

Previous goals

1. Learn to parse CVE API (might use different API because the one we using is limited)
We learned to work with NATIONAL VULNERABILITY DATABASE
nvd.nist.gov
2. Check for secrets and passwords
Working on it
3. Check what ports are used and which ports are opened but not used
Checking assigned ports with actual running ports

Process of checking for vulnerabilities

Reading Dockerfile
For information
running software,
ports, users...

Checking Database
for potential
vulnerabilities

Showing results of
potential CVE results
and other information



Results of running Script right now

```
Type the path to Docker Container
/Users/timurzhunusov/Downloads/vulhub-master/httpd/CVE-2021-42013/
User User is not defined running as root!
No exposed ports
Ports Ports used by Docker right now:
CONTAINER ID    IMAGE           COMMAND          CREATED
4c5f5f06fc14    mongo:latest    "docker-entrypoint.s..." 20 hours ago

Running software: httpd 2.4.50
CVE Possible CVE Vulnerabilities:
Potential threat: CVE-2021-41773
Score ['V3', 7.5, 'HIGH']
URL https://nvd.nist.gov/vuln/detail/CVE-2021-41773
Potential threat: CVE-2021-42013
Score ['V3', 9.8, 'CRITICAL']
```

Not working if image pulled from docker hub repository

Find a way to pull image from repository and parse the log and history to understand what is running inside container from repository

```
(base) timurzhunusov@MacBook-Pro-Timur HW5 % docker pull node
(base) timurzhunusov@MacBook-Pro-Timur HW5 % docker history node
```

| IMAGE | CREATED | CREATED BY | SIZE |
|--------------|-------------|--|--------|
| cb9aad9080ca | 6 days ago | /bin/sh -c #(nop) CMD ["node"] | 0B |
| <missing> | 6 days ago | /bin/sh -c #(nop) ENTRYPOINT ["docker-entry... | 0B |
| <missing> | 6 days ago | /bin/sh -c #(nop) COPY file:4d192565a7220e13... | 388B |
| <missing> | 6 days ago | /bin/sh -c set -ex && for key in 6A010... | 7.6MB |
| <missing> | 6 days ago | /bin/sh -c #(nop) ENV YARN_VERSION=1.22.19 | 0B |
| <missing> | 6 days ago | /bin/sh -c ARCH= && dpkgArch="\$(dpkg --print... | 151MB |
| <missing> | 6 days ago | /bin/sh -c #(nop) ENV NODE_VERSION=19.0.1 | 0B |
| <missing> | 2 weeks ago | /bin/sh -c groupadd --gid 1000 node && use... | 334kB |
| <missing> | 2 weeks ago | /bin/sh -c set -ex; apt-get update; apt-ge... | 529MB |
| <missing> | 2 weeks ago | /bin/sh -c apt-get update && apt-get install... | 152MB |
| <missing> | 2 weeks ago | /bin/sh -c set -ex; if ! command -v gpg > /... | 19MB |
| <missing> | 2 weeks ago | /bin/sh -c set -eux; apt-get update; apt-g... | 10.7MB |

Check whether secrets are protected properly

- Secrets are usually stored in environment variables, but this is not recommend because for docker container, environment variables are not secure.
- We write codes to check whether secrets are in environment variables.

```

1  import os
2  import subprocess
3
4  os.chdir('desktop/DockerFile')
5  result = subprocess.check_output('docker inspect secretsimage', shell=True)
6
7  output = result.decode()
8
9  if 'PASSWORD' in output:
10     print('secrets are not protect well')
11

```

If sensitive data such as password are stored in the environment variables, it will display secrets are not protected well.

```

},
  "Tty": false,
  "OpenStdin": false,
  "StdinOnce": false,
  "Env": [
    "PATH=/usr/local/sbin:/usr/local/bin:",
    "MYSQL_UNIX_PORT=/var/lib/mysql/mysql",
    "MYSQL_ROOT_PASSWORD=toor",
    "MYSQL_DATABASE=users",
    "MYSQL_USER=root",
    "MYSQL_PASSWORD=toor",
    "MYSQL_ROOT_HOST=mysql-db"
  ],
  "Cmd": [
    "mvnsald"
  ],
  "Metadata": {
    "LastTagTime": "2022-11-16T00:26:58.842769031Z"
  }
}
]

secrets are not protect well

```

Future goals

1. Adapt scrip for container images from web repositories
2. Check what ports are used and which ports are opened but not used