Initial Configuration

Docker Setup

Install Docker from the Docker repository. Use the following commands to install Docker on a fresh OS.

Procedure

Step 1 Uninstall all other packages.

```
for pkg in docker.io docker-doc docker-compose docker-compose-v2 podman-docker containerd runc; do
sudo apt-get remove $pkg; done
```

Step 2 Set up Docker's APT repository.

a) Add Docker's official GPG key.

```
sudo apt-get update

sudo apt-get install ca-certificates curl

sudo install -m 0755 -d /etc/apt/keyrings

sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc

sudo chmod a+r /etc/apt/keyrings/docker.asc
```

b) Add the repository to APT sources.

```
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc]

https://download.docker.com/linux/ubuntu \
  $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update
```

Step 3 Install the Docker packages.

```
sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-plugin docker-compose-plugin
```

Note

Reboot the system to start Docker.

Step 4 Verify that the Docker Engine installation is successful by running the hello-world image using the command:

```
sudo docker run hello-world
```

This command downloads a test image and runs it in a container. When the container runs, it prints a confirmation message and exits.

Cisco Cyber Vision Docker Sensor Host Configuration

Docker Registry

Docker uses registries to pull container images. It classifies a registry as either secure or insecure.

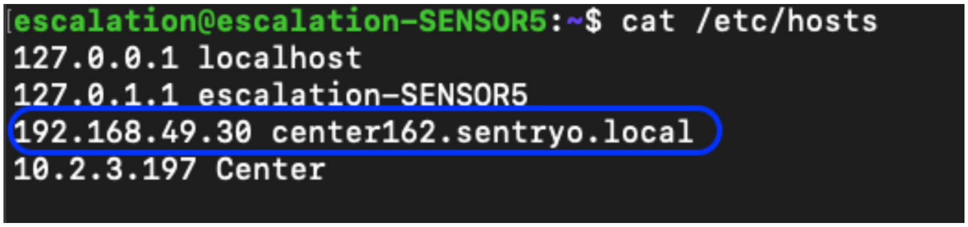
- A secure registry uses TLS and places a copy of its CA certificate on the Docker host at /etc/docker/certs.d/registry-FQDN:443/ca.crt.
- An insecure registry does not use TLS (listens on plaintext HTTP) or uses TLS with a CA certificate that is not recognized by the Docker daemon.

This issue occurs when the certificate is missing in /etc/docker/certs.d/registry-FQDN:443/ or when certificate verification fails due to an incorrect CA.

By default, Docker assumes all registries are secure, except for local ones. If Docker assumes a registry is secure, communication with an insecure registry fails. Configure the Docker daemon specifically to communicate with an insecure registry.

Docker Registry Secure Configuration

1. **FQDN**: Ensure that the host resolves the Cyber Vision Center FQDN. If the host cannot resolve the Center FQDN, specify the correct IP address in the host's configuration file.



```
escalation@escalation-SENSOR5:~$ cat /etc/hosts
127.0.0.1 localhost
127.0.1.1 escalation-SENSOR5
192.168.49.30 center162.sentryo.local
10.2.3.197 Center
```

2. **Certificate**: If the Cyber Vision Center and the host share the same Certificate Authority, you do not need additional configuration. Otherwise, add the Center certificate to a specific folder on the host for authentication.

   Download the ca.pem file of your Center, rename it as ca.crt, and copy it into the folder **/etc/docker/certs.d/Center FQDN:443/**.

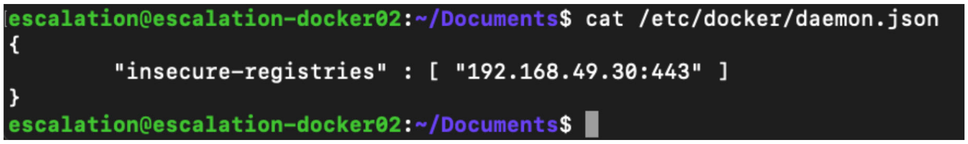
   Create a folder with the Center FQDN + ':443', for example, center162.sentryo.local:443, and add the ca.crt file to it.

/etc/docker/certs.d/center162.sentryo.local:443/ca.crt

Docker Registry Insecure Configuration

Use the Center IP directly without FQDN resolution. Define the Center IP as an insecure registry in the Docker configuration by adding it to **/etc/docker/daemon.json**. Restart Docker with **sudo systemctl restart docker.service**.

For example:



```
escalation@escalation-docker02:~/Documents$ cat /etc/docker/daemon.json
{
        "insecure-registries" : [ "192.168.49.30:443" ]
}
escalation@escalation-docker02:~/Documents$ 
```

Configuration example:

```
{

        "insecure-registries" : ["192.168.49.30:443"]

}
```

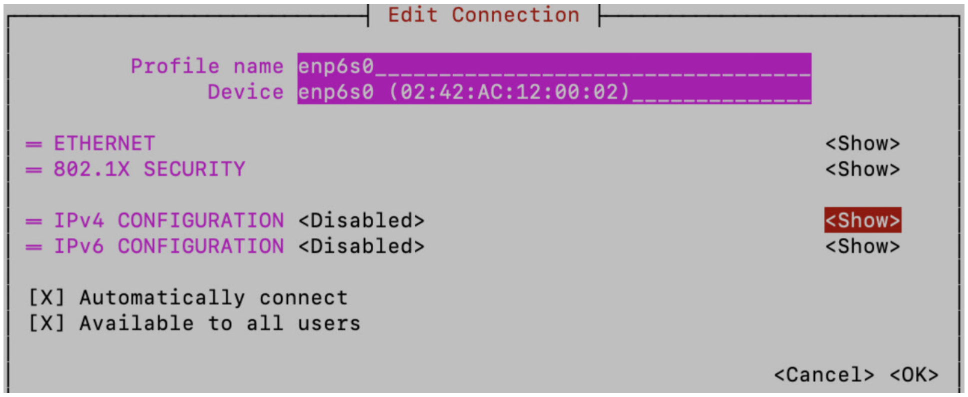
Other Host Configurations

1. **Network Configuration**: Disable IPv4 and IPv6 on all interfaces that are used for capturing traffic or performing active discovery.

   ```
   ipv4.method: "disabled"
   ipv6.method: "disabled"
   ```

   Use nmtui (Network Manager Text User Interface) to configure each interface for the Docker sensor application that is used in passive monitoring and active discovery. Launch the tool by typing: sudo nmtui, then use the graphical interface to disable IPv4 and IPv6 on all sensor interfaces.

   For example:

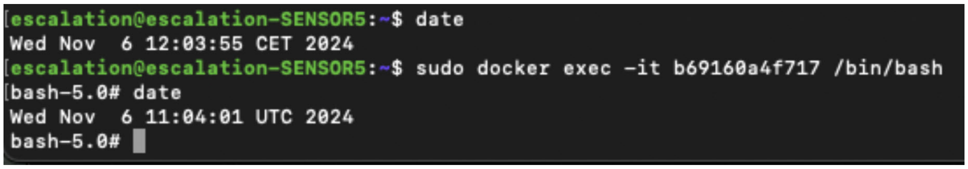


```
                          Edit Connection

         Profile name enp6s0___________________________________
               Device enp6s0 (02:42:AC:12:00:02)_______________

 ═ ETHERNET                                                 <Show>
 ═ 802.1X SECURITY                                          <Show>

 ═ IPv4 CONFIGURATION <Disabled>                            <Show>
 ═ IPv6 CONFIGURATION <Disabled>                            <Show>

 [X] Automatically connect
 [X] Available to all users

                                                      <Cancel> <OK>
```

2. **Time Zone**: Set the host time zone to UTC using the command 'sudo timedatectl set-timezone UTC'. Alternatively, set it to any other time zone with a valid source of synchronization. The system requires a valid NTP server.

```
sudo nano /etc/systemd/timesyncd.conf (add at the end NTP=valid ntp server)

sudo systemctl restart systemd-timesyncd
```

For example:



```
[escalation@escalation-SENSOR5:~$ date
Wed Nov  6 12:03:55 CET 2024
[escalation@escalation-SENSOR5:~$ sudo docker exec -it b69160a4f717 /bin/bash
[bash-5.0# date
Wed Nov  6 11:04:01 UTC 2024
bash-5.0#
```