Creating Xray Policies and Rules

Overview

Policies define security and license compliance behavior specifications. Policies enable you to create a set of rules, in which each rule defines a license/security criteria, with a corresponding set of automatic actions according to your needs. Policies are enforced when applying them to Watches. A policy is contextless, which means that it only defines what to enforce and not what to enforce it on.

Separating the behavior you want to enforce from the context you want to enforce it on provides you with the following values:

- **Efficiency**. Reduce work and save time by configuring your policies once and assigning them to multiple watches.
- **Flexibility**. Configure multiple behaviours with additional functionality such as priority of your security rules.
- **Separate Concerns**. Delegate permissions to different teams in your organization. Everything related to resources and filters is in the watch, and everything related to security and license compliance is in policies.

Starting from Xray 3.21.2, the Watches configuration has been moved from the Application Module to the Administration Module in the JFrog Platform UI.

Page Contents

Triggering Violations Using Policy Rules

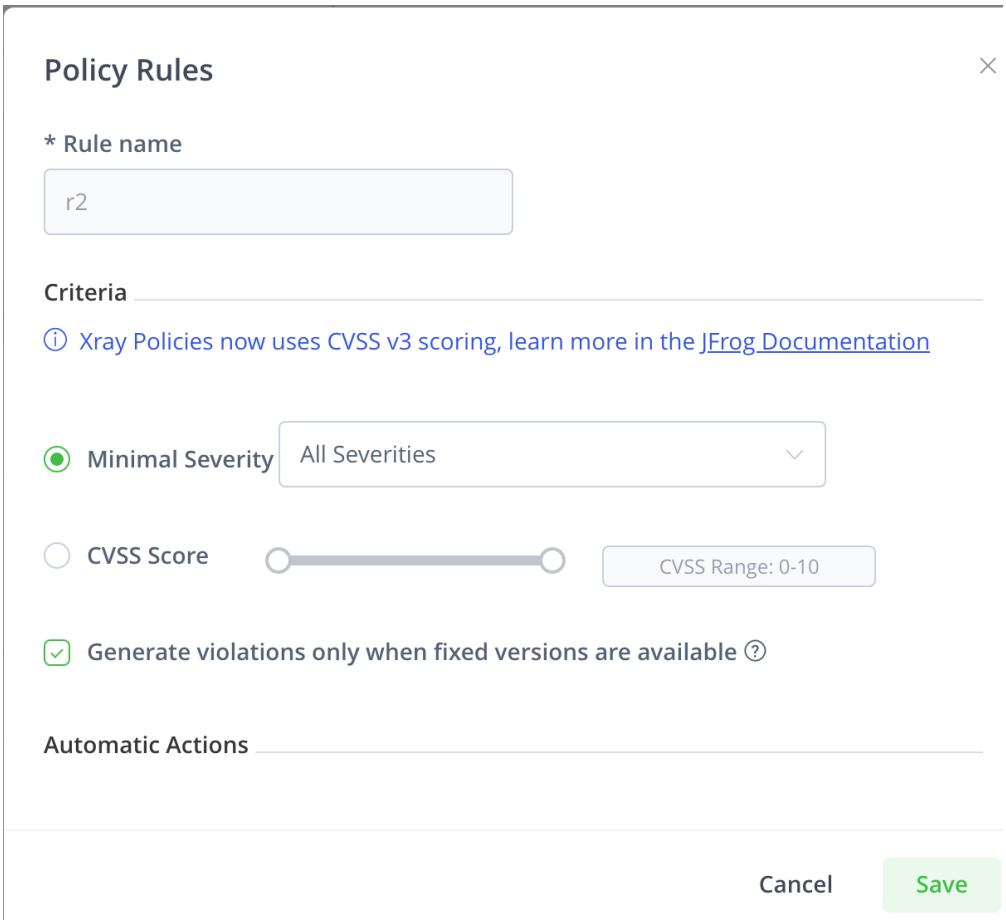
Policies contain user-defined rules allowing you to trigger violations for specific vulnerability or license breaches by setting a license or security criteria, with a corresponding set of automatic actions according to your needs. Rules are processed according to the ascending order in which they are placed in the Rules list on the Policy. If a rule is met, the subsequent rules in the list will not be applied.

Xray supports the following policy types:

Security Rules

A Security Rule allows you to create a set of rules around security vulnerabilities. There are two possible criteria:

1. **Minimal Severity** (Minor, Major, Critical, All): The minimal security vulnerability severity as it is in the JFrog vulnerabilities database. If the artifact or build contains a vulnerability with the selected severity or higher, the rule will meet the criteria, the automatic actions will be executed, and the policy will stop processing.
2. **CVSS Score** (1-10): The CVSS score range to apply to the rule. This is used for a fine-grained control, rather than using the predefined severities. The score range is based on CVSS v3 scoring, and CVSS v2 score is CVSS v3 score is not available.
3. **Generate violations only when fixed versions are available**: Xray will not generate violations for issues that do not contain a fixed version. If a fixed version is available later, the violation will be generated.



License Rules

A licence Rule allows you to create a set of rules around license compliance. There are three possible criteria:

- **Allowed Licenses**: Specifies an Allow List of OSS licenses that **may** be attached to a component. If a component has an OSS license outside the specified Allow List, The rule will meet the criteria, a violation will be generated, automatic actions will be executed, and the policy will stop processing.
- **Banned Licenses**: Specifies a Block List of OSS licenses that **may not** be attached to a component. If a component has any of the OSS licenses specified, The rule will meet the criteria, a violation will be generated, automatic actions will be executed, and the policy will stop processing.
- **Disallow Unknown License**: Specifies the wanted behavior for components whose license cannot be determined. A violation will be triggered if a component with unknown license is found.

Policy Violation Automatic Actions

An action determines the automatic response to a detected Policy violation. You can define one or more action within each Policy Rule. Actions include the following:

- **Generate Violation (Minor, Major, Critical)**: The severity of the violations that is generated if the criteria is met.
- **Notify Email:** This action lets you specify email addresses to which Xray should send an email message about a violation when one is triggered. For this to work, you need to have a mail server configured in Xray.
- **Notify Watch's Recipients:** This action lets you send an email to all the watch recipients about a violation when triggered.
- **Notify Deployer:** This action lets you send an email to the user that deployed the component about a violation when triggered.
- **Create Jira Ticket**: This action enables the Jira ticket creation in the Policy rules
- **Trigger Webhook:** This action lets you specify webhooks you have configured in Xray that should be invoked when a violation is triggered (See payload below).
- **Block Download:** This action lets you specify that artifacts should be blocked for download and allows you to select one of these options:
    - Block Download: When set, Artifactory will block download of artifacts that meet the Artifact Filter and Severity Filter specifications for this watch.
    - Block Unscanned: When set, Artifactory will block download of artifacts that meet the Artifact Filter specifications for this watch, but have not been scanned yet or the Xray data has been removed due to a retention policy. For more information on Xray Data Retention, see Indexing Xray Resources.
- **Block Release Bundle distribution:** This action lets you specify that Release Bundles should be blocked for download if they meet the policy criteria rule.

- **Fail Build:** This action lets you specify that if a CI server requests a build to be scanned, and the Watch triggers a violation, Xray will respond with an indication that the build job should fail.
  This action is only available if the Watch is defined with Builds as target type.

    - **Grace Period**: There are many cases where you do not want to fail the first build, for example, some violations are not showstoppers, and you would like to look into them later without stopping the build creation. You can set a grace period for a number of days that you define according to your needs. During the grace period you define, the build will not fail and all violations are ignored during this period. An automatic Ignore Rule is created for the grace period with the following criteria:
        - On the specific vulnerability/license
        - On the specific component
        - On any version of the specific build
        - On the specific Policy
        - On the specific Watch

      Once the grace period ends, the Ignore Rule is deleted, and if the build contains violations, those violations are created and the build will fail.

Creating a Policy

Step 1 Setting the General Policy Settings

1. In the **Administration** module, select **Watches & Policies** and from the **Policies** tab click **New Policy**.
2. Select the policy rule type and configure the rule.

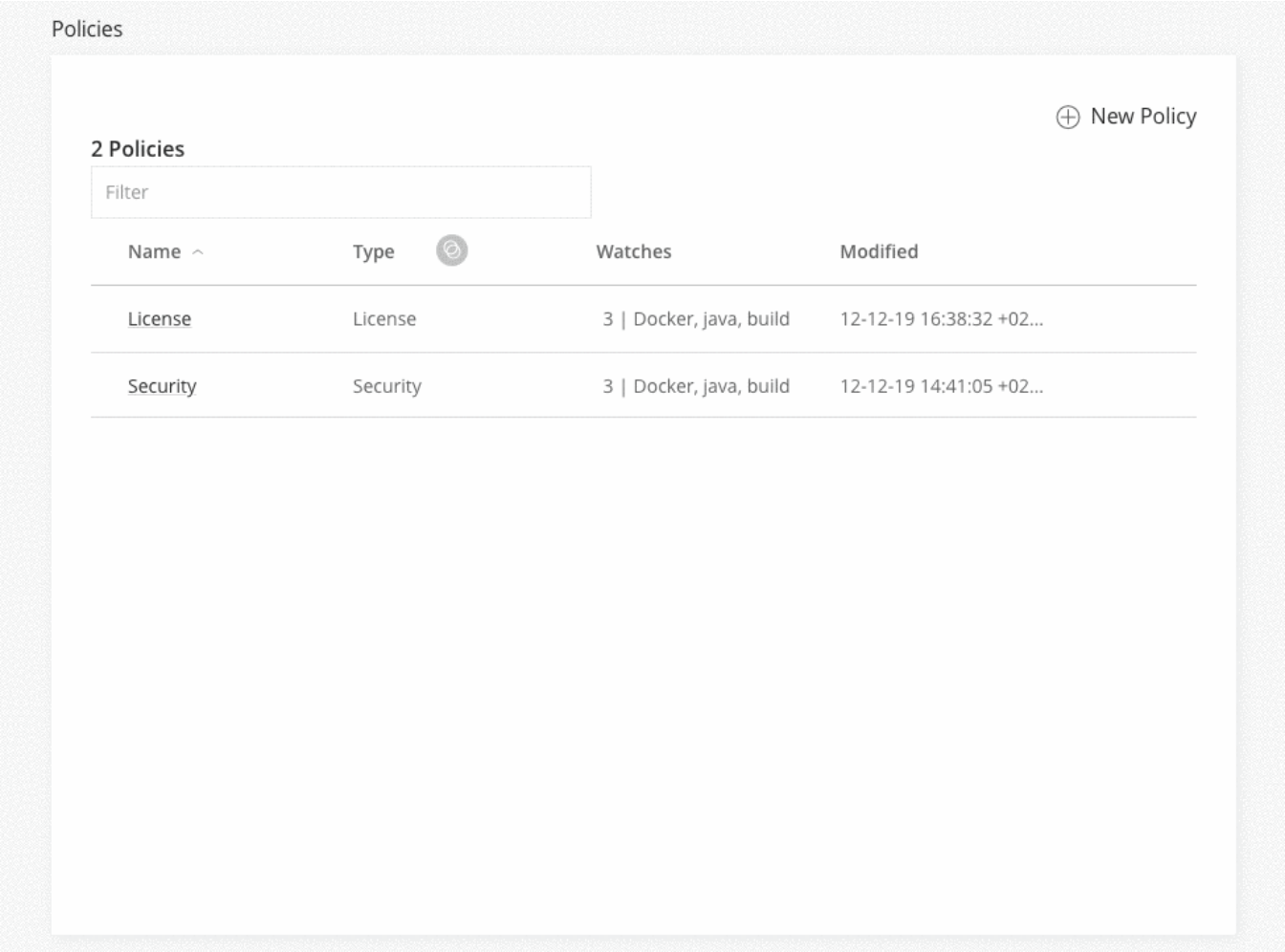
    - **Security**: Lets you create a set of rules around security vulnerabilities. Choose how you want Xray to respond to each vulnerability severity.
    - **License**: Lets you create a set of rules around allowed/banned sets of licenses.
    - **Operational Risk**: Lets you create a set of rules about the operational risk of using open source software components. For more information, see Components Operational Risk.
3. Set the priority in which rules are processed. Drag and drop the rules to place them according to their priority.
4. Set the Rule Criteria.
   If the criteria is met, then the automatic actions of this rule are executed and the policy is considered as processed (no further rules will be checked).
5. Set automatic actions to run if a criteria is met.

Step 2 Creating a Policy Rule

Configure the the basic policy settings and select your policy type - Security or License.

Configure a Security Rule

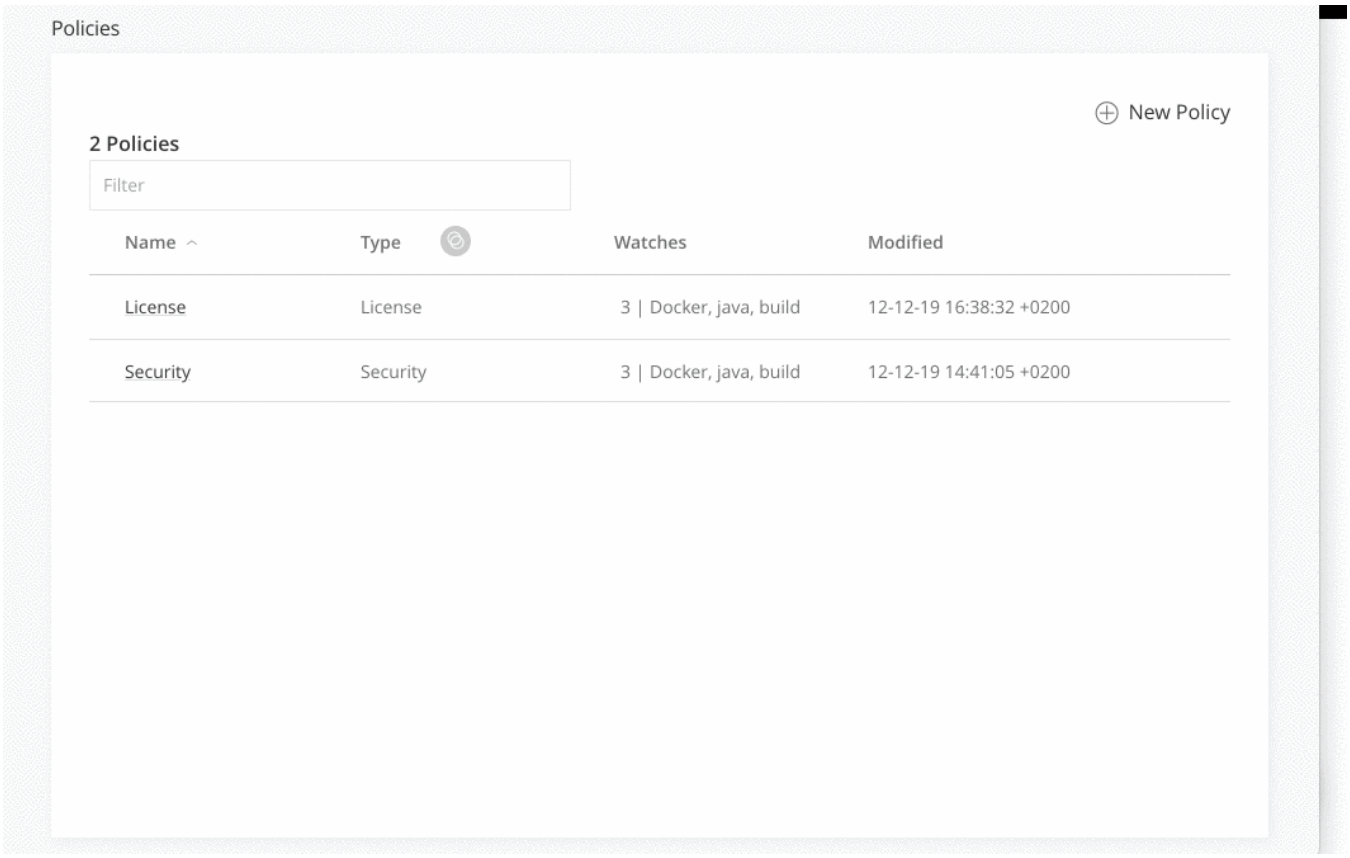
Select **Security** from the drop-down list and click **New Rule** to set criteria and assign automatic actions.



Rule Name	A logical name for this Rule.
Criteria	The set of security conditions to examine when an scanned artifact is scanned.
Automatic Actions	Specifies the actions to take once a security policy violation has been triggered.

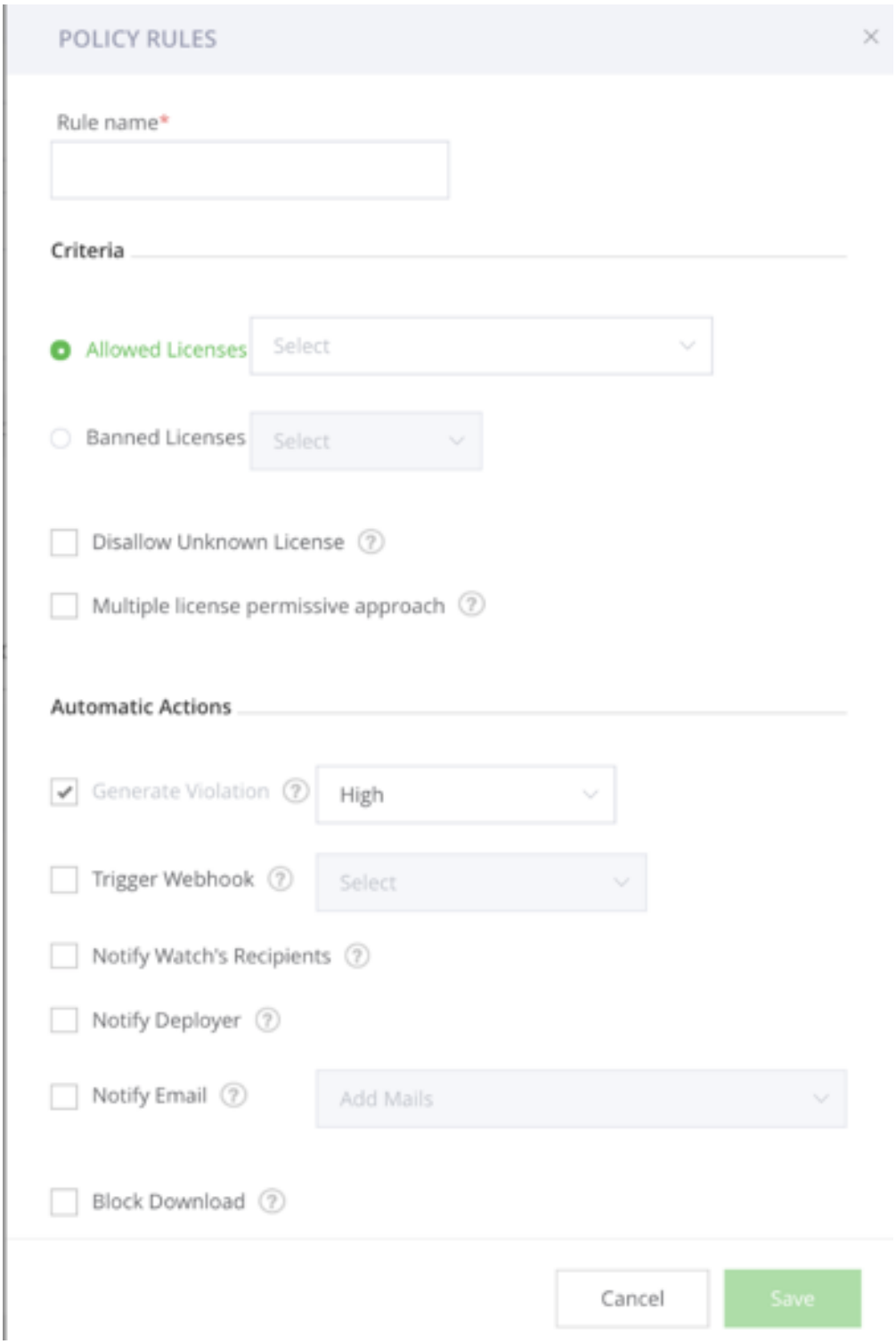
Configure a License Rule

To create a new License Rule, select **License** from the drop-down list and click **New Rule**.



Assign an Automatic Action to a Policy Rule

You can define one or more actions within each Policy Rule. To view a list of actions, see Automatic Actions.



Block Unscanned Artifacts

This configuration will block unscanned artifact download requests. The download timeout should be set by your system administrator.

Multiple License Permissive Approach

When a component is detected with multiple licenses, the policy rules apply on all of the licenses, thus if one of the multiple licenses meets the policy rule, a violation will be created anyways. The multiple license permissive approach enables you to have more flexibility in the policy level and to configure a more permissive approach that allows components that have at least one of the licenses as permitted to go through without triggering a violation even if some licenses are not allowed.

Triggering a Webhook

You can select a predefined Webhook as an automatic action in case a violations is found.

- Select the **Trigger Webhook** checkbox and select predefined Webhook from the list.

The payload provided to any triggered webhook is a JSON object describing a list of Alerts with the following format:

```
{
  "created": "<Alert creation time stamp in ISO8601 (yyyy-MM-dd'T'HH:mm:ss.SSSZ)>",
  "top_severity": "<Top severity of any issue in the alert>",
  "watch_name": "<Logical name for the watch>",
  "issues": [
        {
                "severity": "<Issue severity>",
            "type": "<Issue type>",
            "provider": "<Issue provider>",
            "created": "<Issue creation time stamp in ISO8601 (yyyy-MM-dd'T'HH:mm:ss.SSSZ)>",
            "summary": "<Issue summary>",
              "description": "<Issue description>",
              "impacted_artifacts": [
                        {
                              "name": "<Artifact name>",
                          "display_name": "<Artifact display name>",
                          "path": "<Artifact path in Artifactory>",
                          "pkg_type": "<Package type>",
                          "sha256": "<Artifact SHA 256 checksum>",
                          "sha1": "<Artifact SHA 1 checksum>",
                          "depth": <Artifact depth in its hierarchy>,
                         "parent_sha": "<Parent artifact SHA 1 checksum>",
                                 "infected_files": [
                            {
                                      "name": "<File name>",
                                      "path": "<File path>",
                                      "sha256": "<File SHA 256 checksum>",
                                      "depth": <File depth in hierarchy>,
                                      "parent_sha": "<File's parent SHA 1 checksum>",
                                      "display_name": "<File's display name>",
                                      "pkg_type": "File's package type"
                            }
                                ]
                        }
                ]
        }
  ]
}
```

The following shows an example payload for a webhook.

```
{
  "created": "<Alert creation time stamp in ISO8601 (yyyy-MM-dd'T'HH:mm:ss.SSSZ)>",
  "top_severity": "<Top severity of any issue in the alert>",
  "watch_name": "<Logical name for the watch>",
  "issues": [
        {
                "severity": "<Issue severity>",
            "type": "<Issue type>",
            "provider": "<Issue provider>",
            "created": "<Issue creation time stamp in ISO8601 (yyyy-MM-dd'T'HH:mm:ss.SSSZ)>",
            "summary": "<Issue summary>",
              "description": "<Issue description>",
              "impacted_artifacts": [
                        {
                              "name": "<Artifact name>",
                          "display_name": "<Artifact display name>",
                          "path": "<Artifact path in Artifactory>",
                          "pkg_type": "<Package type>",
                          "sha256": "<Artifact SHA 256 checksum>",
                          "sha1": "<Artifact SHA 1 checksum>",
                          "depth": <Artifact depth in its hierarchy>,
                         "parent_sha": "<Parent artifact SHA 1 checksum>",
                                 "infected_files": [
                            {
                                      "name": "<File name>",
                                      "path": "<File path>",
                                      "sha256": "<File SHA 256 checksum>",
                                      "depth": <File depth in hierarchy>,
                                      "parent_sha": "<File's parent SHA 1 checksum>",
                                      "display_name": "<File's display name>",
                                      "pkg_type": "File's package type"
                            }
                                ]
                        }
                ]
        }
  ]
}
```

Editing a Policy

Edit an existing Policy, from the Policy page, by hovering over it and clicking on the Edit icon on the right.

Edits made to a policy will automatically be applied to all watches the policy is assigned to. This will take affect only for newly scanned artifacts. You can manually activate the watch on existing artifacts.