Kubernetes service accounts with failed or forbidden status

You might want information about which Kubernetes service accounts received failed or forbidden status during access attempts when doing the following:

- Monitoring Kubernetes sensitive object access

Prerequisites

In order to execute this procedure in your environment, the following data, services, or apps are required:

- Splunk Enterprise or Splunk Cloud Platform
- Kubernetes
- One of the following:
  - Amazon: Splunk Add-on for Amazon Web Services, Splunk App for AWS, and AWS CloudWatch data
  - Microsoft: Splunk Add-on for Microsoft Cloud Services and Azure storage data
  - Google: Splunk Add-on for Google Cloud Platform and Pub/Sub data

Example

You want to investigate failed or forbidden status accounts on your network to determine if they represent a threat.

To optimize the search shown below, you should specify an index and a time range.

AWS

1. Ensure that your deployment is ingesting CloudWatch logs.
2. Run the following search:

   source="aws:cloudwatchlogs:eks" user.groups{}=system:serviceaccounts responseStatus.status=Failure | table sourceIPs{} src_user userAgent verb responseStatus.status requestURI

Search explanation

The table provides an explanation of what each part of this search achieves. You can adjust this query based on the specifics of your environment.
**Splunk Search**

```plaintext
sourcetype="aws:cloudwatchlogs:eks"
```

**Explanation**

Search only AWS EKS Kubernetes data.

```plaintext
user.groups{}=system:serviceaccounts
```

**Explanation**

Search the service accounts user group.

```plaintext
responseStatus.status=Failure
```

**Explanation**

Search for requests with a failure status.

```plaintext
| table sourceIPs{} src_user userAgent verb responseStatus.status requestURI
```

**Explanation**

Display the results in a table with columns in the order shown.

---

**Azure**

1. Ensure that you have configured Kube-Audit data diagnostics.
2. Run the following search:

```plaintext
sourcetype=mscs:storage:blob:json category=kube-audit
| spath input=properties.log
| search user.groups{}=system:serviceaccounts*  responseStatus.reason=Forbidden
| table sourceIPs{} user.username userAgent verb responseStatus.reason responseStatus.status properties.pod objectRef.namespace
```

**Search explanation**

The table provides an explanation of what each part of this search achieves. You can adjust this query based on the specifics of your environment.

---

**Splunk Search**

```plaintext
sourcetype:mscs:storage:blob:json
```

**Explanation**

Search only the source type mscs:storage:blob:json.

```plaintext
category=kube-audit
```

**Explanation**

Search the data source kube-audit from the diagnostic logs in Azure Cloud services.

```plaintext
| spath input=properties.log
```

**Explanation**

Extract fields from the properties Kube-Audit log.

```plaintext
| search user.groups{}=system:serviceaccounts*
```

**Explanation**

Search for service accounts with a response status of Forbidden.
Splunk Search

responseStatus.reason=Forbidden

| table sourcIPs() user.username userAgent verb responseStatus.reason responseStatus.status properties.pod objectRef.namespace

Display the results in a table with columns in the order shown.

Explanation

Forbidden.

GCP

1. Ensure that your deployment is ingesting Pub/Sub messaging logs.
2. Run the following search:

```
| table src_ip src_user http_user_agent data.protoPayload.response.spec.resourceAttributes.namespace data.resource.labels.cluster_name data.protoPayload.response.spec.resourceAttributes.verb data.protoPayload.request.status.allowed data.protoPayload.response.status.reason data.labels.authorization.k8s.io/decision | dedup src_ip src_user
```

Search explanation

The table provides an explanation of what each part of this search achieves. You can adjust this query based on the specifics of your environment.

Splunk Search

sourceType="google:gcp:pubsub:message" system:serviceaccounts data.protoPayload.response.status.allowed!=*

Search only GCP Pub/Sub messages.

Explation

Search for service accounts that do not have a response status of allowed.

| table src_ip src_user http_user_agent data.protoPayload.response.spec.resourceAttributes.namespace data.resource.labels.cluster_name data.protoPayload.response.spec.resourceAttributes.verb data.protoPayload.request.status.allowed data.protoPayload.response.status.reason data.labels.authorization.k8s.io/decision

Display the results in a table with columns in the order shown.
### Splunk Search

| dedup src_ip src_user |

### Explanation

Remove duplicate results from the same IPs and users.

---

## Result

You can extend this search by using top or rare operators to find trends or rarities in failure status, user agents, source IP addresses, and request URIs. Note that this search can give false positives as there might be inherent issues with authentications and permissions at cluster.

For additional information about this search, such as its applicability to common frameworks and standards, see this project on GitHub for [AWS](https://aws.amazon.com), [Azure](https://azure.microsoft.com), or [GCP](https://cloud.google.com).