Configure packet forwarding for pods in EKS

Published: 2022-01-07

By default, if you have configured traffic mirroring for EC2 instances that host an AWS Elastic Kubernetes Service (EKS) cluster, all of the traffic between nodes in the cluster is seen by the ExtraHop system. Most ExtraHop security detections can be generated from node-level traffic monitoring; however, if you want to monitor traffic between pods for added visibility, you must enable packet forwarding in your EKS cluster.

This guide shows you how to deploy the rpcapd software tap as a DaemonSet service that automatically configures packet forwarding for each pod in a cluster backed by EC2 instances. In addition to configuring packet forwarding, the rpcapd container also deduplicates packets that would otherwise be forwarded multiple times to the ExtraHop sensor.

Retrieve subnets for pods and services

Before you can configure the ExtraHop system to monitor pods in EKS, you must retrieve the subnets allocated to the pods and allocated to the services the pods support.

Important: Note the subnets you retrieve; you will need the subnets in the deployment procedure.

1. Retrieve the subnets for pods.
   a) In the AWS console, click **Services** and then select **Elastic Kubernetes Service**.
   b) Click **Clusters**.
   c) Click the name of the cluster that contains the pods you want to monitor.
   d) Click the **Configuration** tab.
   e) Click the **Networking** tab.
   f) For each subnet in the Subnets section, click the subnet, and then note the CIDR block in the IPv4 CIDR column of the Subnets table.
2. Retrieve the subnets for services.
   a) Return to the **Networking** tab on the Cluster page.
   b) Note the CIDR block under Service IPv4 range.

Configure the ExtraHop system to discover pods

With L2 discovery, the ExtraHop system assigns all IP addresses to an associated L2 device; this is the default setting for ExtraHop systems. If L2 discovery is enabled, you must configure the ExtraHop system to discover Kubernetes pods as remote devices, even if the pods are located on nodes inside your local network. Otherwise, the pod IP addresses will only be associated with the corresponding L2 devices for the Kubernetes nodes, and the system will not track the pods as separate devices.

Enable RPCAP on the ExtraHop system.

a) Configure RPCAP on the ExtraHop system.
b) Configure a packet-forwarding rule for the pod subnet on the ExtraHop system.
   - Note the port number you select. You will need the number in the deployment procedure.
   - In the Interface Address field, specify the pod subnet as a CIDR block.
   - Leave the Interface Name field blank.
   - Leave the Filter field blank.
c) Save the running configuration file.

Create the rpcapd container image

Create a container image for the containers that will forward packets to the ExtraHop system. After you create the container image, you must store the image in a registry that is accessible to all nodes in the EKS cluster. The registry can be the AWS Elastic Container Registry (ECR) or another third-party registry.

Note: The following instructions show you how to create the container image with Docker. However, you can create the image with any tool that produces Open Container Initiative (OCI) compliant images.

1. Download the RPCAP install files.
   Click the download link under Installation package for Other Linux distributions.
2. Open a terminal application and run the following command to extract the install script from the file:

```
tar xf rpcapd-8.0.5.3940.tar.gz
```

3. Enter the rpcapd directory:

```
cd rpcapd
```

4. Change the name of the rpcapd executable file to rpcapd:

```
mv rpcapd-64bit-linux rpcapd
```

5. Go to the ExtraHop code-examples GitHub repository and download the `deploy_kubernetes_daemon/init.sh` and `deploy_kubernetes_daemon/Dockerfile` files to the `rpcapd` directory.
6. In the `rpcapd` directory, create the docker image:

```
docker build -t rpcapd .
```

7. If you are storing the image in ECR, log in to the registry:

```
aws ecr get-login-password --region REGION | docker login --username AWS
 --password-stdin EXAMPLE_REGISTRY
```

8. Tag the image in a registry that can be accessed by all nodes in your Kubernetes cluster:

```
docker tag rpcapd EXAMPLE-REGISTRY/rpcapd:latest
```

   Note: You must replace `EXAMPLE_REGISTRY` with the name of your registry.

9. Push the image to the registry:

```
docker image push EXAMPLE-REGISTRY/rpcapd:latest
```

Deploy the rpcapd DaemonSet service

1. Write the DaemonSet spec file.
   a) Go to the ExtraHop code-examples GitHub repository and download the `deploy_kubernetes_daemon/rpcapd_daemon.yaml` file to the `rpcapd` directory.
   b) In the `rpcapd` directory, open the `rpcapd_daemon.yaml` file in a text editor.
   c) Replace the values for the following variables with information from your environment:

image

The name and registry location of the image you created in the previous procedure. For example:

```
EXAMPLE-REGISTRY/rpcapd:latest
```

Tip: If you are storing the image in ECR, you can retrieve this string from the AWS Console by clicking **Services**, and then selecting **Elastic Container Registry**, then clicking the name of the repository you pushed the image to, and then clicking the copy icon in the Image URI column.

env.name = EXTRAHOP_SENSOR_IP

The IP address of the ExtraHop sensor

env.name = RPCAPD_TARGET_PORT

The port on the ExtraHop sensor that you assigned the packet-forwarding rule to.

env.name = PODNET

The subnets of the pods in your cluster that you retrieved earlier, in a comma-separated list.

env.name = SVCNET

The subnets of the services in your cluster that you retrieved earlier, in a comma-separated list.

d) Save and close the `rpcapd_daemon.yaml` file.

2. Deploy the DaemonSet by running the following command:

```
kubectl apply -f rpcapd_daemon.yaml
```

The system displays output similar to the following text:

```
namespace/extrahop created
daemonset.apps/extrahop-rpcapd created
```

3. Confirm that the deployment was successful:

```
kubectl wait pod -n extrahop -l component=extrahop-rpcapd --
for=condition=Ready
```

When a pod is deployed, the command displays output similar to the following text:

```
pod/extrahop-rpcapd-vfctb condition met
```

After every pod is deployed, the command exits.

You can now view metrics for pods in your EKS cluster in the ExtraHop system.