Performing CyberOps Using Cisco Security Technologies v1.0 (350-201)

試験の概要：CyberOps Using Cisco Security Technologies v1.0（CBRCOR 350-201）は、Cisco CyberOps Professional 認定に関連する試験であり、試験時間は 120 分です。この試験では、サイバーセキュリティの基礎、技術、プロセス、自動化を含む、サイバーセキュリティの中核となる運用に関する知識が試されます。本試験の受験対策として、Performing CyberOps Using Cisco Security Technologies コースの受講をお勧めします。

以下に、この試験の出題内容の概要を示します。ただし、試験によっては、ここに示されていない関連項目も出題される場合があります。試験内容をより適切に反映し、明確にするために、次のガイドラインは予告なく変更されることがあります。

20% 1.0 基礎

- 1.1 プレイブック内のコンポーネントの解釈
- 1.2 プレイブックのシナリオに基づいた必要なツールの決定
- 1.3 一般的なシナリオ（特権の不正昇格、DoS や DDoS、ウェブサイトの改ざんなど）へのプレイブックの適用
- 1.4 様々なコンプライアンス規格（PCI、FISMA、FedRAMP、SOC、SOX、PCI、GDPR、Data Privacy、ISO 27101 など）を巡る業界予測
- 1.5 サイバーリスク保険の概念と限界
- 1.6 リスク分析の要素の分析（情報資産、脆弱性、脅威の組み合わせ）
- 1.7 インシデント対応ワークフローの適用
- 1.8 一般的なインシデント対応メトリックを使用した改善の特徴と領域
- 1.9 クラウド環境の種類
- 1.10 クラウド プラットフォーム（IaaS、PaaS など）におけるセキュリティ運用上の考慮事項の比較

30% 2.0 技術

- 2.1 特定のニーズを満たすため、または特定の質問に答えるために推奨されるデータ分析の技術
- 2.2 セキュリティ強化されたマシンイメージを使用した展開
- 2.3 資産のセキュリティ体制を評価するプロセス
- 2.4 環境のセキュリティ管理の評価、ギャップの診断、改善の提案
- 2.5 システムのセキュリティ強化のための業界標準および推奨事項のリソース
- 2.6 （所定のシナリオにおける）推奨されるパッチ適用の特定
- 2.7 （所定のシナリオにおける）無効化が推奨されるサービス
- 2.8 ネットワークへのセグメンテーションの適用
- 2.9 ネットワークコントロールを活用したネットワーク セキュリティの強化
- 2.10 SecDevOps の推奨事項（含意）

- 2.11 インテリジェンスを自動化する脅威インテリジェンス プラットフォーム（TIP）の使用方法と概念
- 2.12 ツールを使用した脅威インテリジェンスの適用
- 2.13 データ損失、データ漏洩、移動中／使用中／静止中のデータといった概念の一般的な基準に基づいた適用
- 2.14 データ損失防止技術の検出および実施のためのさまざまな仕組み
  - 2.14.a ホストベース
  - 2.14.b ネットワークベース
  - 2.14.c アプリケーションベース
  - 2.14.d クラウドベース
- 2.15 デバイスおよびソフトウェアに対して実施が推奨される調整／適用（ルール、フィルタ、ポリシー）
- 2.16 セキュリティ データ管理の概念
- 2.17 セキュリティ データ分析のためのツールの使用と概念
- 2.18 エスカレーションを通して説明された問題に対しての解決に必要な自動化までのワークフローの提言
- 2.19 利害関係者（テクニカル／リーダーシップ／エグゼクティブ）とのコミュニケーションを図るためのダッシュボード データの適用
- 2.20 ユーザ／エンティティの行動分析（UEBA）
- 2.21 ユーザの行動アラートに基づいた次のアクションの決定
- 2.22 ネットワーク解析のためのツール（パケットキャプチャ ツール、トラフィック解析ツール、ネットワーク ログ解析ツールなど）とその限界
- 2.23 パケット キャプチャ ファイルのアーティファクトおよびストリームの評価
- 2.24 既存の検出ルールのトラブルシューティング
- 2.25 攻撃の戦術／技術／手順（TTP）の見極め

30% 3.0 プロセス

- 3.1 脅威モデルの構成要素の優先順位付け
- 3.2 一般的なタイプのケースを調査するステップ
- 3.3 マルウェア解析プロセスにおけるステップの概念およびシーケンスの適用
  - 3.3.a （パケットキャプチャやパケット分析ツールから）分析用のサンプルを抽出して識別する
  - 3.3.b リバース エンジニアリングを行う
  - 3.3.c サンドボックス環境を使用して動的マルウェア解析を実行する
  - 3.3.d 静的マルウェア分析を追加する必要性を特定する
  - 3.3.e 静的マルウェア分析を実行する
  - 3.3.f 結果をまとめて共有する
- 3.4 攻撃時の一連のイベントのトラフィックパターンの分析に基づいた解釈
- 3.5 さまざまな種類のプラットフォーム（デスクトップ、ラップトップ、IoT、モバイルデバイスなど）が関わる環境でのエンドポイント侵入の可能性を調査する手順
- 3.6 （所定のシナリオにおける）侵害の痕跡（IOC）と攻撃の痕跡（IOA）
- 3.7 サンドボックス環境での IOC（複雑な指標の生成を含む）
- 3.8 （所定のシナリオにおける）モダリティのさまざまなベクトル（クラウド、エンドポイント、サーバー、データベース、アプリケーション）で発生するデータ損失の危険性を調査する手順
- 3.9 脆弱性の問題に対処するために一般に推奨されている緩和策の手順

3.10 業界のスコアリング システム（CVSS など）やその他の手法を用いた脆弱性のトリアージおよびリスク分析のために推奨される次のステップ

20% 4.0 自動化

4.1 オーケストレーションおよび自動化の概念、プラットフォーム、メカニズムの比較
4.2 基本的なスクリプトの解釈（Python など）
4.3 セキュリティ操作タスクの自動化（指定されたスクリプトを修正する）
4.4 一般的なデータ形式（JSON、HTML、CSV、XML など）
4.5 自動化およびオーケストレーションの機会
4.6 API を消費する際の制約（レート制限、タイムアウト、ペイロードなど）
4.7 REST API に関連する一般的な HTTP レスポンスコード
4.8 HTTP レスポンスの構成要素（レスポンスコード、ヘッダ、ボディ）の評価
4.9 API 認証メカニズムの解釈（ベーシック、カスタムトークン、API キー）
4.10 Bash コマンドの活用(ファイル管理、ディレクトリ ナビゲーション、環境変数)
4.11 CI/CD パイプラインの構成要素
4.12 DevOps プラクティスの原則の適用
4.13 コードとしてのインフラストラクチャの原則