技術資料

Contrast Assess セッションベースの脆弱性自動検証機能紹介

Contrast Security Japan 合同会社
jpnsales@contrastsecurity.com
https://www.contrastsecurity.com/jp

Contrast Assess 自動検証ポリシーについて

自動検証 (Auto-verification) ポリシーとは

- 脆弱性管理の支援機能の一つで、下記のような場合に**自動的に**脆弱性のステータスを**「修復済 - 自動検証」**に更新する(= **修正されたとみなす**)機能
  - 以前の検出内容と同じルート(プログラムの実行経路[コントローラメソッド])を再度検査した際に、その脆弱性が見つからなくなった場合

    (「ルートをもとにした自動検証」または「[NEW]セッションを元にした自動検証」※Contrast推奨)
  - 検出されてから指定した日数が経過した場合 (「日数による自動検証」※Contrast非推奨)

メリット/目的

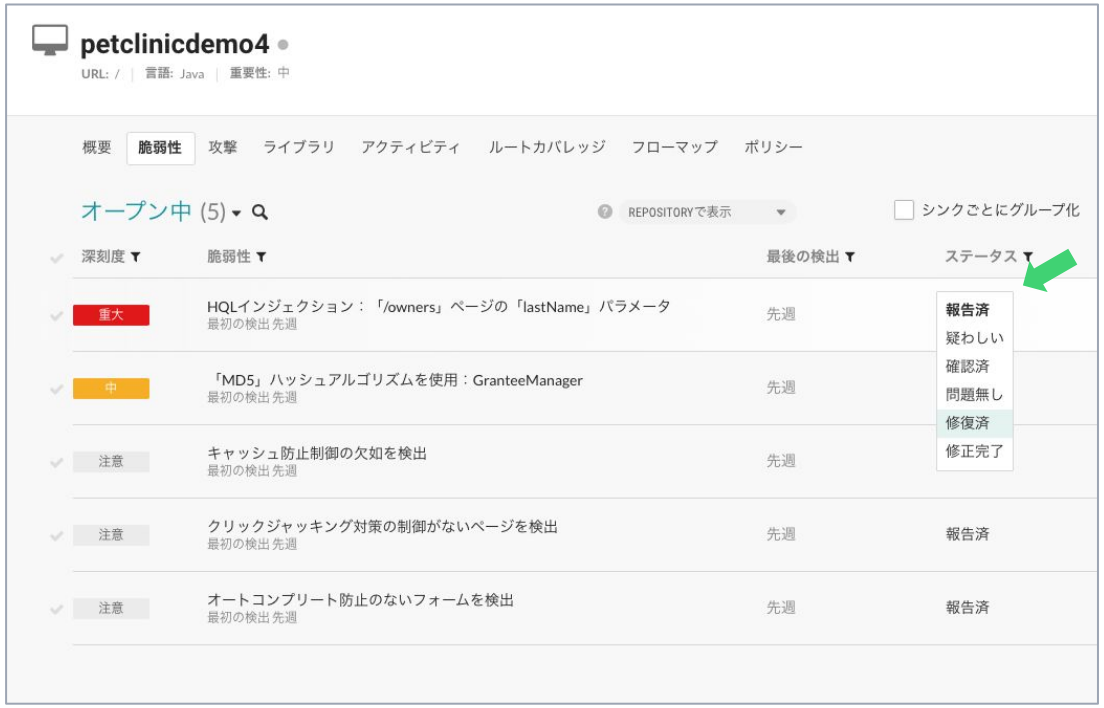
- 新たな検査による検出内容を残し、以前の脆弱性の表示を自動抑制することで、管理上の「ノイズ(ヒューマンエラー含む)」を減らし、管理効率を上げることができる。
- 特に、変更が頻繁に施されるプログラムで継続的にAssessによるテストを行う場合、古い検出内容が有益でないことも多く、新たな結果を優先表示する仕組みが有用となる。

Contrast SECURITY

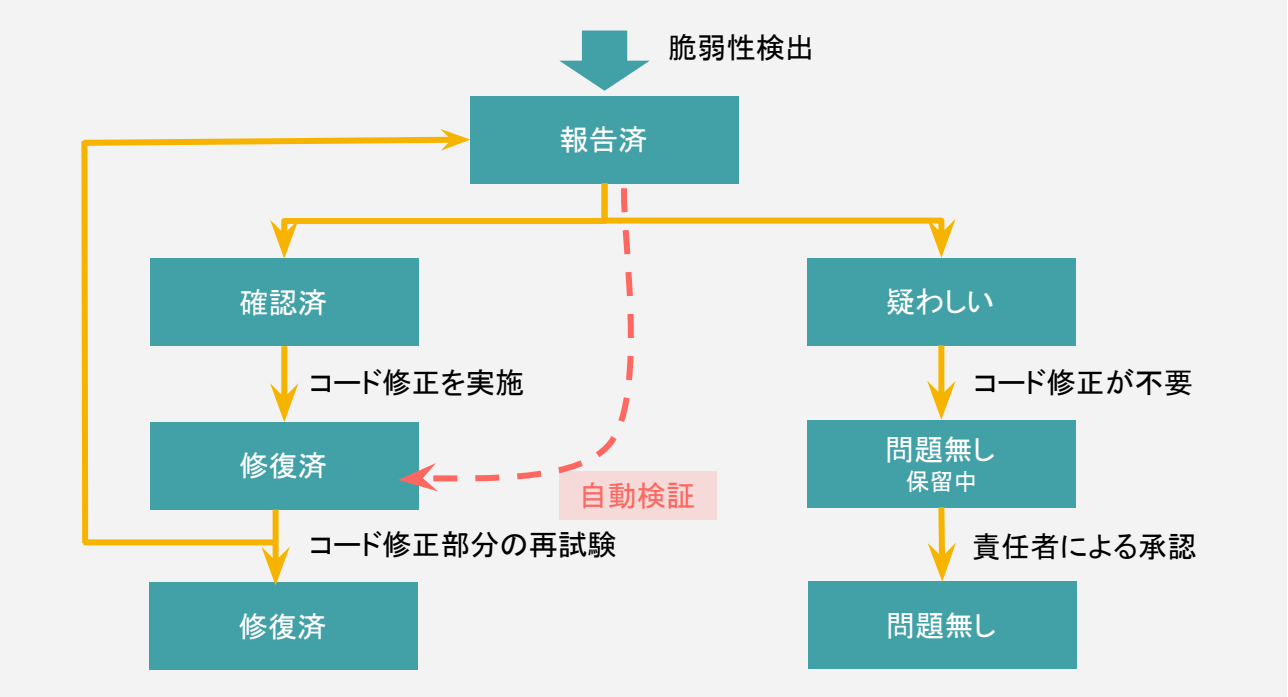
「ステータス」列を使った脆弱性の管理

- Assessによって検出された直後は「報告済」のステータス
- 修正した脆弱性は、「修復済」に更新してクローズ
- 自動検証ポリシーは、古い脆弱性を「修復済- 自動検証」ステータスに自動更新

手動でのステータス更新



Assessでの脆弱性ライフサイクル（典型的な例）



Contrast SECURITY

自動検証ポリシーの設定

ポリシーの管理> 脆弱性の管理> ＋ポリシーを追加

Contrast SECURITY

自動検証ポリシーの設定(続き)

ポリシーの管理画面

適用範囲（ルール、アプリケーション、環境）

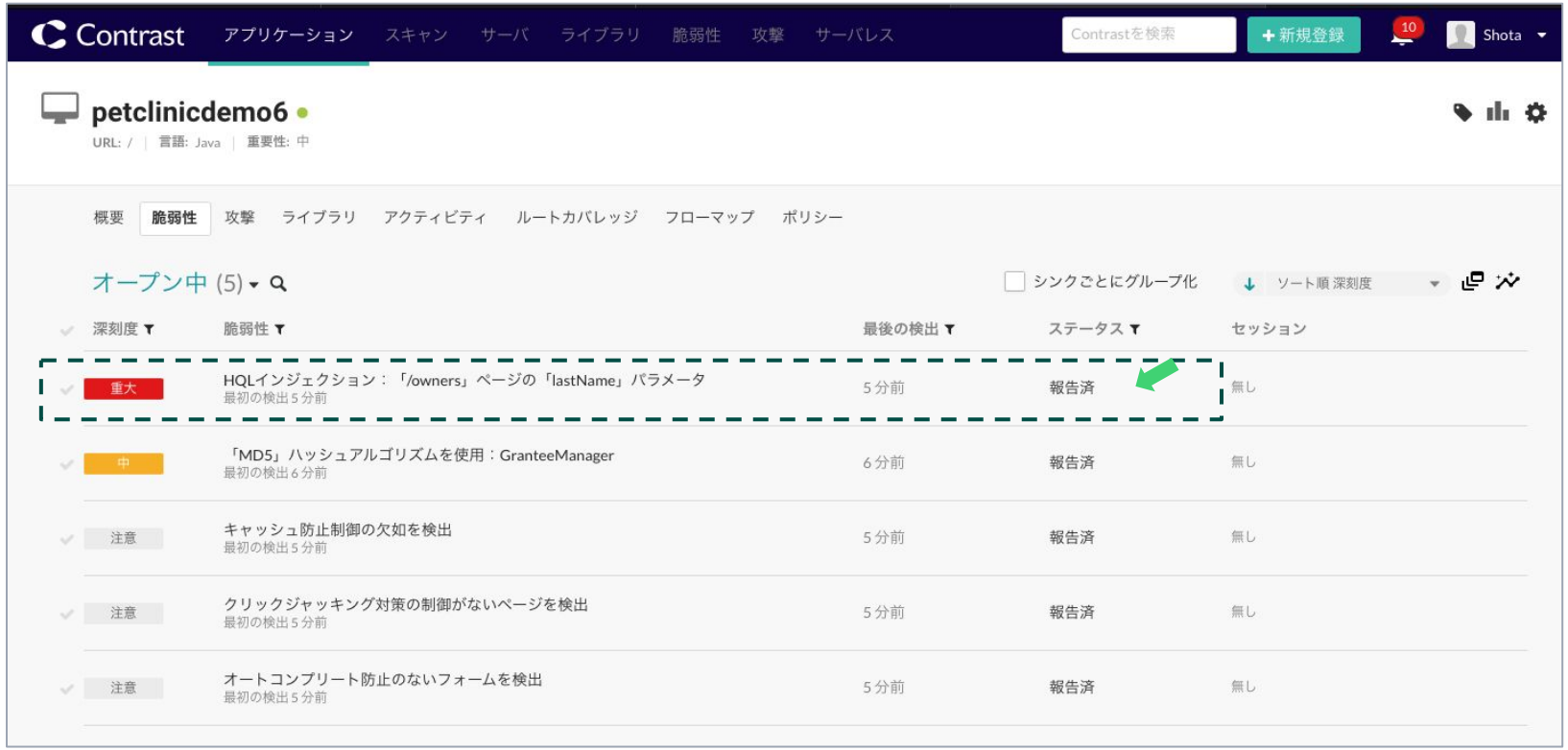
適用するタイミング（トリガー）の種類
- 検出からの**経過日数**により適用される（「日数ベース」）
- **同じルートで検出されなくなった際に** 適用される（「ルートベース」）
- **同じルートで検出されなくなった際＋セッションクローズ時に** 適用される（「セッションベース」）

ルートベースの自動検証機能による ステータス変更のウォークスルー

前提: ルートベースでの自動検証ポリシーを追加

自動検証機能によるステータス変更のウォークスルー (① 検出当初)

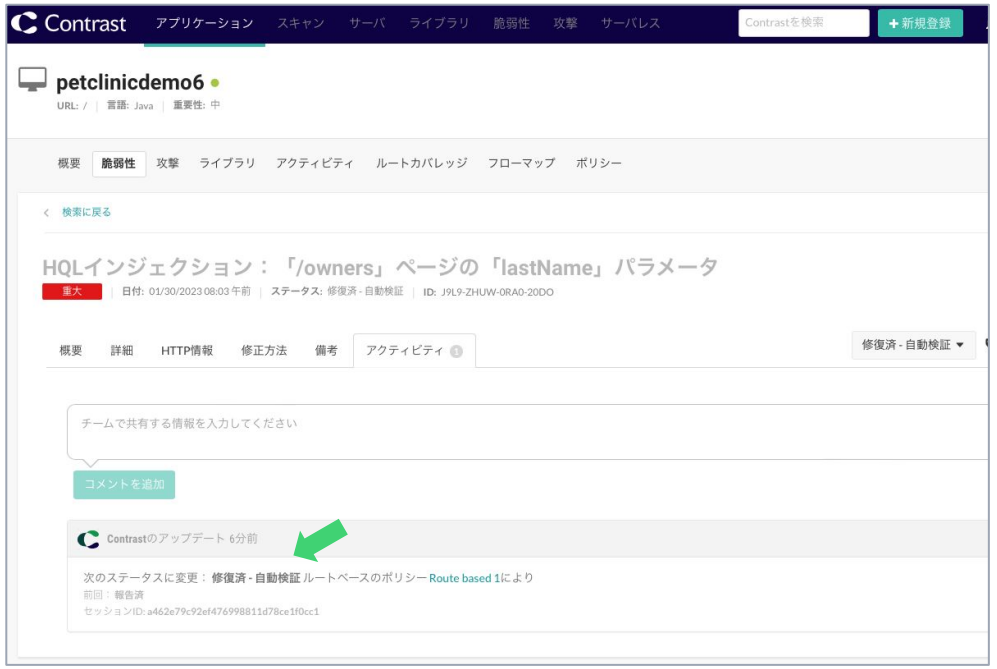
- 脆弱性が最初に検出される際、ステータスは「報告済」に。



Contrast SECURITY

自動検証機能によるステータス変更のウォークスルー (② 脆弱性を修正し、再検査)

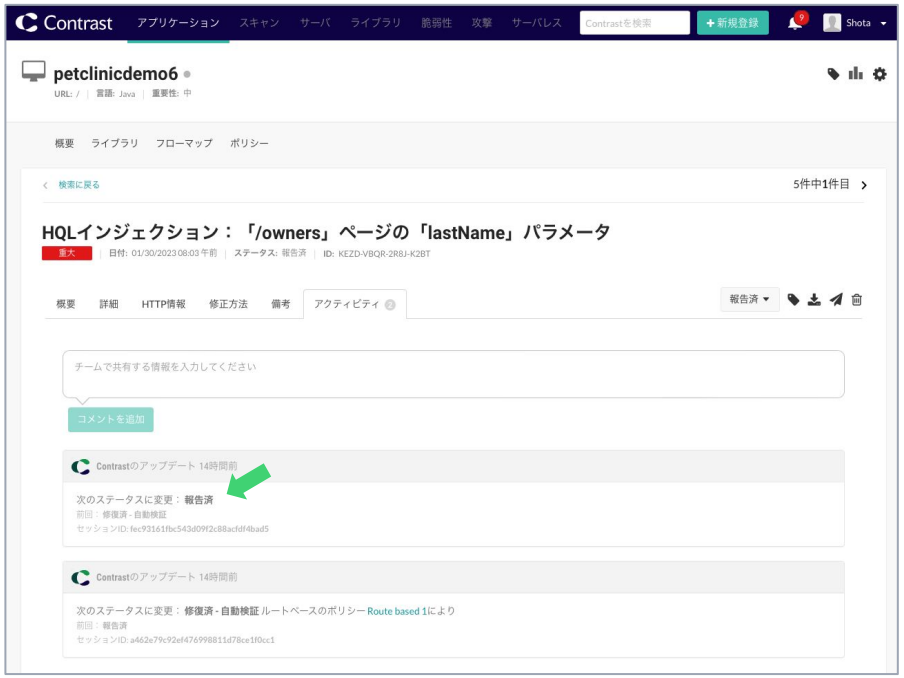
- 脆弱なコードを修正後、再度Assessによる検査を実施。
- 同じ実行ルートに以前検出された脆弱性が検出されない。
  → 自動検証ポリシーが働き、ステータスが「修復済- 自動更新」に。



Contrast SECURITY

自動検証機能によるステータス変更のウォークスルー (③ さらに再検査、同じ脆弱性が再度検出)

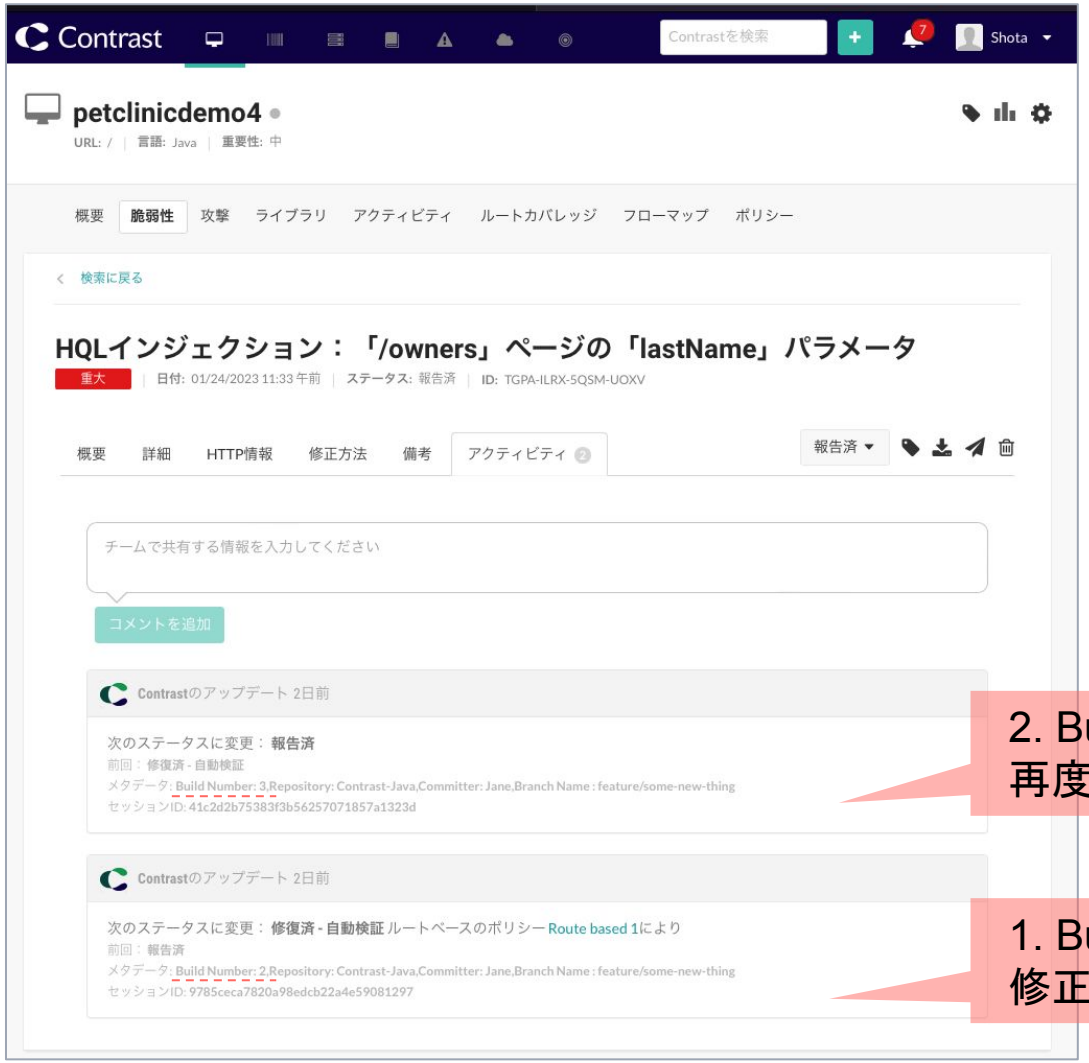
- 同じ脆弱性が再度検出。ステータスは再び「報告済」に。



Contrast SECURITY

セッションメタデータの併用

セッションメタデータ機能を併用することで、より明確なステータス追跡が可能



セッションメタデータ機能… Assessエージェントを使った検査セッションの付加情報（例: ビルド番号）を、アプリケーション起動オプションとして指定することで、脆弱性の管理をしやすくする機能

自動検証ポリシーと併用すると、ステータス変更のタイムライン表示とあわせ、検査セッションのメタデータが表示されることで、どのタイミングでステータスが変更されたか、把握がより容易に。

2. Build Number = 3
再度検出され、再び「報告済」に戻る

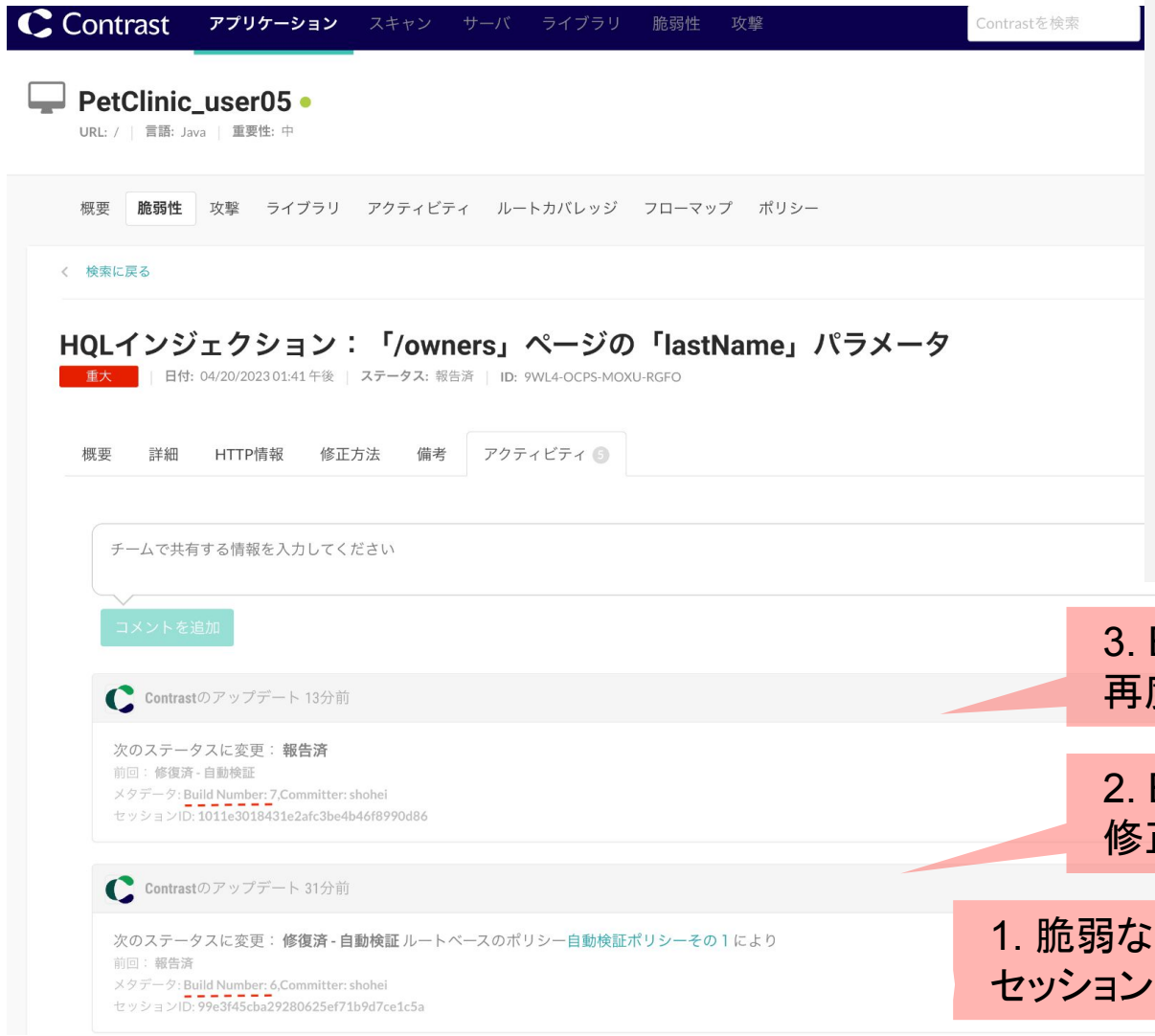
1. Build Number = 2
修正による自動検証。ステータスは「修復済自動検証」に

Contrast SECURITY

セッションベースの自動検証機能による ステータス変更のウォークスルー

前提: セッションベースでの自動検証ポリシーを追加

セッションベースの自動検証機能を使用することで、管理者がセッションをクローズしたタイミングでステータス追跡が可能



セッションベースの自動検証機能… セッションメタデータ機能を盛り込む必要がある。前述の「ルートベースの自動検証機能＋セッションメタデータの併用」と機能は同じだが、脆弱性を自動検証するタイミングが異なる。

ルートベースの自動検証機能では脆弱性が検出されなくなったタイミングで修正による自動検証が発火されるが、セッションベースでの自動検証ではAPIコールよりセッションをクローズしたタイミングで自動検証が発火される。

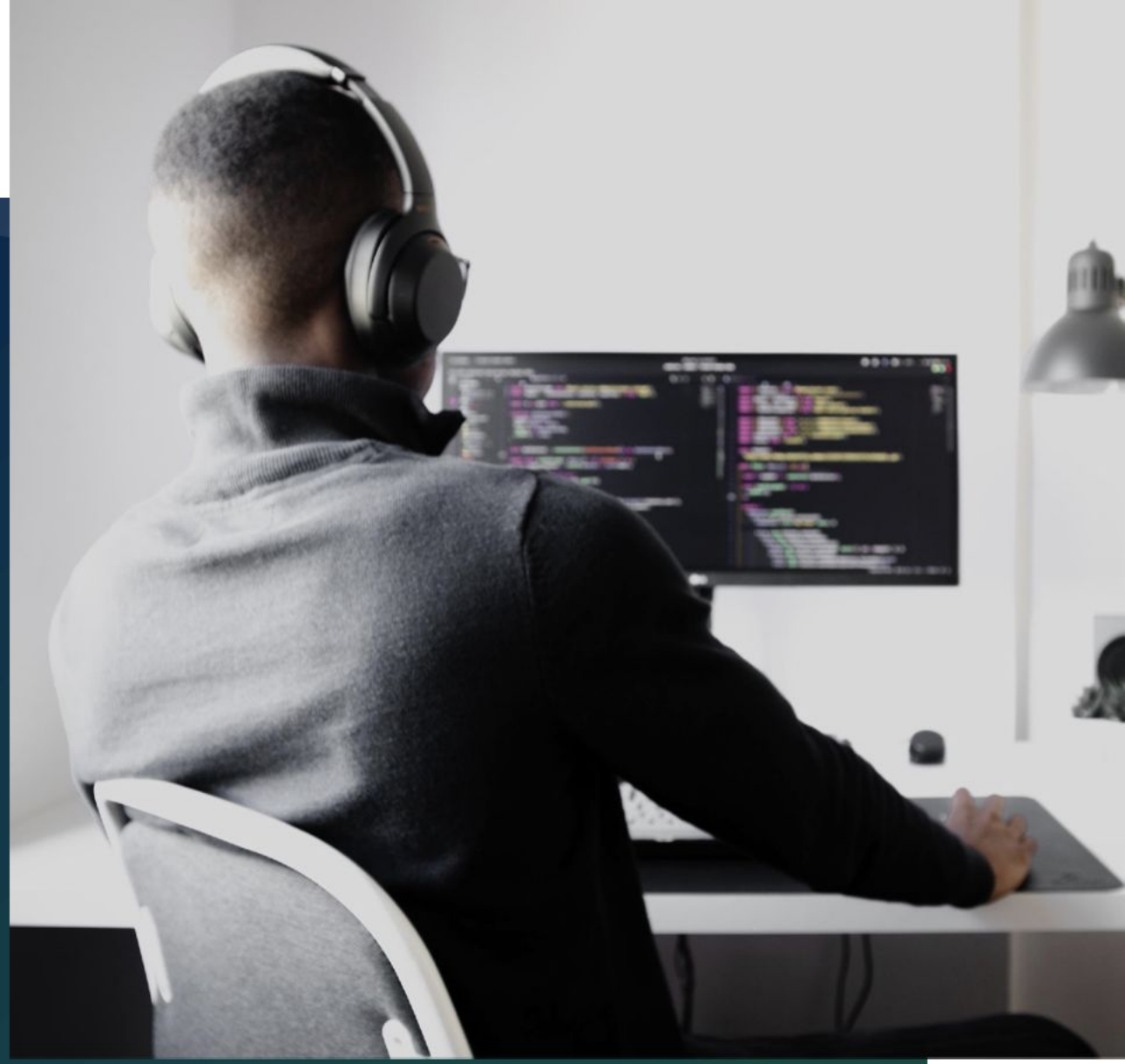
従って管理者がメタデータを追加（オープン）、テスト後にクローズすることで自動検証の発火タイミングをマネージできる。

3. Build Number = 7
再度検出され、再び「報告済」に戻る

2. Build Number = 6
修正による自動検証。ステータスは「修復済自動検証」に

1. 脆弱なコードを修正後、再度Assessによる検査を実施。Build Number = 6のセッションをターミナルよりAPIコールし、セッションをクローズ。

Demo



まとめ

- **「ルートベース」の自動検証ポリシーに加えて、「セッションベース」の自動検証ポリシーが追加された。**
- 「セッションベース」の自動検証ポリシーはセッションメタデータを追加する必要がある。
- 「セッションベース」の自動検証ポリシーは「「ルートベース」の自動検証ポリシーとセッションメタデータとの併用」機能を含んでいるが、自動検証の発火タイミングが異なる。
- 管理者が「セッションベース」の自動検証ポリシーを使用、メタデータを追加（オープン）、テスト後にクローズすることで自動検証の発火タイミングをマネージできる。

注意事項

よくあるお問い合わせ

自動検証ポリシーの設定が見当たらない

- デフォルトでは無効となっているので、Super Adminが組織の設定を変更する必要があります。

期待通りに脆弱性のステータスが変更されない

- アプリケーションコードを更新して再度検査を行う際に、セッションメタデータもそれに合わせて更新してください。

脆弱性の「アクティビティ」セクションを見ると「報告済」 -> 「修復済 - 自動検証」 -> 「報告済」 -> ... と頻繁にステータスが 変更されている。

- ルート（コントローラメソッド）に分岐がある場合、どの分岐を先に検査するかによって、修正が行われていないにも関わらず一旦「修復済 - 自動検証」となることがあります。
  このようなケースでは、テストの際にコントローラー内の全ての分岐を検査しないと、実態と異なるステータスとなってしまうので注意が必要です。

Contrast SECURITY

参考

参考資料

- Contrast Security サポートポータル

  https://support.contrastsecurity.com/hc/ja

  > 使い方 > Contrast UI > 検出された脆弱性が修復済みであるかを確認する

- Contrast Security ドキュメント

  https://docs.contrastsecurity.jp/index.html?lang=ja

  “自動検証”で検索 > セッションベースの自動検証のためのテストランの設定

  https://docs.contrastsecurity.jp/ja/session-metadata.html

  セッションメタデータ及び セッションメタデータの設定

Contrast SECURITY

Contrast Security Japan合同会社
jpnsales@contrastsecurity.com
https://www.contrastsecurity.com/jp

技術資料

Contrast SECURITY