Qualys.

Container Security

Release Notes for Sensor

Version 1.5.0
August 14, 2020

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

CRI-O Runtime Support

Starting with Container Security v1.6.8, CRI-O Runtime will be supported in OpenShift 4.4 and above, and in Kubernetes without OpenShift. With this support, you'll be able to use the general (host) sensor to scan images and containers when the underlying runtime is CRI-O. CRI-O Runtime is not supported with the Registry and CI/CD modes. See the Qualys Container Security Sensor Deployment Guide for deployment steps.

New parameters for LogFileSize and LogFilePurgeCount

Now, during sensor installation, you'll be able to specify new parameters to define the maximum size per log file and maximum number of archived log files. You can provide these parameters as command line arguments or as arguments in deployment files.

New command line parameters for "installsensor.sh" script:

LogFileSize: Configuration to set the maximum size per log file for sensor in bytes. This configuration accepts "<digit><K/M/>" where "K" is kilobyte and "M" is megabyte. For example, specify "10" for 10 bytes, "10K" for 10 kilobytes, "10M" for 10 megabytes. The default is "10M".

LogFilePurgeCount: Integer value that specifies the maximum number of archived log files. The default is "5".

New values you can provide in command or args parameter when deploying a sensor:

"--log-filesize" to set the maximum size per log file for sensor in bytes. This parameter accepts "<digit><K/M/>" where "K" is kilobyte and "M" is megabyte. For example, specify "10" for 10 bytes, "10K" for 10 kilobytes, "10M" for 10 megabytes. The default is "10M".

"--log-filepurgecount" to define the number of archived qpa.log files to be generated. The default is "5".

Sample args value for deploying a Registry Sensor:

args: ["--k8s-mode","--registry-sensor","--log-level","5"," --log-filesize","5M","--log-filepurgecount","4"]

Support sensor deployment on master node

In this release, we added toleration to the sensor yml file. This will allow you to schedule sensor pods on the master node. By default, the toleration is commented out; you'll need to uncomment it in the yml file to use this feature.

Use Kubectl to launch image scanning pod

Now you can launch the image scanning pod using the kubectl command line tool by specifying the command line argument --use-kubectl. The sensor will perform image scanning pod creation, execution and deletion in the 'qualys' namespace only.

The Sensor v1.5.0 has to be installed using new yml files for docker and containerD as they are modified to launch the Sensor daemonset in 'qualys' namespace with sensor permissions controlled by RBAC.

Introducing node affinity rule for image scanning

In Kubernetes environments, the sensor launches a pod to scan the images when --use-kubectl flag is used. The image scanning pods are by default launched on the same node as the sensor. This is required to prevent the images being replicated across the cluster. Sometimes due to lack of resources, the Kubernetes API server may not be able to schedule the image scanning pod on the same node, which may lead to scan failure. In this case, you can define the new environment variable "QUALYS_SCANNING_CONTAINER_SCOPECLUSTER" in the yml file and set it to 1. This will cause the sensor to launch image scanning pods across the cluster.

Issues Addressed

- The Registry sensor will now chunk a tag list authorization token request into multiple requests, if the URL length goes over 2048 characters. This can happen if a large number of repositories present in the registry matches the filter.

- The Registry sensor will now regenerate the token in the case of authentication failure due to token expiry. Token expiry scenario will be handled during all three operations: 1) tag listing request, 2) manifests request, and 3) blobs request.

- We fixed an issue where a sensor deployed in kubernetes clusters which have systemd as the cgroup driver for kubelet and containerd runtime was not able to start because the sensor could not read its own containerd ID.