ATTACK DEFENSE

by PentesterAcademy

Name	Dockerfilelint
URL	https://attackdefense.com/challengedetails?cid=2162
Type	Docker Security: Dockerfile Linting

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.

Dockerfile Linting is a process to check and modify the Dockerfile as per the industry's best practices.

Dockerfilelint is a node module that analyzes a Dockerfile and looks for common traps, mistakes and helps enforce best practices.

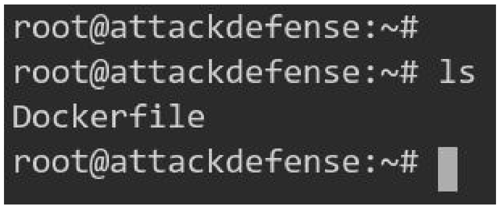
A Dockerfile is provided in the home directory of the root user (i.e. /root). Dockerfilelint is installed on the machine.

Objective: Analyze the Dockerfile with Dockerfilelint. Edit Dockerfile to remove all issues detected by Dockerfilelint!

Solution:

Step 1: List files present in current directory.

Command: ls

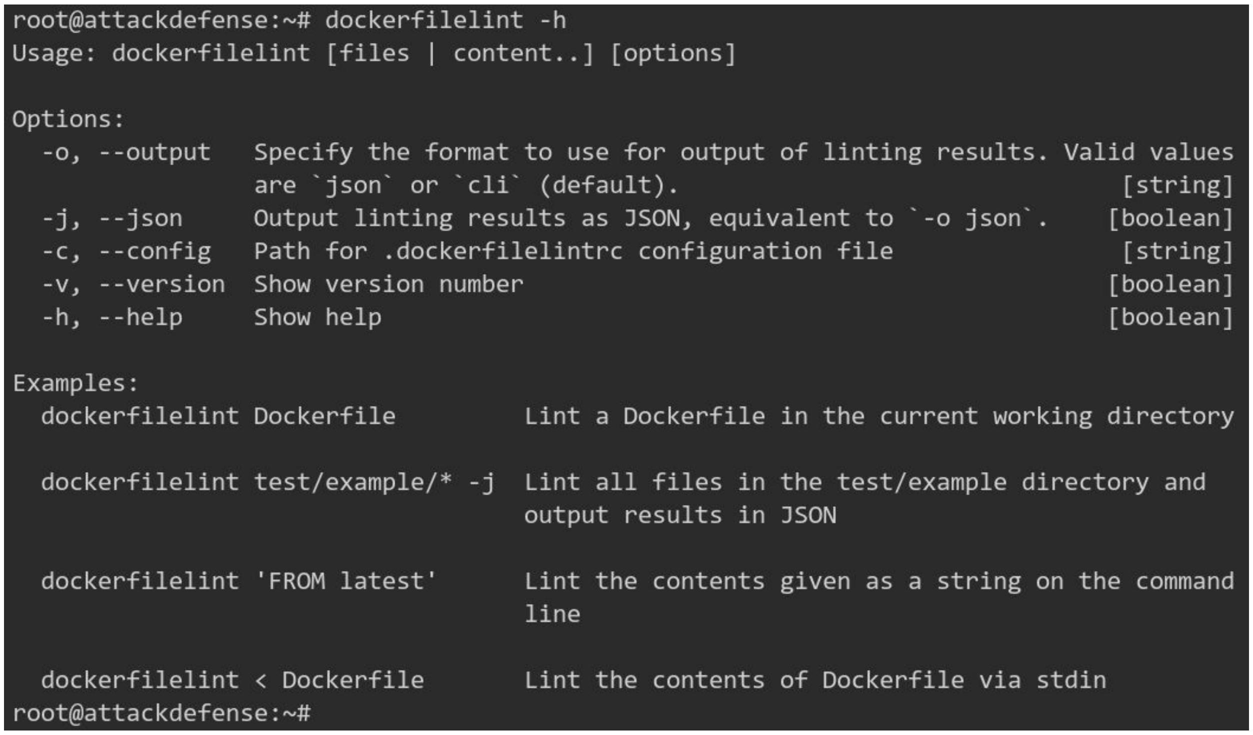


```
root@attackdefense:~#
root@attackdefense:~# ls
Dockerfile
root@attackdefense:~#
```

There is a Dockerfile present in the current directory.

Step 2: Check help menu for dockerfilelint.

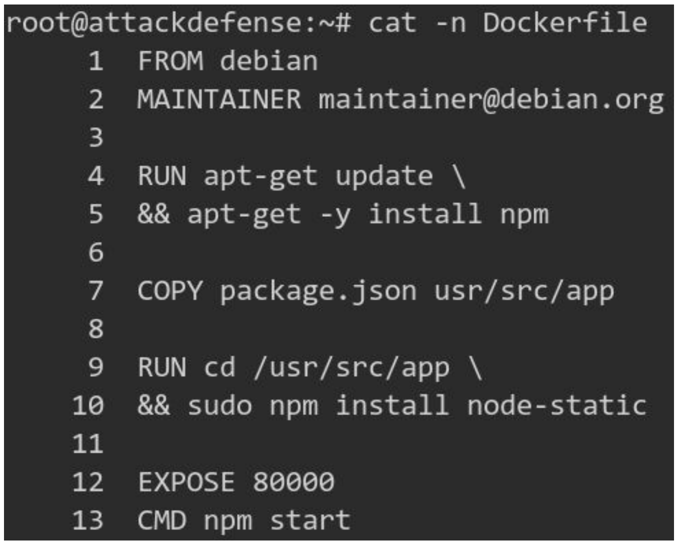
Command: dockerfilelint -h



```
root@attackdefense:~# dockerfilelint -h
Usage: dockerfilelint [files | content..] [options]

Options:
  -o, --output   Specify the format to use for output of linting results. Valid values
                 are `json` or `cli` (default).                             [string]
  -j, --json     Output linting results as JSON, equivalent to `-o json`.   [boolean]
  -c, --config   Path for .dockerfilelintrc configuration file              [string]
  -v, --version  Show version number                                        [boolean]
  -h, --help     Show help                                                  [boolean]

Examples:
  dockerfilelint Dockerfile         Lint a Dockerfile in the current working directory

  dockerfilelint test/example/* -j  Lint all files in the test/example directory and
                                    output results in JSON

  dockerfilelint 'FROM latest'      Lint the contents given as a string on the command
                                    line

  dockerfilelint < Dockerfile       Lint the contents of Dockerfile via stdin
root@attackdefense:~#
```

Step 3: Read the contents of Dockerfile.

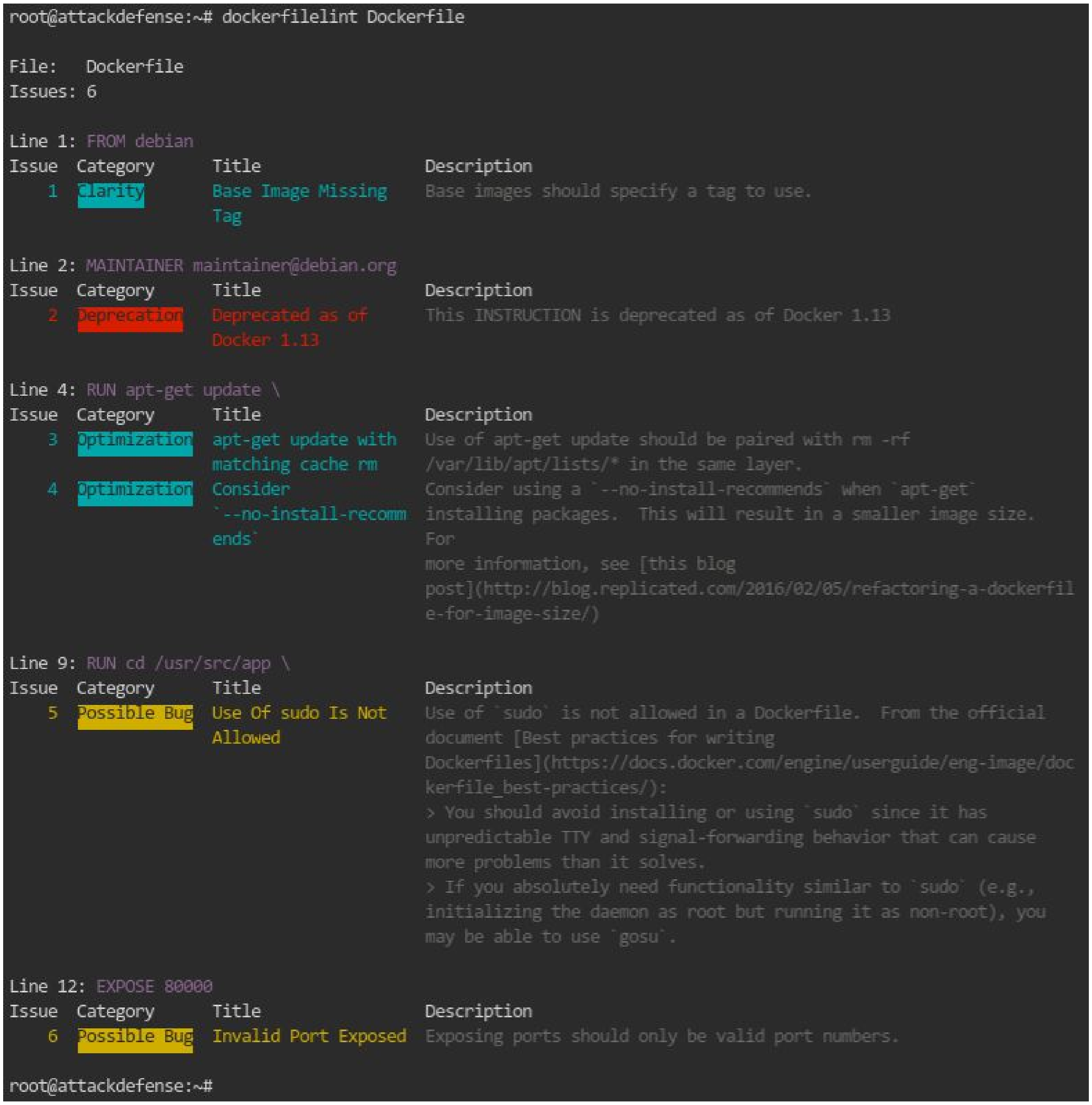
Command: cat -n Dockerfile



```
root@attackdefense:~# cat -n Dockerfile
     1  FROM debian
     2  MAINTAINER maintainer@debian.org
     3
     4  RUN apt-get update \
     5  && apt-get -y install npm
     6
     7  COPY package.json usr/src/app
     8
     9  RUN cd /usr/src/app \
    10  && sudo npm install node-static
    11
    12  EXPOSE 80000
    13  CMD npm start
```

Step 4: Run dockerfilelint on Dockerfile.

Command: dockerfilelint Dockerfile



```
root@attackdefense:~# dockerfilelint Dockerfile

File:   Dockerfile
Issues: 6

Line 1: FROM debian
Issue  Category      Title                 Description
    1  Clarity       Base Image Missing    Base images should specify a tag to use.
                     Tag

Line 2: MAINTAINER maintainer@debian.org
Issue  Category      Title                 Description
    2  Deprecation   Deprecated as of      This INSTRUCTION is deprecated as of Docker 1.13
                     Docker 1.13

Line 4: RUN apt-get update \
Issue  Category      Title                 Description
    3  Optimization  apt-get update with   Use of apt-get update should be paired with rm -rf
                     matching cache rm     /var/lib/apt/lists/* in the same layer.
    4  Optimization  Consider              Consider using a `--no-install-recommends` when `apt-get`
                     `--no-install-recomm  installing packages.  This will result in a smaller image size.
                     ends`                 For
                                           more information, see [this blog
                                           post](http://blog.replicated.com/2016/02/05/refactoring-a-dockerfil
                                           e-for-image-size/)

Line 9: RUN cd /usr/src/app \
Issue  Category      Title                 Description
    5  Possible Bug  Use Of sudo Is Not    Use of `sudo` is not allowed in a Dockerfile.  From the official
                     Allowed               document [Best practices for writing
                                           Dockerfiles](https://docs.docker.com/engine/userguide/eng-image/doc
                                           kerfile_best-practices/):
                                           > You should avoid installing or using `sudo` since it has
                                           unpredictable TTY and signal-forwarding behavior that can cause
                                           more problems than it solves.
                                           > If you absolutely need functionality similar to `sudo` (e.g.,
                                           initializing the daemon as root but running it as non-root), you
                                           may be able to use `gosu`.

Line 12: EXPOSE 80000
Issue  Category      Title                 Description
    6  Possible Bug  Invalid Port Exposed  Exposing ports should only be valid port numbers.

root@attackdefense:~#
```

Modify the Dockerfile to address the issues mentioned by dockerfilelint.
Please note that line numbers below are respective to unmodified Dockerfile.

Modifications:

Line 1: Specify tag for the base image used.
Before Modification: FROM debian
After Modification: FROM debian:9

Line 2: Use LABEL for specifying maintainer.
Before Modification: MAINTAINER maintainer@debian.org
After Modification: LABEL maintainer="maintainer@debian.org"

Line 5: Add --no-install-recommends flag to apt-get statement.
Before Modification: && apt-get -y install npm
After Modification: && apt-get -y --no-install-recommends install npm

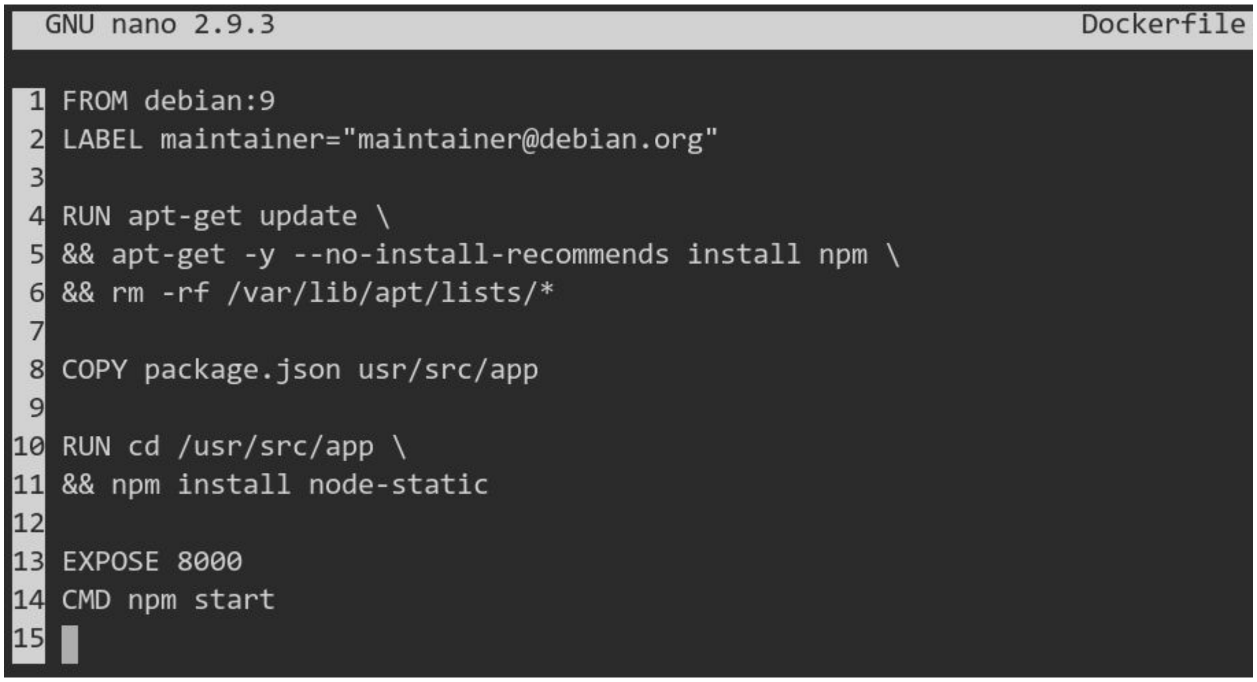
Line 5: Remove apt cache after installing packages.
Before Modification: && apt-get -y --no-install-recommends install npm
After Modification:
```
&& apt-get -y --no-install-recommends install npm \
&& rm -rf /var/lib/apt/lists/*
```

Line 10: Remove sudo from command.
Before Modification: && sudo npm install node-static
After Modification: && npm install node-static

Line 12: Use a valid port number.
Before Modification: EXPOSE 80000
After Modification: EXPOSE 8000

Step 5: Check the file in nano after applying the above mentioned modifications.

Command: nano -l Dockerfile

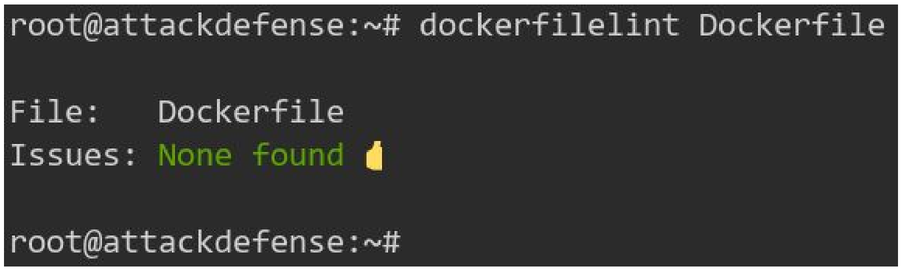


```
GNU nano 2.9.3                                                   Dockerfile

FROM debian:9
LABEL maintainer="maintainer@debian.org"

RUN apt-get update \
&& apt-get -y --no-install-recommends install npm \
&& rm -rf /var/lib/apt/lists/*

COPY package.json usr/src/app

RUN cd /usr/src/app \
&& npm install node-static

EXPOSE 8000
CMD npm start

```

Save the file and exit nano. Press ‘Ctrl + X’ followed by ‘Y’ and Enter to exit and save changes.

Step 6: Run dockerfilelint again on the modified Dockerfile.

Command: dockerfilelint Dockerfile



```
root@attackdefense:~# dockerfilelint Dockerfile

File:    Dockerfile
Issues:  None found 👍

root@attackdefense:~#
```

No issues were found in the Dockerfile after modification.

References:

- dockerfilelint (https://github.com/replicatedhq/dockerfilelint)