Container Security

Release Notes for Sensor

Version 1.11.0
February 3, 2022 (Updated February 15, 2022)

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

Support for JFrog Artifactory Private Registry Scanning

We now support JFrog Artifactory Private registry for the Registry sensor. This new functionality allows users to scan JFrog Artifactory repositories. The sensor will use the Artifactory Native API with AQL (Artifactory Query Language) for the listing phase of the registry scan to collect image metadata information for the repository provided in the registry scan schedule. The Artifactory Native API provides a more efficient method of finding images which will result in performance improvements.

To scan images in your JFrog Artifactory Private registry, you'll need to complete these steps:

1) Download the Registry sensor (Sensor version 1.11 or later). Go to **Configurations** > **Sensors > Download Sensor** and pick **Registry**. Select the 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 registry in the Container Security UI and set up a scanning schedule. Go to **Assets** > **Registries** > **New Registry**. Choose registry type **JFrog Artifactory Private**. Provide the following information:

URL - Enter the registry URL.

Access Method - Choose **Path** (for direct access to Docker registries) or **Sub Domain** (for access to Docker registries through a reverse proxy).

Service Context - Enter the service context value configured as part of the Base URL for reverse proxy configuration.

3) After adding registry information, click **Next** to enter scan settings. Like with other registry types, you can choose to scan immediately (On Demand) or on an on-going basis (Automatic). See the online help for guidance on scan settings.

Support for RedHat Quay Registry Scanning

The RedHat Quay registry is now supported for the Registry sensor. This new option allows users to scan the images that are stored in the Quay registry.

To scan images in your RedHat Quay registry, you'll need to complete these steps:

1) Download the Registry sensor (Sensor version 1.11 or later). Go to **Configurations** > **Sensors > Download Sensor** and pick **Registry**. Select the 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 registry in the Container Security UI and set up a scanning schedule. Go to **Assets** > **Registries** > **New Registry**. Choose registry type **RedHat Quay**. Provide the registry URL (e.g. https://quay.io) and authentication credentials for connecting to your registry. You can provide a standard username and password (for an account with Admin or Super User privileges) or robot account credentials. For a robot account, the username is formatted as UserName+RobotAccountName and the password is the password token for the robot account.

3) After adding registry information, click **Next** to enter scan settings. Like with other registry types, you can choose to scan immediately (On Demand) or on an on-going basis (Automatic). See the online help for guidance on scan settings.

Static Scanning Now Supported on Containerd Runtime

With this release, we now support static scanning for sensor deployments in Kubernetes with Containerd Runtime.

The sensor will perform static scanning for container images as a fallback mechanism to current dynamic scanning in case container image does not have a shell. Static scanning will also be performed for Google distroless images without shell. Static scanning will not be performed on container or container images having a shell.

Static scanning collects the list of installed software from the container image file system to find vulnerabilities in the container images. The installed software list is retrieved from the Package manager metadata files. Package managers supported are RPM, DPKG and Alpine. If you have large images without shell on the host where sensor is running, the requirement for disk space may exceed the minimum requirement of 1GB.

Containerd Runtime and CRI-O Runtime Now Supported by Registry Sensor

Containerd Runtime and CRI-O Runtime are now supported for registry sensor. Prior to this release, we were only supporting Docker Runtime for the registry sensor. Now these other runtimes are also supported.

New Option for Optimizing Image Scans

By default, the Container Security Sensor scans every image that it detects on the host. This results in redundant scanning of images. In order to optimize image scans, we've introduced a new argument called "--optimize-image-scans". When you install the General sensor with this new argument, the sensor will communicate with the Qualys Cloud Platform and perform informed scans to avoid redundant image scans. The sensor will determine if the images present on the host are already scanned by other sensors for the same manifest and version and will not scan those images again. Please note that this feature is available for General sensor type only.

Support for Deploying Sensor On Kubernetes Clusters Using Cgroup V2

Starting in this release, the Container Sensor will use the Kubernetes Downward API to fetch its own containerid. Prior to this release, the sensor used the /proc/self/cgroup file to fetch its containerid but the structure of the file varies between different cgroup drivers. Now we've removed this dependency on the cgroup file.

Make sure the cssensor deployment file is updated to the latest version and env contains the following:

```
- name: QUALYS_POD_NAME
  valueFrom:
    fieldRef:
      fieldPath: metadata.name
- name: QUALYS_POD_NAMESPACE
  valueFrom:
    fieldRef:
      fieldPath: metadata.namespace
```

Update to apiVersion in Container Sensor Deployment YAML Files for RBAC

We changed the apiVersion from **v1beta1** to **v1** in the sensor deployment template yaml files. These yaml files were updated: cssensor-ds.yml, cssensor-containerd-ds.yml, cssensor-crio-ds.yml, cssensor-openshift-crio-ds.yml, cssensor-ds_pv_pvc.yml.

Note - If your Kubernetes version is 1.17 or older, then you will need to change the apiVersion in the yaml file back to v1beta1.

This change appears in the following sections of the yaml files:

```
...

 # Role for all permission to qualys namespace
  - kind: Role
    apiVersion: rbac.authorization.k8s.io/v1
    metadata:
      name: qualys-reader-role
      namespace: qualys
...

# ClusterRole for read permission to whole cluster
  - kind: ClusterRole
    apiVersion: rbac.authorization.k8s.io/v1
    metadata:
      name: qualys-cluster-reader-role
...

# RoleBinding to assign permissions in qualys-reader-role to qualys-service-account
  - kind: RoleBinding
    apiVersion: rbac.authorization.k8s.io/v1
    metadata:
      name: qualys-reader-rb
      namespace: qualys
...

# ClusterRoleBinding to assign permissions in qualys-cluster-reader-role to qualys-service-account
  - kind: ClusterRoleBinding
    apiVersion: rbac.authorization.k8s.io/v1
    metadata:
      name: qualys-cluster-reader-rb
…
```

Deploying Sensor in Kubernetes using Helm Now Supported

Helm is the package manager for Kubernetes and Helm Charts can be used to deploy the Container Security Sensor in Kubernetes. See the Container Security Sensor Deployment Guide for complete details.

How to Deploy a Helm Chart on Kubernetes Cluster

Follow the steps below to deploy the helm chart on the Kubernetes cluster.

1) Download the appropriate helm chart for your Kubernetes deployment based on the Container Runtime. Helm chart templates for sensor installation are available in the QualysContainerSensor.tar.xz as helm-chart-scripts.tar.xz. You can also download them directly from GitHub at https://github.com/Qualys/cs_sensor/helm-chart-scripts/.

2) Update the following values in the <chart-name>/values.yaml file: customer id, activation id, and Qualys URL. Update any other options in the values.yaml file, as needed for your environment.

3) Deploy the helm chart using following helm command:

```
helm install qualys-sensor sensor-chart
```

Lowered Sensor Health Status Check Interval

To make the Container Security Sensor Docker policy compliant we added the HEALTHCHECK instruction in Dockerfile starting in Sensor version 1.10, and set the interval to 24 hours. Now we've lowered this interval to default 30 seconds. The status will start as "health: starting" and change to "healthy" after the first health check interval 30 seconds after the launch. Please note that as long as the sensor STATUS in 'docker ps' is "Up" the sensor is running successfully.

Here's an example of sensor health status:

```
root@ip-10-11-12-13:/home/ubuntu# docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
e12ddf3e4ff5 qualys/sensor:latest "/usr/bin/tini -- /u…" About a minute ago
Up About a minute (healthy) qualys-container-sensor
```

End of Support for Environment Variable QUALYS_SCANNING_CONTAINER_SCOPECLUSTER

In the Container Security Sensor 1.10 release, we announced that the environment variable QUALYS_SCANNING_CONTAINER_SCOPECLUSTER, which allowed users to disable node affinity, was being deprecated. Starting in Container Security Sensor 1.11, the environment variable is no longer supported and users will no longer be allowed to disable node affinity. Node affinity is enforced by default, and this means the image scanning takes place on the same node as the sensor that initiated the scan.

RedHat OpenShift 4.9 Supported

Container Security Sensor version 1.11 has been tested and verified with RedHat OpenShift 4.9.

Issues Addressed

- Registry sensor will remove previously pulled container images on restart.
- The sensor can run with tolerations specified in the deployment YAML file. We made an improvement to carry the same tolerations to image scanning pods when they are launched.