Penetration Testing with Kali Linux as a Docker Container

thenewstack.io/penetration-testing-with-kali-linux-as-a-docker-container

Jack Wallen November 11, 2023

Feature image courtesy of Kali Linux.

Penetration testing is a requirement for so many businesses. After all, you're going to need to know if your systems have vulnerabilities, so they can be mitigated as quickly as possible. One the if best ways to do this is to attempt to break into the system itself.

One of the most widely used pen testing platforms on the market is Kali Linux. With this Linux distribution, you have a plethora of tools at your disposal.

But what if you want to be able to run penetration testing without having to install a full-blown operating system? And if your security staff (or admins) have at least a fundamental understanding of Docker containers, they could always deploy Kali Linux as a Docker container and run penetration testing from within a headless container.

It's actually quite a fascinating and flexible method of penetration testing and I'm going to show you how to make it happen.

What You Need

I'll demonstrate this process using a Ubuntu Server 22.04 base. You can pull this off with any operating system that supports Docker. If you opt for an OS other than Ubuntu (or a derivative), you'll need to alter the Docker installation commands. You'll also need a user with sudo privileges.

That's it. Let's make some pen-testing magic.

How to Install Docker

The first thing we must do is install Docker. If you've already installed the container runtime, feel free to skip to the next section.

TRENDING STORIES

Before we can install Docker, we must download and add the official Docker GPG key. This is completed with the following command:

```
1 curl-fsSL https://download.docker.com/linux/ubuntu/gpg|sudo gpg--dearmor-o/usr/share/keyrings/docker-archive-keyring.gpg
```

One more step before the install is the adding of the official Docker repository, that will allow us to install Docker CE. Add the repository with the command:

view raw gistfile1.txt hosted with ❤ by GitHub
Next, we'll install a few basic dependencies:

```
1 sudo apt-get install apt-transport-https ca-certificates curl gnupg lsb-release-y
```

You can now update apt with the following:

```
1 sudo apt-get update
```

Finally, install Docker CE with the following command:

```
1 sudo apt-get install docker-ce docker-ce-cli containerd.io-y
```

In order to manage Docker without using sudo (which is a security issue), you must add your user to the docker group using the command:

```
1 sudo usermod-aG docker$USER
```

Log out and log back in so the changes take effect.

Pulling the Kali Linux Image

We can now pull the official Kali Linux image. This is done with the command:

```
1 docker pull kalilinux/kali-rolling
```

Deploy the Kali Linux Container

With the image pulled, we can now deploy the Kali Linux container with the command:

```
1 docker run-ti kalilinux/kali-rolling/bin/bash
```

This command will not only deploy the container but it will also land us on the Kali Linux bash prompt, so we can start working with the newly-deployed container.

Install Kali Headless

The Kali Linux image doesn't actually ship with any pen-testing tools. For that, we have to install a specific package.

Before we can install the required software, we must first update apt with the command:

```
1 apt update
```

After the update, install the package with:

```
1 apt install kali-linux-headless-y
```

During the installation, you'll be asked a few questions, each of which could be unique to your situation. Pay attention to those questions and answer either yes or no for each. This installation will take some time (between 5-20 minutes, depending on the speed of your hardware and network connection).

When this installation completes, you'll need to exit the running container and issue a commit to save the changes to the Kali container, so it can be reused without having to walk through the installation process again.

This next step is important. If you simply exit from the container (using the exit command) you'll lose all of your work and will have to go through the kali-linux-headless install once again. That's not an efficient way of working. Instead, SSH into the hosting machine from a different desktop, while remaining within the Kali container on the original login. From that new login, locate the container ID for the running Kali Linux container with the command:

```
1 docker ps
```

Using the first four characters of the running container's ID, commit the changes with a command like:

```
1 docker commit IDkalitools
```

Where ID is the first four digits of the container ID. You can also name the image anything you like. I used kalitools as an example.

Now that you've created the new image (which includes all of the headless pen testing tools), you can exit out of the original container. Verify the new image was created with:

```
1 docker images
```

You should see an image named kalitools (or whatever you've chosen to name it). You can then deploy a Kali Linux container from the new image (which includes all of the Kali Linux headless tools) with a command like:

```
1 docker run-it kalitools/bin/bash
```

You'll once again find yourself at the Kali Linux container bash prompt, where you can start running your penetration testing, all from within the convenience of a Docker container. When you exit out of the container this time, you'll still have the Kali Linux container image that contains all the tools. To run more penetration testing, you simply have to deploy another container with the above command (*docker run -it kalitools /bin/bash*) and have at the testing.

Jack Wallen is what happens when a Gen Xer mind-melds with present-day snark. Jack is a seeker of truth and a writer of words with a quantum mechanical pencil and a disjointed beat of sound and soul. Although he resides...