Security and compliance

Configuring security context constraints on AWS clusters
Configuring security context constraints on AWS clusters
Abstract
This document provides instructions for configuring security context constraints.
Table of Contents

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

CHAPTER 2. ADDING ADDITIONAL CONSTRAINTS FOR IP-BASED AWS ROLE ASSUMPTION ........ 6
  2.1. CREATE AN IDENTITY-BASED IAM POLICY .................. 6
  2.2. ATTACHING THE IDENTITY-BASED IAM POLICY .......... 9
  2.3. ADDITIONAL RESOURCES .................................... 9
CHAPTER 1. AUDIT LOGS

Red Hat OpenShift Service on AWS 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, 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 impersonating another user.</td>
</tr>
<tr>
<td>sourceIPs</td>
<td>Optional. The source IPs, from where the request originated and any intermediate proxies.</td>
</tr>
<tr>
<td>userAgent</td>
<td>Optional. The user agent string reported by the client. Note that the user 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 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 not a Status type. For successful responses, this will only include the code. For non-status type error responses, this will be auto-populated with the error message.</td>
</tr>
<tr>
<td>Field</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>requestObject</td>
<td>Optional. The API object from the request, in JSON format. The RequestObject is recorded as is in the request (possibly re-encoded as JSON), prior to version conversion, defaulting, admission or merging. It is an external versioned object type, and might not be a valid object on its own. This is omitted for non-resource requests and is only logged at request level and higher.</td>
</tr>
<tr>
<td>responseObject</td>
<td>Optional. The API object returned in the response, in JSON format. The ResponseObject is recorded after conversion to the external type, and serialized as JSON. This is omitted for non-resource requests and is only logged at response level.</td>
</tr>
<tr>
<td>requestReceivedTimestamp</td>
<td>The time that the request reached the API server.</td>
</tr>
<tr>
<td>stageTimestamp</td>
<td>The time that the request reached the current audit stage.</td>
</tr>
<tr>
<td>annotations</td>
<td>Optional. An unstructured key value map stored with an audit event that may be set by plugins invoked in the request serving chain, including authentication, authorization and admission plugins. Note that these annotations are for the audit event, and do not correspond to the metadata.annotations of the submitted object. Keys should uniquely identify the informing component to avoid name collisions, for example podsecuritypolicy.admission.k8s.io/policy. Values should be short. Annotations are included in the metadata level.</td>
</tr>
</tbody>
</table>

Example output for the Kubernetes API server:

```
{"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":null,"sourceIPs":null,"userAgent":"cluster-kube-controller-manager-operator/v0.0.0 (linux/amd64) kubernetes/$Format"},"objectRef":null,"responseStatus":null,"requestReceivedTimestamp":"2020-04-02T08:27:20.000000Z","stageTimestamp":null,"annotations":null}
```

1.2. GATHERING AUDIT LOGS
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:

   ```
   $ 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:

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

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

CHAPTER 2. ADDING ADDITIONAL CONSTRAINTS FOR IP-BASED AWS ROLE ASSUMPTION

You can implement an additional layer of security in your AWS account to prevent role assumption from non-allowlisted IP addresses.

2.1. CREATE AN IDENTITY-BASED IAM POLICY

You can create an identity-based Identity and Access Management (IAM) policy that denies access to all AWS actions when the request originates from an IP address other than Red Hat provided IPs.

Prerequisites

- You have access to see the AWS Management Console with the permissions required to create and modify IAM policies.

Procedure

1. Sign in to the AWS Management Console using your AWS account credentials.
2. Navigate to the IAM service.
3. In the IAM console, select Policies from the left navigation menu.
4. Click Create policy.
5. Select the JSON tab to define the policy using JSON format.
6. Copy and paste the following JSON policy document into the editor:

```json
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Deny",
      "Action": "*",
      "Resource": "*",
      "Condition": {
        "NotIpAddress": {
          "aws:SourceIp": [
            "3.223.162.20/32",
            "3.223.177.185/32",
            "54.209.120.28/32",
            "23.21.192.204/32",
            "23.23.16.23/32",
            "3.217.67.187/32",
            "34.206.248.211/32",
            "34.237.192.147/32",
            "52.1.97.230/32",
            "18.214.192.218/32",
            "3.218.132.183/32",
            "52.202.67.83/32",
            "18.220.162.161/32",
            "18.224.36.208/32",
          ]
        }
      }
    }
  ]
}
```
"3.143.200.173/32",
"54.197.245.192/32",
"3.23.162.248/32",
"44.217.70.145/32",
"52.202.89.184/32",
"54.174.41.137/32",
"3.231.181.77/32",
"44.193.253.218/32",
"52.201.38.139/32",
"34.205.217.112/32",
"23.22.217.39/32",
"44.193.121.36/32",
"54.211.144.4/32",
"34.194.251.19/32",
"44.196.79.250/32",
"52.45.208.183/32",
"100.20.120.76/32",
"100.20.197.29/32",
"52.26.177.23/32",
"34.197.214.203/32",
"35.170.167.51/32",
"52.23.44.32",
"44.228.245.162/32",
"44.238.205.35/32",
"54.203.216.175/32",
"34.237.49.153/32",
"44.196.177.146/32",
"52.23.117.40/32",
"44.225.234.235/32",
"44.241.225.78/32",
"44.241.55.3/32",
"34.237.180.56/32",
"44.205.240.205/32",
"52.54.93.238/32",
"35.155.66.53/32",
"44.231.249.237/32",
"44.233.161.100/32",
"3.229.185.234/32",
"54.147.98.63/32",
"54.163.100.197/32",
"23.20.194.86/32",
"23.22.242.238/32",
"54.147.218.140/32",
"52.21.229.141/32",
"54.227.5.10/32",
"54.146.138.135/32",
"23.21.239.1/32",
"52.20.145.130/32",
"54.157.89.24/32",
"107.22.162.110/32",
"3.223.147.2/32",
"54.188.225.66/32",
"54.177.143.128/32",
"54.219.250.189/32",
"18.135.14.84/32",
"18.135.218.119/32",

CHAPTER 2. ADDING ADDITIONAL CONSTRAINTS FOR IP-BASED AWS ROLE ASSUMPTION
NOTE

This list is subject to change. Additionally, you must specify the IP addresses in CIDR notation.

7. Click **Review and create**.

8. Provide a name and description for the policy, and review the details for accuracy.
9. Click Create policy to save the policy.

**NOTE**

The condition key `aws:ViaAWSService` must be set to false to enable subsequent calls to succeed based on the initial call. For example, if you make an initial call to `aws ec2 describe-instances`, all subsequent calls made within the AWS API server to retrieve information about the EBS volumes attached to the ec2 instance will fail if the condition key `aws:ViaAWSService` is not set to false. The subsequent calls would fail because they would originate from AWS IP addresses, which are not included in the AllowList.

### 2.2. ATTACHING THE IDENTITY-BASED IAM POLICY

Once you have created an identity-based IAM policy, attach it to the relevant IAM users, groups, or roles in your AWS account to prevent IP-based role assumption for those entities.

**Procedure**

1. Navigate to the IAM console in the AWS Management Console.

2. Select the default IAM `ManagedOpenShift-Support-Role` role to which you want to attach the policy.

   **NOTE**

   You can change the default IAM `ManagedOpenShift-Support-Role` role. For more information about roles, see [Red Hat support access](#).

3. In the Permissions tab, select Add Permissions or Create inline policy from the Add Permissions drop-down list.

4. Search for the policy you created earlier by:
   a. Entering the policy name.
   b. Filtering by the appropriate category.

5. Select the policy and click Attach policy.

**IMPORTANT**

To ensure effective IP-based role assumption prevention, you must keep the allowlisted IPs up to date. Failure to do so may result in Red Hat site reliability engineering (SRE) being unable to access your account and affect your SLA. If you have further questions or require assistance, please reach out to our support team.

### 2.3. ADDITIONAL RESOURCES

- For more information about denying access based on the source IP, see [AWS: Denies access to AWS based on the source IP](#) in the AWS documentation.