Hacking and Securing Docker Containers (2019 Edition) - Additional Documentation

Lab setup: This section is to accompany the lecture “Lab setup”

Step 1: Install Ubuntu 18.04 Desktop Virtual Machine and launch a terminal after logging in. Next, run the following command.

```
dev@dev:~$ sudo apt update
Hit:1 http://security.ubuntu.com/ubuntu bionic-security InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu bionic InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu bionic-updates InRelease
Get:4 http://us.archive.ubuntu.com/ubuntu bionic-backports InRelease [74.6 kB]
Get:5 http://us.archive.ubuntu.com/ubuntu bionic-backports/universe amd64 DEP-11 Metadata [7,224 B]
Fetched 70.8 kB in 12s (6,085 B/s)
Reading package lists... Done
Building dependency tree
Reading state information... Done
376 packages can be upgraded. Run 'apt list --upgradable' to see them.
dev@dev:~$
```

Step 2: Install Docker as shown in the following screenshot.

```
dev@dev:~$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  bridge-utils cgroupfs-mount containerd git git-man liberror-perl pigz runc ubuntu-fan
Suggested packages:
  aufs-tools btrfs-progs debootstrap docker-doc rinse zfs-fuse | zfsutils git-daemon-run | git-daemon-sysvinit g
  gitweb git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  bridge-utils cgroupfs-mount containerd docker.io git git-man liberror-perl pigz runc ubuntu-fan
0 upgraded, 10 newly installed, 0 to remove and 376 not upgraded.
Need to get 56.9 MB of archives.
After this operation, 291 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us.archive.ubuntu.com/ubuntu bionic/universe amd64 pigz amd64 2.4-1 [57.4 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 bridge-utils amd64 1.5-15ubuntu1 [30.1 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu bionic/universe amd64 cgroupfs-mount all 1.4 [6,320 B]
Get:4 http://us.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 runc amd64 1.0.0~rc7+git20190403.029124d
Get:5 http://us.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 containerd amd64 1.2.6-0ubuntu1~18.04.1
Get:6 http://us.archive.ubuntu.com/ubuntu bionic-updates/universe amd64 docker.io amd64 18.09.5-0ubuntu1~18.04.2
Get:7 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 liberror-perl all 0.17025-1 [22.8 kB]
Get:8 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git-man all 1:2.17.1-1ubuntu0.4 [803 kB]
Get:9 http://us.archive.ubuntu.com/ubuntu bionic-updates/main amd64 git amd64 1:2.17.1-1ubuntu0.4 [3,907 kB]
Get:10 http://us.archive.ubuntu.com/ubuntu bionic/main amd64 ubuntu-fan all 0.12.10 [34.7 kB]
Fetched 56.9 MB in 1min 27s (658 kB/s)
Preconfiguring packages ...
Selecting previously unselected package pigz.
(Reading database ... 125146 files and directories currently installed.)
Preparing to unpack .../0-pigz_2.4-1_amd64.deb ...
Unpacking pigz (2.4-1) ...
Selecting previously unselected package bridge-utils.
Preparing to unpack .../1-bridge-utils_1.5-15ubuntu1_amd64.deb ...
Unpacking bridge-utils (1.5-15ubuntu1) ...
Selecting previously unselected package cgroupfs-mount.
Preparing to unpack .../2-cgroupfs-mount_1.4_all.deb ...
Unpacking cgroupfs-mount (1.4) ...
```

Step 3: In most of the sections of the course, we want to run docker commands without "sudo". To be able to run docker commands without sudo, run the following commands as shown in the screenshot below.

```
dev@dev:~$ sudo groupadd docker
groupadd: group 'docker' already exists
dev@dev:~$
dev@dev:~$ sudo usermod -aG docker $USER
dev@dev:~$
```

Step 4: Run the following commands to complete docker installation.

```
dev@dev:~$ sudo systemctl start docker
dev@dev:~$
dev@dev:~$
dev@dev:~$ sudo systemctl enable docker
Synchronizing state of docker.service with SysV service script with /lib/systemd/systemd-sysv-install.
Executing: /lib/systemd/systemd-sysv-install enable docker
dev@dev:~$
```

Step 5: If you run any docker commands at this moment, it will not work and you will see an error as shown below. A restart is required for the changes to take effect.

```
dev@dev:~$ docker images
Got permission denied while trying to connect to the Docker daemon socket at
unix:///var/run/docker.sock: Get http://%2Fvar%2Frun%2Fdocker.sock/v1.39/imag
es/json: dial unix /var/run/docker.sock: connect: permission denied
dev@dev:~$
```

Step 6: Restart the Virtual Machine and run the following command. It should work without any errors as shown below.

```
dev@dev:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
dev@dev:~$
```

Step 7: Now, let us verify if we can pull a docker image by running the following command. This command downloads alpine image onto your local machine.

```
dev@dev:~$ docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
921b31ab772b: Pull complete
Digest: sha256:ca1c944a4f8486a153024d9965aafbe24f5723c1d5c02f4964c045a16d19dc54
Status: Downloaded newer image for alpine:latest
dev@dev:~$
```

Step 8: Next, start a container using the alpine image we just downloaded. This can be done as shown below.

```
dev@dev:~$ docker run -itd alpine
723d0333b1af6c913b145c74e40fa82ea15c0209eea6a109afc7c91a59c2cfd6
dev@dev:~$
```

Step 9: The following commands show how we can interact with the running container.

```
dev@dev:~$ docker ps -aq
723d0333b1af
dev@dev:~$
dev@dev:~$
dev@dev:~$ docker exec -it 723d0333b1af sh
/ # id
uid=0(root) gid=0(root) groups=0(root),1(bin)
/ #
```

Building a docker image: This section is to accompany the lectures “Building your First Docker Image” and “Running your First Docker Container”

Step 1: The following figure shows the Dockerfile we are going to use for building the Docker image

```
dev@dev:~/Desktop/web$ ls
Dockerfile  index.html
dev@dev:~/Desktop/web$
dev@dev:~/Desktop/web$ cat Dockerfile
FROM ubuntu:16.04

RUN apt-get update -y
RUN apt-get install -y apache2
RUN chown -R www-data:www-data /var/www/

ENV APACHE_RUN_USER www-data
ENV APACHE_RUN_GROUP www-data
ENV APACHE_LOG_DIR /var/log/apache2
ENV APACHE_LOCK_DIR /var/lock/apache2
ENV APACHE_PID_FILE /var/run/apache2.pid

EXPOSE 80
ENTRYPOINT ["/usr/sbin/apache2ctl"]
CMD ["-D","FOREGROUND"]
dev@dev:~/Desktop/web$
```

Step 2: Run the following command to build a new docker image with the custom web application.

```
dev@dev:~/Desktop/web$ docker build -t webserver:latest .
Sending build context to Docker daemon  3.072kB
Step 1/12 : FROM ubuntu:16.04
 ---> 13c9f1285025
Step 2/12 : RUN apt-get update -y
 ---> Using cache
 ---> 4d2cce3a3370
Step 3/12 : RUN apt-get install -y apache2
 ---> Using cache
 ---> 08fdaca6481e
Step 4/12 : RUN chown -R www-data:www-data /var/www/
 ---> Using cache
 ---> 6576c06aac43
Step 5/12 : ENV APACHE_RUN_USER www-data
 ---> Using cache
 ---> 4c595d8da1d7
Step 6/12 : ENV APACHE_RUN_GROUP www-data
 ---> Using cache
 ---> 7e2156a462d9
Step 7/12 : ENV APACHE_LOG_DIR /var/log/apache2
 ---> Using cache
 ---> 5f87f309e2e9
Step 8/12 : ENV APACHE_LOCK_DIR /var/lock/apache2
 ---> Using cache
 ---> 132319f38749
Step 9/12 : ENV APACHE_PID_FILE /var/run/apache2.pid
 ---> Using cache
 ---> 9246d09bb84c
Step 10/12 : EXPOSE 80
 ---> Using cache
 ---> 72a1c4e21318
Step 11/12 : ENTRYPOINT ["/usr/sbin/apache2ctl"]
 ---> Using cache
 ---> 4d774a9194b8
Step 12/12 : CMD ["-D","FOREGROUND"]
 ---> Using cache
 ---> 3f066cba41cb
Successfully built 3f066cba41cb
Successfully tagged webserver:latest
dev@dev:~/Desktop/web$
```

Step 3: Run the following command to start a container. “**docker ps**” command shows that the container is running fine.

```
dev@dev:~/Desktop/web$
dev@dev:~/Desktop/web$ docker run -itd -p 8080:80 webserver:latest
31d52c2017ad2d36b4ba66292cdca92f0f454006fce38fbc2ea69283101a7bc5
dev@dev:~/Desktop/web$
dev@dev:~/Desktop/web$ docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS              PORTS                  NAMES
31d52c2017ad        webserver:latest    "/usr/sbin/apache2ct…"   5 seconds ago       Up 4 seconds        0.0.0.0:8080->80/tcp   practical_stonebraker
723d0333b1af        alpine              "/bin/sh"                39 minutes ago      Up 39 minutes                              agitated_shtern
dev@dev:~/Desktop/web$
dev@dev:~/Desktop/web$
```

Step 4: Interacting with the newly started container.

```
dev@dev:~/Desktop/web$ docker exec -it 31d52c2017ad /bin/bash
root@31d52c2017ad:/#
root@31d52c2017ad:/# id
uid=0(root) gid=0(root) groups=0(root)
root@31d52c2017ad:/#
root@31d52c2017ad:/#
```

Note: python3 comes preinstalled with Ubuntu 18.04

```
dev@dev:~$ python3 --version
Python 3.6.7
dev@dev:~$
```