.

Lesson 05 Demo 05

Blocking All Traffic from an Application

Objective: To effectively block all the network traffic from a specific application to ensure enhanced security

Tools required: kubeadm, kubectl, kubelet, and containerd

Prerequisites: A Kubernetes cluster (refer to Demo 01 from Lesson 01 for setting up a cluster)

Steps to be followed:

1. Set up the application pod and policy
2. Verify the network policy

Step 1: Set up the application pod and policy

1.1 Create an nginx pod with the label **app=simplilearn** and expose it at port 80 using the following command:
kubectl run simplilearn --image=nginx --labels="app=simplilearn" --expose --port=80

```
labsuser@master:~$ kubectl run simplilearn --image=nginx --labels="app=simplilearn" --expose --port=80
service/simplilearn created
pod/simplilearn created
labsuser@master:~$
```

.

1.2 Execute a temporary pod and make a request to the web service using the following commands:
kubectl run --rm -i -t --image=alpine test-$RANDOM -- sh
wget -qO- http://simplilearn

```
labsuser@master:~$ kubectl run --rm -i -t --image=alpine test-$RANDOM -- sh
If you don't see a command prompt, try pressing enter.
/ # wget -qO- http://simplilearn
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
html { color-scheme: light dark; }
body { width: 35em; margin: 0 auto;
font-family: Tahoma, Verdana, Arial, sans-serif; }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
```

1.3 Create a configuration file named **simplilearn-deny-all.yaml** using the following command:
nano simplilearn-deny-all.yaml

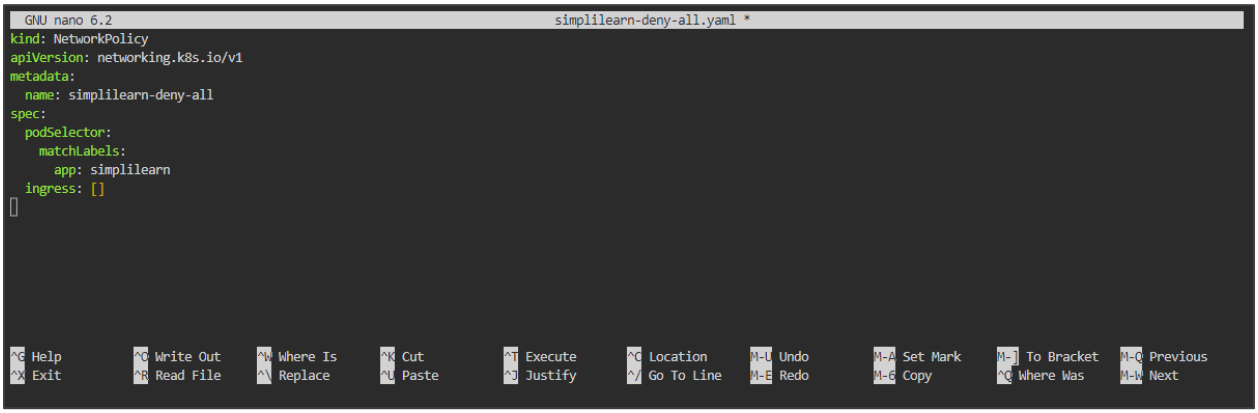
```
<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
/ # exit
Session ended, resume using 'kubectl attach test-26731 -c test-26731 -i -t' command when the pod is running
pod "test-26731" deleted
labsuser@master:~$ nano simplilearn-deny-all.yaml
```

.

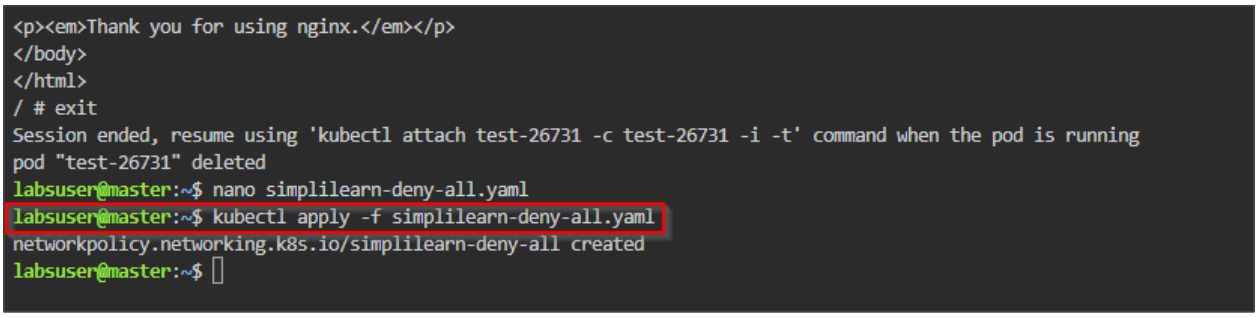
1.4 Add the following YAML code to the **simplilearn-deny-all.yaml** file:

apiVersion: networking.k8s.io/v1
kind: NetworkPolicy
metadata:
name: simplilearn-deny-all
spec:
podSelector:
matchLabels:
app: simplilearn
ingress: []



Step 2: Verify the network policy

2.1 Create the network policy using the following command:
kubectl apply -f simplilearn-deny-all.yaml



.

2.2 Verify the network policy using the following command:
kubectl get networkpolicy

```
<p><em>Thank you for using nginx.</em></p>
</body>
</html>
/ # exit
Session ended, resume using 'kubectl attach test-26731 -c test-26731 -i -t' command when the pod is running
pod "test-26731" deleted
labsuser@master:~$ nano simplilearn-deny-all.yaml
labsuser@master:~$ kubectl apply -f simplilearn-deny-all.yaml
networkpolicy.networking.k8s.io/simplilearn-deny-all created
labsuser@master:~$  kubectl get networkpolicy
NAME                 POD-SELECTOR      AGE
simplilearn-deny-all   app=simplilearn   89s
labsuser@master:~$
```

2.3 Validate if the network policy blocks the traffic using the following commands:
kubectl run --rm -i -t --image=alpine test-$RANDOM -- sh
wget -qO- --timeout=2 http://simplilearn

```
labsuser@master:~$ nano simplilearn-deny-all.yaml
labsuser@master:~$ kubectl apply -f simplilearn-deny-all.yaml
networkpolicy.networking.k8s.io/simplilearn-deny-all created
labsuser@master:~$  kubectl get networkpolicy
NAME                 POD-SELECTOR      AGE
simplilearn-deny-all   app=simplilearn   89s
labsuser@master:~$ kubectl run --rm -i -t --image=alpine test-$RANDOM -- sh
If you don't see a command prompt, try pressing enter.
/ # wget -qO- --timeout=2 http://simplilearn
wget: download timed out
/ #
```

Note: The provided network policy with an empty spec ingress does not allow any traffic into the pod. However, if there's at least one network policy with a rule permitting the traffic, it is directed to the pod, bypassing other blocking policies.

By following these steps, you have successfully restricted all network traffic from an application to enhance its security.