Docker Forensics - HackTricks

Docker Forensics

Container modification

There are suspicions that some docker container was compromised:

docker ps

CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES

cc03e43a052a lamp-wordpress "./run.sh" 2 minutes ago Up 2 minutes 80/tcp wordpress

You can easily **find the modifications done to this container with regards to the image** with:

docker diff wordpress

C /var

C /var/lib

C /var/lib/mysql

A /var/lib/mysql/ib_logfile0

A /var/lib/mysql/ib_logfile1

A /var/lib/mysql/ibdata1

A /var/lib/mysql/mysql

A /var/lib/mysql/mysql/time_zone_leap_second.MYI

A /var/lib/mysql/mysql/general_log.CSV

...

In the previous command **C** means **Changed** and **A, Added**. If you find that some interesting file like `/etc/shadow` was modified you can download it from the container to check for malicious activity with:

docker cp wordpress:/etc/shadow.

You can also **compare it with the original one** running a new container and extracting the file from it:

docker run -d lamp-wordpress

docker cp b5d53e8b468e:/etc/shadow original_shadow #Get the file from the newly created container

diff original_shadow shadow

If you find that **some suspicious file was added** you can access the container and check it:

docker exec -it wordpress bash

Images modifications

When you are given an exported docker image (probably in `.tar` format) you can use **container-diff** to **extract a summary of the modifications**:

docker save <image> > image.tar #Export the image to a .tar file

container-diff analyze -t sizelayer image.tar

container-diff analyze -t history image.tar

container-diff analyze -t metadata image.tar

Then, you can **decompress** the image and **access the blobs** to search for suspicious files you may have found in the changes history:

tar -xf image.tar

Basic Analysis

You can get **basic information** from the image running:

docker inspect <image>

You can also get a summary **history of changes** with:

docker history --no-trunc <image>

You can also generate a **dockerfile from an image** with:

alias dfimage="docker run -v /var/run/docker.sock:/var/run/docker.sock --rm alpine/dfimage"

dfimage -sV=1.36 madhuakula/k8s-goat-hidden-in-layers>

Dive

In order to find added/modified files in docker images you can also use the **dive** (download it from **releases**) utility:

#First you need to load the image in your docker repo

sudo docker load < image.tar 1

Loaded image: flask:latest

#And then open it with dive:

sudo dive flask:latest

This allows you to **navigate through the different blobs of docker images** and check which files were modified/added. **Red** means added and **yellow** means modified. Use **tab** to move to the other view and **space** to collapse/open folders.

With die you won't be able to access the content of the different stages of the image. To do so you will need to **decompress each layer and access it**. You can decompress all the layers from an image from the directory where the image was decompressed executing:

tar -xf image.tar

for d in `find * -maxdepth 0 -type d`; do cd $d; tar -xf ./layer.tar; cd ..; done

Credentials from memory

Note that when you run a docker container inside a host **you can see the processes running on the container from the host** just running `ps -ef`

Therefore (as root) you can **dump the memory of the processes** from the host and search for **credentials** just **like in the following example**.

.