように設計された次の Python スクリプトを検討してください。

```
import requests

VIRUSTOTAL_API_KEY = "YOUR_VIRUSTOTAL_API_KEY"
WILDFIRE_API_KEY = "YOUR_WILDFIRE_API_KEY"

def query_virustotal(hash_value):
    url = f"https://www.virustotal.com/api/v3/files/{hash_value}"
    headers = {"x-apikey": VIRUSTOTAL_API_KEY}
    response = requests.get(url, headers=headers)
    return response.json()

def query_wildfire(hash_value):
    # Simplified for conceptual understanding, actual WildFire API may differ
    url = "https://wildfire.paloaltonetworks.com/publicapi/v2/get/report"
    params = {"apikey": WILDFIRE_API_KEY, "hash": hash_value}
    response = requests.get(url, params=params)
    return response.json()

file_hash = "d41d8cd98f00b204e9800998ecf8427e" # Example MD5 hash
vt_report = query_virustotal(file_hash)
print("\nVirusTotal Report:")
print(vt_report.get('data', {}).get('attributes', {}).get('last_analysis_stats', {}))

# Assuming 'd41d8cd98f00b204e9800998ecf8427e' is also available in WildFire
# For a real scenario, you'd submit unknown files to WildFire first
wildfire_report = query_wildfire(file_hash)
print("\nWildFire Report (Simplified):")
print(wildfire_report.get('malware', 'Not found in WildFire'))
```

提供されたスクリプトと、WildFire、Unit 42、VirusTotal に関する理解に基づいて、ファイル ハッシュが未知の、潜在的にゼロデイのマルウェア サンプルであると仮定した場合、特に独自のインテリジェンスと動作分析に関して、高度な脅威分析に query_virustotal ではなく query_wildfire の結果を使用することの比較上の利点を正確に説明している次の記述のうちどれですか。

**A.** query_virustotal は、より広範なコミュニティの貢献により、より詳細な動作分析と独自の脅威インテリジェンスを常に提供します。

**B.** query_wildfire は、ファイルが分析用に送信されると (ハッシュによるクエリだけでなく)、詳細なプロセス ツリー、ネットワーク接続、システムの変更など、独自のサンドボックス結果を提供します。これらの結果は、一般に、パブリック VirusTotal スキャン エンジンでは包括的に利用できず、詳細に分析されることもありません。

**C.** query_wildfire は主に静的分析とシグネチャ検索を目的としていますが、query_virustotal はゼロデイ脅威の動的分析に優れています。

**D.** どちらの機能も、未知のマルウェア サンプルに対して、同一レベルの独自の脅威インテリジェンスと動作分析を提供します。
**E.** query_wildfire の主な利点は、query_virustotal ではできない、Palo Alto Networks 以外のセキュリティ デバイスに新しいシグネチャを直接プッシュできることです。
**Answer: B (**メーッセージを残す**)**
WildFireの強みは、高度な独自開発の動的分析サンドボックスにあります。未知のファイルがWildFireに送信されると、制御された環境でマルウェアが起動し、プロセスの作成、ファイルシステムの変更、レジストリの変更、ネットワーク通信など、その動作が綿密に記録されます。この詳細な動作分析と、Palo Alto Networks独自の脅威インテリジェンスの生成は、VirusTotalに公開されている様々なウイルス対策エンジンから集約される情報よりもはるかに包括的で、独自のものです。VirusTotalはサンドボックスの結果（多くの場合、公開サンドボックスの結果）を表示する場合もありますが、WildFireの奥深さとPalo Alto Networksエコシステムとの統合（NGFWへの自動シグネチャ配信）は、特にゼロデイ脅威や回避型脅威に対して、WildFireの重要な差別化要因となります。

最新問題**: 85**
ある大規模企業が、NGFW、セキュリティオーケストレーション、自動化、レスポンスのためのCortex XSOAR、そして一元化されたSIEMを含む、Palo Alto Networksのセキュリティインフラストラクチャを活用しています。アナリストは、広く使用されている社内アプリケーションに影響を与える重大な脆弱性（CVE-2023-XXXX）を発見しました。脅威インテリジェンスによると、この脆弱性は既知のAPTグループによって積極的に悪用されています。SOCの現在の検出ルールとXSOARのプレイブックでは、この特定のCVEが明確にカバーされていません。検出分類の観点から、このギャップに関連する最も重大なリスクは何でしょうか？また、これをプロアクティブに軽減するために、Cortex XSOARをどのように活用すべきでしょうか？
**A.** 脆弱性のすべてのスキャンでアラートが生成されるため、リスクは真陽性のオーバーロードです。これらのアラートを自動的に抑制するには、XSOARを使用する必要があります。
**B.** リスクの主な原因は、ルールの設定ミスによる誤検知です。この設定ミスを監視するためのカスタムレポートを作成するには、XSOARを使用する必要があります。
**C.** 主なリスクは偽陰性です。XSOARを活用して、新たな脅威インテリジェンスを取り込み、SIEMとNGFW内に新たな侵害指標（IOC）と検出ルールを自動的に作成し、確認されたエクスプロイトへの自動対応のためのプレイブックを更新する必要があります。
**D.** リスクは真陰性です。XSOARを使用して、脆弱性がどのシステムにも存在しないことを確認し、脅威がないことを確認する必要があります。
**E.** リスクは「不明」状態です。XSOAR は、インシデントが発生した後にのみ、事後対応的に使用できます。
**Answer: C (**メーッセージを残す**)**
ここで最も重大なリスクは、False Negative（偽陰性）です。脆弱性が積極的に悪用され、現在のセキュリティ対策（検出ルール）がそれをカバーしていない場合、成功したエクスプロイトは検出されません。Cortex XSOARは、このシナリオ（オプションC）におけるプロアクティブな緩和策とし

て不可欠です。Cortex XSOARは、新しい脅威インテリジェンス（CVE-2023-XXXXに関連するIOC、TTPなど）を取り込み、それらを新しい検出ルールとしてSIEMおよびNGFWに自動的にプッシュします。また、検出時にインシデント対応プレイブックを更新し、この脆弱性に対する具体的な手順（ホスト分離、パッチ管理、フォレンジック収集、通信プロトコルなど）を追加します。このプロアクティブなアプローチは、実際の攻撃が発生した際に、潜在的なFalse NegativeをTrue Positiveに変換することを目的としています。

最新問題: **86**
高度なフィッシング攻撃は、最初のメールゲートウェイを回避します。XSOARプレイブックは、インシデントデータで見つかった不審なURLを分析するように設計されています。プレイブックには以下の機能が必要です。
1. インシデントの詳細からすべての URL を抽出します。
2. 一意の URL ごとに、複数の脅威インテリジェンス フィード (VirusTotal、URLscan.io など) に対してレピュテーション チェックを実行します。
3. URL が悪意のあるものと判断された場合、Web アプリケーション ファイアウォール (WAF) にブロック ルールが自動的に作成され、関連するプロキシ サーバーが更新されます。
4. URL が疑わしいものの、確実に悪意のあるものではない場合は、隔離された分析環境 (サンドボックス) に送信し、結果を待ちます。
5. すべての調査結果を構造化されたインシデント ノートに統合します。
抽出された各 URL を反復的に処理するのに最適な XSOAR プレイブック コンポーネントはどれですか。また、XSOAR 内でこれを実現するための一般的なプログラムによるアプローチは何ですか。
**A.** 「条件付きタスク」は反復処理に最適です。プログラム的には、条件付きタスク内のPython自動化スクリプトのforループでURLを反復処理し、サブタスクを実行できます。
**B.** 「Whileループ」タスクは反復処理用に特別に設計されています。一般的なプログラム的アプローチとしては、コンテキストからURLのリストを取得し、すべてのURLが処理されるまでカウンターをデクリメントし、各URLの分析ごとにサブプレイブックを用意するというものがあります。
**C.** 「データ収集タスク」は反復処理に最適です。プログラム的に設定することで、アナリストが各URLを1つずつ手動で処理するように促すことができます。
**D.** 「Playbook Inputs」メカニズムが理想的です。各URLを個別の入力として渡し、URLごとに新しいPlaybookインスタンスをトリガーする必要があります。
**E.** 「リンクタスク」が最適です。各URLには、事前に設定された分析タスクへの専用リンクが設定されます。
**Answer: B (**メーッセージを残す**)**
「Whileループ」タスク（新しいXSOARバージョンでは「ループ」）は、プレイブック内の反復処理のために明示的に設計されています。一般的なプログラム的アプローチでは、インシデントコンテキストに保存されたアイテムのリスト（この場合はURL）を使用します。ループ条件は、リストが空かどうか、またはカウンターが制限に達したかどうかを確認します。ループ内では、サブプレイ

ブックまたは一連のタスクがリストから1つのURLを処理し、それを削除してから、ループ条件を再評価します。オプションAは不正解です。条件付きタスクは分岐用であり、直接的な反復処理ではありません。オプションCは手動で実行され、自動化されていません。オプションDはインシデントが急増し、非効率的です。オプションEは関連タスクをリンクするためのものであり、反復処理用ではありません。

最新問題: **87**
Okta 統合からの不審なログインアラートに対応するように XSOAR プレイブックが設計されているシナリオを考えてみましょう。プレイブックのロジックでは、ログインが外部 GeoIP サービスによって 「高リスク」と識別された国から行われた場合、Okta を介してユーザーのパスワードを即座にリセットし、発信元 IP のブロックルールを Palo Alto Networks NGFW に作成するように指示されています。さらに、Jira チケットがレビュー用にオープンされます。特定のインシデントに対するプレイブック実行中に GeoIP サービス統合が失敗したりエラーを返したりした場合、次のどの XSOAR メカニズムによって、プレイブックがこの失敗を適切に処理し、エラーをログに記録し、プロセス全体を停止したりインシデントを未解決のままにしたりすることなく、インシデントをエスカレーションできるでしょうか。
**A.** GeoIP 統合の成功をチェックし、失敗した場合は人間のアナリストが介入するための 「手動」タスクに移行する 「条件付き」タスクを実装します。
**B.** プレイブック内の 「エラー処理」ブロックを活用し、GeoIPサービス統合呼び出しからの例外を具体的に捕捉します。このブロックは、SOCマネージャーへの 「メール送信」コマンドを実行し、「demisto.logError(Y」を使用して詳細なエラーメッセージをログに記録した後、Oktaパスワードリセットやngfwブロックを実行せずに、「インシデントステータスを 「レビュー保留中」に設定」タスクに進みます。
**C.** プレイブックの実行が完了した後、XSOAR システム ログのみを使用して統合の失敗を特定し、プレイブックを手動で再起動します。
**D.** GeoIP 統合のタイムアウト設定を非常に高い値に構成します。これは、最終的に成功することを想定しており、成功しない場合は、プレイブックはそのステップで停止します。
**E.** プレイブックの入力にデフォルトの 「低リスク」国を事前定義します。これにより、GeoIP サービスが失敗した場合、デフォルトでそれほど積極的ではない応答パスが設定されます (例: Jira チケットのみを開く)。
**Answer: B (**メーッセージを残す**)**
オプション B は、最も堅牢で XSOAR ネイティブのエラー処理メカニズムについて説明しています。XSOAR プレイブックは、明示的なエラー処理ブロックをサポートしています。GeoIP 統合からの例外を具体的にキャッチすることで、プレイブックは次のことを行うことができます。1. プレイブック全体のクラッシュを防止します。2. デバッグとインシデント後の分析に不可欠な 'demisto.logError()' を使用して詳細なエラー情報をログに記録します。3. SOC マネージャーに認識を促すため、即時通知 (メール) を送信します。4. 不完全な情報に基づいて潜在的にリスクのあるアクション (パスワードのリセット、ブロック) を実行せずに、自動化されたステップが不完全で人間の介入が必要であることを示す、インシデントを 'レビュー保留中' ステータスに正常に移

行します。これにより、外部統合の障害が発生した場合でも、継続性と適切なインシデント管理が保証されます。オプションAとEは部分的な解決策を提供しますが、オプションBのような包括的なエラーのキャプチャとレポートがありません。オプションCとDは事後対応的であるか、非現実的です。

最新問題: **88**
高度なゼロデイ攻撃により、複数の重要なサーバーが侵害を受けました。インシデント対応チームはCortex XSOARのWar Roomを使用しています。攻撃の斬新さゆえに、既存の自動化されたプレイブックでは完全な修復を行うには不十分です。チームは、War Room内の制御された環境で、新しい検知・対応ロジックを共同で開発・テストし、カスタムスクリプトを共有し、その有効性を検証する必要があります。War Roomは、インシデント発生中に、このアジャイルで反復的な開発・テストプロセスをどのように促進するのでしょうか？
**A.** War Room は主にコミュニケーションログとして機能します。すべての開発は外部の別の IDE で実行する必要があります。カスタムスクリプトはコンテンツパックとして XSOAR に手動でインポートされるため、イテレーションごとにプラットフォーム全体を再起動する必要があります。
**B.** アナリストは、「run_script」コマンドを使用して、PythonスクリプトをWar Roomのエントリとして直接共有できます。War Roomの「Automation」タブでは、これらのスクリプトを実際のインシデント状況に対して即座にテストできます。新しい検出ルールをメモとして作成し、外部セキュリティツールで手動で設定することも可能です。
**C.** War Room は、コマンドライン経由でアドホック Python スクリプトまたはコマンドの実行をサポートしており、インシデントデータに対する即時のテストを可能にします。Venrich_ioc' などのコマンドを使用することで、新たな侵害指標 (IOC) を共有し、自動的に強化することができます。共有メモを通じて新しいプレイブックロジックを共同で作成し、それをプレイブックの一部のスニペットとしてエクスポートすることも可能です。
**D.** War Room は「サンドボックス環境」と統合されており、新しいロジックやスクリプトを個別に開発・テストできます。検証が完了すると、本番環境の XSOAR インスタンスに自動的にデプロイされ、War Room に反映されます。
**E.** ウォールームの主な機能はデータの可視化です。新しいロジックを開発・テストするには、チームはすべてのインシデントデータをエクスポートし、オフラインで分析を実行し、新たな発見事項やスクリプトを「エビデンス」エントリとして手動で再インポートする必要があります。
**Answer: (**解答を表示する**)**
オプションCは、ウォールームがライブインシデント発生中のアジャイル開発とテストをいかにサポートするかを的確に示しています。ウォールームのコマンドラインからアドホックPythonスクリプトやコマンドを直接実行できる機能は、プレイブック全体を作成・修正することなく、ライブインシデントデータに対して新しいロジックを即座にテストできる非常に強力な機能です。ウォールームは、コマンドを使用して新しいIOCを即座に共有・強化することを容易にします。完全なIDEではありませんが、ウォールームのコラボレーション機能（メモや共有エントリを通じ

た)により、チームは新しい検知・対応ロジックのコンセプトを共同で草案・改良することができ、その後、プレイブックに正式に統合することができます。この反復的な「オンザフライ」機能は、複雑で斬新なインシデントシナリオにおけるXSOARのウォールームの真髄です。

最新問題: **89**
SOCチームは、「真のAIを活用した脅威インテリジェンス統合」を謳う新しいベンダーを評価しています。現在のプロセスでは、脅威インテリジェンスフィードを手動で確認し、ファイアウォールルールまたはSIEM相関ルールを手動で更新しています。CISOは、「真のAI」が、単純なスクリプトや基本的な機械学習ベースのキーワード抽出では実現できない、このプロセスを根本的に変革する仕組みを理解したいと考えています。この文脈において、「機械学習」を超えた最も先進的で独特な「AI」機能は、次のうちどれでしょうか？
**A.** AI システムは、教師あり ML を使用して脅威インテリジェンス記事をカテゴリ (マルウェア、APT、脆弱性など) に分類し、アナリストによる分類を容易にします。
**B.** AI システムは自然言語生成 (NLG) を採用し、脅威インテリジェンス レポートをアナリスト向けに簡潔で実用的な箇条書きに要約します。
**C.** AIシステムは、自然言語理解（NLU）とナレッジグラフを活用して、非構造化脅威インテリジェンスを読み取り、理解します。TTP、IOC、アクタープロファイルを自動的に抽出し、それらが組織の特定の資産や脅威態勢とどのように関連しているかを推論し、人間の介入を最小限に抑えながら、適応型防御メカニズム（新しいファイアウォールポリシー、エンドポイント強化ルールなど）を動的に生成・展開します。これは、シンボリックAIと自律推論の実証です。
**D.** AI システムは強化学習を使用して、新しいインテリジェンスがインシデント削減に及ぼした影響の履歴に基づいて、脅威インテリジェンス フィードの更新頻度を最適化します。
**E.** AI システムは、教師なし ML を適用して、さまざまな脅威インテリジェンス ソースからの一見異なる IOC 間の新たな相関関係を発見します。
**Answer: (**解答を表示する**)**
課題は、「単純なスクリプトや基本的な ML ベースのキーワード抽出で達成できることを超えて」、「真の AI」を実証することです。オプション A、B、E は ML の高度なアプリケーション (分類、要約、相関) について説明していますが、主に情報の処理と提示に焦点を当てています。価値あるものではありますが、複雑で進化するインテリジェンスに基づく「理解」と「行動」のパラダイムを根本的に変えるものではありません。オプション D は AI 最適化機能について説明していますが、インテリジェンス統合の核心的な変革ではありません。オプション C は、このコンテキストにおける AI の頂点を表しています。これは、複雑で非構造化の脅威インテリジェンスに基づいて、システムが理解 (NLLJ)、推論 (シンボリック AI、ナレッジ グラフ)、自律的に行動 (動的ポリシーの生成と展開) する能力について説明しています。これは、単なるデータ処理を超えて、コンテキストと関連性を真に理解し、防御を自律的に適応させることにまで進み、これが高度な AI と I ML の主な差別化要因です。意味的な理解を構築し、その理解を特定の環境にどのように適用するかについて推論します。

最新問題: **90**
Cortex XDRが 「悪意あり」と判定されたXDRストーリーを検出し、一連のイベントが発生したシナリオを考えてみましょう。「Outlook.exe」が 「cmd.exe」を起動し、次に 「mshta.exe」を実行してリモートHTAファイルを実行し、その後 「evil.exe」をドロップして実行しました。その後、「evil.exe」は外部IPへのC2接続を確立しようとしました。Causality ViewがこのXDRストーリーの調査をどのように強化し、セキュリティ運用担当者にとってなぜ重要であるかを正確に説明している記述は、次のうちどれですか？
**A.** Causality View は、各イベントのすべての生のログを単一の検索可能なテキスト ファイルに集約し、視覚的な表現なしでログ分析を簡素化します。
**B.** プロセス ツリーの時系列のインタラクティブ グラフを表示します。ルートとして 'Outlook.exe' が表示され、'cmd.exes'、次に 'mshta.exe'、最後に 'evil.exe' と分岐します。これにより、アナリストは攻撃フローの全体を追跡し、最初の侵害ベクトルを特定できます。
**C.** Causality View は、関連するすべてのファイルを自動的に隔離し、XDR ストーリー内のすべてのプロセスを終了するため、アナリストによる手動介入は必要ありません。
**D.** 悪意のあるプロセスによって行われたすべてのシステム変更を感染前の状態にロールバックする直接的な 「ワンクリック」修復ボタンを提供し、詳細な調査の必要性を排除します。
**E.** 因果関係ビューはネットワーク接続のみに焦点を当て、親プロセスに関係なく、「evil.exe」によって確立されたすべてのアクティブな接続のリアルタイム マップを提供します。
**Answer: B (**メーッセージを残す**)**
複雑なXDRストーリーを理解するには、Causality Viewが不可欠です。オプションBは、関連するプロセスとイベントをインタラクティブな時系列グラフで表示するという、Causality Viewの中核機能を正確に説明しています。これにより、セキュリティ運用担当者は、最初のトリガー（悪意のある添付ファイルまたはリンクによって 「Outlook.exe」が 「cmd.exe」を起動する）から最終的な悪意のあるアクティビティ（「evil.exe」がC2を確立する）まで、攻撃チェーン全体を視覚化できます。親子関係や関連するネットワーク/ファイル/レジストリアクティビティを含む一連のイベントを視覚的に理解することは、攻撃の範囲を特定し、持続メカニズムを特定し、効果的な封じ込めおよび根絶戦略を策定する上で不可欠です。オプションA、C、D、およびEは、Causality Viewの機能を誤って表現しているか、調査後に実行される可能性のある自動アクションを説明しているものの、ビュー自体の主目的ではありません。

最新問題: **91**
最近の監査で、一部のXSOARプレイブックが、レート制限の厳しい外部サービスへの冗長なAPI呼び出しを実行していることが明らかになりました。チームは、この特定のサービスのレスポンスに対してグローバルキャッシュメカニズムを実装したいと考えています。そこで、データを15分間保存するカスタムキャッシュを使用することにしました。このキャッシュは、複数のプレイブックとそこに埋め込まれたスクリプトからアクセスできる必要があります。スクリプトとジョブの違いを考慮すると、XSOARでこの共有型時間ベースキャッシュを実装する上で、最もスケーラブルで保守性の高いアプローチはどれでしょうか？

**A.** レート制限サービスを呼び出す各スクリプト内にキャッシュ ロジックを実装し、インシデント コンテキストにデータを保存し、スケジュールされたジョブでデータをクリアします。
**B.** レート制限サービスをラップし、XSOARの組み込みキーバリューストアを使用して内部的にキャッシュロジックを実装する新しいXSOAR統合を作成します（
`demisto.setContext()`

and
`demisto.getContext()`
**C.** レート制限されたサービスから定期的にデータを取得し、それをグローバル XSOAR リストに保存し、スクリプトがこのリストをクエリするジョブを利用します。
**D.** XSOAR サーバー構成を変更して、すべてのスクリプトと統合が直接アクセスできる外部 Redis キャッシュを有効にします。
**E.** コマンドとして公開される専用の Python スクリプトを開発します。このスクリプトは、レート制限されたサービスへのすべての呼び出しを処理し、メモリ内のキャッシュを実装し、他のスクリプトによって呼び出されます。
**Answer: B (**メーッセージを残す**)**
最もスケーラブルで保守性の高いアプローチは、レート制限サービスをラップし、キャッシュロジックを内部的に実装する新しいXSOAR統合を作成する または既存の統合を修正する）ことです。その理由は以下のとおりです。1. 統合は、外部APIとのやり取りを抽象化し、その状態/キャッシュを管理するのに適した場所です。2. XSOARのキーバリューストア（
`demisto.setContext()`

and
`demisto.getContext()`
3. このアプローチは、キャッシュ ロジックを一元管理するため、この統合を使用するすべてのプレイブックまたはスクリプトで再利用でき、適切な有効期限が確保されます。オプション A は、インシデント コンテキストがインシデントごとにあり、グローバルではないため、ジョブでクリアするのが非効率的であるために問題があります。オプション C は、キャッシュの効率的なキーと値の参照と有効期限のために設計されていないリストを使用します。オプション D は、内部キャッシュの標準的な XSOAR プラクティスではなく、外部依存関係を導入します。オプション E (スクリプト内のメモリ内キャッシュ) は、異なるスクリプトの実行間、さらには異なるプレイブックの実行間では保持されないため、グローバル キャッシュには効果がありません。

最新問題: **92**
高度なAPTグループが、カスタムのポリモーフィック型マルウェア亜種を使用していることが確認されました。初期の侵害において唯一共通する指標は、C2通信に新規登録された固有のドメイン（evil-command-control.xyz）を使用していることです。これは、公開されている脅威インテリジェンスフィードではまだ広く知られていません。セキュリティチームは、このドメイン指標をCortexエコシステム内で迅速に運用化し、防御と検知の両方を実現する必要があります。
**A.** ドメインをWildFireに送信して分析を行い、判定結果を待ちます。その後、NGFW上でドメイン用のカスタムURLフィルタリングプロファイルを手動で作成します。Cortex XDRの「Search」機能を使用して、ドメインへのDNSクエリを検索します。
**B.** Cortex XSOAR内のカスタム「脅威インテリジェンスフィード」にドメインを取り込み、すべての次世代ファイアウォールの外部動的リスト（EDL）に自動的にプッシュします。同時に、Cortex XDRに新しい「分析ルール」を設定し、evil-command-control.xyzへのネットワーク接続またはDNS解決をアラートで通知します。
**C.** Cortex XDRの「インジケーター管理」機能を利用してドメインを直接インポートします。これにより、ドメインへのトラフィックが自動的にブロックされ、既存の接続に対してアラートがトリガーされます。
**D.** NGFW 上の既存の「DNS セキュリティ ポリシー」を変更して、.xyz トップレベル ドメインへのすべてのクエリをブロックし、影響を受けるエンドポイントで「ライブ ターミナル」セッションを開始して、ブラウザー履歴でドメインを検索します。
**E.** ドメイン evil-command-control.xyz のカスタム「AutoFocus プロファイル」を作成し、Cortex XSOAR を使用して手動調査用の「War Room」を作成します。
**Answer: B (**メーッセージを残す**)**
オプションBは、最も堅牢で自動化されたソリューションです。ドメインをカスタムXSOAR脅威インテリジェンスフィードに取り込むことで、一元管理とNGFW EDLへの自動配信が可能になり、ネットワーク全体を即座にブロックできます。同時に、XDRで分析ルールを作成することで、エンドポイントでドメインへの接続や解決を試みた場合、継続的な検出とアラート通知が保証されます。これにより、プロアクティブな防御とリアクティブ検知の両方が実現します。オプションAは手動操作が多く、リアクティブです。オプションCは誤りです。XDRはインジケータを使用できますが、インジケータのインポートのみに基づいてネットワーク全体を直接自動ブロックすることは、NGFWとの統合や特定のポリシーがなければ、XDRの主要なメカニズムではありません。オプションDは範囲が広すぎて、正当なサービス中断を引き起こす可能性があります。オプションEは調査手順であり、自動防御や検知は提供しません。

最新問題: **93**
NISTインシデント対応計画の「事後活動」フェーズにおいて、ある組織は、高度な持続的脅威（APT）技術を用いた複雑な多段階攻撃によって機密性の高いデータが盗み出されたことを発見し

ました。事後分析の結果、脅威インテリジェンスの統合と自動対応機能にギャップがあることが明らかになりました。Palo Alto Networksのセキュリティプラクティスと整合した以下の改善策のうち、これらのギャップを最も効果的に解消し、同様の高度な脅威に対する将来の「準備」および「検知・分析」フェーズを強化するには、どの改善策が効果的でしょうか？

**A.** Cortex XSOAR プレイブックを実装して、AutoFocus および WildFire インテリジェンスを使用してアラートを自動的に強化し、信頼性の高い IOC 一致に基づいてターゲットを絞った応答 (エンドポイントの隔離、NGFW での C2 ドメインのブロックなど) を調整します。

**B.** 組織全体でパッチが適用されていないすべてのソフトウェアの監査を実施し、未適用のパッチをすべて直ちに適用して、攻撃対象領域を最小限に抑えます。

**C.** 外部境界防御に重点を置き、ネットワーク全体の脆弱性スキャンと侵入テストの頻度を増やします。

**D.** きめ細かなマイクロセグメンテーション ポリシーを適用するために、すべての内部ネットワーク セグメントに追加の次世代ファイアウォールを導入します。

**E.** より強力なスパムおよびフィッシング フィルターを使用して電子メール セキュリティ ゲートウェイを強化することにのみ重点を置きます。

**Answer: A (**メーッセージを残す**)**

「インシデント後の活動」フェーズには、得られた教訓と改善点が含まれます。このシナリオでは、複雑な多段階攻撃に対する「脅威インテリジェンス統合と自動対応機能のギャップ」を具体的に指摘しています。- A: Cortex XSOARプレイブックをAutoFocusおよびWildFireと統合することで、両方のギャップに直接対処できます。XSOARは、グローバル脅威インテリジェンス（AutoFocus、WildFire）からのコンテキスト情報を用いたアラートの拡充を自動化し、自動対応をオーケストレーションすることで、「検知と分析」の精度と、将来の類似インシデントに対する自動アクションを定義することで「準備」のスピードと効率を大幅に向上させます。これはまさに、インテリジェンスの統合と対応の自動化に関するものです。- B、C、D、Eはすべて有効なセキュリティ改善策ですが、特定された特定のギャップ（脅威インテリジェンス統合と自動対応）に、XSOARとその機能ほど効果的に直接対処するものではありません。パッチ適用（B）、スキャン（C）、マイクロセグメンテーション（D）は攻撃対象領域の縮小とネットワーク制御の改善を目的としていますが、メールセキュリティ（E）は単一の攻撃ベクトルに焦点を当てています。有益ではありますが、XSOAR のように、分析のための脅威インテリジェンスの統合を具体的に強化したり、APT に対する複雑なマルチツールの対応を自動化したりするものはありません。

最新問題**: 94**

セキュリティオペレーションセンター（SOC）は、ログ取り込み戦略をCortex XSIAMに移行しています。重要な業務アプリケーションが、ネストされたオブジェクトと配列を含むカスタムJSON形式でログを生成しています。既存のSIEMでは、このログを効率的に解析できず、セキュリティ分析が不完全でした。これらの複雑なJSONログを正確に解析・拡充するために、最も効果的なCortex XSIAMデータ取り込みプロセスとその理由は何でしょうか？

**A.** Cortex XSIAM のデフォルトの JSON パーサーのみに依存して、syslog 経由で直接取り込みます。

**B.** エンドポイント ログに Cortex XDR エージェントを使用し、ネットワーク デバイス ログをローカル コレクター経由で転送し、JSON 形式の XSIAM 内でカスタム解析ルールを構成します。
**C.** 専用のログ コレクターをオンプレミスで展開し、JSON 構造のカスタム XQL 解析ルールを使用してログ プロファイルを構成し、特定の属性に対してフィールド抽出ルールを活用します。
**D.** ログをクラウド ストレージ バケット (S3 など) にプッシュし、事前定義されたスキーマと JSON をフラット化する変換関数を使用してデータ取り込みルールを構成します。
**E.** サードパーティの ETL ツールを使用して、JSON ログを Cortex XSIAM に取り込む前に前処理し、フラットな CSV 形式に正規化します。
**Answer: C (メーッセージを残す)**
ネスト構造を持つ複雑なカスタムJSON形式の場合、デフォルトのパーサー（A）やシンプルなエージェント（B）に頼るだけでは不十分です。クラウドストレージ（D）も選択肢の一つですが、オンプレミスのカスタムログを扱うCortex XSIAMで最も堅牢かつ柔軟なアプローチは、専用のログコレクターを導入することです。これにより、カスタムXQL解析ルールを含むログプロファイルを作成でき、ネストされたJSONをナビゲートして特定のフィールドを抽出できる強力なツールです。フィールド抽出ルール（FER）はこのプロセスをさらに洗練させ、正確なデータエンリッチメントを実現します。Cortex XSIAMがネイティブ機能を備えている場合、サードパーティ製のETLツール（E）は不要な複雑さとコストを増大させます。

最新問題: **95**
貴社では、脅威の検知と対応にCortex XDRを使用しています。最近の社内セキュリティ監査で、重大な脆弱性が明らかになりました。権限のないユーザー（user_developer）が本番サーバー上の機密設定ファイルにアクセスでき、最小権限の原則に違反していたのです。データの流出は発生していませんが、これはユーザーおよびロール管理における体系的な問題を示唆しています。監査では、ユーザー行動分析、ロール定義、データ保護に重点を置いた、同様のインシデントを防止するための堅牢なシステムの導入を推奨しています。導入されていれば、このアクセスを阻止し、即時の検知と実用的な洞察を提供できたであろうCortex XDRの機能とベストプラクティスをすべて選択してください。
**A.** 機密性の高い構成ファイル (例: /etc/apache2/sites-avai1ab1e/、/var/lib/mysql/) を含むパスへの user_developer のアクセスを明示的にブロックするデータ保護ポリシーを実装します。
**B.** Cortex XDRのユーザー行動分析（UBA）を活用して、user_deve10perの典型的なアクティビティをベースライン化します。本番環境の設定ファイルへのアクセスはすべて異常なアクティビティとしてフラグ付けされ、アラートがトリガーされます。
**C.** Cortex XDR で、user_developer に対して、機密性の高い本番サーバー構成の表示または変更権限を明示的に除外するカスタム ロールを定義し、対象プロファイルを通じてこのロールをエンドポイント エージェントに適用します。
**D.** 'file_access' イベントに基づいてカスタム XQL アラートを作成し、特に管理者以外のユーザーによる既知の機密構成ファイル パスへのアクセスを探します。

```
dataset = file_access | filter action_file_path in ('/etc/apache2/sites-available/', '/var/lib/mysql/') and not (actor_process_user_id in (1000, 0)
or actor_process_user_name in ('root', 'admin')) | groupby actor_process_image_name, actor_process_user_name, action_file_path limit 100
```

**E.** 本番サーバーでCortex XDRのフルディスク暗号化を有効にします。これにより、権限のないユーザーは、役割やファイルの権限に関係なく、ファイルを読み取ることができなくなります。
**Answer: A,B,D (**メーッセージを残す**)**
この質問では、プロアクティブな予防、行動検出、正確なルールベースの検出を特定する必要があります。A (データ保護ポリシー): これは直接的な予防策です。Cortex XDR のデータ保護モジュールは、ユーザーまたはユーザー グループに基づいて特定のファイル パスへのアクセスを明示的にブロックまたは制限し、機密性の高い構成ファイルへのアクセスを効果的に防止できます。B (ユーザー行動分析): UBA は、異常な行動を検出するために user_developer が不可欠です。の通常のアクティビティにこれらのパスへのアクセスが含まれていない場合、UBA はこの user_developer をベースラインとして設定し、逸脱があれば疑わしいとフラグを付けて、即時検出を提供します。C (カスタム ロール定義): このオプションには問題があります。Cortex XDR のロールでは、主に XDR コンソールとその機能内のアクセスが管理され、エンドポイント自体のファイル システム権限は直接管理されません。XDR ロールでは、そのユーザーに関してアナリストが XDR で表示または実行できることが制限される場合がありますが、OS の権限で許可されている場合、ユーザーが OS 上のファイルにアクセスすることを直接防止することはありません。脆弱性は OS レベルにあり、XDR コンソール レベルではありません。したがって、アクセス自体を防ぐことはできません。D (カスタム XQL アラート): これは、具体的で実用的な検出を提供します。細かく調整された XQL クエリは、アクセスすべきでないユーザーによるこれらの特定のパスへのアクセスを直接監視します。これは、SOC に即座に警告できる強力な検出メカニズムです。E (フルディスク暗号化): フルディスク暗号化は保存中のデータには重要ですが、主にディスクが物理的に取り外されている場合やシステムがオフラインの場合にのみデータを保護します。システムが稼働し、OS 操作のためにディスクが復号化されると、ファイル アクセスは暗号化自体ではなく、OS レベルの権限によって制御されます。OS にアクセスできる権限のないユーザーは、ディスクが暗号化されていても、OS の権限で許可されていればファイルを読み取ることができます。シナリオで強調表示されている特定のアクセスを防ぐことはできません。

最新問題**: 96**
広く利用されているウェブサーバーに、重大なゼロデイ脆弱性が公開されました。貴社のインシデント対応計画では、潜在的な悪用試行を特定するための迅速な対応が規定されています。Palo Alto NetworksのNGFW、WildFireへのアクセス、Unit 42の脅威インテリジェンスへの加入を保有しています。さらに、初期偵察にはVirusTotalを頻繁に使用しています。潜在的な悪用試行を迅速に特定し、封じ込めるために、以下の戦略の組み合わせのうち、最も迅速な対応能力と長期的なインテリジェンス収集能力を兼ね備えているのはどれでしょうか？
**A.** 影響を受ける Web サーバーへのすべてのトラフィックを事前にブロックし、そのログを VirusTotal に送信して事後分析を行います。
**B.** Unit 42 の迅速な脆弱性調査とエクスプロイト インテリジェンスを活用して特定のエクスプロイト パターンを識別し、NGFW でカスタム シグネチャまたは脅威防止プロファイルを構成し、観測された疑わしいペイロードに対して WildFire を使用します。

**C.** パッチがリリースされるまで脆弱な Web サーバーを完全に無効にし、関連するハッシュについて VirusTotal の過去の送信内容を確認します。
**D.** 脆弱性に関する言及について公開フォーラムとソーシャル メディアを監視し、一般的なネットワーク侵入検知システム (NIDS) ルールを適用します。
**E.** Web サーバーの悪用は主にエンドポイントの問題であるため、エンドポイント検出および応答 (EDR) アラートのみに焦点を当てます。
**Answer: (**解答を表示する**)**
ゼロデイ脆弱性には、迅速かつ的確な対応と、潜在的なエクスプロイトに対する深い理解が必要です。Unit 42は、迅速な脆弱性調査とエクスプロイトインテリジェンスに優れており、脆弱性が実際にどのように武器化されているかについての詳細な分析を頻繁に提供しています。このインテリジェンスは、NGFW上で具体的かつ効果的な脅威防止ルールを作成するために不可欠です。WildFireは、確認された新たなペイロードやエクスプロイト後のツールを分析し、リアルタイムのシグネチャを提供します。この組み合わせたアプローチにより、専門家によるインテリジェンスと新たな脅威の動的分析に基づいた、ネットワークレベルのプロアクティブな防御が可能になります。

最新問題**: 97**
SOC（セキュリティオペレーションセンター）は、脅威検知とインシデント対応ワークフローの強化を目的とした、新しいセキュリティ情報イベント管理（SIEM）ソリューション、Palo Alto Networks Cortex XSIAMの評価を行っています。重要な要件は、エンドポイントテレメトリ、ネットワークフローデータ、クラウドログなど、多様なセキュリティイベントを自動的に相関させ、高度な持続的脅威（APT）を特定することです。この要件を直接サポートするXSIAMの中核機能はどれですか？また、XSIAMの効果的な導入によってSOC内のどの役割が最も影響を受けるでしょうか？
**A.** 統合データレイク; セキュリティアナリスト Tier 1
**B.** 機械学習と行動分析; セキュリティアナリスト Tier 2/3
**C.** オーケストレーションと自動化（SOAR）; SOC マネージャー
**D.** 攻撃対象領域管理; 脆弱性管理スペシャリスト
**E.** 脅威インテリジェンス管理; 脅威ハンター
**Answer: B (**メーッセージを残す**)**
Palo Alto Networks Cortex XSIAMは、機械学習と行動分析を活用し、多様なデータソースを相関分析することで、APT攻撃に特徴的な巧妙で多段階的な攻撃を特定します。これは、単純なルールベースのアラート通知の域を超えています。この高度な相関分析機能は、複雑な攻撃チェーンの詳細な調査と理解を担うTier 2およびTier 3のセキュリティアナリストに直接的なメリットをもたらします。これにより、ノイズではなく真陽性と信頼性の高いアラートに集中できるようになります。XSIAMの機能やSOCロールといった選択肢もありますが、「機械学習と行動分析」は高度な相関分析に特化しており、「Tier 2/3セキュリティアナリスト」が複雑な脅威の特定におけるその有効性の恩恵を最も受けます。

最新問題: **98**
SOC（セキュリティオペレーションセンター）は、エンドポイントの検知と対応にPalo Alto Networks Cortex XDRを使用しています。疑わしいPowerShellアクティビティを特定するために、新しいカスタム行動脅威検知ルールを実装しました。特に、エンコードされたコマンドとセキュリティ機能を無効化しようとする試みに焦点を当てています。導入から数日後、SOCは大量のアラートに悩まされています。そのほとんどは、正規のIT管理スクリプトやソフトウェアインストーラーに起因するものでした。この大量のアラートは、実際の脅威への対応能力に深刻な影響を与えています。この状況と、最も効果的な戦略的対応策について、以下の記述のうちどれが的確に説明されていますか？
**A.** これは真陰性のシナリオです。ルールは意図したとおりに機能しています。SOCはアナリストをさらに雇用する必要があります。
**B.** これはFalse Negative（偽陰性）を表します。ルールは真の脅威を検出できていません。ルールをより積極的に適用する必要があります。
**C.** これは誤検知の蔓延です。戦略的な調整には、カスタムルールをより具体的な除外基準で改良し、コンテキスト情報（信頼できる発行元、特定のファイルパスなど）を活用し、場合によっては「通常の」アクティビティのベースラインを実装して逸脱を特定することが含まれます。
**D.** これは真陽性の過負荷です。真の脅威が検出されています。解決策としては、すべてのアラートへの対応を自動化することです。
**E.** これは「未検出」イベントの例です。再評価されるまで、ルールは直ちに無効化する必要があります。
**Answer: C (**メーッセージを残す**)**
このシナリオは、明らかに誤検知の蔓延を示しています。カスタム ルールが範囲が広すぎるため、無害なアクティビティに対して多くのアラートが生成されています。最も効果的な戦略的調整 (オプション C) は、ルールを改良することです。これには、より具体的な除外基準の追加 (例: 信頼できるベンダーによって署名された PowerShell スクリプトや特定の IT 自動化ディレクトリからの PowerShell スクリプトを許可する)、無害なものと悪意のあるものを区別するためのコンテキスト情報の組み込み (例: PowerShell が特権コンテキストで実行されているかユーザー コンテキストで実行されているか、既知の悪意のある兆候に関連する場合にのみセキュリティ機能を無効にしようとしているか)、そして真の異常を識別するために通常の PowerShell 動作のベースラインを構築することが含まれます。オプション A と B は状況を誤って分類しています。オプション D は対応の自動化を提案していますが、誤検知率が高い場合は危険です。オプション E は過剰反応です。ルールを完全に無効にすると、ルールを改良するのではなく、誤検知のリスクが生じます。

最新問題: **99**
Palo Alto Networks（PAN-OS）次世代ファイアウォールを採用したセキュリティオペレーションセンター（SOC）は、重要な社内サーバーから通常とは異なるトップレベルドメインへのアウトバウンドDNSリクエストが急増していることを確認しました。脅威インテリジェンスフィードは、DNS情報漏洩を悪用した最近のキャンペーンを示唆しています。NISTインシデント対応計画

の文脈において、このシナリオにおける「検知と分析」フェーズに最も適したアクションはどれですか？これは、更なる封じ込め活動に先立つものです。
**A.** PAN-OS セキュリティ ポリシー ルールを使用して、影響を受けるサーバーからのすべての送信 DNS トラフィックを直ちにブロックします。
**B.** 影響を受けるサーバーからのすべてのトラフィックに対してファイアウォール上で完全なパケット キャプチャを開始し、DNS クエリの内容を分析して疑わしいパターンを検出するとともに、DNS セキュリティ ログと相関関係を調べます。
**C.** 侵害が発生したと想定して、サーバーをネットワークから分離し、フォレンジック イメージングを開始します。
**D.** 潜在的な違反について経営幹部に通知し、公式声明を準備します。
**E.** ネットワーク全体のエンドポイント上のすべてのウイルス対策署名を更新します。
**Answer: (**解答を表示する**)**
「検知と分析」フェーズでは、イベントがインシデントであるかどうか、その範囲と性質を判断することに重点が置かれます。トラフィックのブロック（A）は封じ込め手順となる場合もありますが、即時の完全なパケットキャプチャとDNSセキュリティログとの相関分析（B）により、正当なサービスに早期に影響を与えることなく分析に必要な重要なデータが得られ、正確なインシデント分類に不可欠です。サーバーの隔離（C）と経営陣への通知（D）は、通常、「封じ込め、根絶、復旧」または「インシデント後の活動」の手順であり、ウイルス対策シグネチャの更新（E）は一般的なセキュリティ対策であり、特定の異常事象に対する主要な検知および分析手順ではありません。

最新問題**: 100**
独自アプリケーション向けに高度にカスタマイズされたデータ取り込みパイプラインを備えたXSIAMの顧客は、独自の解析ロジックと関連データモデルをコンテンツパックとして業界コンソーシアム内の他の組織と共有したいと考えています。彼らは、業界固有の脅威を特定するために、これらのデータモデル用のXQLクエリを開発しました。他のコンソーシアムメンバーによるインポートと操作を確実に成功させるには、コンテンツパックのマニフェストのどの側面、特にデータの可用性と正規化に関して、慎重に定義する必要がありますか？
**A.** コンテンツパックのマニフェストには、独自のアプリケーションログを定義済みデータモデルに正規化するために使用する「データモデル」定義と「XQLパーサー」設定（Grokパターン、JSONパスなど）を明示的に記載する必要があります。さらに、想定される生ログ形式に関する詳細なドキュメントも不可欠です。
**B.** XSIAM はクエリ自体から必要なデータ モデルと解析ロジックを自動的に推測するため、XQL で記述された「検出ルール」のみを含める必要があります。
**C.** コンテンツ パックには、独自のアプリケーションからの生のログ ファイルをバンドルして、他のメンバーがトレーニング目的で直接取り込めるようにする必要があります。
**D.** 他のコンソーシアム メンバーの独自のアプリケーションから直接データを取得するには、コンテンツ パック内で「外部統合」を構成する必要があります。
**E.** コンテンツ パックには、脅威インテリジェンスを共有するための最も移植性の高い要素である「インシデント レイアウト」と「レスポンス プレイブック」のみを含める必要があります。

**Answer: A (**メーッセージを残す**)**
独自のアプリケーション用のカスタム解析ロジックとデータ モデルを共有することは、コンテンツ パック内では複雑なタスクです。
*データモデル定義 :これらは基本的なものです。コンソーシアムの他のメンバーは、正規化されたデータの構造とスキーマを理解する必要があります。
*XQLパーサーの設定 :これは非常に重要です。データは独自仕様でカスタムであるため、コンテンツパックには、生のログを定義されたデータモデルに変換する正確な解析ロジック（例XQL関数の使用、カスタムパーサーの定義など）が含まれている必要があります。
*生ログ形式に関するドキュメント：技術マニフェストに直接含まれているわけではありませんが、想定される生ログ形式を明確に説明した外部ドキュメントは不可欠です。これがなければ、他のメンバーはコンテンツパックの解析要件に合わせてデータ取り込みを設定する方法がわかりません。
オプションBは誤りです。XSIAMはXQLクエリから複雑なカスタム解析を自動的に推論しません。
オプションCは非現実的であり、セキュリティリスクを伴います。
選択肢Dは不正解です。コンテンツパックは、このように他の組織のシステムから直接データを取得することはありません。選択肢Eは、検出後の側面に焦点を当てており、重要なデータ取り込みと正規化の課題を無視しています。

最新問題**: 101**
高度な持続的脅威（APT）グループが大規模組織のネットワークへの侵入に成功し、SOCは「根絶」段階にあります。攻撃者が使用していた複数の侵害されたエンドポイントとC2サーバーを特定しました。このAPTグループは、カスタムマルウェアの亜種と高度な回避技術を使用することで知られています。脅威の根絶、再感染の防止、そして将来の検出能力の向上において、以下のどのアクションとPalo Alto Networksツールを組み合わせた場合、最も強力かつプロアクティブなアプローチとなるでしょうか？
**A.** リアルタイム保護のためにすべてのエンドポイントに Cortex XDR エージェントを展開し、NGFW ですべての C2 IP アドレスをブロックします。
**B.** 侵害されたすべてのエンドポイントの完全な再イメージ化を実行し、NGFW 上のウイルス対策シグネチャを更新します。
**C.** NGFW上のNSX統合（または類似のツール）を介してマイクロセグメンテーションポリシーを適用したネットワークセグメンテーションを実装し、WildFireを活用して新たに発見されたマルウェアに関するカスタム脅威インテリジェンスを生成し、これらのIOCをMineMeldまたはカスタム統合を介してすべてのセキュリティコントロール（NGFW、XDR、SIEM）にプッシュします。同時に、Cortex XDRでXQLハントを実行し、環境全体にわたって類似の攻撃パターンを探します。
**D.** データの流出を防ぐために内部ネットワークからのすべての送信トラフィックをブロックし、すべてのユーザー アカウントに多要素認証 (MFA) を適用します。
**E.** 疑わしいユーザー アカウントをすべて無効にし、ネットワーク全体の脆弱性スキャンを実行します。
**Answer: (**解答を表示する**)**

この質問には、再感染の防止と将来の検出の改善に重点を置いた、根絶段階での APT に対処するための多面的なアプローチが必要です。
1. ネットワークセグメンテーション／マイクロセグメンテーション :ラテラルムーブメント（横方向の移動）を防止し、将来の侵害を封じ込めるために不可欠です。ネットワークをセグメント化することで、たとえ1つのセグメントが侵害されたとしても、影響範囲は限定されます。NSXについて言及しましたが、中核となる概念はマイクロセグメンテーションであり、Palo Alto NGFWでもこれを適用できます。
2. カスタム脅威インテリジェンスのための WildFire: APT はカスタム マルウェアを使用するため、これらの固有のサンプルを分析し、新しいシグネチャと IOC を生成するには WildFire が不可欠です。
3. すべてのセキュリティコントロールへのIOCのプッシュ（MineMeld/カスタム統合）これはプロアクティブな防御にとって非常に重要です。WildFireで新たに生成されたIOCは、NGFW（境界/内部セグメントでのブロック）、Cortex XDR（エンドポイントでの検知と防御）、SIEM（相関分析とアラート生成）に即座にプッシュする必要があります。MineMeldは、脅威インテリジェンスを共有および活用するためのPalo Alto Networksツールです。
4. Cortex XDRにおけるXQLハント :APT攻撃は、持続的かつ広範な侵害を意味します。環境全体を対象としたXQLハントは、攻撃の他のインスタンス、未確認の侵害システム、またはAPT活動の残骸を見つけるために不可欠です。これは、単純な駆除にとどまらず、全範囲のセキュリティを確保し、見落とされたコンポーネントからの再感染を防ぐことにもつながります。
他のオプションを評価してみましょう:
A: 良いことですが、XDR を展開して IP をブロックするだけでは、回避的なカスタム マルウェアや潜在的に動的な C2 を使用する APT には不十分です。
B: 再イメージ化は根絶の一環ですが、AV シグネチャを更新するだけでは、カスタムのゼロデイマルウェアから保護することはできません。
D: すべての送信トラフィックをブロックするのは、混乱を招きやすく、持続可能ではありません。MFAは重要ですが、予防策であり、活動中のAPTに対する根絶戦略ではありません。
E: アカウントの無効化と脆弱性スキャンは重要なステップですが、高度な APT を根絶し、将来の回復力を構築するには十分ではありません。

最新問題: **102**
Palo Alto Networksのセキュリティアーキテクトが、クライアントに「AIドリブンSecOps」と「MLドリブンSecOps」の概念を説明しているところです。経験豊富なSOCマネージャーであるクライアントは、アーキテクトにこう問いかけます。「AIというのは、高度なMLモデルのマーケティング用語に過ぎないのではないでしょうか。膨大なデータを扱うMLのみのシステムでは根本的に不可能なセキュリティタスクを、AIドリブンシステムが確実に実行できる具体的なシナリオを教えてください。」次のシナリオのうち、セキュリティ運用におけるAI独自の能力を最も明確かつ最もよく表しているのはどれでしょうか？

**A.** 機械学習システムは、異常なファイル暗号化パターンを識別することでランサムウェアを検知できます。一方、AIシステムは、攻撃者のTTPを理解し、新しい亜種であっても高い信頼性で前兆となる活動を相関させることで、暗号化が開始される前にランサムウェア攻撃を予測できます。
**B.** 機械学習システムは、学習した特徴に基づいてネットワークトラフィックを悪意のあるものと無害なものに分類できます。AIシステムは、攻撃の意図と影響を理解することで、人間の介入や事前定義されたテンプレートなしに、新しいセキュリティポリシーとファイアウォールルールをリアルタイムで自律的に設計し、新たな攻撃に対抗できます。
**C.** MLシステムは、通常のユーザー行動のベースラインからの逸脱を検出することで、内部脅威を特定できます。AIシステムは、内部関係者と自然言語による対話を行い、より多くのコンテキストを収集し、意図を評価し、人間のアナリストを模倣して修復手順を案内することができます。
**D.** MLシステムは、重大度と信頼度スコアに基づいてアラートの優先順位を決定できます。AIシステムは、アラートの背後にある理由を、具体的な証拠や相関関係を引用しながら、人間が理解できる形式で説明できます。これは、MLシステムでは通常、本質的には不可能です。
**E.** MLシステムはディープラーニングを用いてポリモーフィック型マルウェアを検出できます。AIシステムはポリモーフィック型のデコイファイルを自律的に生成し、ネットワーク全体に配布することで、新たなマルウェアの亜種を捕捉・分析し、インテリジェントなハニーポットシステムとして効果的に機能します。
**Answer: B (**メーッセージを残す**)**
この設問では、AIが「膨大なデータを扱う高度な機械学習」をも凌駕する基礎的な能力を発揮するシナリオを求めています。選択肢Aは予測分析について説明しています。これは高度ではあるものの、依然として高度な機械学習の領域に大きく限定されます。機械学習モデルはパターンに基づいて予測を学習できます。選択肢Cは自然言語処理／理解について説明しています。これはAIの一分野ですが、「対話」の部分はNLPの特定の応用であることが多く、一般的なセキュリティ運用における機械学習を超えたAI全体の根本的な差別化ではありません。また、「修復を通じたガイド」はスクリプト駆動型でも可能です。選択肢Dは説明可能なAI（XAI）について説明しています。これは現代のAIの重要な側面ですが、中核となる「検出」や「アクション」は依然として機械学習に根ざしていることが多いです。説明は機械学習の出力に基づいて構築できます。選択肢Eは、最先端の高度な研究指向のAI機能（防御／欺瞞のための生成AI）について説明しています。これはまだ広く普及しておらず、すべてのAIシステムが実行でき、機械学習が根本的に実行できない「セキュリティ運用」の中核タスクです。これはAIの応用ではありますが、一般的な概念における最も根本的な違いではないかもしれません。オプションBは真に根本的な飛躍を表しています。攻撃の意図と影響を理解し、事前にプログラムされたテンプレートや人間の介入（初期の「学習」フェーズ以降）に頼ることなく、コンテキストアウェアな新しいセキュリティポリシーとファイアウォールルールを自律的に設計する能力は、パターン認識（ML）から、新しい状況における認知的、創造的な問題解決、そして自律的な意思決定へと境界を越え、強いAIの特徴となっています。MLのみのシステムは分類や検知はできますが、真に自律的かつ適応的な方法で新しいルールやポリシーを「設計」することはできません。

最新問題: **103**
Cortex XSIAM によって、内部者によるデータ流出の可能性がある重大なインシデントが検出されました。このインシデントは、特定のユーザーアカウントが機密データ共有にアクセスし、承認されていないクラウドストレージサービスへの大規模なアウトバウンドファイル転送を開始したことを示しています。法的手続きのためにフォレンジック証拠を収集し、さらなる流出を阻止する必要があります。XSIAM の機能を活用した以下のアクションのうち、このシナリオにおいて最も適切かつ重要なものはどれですか？
**A.** 実行する
**B.** さらなるデータ損失を防ぐために、直ちにユーザーのパスワードを変更し、アカウントを無効にします。
**C.** XSIAM では詳細なフォレンジック データが提供されないため、サードパーティ ツールを使用してユーザーのワークステーションの完全なディスク フォレンジック イメージを開始します。
**D.** 境界ファイアウォールでのネットワーク トラフィック分析のみに焦点を当て、流出先を特定してブロックします。
**E.** レビューのみ
**Answer: A (**メーッセージを残す**)**
このシナリオでは、法的手続きのために封じ込めと詳細なフォレンジック調査の両方が必要です。オプションAは最も包括的かつ適切です。エンドポイント分離により、脅威は即座に封じ込められます。XQLを使用してfile_eventおよびnetwork_connectionデータセットをクエリすることは、アクセスされたデータとその移動先を把握するために不可欠です。ユーザーアクティビティログと監査ログを収集することで、ユーザーの行動とアクセスの詳細を含む、法的手続きに必要な証拠が得られます。オプションBは対応策ですが、フォレンジック証拠は提供されません。Cは誤りです。XSIAMは豊富なフォレンジックデータを提供しますが、完全なディスクイメージは多くの場合時間がかかりすぎ、初期段階として必ずしも必要ではありません。Dは範囲が狭すぎ、内部ユーザーの行動が欠落しています。Eは、内部者によるデータ流出シナリオには無関係です。

最新問題: **104**
ある組織はCortex XSIAMを導入しており、その機能を最大限に活用して、ラテラルムーブメントやコマンドアンドコントロール（C2）通信を伴う高度な攻撃を検知したいと考えています。オンプレミスのデータセンター、AWSクラウドインフラストラクチャ、そして多数のリモートワークフォースが混在する環境で運用されています。包括的な可視性を実現するために、Cortex XSIAMセンサーのどの組み合わせが最も効果的でしょうか？また、それぞれのセンサーはどのような種類のデータを活用すれば、このような脅威の特定に役立つでしょうか？
**A.** ネットワークフローとプロセスデータ用のホストセンサー（エンドポイントエージェント）と、API呼び出し用のクラウドセンサー（CloudTrail）。この組み合わせにより、ホストコンテキストとクラウド環境におけるC2とラテラルムーブメントをそれぞれ効果的に検出します。
**B.** ネットワーク会話とDNSクエリを検知するネットワークセンサー（NetFlow、パケットキャプチャ）と、プロセス実行とファイルアクセスを検知するホストセンサー（エンドポイントエージェ

ント）。この組み合わせにより、C2（ネットワーク層）とラテラルムーブメント（ホスト間アクティビティ）を検知するための強力な基盤が構築されます。
**C.** 認証試行に関するアイデンティティセンサー（Active Directoryログ）と、クラウドネットワーク内部トラフィックに関するクラウドセンサー（VPCフローログ）。この組み合わせは、認証異常とクラウドネットワークの可視性に主眼を置いており、C2やホストレベルの詳細な横方向の移動にはそれほど重点を置いていません。
**D.** コンテナアクティビティ用のコンテナセンサー（Kubernetes監査ログ）と、産業用制御システムデータ用のOT/IoTセンサー。特定の環境では重要ですが、この組み合わせは一般的な企業におけるラテラルムーブメントやC2を広範囲にカバーすることはできません。
**E.** 環境に関係なく、横方向の移動と C2 検出の両方に必要なすべてのデータをキャプチャできるため、ホスト センサー (エンドポイント エージェント) のみで十分です。
**Answer: B (**メーッセージを残す**)**
ラテラルムーブメントとC2を含む高度な攻撃を検知するには、多面的なセンサーアプローチが不可欠です。ネットワークセンサー（NetFlowや専用のパケットキャプチャセンサーなど）は、ネットワークの会話、DNSクエリ、そして全体的なトラフィックパターンを観察するのに優れており、これらはC2チャネルの特定に不可欠です。ホストセンサー（エンドポイントエージェント）は、プロセス実行、ファイルシステムのアクティビティ、レジストリの変更、そしてローカルネットワーク接続に関する詳細な可視性を提供し、攻撃者がホスト内およびホスト間でどのようにラテラルムーブメントを行っているかを把握するために不可欠です。ネットワークテレメトリとホストテレメトリを組み合わせることで、これらのタイプの脅威に対する最も包括的な視点が得られます。

最新問題**: 105**
カスタムアプリケーションに影響を与える重大なゼロデイ脆弱性が公開されました。SOCは、現在非標準の複数行形式でローカルファイルに書き込まれているアプリケーション固有の監査ログをCortex XSIAMに取り込み、即時の脅威ハンティングを行う必要があります。この特定のアプリケーションには、既存の統合環境がありません。XSIAM内での迅速な取り込みとその後の脅威ハンティングに最も適したアプローチはどれですか？また、対処すべき主要な課題は何ですか？
**A.** アプリケーションサーバーにCortex XDRAgentをインストールし、ログファイルを監視するためのデータ収集プロファイルを設定します。重要な課題は、複数行形式に対応した堅牢なXQL解析ルールを作成することです。
**B.** アプリケーションを変更してログをSyslogサーバーに直接送信し、XSIAMでSyslogコレクターを設定します。主な課題は、アプリケーションの変更と、複数行のイベントからコンテキストが失われる可能性があることです。
**C.** ログファイルの末尾を追跡し、複数行のイベントを1行のJSONに正規化し、XSIAM Ingestion API経由でプッシュするカスタムスクリプトを作成します。重要な課題は、カスタムスクリプトの開発と保守です。

**D.** 専用のログコレクターを導入し、「ファイル」データソースを指定したログプロファイルを設定し、カスタム解析ルール内でGrokパターンを使用して複数行形式を処理します。重要な課題は、複数行イベントに対して複雑なGrokパターンを正確に定義することです。
**E.** Filebeatなどのサードパーティ製ログフォワーダーを使用してログをKafkaトピックに送信し、Cortex XSIAMを設定してKafkaからログを取得します。重要な課題は、Kafkaインフラストラクチャの設定と管理です。
**Answer: D (**メーッセージを残す**)**
アプリケーションの変更やカスタム スクリプトを使用せずに、ローカルで非標準の複数行ファイルを迅速に取り込むには、専用のログ コレクターを導入するのが一般的に最も適したネイティブ XSIAM アプローチです。ログ コレクターの 「ファイル」データ ソース タイプは、このために設計されています。正しく認識されているように、主な課題は、複数行イベントを処理して関連フィールドを抽出するために、カスタム解析ルール内で正確で堅牢な grok パターンを作成することです。XDR エージェント (A) はファイルを収集できますが、高度にカスタマイズされた複数行形式の解析機能は、grok を備えた専用のログ コレクターよりも柔軟性が低い可能性があります。Syslog (B) は複数行イベントの処理に苦労することがよくあります。カスタム スクリプト (C) は強力ですが、開発時間と継続的なメンテナンスが必要です。Kafka (E) は、より直接的な取り込みを可能にする重要な追加インフラストラクチャを導入します。したがって、D はこの特定の課題に対する最も直接的で効果的な XSIAM ネイティブ ソリューションです。