2026 年 5 月

Qualys Custom Assessment and Remediation (CAR)

トレーニングガイド

はじめに

このコースでは、環境内のデータを正確に維持し、一貫性と信頼性の高いレポートを生成するためのベストプラクティスを学びます。また、組織内の様々なステークホルダー向けのレポートを生成するための、様々な戦略とツールの活用方法も学びます。

下記のフリートレニング Custom Assessment and Remediation にエントリし、この資料とともにコースを完了する事を推奨します。
https://www.qualys.com/training/

アジェンダ

このレッスンでは、以下のトピックについて演習やビデオを見ながら説明してゆきます。各セクションの後にショートテストがあります。

- Qualys センサー概要
- Qualys クラウドエージェント
- カスタム評価・修復(CAR)入門
- CAR スクリプトワークフロー
- カスタムアプリケーション制御用スクリプト
- スクリプト結果チェック UDC
- 緊急のセキュリティ脅威にスクリプトで対応

はじめに

カスタム評価および修復（CAR）に関するトレーニングへようこそ。CAR アプリケーションを使用すると、既存の脆弱性対策プログラム、セキュリティプロセス、および分析とシームレスに統合されたカスタムスクリプト制御を使用して、戦術的なセキュリティワークフローを自動化できます。

コースの目標
このコースを修了すると、以下のことができるようになります。

- CAR が特殊な環境においてなぜ重要なのかを理解しましょう。
- CAR UI の基本を確認し、ジョブ、スクリプト、ライブラリの機能を理解してください。
- CAR がゼロデイ攻撃への対応にどのように役立つかを理解しましょう。
- Qualys ライブラリからカスタムスクリプトをアカウントにインポートする方法をご覧ください。
- 固有の QID を作成し、Qualys ナレッジベースに追加してください。
- 非標準的な脆弱性問題への対処に役立つ、独自のユースケースをご覧ください。
- Qualys CAR がサポートする Windows および Linux のスクリプト言語を特定します。
- Qualys CAR ワークフローのすべての手順を実行します。
- 事前に定義された CAR ユーザーロールを使用して、職務分掌を徹底し、スクリプトの整合性を確保します。
- CAR と Qualys Policy Compliance を統合してユーザー定義コントロール（UDC）を構築するプロセスについて解説します。
- Qualys Policy Compliance で「スクリプト結果チェック」UDC を作成します。

なぜ CAR が必要か？

企業の IT 環境はそれぞれ異なり、それに伴って直面するセキュリティ上の課題も多様です。標準的なセキュリティツールでは一般的な脆弱性や設定ミスを検知することはできますが、組織ごとに固有の要件や状況に完全に対応できるとは限りません。例えば、特定のソフトウェアやライブラリが環境全体で使われているかを確認したい場合や、既知の不正ファイルのハッシュ値をもとに危険なファイルを検出したい場合など、より個別的で高度な調査や対応が求められることがあります。また、ゼロデイ脆弱性への対応や、パッチ適用後の追加確認といった場面でも、迅速かつ柔軟な対応が必要になります。

このような背景から、CAR は「**標準では対応しきれない課題に対して、独自の検知や対応を実装できる仕組み**」として提供されています。ユーザーは自身の環境に合わせてカスタムルールを作成し、必要な情報を収集したり、特定の条件に合致した場合に自動的に対処を行ったりすることができます。つまり、CAR は単なる診断ツールではなく、「調査」と「対処」を一体的に実現する機能です。

具体的には、CAR ではまず「**Custom Assessment（カスタム検知）**」を定義します。これは、ファイルの存在確認や設定値のチェック、特定のハッシュ値との照合といった調査内容を自由に記述できる仕組みです。そして、問題が検知された場合には「**Remediation（対応）**」として、ファイルの削除や設定の変更、コマンドの実行などのアクションを自動で実施することが可能です。このように、ユーザー自身が検知と対応のロジックを定義できる点が、CAR の大きな特徴です。

また、この機能の重要な利点として、既存の Qualys エージェントをそのまま活用できる点が挙げられます。多くの環境では、すでに脆弱性管理（VMDR）やポリシーコンプライアンスのために Qualys エージェントが導入されていますが、CAR はそれらと同じエージェントを利用します。そのため、新たなソフトウェアを導入する必要はなく、設定を有効化することで追加機能として利用を開始できます。これにより、導入の負担を抑えつつ、より高度なセキュリティ対応が可能になります。

資料ではさらに、CAR の利用を理解するために、基本的な操作画面の説明や代表的なユースケースの紹介、チュートリアルの実施を通じて、実際の使い方を学ぶ構成になっています。代表的な利用シナリオとしては、不正ファイルや不審な設定の検出といったセキュリティ調査、脆弱性対応の確認、インシデント発生時の迅速な対応、さらには独自のコンプライアンスチェックなどが挙げられます。

このように、**CAR は組織ごとに異なるセキュリティ課題に柔軟に対応するための強力な仕組み**であり、特に標準機能ではカバーしきれないケースや、緊急性の高い対応が求められる場面で大きな価値を発揮します。一方で、自由度が高い分、設定内容によっては誤検知や意図しない影響を引き起こす可能性もあるため、運用設計やテストを十分に行うことが重要です。

CAR の 4 つの柱

本セクションでは、Qualys の「Custom Assessment and Remediation（CAR）」機能に関する基本概念、アーキテクチャ上の特徴、および主要なユースケースについて体系的に解説します。

前提として、CAR は既存の Qualys Cloud Agent を基盤として動作する点が特徴です。このエージェントは、既に多くの環境で脆弱性管理（VMDR）、ポリシーコンプライアンス（PA）、パッチ管理（PM）、ファイル整合性監視（FIM）などに活用されています。CAR はこれらと同一エージェントを利用するため、新たなエージェントの導入を必要とせず、既存の運用基盤の延長としてシームレスに機能を拡張できます。この統合アーキテクチャにより、収集されたデータは単一プラットフォーム上で相関分析され、セキュリティ可視性の向上および運用効率の最適化が実現されます。

CAR の中核機能は、「**カスタムスクリプトの実行による検知および対応の自動化**」にあります。ユーザーは Windows、Linux、Unix、macOS といった多様な OS 環境に対してスクリプトを展開し、任意の評価ロジック（Assessment）および対処ロジック（Remediation）を実装することが可能です。これにより、標準的な脆弱性スキャンではカバーできない組織固有の要件や、環境依存のリスクに対して柔軟に対応できます。

加えて、CAR では**ユーザー独自の脆弱性識別子（QID: Qualys ID）を作成**することが可能である。これにより、標準の Qualys KnowledgeBase に登録されていない独自のリスク要因を定義し、それを

既存の脆弱性管理プロセスに統合できます。作成されたカスタム QID は、True Risk Score などのリスク評価指標に組み込むことができるため、環境固有の脅威コンテキストを考慮したリスクベースの優先順位付けが実現されます。この点は、標準的な脆弱性管理を高度化する上で重要な機能です。

さらに、CAR は**柔軟な修復ワークフローの実装をサポート**します。ユーザーは特定の条件に基づいて自動的に修復処理を実行するワークフローを構築でき、組織の運用ポリシーや対応優先度に応じたカスタム対応が可能となります。これにより、脆弱性の検知から修復までのリードタイム短縮、すなわち Mean Time To Remediate（MTTR）の削減が期待できます。

CAR の活用を支援する要素として、「**Qualys Library**」の存在も重要です。このライブラリには、Qualys が事前に作成したスクリプトが格納されており、ユーザーはこれらをそのまま利用するか、または自社環境に合わせてカスタマイズすることができます。ライブラリは継続的に更新されるため、新たに報告された脆弱性や緊急度の高いセキュリティ課題に対して迅速に対応できます。この仕組みは、インシデント対応時の初動を大幅に効率化します。

具体的なユースケースとしては、**オープンソースライブラリの脆弱性検査**が挙げられます。たとえば、アプリケーションに含まれる特定のライブラリ（例：JAR ファイル）について、脆弱なバージョンの有無をシステム全体で検出することが可能です。さらに、検出後には自動的な修復処理や、パッチ適用後の追加検証、あるいは一時的なリスク軽減措置を実施することができます。また、平文で保存された認証情報の検出、不正なハッシュ値の探索、非標準のブラウザ拡張の検知など、広範なセキュリティチェックに適用可能です。

CAR は Qualys の VMDR と密接に統合されており、資産の重要度や既存の脆弱性情報と組み合わせて、よりコンテキストに基づいたリスク評価を提供します。これにより、単なる「**脆弱性の有無**」ではなく、「**ビジネス影響やリスク優先度を加味した判断**」が可能となります。

総じて、CAR の導入により、従来は複数のツールやプロセスを組み合わせて実現していたカスタム検知・対応が、単一プラットフォーム上で完結するようになります。この結果、ツール間連携の複雑性が低減され、総所有コスト（TCO）の削減とともに、セキュリティ運用の迅速化および高度化が実現されます。

このように、CAR は従来の脆弱性管理を拡張し、組織固有のリスクに対して俊敏かつ柔軟に対応するための重要なコンポーネントです。特に、標準検知では対応が困難なケースや、迅速な対応が求められる状況において、その価値を最大限に発揮するでしょう。

CAR 概要

本セッションでは、Qualys の「Custom Assessment and Remediation（CAR）」機能について、実践的に活用できるレベルまで理解を深めていきます。本コースを修了する頃には、受講者の皆様は、自身

の環境におけるシステムやアプリケーションに対して、カスタムデータ収集および分析を行い、独自の検知および対応を実施できるようになります。

まず、CAR の主要な機能として、**Qualys Library から既存のスクリプトをインポートして活用**する方法を学びます。これにより、ゼロからスクリプトを開発することなく、事前に用意された検知・対応ロジックを迅速に導入することが可能になります。また、自身でカスタムスクリプトを作成する場合においても、既存のスクリプトをベースにカスタマイズすることで、効率的に実装を進めることができます。

さらに、CAR は Qualys の VMDR（Vulnerability Management, Detection and Response）アプリケーションと密接に統合されています。受講者の皆様は、**独自の脆弱性識別子（QID）を作成し、それを KnowledgeBase に登録する方法を学びます。**このカスタム QID を既存の脆弱性情報と統合することで、True Risk Score などのリスク評価指標に反映させることができ、環境固有のリスクを考慮したより高度な脆弱性管理が可能になります。

加えて、**CAR は Policy Audit とも連携可能**です。カスタムコントロールを作成し、それらの評価結果を取得・分析することで、組織固有のセキュリティポリシーやコンプライアンス要件に対応したチェックを実現できます。このように、CAR は単なる検知ツールではなく、複数のセキュリティ機能と統合された拡張プラットフォームとして機能します。

また、CAR の運用においては**ロールベースアクセス制御（RBAC）**が適用されます。**スクリプトの作成、テスト、実行といった各プロセスに対して適切な権限管理を行うことで、安全かつ統制された運用を実現することができます。**

CAR の位置づけとして重要なのは、従来の脆弱性スキャンの「**次のステップ**」であるという点です。標準的な脆弱性スキャンやコンプライアンスチェックは、セキュリティ対策の第一段階として非常に重要ですが、それだけでは対応しきれないケースが存在します。たとえば、**ゼロデイ脆弱性や緊急性の高いセキュリティ脅威、あるいは組織固有の要件に基づく非標準的なチェック**などがこれに該当します。このような状況において、CAR は柔軟かつ迅速な対応手段を提供します。

従来の運用では、こうした対応を実現するために、**別チームによるスクリプト開発、別ツールによる配布、データの収集・統合、そして分析といった複数のプロセスが必要でした。**しかし、**CAR を活用することで、これらの一連の流れを単一プラットフォーム上で完結させることができます。**これにより、修復までの時間（MTTR: Mean Time To Remediate）を大幅に短縮し、セキュリティ運用の効率化を実現します。

CAR の利用を開始するには、まず **Qualys の Technical Account Manager（TAM）もしくはカスタマーサービスに依頼し、サブスクリプション上で機能を有効化**する必要があります。有効化後は、既に Cloud Agent が展開されていれば、そのまま CAR を利用開始することが可能です。プラットフォーム上では、「**Resources**」セクション内の IT Security カテゴリから「**Custom Assessment and Remediation**」にアクセスし、各種ガイドやドキュメントを参照できます。

また、CAR は **Qualys Cloud Agent を基盤として動作します**。このエージェントは各ホストに 1 つずつインストールされ、ローカルサービスとして実行されます。エージェントはデータ収集機能を担うだけでなく、CAR で定義されたスクリプトを実行する役割も持っています。収集されたデータは Qualys プラットフォーム上で評価され、脆弱性情報やポリシー違反などと相関分析されます。

なお、CAR を利用する際には、対象となる OS や技術スタックが対応しているかを「**Platform Availability Matrix**」で事前に確認することが重要です。対応範囲は継続的に拡張されているため、最新の情報については TAM に確認することを推奨します。

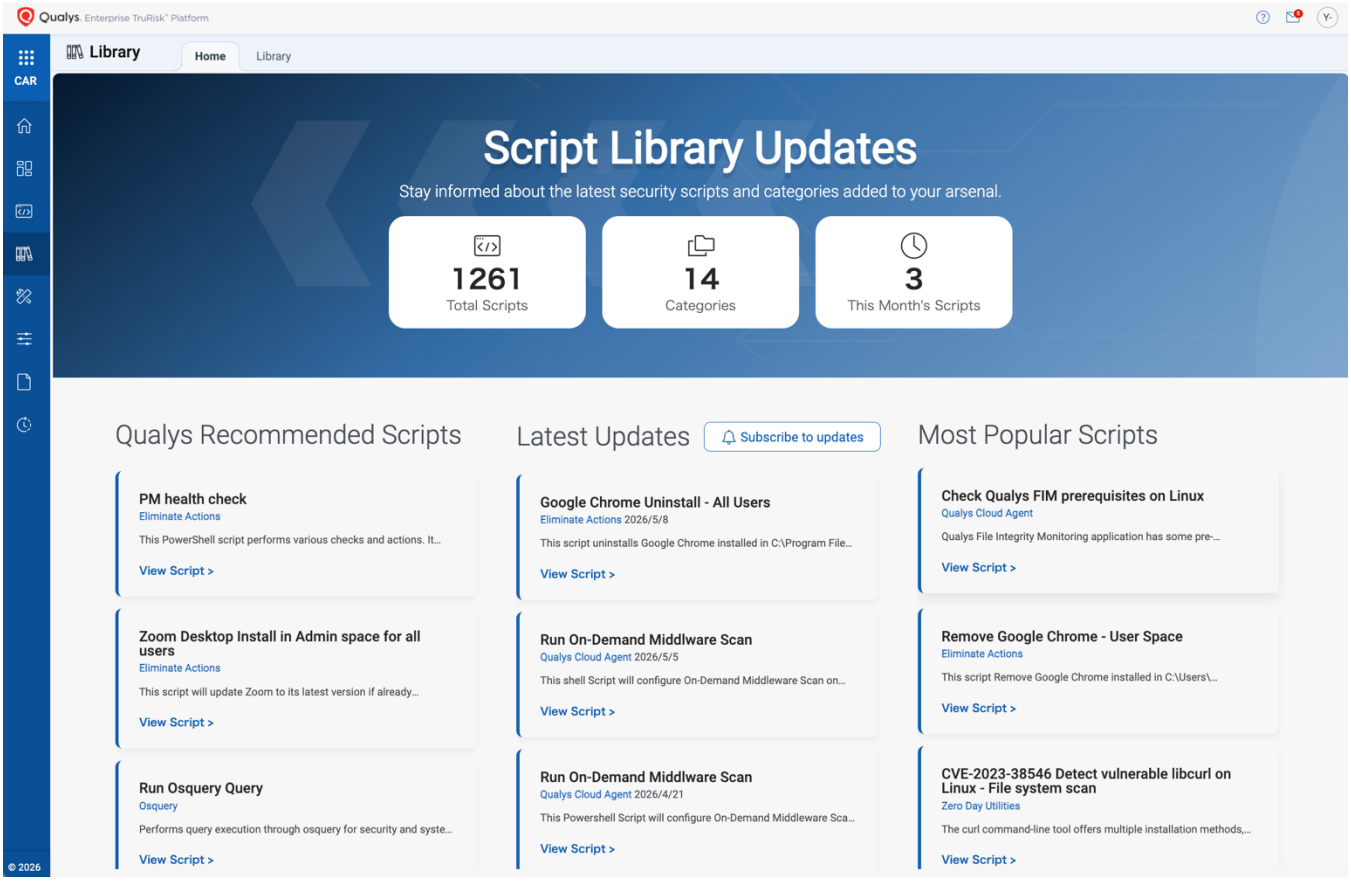
以上のように、CAR は従来の脆弱性管理を補完・拡張し、組織固有のセキュリティ課題に対して柔軟かつ迅速に対応するための重要な機能です。本トレーニングを通じて、その活用方法を体系的に習得していきましょう。

CAR Library Demo

本セクションでは、Qualys の「Custom Assessment and Remediation（CAR）」機能における重要な構成要素の一つである「**Qualys Library**」について解説します。**CAR を活用し始める際には、このライブラリから理解することを強く推奨します。Video はこちらで視聴できます。**

Qualys Library には、Qualys によって事前に作成されたスクリプトが格納されています。これらのスクリプトは、そのまま**各ユーザーのサブスクリプションにインポートして利用することが可能**です。特に、緊急性の高いセキュリティ対応が求められる場面においては、ゼロからスクリプトを開発するのではなく、ライブラリに存在するスクリプトを活用することで、迅速な対応を実現できます。

さらに、これらのスクリプトは単にインポートして使用するだけでなく、ユーザー自身の環境に合わせて編集・カスタマイズすることもできます。たとえば、既存のスクリプトが自社の要件に類似している場合、その内容をベースに微調整を行うことで、効率的に独自の対応ロジックを構築することが可能です。このように、Qualys Library は「**再利用可能なナレッジベース**」として機能し、CAR 活用の出発点として重要な役割を担います。



画面上では、左側にカテゴリ別のフィルタが用意されており、用途に応じたスクリプトを効率的に検索することができます。代表的なカテゴリの一つに「**QID without Patch**」があります。このカテゴリには、ベンダーから正式なパッチが提供されていない脆弱性に対処するためのスクリプトが含まれています。これらは主に「**レスポンス型スクリプト**」であり、たとえば対象ソフトウェアのアンインストールや存在確認といった対策を自動化するものです。具体例として、サポートが終了した Adobe Flash Player を検出し、自動的にアンインストールするスクリプトが挙げられます。

また、各スクリプトには「**Script ID**」が付与されており、ダッシュボードやレポート、ウィジェットなどで参照する際に利用されます。スクリプトの詳細画面では、対象プラットフォーム（例：Windows、Linux）、カテゴリ、スクリプト種別（カスタム／レスポンスなど）、および実際のスクリプト内容を確認することができます。必要に応じてスクリプトをコピーし、ローカルで編集して再利用することも可能です。

次に「**Post Patch」カテゴリ**について説明します。このカテゴリには、パッチ適用後に必要となる追加対応を実行するスクリプトが含まれています。多くの場合、脆弱性の完全な修復には単一のパッチ適用だけでは不十分であり、レジストリ変更や設定変更といった追加の手順が必要となります。これらの後処理を自動化することで、修復の完全性と運用効率を向上させることができます。

さらに、**「0 Day Utilities」カテゴリ**では、ゼロデイ脆弱性への対応を目的としたスクリプトが提供されています。これらのスクリプトは、特定の脆弱なソフトウェアバージョンの検出や、一時的なリスク軽減策の実装を支援します。たとえば、特定の libcurl の脆弱バージョンを検出するスクリプトなどが含まれており、関連する CVE 情報とともに提供されます。これにより、どの脆弱性に対処しているのかを明確に把握しながら対応を進めることができます。

スクリプトを実際に利用する際には、「Quick Actions」メニューから簡単にインポートすることができます。インポートされたスクリプトは一覧画面で確認でき、ライブラリ由来のものには専用のアイコンが表示されるため、視覚的に識別することが可能です。

なお、必要なスクリプトがライブラリに存在しない場合には、自身で新たにスクリプトを作成する方法に加え、Technical Account Manager（TAM）に相談することで、今後の提供予定や代替手段について情報を得ることができます。

以上のように、Qualys Library は CAR の活用において非常に重要な役割を果たします。まずは既存のスクリプトを理解・活用し、その上で自社環境に最適化していくことが、効率的かつ効果的な CAR 運用への第一歩となります。

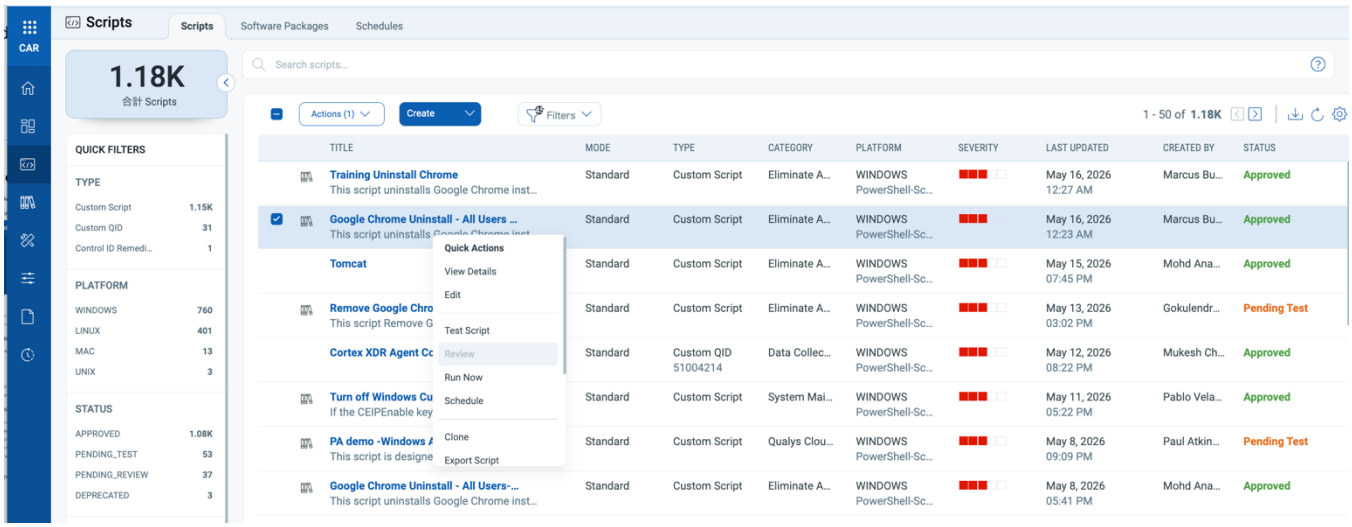
Jobs Demo

本セクションでは、Qualys CAR における**「Library」**に続く重要な機能である**「Scripts」およびスクリプト実行と結果確認のプロセスについて解説します。Jobs Demo** はこちらからご視聴ください。

まず、「Library」が Qualys によって提供される事前構築済みスクリプトの保管場所であるのに対し、「Scripts」セクションは、自組織で管理するスクリプトを確認・運用するための領域です。このセクションでは、現在**テスト中のスクリプト、承認プロセス中のスクリプト、そして本番環境で実行されているスクリプトを一元的に管理**することができます。これにより、スクリプトのライフサイクル全体を可視化し、統制の取れた運用を実現できます。

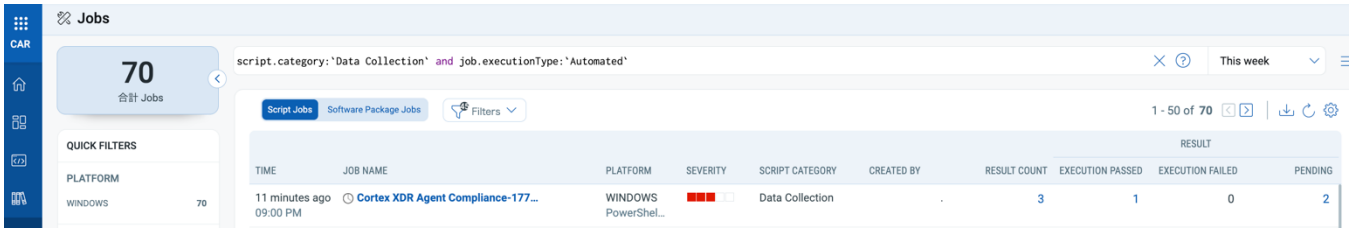
スクリプトは、作成後すぐに本番環境で実行されるわけではなく、**テストおよび承認のプロセスを経て運用に移行**します。本トレーニングでは詳細なワークフローを別途解説しますが、スクリプトが承認された後は、適切な権限を持つユーザーが実行またはスケジュール実行を行うことが可能になります。

スクリプトの実行は、**「Quick Actions」**メニューから行います。ここでは**「Run Now（即時実行）」**または**「Schedule（スケジュール実行）」**を選択できます。なお、承認済みのスクリプトであっても、必要に応じて再度テストを行うことが可能です。特にスクリプト内容を変更した場合には、再テストを実施することが推奨されます。



実行時の重要なポイントとして、「**タグの選択**」が挙げられます。CAR ではスクリプトの適用対象をタグによって指定します。たとえば、Linux 向けスクリプトであれば Linux 環境に対応するタグを選択する必要があります。適切でないタグを選択した場合、スクリプトは対象ホストで実行されない、あるいは結果が返らない可能性があります。このため、対象環境に応じた正しいタグ選択が重要です。

また、**タグの柔軟性により、スクリプトを段階的に展開する運用も可能**です。たとえば、特定のロケーションや部門に属するホストのみに対して実行し、問題がないことを確認してから全体に展開するといった方法を取ることができます。
スクリプト実行後は、「**Jobs**」機能を利用して結果を確認します。「View Latest Jobs」を選択することで、直近の実行結果にアクセスできます。実行結果では、**スクリプトが適用された資産数や、成功・失敗・保留といったステータスを確認**できます。



各資産に対する結果は「**Return Code**」によって示されます。たとえば、あるコードは「**脆弱性未検出（正常）**」を意味し、別のコードは「**脆弱性検出**」を示します。このように、Return Code を用いることで詳細画面に入らずとも結果の概要を迅速に把握することが可能です。また、特定の Return Code でフィルタリングを行うことで、問題のある資産のみを抽出することもできます。

一方で、「**Pending（保留）**」となるケースも存在します。これは主に、対象資産がスクリプトの対応プラットフォームではない場合や、すでに環境から削除された資産に対してスクリプトが割り当てられている場合に発生します。このような状況を防ぐためには、資産管理の精度を維持する、いわゆるアセットハウスキーピングが重要となります。

さらに、**CAR で作成したカスタム QID は、VMDR と連携して活用することができます**。KnowledgeBase に登録された QID をもとに、Vulnerability 管理機能から対象資産を検索・分析できます。これにより、CAR で検出した独自のリスクを、既存の脆弱性管理プロセスの中で一元的に管理することができます。

また、**これらの結果はレポートとして出力することができる**ほか、ダッシュボードやウィジェットとして可視化することも可能です。特に、ゼロデイ対応や緊急のリスク管理が必要なシナリオでは、継続的な監視を行うためにダッシュボード化することが有効です。

以上のように、「Scripts」セクションおよびスクリプト実行機能は、CAR における実運用の中核を担う要素です。スクリプトの作成からテスト、承認、実行、結果分析、そして可視化までを一貫して管理できるため、効率的かつ高度なセキュリティ運用を実現することが可能となります。本トレーニングでは、これらのプロセスを実際の操作とともに習得していきます。

ユースケース 1　ゼロデイ

本セクションでは、**ゼロデイ脆弱性への対応における CAR の活用方法について**、具体例(**Lab Tutorial: Import and run a zero-day detection script**)を用いて解説します。ここでは、**libcurl の特定バージョンに存在する脆弱性**を題材に、既存のスクリプトを活用した検知プロセスを理解していきます。

まず前提として、ゼロデイ脆弱性とは、ベンダーによる正式なパッチや検知ロジックが十分に提供されていない段階で公開される脆弱性を指します。このような脆弱性は、標準的な QID（Qualys ID）や通常のスキャンのみでは十分に検知できないケースがあります。今回の libcurl の脆弱性もその一例であり、特定のバージョンにおいて、実行中のプログラムに対して不正に Cookie を挿入される可能性があるというリスクが存在します。

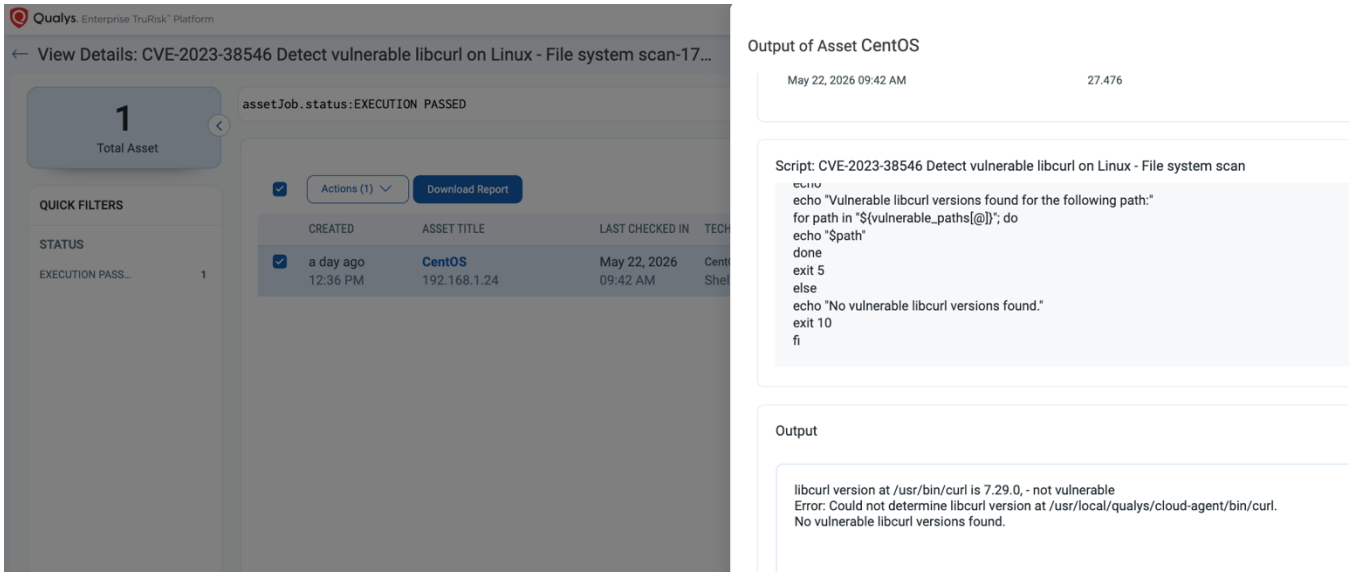
このようなケースでは、対象となるソフトウェアのインストール方法や配置場所、利用状況によって挙動が異なるため、単純なチェックでは正確な検知が難しくなります。そのため、CAR を活用し、より柔軟な検知ロジックを実装することが重要になります。

具体的には、Qualys Library に用意されている事前構築済みのスクリプトを利用することで、ゼロデイ脆弱性に対する検知を迅速に開始することができます。このスクリプトをインポートし、環境内で実行することで、対象となる libcurl の脆弱バージョンを特定することが可能です。

さらに重要な点として、CAR で作成または活用した検知結果は、単なる検出情報として扱われるだけではありません。これらはカスタム QID として定義され、Qualys の脆弱性管理機能と統合されます。すでに資産の重要度（Asset Criticality）が設定されている場合、このカスタム QID も含めた形でリスクスコア（True Risk Score）に反映されます。その結果、単なる検知結果にとどまらず、「**どの資産においてどの程度のリスクがあるか**」というコンテキストを踏まえた評価が可能になります。

操作手順としては、まず Qualys Library から対象スクリプトを確認し、必要に応じてインポートを行います。今回の例ではすでにスクリプトがインポートされている前提となっており、その後「**Scripts**」セクションに移動してスクリプトを確認します。
スクリプトが利用可能な状態であれば、**Quick Actions** メニューから実行することができます。実行後は「**Jobs**」セクションを通じて実行結果を確認します。ここでは、どの資産に対してスクリプトが実行されたか、実行が成功したか、あるいは未実行（Pending）となっているかといった状況を把握することができます。

本演習では、これらの一連の流れを実際に操作しながら確認していきます。具体的には、ライブラリからスクリプトを取得し、スクリプトを実行し、その結果を確認するという基本的なワークフローを体験します。
このプロセスを通じて、受講者の皆様には、ゼロデイ脆弱性のような緊急かつ非標準的なリスクに対して、CAR を用いて迅速に検知・対応する手法を理解していただきます。CAR を活用することで、標準機能だけでは対応が難しい状況に対しても柔軟かつ効率的に対処できるようになります。



以上のように、CAR と Qualys Library を組み合わせることで、ゼロデイ脆弱性への対応を迅速に開始し、かつリスクコンテキストを保ったまま管理することが可能となります。本コースでは、このような実践的な活用方法を段階的に習得していきます。

ユースケース 2 Chrome

今回は、ユーザーがインストールしている**ブラウザー拡張機能**の可視化について説明します。
ブラウザー拡張機能は利便性を高める一方で、**新たな脆弱性を持ち込む可能性があり**、攻撃の入り

口となることが多いのが特徴です。そのため、どの拡張機能がインストールされているのかを把握し、監視できるようにすることが重要です。
しかし、このシナリオでは、拡張機能を確認する仕組みがまだ用意されていない前提とします。そこで、スクリプトを新たに作成・実行し、環境の可視性を高める対応を行います。

スクリプトの作成手順
まず、**スクリプトセクション**に移動し、新しいスクリプトを作成します。
このスクリプトはライブラリにまだ存在しないため、自分たちで作成してデプロイする必要があります。
スクリプト作成時には、以下の項目を設定します。
- スクリプトのタイトル（例：Chrome 拡張機能の検出など）
- 説明文
- 重要度（Severity）

これにより、**新しい QID** を作成し、Qualys のナレッジベースに登録します。この QID は、特定のホストに対するリスクと関連付けられます。

QID の詳細設定
次に、対象となるプラットフォームを選択し、タイトルが正しいことを確認します。
あわせて QDS（Qualys Detection Score） を設定します。QDS は 0～100 の範囲で指定し、数値が高いほど緊急度の高い脆弱性であることを示します。どの数値を設定するかは、組織のポリシーに従って判断してください。
また、以下の設定も行います。
- 潜在的な脆弱性か、確認済みの脆弱性かの指定
- 関連する CVE があれば追加
- 使用するスクリプト言語
- カテゴリ
- タイムアウト値

スクリプトの入力方法は 3 通りありますので、いずれかを選択してスクリプトを貼り付けます。

リターンコードの設定と保存
リターンコードも設定します。
- **10**：結果が空（検出なし）
- **5**：デフォルト以外の拡張機能が検出された場合

設定が完了したら「次へ」をクリックし、作成内容のサマリーを確認します。問題がなければ保存し、これでスクリプトがナレッジベースに追加されます。

結果の確認方法
スクリプトを環境で実行した後は、Vulnerability（脆弱性）セクションに移動します。
これまでと同様に、作成した QID を指定して検索してください。
すると、その QID に該当する結果を返したアセットが表示され、デフォルト以外の Chrome ブラウザー拡張機能がインストールされている端末を確認することができます。

このように、スクリプトを活用することで、これまで見えなかったブラウザー拡張機能の状況を可視化し、セキュリティリスクの把握につなげることができます。

演習名：　Custom QID Creation
演習名：　Run Custom QID

追加のユースケース

ここでは、いくつかの代表的なユースケースを簡単に紹介します。

ハッシュ値を使ったファイル検出のユースケース

ライブラリの中には、ハッシュ値を基にファイルの存在を確認するスクリプトがあります。
この検出では、Linux と Windows の両方に対応したスクリプトが用意されており、詳細画面から内容を確認できます。
このスクリプトは、指定したハッシュ値を持つファイルがシステム上に存在するかを検索します。
さらに応用として、そのハッシュ値がマルウェアであると判明している場合は、検出だけでなく該当ファイルを削除するスクリプトを作成することも可能です。
既存のスクリプトは、そのままインポートすることも、コピーしてカスタマイズすることもできます。

Active Directory セキュリティ管理のユースケース

ライブラリをカテゴリ「AD Security Posture」で絞り込むと、Active Directory のセキュリティ管理を支援するスクリプトが複数表示されます。
その中の一例として、非アクティブアカウント分析（Inactive Account Analyzer）があります。
このスクリプトでは、過去 180 日間使用されていないアカウントを検出し、組織がユーザーアカウントを評価・管理する際に役立ちます。
不要になったアカウントを特定することで、削除などの対応を検討でき、セキュリティ向上につながります。

サービス拒否（DoS）脆弱性の緩和ユースケース

別のユースケースとして、サービス拒否（DoS）脆弱性の緩和があります。
この脆弱性については、ナレッジベースにすでに QID が登録されており、比較的最近の CVE に基づいた検出が可能です。
この脆弱性は、Web アプリケーションスキャンでも検出できますが、CAR（Custom Assessment and Response） を使用することで、緩和スクリプトによる対応が可能です。
ただし、この方法が組織にとって適切かどうかは、十分に調査・検討する必要があります。別の対応方法を選択することも考慮してください。

検出とレスポンスの違いについて

CAR のライブラリ内にある該当スクリプトの詳細を確認すると、
このスクリプトは脆弱性を検出するものではなく、レスポンス（対応）を行うものであることが分かります。具体的には、対象となるプロトコルを無効化することで、サービス拒否脆弱性を回避する仕組みです。このように、ライブラリには検出用だけでなく、脆弱性への対処を目的としたスクリプトも含まれています。

まとめ

このように、ライブラリは常に成長し続ける仕組みであり、さまざまな検出・対応ユースケースが追加されていきます。
定期的に内容を確認し、自社環境に合ったスクリプトを活用することが重要です。

CAR SCRIPT ワークフロー

このセクションでは、これまで詳しく扱ってこなかった CAR スクリプトのワークフローについて説明します。
CAR におけるスクリプトの実行には、いくつかの重要なステップがあります。基本的な流れは次の通りです。

1. スクリプトの作成
2. テストの実施
3. レビューと承認
4. 実行およびスケジュール設定

まず、誰かがスクリプトを作成し、Qualys に登録するか、ライブラリからインポートします。
その後、そのスクリプトは**最大 10 台までのアセットに対してテスト**を行います。 テストが完了した後、スクリプトはレビューおよび承認のプロセスに進みます。この承認が完了して初めて、スクリプトを本番環境で実行し、スケジュール設定することができます。

ロールベースアクセス制御（RBAC）の重要性

CAR ではセキュリティを強化するために、ロールベースアクセス制御（RBAC）が組み込まれています。
スクリプトはシステムに対して大きな影響を与える可能性があるため、**役割ごとに権限を分離することでリスクを低減**します。

主なロールは以下の通りです。

- CAR Author：スクリプトの作成およびテストを実施
- CAR Manager：スクリプトのレビューおよび承認を担当

- **Operations ロール**：承認されたスクリプトの実行を担当
- **Viewer / Auditor**：閲覧や監査目的で利用

例えば、Operations ロールはスクリプトの実行はできますが、承認はできないように制御されています。
このように職務分離（Separation of Duties）が実現されています。

スクリプトのテストと承認

スクリプト作成後、最初に行うべきは**テスト**です。
テストの目的は、スクリプトが正しく動作するか、またシステムに悪影響を与えないかを確認することです。
テスト時のポイントは以下の通りです。
- 最大 10 台までのホストを対象にテストする
- タグではなく個別にホストを選択する

これは、初回テストで影響範囲を最小限に抑えるためです。

その後、CAR Manager によるレビューと承認が行われます。なお、テストやレビューの過程では、危険性のあるコマンドがハイライト表示されることがありますが、システム側で実行を自動的に停止するわけではありません。そのため、組織として十分にスクリプト内容を確認し、慎重に承認することが重要です。

アクティビティログによる監査

CAR の操作はすべてアクティビティログに記録されます。
ログでは以下の情報を確認できます。
- 実行されたアクション（作成、テスト、実行、更新など）
- 対象（スクリプトまたはジョブ）
- 操作を行ったユーザー

また、ログはフィルタリングや検索、ダウンロードが可能です。これにより、運用状況を常に把握し、監査にも対応できます。

スクリプトとジョブの保持期間に関する注意点

最後に、運用上の重要なポイントです。
- スクリプトは 6 か月間更新または実行されない場合、自動的に無効化（Deprecated）されます
- 無効化されたスクリプトは、7 日後に削除（Purge）されます
- ジョブも同様に、7 日後に削除されます

そのため、多くの組織では週 1 回程度の定期実行を設定し、スクリプトやジョブを維持する運用を行っています。
また、必要に応じて API を使用して結果を取得・保存することも可能です。

出力およびスクリプトサイズの制限

CAR には以下の制限があります。
- スクリプト出力サイズ：最大 1MB
- スクリプトサイズ：最大 500KB

これらを考慮してスクリプトを設計してください。

まとめ
CAR のスクリプト運用では、
作成 → テスト → 承認 → 実行
という明確なプロセスを踏むことが重要です。
また、RBAC による権限分離やログ管理を活用しながら、安全かつ確実に運用していくことが求められます。

トレーニング内の演習を実施すると、より理解が深まります。

演習名： Create a New Script
演習名： Testing a Script
演習名： Review and Approve a Script
演習名： Schedule and Run Script

ポリシーの適用

ここでは、Collection & Analysis（CAR）のその他の活用方法について説明します。

機密情報（PII）の検出
まず一つ目のユースケースとして、PII（個人識別情報）が平文で保存されているホストの検出があります。 CAR を使用すると、独自のスクリプトを作成し、システム内に機密情報が保存されていないかを調査することができます。
例えば、以下のような対応が可能です。

- システム内に平文の PII が含まれているかをスキャンする
- 検出された場合にレポートする
- 必要に応じて、データの削除や対象システムの保護を実施する

このように、標準機能ではカバーできない内容でも、スクリプトを活用することで可視化・対処が可能になります。

修復およびセキュリティ強化の実施

CAR スクリプトは、単なる検出だけでなく、修復（リメディエーション）にも活用できます。
これまで説明した通り、CAR では以下のような対応が可能です。

- 脆弱性の緩和（Mitigation）
- 設定ミス（Misconfiguration）の是正
- システムのハードニング（セキュリティ強化）の強制

つまり、問題を見つけるだけでなく、自動的に修正・対策を実行することもできる点が大きな特徴です。

Policy Compliance との連携

Qualys の大きな強みの一つとして、CAR と Policy Audit の連携があります。
この連携では、「スクリプト結果チェック（Script Result Check）」という機能を利用します。

スクリプト結果チェックの仕組み

スクリプト結果チェックでは、CAR スクリプトの実行結果をもとに、ポリシーに準拠しているかどうかを判定できます。
基本的な流れは以下の通りです。

1. CAR スクリプトを作成し、対象ホストで実行する
2. 必要なデータを収集する
3. Policy Compliance でカスタムコントロールを作成する
4. スクリプトの出力結果を評価し、合否を判定する

具体例（PII の検出）

例えば、PII の検出を行った場合、スクリプトの出力に特定の情報（例：API キーや個人情報）が含まれていることがあります。
この場合、Policy Compliance 側で以下のような判定が可能です。

- PII が含まれている → コントロール失敗（Fail）
- 含まれていない → コントロール成功（Pass）

このようにして、収集したデータをもとにコンプライアンス状況を自動的に評価できます。

対応プラットフォーム

スクリプト結果チェックは、以下の環境で利用可能です。

- Unix / Linux ホスト
- Windows ホスト

どちらの環境でも、スクリプトの出力結果を読み取り、
コンプライアンス判定に活用することができます。

まとめ

CAR を活用することで、以下のような高度な運用が可能になります。

- 平文の機密情報（PII）の検出
- 脆弱性や設定不備の自動修復
- システムのセキュリティ強化（ハードニング）
- Policy Audit と連携したコンプライアンス評価

単なる検出にとどまらず、「収集 → 分析 → 評価 → 対応」まで一貫して実行できる点が、CAR の大きな利点です。

トレーニング内の演習を実施すると、より理解が深まります。

演習名： PII Discovery Script
演習名： Script Results Check UDC

API の利用

CAR API を活用したスケーリングについて

ここでは、CAR（Custom Assessment and Response）をより効率的に活用するための方法として、**CAR API の利用**について説明します。

CAR には API が用意されており、Qualys のユーザーインターフェース（UI）から **API ガイドを参照することができます**。
API ガイドは、画面右上の「Help（ヘルプ）」メニューから「Resources」を選択し、CAR アプリケーションの **API User Guide** をクリックすることで確認できます。

CAR API でできること

CAR API を使用すると、これまで画面操作で行っていたさまざまな処理を自動化・効率化することができます。
主な機能は以下の通りです。

1. スクリプト関連の操作

- スクリプト数の取得（カウント）
- スクリプトの検索・照合

- スクリプト一覧の取得

これにより、多数のスクリプトを管理・把握しやすくなります。

2. アセット情報の取得

- スクリプトを実行する対象となる**アセットの一覧取得**

対象環境の情報を API 経由で取得できるため、柔軟な運用が可能です。

3. ジョブ関連の操作

- スクリプト実行ジョブの確認
- スクリプト ID を基にしたジョブ情報の取得

これにより、どのスクリプトがどのように実行されているかを把握できます。

4. アクティビティログの取得

- API を用いた**アクティビティログの取得**

誰がどの操作を行ったかをプログラムから取得でき、監査や分析に役立ちます。

5. スクリプトのエクスポート／インポート

- スクリプトを API 経由でエクスポート・インポート

複数環境間でのスクリプト共有やバックアップが容易になります。

6.スケジューリング

- スクリプトの実行スケジュール設定

定期実行を自動化し、運用負担を軽減できます。

7.ライブラリのスクリプト取得

- ライブラリに登録されているスクリプト一覧の取得

最新のスクリプト活用にもつながります。

以上

まとめ

CAR API を活用することで、以下のようなメリットがあります。

- 手動操作の削減による**運用効率の向上**
- スクリプトやジョブの**一元管理**
- 自動化による**スケーラブルな運用**

API の詳細な使い方や具体的なリクエスト方法については、
Qualys UI から参照できる **API User Guide** にすべて記載されていますので、必ず確認してください。