Using Kernel Hardening Tools: AppArmor

Every Kubernetes Host (Linux Machine) has a kernal space and user space

when you install an application in User Space, it does system calls to Kernal Space like system specific and i/o calls (open, close, write, touch)

AppArmor is a kernal security module for kubernetes

it restricts the container’s access to resources.

Linux capabilities, network access and file permissions.

Objective is to secure/restrict the container what are allowed?

It is configured through profiles, which are a set of rules. Which we load in kernal.

Profile is loaded in 2 different modes

Enforce Mode (Prevent the access)

Complain Mode (Complaint and create system logs)

default path

/etc/apparmor.d/

Install AppArmor

Create Profiles

Enforce/load AppArmor profiles (in all nodes)

Apply profile to pod (add annotations in the yaml file)

Containers & system are secure

sudo systemctl status apparmor (it comes by default in ubuntu)

```
#check enabled in nodes
cat /sys/module/apparmor/parameters/enabled
aa-status
sudo aa-status

cd /etc/aparmor.d/
```

sudo vim k8s-apparmor-example-deny-write

```
#include <tunables/global>

profile k8s-apparmor-example-deny-write flags=(attach_disconnected) {
  #include <abstractions/base>

  file,

  # Deny all file writes.
  deny /** w,
}
```

ls -l

#load the profile

sudo apparmor_parser /etc/apparmor.d/k8s-apparmor-example-deny-write

sudo aa-status #to confirm if the profile is loaded or not.

Now next is to create an annotation…

cd ~

vim pod.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: hello-apparmor
spec:
  containers:
  - name: hello
    image: busybox:1.28
    command: [ "sh", "-c", "echo 'Hello AppArmor!' && sleep 1h" ]
```

kubectl apply -f pod.yaml

kubectl get pod

kubectl exec -it hello-aparmor -- sh

echo ‘12345’ > test.txt

ls

exit

vim pod2.yaml

```
apiVersion: v1
kind: Pod
metadata:
  name: hello-apparmor
  annotations:
    # Tell Kubernetes to apply the AppArmor profile "k8s-apparmor-example-deny-write".
    # Note that this is ignored if the Kubernetes node is not running version 1.4 or
greater.
```

```
    container.apparmor.security.beta.kubernetes.io/hello: localhost/k8s-apparmor-example-deny-write
spec:
  containers:
  - name: hello
    image: busybox:1.28
    command: [ "sh", "-c", "echo 'Hello AppArmor!' && sleep 1h" ]
```

kubectl delete pod hello-apparmor

kubectl apply -f pod2.yaml

kubectl describe pod hello-apparmor

kubectl exec -it hello-aparmor -- sh

touch test

echo ‘123445’ > text2.txt

to delete

sudo apparmor_parser -R /etc/apparmor.d/k8s-apparmor-example-deny-write

sudo aa-status