sysdig

イメージスキャン

本文の内容は、Sysdig Secure キャプチャーのドキュメント (https://docs.sysdig.com/en/captures-122718.html）を元に日本語に翻訳・再構成した内容となっております。

イメージスキャン

イメージスキャンを使用すると、コンテナイメージの脆弱性、シークレット、ライセンス違反などをスキャンできます。 開発ビルドプロセスの一部として使用でき、コンテナレジストリに追加されたイメージを検証でき、インフラストラクチャでコンテナを実行することで使用されるイメージをスキャンできます。

前提条件

- ネットワークとポートの要件
  イメージスキャンには、外部の脆弱性フィードへのアクセスが必要です。 最新の定義に適切にアクセスするには、ネットワークとポートの要件を参照してください。
- イメージスキャンリクエストのホワイトリストに登録されたIP
  イメージスキャンリクエストとSplunkイベント転送は両方とも18.209.200.129から発生します。Sysdigがプライベートリポジトリをスキャンできるようにするには、ファイアウォールでこのIPアドレスからのインバウンドリクエストを許可する必要があります。

Sysdigイメージスキャンの仕組み

イメージスキャンの基本設定は簡単です。イメージが保存されているレジストリ情報を提供し、スキャンをトリガーして、結果を確認します。

動作のステップ：

- イメージの内容が分析されます。
- コンテンツレポートは、複数の脆弱性データベースに対して評価されます。
- 次に、デフォルトまたはユーザー定義のポリシーと比較されます。

- 結果は、Sysdig Secureと（該当する場合）開発者の外部CIツールの両方で報告されます。

イメージコンテンツレポート

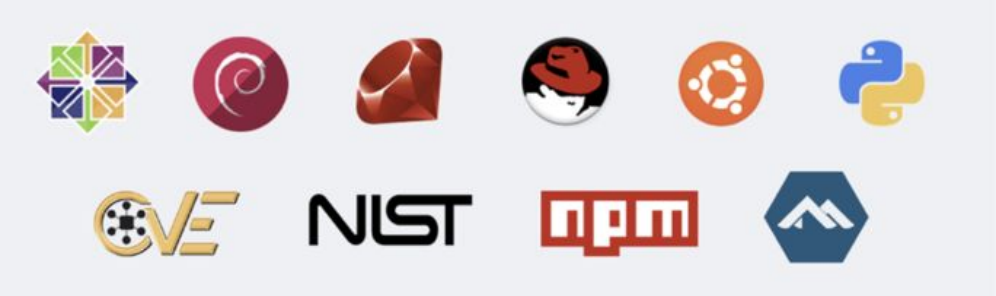
分析により、以下を含むイメージコンテンツの詳細レポートが生成されます。

- 公式OSパッケージ
- 非公式OSパッケージ
- 構成ファイル
- 資格情報ファイル
- ローカリゼーションモジュールとソフトウェア固有のインストーラー：
    - NPMを使用したJavascript
    - Python PiP
    - GEMを使用したRuby
    - .jarアーカイブを使用したJava/JVM
- イメージのメタデータと構成属性

使用される脆弱性データベース

Sysdig Secureは、さまざまな脆弱性データベースに対して継続的にチェックし、新しく検出されたCVEでランタイムスキャン結果を更新します。

現在のデータベースリストには次のものが含まれます。



Centos Debian Ruby Red Hat Ubuntu Python
CVE NIST NPM Alpine NVD

ユースケース

組織として、環境内で実行される許容可能な、安全で信頼できるイメージを定義します。開発パイプラインのイメージスキャンは、セキュリティ担当者とは多少異なるフローに従います。

コンテナ開発中のスキャン（DevOps）

開発パイプラインの一部としてイメージスキャンを使用して、ベストプラクティス、脆弱性、およびシークレットコンテンツをチェックします。

始めに：

- **レジストリの追加**：イメージにアクセスするために必要な資格情報とともに、イメージが保存されているレジストリを追加します。
- **CIツールの統合**：Jenkinsプラグインを使用するか、SysdigLabsソリューションから独自の統合を構築して、イメージスキャンを外部CIツールと統合します。
- **イメージのスキャン**：プラグインまたはCLI統合により、イメージスキャンプロセスがトリガーされます。失敗したビルドは、構成されている場合は停止されます。
- **結果の確認（CIツール）**：開発者は、統合CIツール（Jenkins）で結果を分析できます。

（オプション：ポリシーを追加するか、ニーズに合わせてデフォルトポリシーを調整し、特定のイメージまたはタグにポリシーを割り当て、アラートと通知を構成します。）

実行中のコンテナのスキャン（セキュリティ担当者）

セキュリティ担当者は、イメージスキャンを使用して、実行中のコンテナ、スキャンステータス、およびイメージに新しい脆弱性が存在するかどうかを監視します。

始めに：

- **レジストリの追加**：イメージにアクセスするために必要な資格情報とともに、イメージが保存されているレジストリを追加します。
- **イメージのスキャン**：イメージスキャンをトリガーします（手動で、または自動スキャンするようにアラートを構成します）。
- **レビュー結果（Sysdig Secure）**：セキュリティ担当者は、Sysdig Secureイメージスキャンで結果を分析できます。

（オプション：ポリシーを追加するか、ニーズに合わせてデフォルトポリシーを調整し、特定のイメージまたはタグにポリシーを割り当て、アラートと通知を構成します。）

注意

イメージスキャンには、外部の脆弱性フィードへのアクセスが必要です。 最新の定義に適切にアクセスするには、ネットワークとポートの要件を参照してください。

イメージスキャンを開発パイプラインに統合する

開発パイプラインの一部としてイメージスキャンを使用して、ベストプラクティス、脆弱性、およびシークレットコンテンツをチェックするオプションがあります。

インラインスキャン

バージョン2.5.0以降、Sysdig Secureユーザーは、ローカルでイメージをスキャンおよび分析し、レジストリへのアクセスを提供せずにインフラストラクチャメタデータをSysdigプラットフォームに送り返すことができます。この機能は、さまざまな場合に必要になることがあります。

- イメージは自分の環境を離れません
- SaaSユーザーは、SysdigのSaaSサービスにイメージと独自コードを送信しません
- レジストリを公開する必要はありません
- イメージをより簡単に並行してスキャンできます
- レジストリに到達する前にイメージをスキャンできます。
    - レジストリのコストを削減
    - ビルドパイプラインを簡素化する

前提条件

- Sysdig SecureおよびSysdigインストールに接続する機能
- Dockerエンジン
- DockerHubへのアクセス
- Bash

インラインスキャンを実装する

- スクリプトにアクセスする
- ここ(https://raw.githubusercontent.com/sysdiglabs/secure-inline-scan/master/inline_scan.sh)からinline_scan.shスクリプトをダウンロードします。

- パラメーターと例を確認する
- GitHubのReadMeファイルには、スクリプトパラメーターとその使用法が説明されており、詳細な例が示されています。
- 期待される出力

  スキャンがトリガーされた後、コマンドラインは成功または失敗の結果メッセージを送信します。

  詳細な結果分析を表示するには、Sysdig Secureダッシュボードにログインして、[Scan Result]ページを確認します。

パイプライン統合の例

さまざまなパイプラインについて、詳細に文書化された例があります。

- Gitlab: https://sysdig.com/blog/gitlab-ci-cd-image-scanning/
- Githubアクション: https://sysdig.com/blog/image-scanning-github-actions/
- AWS Codepipeline: https://sysdig.com/blog/image-scanning-aws-codepipeline-codebuild/
- Azureパイプライン: https://sysdig.com/blog/image-scanning-azure-pipelines/
- CircleCI: https://sysdig.com/blog/image-scanning-circleci/

Jenkinsと統合する

Sysdigには、SysdigイメージスキャンをJenkinsベースのビルドプロセスに統合するプラグインがあります。

Jenkinsプラグインのインストールと構成

Sysdig Secure Jenkins Plugin(https://plugins.jenkins.io/sysdig-secure/)のドキュメント（jenkins.ioにあります）には次のことが記載されています。

- 前提条件
- プラグインの入手
- Jenkins UIで必要なシステム設定手順

- ビルドステップとしてのSysdig Secure Image Scanningの追加（Jenkins UIで）
- スキャンされたビルドで実行するアクションの設定（ビルドの失敗や警告の発行のタイミングなど）

Jenkinsでスキャン結果を取得する

Sysdigプラグインは、Jenkinsビルドリストにリストされているスキャンレポートを生成します。

Sysdig Scanning Reportをクリックして、概要情報とポリシーチェックと結果のリストを表示します。

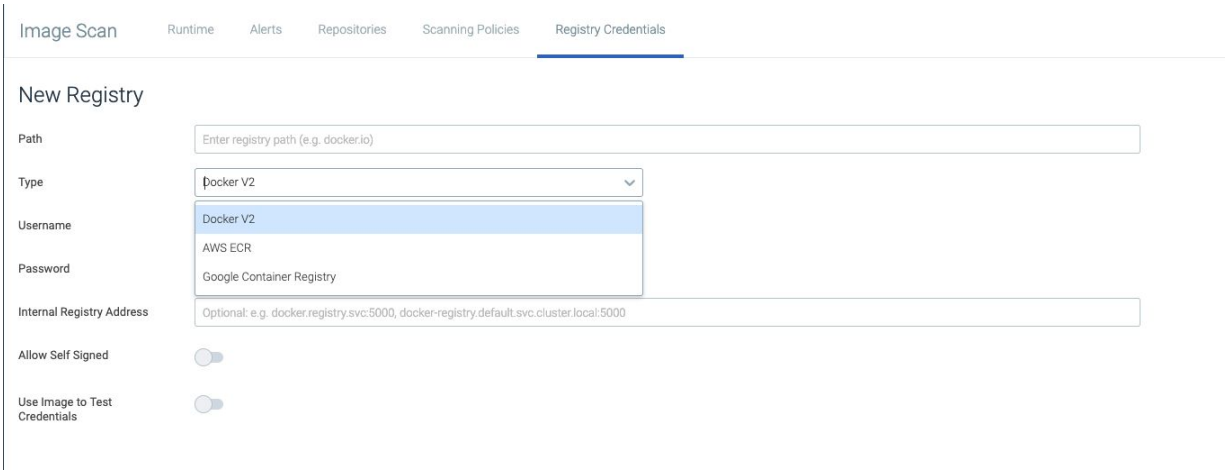
レジストリ資格情報を管理する

Sysdig Secureがイメージをプルおよび分析するには、レジストリ資格情報が必要です。 各レジストリタイプには、必要な資格情報の一意の入力フィールドがあります（例：docker.ioのユーザー名/パスワード、Google Container RegistryのJSONキー）。

新しいレジストリを追加する

1. `Image Scanning`モジュールから、`Registry Credentials`を選択します。
2. `Add Registry`をクリックします。

   [New Registry]ページが表示されます。



3. レジストリへのパスを入力します。 例えば、 docker.io
4. ドロップダウンメニューからレジストリ`Type`を選択します。
5. レジストリ固有の`credentials`を選択します（選択した`Type`に基づいて）

a. Docker V2には多くのDocker V2レジストリがあり、資格要件は異なる場合があります。

   たとえば、Azure Container Registryの場合：

   i. Admin Account
      `Username`: in the `'az acr credentials show --name <registry name>'` command result
      `Password`: The password or password2 value from the `'az acr credentials show'` command result
   ii. Service Principal
      `Username`: The service principal app id
      `Password`: The service principal password

b. **AWS ECR:**
   i. AWS access key
   ii. AWS secret key
c. **Google Container Registry:**
   i. JSON Key

6. （主にOpenShiftクラスターの場合）：内部レジストリアドレスを追加します。

OpenShiftクラスターのイメージレジストリーを実行する推奨方法は、ローカルで実行することです。 Sysdigエージェントは内部レジストリ名を検出しますが、Anchoreエンジンがイメージをプルおよびスキャンするには、内部レジストリ自体にアクセスする必要があります。

例：

外部名：mytestregistry.example.com

内部名：docker-registry.default.svc：5000

注意

Sysdigは内部レジストリ名を外部レジストリ名にマップするため、ランタイムリストとリポジトリリストには外部名のみが表示されます。

7. オプション：スイッチをトグルして、`Allow Self-Signed`証明書を許可します。

   デフォルトでは、UIはTLS / SSL対応のレジストリからのみイメージをプルします。

   （自己署名証明書または不明な認証局からの証明書でレジストリが保護されている場合）証明書を検証しないようUIに指示するには、[`Allow Self-Signed`]を切り替えます。

8. オプション：「`Test Credentials`」スイッチを切り替えて、エントリーを検証します。

   有効にすると、Sysdigは入力された資格情報を使用してイメージをプルしようとします。成功すると、レジストリが保存されます。失敗した場合、エラーが表示され、資格情報またはイメージの詳細を修正できます。

   有効にした場合は、`test registry path`を次の形式で入力します。

   ```
   registry/repo:tag
   ```

9. 例えば。 **`quay.io/sysdig/agent:0.89`**

10. Saveをクリックします。

レジストリを編集する

1. `Image Scanning`モジュールから、`Registry Credentials`を選択します。
2. 既存のレジストリを選択して、`Edit`ウィンドウを開きます。

3. 必要に応じてパラメーターを更新し、[Save]をクリックします。

> **注意**
>
> レジストリタイプは編集できません。

レジストリを削除する

1. Image Scanningモジュールから、Registry Credentialsを選択します。
2. 既存のレジストリを選択して、編集ウィンドウを開きます。
3. [Delete Registry]をクリックし、[Yes]をクリックして変更を確認します。

次のステップ

少なくとも1つのレジストリが正常に追加されると、提供されているDefaultのスキャンポリシーを利用して、イメージをスキャンし、スキャン結果を確認することができます。

スキャンイメージ

イメージが保存されるレジストリクレデンシャルを設定したら、手動または自動でイメージスキャンをトリガーできます。

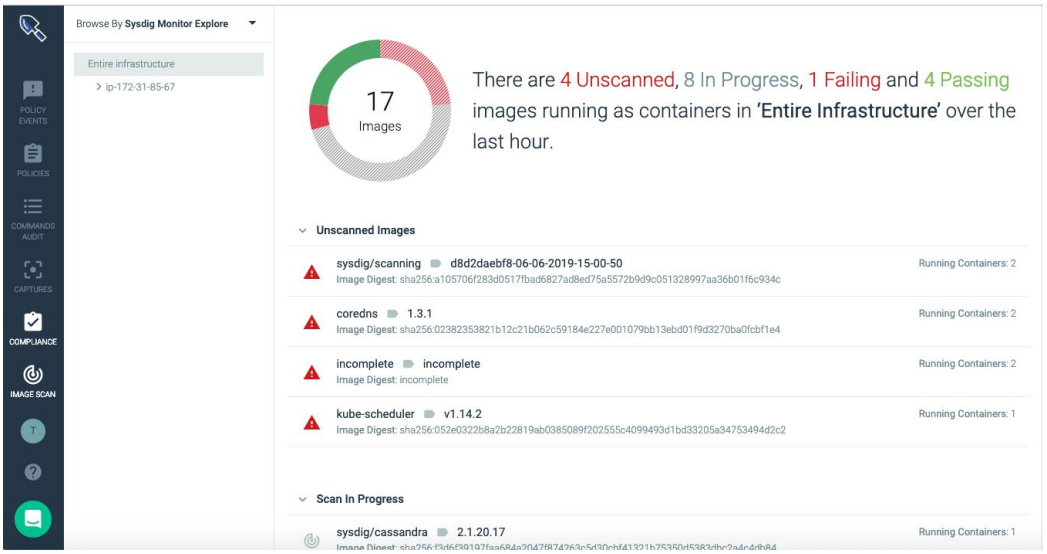
イメージを手動でスキャンする

実行中の環境に新しいイメージを追加する場合、手動でスキャンする必要がある場合があります。 これは、[`Runtime`]タブまたは[`Repositories`]タブから実行できます。

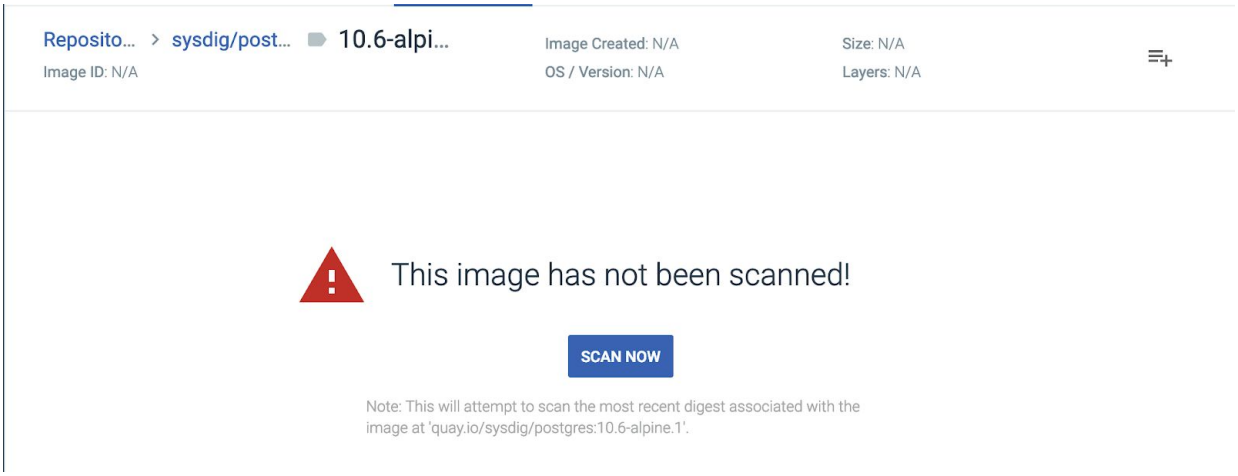
ランタイムタブから

[`Runtime`]タブからイメージを手動でスキャンするには：

1. イメージスキャンモジュールから、[`Runtime`]タブを選択します。



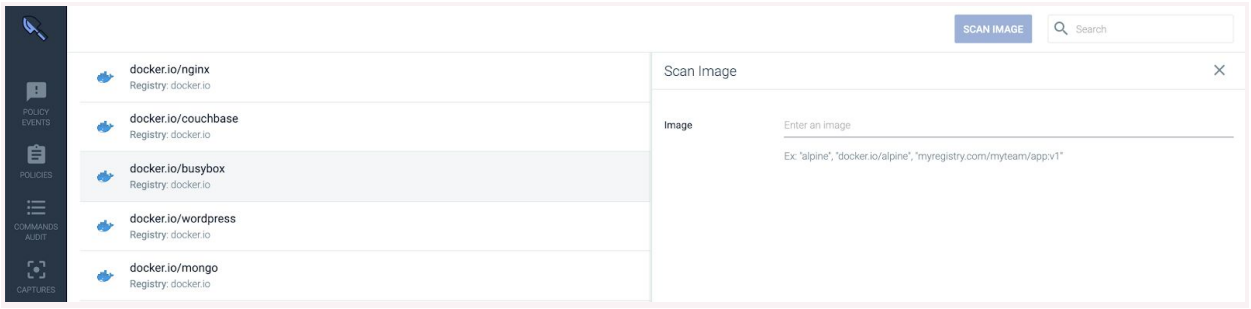
2. スキャンされていないイメージのリストからイメージを選択します。



3. `Scan Now`をクリックします。

リポジトリタブから

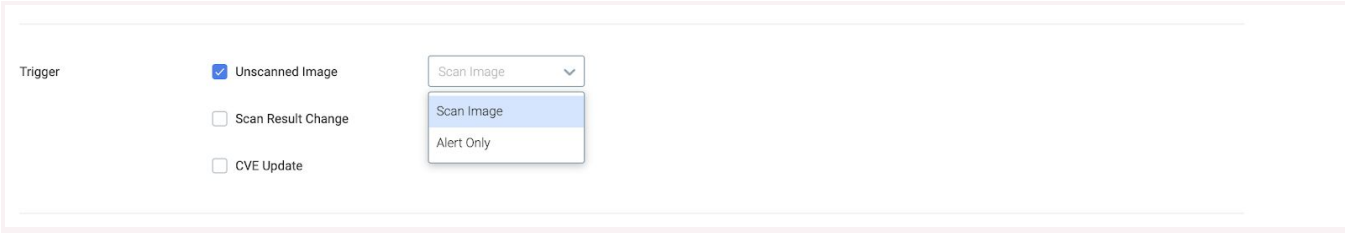
1. `Image Scanning`モジュールから、[`Repositories`]タブを選択します。
2. [`Scan Image`]をクリックします



3. イメージへのパスを定義し、[`Scan`]をクリックします。

イメージを自動的にスキャンする

[`Unscanned Image`]のトリガードロップダウンメニューを[`Scan Image`]に設定して、スキャンされていないイメージが見つかったときに自動的にイメージスキャンをトリガーするスキャンアラートを構成します。



スキャンアラートの管理も参照してください。

スキャンポリシーを管理する

イメージスキャンポリシーは、次のようないくつかのシナリオを定義します。

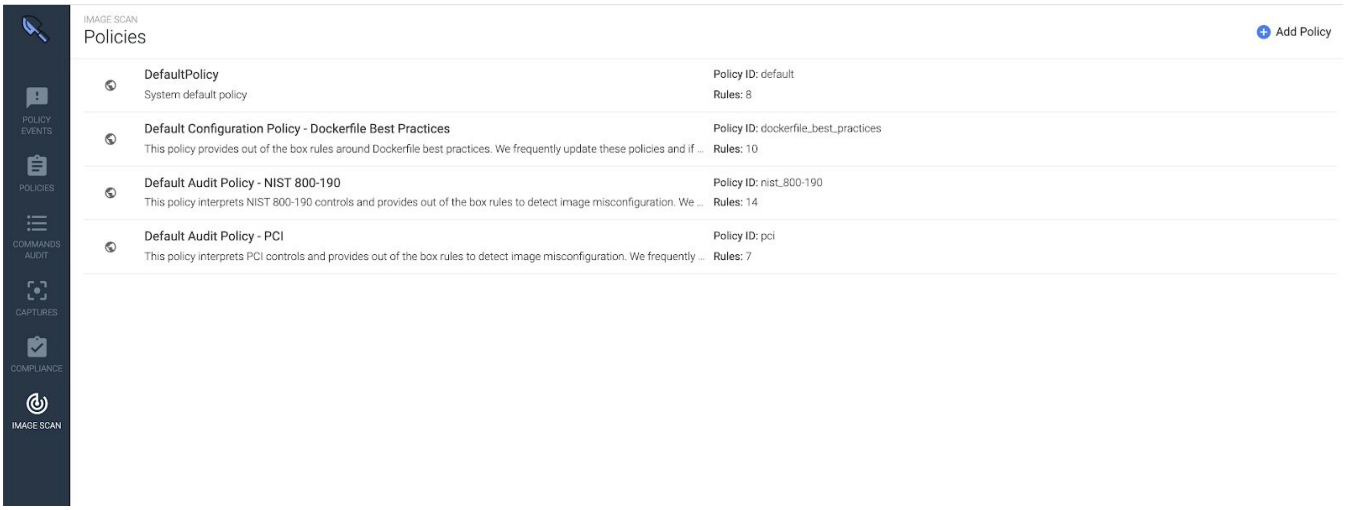
- ビルドプロセスが停止している可能性があります。
- 管理者は、コンテナイメージ内の潜在的なリスクについて警告を受けることがあります。

各スキャンポリシーは、ゲートとトリガーで構成されるルールで構成されます。Sysdigには、レジストリ資格情報が設定されるとすぐにスキャンを実行するために使用できるデフォルトポリシーが含まれています。

ユーザーは、利用可能なスキャンポリシーゲートおよびトリガーから追加のルールまたはポリシーを作成できます。

事前設定されたポリシー

Sysdigには、現状のまま、または構築するテンプレートとして使用できる4つのベースラインポリシーが用意されています。



デフォルトポリシー

このポリシーは、次のような最も一般的なイメージスキャンのケースを対象としています。

- 中および高脆弱性のチェック
- 設定アイテムのチェック（例：イメージのヘルスチェックの確認、公開ポートの禁止）
- 脆弱性フィードデータが最新であることを検証します。

このポリシーは、削除できない基本的な包括的なポリシーです。 他のポリシーの割り当てが行われない場合、デフォルトのポリシーが自動的に使用されます。

ヒント

デフォルトポリシーを編集でき、Sysdig Secureをアップグレードしても編集内容は保持されます

事前設定されたコンプライアンスポリシー

構成済みの他の3つのポリシーは、コンプライアンスルールを扱います。 それらを使用するには、ポリシー割り当てリストに追加する必要があります。

警告

事前設定されたコンプライアンスポリシーを編集する場合は、一致するルールを含む新しいポリシーを作成して編集します。

そうしないと、Sysdig Secureのアップグレード中にカスタマイズが上書きされて失われる可能性があります。

ポリシー設定-Dockerfileのベストプラクティス

このポリシーは、許可しないなど、Dockerfileのベストプラクティスに関するすぐに使用できるルールを提供します。

- 環境変数として焼き付けられた秘密
- ルートユーザー設定
- 露出ポート
- .yumアップグレードを含む指示を実行します。

監査ポリシー-NIST 800-190

このポリシーは、NIST 800-190コントロールを許可しないなどのSysdig Secureスキャニングポリシーにマップします。

- 非公式ノードまたはRubyパッケージ
- Dockerファイルに指示を追加する
- 予期された値以外のベースディストリビューションの使用

監査ポリシー-PCI

このポリシーは、PCI（Payment Card Industry）コントロールをSysdig Secureスキャンポリシーにマップします（脆弱性や資格情報をイメージに含めないようにするなど）。

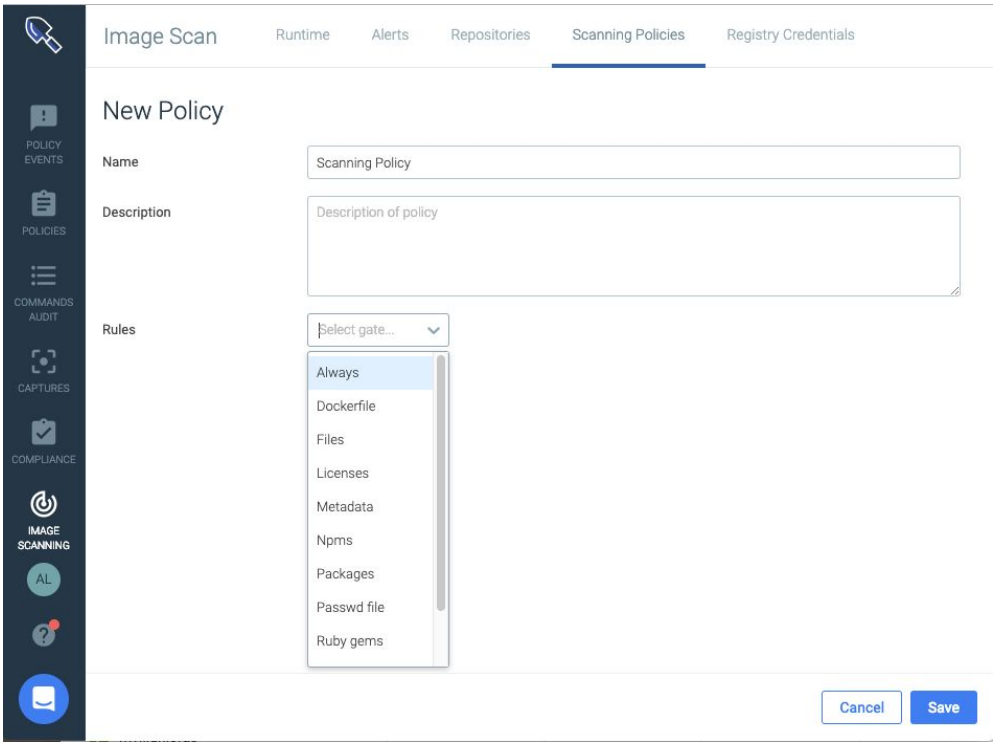
カスタマイズされたポリシー

事前設定されたコンプライアンススキャンポリシーを直接編集しないでください。 一致するポリシーを作成し、編集します。

ポリシーを作成する

1. `Image Scanning`モジュールから、`Scanning Policies`を選択し、`Add Policy(+)`をクリックします。

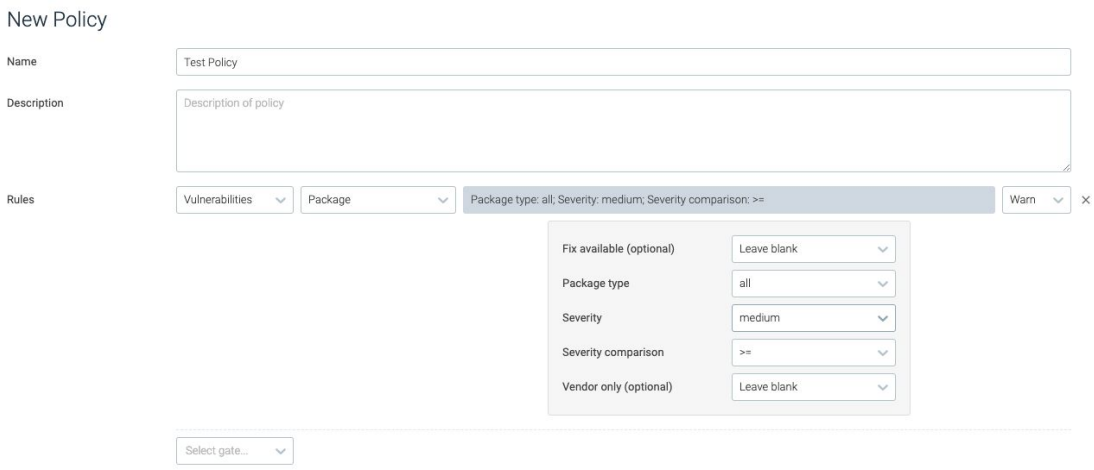
   [New Policy]ページが表示されます



2. 新しいポリシーの`Name`とオプションの`Description`を定義します。
3. `Rule`を追加します。
    a. ドロップダウンメニューから`Gate`を選択し、`Trigger`を選択します。
    b. 関連するパラメーターを設定します。（一部のトリガーでは、パラメーターを設定する必要はありません。）

    各オプションの詳細については、ポリシーゲートおよびトリガーのスキャンを参照してください。

    以下の例では、`package`トリガーで`vulnerabilities`ゲートを使用しています。



4. オプション：ステップ5を繰り返して、必要に応じてルールを追加します。

5. `Save`をクリックします。

ポリシーを編集する

1. `Image Scanning`モジュールから、`Scanning Policies`を選択します。
2. リストから目的のポリシーを選択します。
3. 必要に応じてポリシールールを編集し、[`Save Policy`]をクリックします。

ポリシーを削除する

1. `Image Scanning`モジュールから、`Scanning Policies`を選択します。。
2. リストから目的のポリシーを選択します。
3. [`Delete`]（ゴミ箱）アイコンをクリックし、[`Yes`]を選択して変更を確認します。

ホワイトリスト/ブラックリスト

必要に応じて、特定のイメージまたはCVEをグローバルにホワイトリストまたはブラックリストに登録できます。ホワイトリスト|ブラックリストCVEとイメージを参照ください。これは、ポリシーの評価順序には影響しません。

ポリシーの割り当てを管理する

スキャンに対して非常に単純な単一ポリシーのアプローチを使用しない限り、特定のポリシーを特定のレジストリ、リポジトリ、またはタグに割り当てる可能性があります。

これを行うには、[`Policy Assignments`]ページを使用します。

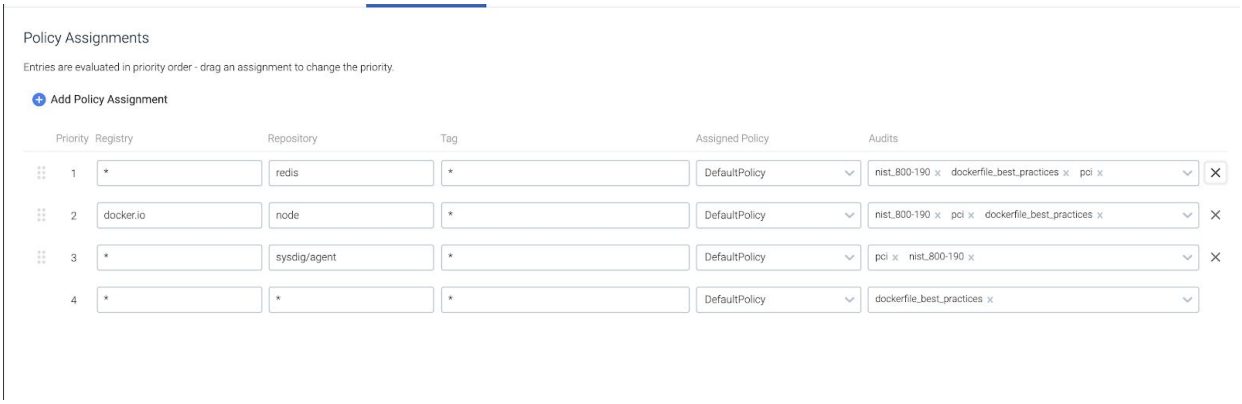
例えば：

- サンプルProdイメージポリシーで「Prod」タグを持つすべてのイメージを評価するには、次の割り当て（registry/repo/tag）を使用します：*/*/Prod
- サンプルGoogleポリシーを使用してgcr.ioからすべてのイメージを評価するには、割り当て（registry/repo/tag）を使用します：gcr.io/*/*

ポリシーを割り当てる

1. `Image Scanning`モジュールから、[**`Scanning Policies`**]を選択し、[+**`Policy Assignments`**]を選択します。

以前に定義された割り当てが優先度順にリストされます。



2. [`+Add Policy Assignment`]をクリックします。

[Assignment]ページの上部に新しいエントリ行が表示されます。目的の割り当ての詳細を入力します。

   a. **`Priority`**：優先度は、割り当てられたポリシーに対する評価の順序です。新しい割り当てはそれぞれ優先度1に自動的に配置されます。ポリシー割り当てを作成して保存したら、リストの新しい位置にドラッグして優先度の順序を変更できます。優先順位の使用も参照してください。
   b. **`Registry`**：任意のレジストリドメイン（例：quay.io）。ワイルドカードがサポートされています。アスタリスク*はレジストリを指定します。
   c. **`Repository`**：任意のリポジトリ（通常=イメージの名前）。ワイルドカードがサポートされています。アスタリスク*は、任意のリポジトリを指定します。
   d. **`Tag`**：任意のタグ。ワイルドカードがサポートされています。アスタリスク*は任意のタグを指定します。
   e. **`Assigned Policy`**：評価に使用するポリシーの名前。ドロップダウンメニューから選択します。

3. 保存をクリックします。

4. オプション：ドラッグハンドル（行の左側にある4つのドット）をクリックし、割り当てをリスト上の別の場所にドラッグして、優先順位を再編成します。

優先順位を使用する

複数のスキャンポリシーを使用する場合、Anchoreエンジンは、ポリシー割り当てリストの優先度1から開始して、トップダウンの順序でそれらを評価します。入力イメージに一致する最初のポリシー割り当てルールが評価され、以降のすべてのルールは無視されます。したがって、優先順位は重要です。

ヒント

たとえば、2つのポリシー定義が定義されたリストを想像してください。

```
Priority 1 Registry = quay.io Repository = sysdig/*

Priority 2 Registry = quay.io Repository = sysdig/myrepo
```

最初のルールはワイルドカードを使用するため、評価はsysdig /で始まるすべてのリポジトリに適用され、sysdig/myrepoを評価する前に停止します。

優先順位を逆にして、目的の動作を取得します。

ポリシー割り当てリストの下部には、削除できないキャッチオールエントリがあります。 形式は次のとおりです。

```
registry = * repository = * tag = * assigned policy = default
```

（`assigned policy`は変更できますが、他のフィールドは編集できません。）

この行の目的は、別のポリシー評価に該当しないレジストリが、少なくともシステム設定の`Default`ポリシーに対して評価されるようにすることです。

ホワイトリスト| ブラックリストCVEとイメージ

Sysdig Secureを使用すると、ユーザーはCVEとイメージをグローバルに信頼されたまたはブラックリストに登録されたものとして定義できます。たとえば、低リスクのCVEをグローバルに承認して、より重要な修正を含むビルドに影響を与えないようにすることができます。または、特定のイメージをグローバルに承認済みまたは未承認としてマークして、常に/決してスキャンに合格しないようにすることができます。

ヒント

ユーザー、ポート、パッケージなど、他のエンティティのブラックリストオプションは、ポリシーゲートとトリガーのスキャンにリストされています。

ホワイトリスト化/ブラックリスト化されたCVEとイメージを確認する

ホワイトリスト/ブラックリストに登録されたCVEとイメージの現在のリストを確認するには：

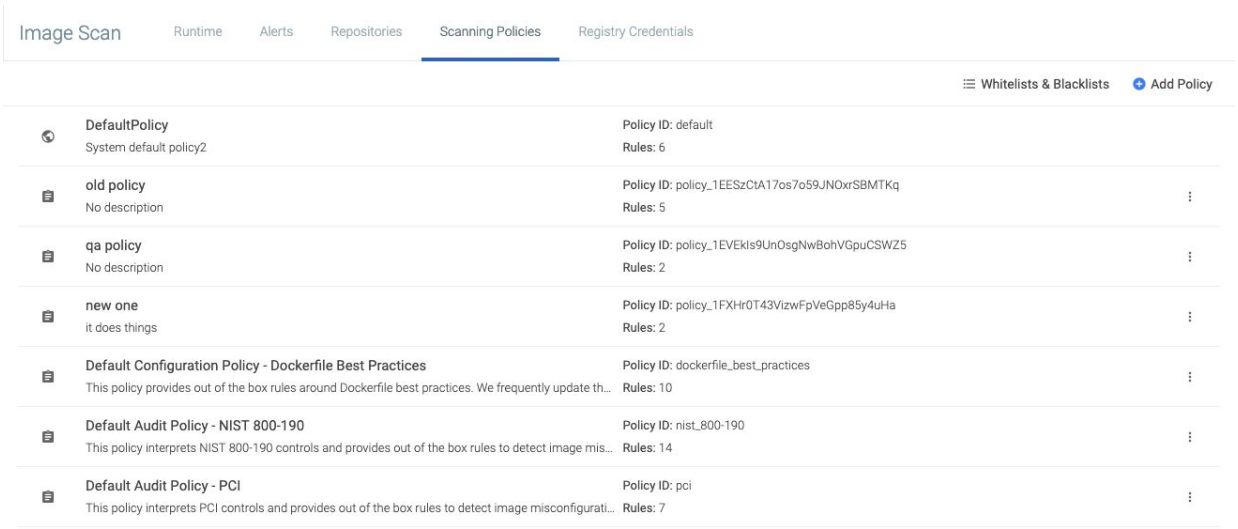
1. `Image Scanning`モジュールから、`Scanning Policies`を選択します。
2. [`Whitelists and Blacklist`]ボタンをクリックします。
3. 関連するタブ（**`CVE Whitelist`**、**`Global Trusted Images`**、または**`Global Blacklisted Images`**）を選択します。

CVEをホワイトリストに登録する

CVEをホワイトリストに登録するには、[`Scanning Policies`]タブと[`Repositories`]タブの2つの方法があります。

[Scanning Policies]タブから：

1. `Image Scanning`モジュールから、`Scanning Policies`を選択します。



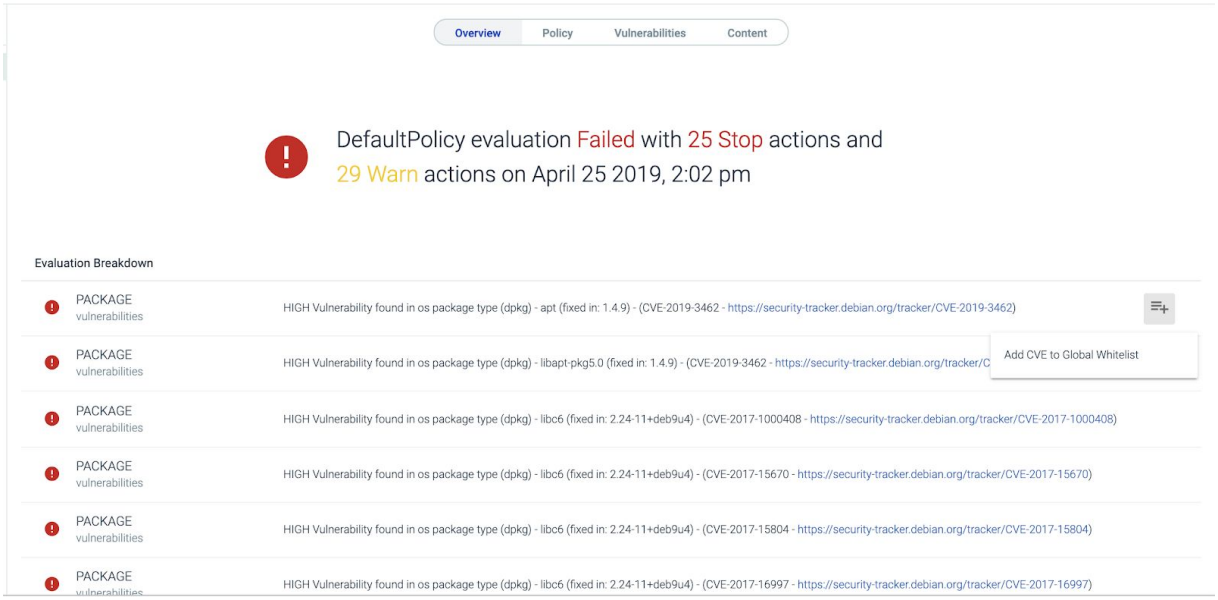
2. [`Whitelists and Blacklists`]をクリックします。

3. [`Add CVE`]をクリックし、各CVEをコンマ区切りリストに追加し、[`Ok`]をクリックして保存します。

   リスト内の各項目は、CVE命名形式（`CVE-YEAR-ID`）に従う必要があります。

リポジトリタブから

1. `Image Scanning`モジュールから、[`Repositories`]を選択し、リストされているリポジトリのいずれかを選択します。
   脆弱性に関連するポリシー結果がある場合、そのCVEをホワイトリストに登録できます。
2. 関連するCVEの横にある[`More Options`]（+）アイコンをクリックします。



3. [Add CVE to Global Whitelist]を選択します。

   これで、CVEが[CVE Whitelist]タブにリストされます。

イメージのホワイトリスト/ブラックリスト

注意

イメージが「信頼できるイメージ」リストと「ブラックリストに登録されたイメージ」リストの両方に追加された場合、ブラックリストに登録されたものが優先されます。

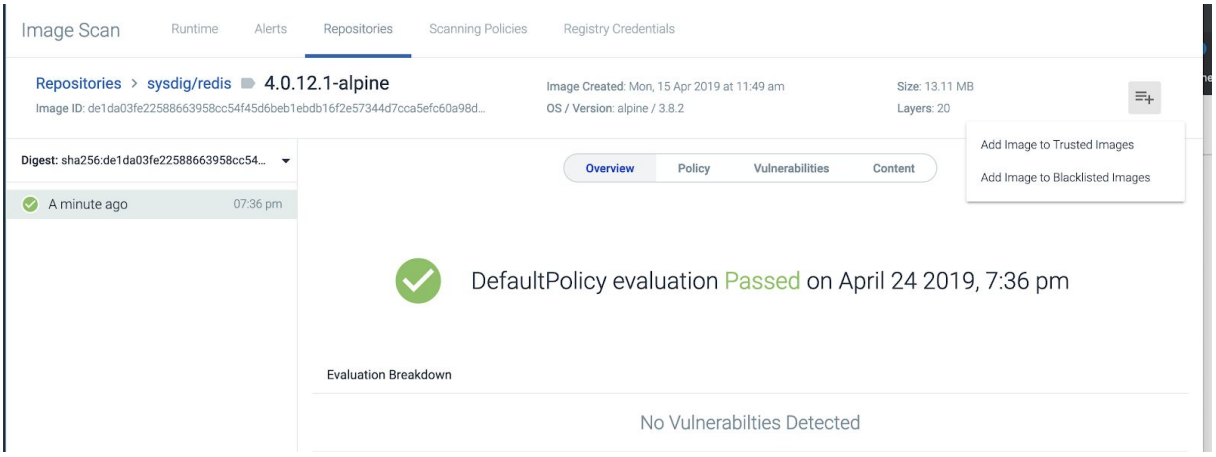
イメージをホワイトリストまたはブラックリストに登録するには、[Scanning Policies]タブと[Repositories]タブの2つの方法があります。

[**Scanning Policies**]タブから：

1. Image Scanningモジュールから、Scanning Policiesを選択します。
2. [Whitelists and Blacklists]をクリックします。
3. 関連するタブ（**Global Trusted Images**、または**Global Blacklisted Images**）を選択し、Add Imageボタンをクリックします。
4. 各イメージをコンマ区切りリストに追加し、[Ok]をクリックします。
   タグ名は有効なASCIIでなければならず、小文字と大文字、数字、アンダースコア、ピリオド、ダッシュを含めることができます。
   タグ名はピリオドまたはダッシュで始まってはならず、最大128文字を含めることができます。

[**Repositories**]タブから：

1. Image Scanningモジュールから、Repositoriesを選択します。
2. リストから関連するリポジトリを選択し、関連するイメージを開きます。
3. ページ上部の[More Options]（+）アイコンをクリックします

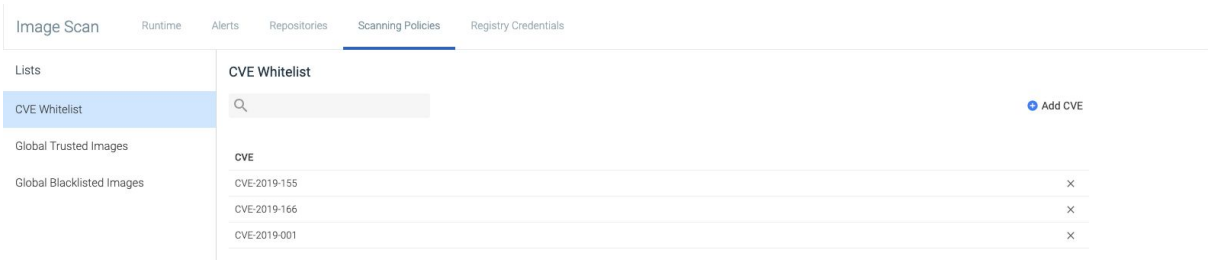


1. 必要に応じて、[`Add Image to Trusted Images`]または[`Add Image to Blacklisted Images`]を選択します。
2. これで、必要に応じて、CVEが[`Global Trusted Images`]タブまたは[`Global Blacklisted Images`]タブにリストされます。

ホワイトリスト/ブラックリストからCVEまたはイメージを削除する

さまざまなリストから1つ以上のCVEまたはイメージを削除するには：

1. `Image Scanning`モジュールから、`Scanning Policies`を選択します。
2. [`Whitelists and Blacklists`]をクリックします。
3. 関連するタブ（**`CVE Whitelist`**、**`Global Trusted Images`**、または**`Global Blacklisted Images`**）に移動します。
4. 関連するCVE/イメージの横にある`Delete`（X）アイコンをクリックします



5. `Save`をクリックします。

Scanning ポリシーゲート、ルール、 及び、トリガー

この文書はSysdig Secureポリシーバンドル内でサポートされているゲート（およびそれらのそれぞれのトリガー/パラメーター）の包括的なリストを提供します。 これらのポリシーゲート、トリガー、およびパラメータを使用して、部分ファイル名のホワイトリスト/ブラックリストから、どのログインシェルが承認されるかを定義するまで、詳細なスキャンポリシーを構築できます。

この情報はまたCLIを使用して得ることができます：

```
user@host:~$ anchore-cli policy describe (--gate <gatename> (--trigger <triggername))
```

スキャンポリシーの構築の詳細については、Image Scanningのドキュメントを参照してください。

Always

このゲートは無条件に起動されるため、有用なテストリソースをユーザーに提供します。

always

ポリシーに存在する場合、alwaysトリガー/ ゲートは作動します。

Alwaysゲートは、イメージブラックリスト/ホワイトリストが期待どおりに機能しているかどうかをテストするのに役立ちます。

Dockerfile

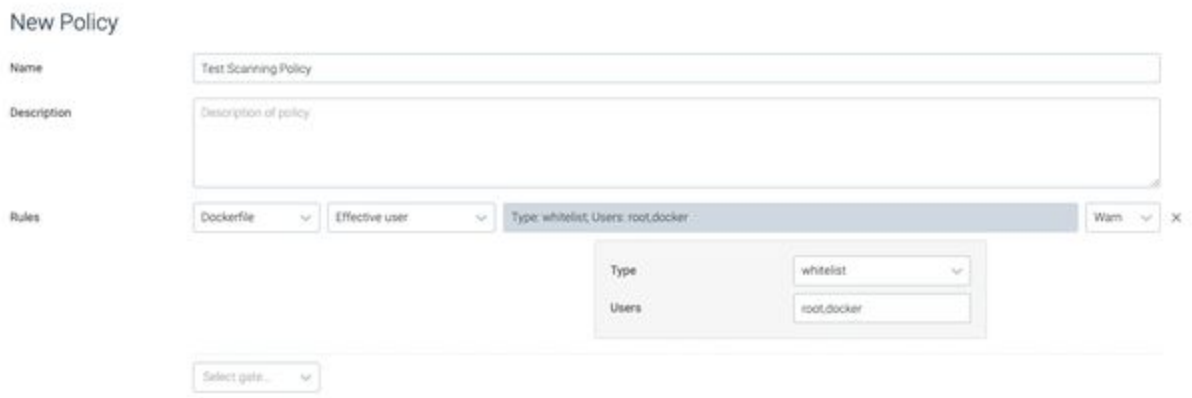
dockerfileゲートは、ベスト・プラクティスに従わない公開ポートおよび命令について、dockerfileの内容、または提供されていない場合はdockerfileの想定内容を確認します。

ゲートは、コンテンツがdocker layer historyに基づいていることを前提としています。

effective_user

このトリガーは、有効なユーザーが提供されたユーザーと一致するかどうかを確認し、構成されたタイプに基づいて起動します。

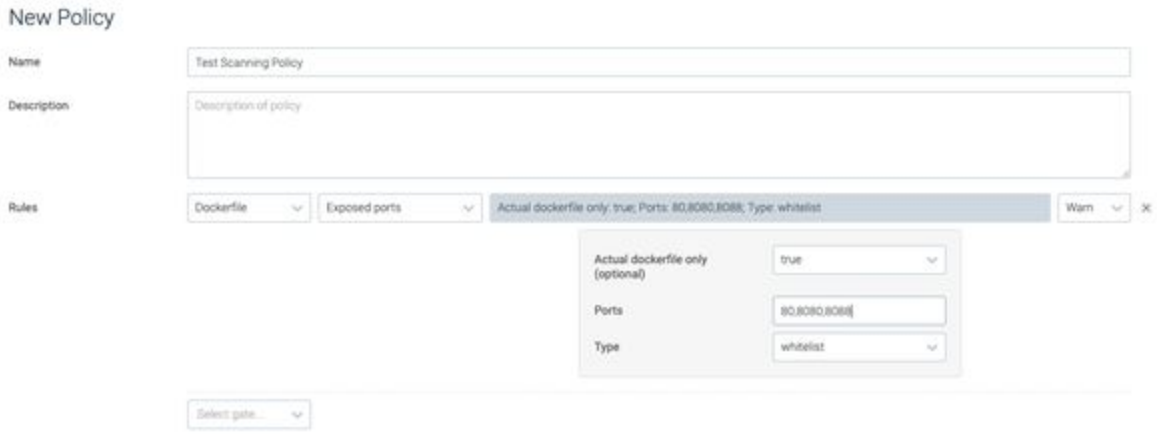
パラメーター	ディスクリプション	例
`type`	ユーザーをホワイトリストに登録するかブラックリストに登録するかを決定します。	N/A
`user`	ユーザーの名前	root,docker



exposed_ports

このトリガーは、公開されているポートのセットを評価して、ホワイトリストに登録するかブラックリストに登録するかを決定します。

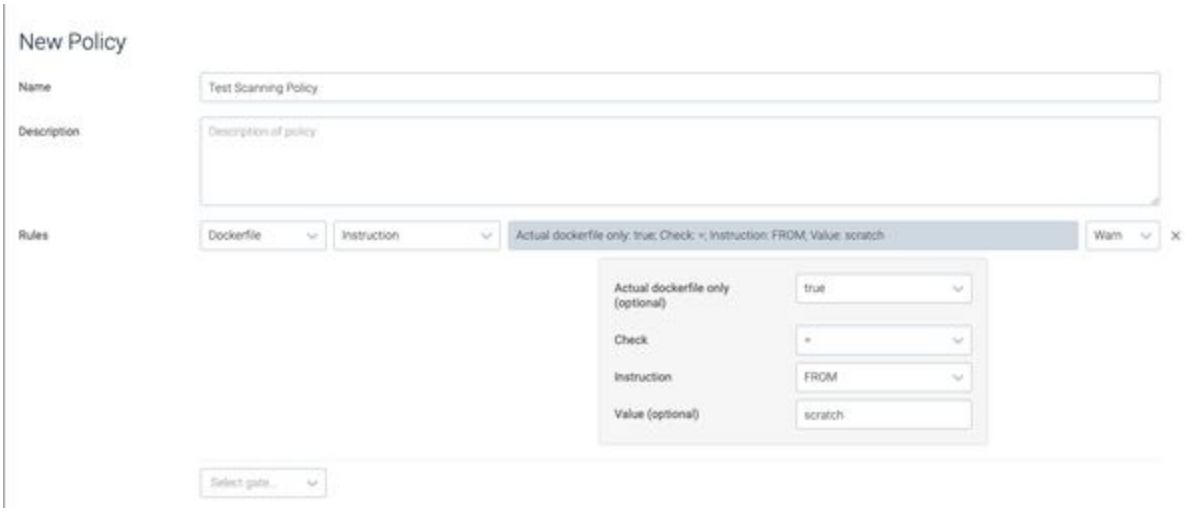
パラメーター	ディスクリプション	例
`actual_dockerfile_only`	評価が推測または推測されたdockerfileをスキップするかどうかを定義し、ユーザー提供のdockerfilesのみを評価します。 デフォルト値はfalseです。	true
`ports`	ポート番号のカンマ区切りリスト	80,8080,8088
`type`	ポートをホワイトリストに登録するかブラックリストに登録するかを定義します	N/A



instruction

このトリガは、リスト内のdirectives/instructionsがdockerfile内の条件と一致するかどうかを評価します。

パラメーター	ディスクリプション	例
`actual_dockerfile_only`	評価が推測または推測されたdockerfileをスキップするかどうかを定義し、ユーザー提供のdockerfilesのみを評価します。 デフォルト値はfalseです。	true
`check`	実行するチェックのタイプ	=
`instruction`	dockerfile instruction の確認	FROM
`value`	dockerfile instructionをチェックする値	scratch



no_dockerfile_provided

イメージにdockerfileが提供されていない場合、このトリガーは作動します。 このトリガーにはパラメーターは必要ありません。

Files

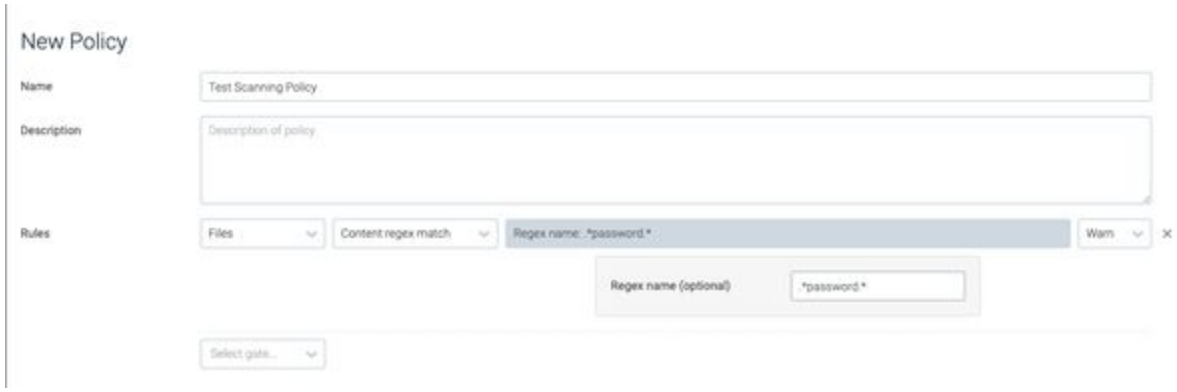
ファイルゲートは、分析されたイメージ内のファイルを確認します。 この評価はファイルの内容、名前、そしてファイルシステムの属性をカバーします。

content_regex_match

このトリガーは、analyzer_config.yamlのcontent_searchセクションで構成された正規表現を使用して一致が見つかったファイルごとに発生します。

正規表現の値に関する詳細は、analyzer_config.yamlファイルを参照してください。

パラメーター	ディスクリプション	例
`regex_name`	FILECHECK_CONTENTMATCHアナライザー・パラメーターに表示される正規表現文字列	.*password.*

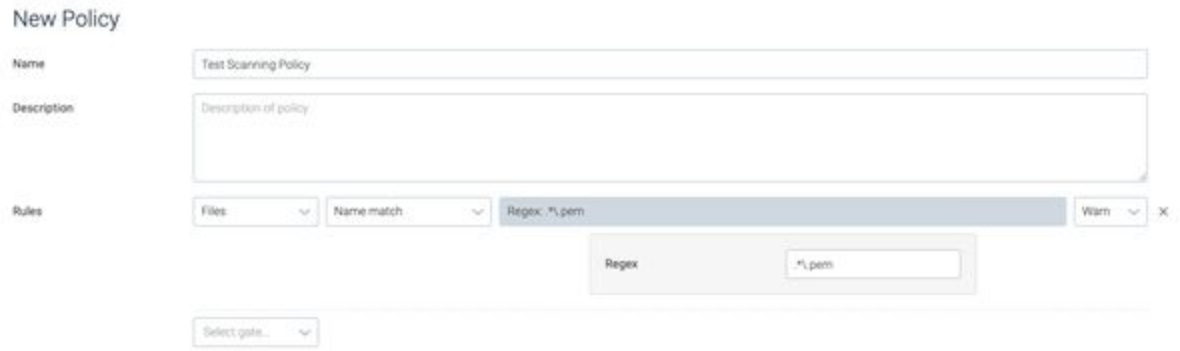


name_match

コンテナ内のファイルの名前が提供された正規表現と一致する場合、このトリガは作動します。

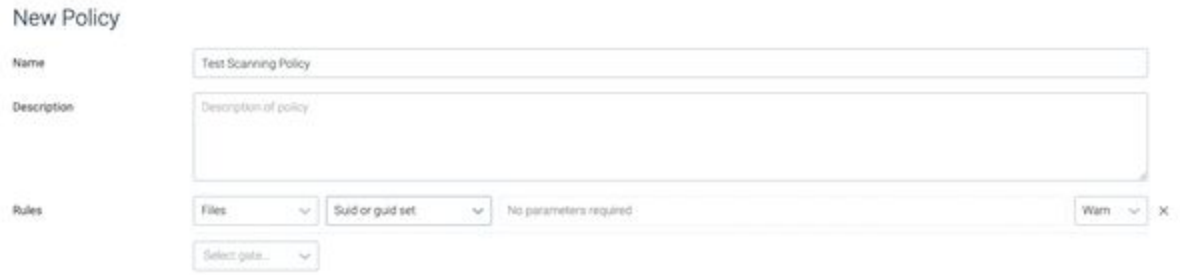
このトリガーは、ポリシー評価にパフォーマンス上の影響を与えます。

パラメーター	ディスクリプション	例
`regex`	検索する正規表現	.*\.pem



suid_or_guid_set

このトリガーは、セットユーザーID（SUID）またはセットグループID（SGID）が構成されているファイルごとに発生します。 パラメータは必要ありません。



Licenses

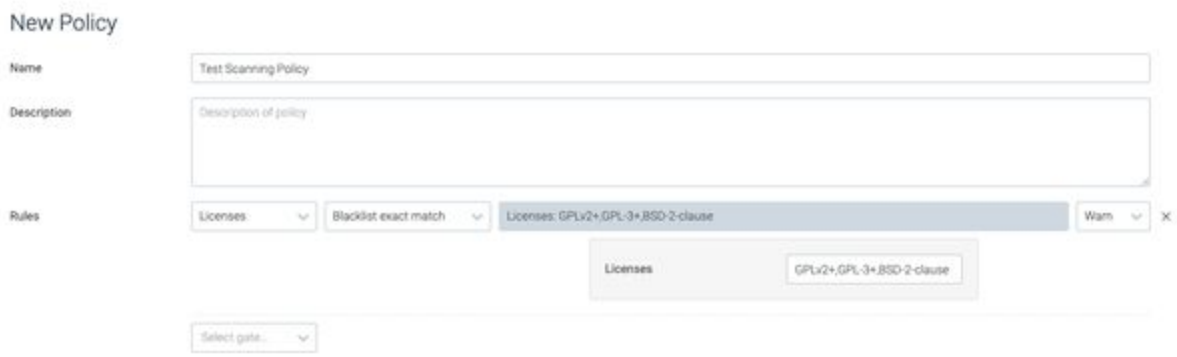
このゲートは、たとえば社内のポリシーに違反しているパッケージが使用されていないことを確認するために、コンテナイメージにあるソフトウェアライセンスを確認するために使用されます。

blacklist_exact_match

指定された正確なライセンスの下で配布されたパッケージがイメージに含まれている場合、このトリガーは作動します。

パラメーター	ディスクリプション	例

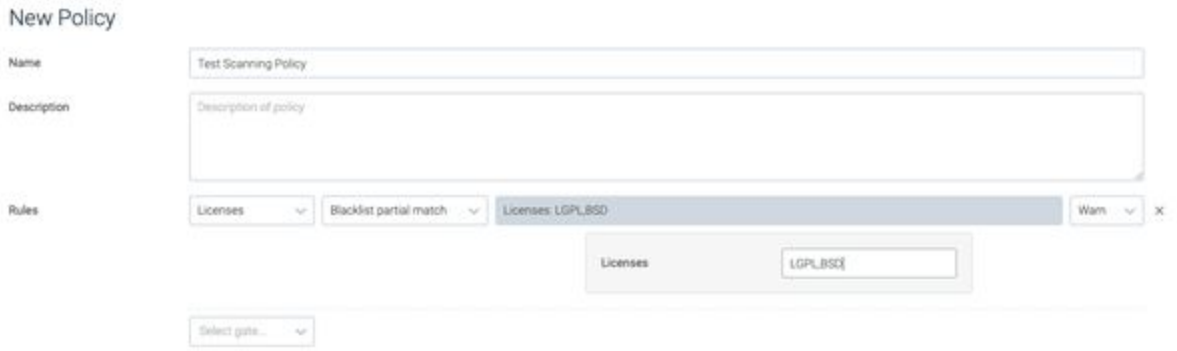
`licenses`	ブラックリストに登録するライセンス名のコンマ区切りリスト	GPLv2+,GPL-3+,BSD-2-clause



blacklist_partial_match

提供された部分文字列を含むライセンスに基づいて配布されたパッケージがイメージに含まれている場合、このトリガーは作動します。

パラメーター	ディスクリプション	例
`licenses`	ライセンスをブラックリストに入れる文字列のコンマ区切りリスト	LGPL,BSD



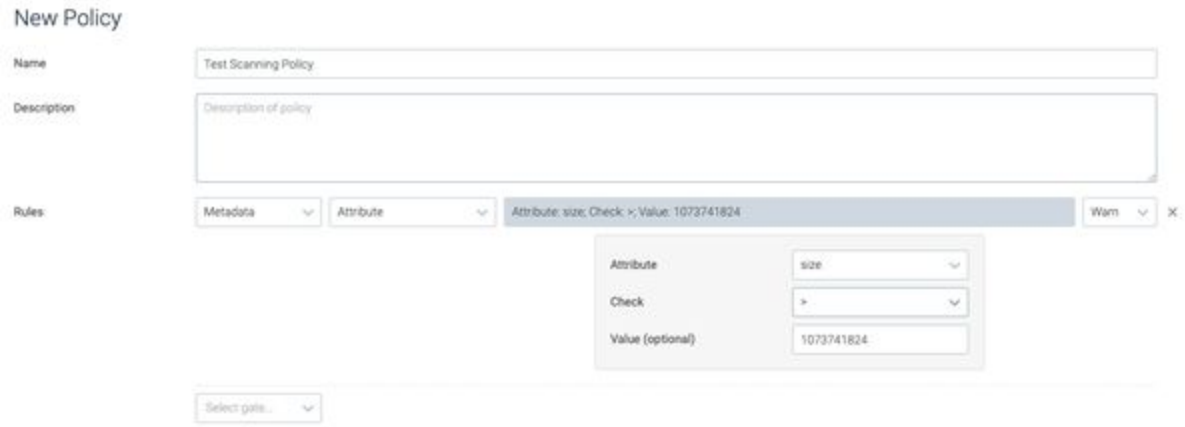
Metadata

このゲートは、サイズ、オペレーティングシステム、アーキテクチャなどのイメージメタデータを確認します。

attribute

名前付きイメージメタデータ値が指定の条件に一致すると、属性トリガーが発生します。

パラメーター	ディスクリプション	例
`attribute`	属性名の確認	size
`check`	評価のために実行する操作	>
`value`	評価に使用する値	1073741824



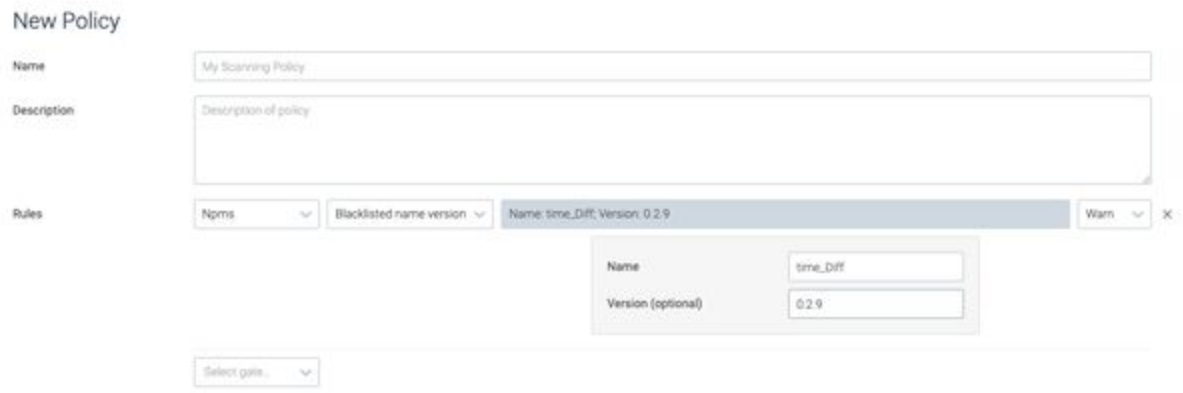
NPMs

NPMゲートは、NPMパッケージがインストールされているすべてのイメージを確認します。

blacklisted_name_version

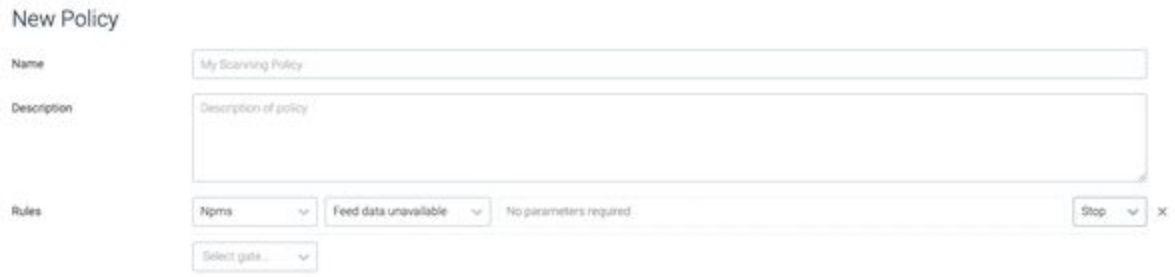
評価されたイメージに、ブラックリストに記載されているNPMパッケージがインストールされている場合は、名前順、またはオプションで名前とバージョン順でトリガされます。

パラメーター	ディスクリプション	例
`name`	ブラックリストに載っている NPMパッケージの名前	time_diff
`version`	ブラックリストに登録するNPM パッケージの特定のバージョン	0.2.9



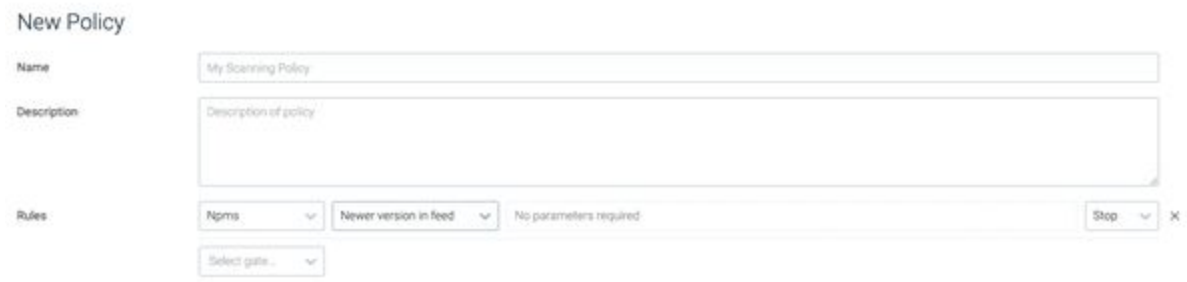
feed_data_unavailable

エンジンがNPMデータフィードにアクセスできない場合、このトリガーは作動します。 パラメータは必要ありません。



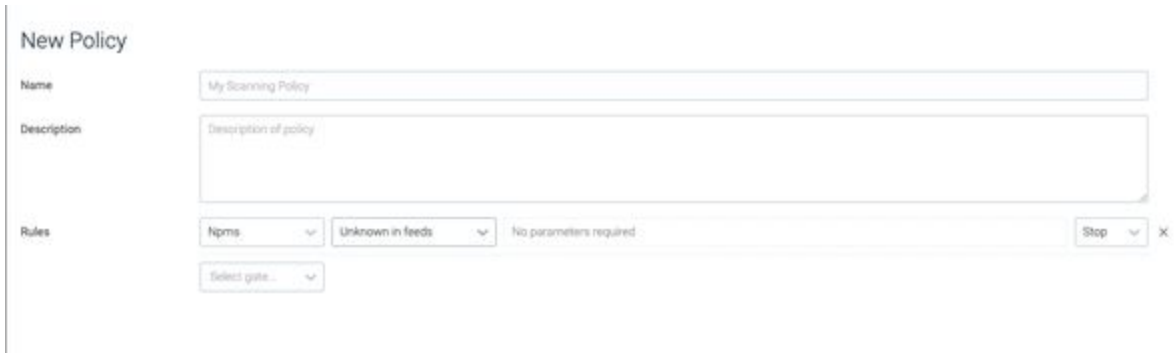
newer_version_in_feed

NPMデータ・フィードにパッケージの新しいバージョンがリストされている場合、このトリガーは作動します。 パラメータは必要ありません。



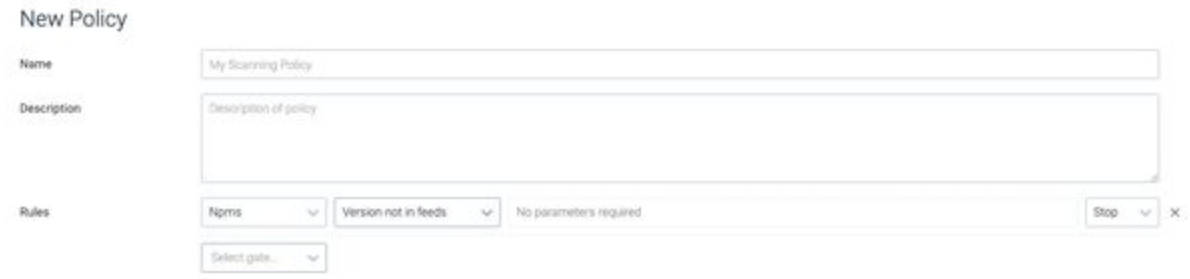
unknown_in_feeds

インストールされているNPMが公式のNPMデータベースにない場合、このトリガーは作動します。 パラメータは必要ありません。



version_not_in_feeds

NPMバージョンが公式のNPMフィードに有効なバージョンとしてリストされていない場合、このトリガーは作動します。 パラメータは必要ありません。



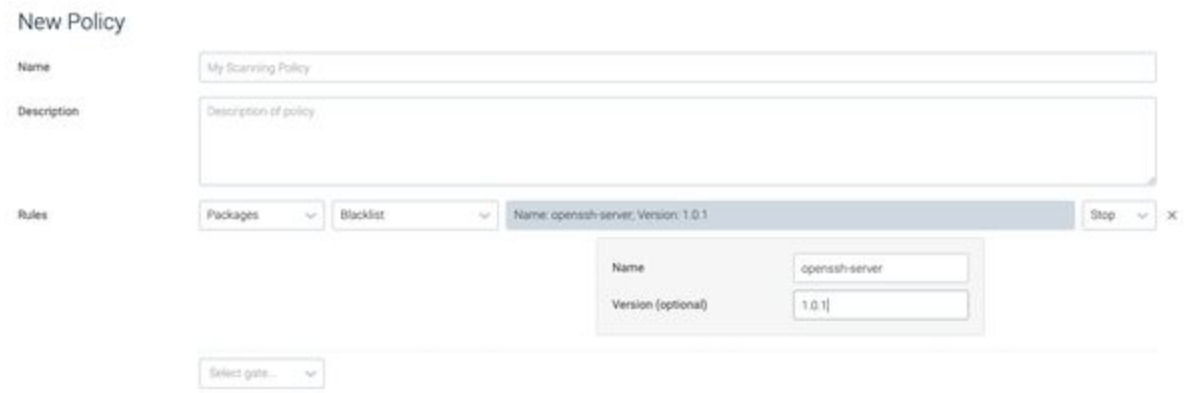
Packages

パッケージゲートは、イメージ内のすべてのパッケージをレビューし、名前、バージョン、およびホワイトリスト/ブラックリストに登録されたパッケージを確認します。

blacklist

このトリガーは、名前、または名前とバージョンのいずれかによってブラックリストに登録されているパッケージがイメージに含まれている場合に発生します。

パラメーター	ディスクリプション	例
`name`	ブラックリストに載っているパッケージの名前	openssh-server
`version`	ブラックリストに載せるべきパッケージの正確なバージョン	1.0.1

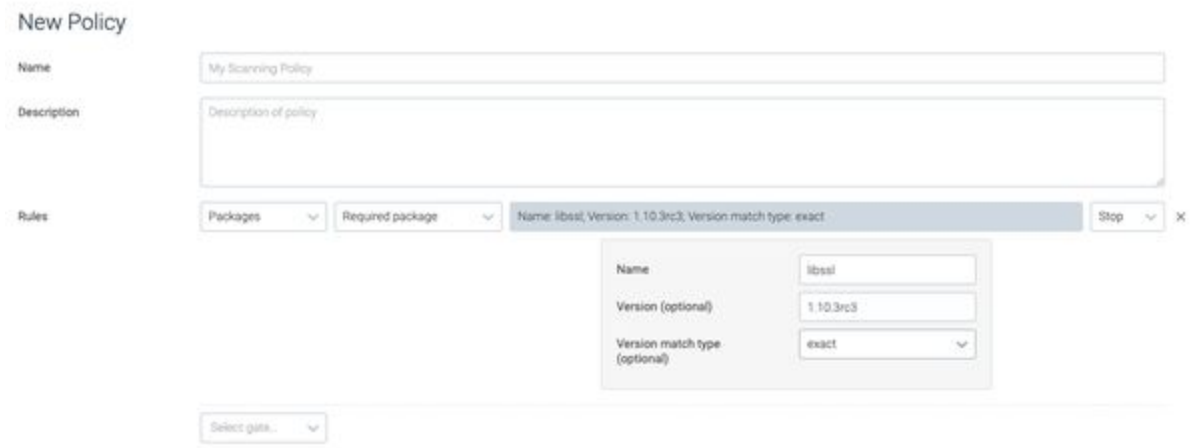


required_package

指定されたパッケージ/バージョンがイメージに見つからない場合は、required_packageトリガーが発生します。

パラメーター	ディスクリプション	例

`name`	必要なパッケージの名前	libssl
`version`	必要なパッケージのバージョン	1.10.3rc3
`version_match_type`	トリガーが正確なパッケージとバージョン（厳密）、または単にパッケージのバージョン（最小）のどちらを必要とするかを定義します。 これはバージョンが定義されている場合にのみ関係します。	exact

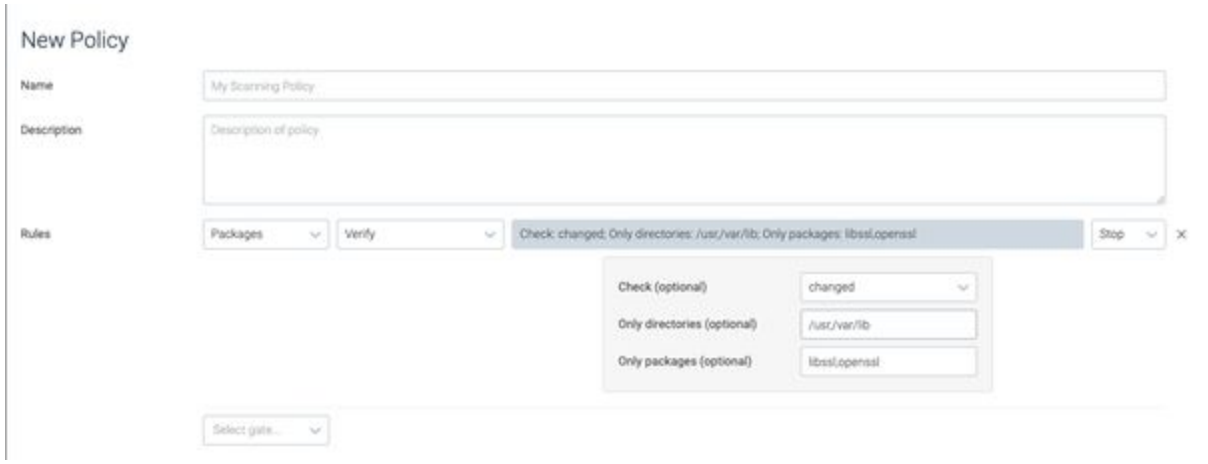


verify

このトリガーは、イメージ内のパッケージデータベースに対するパッケージの整合性を確認し、指定されたディレクトリのすべてまたは定義済みリストのいずれかに含まれるコンテンツの変更または削除を試みます。

パラメーター	ディスクリプション	例
`check`	チェックが不足しているパッケージ、変更されたパッケージ、またはすべてに焦点を合わせるべきかどうかを定義します。	changed
`only_directories`	チェックが制限されるべきディレクトリのリストを定義します	/usr,/var/lib

`only_packages`	検証する必要があるパッケージのリストを定義します	libssl,openssl



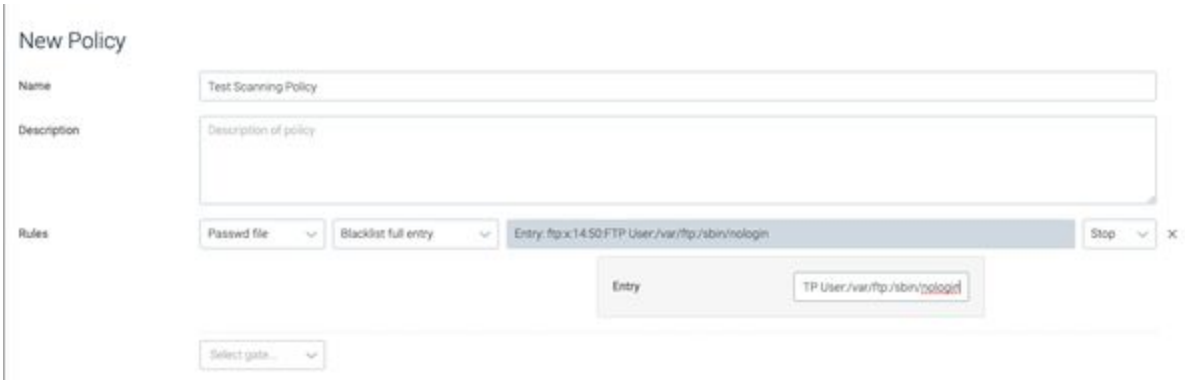
Passwd File

このゲートは、ブラックリストに載っているユーザー、グループ、およびシェルの/etc/ passwdを確認します。

blacklist_full_entry

パスワード全体が/ etc / passwdファイルに見つかると、このトリガーは作動します。

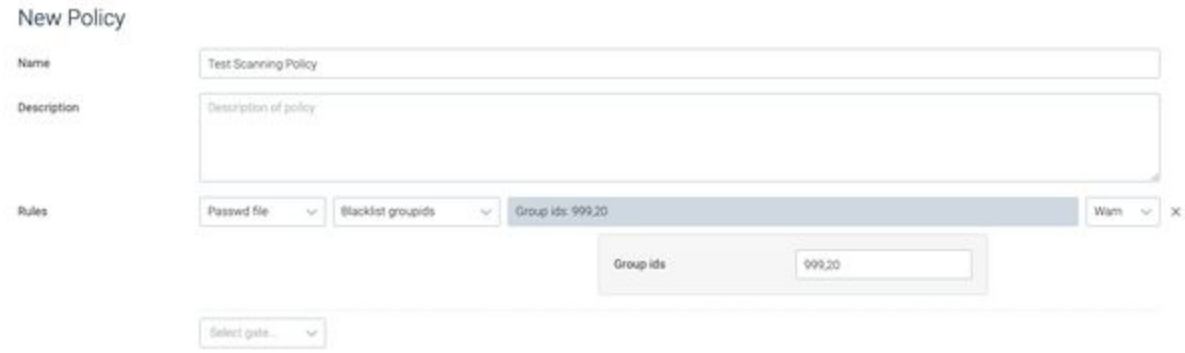
パラメーター	ディスクリプション	例
`entry`	/ etc / passwdに一致する完全なエントリ	ftp:x:14:50:FTP User:/var/ftp:/sbin/nologin



blacklist_groupids

指定されたグループIDが/etc/passwdファイルに見つかった場合、このトリガーは作動します。

パラメーター	ディスクリプション	例
`group_ids`	トリガーを作動させるグループIDのカンマ区切りの数値リスト	999,20

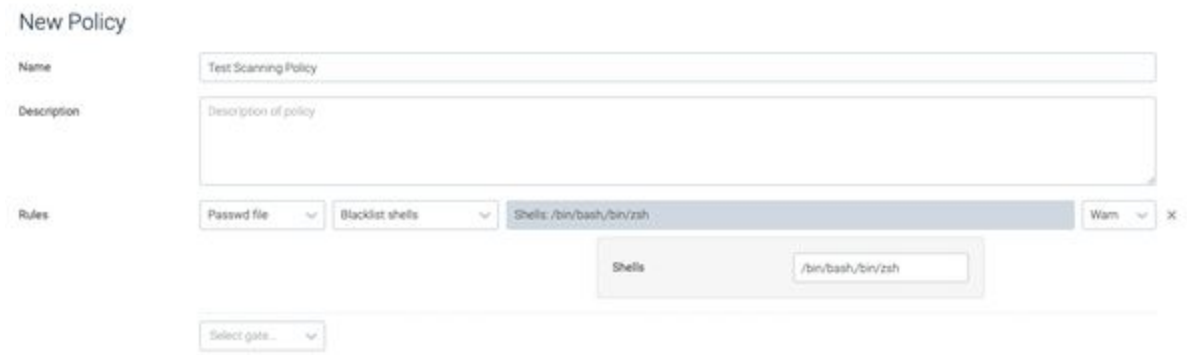


blacklist_shells

指定されたログインシェルが/etc/passwdファイル内の任意のユーザーの下に見つかった場合、このトリガーは作動します。

パラメーター	ディスクリプション	例

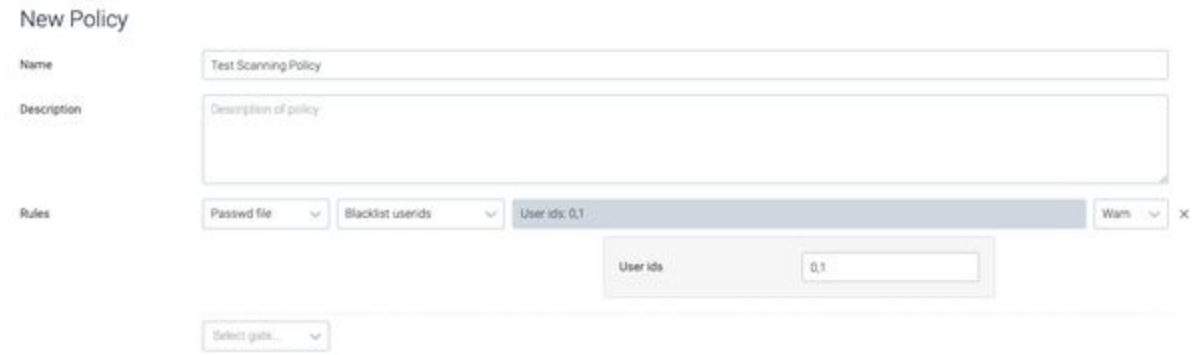
shells	ブラックリストへのシェルコマンドのリスト	/bin/bash,/bin/zsh



blacklist_userids

指定されたユーザーIDが/etc/passwdに存在する場合、このトリガーは作動します。

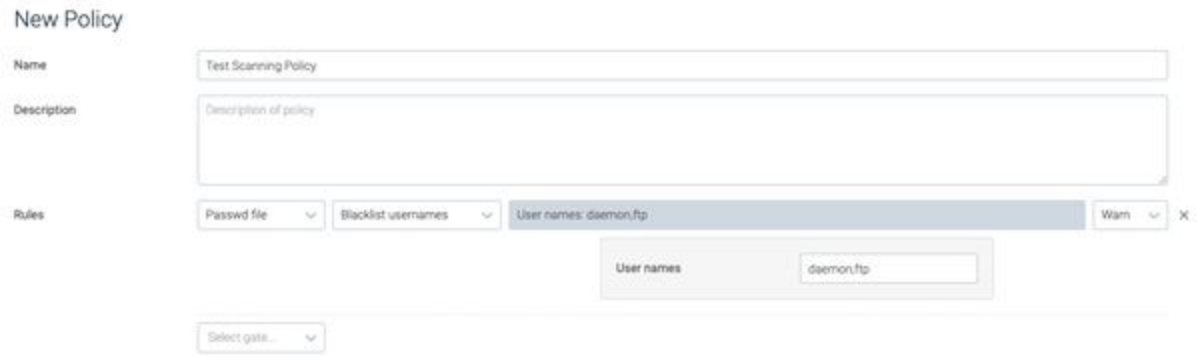
パラメーター	ディスクリプション	例
user_ids	ブラックリストへのユーザーIDの数値、カンマ区切りリスト	0,1



blacklist_usernames

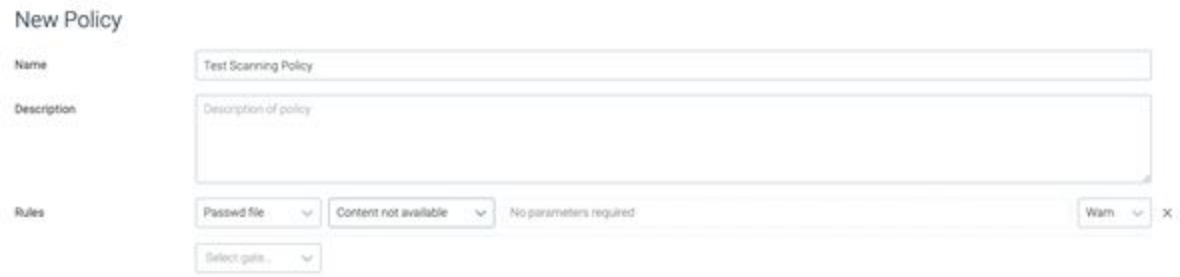
指定されたユーザー名が/etc/passwdファイルに見つかった場合、blacklist_usernamesトリガーは作動します。

パラメーター	ディスクリプション	例
`user_names`	ブラックリストに入れるユーザー名のカンマ区切りリスト	daemon,ftp



content_not_available

/etc/passwdファイルがイメージに存在しない場合、content_not_availableトリガーは作動します。 パラメータは必要ありません。



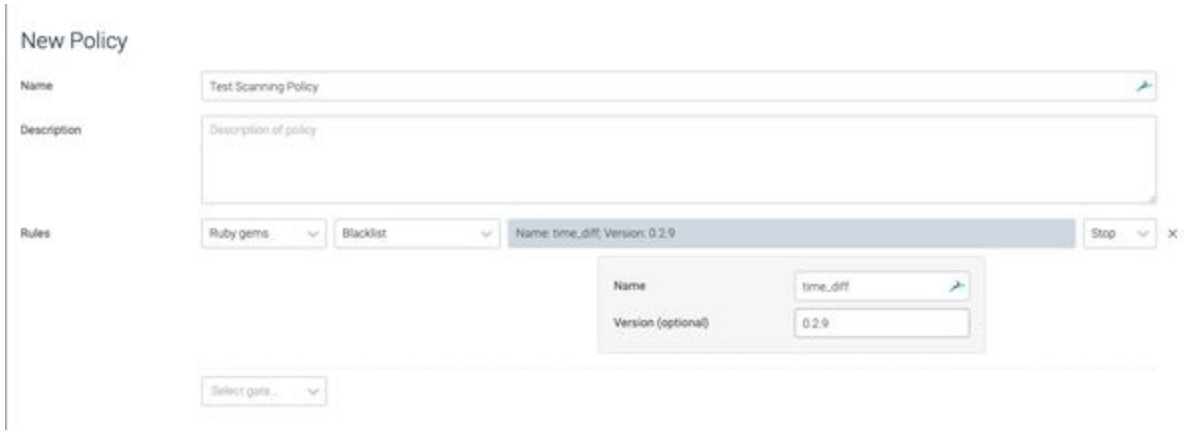
Ruby Gems

Ruby Gemsゲートは、開発者が正式なGEMデータベースからの正式なパッケージを使用していることを保証し、サポートされなくなったバージョンのパッケージを使用していないことを保証します。

blacklist

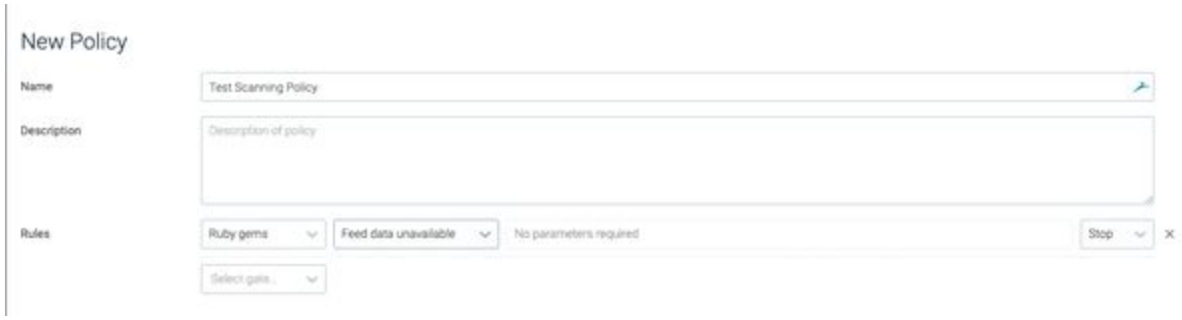
設定された名前とバージョンに一致するGEMパッケージが評価されたイメージで見つかった場合、ブラックリストトリガーは作動します。

パラメーター	ディスクリプション	例
`name`	gemの名前	time_diff
`version`	ブラックリスト gemのバージョン	0.2.9



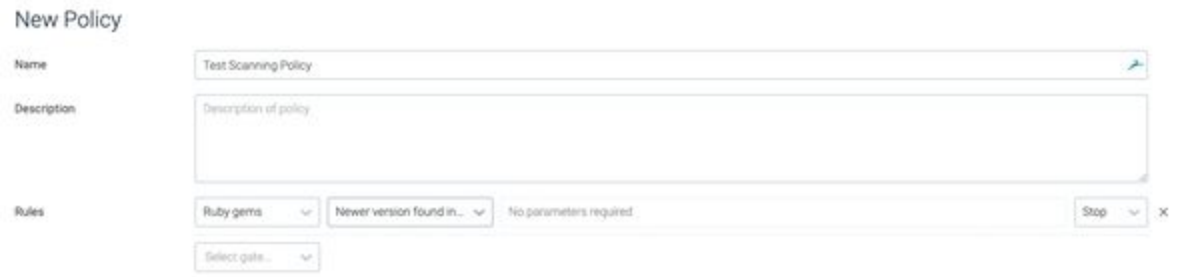
feed_data_unavailable

SysdigがGEMデータフィードにアクセスできない場合、このトリガーは作動します。 パラメータは必要ありません。



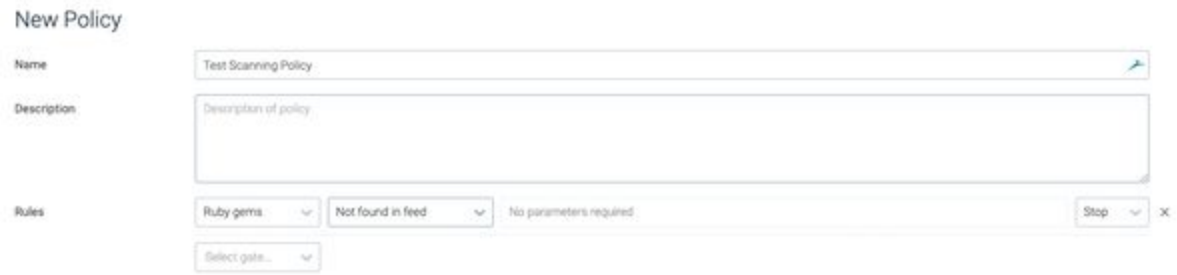
newer_version_found_in_feed

インストールされているGEMパッケージが最新バージョンではない場合、このトリガーは作動します。 パラメータは必要ありません。



not_found_in_feed

インストールされたGEMが公式GEMデータベースにない場合、このトリガーは作動します。 パラメータは必要ありません。



version_not_found_in_feed

GEMが正式なGEMフィードに有効/サポートされているバージョンとしてリストされていない場合、このトリガーは作動します。 パラメータは必要ありません。

Secret Scans

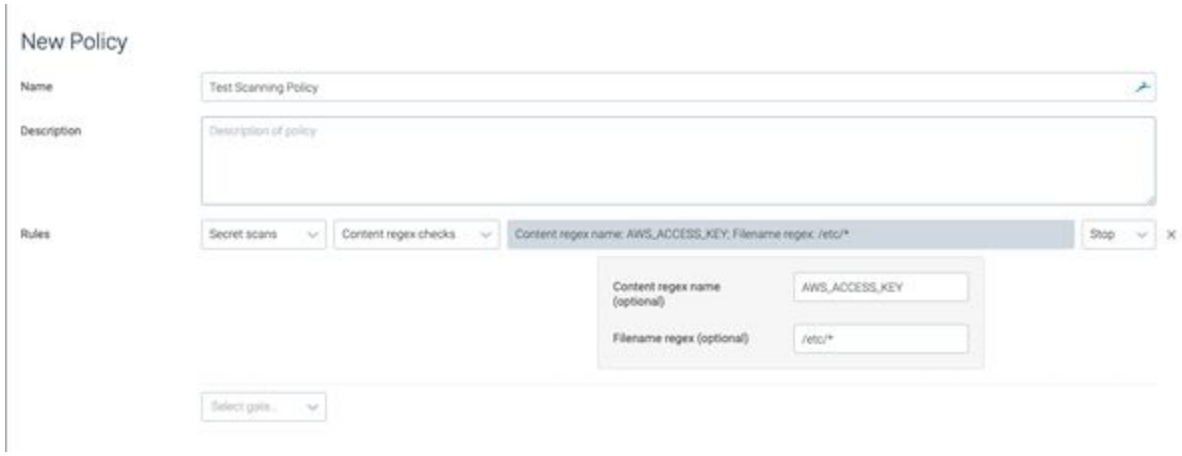
シークレットスキャンは、設定された正規表現に基づいて、イメージが侵害された場合に利用できる可能性のあるシークレットがイメージに焼き付けられているかどうかを判断します。

content_regex_checks

content_regex_checksは、コンテンツ検索アナライザーが構成済みの名前付き正規表現との一致を検出した場合にトリガーを起動します。 一致は、content_regex_nameとfilename_regexのいずれかが設定されている場合は、それらによってフィルタリングされます。

content_regex_nameは、analyzer_config.yamlのsecret_searchセクション値にする必要があります。

パラメーター	ディスクリプション	例
`content_regex_name`	変数/コンテンツの名前。 イメージで見つかった場合にトリガーを作動させます デフォルトで使用可能な名前は AWS_ACCESS_KEY、 AWS_SECRET_KEY、PRIV_KEY、 DOCKER_AUTH、およびAPI_KEYです。	AWS_ACCESS_KEY
`filename_regex`	content_regex_nameの存在について分析する必要があるファイルをフィルタリングします	/etc/.*



Vulnerabilities

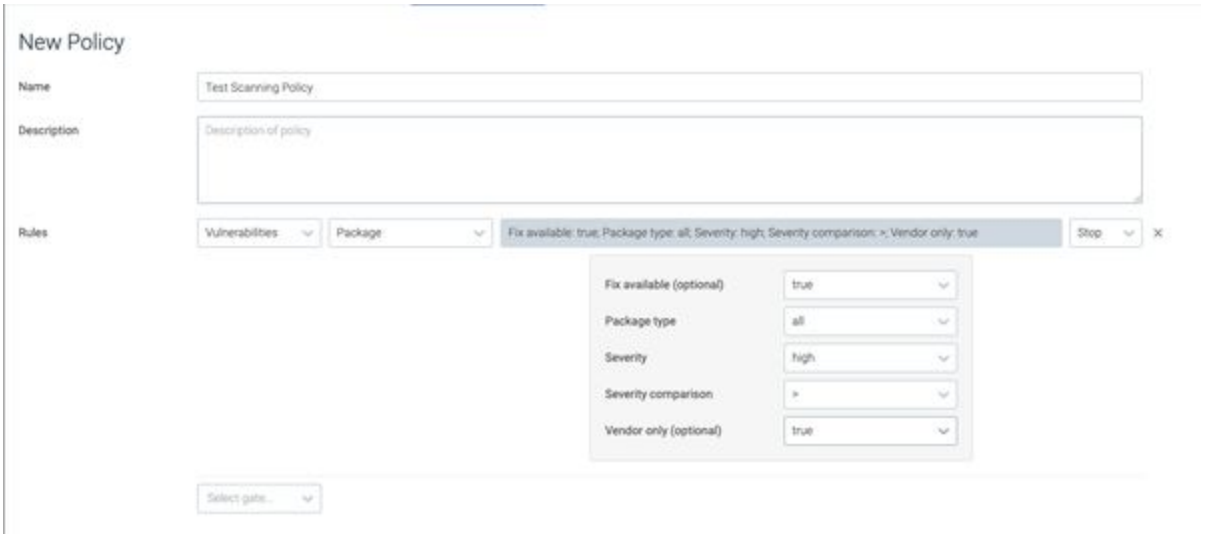
CVE /脆弱性チェックを使用して、含まれているパッケージに設定レベルを超える脆弱性がないこと、指定された期間より古いこと、またはデータが利用できない場合を確認できます。

package

イメージ内の脆弱性が設定された比較基準に一致すると、パッケージトリガーが作動します。 以下の表は、利用可能なパラメータと基準をまとめたものです。

パラメーター	ディスクリプション	例
`fix_available`	存在する場合、脆弱性レコードの修正プログラムの可用性はパラメータの値と一致する必要があります	true
`package_type`	特定のタイプのパッケージ	all
`severity`	脆弱性の深刻度	high
`severity_comparison`	セキュリティ評価のために実行する比較のタイプ	>

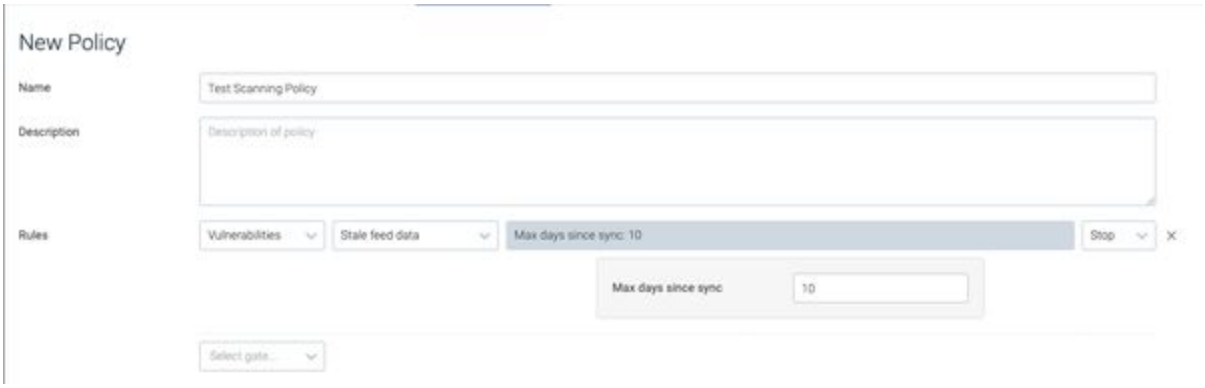
`vendor_only`	trueの場合、このCVEに対して利用可能な修正は、「Won't be addressed by the vendor」として明示的にマークされません	true



stale_feed_data

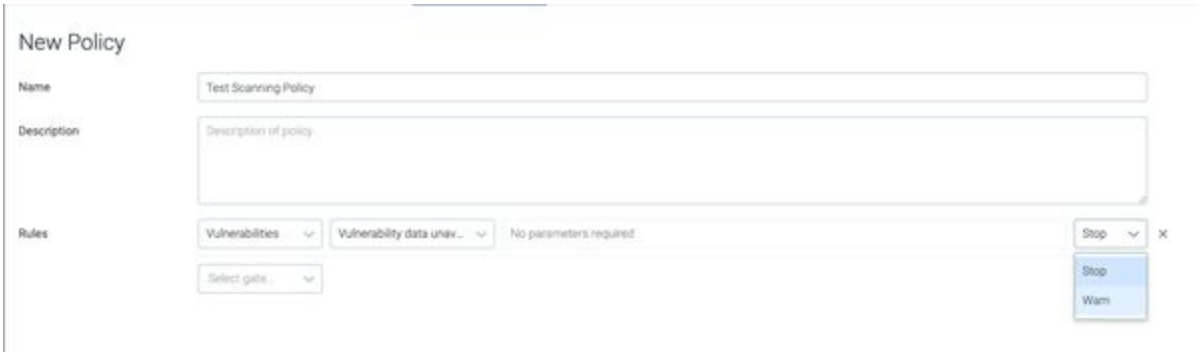
CVEデータが指定されたウィンドウよりも古い場合、stale_feed_dataトリガーが発生します。

パラメーター	ディスクリプション	例
`max_days_since_sync`	トリガーが作動するまでの同期データの日数を決定します	10



vulnerability_data_unavailable

利用可能な脆弱性データがない場合は、脆弱性データ有効化トリガーが作動します。 このトリガーにはパラメーターは必要ありません。



スキャンアラートの管理

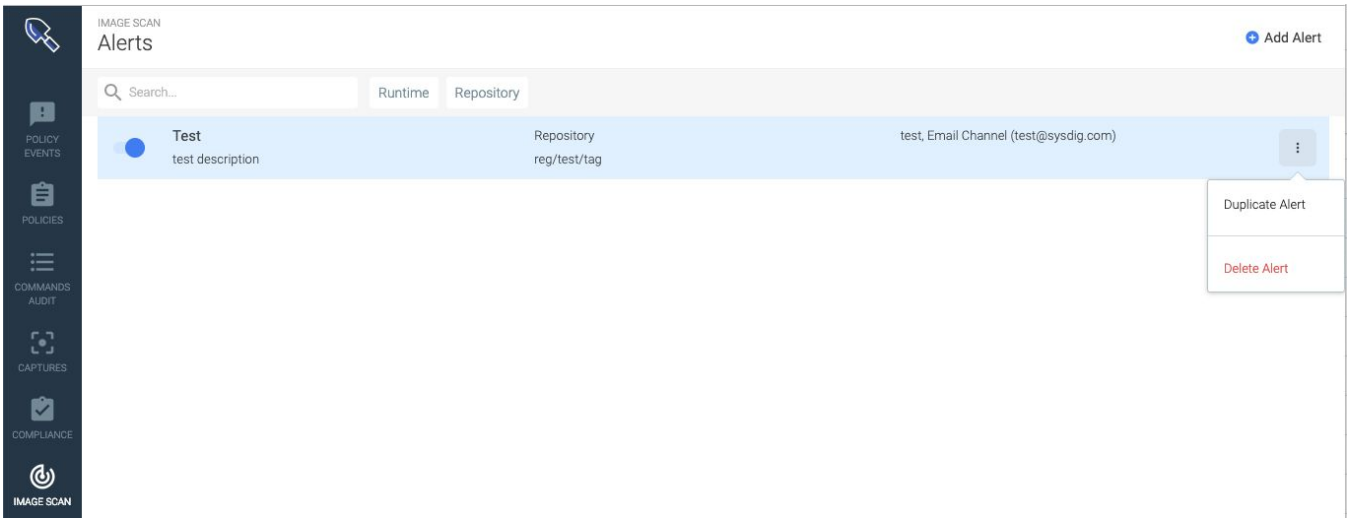
すべてのSysdigアラートと同様に、イメージスキャンアラートは、インフラストラクチャで問題が発生したときにユーザーに通知するように構成できます。 スキャンアラートは、リポジトリ内の静的イメージまたは実行中（実行時）イメージに対して作成できます。スキャンアラートは、スキャンされていないイメージが環境に追加されたとき、イメージがポリシー評価に失敗したとき、スキャン結果が変更されたとき、またはCVEが更新されたときに焦点を合わせます。

ユーザーがアラートを実装する場合の例：

- 処理する3つの異なるイメージの新しいCVE更新があるかどうかを知りたい
- 組織全体で使用されているdocker hubからの共通イメージのいずれかが変更されたポリシーステータスを持っている場合、通知を受けたい

スキャンアラートリストの管理

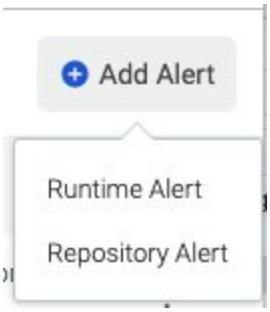
`Image Scanning`モジュールから、[`Alerts`]タブを選択します。 スキャン警告リストが表示されます。



ここから、既存のアラートを検索し、アラートを作成、複製、または削除できます。

アラートを追加する

1. 新しいアラートを作成するには：`Image Scanning`モジュールから、[`Alerts`]タブを選択し、[`Add Alert`]をクリックします。
2. `Runtime`または`Repository`のアラートタイプを選択します。

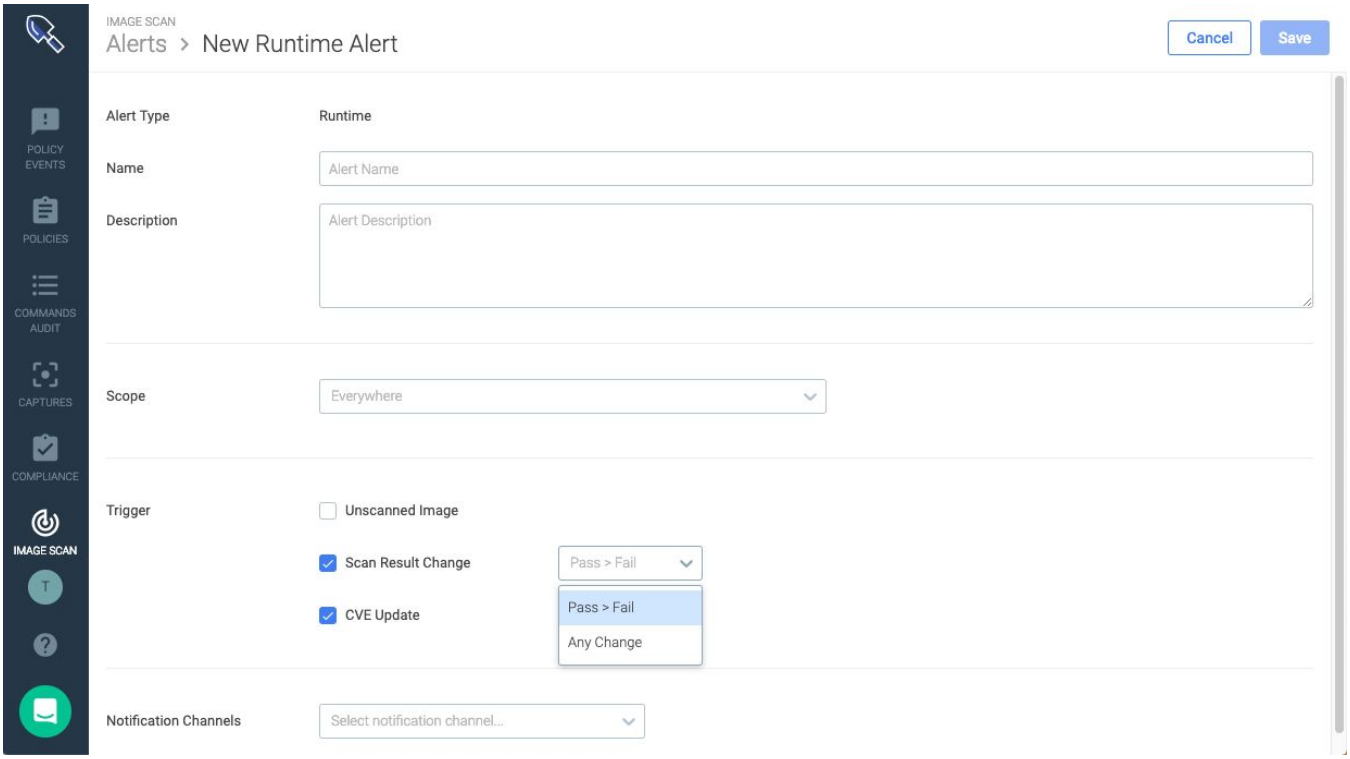


3. 適切な[New Alert]ページに入力します。

ランタイムアラートを作成する

ランタイムアラートを使用して、実行中のイメージをスキャンし、ポリシー違反、ステータス変更、またはスキャンされていないイメージが環境に追加された場合に通知をトリガーします。 アラートパ

ラメータを入力し、[`Save`]をクリックします。

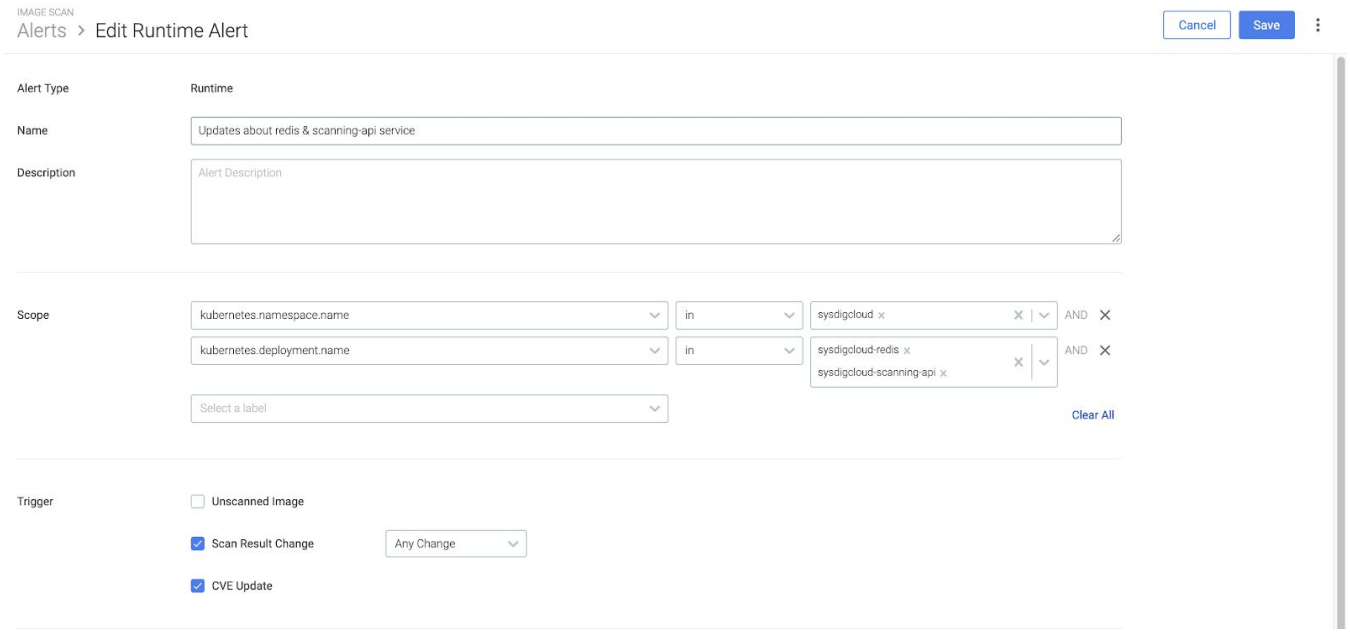


基本パラメータ

名前とオプションの説明を入力します。

Scope

`Entire Infrastructure`を使用するか、より狭いスコープを定義します。



トリガー

スキャンされていないイメージ

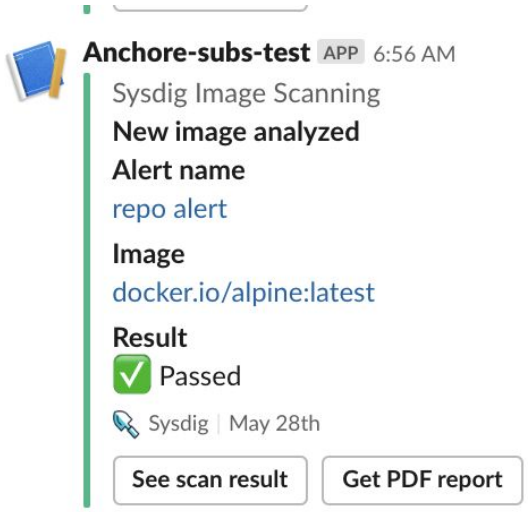
- `Scan`：ボックスをチェックし、スキャンを選択して、スコープ内で検出されたイメージを自動的にスキャンします。
  イメージのスキャンもご覧ください。
- `Alert Only`：警告を送信しますが、イメージを自動的にスキャンしません。

スキャン結果の変更

- `Pass/Fail`：以前に合格したイメージがポリシー評価に失敗したときに通知を受けるには、このオプションを選択します。
- `Any Change`：このオプションを選択すると、以前にスキャンしたイメージ結果に変更があったときに通知されます。

[スキャン結果の変更]がオンになっており、通知チャネルが構成されている場合、アラートが送信されることに注意してください。チャネルが設定されていない場合、何も起こりません。

たとえば、次のイメージは「Any Change」が設定されたときにトリガーされたSlack通知を示しています。



CVEの更新

実行中のイメージに脆弱性が追加、更新、または削除されるたびに通知されるようにするには、このオプションを選択します。

通知チャンネル

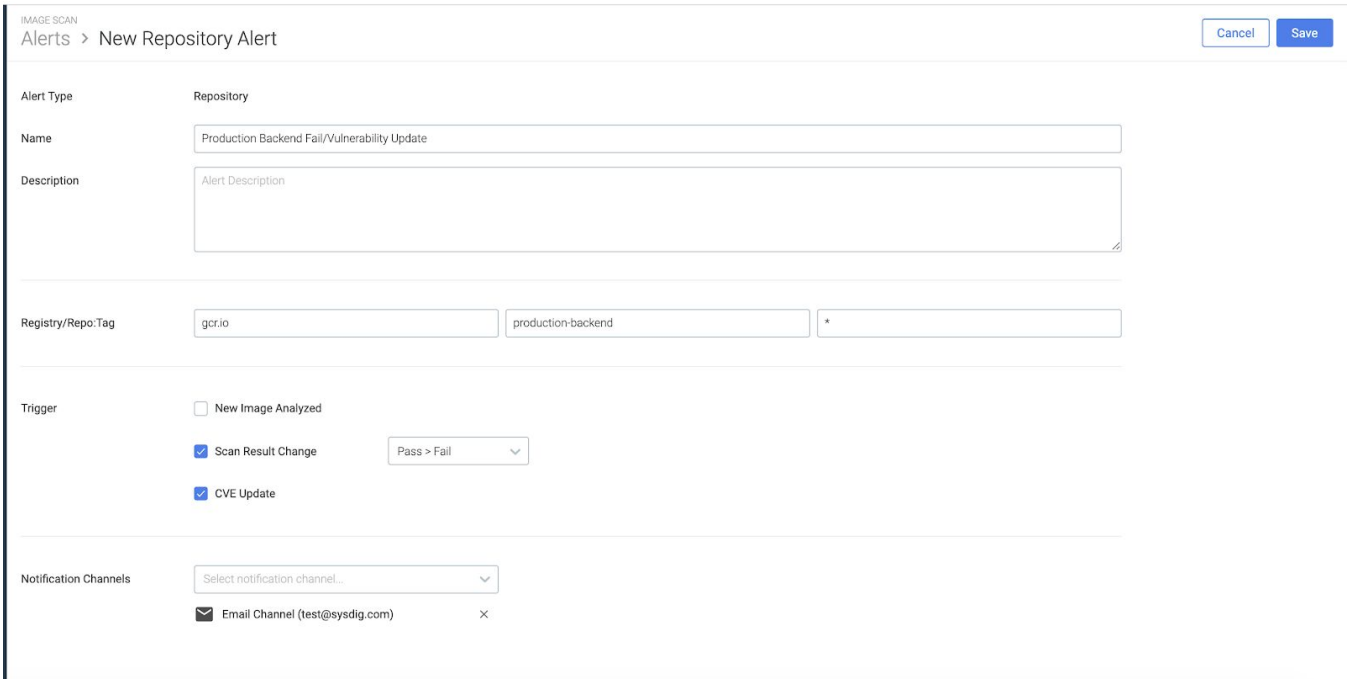
[`+ Add Channel`]をクリックして、アラート通知に使用する構成済みの通知チャネル（電子メールなど）を選択します。

Sysdig Secure環境に通知チャネルがまだ定義されていない場合は、通知チャネルのセットアップを参照してください。

リポジトリアラートを作成する

リポジトリアラートを使用して、リポジトリ内の静的イメージをスキャンし、ポリシー違反、ステータス変更、または環境に新しいイメージが追加された場合に通知をトリガーします。アラートパラ

メータを入力し、[Save]をクリックします。



基本パラメータ

名前とオプションの説明を入力します。

Registry/Repo/Tag

アラートで考慮するレジストリスコープを入力します。ワイルドカード*がサポートされています。レジストリまたはレポジトリにワイルドカードが使用されている場合、唯一のアラートオプションは[新しいイメージの分析]になります。

トリガー

分析された新しいイメージ

結果に関係なく、新しいイメージが分析されるたびに警告するボックスをオンにします。

スキャン結果の変更

- `Pass/Fail`：以前に合格したイメージがポリシー評価に失敗したときに通知を受けるには、このオプションを選択します。

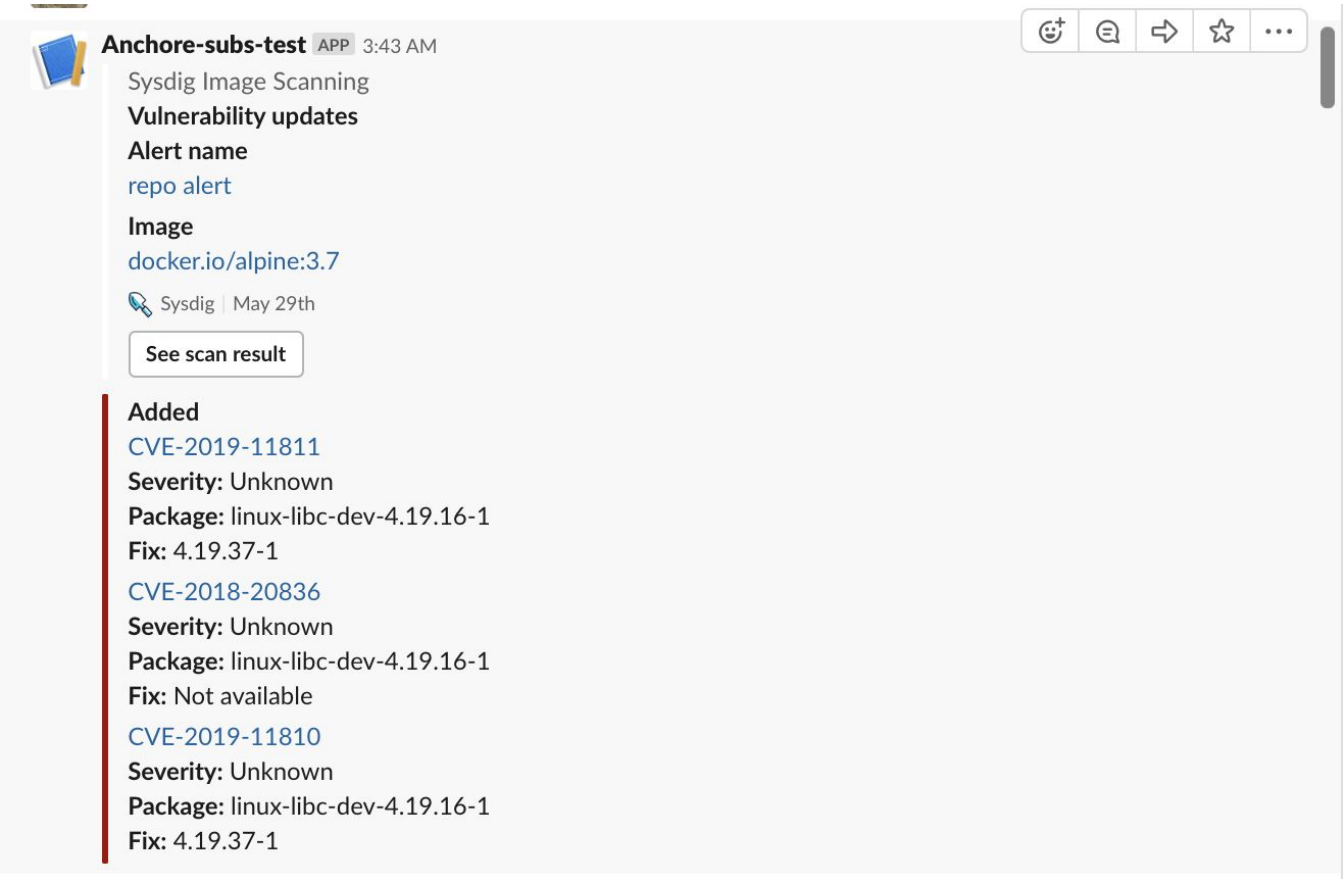
- `Any Change`：このオプションを選択すると、以前にスキャンしたイメージ結果に変更があったときに通知されます。

  [スキャン結果の変更]がオンになっており、通知チャネルが構成されている場合、アラートが送信されることに注意してください。チャネルが設定されていない場合、何も起こりません。

CVEの更新

このオプションを選択すると、リポジトリアラートスコープ内のイメージに脆弱性が追加、更新、または削除されるたびに通知されます。

たとえば、次のイメージは、「CVE Update」が設定されたときにトリガーされたSlack通知を示しています。



通知チャンネル

[`+ Add Channel`]をクリックして、アラート通知に使用する設定済みの通知チャネル（電子メールなど）を選択します。

Sysdig Secure環境に通知チャネルがまだ定義されていない場合は、通知チャネルのセットアップを参照してください。

アラートを編集する

- `Image Scanning`モジュールから、[`Alerts`]タブを選択します。
- リストから目的のアラートを選択します。
- 必要に応じてアラートトリガー、スコープ、および通知チャネルを編集し、[`Save`]をクリックします。

アラートを複製する

- `Image Scanning`モジュールから、[`Alerts`]タブを選択します。
- リストから目的のアラートを選択します。
- [`More`]（3つのドット）アイコンをクリックし、ドロップダウンから[`Duplicate Alert`]をクリックし、[はい]をクリックして確認します。

アラートを削除する

- `Image Scanning`モジュールから、[`Alerts`]タブを選択します。。
- リストから目的のアラートを選択します。
- [`More`]（3つのドット）アイコンをクリックし、ドロップダウンから[`Delete Alert`]をクリックしてから、[`Yes`]をクリックして確認します。

スキャン結果を確認する

スキャン用のビルド環境をセットアップし（該当する場合）、目的のレジストリを追加し、スキャンを手動でトリガーするか、自動的にスキャンするようにアラートを構成すると、イメージスキャンレポートが生成されます。

スキャン結果にアクセスするにはさまざまな方法があります。

- 外部（開発者向け）：Jenkinsなどの外部継続的統合（CI）ツールから。
- 内部（セキュリティ担当者向け）：Sysdig Secureのイメージスキャンモジュールの[**Runtime**]タブまたは[**Scan Results**]タブ（旧称「Repositories」）から

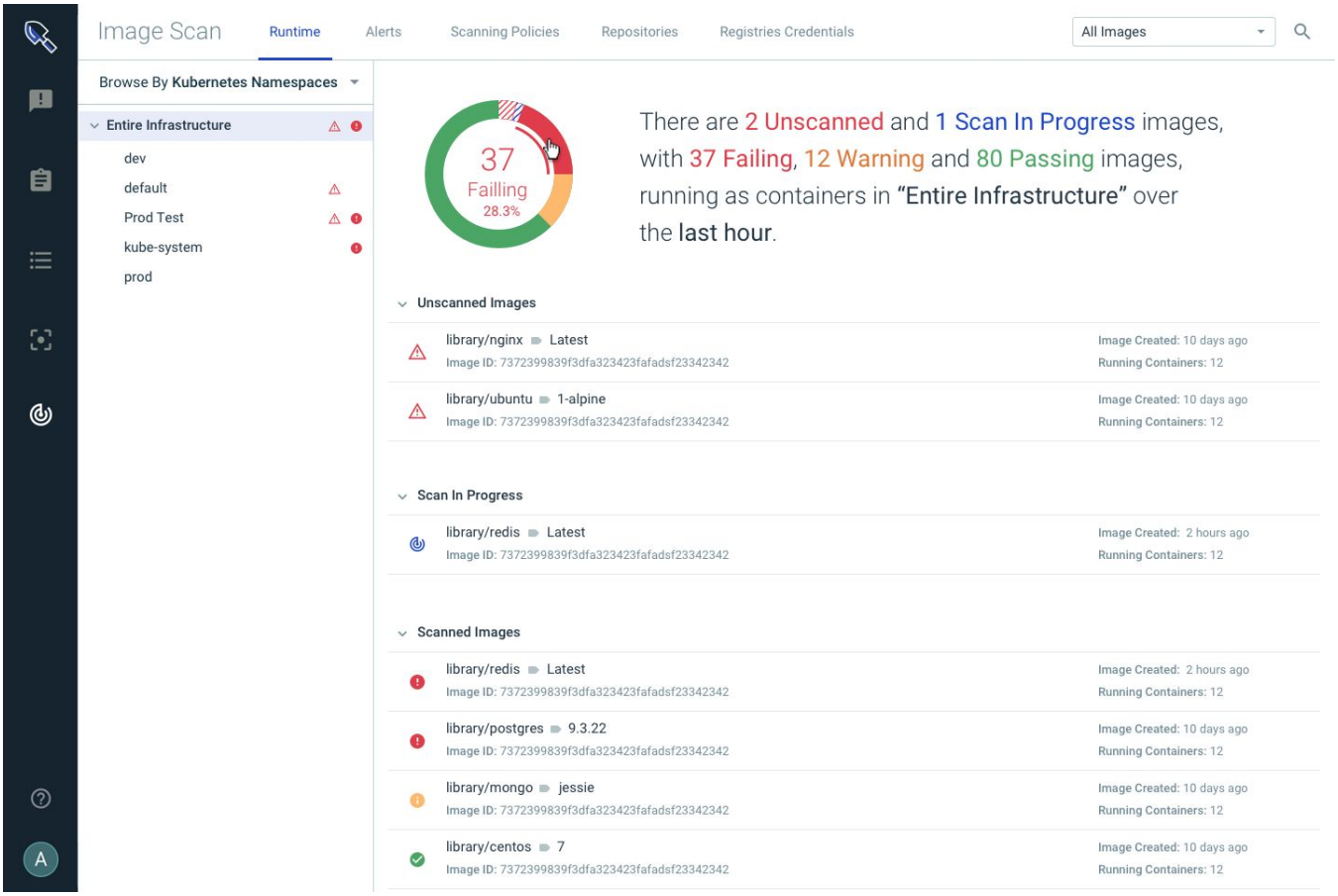
注意

データ保持設定を使用して、スキャン結果リストの管理に役立てることができます。 詳細については、データ保持制限の設定を参照してください。

（オンプレミスインストール用ではSysdig Platformバージョン3.2.0から利用可能です。）

ランタイムビュー

`Runtime`は、過去1時間に環境で実行されたイメージに関する常に更新されたレポートを提供します。



左側の列で、`Entire Infrastructure`を表示するか、ネームスペースにドリルダウンします。

右の列のレポートには、`Unscanned`イメージと`Scanned`イメージが一覧表示されます。[スキャン結果の詳細]ビューにドリルダウンできます。

スキャンされていないイメージ

スキャンを手動でトリガーするには、スキャンされていないイメージを選択します。

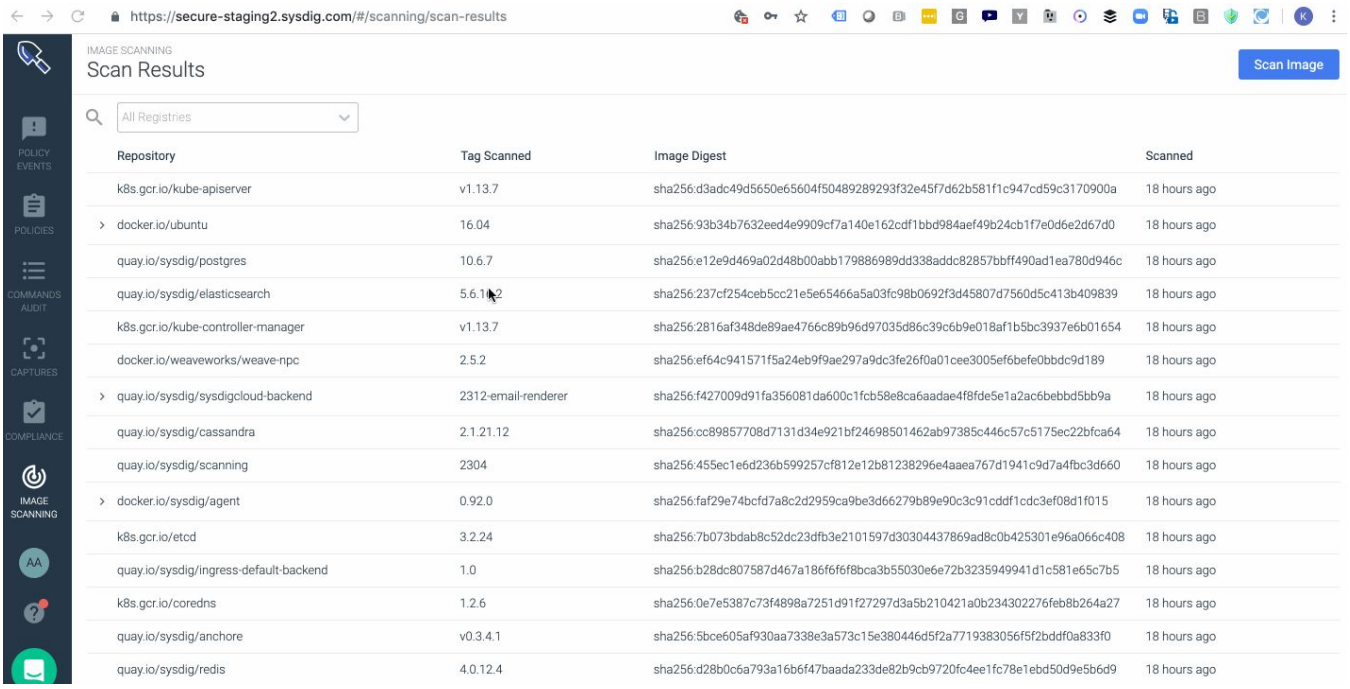
スキャンイメージ

スキャンされたイメージを選択して、詳細にドリルダウンします：`Summary`ページ、`Policy`の詳細、`Vulnerability`の詳細、`Content`違反（ライセンスなど）。

スキャン結果ビュー

`Scan Results`リストを使用して、スキャン結果のリストを表示します。

- 実行されていないものを含む特定のイメージを検索する
- イメージがデプロイされている場所に基づくフィルター
- さまざまなリポジトリを簡単に参照/展開して、評価されたimage：tagsとその結果を確認します。



リスト上のスキャンされたイメージにドリルダウンすると、スキャン結果の詳細ビューはランタイムパネルからのものと同じです。

スキャン結果の詳細

[Scan Results]リストにドリルダウンすると、詳細メニューに脆弱性とポリシー違反のデータが一目でわかるさまざまな方法が表示されます。

- ポリシー概要ビュー
- 脆弱性の概要
- コンテンツの概要

これらの要約は以下を提供します。

- 特定のイメージが失敗した理由の解析しやすいビュー
- どのルールが最も警告および停止アクションを生成したか
- 配置されたさまざまな監査ポリシーに対してイメージがどのように実行されたかの概要
- 重大度の高いCVEをフィルタリングし、利用可能な修正があるものを確認する機能

ポリシーの概要をPDFに、脆弱性の概要をCSVファイルにダウンロードすることもできます。

ポリシー結果ビュー

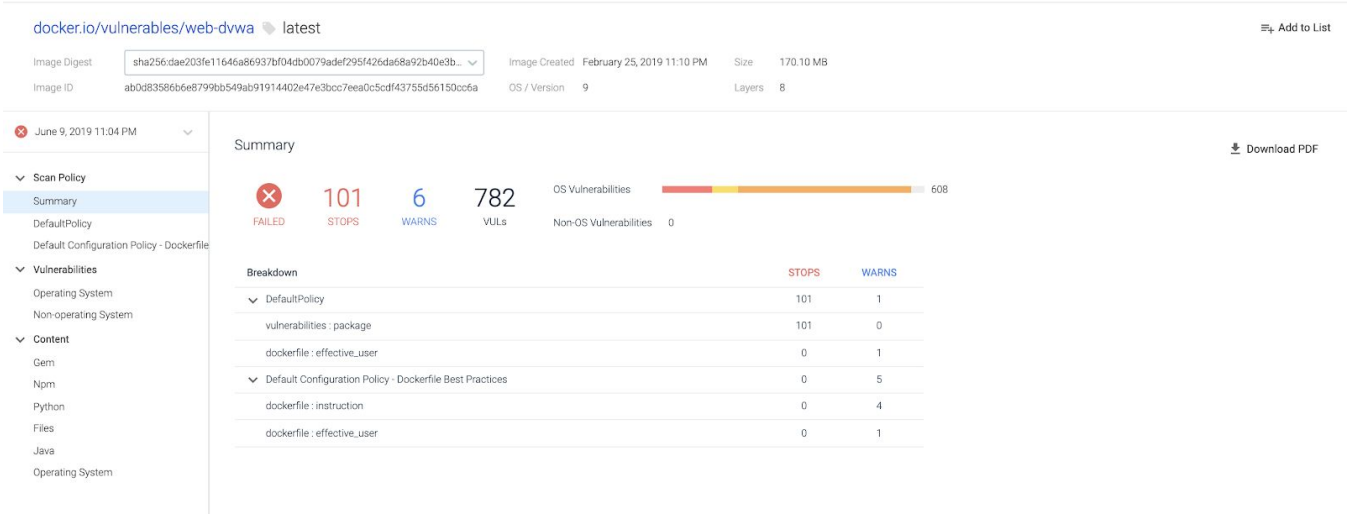
概要

スキャン結果の詳細のランディングページは、Policy Summaryビューです。

できる事：

- スキャン状態の鳥瞰図を取得する
- 別のスキャン日付を選択してください
- 詳細ページへのドリルダウン

- [Download as PDF]をクリックして、基礎となるすべてのCVEを含む完全なレポートを取得します



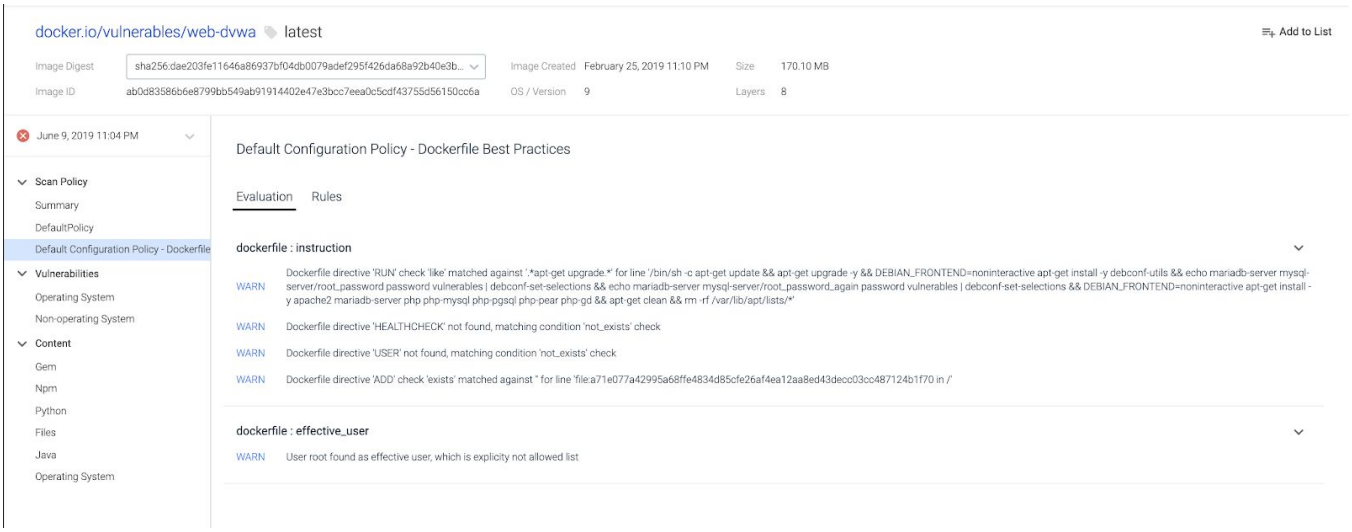
過去のスキャンの日付を選択

ドロップダウンから、分析するスキャンの日付を選択します。

スキャンポリシーの詳細を確認する

リストされたポリシーを選択して、評価でトリガーされたSTOPおよびWARNアクションの詳細を確認し、

影響を受ける基本的なルールも同様です。

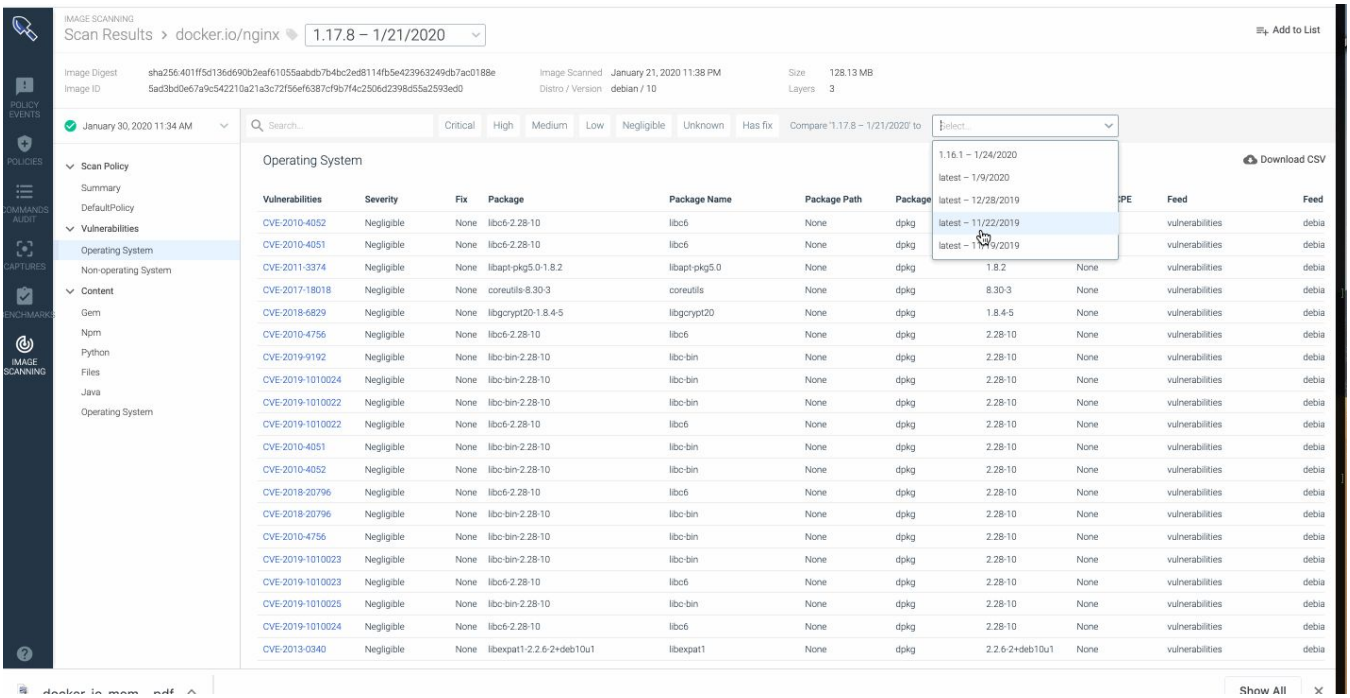


脆弱性の概要を確認する

確認するオペレーティングシステム関連または非オペレーティングシステム関連の脆弱性の概要を選択します。

できる事：

- 脆弱性ステータスの鳥瞰図を取得する
- 完全な詳細を取得するには、CVE番号をクリックします
- 重大度による検索またはフィルター
- 異なるタグを選択して、それらの間の脆弱性の比較を確認します
- [Download CSV]をクリックして、脆弱性データをCSVファイルとして取得します



脆弱性の比較

脆弱性の比較により、ユーザーは同じレポジトリ内の2つの異なるタグを比較して、バージョンXでバージョンYと比較して新しい脆弱性または修正された脆弱性を確認できます。

これにより、開発者は最新のイメージを以前のバージョンと簡単に比較して、対処された脆弱性と新規の脆弱性を簡単に報告できます。

- ページ上部のドロップダウンメニューから、タグによる脆弱性スキャン結果を選択します。
- 右側の[Compare to]ドロップダウンから別のタグを選択します。
- 比較レポートが表示され、TOTAL、NEW、FIXED、およびバージョン間の残りのSHAREDによる脆弱性が強調表示されます。 見出しをクリックして、詳細な脆弱性リストを確認します。

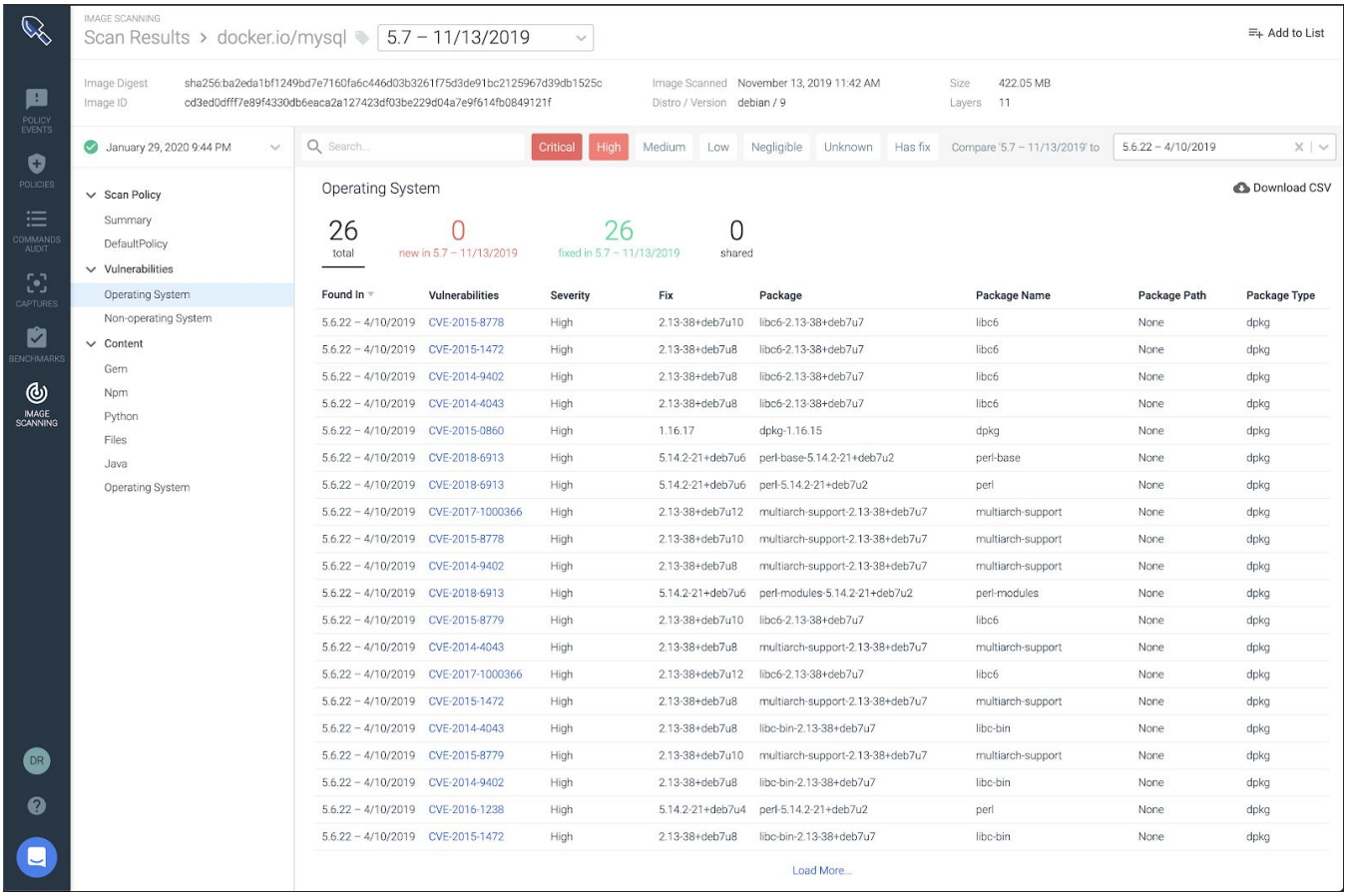
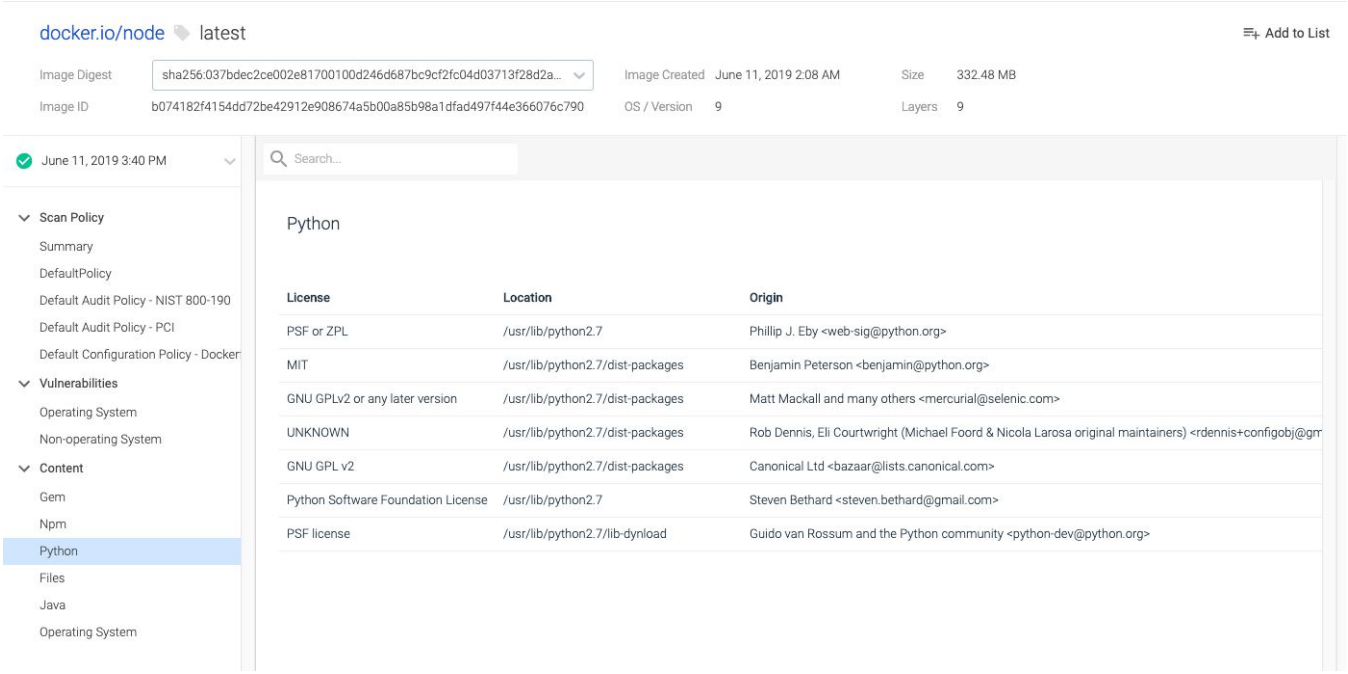


IMAGE SCANNING
Scan Results > docker.io/mysql
5.7 – 11/13/2019
Add to List
Image Digest sha256:ba2eda1bf1249bd7e7160fa6c446d03b3261f75d3de91bc2125967d39db1525c
Image ID cd3ed0dfff7e89f4330db6eaca2a127423df03be229d04a7e9f614fb0849121f
Image Scanned November 13, 2019 11:42 AM
Distro / Version debian / 9
Size 422.05 MB
Layers 11
January 29, 2020 9:44 PM
Search...
Critical
High
Medium
Low
Negligible
Unknown
Has fix
Compare '5.7 – 11/13/2019' to
5.6.22 – 4/10/2019
POLICY EVENTS
POLICIES
COMMANDS AUDIT
CAPTURES
BENCHMARKS
IMAGE SCANNING
Scan Policy
Summary
DefaultPolicy
Vulnerabilities
Operating System
Non-operating System
Content
Gem
Npm
Python
Files
Java
Operating System
Operating System
Download CSV
26 total
0 new in 5.7 – 11/13/2019
26 fixed in 5.7 – 11/13/2019
0 shared
Found In
Vulnerabilities
Severity
Fix
Package
Package Name
Package Path
Package Type
5.6.22 – 4/10/2019 CVE-2015-8778 High 2.13-38+deb7u10 libc6-2.13-38+deb7u7 libc6 None dpkg
5.6.22 – 4/10/2019 CVE-2015-1472 High 2.13-38+deb7u8 libc6-2.13-38+deb7u7 libc6 None dpkg
5.6.22 – 4/10/2019 CVE-2014-9402 High 2.13-38+deb7u8 libc6-2.13-38+deb7u7 libc6 None dpkg
5.6.22 – 4/10/2019 CVE-2014-4043 High 2.13-38+deb7u8 libc6-2.13-38+deb7u7 libc6 None dpkg
5.6.22 – 4/10/2019 CVE-2015-0860 High 1.16.17 dpkg-1.16.15 dpkg None dpkg
5.6.22 – 4/10/2019 CVE-2018-6913 High 5.14.2-21+deb7u6 perl-base-5.14.2-21+deb7u2 perl-base None dpkg
5.6.22 – 4/10/2019 CVE-2018-6913 High 5.14.2-21+deb7u6 perl-5.14.2-21+deb7u2 perl None dpkg
5.6.22 – 4/10/2019 CVE-2017-1000366 High 2.13-38+deb7u12 multiarch-support-2.13-38+deb7u7 multiarch-support None dpkg
5.6.22 – 4/10/2019 CVE-2015-8778 High 2.13-38+deb7u10 multiarch-support-2.13-38+deb7u7 multiarch-support None dpkg
5.6.22 – 4/10/2019 CVE-2014-9402 High 2.13-38+deb7u8 multiarch-support-2.13-38+deb7u7 multiarch-support None dpkg
5.6.22 – 4/10/2019 CVE-2018-6913 High 5.14.2-21+deb7u6 perl-modules-5.14.2-21+deb7u2 perl-modules None dpkg
5.6.22 – 4/10/2019 CVE-2015-8779 High 2.13-38+deb7u10 libc6-2.13-38+deb7u7 libc6 None dpkg
5.6.22 – 4/10/2019 CVE-2014-4043 High 2.13-38+deb7u8 multiarch-support-2.13-38+deb7u7 multiarch-support None dpkg
5.6.22 – 4/10/2019 CVE-2017-1000366 High 2.13-38+deb7u12 libc6-2.13-38+deb7u7 libc6 None dpkg
5.6.22 – 4/10/2019 CVE-2015-1472 High 2.13-38+deb7u8 multiarch-support-2.13-38+deb7u7 multiarch-support None dpkg
5.6.22 – 4/10/2019 CVE-2014-4043 High 2.13-38+deb7u8 libc-bin-2.13-38+deb7u7 libc-bin None dpkg
5.6.22 – 4/10/2019 CVE-2015-8779 High 2.13-38+deb7u10 multiarch-support-2.13-38+deb7u7 multiarch-support None dpkg
5.6.22 – 4/10/2019 CVE-2014-9402 High 2.13-38+deb7u8 libc-bin-2.13-38+deb7u7 libc-bin None dpkg
5.6.22 – 4/10/2019 CVE-2016-1238 High 5.14.2-21+deb7u4 perl-5.14.2-21+deb7u2 perl None dpkg
5.6.22 – 4/10/2019 CVE-2015-1472 High 2.13-38+deb7u8 libc-bin-2.13-38+deb7u7 libc-bin None dpkg
Load More...

コンテンツの詳細を確認する

node、ruby、python、java、OSパッケージ、およびコンテナー内のファイルをナビゲートして、特定のパッケージまたはファイルに関する詳細を検索します。



データ保持制限を設定する

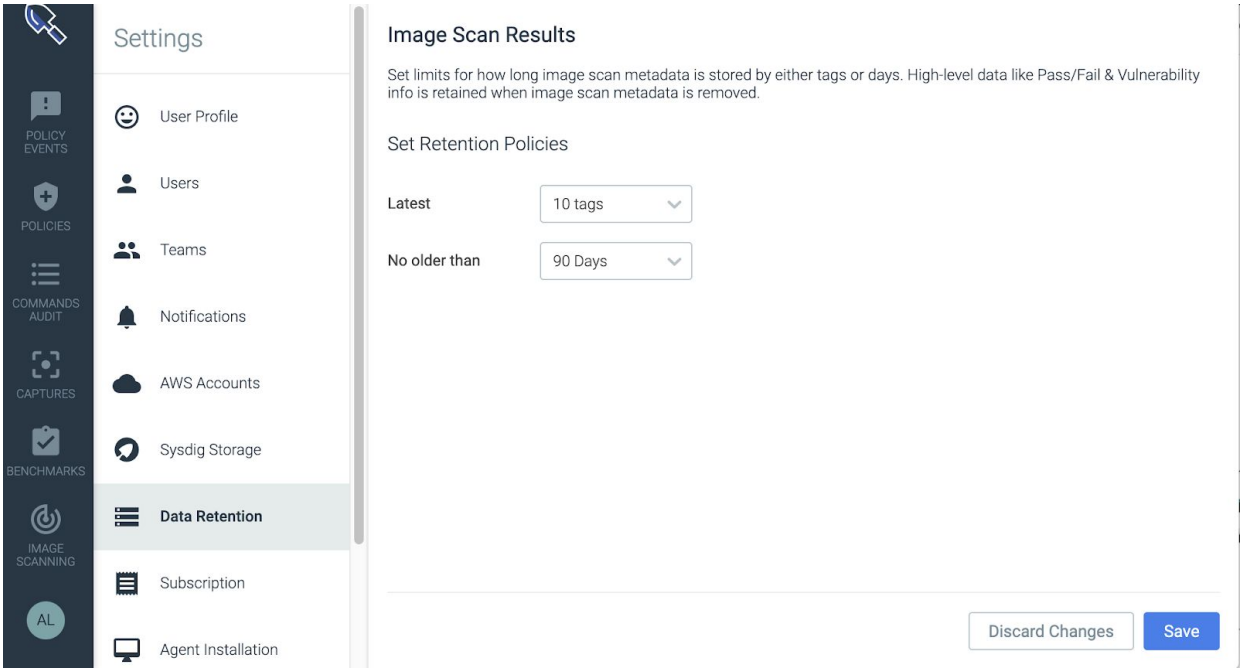
Sysdig Secure管理者は、次の目的で、イメージスキャン結果のデータ保持制限を設定できます。

- ユーザーを混乱させる古いイメージを保存しない
- Sysdigアプリケーションのパフォーマンスを改善する
- ストレージコストを削減します。

制限は、タグまたは日付で定義できます。

デフォルトのデータ保持制限を調整するには：

1. 管理者としてSysdig Secureにログインし、ナビゲーションバーのチームスイッチャーボタンから[**Settings**]メニューを選択します。
2. データ保持を選択します。



3. 制限を設定します。

- 最新：スキャン結果を保持する必要があるバージョンを（タグで）いくつ定義するか。
- 以下よりも古い：スキャン結果を保持する日数（30/60/90）を設定します。

これらの設定は、Sysdig Secure UIのスキャン結果ビューに影響します。

レポート

イメージスキャンレポート

概要

この機能を有効にするには、Sysdigサポートにお問い合わせください

レポート機能により、ユーザーは静的スコープまたはランタイムスコープに対してスキャンのコンテンツを照会し、イメージのリスク、露出、またはコンポーネントを示すレポートを生成できます。

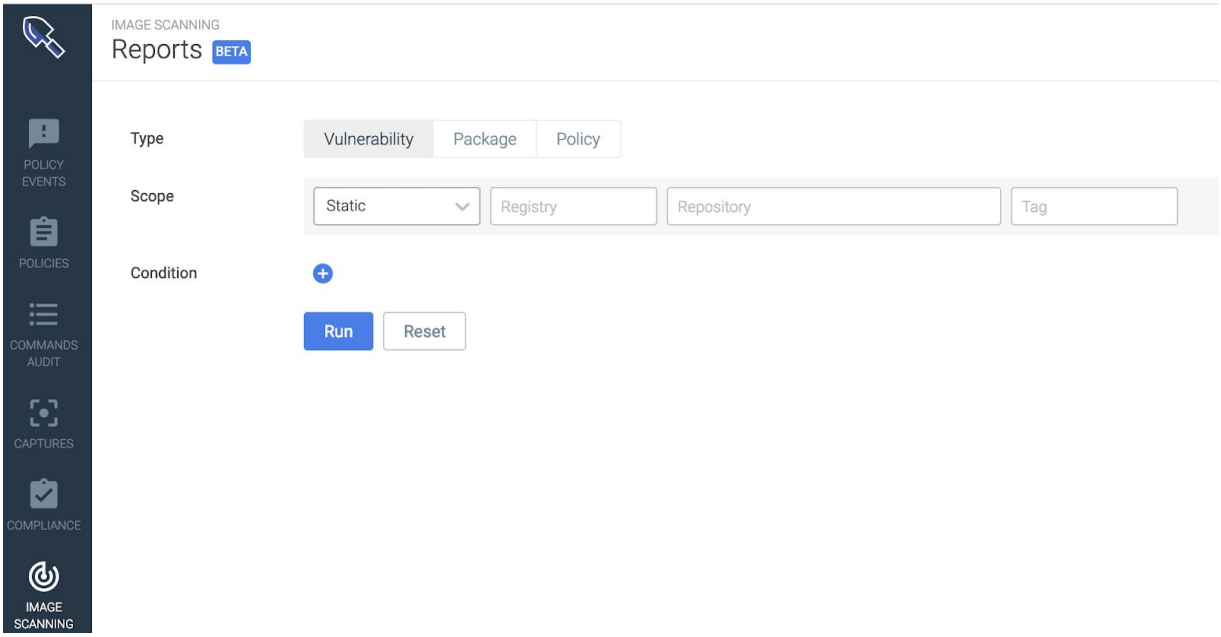
ユースケースには次のものが含まれます。

- 新しいCVEが発表されました。そのCVEにさらされている私の米国東部クラスターで実行中のすべてのイメージを見つけてみましょう
- タグprodがあり、30日以上前に修正された脆弱性があるGoogle Container Registry内のすべてのイメージを表示する
- billingネームスペースで実行されている修正を含む重大度の高い脆弱性を持つすべてのイメージを表示する

レポートを実行する

1. [**Scanning**]> [**Reports**]を選択します。

レポートインターフェイスが表示されます。



2. 適切なクエリパラメータを選択し、[Run]をクリックします。
   - **Type**：表示される列と、レポート出力のフィルタリングに使用できる条件を変更します
     - 脆弱性脆弱性ID、重大度、修正、パッケージ名などに基づいて脆弱性のリストを取得します。
     - パッケージ
     - ポリシー
   - **Scope**：このレポートの一部で照会されているイメージ
     - **Static**：レジストリコンテキストに基づいてイメージを評価します。 サンプルProdイメージポリシーで「Prod」タグを持つすべてのイメージを評価するには、次の割り当て（registry/repo/tag）を使用します。*/*/Prod
   - **Runtime**：ランタイムコンテナ、クラウドプロバイダー、Kubernetesなどのオーケストレーターから公開されたラベルに基づいてイメージを評価します
   - **Condition**：レポート結果をさらにフィルター処理して、意味のある結果を生成する方法。クエリタイプの内訳ごとに詳細が表示されます。
3. オプション：[Download CSV]をクリックして、レポートをキャプチャします。

注意

レポートを正常に生成するには、少なくとも1つの条件を選択するか、リポジトリスコープを追加する必要があります。

Query by Vulnerability

このレポートは、静的またはランタイムスコープ内のイメージ内のパッケージにマッピングされた脆弱性の行を返します。下のイメージでは、特定の脆弱性（CVE-2017-8831）を検索すると、CVEを含む環境でアクティブに実行されている2つのイメージが表示されます 。

利用可能な条件フィールドは次のとおりです。

- Vuln ID
- Severity
- Fix Available?
- Package Name
- Package Version
- Age

Query by Package

このレポートには、パッケージのバージョンがある環境でアクティブに実行されているすべてのイメージが表示されます。 また、複数のイメージが同じパッケージ名バージョンを実行しているかどうか、および関連するCVEがあるかどうかも示します。

利用可能な条件フィールドは次のとおりです。

- Package Name
- Package Version

Query by Policy

ポリシーレポートには、発生したすべてのポリシー評価、成功したか失敗したか、およびイメージが成功したか失敗した可能性がある理由が表示されます。 合格または不合格の理由には、ホワイトリスト、ブラックリスト、または単に標準的なポリシー評価が含まれます。

- Evaluation Results (Pass/Fail)
- Reason (Whitelist, Blacklist, Error, Policy Evaluation)
- Age