ATTACK DEFENSE

by PentesterAcademy

Name	Docker Bench Security
URL	https://attackdefense.com/challengedetails?cid=1607
Type	DevSecOps : Docker Security Tools

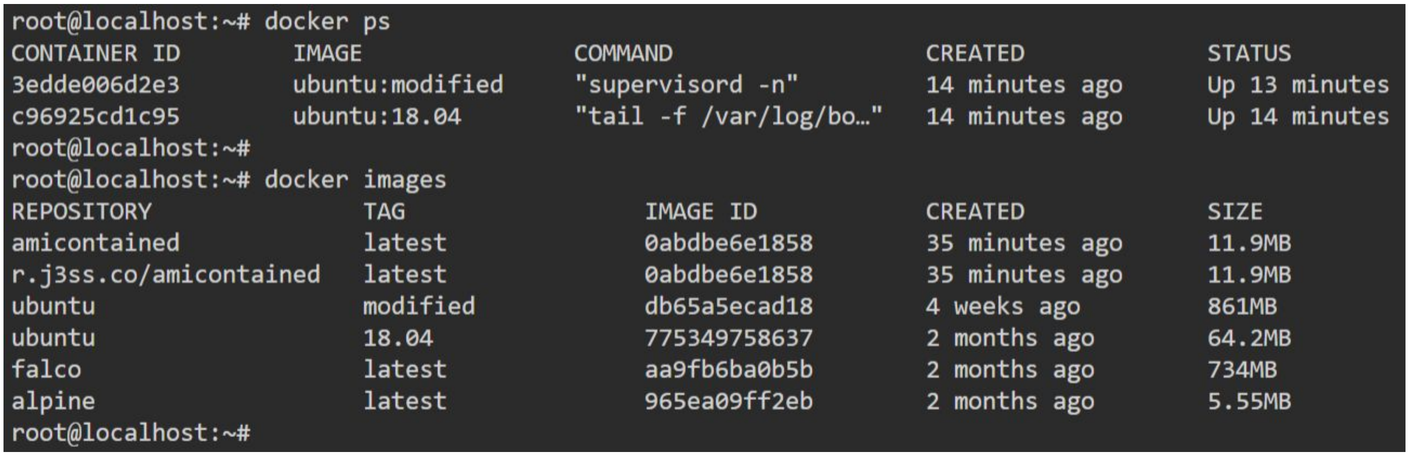
Important Note: This document illustrates all the important steps required to complete this lab. This is by no means a comprehensive step-by-step solution for this exercise. This is only provided as a reference to various commands needed to complete this exercise and for your further research on this topic. Also, note that the IP addresses and domain names might be different in your lab.

Objective: Run Docker Bench Security, locate the misconfigurations and fix those!

Solution:

Step 1: Check running containers and images present on the local machine.

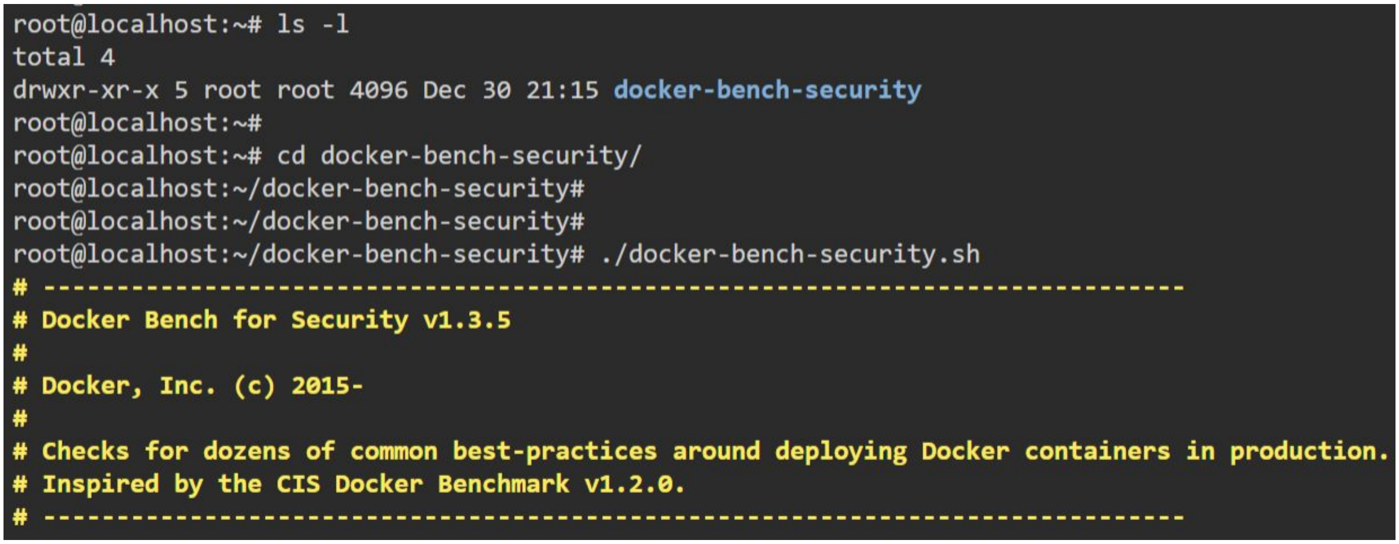
Command:
docker ps
docker images



```
root@localhost:~# docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED             STATUS
3edde006d2e3        ubuntu:modified     "supervisord -n"         14 minutes ago      Up 13 minutes
c96925cd1c95        ubuntu:18.04        "tail -f /var/log/bo…"   14 minutes ago      Up 14 minutes
root@localhost:~#
root@localhost:~# docker images
REPOSITORY               TAG                 IMAGE ID            CREATED             SIZE
amicontained             latest              0abdbe6e1858        35 minutes ago      11.9MB
r.j3ss.co/amicontained   latest              0abdbe6e1858        35 minutes ago      11.9MB
ubuntu                   modified            db65a5ecad18        4 weeks ago         861MB
ubuntu                   18.04               775349758637        2 months ago        64.2MB
falco                    latest              aa9fb6ba0b5b        2 months ago        734MB
alpine                   latest              965ea09ff2eb        2 months ago        5.55MB
root@localhost:~#
```

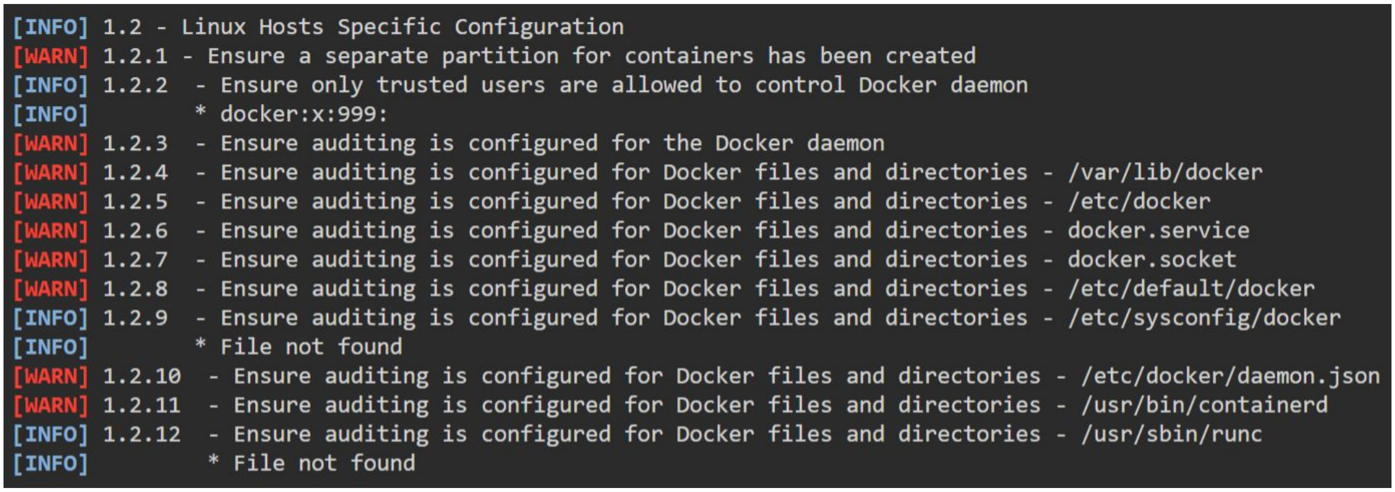
Step 2: Check the contents of the /root directory, change to docker bench security and run the script.

Commands:
ls -l
cd docker-bench-security
./docker-bench-security.sh

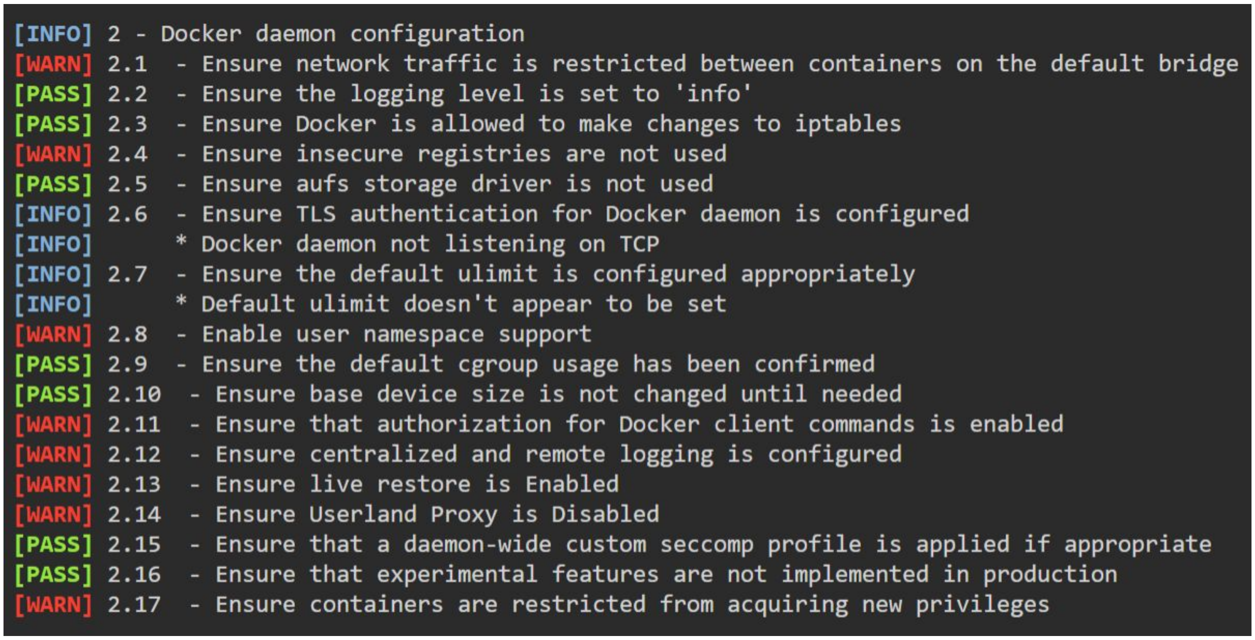


```
root@localhost:~# ls -l
total 4
drwxr-xr-x 5 root root 4096 Dec 30 21:15 docker-bench-security
root@localhost:~#
root@localhost:~# cd docker-bench-security/
root@localhost:~/docker-bench-security#
root@localhost:~/docker-bench-security#
root@localhost:~/docker-bench-security# ./docker-bench-security.sh
# ------------------------------------------------------------------------------
# Docker Bench for Security v1.3.5
#
# Docker, Inc. (c) 2015-
#
# Checks for dozens of common best-practices around deploying Docker containers in production.
# Inspired by the CIS Docker Benchmark v1.2.0.
# ------------------------------------------------------------------------------
```

The script checks for different aspects/configurations and provide waning/information on different aspects..

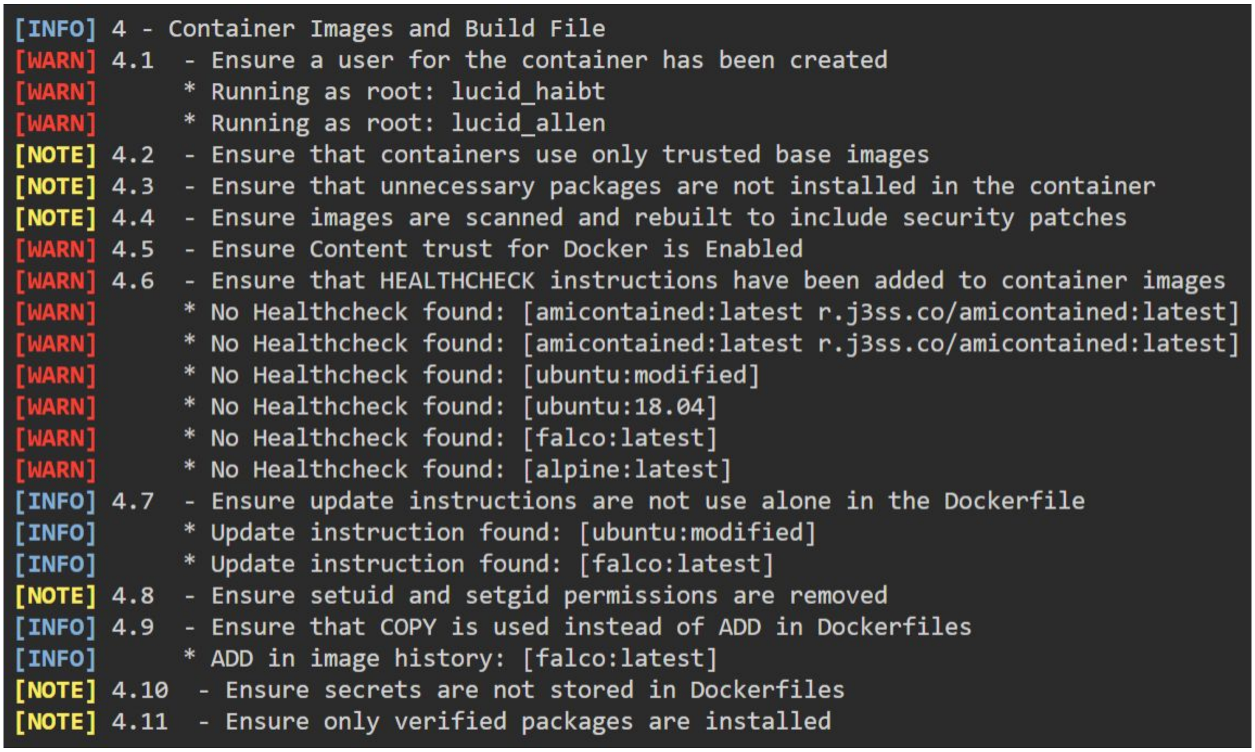


```
[INFO] 1.2 - Linux Hosts Specific Configuration
[WARN] 1.2.1 - Ensure a separate partition for containers has been created
[INFO] 1.2.2  - Ensure only trusted users are allowed to control Docker daemon
[INFO]        * docker:x:999:
[WARN] 1.2.3  - Ensure auditing is configured for the Docker daemon
[WARN] 1.2.4  - Ensure auditing is configured for Docker files and directories - /var/lib/docker
[WARN] 1.2.5  - Ensure auditing is configured for Docker files and directories - /etc/docker
[WARN] 1.2.6  - Ensure auditing is configured for Docker files and directories - docker.service
[WARN] 1.2.7  - Ensure auditing is configured for Docker files and directories - docker.socket
[WARN] 1.2.8  - Ensure auditing is configured for Docker files and directories - /etc/default/docker
[INFO] 1.2.9  - Ensure auditing is configured for Docker files and directories - /etc/sysconfig/docker
[INFO]        * File not found
[WARN] 1.2.10  - Ensure auditing is configured for Docker files and directories - /etc/docker/daemon.json
[WARN] 1.2.11  - Ensure auditing is configured for Docker files and directories - /usr/bin/containerd
[INFO] 1.2.12  - Ensure auditing is configured for Docker files and directories - /usr/sbin/runc
[INFO]         * File not found
```



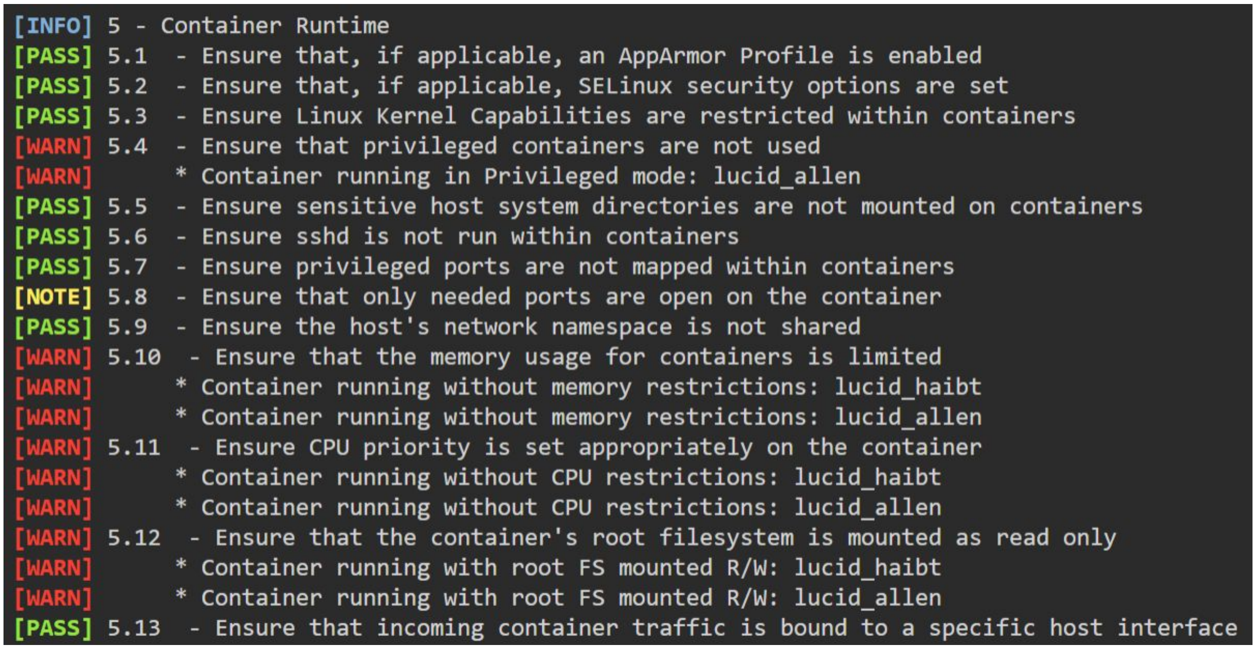
```
[INFO] 2 - Docker daemon configuration
[WARN] 2.1  - Ensure network traffic is restricted between containers on the default bridge
[PASS] 2.2  - Ensure the logging level is set to 'info'
[PASS] 2.3  - Ensure Docker is allowed to make changes to iptables
[WARN] 2.4  - Ensure insecure registries are not used
[PASS] 2.5  - Ensure aufs storage driver is not used
[INFO] 2.6  - Ensure TLS authentication for Docker daemon is configured
[INFO]      * Docker daemon not listening on TCP
[INFO] 2.7  - Ensure the default ulimit is configured appropriately
[INFO]      * Default ulimit doesn't appear to be set
[WARN] 2.8  - Enable user namespace support
[PASS] 2.9  - Ensure the default cgroup usage has been confirmed
[PASS] 2.10  - Ensure base device size is not changed until needed
[WARN] 2.11  - Ensure that authorization for Docker client commands is enabled
[WARN] 2.12  - Ensure centralized and remote logging is configured
[WARN] 2.13  - Ensure live restore is Enabled
[WARN] 2.14  - Ensure Userland Proxy is Disabled
[PASS] 2.15  - Ensure that a daemon-wide custom seccomp profile is applied if appropriate
[PASS] 2.16  - Ensure that experimental features are not implemented in production
[WARN] 2.17  - Ensure containers are restricted from acquiring new privileges
```

The above screenshot shows that the logging, userland proxy and user-remap etc are not enabled.



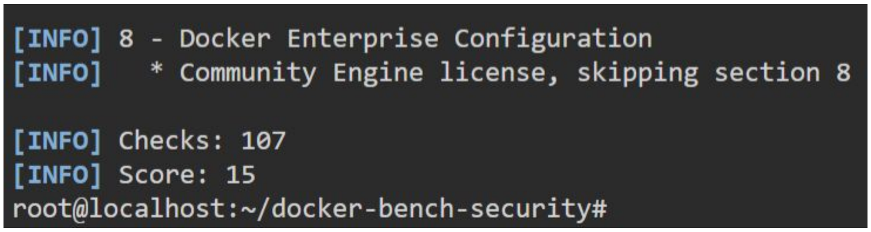
```
[INFO] 4 - Container Images and Build File
[WARN] 4.1  - Ensure a user for the container has been created
[WARN]      * Running as root: lucid_haibt
[WARN]      * Running as root: lucid_allen
[NOTE] 4.2  - Ensure that containers use only trusted base images
[NOTE] 4.3  - Ensure that unnecessary packages are not installed in the container
[NOTE] 4.4  - Ensure images are scanned and rebuilt to include security patches
[WARN] 4.5  - Ensure Content trust for Docker is Enabled
[WARN] 4.6  - Ensure that HEALTHCHECK instructions have been added to container images
[WARN]      * No Healthcheck found: [amicontained:latest r.j3ss.co/amicontained:latest]
[WARN]      * No Healthcheck found: [amicontained:latest r.j3ss.co/amicontained:latest]
[WARN]      * No Healthcheck found: [ubuntu:modified]
[WARN]      * No Healthcheck found: [ubuntu:18.04]
[WARN]      * No Healthcheck found: [falco:latest]
[WARN]      * No Healthcheck found: [alpine:latest]
[INFO] 4.7  - Ensure update instructions are not use alone in the Dockerfile
[INFO]      * Update instruction found: [ubuntu:modified]
[INFO]      * Update instruction found: [falco:latest]
[NOTE] 4.8  - Ensure setuid and setgid permissions are removed
[INFO] 4.9  - Ensure that COPY is used instead of ADD in Dockerfiles
[INFO]      * ADD in image history: [falco:latest]
[NOTE] 4.10  - Ensure secrets are not stored in Dockerfiles
[NOTE] 4.11  - Ensure only verified packages are installed
```

The above screenshot shows the Docker images which have no health check routine, a health check routines allows docker to see if the container is running properly.



```
[INFO] 5 - Container Runtime
[PASS] 5.1  - Ensure that, if applicable, an AppArmor Profile is enabled
[PASS] 5.2  - Ensure that, if applicable, SELinux security options are set
[PASS] 5.3  - Ensure Linux Kernel Capabilities are restricted within containers
[WARN] 5.4  - Ensure that privileged containers are not used
[WARN]      * Container running in Privileged mode: lucid_allen
[PASS] 5.5  - Ensure sensitive host system directories are not mounted on containers
[PASS] 5.6  - Ensure sshd is not run within containers
[PASS] 5.7  - Ensure privileged ports are not mapped within containers
[NOTE] 5.8  - Ensure that only needed ports are open on the container
[PASS] 5.9  - Ensure the host's network namespace is not shared
[WARN] 5.10  - Ensure that the memory usage for containers is limited
[WARN]      * Container running without memory restrictions: lucid_haibt
[WARN]      * Container running without memory restrictions: lucid_allen
[WARN] 5.11  - Ensure CPU priority is set appropriately on the container
[WARN]      * Container running without CPU restrictions: lucid_haibt
[WARN]      * Container running without CPU restrictions: lucid_allen
[WARN] 5.12  - Ensure that the container's root filesystem is mounted as read only
[WARN]      * Container running with root FS mounted R/W: lucid_haibt
[WARN]      * Container running with root FS mounted R/W: lucid_allen
[PASS] 5.13  - Ensure that incoming container traffic is bound to a specific host interface
```

The above screenshot shows the Docker containers running without cgroup restrictions, mounted root filesystem and privileged mode. These containers may lead to host compromise.



```
[INFO] 8 - Docker Enterprise Configuration
[INFO]   * Community Engine license, skipping section 8

[INFO] Checks: 107
[INFO] Score: 15
root@localhost:~/docker-bench-security#
```

Finally, the script provides a score for the system. The system administrator can then resolve the warnings shown by the script and improve the score.

References:

1. Docker (https://www.docker.com/)
2. Docker Bench Security (https://github.com/docker/docker-bench-security)