OpenShift Dedicated 4

Security and compliance

Configuring security context constraints in OpenShift Dedicated
OpenShift Dedicated 4 Security and compliance

Configuring security context constraints in OpenShift Dedicated
Abstract

This document provides instructions for configuring security context constraints in OpenShift Dedicated.
# Table of Contents

- **CHAPTER 1. AUDIT LOGS** .................................................. 3
  - 1.1. ABOUT THE API AUDIT LOG ........................................ 3
  - 1.2. GATHERING AUDIT LOGS .......................................... 4
CHAPTER 1. AUDIT LOGS

OpenShift Dedicated auditing provides a security-relevant chronological set of records documenting the sequence of activities that have affected the system by individual users, administrators, or other components of the system.

1.1. ABOUT THE API AUDIT LOG

Audit works at the API server level, logging all requests coming to the server. Each audit log contains the following information:

Table 1.1. Audit log fields

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>level</td>
<td>The audit level at which the event was generated.</td>
</tr>
<tr>
<td>auditID</td>
<td>A unique audit ID, generated for each request.</td>
</tr>
<tr>
<td>stage</td>
<td>The stage of the request handling when this event instance was generated.</td>
</tr>
<tr>
<td>requestURI</td>
<td>The request URI as sent by the client to a server.</td>
</tr>
<tr>
<td>verb</td>
<td>The Kubernetes verb associated with the request. For non-resource requests,</td>
</tr>
<tr>
<td></td>
<td>this is the lowercase HTTP method.</td>
</tr>
<tr>
<td>user</td>
<td>The authenticated user information.</td>
</tr>
<tr>
<td>impersonatedUser</td>
<td>Optional. The impersonated user information, if the request is</td>
</tr>
<tr>
<td></td>
<td>impersonating another user.</td>
</tr>
<tr>
<td>sourceIPs</td>
<td>Optional. The source IPs, from where the request originated and any</td>
</tr>
<tr>
<td></td>
<td>intermediate proxies.</td>
</tr>
<tr>
<td>userAgent</td>
<td>Optional. The user agent string reported by the client. Note that the user</td>
</tr>
<tr>
<td></td>
<td>agent is provided by the client, and must not be trusted.</td>
</tr>
<tr>
<td>objectRef</td>
<td>Optional. The object reference this request is targeted at. This does not</td>
</tr>
<tr>
<td></td>
<td>apply for List-type requests, or non-resource requests.</td>
</tr>
<tr>
<td>responseStatus</td>
<td>Optional. The response status, populated even when the ResponseObject is</td>
</tr>
<tr>
<td></td>
<td>not a Status type. For successful responses, this will only include the code.</td>
</tr>
<tr>
<td></td>
<td>For non-status type error responses, this will be auto-populated with the</td>
</tr>
<tr>
<td></td>
<td>error message.</td>
</tr>
</tbody>
</table>
1.2. GATHERING AUDIT LOGS

Example output for the Kubernetes API server:

```json
  "uid": "dd4997e3-d565-4e37-80f8-7fc122cdd785",
  "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": {}
  }
},
"requestReceivedTimestamp": "2020-04-02T08:27:02.000000Z",
"stageTimestamp": "2020-04-02T08:27:02.000000Z",
"annotations": {
  "authorization.k8s.io/decision": "allow",
```
You can use the must-gather tool to collect the audit logs for debugging your cluster, which you can review or send to Red Hat Support.

Procedure

1. Run the `oc adm must-gather` command with -- /usr/bin/gather_audit_logs:

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

2. Create a compressed file from the must-gather directory that was just created in your working directory. For example, on a computer that uses a Linux operating system, run the following command:

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

   Replace `must-gather-local.472290403699006248` with the actual directory name.

3. Attach the compressed file to your support case on the the Customer Support page of the Red Hat Customer Portal.