

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。

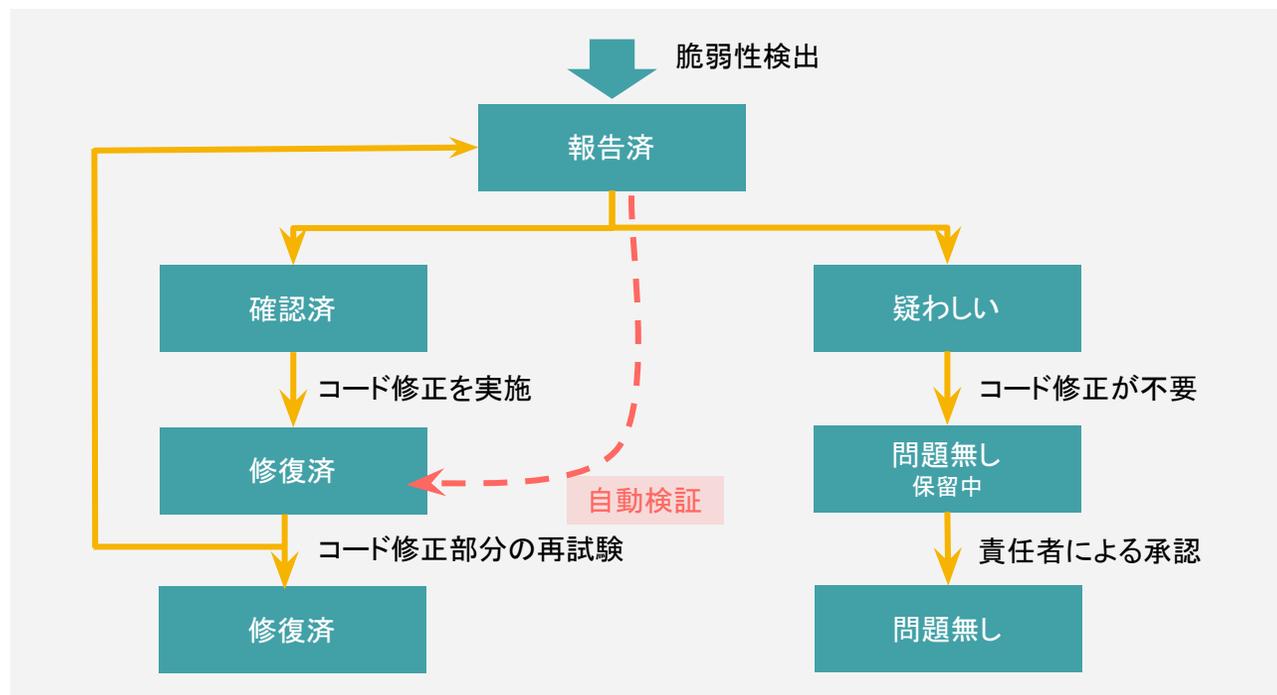
「ステータス」列を使った脆弱性の管理

- 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

The screenshot shows the petclinicdemo4 application security dashboard. The '脆弱性' (Vulnerabilities) tab is active, displaying a list of 5 open vulnerabilities. A green arrow points to the 'ステータス' (Status) column header, which has a dropdown menu open. The dropdown menu options are: 報告済 (Reported), 疑わしい (Suspicious), 確認済 (Confirmed), 問題無し (No issue), 修復済 (Fixed), and 修正完了 (Fixed). The first vulnerability, 'HQLインジェクション: 「owners」ページの「lastName」パラメータ', is currently in the '報告済' status.

Assessでの脆弱性ライフサイクル(典型的な例)



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 アプリケーション スキャン サーバ ライブラリ 脆弱性 攻撃

Contrastを検索 + 新規登録 Shohei

ポリシーの管理

ASSESS

Assessルール

セキュリティ制御

脆弱性の管理

PROTECT

Protectルール

CVEシールド

仮想パッチ

ログエンハンサー

IP管理

概要

アプリケーションの例外

コンプライアンスポリシー

ライブラリポリシー

機密データ

脆弱性の動作

承認ワークフロー

脆弱性のクローズ時に管理者の承認が必要

問題無しステータスのオプション

これにより、脆弱性が「問題無し」の場合の「上記以外」の理由が置き換わります。ユーザに表示・使用させたいラベルを設定してください。

上記以外の理由のカスタムラベルを設定

保存

脆弱性ポリシー

自動検証 違反

+ ポリシーを追加

ポリシーはまだ作成されていません

自動検証ポリシーの設定(続き)

ポリシーの管理画面

ポリシーの管理

ASSESS

Assessルール

セキュリティ制御

脆弱性の管理

PROTECT

Protectルール

CVEシールド

仮想パッチ

ログエンハンサー

IP管理

概要

アプリケーションの例外

コンプライアンスポリシー

ライブラリポリシー

機密データ

脆弱性の自動検証ポリシーを追加

このポリシーは、脆弱性自体の再検出を防ぐものではありません

名前*

XXX脆弱性 (自動検証)

脆弱性ルール

全てのルール (83)

アプリケーション

全てのアプリケーション

環境

全ての環境 (3)

トリガー

少なくとも1つのトリガータイプを選択する必要があります。両方のトリガータイプを選択した場合、先に要件を満たした方がポリシーのトリガーとなります。

トリガータイプを1つ選択する必要があります。

指定日数以降に全ての脆弱性を自動検証によって修復済のステータスにする: 0 日

ルートまたはセッションに基づく自動検証:

セッションベースの自動検証 ? 推奨

ルートベースの自動検証 ?

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」)

キャンセル

保存

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

脆弱性ポリシー						
自動検証		違反				
ポリシーを検索						+ ポリシーを追加
ポリシー	脆弱性ルール	アプリケーション	説明	環境	有効	
自動検証ポリシーその3	全て	全て	ルートベースの自動検証	全て	<input checked="" type="checkbox"/>	

自動検証機能によるステータス変更のワークスルー (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。

The screenshot shows the Contrast Security interface for a target named 'petclinicdemo6'. The top navigation bar includes 'アプリケーション', 'スキャン', 'サーバ', 'ライブラリ', '脆弱性', '攻撃', and 'サーバレス'. A search bar contains 'Contrastを検索' and a '+ 新規登録' button. The main content area shows a list of vulnerabilities under the '脆弱性' tab. The first vulnerability is highlighted with a dashed green box and a green arrow pointing to its '報告済' status.

✓ 深刻度 ▼	脆弱性 ▼	最後の検出 ▼	ステータス ▼	セッション
✓ 重大	HQLインジェクション: 「owners」ページの「lastName」パラメータ 最初の検出 5 分前	5 分前	報告済	無し
✓ 中	「MD5」ハッシュアルゴリズムを使用: GranteeManager 最初の検出 6 分前	6 分前	報告済	無し
✓ 注意	キャッシュ防止制御の欠如を検出 最初の検出 5 分前	5 分前	報告済	無し
✓ 注意	クリックジャッキング対策の制御がないページを検出 最初の検出 5 分前	5 分前	報告済	無し
✓ 注意	オートコンプリート防止のないフォームを検出 最初の検出 5 分前	5 分前	報告済	無し

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。

修復済になるとフィルタにより当該脆弱性が出て来なくなる場合がありますので、「全て」などに切り替えて表示させます。

深読度	脆弱性	最後の検出	ステータス
重大	HQLインジェクション：「/owners」ページの「lastName」パラメータ 最初の検出 26分前	20分前	修復済-自動検証
中	「MD5」ハッシュアルゴリズムを使用：GranteeManager 最初の検出 27分前	8分前	報告済
注意	キャッシュ防止制御の欠如を検出 最初の検出 26分前	5分前	報告済
注意	クリックジャッキング対策の制御がないページを検出 最初の検出 26分前	5分前	報告済
注意	オートコンプリート防止のないフォームを検出 最初の検出 26分前	7分前	報告済

HQLインジェクション：「/owners」ページの「lastName」パラメータ
重大 | 日付: 01/30/2023 08:03 午前 | ステータス: 修復済-自動検証 | ID: J9L9-ZHUW-ORAO-2000

修復済-自動検証

チームで共有する情報を入力してください
コメントを追加

Contrastのアップデート 6分前

次のステータスに変更: 修復済-自動検証 ルートベースのポリシー Route based 1により
前回: 報告済
セッションID: a462e79c92ef476998811d78ce1f0cc1

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。

The screenshot shows the Contrast dashboard for 'petclinicdemo6'. The '脆弱性' (Vulnerabilities) tab is active. A table lists vulnerabilities with columns for severity, description, last detected time, and status. The first row is highlighted with a red dashed box and a green arrow pointing to the '報告済' (Reported) status.

深刻度	脆弱性	最後の検出	ステータス
重大	HQLインジェクション: /owners ページの 'lastName' パラメータ 最初の検出 1 時間前	1 分前	報告済
中	'MD5' ハッシュアルゴリズムを使用: GranteeManager 最初の検出 1 時間前	2 分前	報告済
注意	キャッシュ防止制御の欠如を検出 最初の検出 1 時間前	1 分前	報告済
注意	クリックジャッキング対策の制御がないページを検出 最初の検出 1 時間前	1 分前	報告済
注意	オートコンプリート防止のないフォームを検出 最初の検出 1 時間前	1 分前	報告済

The screenshot shows the details of a vulnerability: 'HQLインジェクション: /owners ページの "lastName" パラメータ'. The status is '報告済' (Reported). A message indicates a status change: '次のステータスに変更: 報告済' (Next status change: Reported). A green arrow points to this message.

次のステータスに変更: 報告済
前記: 修復済・自動検証
セッションID: fec93161fbc543d092c88acdf4b0ad5

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

Contrast

petclinicdemo4

URL: / | 言語: Java | 重要性: 中

脆弱性

HQLインジェクション: 「/owners」ページの「lastName」パラメータ

重大 | 日付: 01/24/2023 11:33 午前 | ステータス: 報告済 | ID: TGPA-ILRX-SQSM-UOXV

概要 詳細 HTTP情報 修正方法 備考 アクティビティ

チームで共有する情報を入力してください

コメントを追加

Contrastのアップデート 2日前

次のステータスに変更: 報告済
前回: 修復済 - 自動検証
メタデータ: Build Number: 3, Repository: Contrast-Java, Committer: Jane, Branch Name: feature/some-new-thing
セッションID: 41c2d2b75383f3b56257071857a1323d

Contrastのアップデート 2日前

次のステータスに変更: 修復済 - 自動検証 ルートベースのポリシー Route based 1により
前回: 報告済
メタデータ: Build Number: 2, Repository: Contrast-Java, Committer: Jane, Branch Name: feature/some-new-thing
セッションID: 9785ceca7820a98edcb22a4e59081297

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

セッションベースの自動検証機能による ステータス変更のワークスルー

前提: セッションベースでの自動検証ポリシーを追加

脆弱性ポリシー						
自動検証		違反				
ポリシーを検索						+ ポリシーを追加
ポリシー	脆弱性ルール	アプリケーション	説明	環境	有効	
自動検証ポリシーその1	全て	全て	セッションベースの自動検証	全て	<input checked="" type="checkbox"/>	🗑️

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

The screenshot shows the Contrast web application interface. At the top, there is a navigation bar with the Contrast logo and menu items: アプリケーション, スキャン, サーバ, ライブラリ, 脆弱性, 攻撃. Below the navigation bar, the user 'PetClinic_user05' is logged in. The main content area shows a vulnerability report for 'HQLインジェクション: 「/owners」 ページの 「lastName」 パラメータ'. The report is marked as '重大' (Critical) and was discovered on 04/20/2023 at 01:41 PM. The status is '報告済' (Reported). The report details are shown in a tabbed view, with the 'アクティビティ' (Activity) tab selected. Below the report, there is a section for 'チームで共有する情報を入力してください' (Enter information to share with your team) and a 'コメントを追加' (Add comment) button. The bottom of the screenshot shows two update notifications from Contrast, one from 13 minutes ago and one from 31 minutes ago, both indicating a status change to '報告済' (Reported) and providing metadata such as Build Number and Session ID.

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になってしまうので注意が必要です。

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 Japan 合同会社
jpnsales@contrastsecurity.com
<https://www.contrastsecurity.com/jp>