OpenShift Dedicated 4

セキュリティーおよびコンプライアンス

OpenShift Dedicated での Security Context Constraints の設定

Last Updated: 2024-04-12

OpenShift Dedicated 4 セキュリティーおよびコンプライアンス

OpenShift Dedicated での Security Context Constraints の設定

法律上の通知

概要

このドキュメントでは、OpenShift Dedicated でセキュリティーコンテキストの制約を設定する手順を説明します。

目次

第1章 監査ログ

OpenShift Dedicated 監査は、システムに影響を与えた一連のアクティビティーを個別のユーザー、管理者、またはその他システムのコンポーネント別に記述したセキュリティー関連の時系列のレコードを提供します。

1.1. API の監査ログについて

監査は API サーバーレベルで実行され、サーバーに送られるすべての要求をログに記録します。それぞれの監査ログには、以下の情報が含まれます。

表1.1 監査ログフィールド

フィールド	説明
level	イベントが生成された監査レベル。
auditID	要求ごとに生成される一意の監査 ID。
stage	このイベントインスタンスの生成時の要求処理のステージ。
requestURI	クライアントによってサーバーに送信される要求 URI。
verb	要求に関連付けられる Kubernetes の動詞。リソース以外の要求の場合、これは小文字の HTTP メソッドになります。
user	認証されたユーザーの情報。
impersonatedUser	オプション。偽装ユーザーの情報 (要求で別のユーザーを偽装する場合)。
sourceIPs	オプション。要求の送信元および中間プロキシーからのソース IP。
userAgent	オプション。クライアントが報告するユーザーエージェントの文字列。ユーザーエージェントはクライアントによって提供されており、信頼できないことに注意してください。
objectRef	オプション。この要求のターゲットとなっているオブジェクト参照。これは、**List** タイプの要求やリソース以外の要求には適用されません。
responseStatus	オプション。**ResponseObject** が **Status** タイプでなくても設定される応答ステータス。正常な応答の場合、これにはコードのみが含まれます。ステータス以外のタイプのエラー応答の場合、これにはエラーメッセージが自動的に設定されます。

フィールド	説明
requestObject	オプション。JSON 形式の要求からの API オブジェクト。**RequestObject** は、バージョンの変換、デフォルト設定、受付またはマージの前に要求の場合のように記録されます (JSON として再エンコードされる可能性がある)。これは外部のバージョン付けされたオブジェクトタイプであり、それ自体では有効なオブジェクトではない可能性があります。これはリソース以外の要求の場合には省略され、要求レベル以上でのみログに記録されます。
responseObject	オプション。JSON 形式の応答で返される API オブジェクト。**ResponseObject** は外部タイプへの変換後に記録され、JSON としてシリアライズされます。これはリソース以外の要求の場合には省略され、応答レベルでのみログに記録されます。
requestReceivedTimestamp	要求が API サーバーに到達した時間。
stageTimestamp	要求が現在の監査ステージに達した時間。
annotations	オプション。監査イベントと共に保存される構造化されていないキーと値のマップ。これは、認証、認可、受付プラグインなど、要求提供チェーンで呼び出されるプラグインによって設定される可能性があります。これらのアノテーションは監査イベント用のもので、送信されたオブジェクトの **metadata.annotations** に対応しないことに注意してください。キーは、名前の競合が発生しないように通知コンポーネントを一意に識別する必要があります (例: **podsecuritypolicy.admission.k8s.io/policy**)。値は短くする必要があります。アノテーションはメタデータレベルに含まれます。

Kubernetes API サーバーの出力例:


```
{"kind":"Event","apiVersion":"audit.k8s.io/v1","level":"Metadata","auditID":"ad209ce1-fec7-4130-8192-c4cc63f1d8cd","stage":"ResponseComplete","requestURI":"/api/v1/namespaces/openshift-kube-controller-manager/configmaps/cert-recovery-controller-lock?timeout=35s","verb":"update","user":{"username":"system:serviceaccount:openshift-kube-controller-manager:localhost-recovery-client","uid":"dd4997e3-d565-4e37-80f8-7fc122ccd785","groups":["system:serviceaccounts","system:serviceaccounts:openshift-kube-controller-manager","system:authenticated"]},"sourceIPs":["::1"],"userAgent":"cluster-kube-controller-manager-operator/v0.0.0 (linux/amd64) kubernetes/$Format","objectRef":{"resource":"configmaps","namespace":"openshift-kube-controller-manager","name":"cert-recovery-controller-lock","uid":"5c57190b-6993-425d-8101-8337e48c7548","apiVersion":"v1","resourceVersion":"574307"},"responseStatus":{"metadata":{},"code":200},"requestReceivedTimestamp":"2020-04-02T08:27:20.200962Z","stageTimestamp":"2020-04-02T08:27:20.206710Z","annotations":{"authorization.k8s.io/decision":"allow","authorization.k8s.io/reason":"RBAC: allowed by ClusterRoleBinding \"system:openshift:operator:kube-controller-manager-recovery\" of ClusterRole \"cluster-admin\" to ServiceAccount \"localhost-recovery-client/openshift-kube-controller-manager\""}}
```


1.2. 監査ログの収集

must-gather ツールを使用して、クラスターをデバッグするための監査ログを収集できます。このログは、確認したり、Red Hat サポートに送信したりできます。

手順

1. **-- /usr/bin/gather_audit_logs** を指定して **oc adm must-gather** コマンドを実行します。

```
$ oc adm must-gather -- /usr/bin/gather_audit_logs
```

2. 作業ディレクトリーに作成された **must-gather** ディレクトリーから圧縮ファイルを作成します。たとえば、Linux オペレーティングシステムを使用するコンピューターで以下のコマンドを実行します。

```
$ tar cvaf must-gather.tar.gz must-gather.local.472290403699006248 1
```

1 **must-gather-local.472290403699006248** は、実際のディレクトリー名に置き換えます。

3. Red Hat カスタマーポータルの **カスタマーサポート** ページ で、圧縮ファイルをサポートケースに添付します。