Container Security

Release Notes for Sensor

Version 1.8.0

May 17, 2021 (Updated June 3, 2021)

Here's what's new in the Container Security Sensor!

Collection of Kubernetes Cluster Attributes

In this release we have added collection of Kubernetes cluster attributes and made this information searchable in the UI. Kubernetes cluster attributes include node details, pod details (name, uuid, namespace, labels), controller details (name, uuid, type) and more. Use Container Security APIs to see the Kubernetes cluster attributes collected for your containers and sensors.

Important - Kubernetes attributes will only be processed for containers discovered after the Container Security version 1.10 release. Kubernetes attributes are collected as part of container inspect processing when containers are discovered for the first time. To fetch Kubernetes cluster attributes for an existing deployment in Kubernetes, you will have to “rollout restart” the existing deployment, which will create new containers and this will start the container inspect processing. Kubernetes attributes will get collected for the newly created containers on Kubernetes clusters.

Use the following command for the “rollout restart”:

```
kubectl rollout restart deployment <deployment-name> -n <namespace>
```

Kubernetes cluster attributes:

- Cluster type (Kubernetes)
- Cluster version
- Project name (collected for projects in Google Cloud Platform)
- Node name and flag indicating whether the node is the master node
- Pod name
- Pod UUID
- Pod namespace
- Pod labels (key and value pairs)
- Controller name
- Controller UUID
- Controller type (e.g. DaemonSet, Deployment, ReplicaSet, etc)

Static Scanning Enhancements

We made the following enhancements to static scanning in this release.

Distroless image scanning support

Static scanning is now supported for Google distroless images without shell.

Static scan performance

To improve scan duration we are using persistent storage to store temporary artifacts generated during the static scan. As a result of this change, if you have large images without shell, the sensor requirement for disk space may exceed the minimum requirement of 1GB.

Support for Harbor Registry

The Harbor registry is now supported for the Registry sensor. To scan images in your Harbor registry, you’ll need to complete these steps:

1) Download the Registry sensor. Go to **Configurations** > **Sensors** > **Download Sensor** and pick **Registry**. Select the Docker environment where you want to deploy the sensor and follow the installation instructions on the screen. Ensure the registry sensor is in Running state and continue to the next step.

2) Add your Harbor registry in the Container Security UI and set up a scanning schedule. Go to **Assets** > **Registries** > **New Registry**. Choose registry type **Docker V2-Private** and provide the registry URL and authentication credentials for connecting to your registry. Admin credentials are required. If your Harbor registry version supports Token based authentication, then the sensor will perform the V2 catalog call with the authentication token. If authentication fails, then the sensor will automatically fall back to the Basic authentication method for the V2 catalog call.

Jobs API Now Used to Create Image Scanning PODs

The Container Security Sensor will now use the Kubernetes Jobs API to create image scanning pods instead of creating Kubernetes standalone pods.

Updates to Installsensor and Autoupdate Scripts for Docker.Sock

For installing the sensor on MacOS, we made the following changes:

- Updated the installsensor.sh script to include docker.sock as direct volume mapping
- Updated the autoupdate.sh script to consider /var/run/docker.sock

If you have an older sensor version (older than 1.8.0) running on an older MacOS Catalina version (older than 10.15), then it will carry "/var/run" mapping for docker socket to the auto updated version of the sensor. When the host is upgraded to Catalina 10.15 and the docker desktop is restarted, the sensor restart policy will restart the sensor. After the restart, the auto updated sensor with version 1.8.0 will fail because it will have "/var/run" mapping for docker socket. In this case, you’ll need to re-install sensor version 1.8.0, which will by default use the correct docker socket mapping i.e. "/var/run/docker.sock".

You’ll also need to make sure the sensor has permissions to write to the hostid file in /private/etc/qualys/ directory. To provide the sufficient permissions, execute the following commands before running the install sensor command:

```
sudo mkdir /private/etc/qualys/
sudo touch /private/etc/qualys/hostid
sudo chmod 666 /private/etc/qualys/hosted
```

New Init Parameter Supported for Installsensor Script

Since Docker doesn't run a special init process, a container can end up with zombie processes. We made the following changes in order to reap the zombie processes:

- For a standalone Docker environment, we changed the installsensor script to run with the --init parameter.

- For a Kubernetes environment, we added the tini binary to spawn child processes and reap zombies.

Issues Addressed

- We made a fix for Container Security Sensor in the OpenShift CRI-O environment (in 1.8.0-3). With this fix, CS Sensor will fetch the latest resource version before starting the watch API call to monitor pod events. Please upgrade your sensor to the latest version to get this fix. Prior to this fix, the sensor was using an old resource version and it was going into a continuous loop.
- We fixed an issue where sensor installation was failing on Kali Linux which was using cgroup v2. Now, sensor deployment is supported on Linux host with cgroup v2 support.
- We made a fix to handle registry repositories with empty tags list.