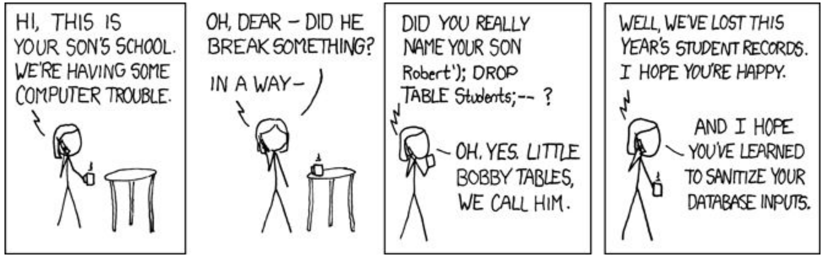
TIC4302
Group 5 - DVPWA

Nuri Irfani Binte Masri A0177806U
Zhang Mengtong A0194255Y

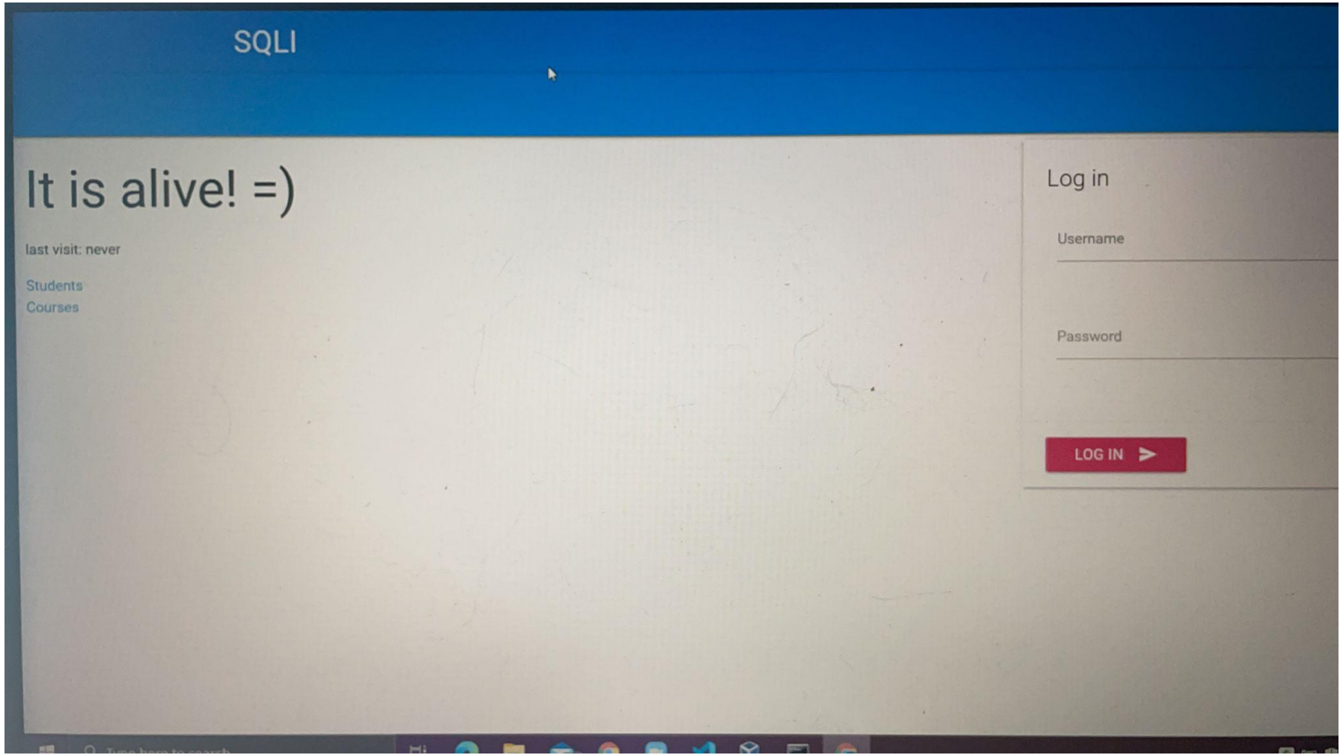
DVPWA? → **Damn Vulnerable Python Web Application**

DVWA

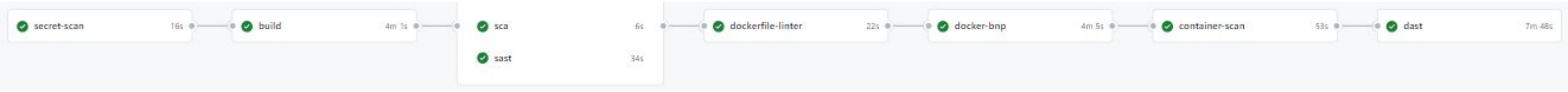
- ➔ Created by **anxolerd**
- ➔ Inspired by the well known DVWA & an XKCD comic
- ➔ Does not have a CI/CD pipeline defined

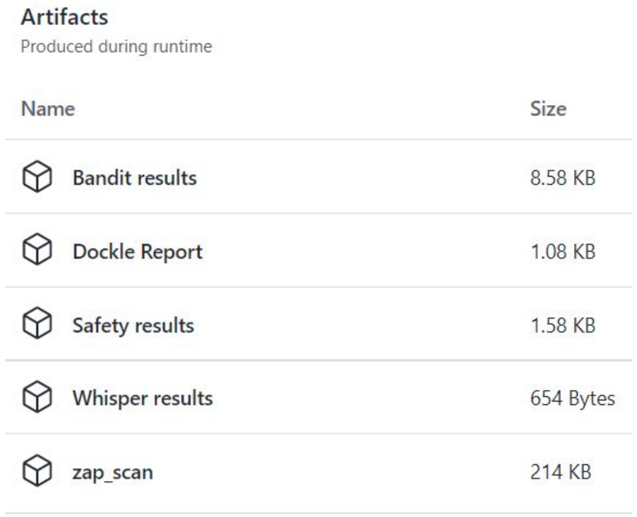

HI, THIS IS YOUR SON'S SCHOOL. WE'RE HAVING SOME COMPUTER TROUBLE.
OH, DEAR – DID HE BREAK SOMETHING?
IN A WAY–
DID YOU REALLY NAME YOUR SON Robert'); DROP TABLE Students;-- ?
OH. YES. LITTLE BOBBY TABLES, WE CALL HIM.
WELL, WE'VE LOST THIS YEAR'S STUDENT RECORDS. I HOPE YOU'RE HAPPY.
AND I HOPE YOU'VE LEARNED TO SANITIZE YOUR DATABASE INPUTS.

Why DVPWA? $\longrightarrow$ Damn Vulnerable Python Web Application



Pipeline and Artifacts





Artifacts

Produced during runtime

Name	Size
Bandit results	8.58 KB
Dockle Report	1.08 KB
Safety results	1.58 KB
Whisper results	654 Bytes
zap_scan	214 KB

On the pipeline: Secret Scan → Whispers, by Skyscanner

Skyscanner / **whispers** Public

- ➔ An open-source static code analysis tool designed to search for **hardcoded credentials and dangerous functions**.
- ➔ Parses **structured text** such as YAML, JSON, XML, npmrc, .pypirc, .htpasswd, .properties, pip.conf, conf / ini, Dockerfile, Shell scripts, and Python3 (as AST) as well as **declarations** and **assignment formats** for Javascript, Java, GO, and PHP

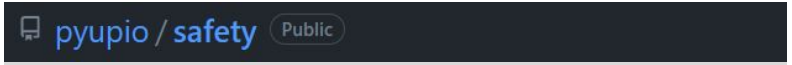
PROS

- ➔ supports a wide range of secret detection formats out of the box
  - ◆ Passwords, AWS keys, API Tokens, Sensitive files, Dangerous functions, etc.
- ➔ Includes a plug-in system that can be used to further extend its scanning capabilities to new file formats.

CONS

- ➔ By default has limited scanning rules; i.e llimited combination of regular expressions, Base64 and Ascii detection.
- ➔ Not a very detailed wiki

On the pipeline: SCA $\longrightarrow$ Safety, by Pyupio


pyupio / safety Public

- ➔ Checks installed dependencies for known security vulnerabilities and licensing issues.
- ➔ The package uses Safety DB, which is the company’s public open source vulnerability database.

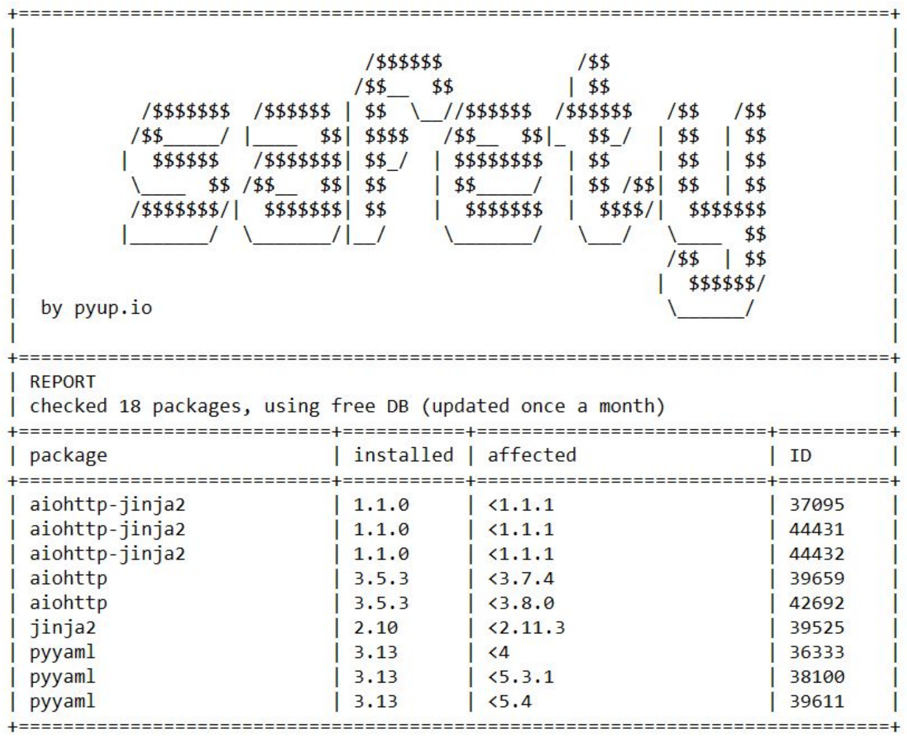
PROS

- ➔ Easy to install, manage and integrates with CI
- ➔ Docker image available

CONS

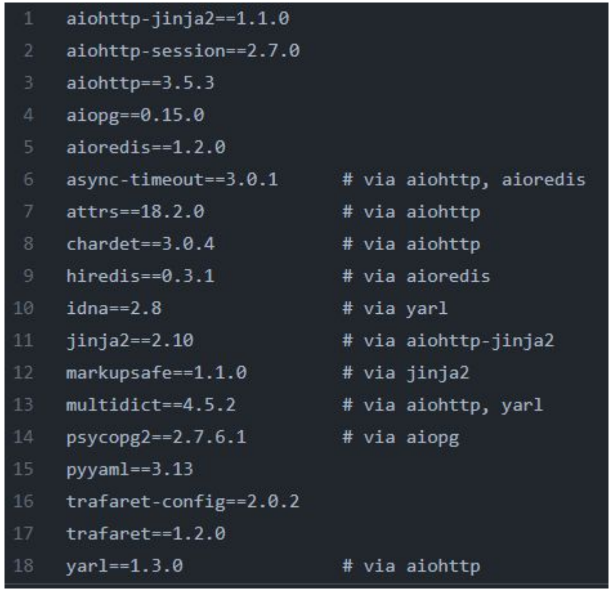
- ➔ The open-source Safety DB is only updated monthly. Paid version has more features, but very expensive:
  - ◆ the ability to sync database in a local system
  - ◆ provides Common Vulnerability Scoring System (CVSS)
- ➔ Outputs do not display severity of vulnerabilities, only ID

DVPWA: SCA

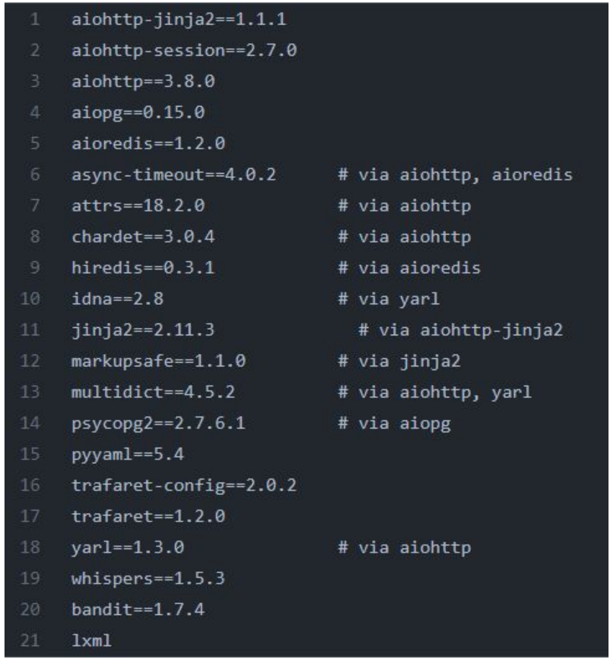


```
+==============================================================================+
|                                                                              |
|                               /$$$$$$            /$$                         |
|                              /$$__  $$          | $$                         |
|           /$$$$$$$  /$$$$$$ | $$  \__//$$$$$$  /$$$$$$   /$$   /$$           |
|          /$$_____/ |____  $$| $$$$   /$$__  $$|_  $$_/  | $$  | $$           |
|         |  $$$$$$   /$$$$$$$| $$_/  | $$$$$$$$  | $$    | $$  | $$           |
|          \____  $$ /$$__  $$| $$    | $$_____/  | $$ /$$| $$  | $$           |
|          /$$$$$$$/|  $$$$$$$| $$    |  $$$$$$$  |  $$$$/|  $$$$$$$           |
|         |_______/  \_______/|__/     \_______/   \___/   \____  $$           |
|                                                          /$$  | $$           |
|                                                         |  $$$$$$/           |
|  by pyup.io                                              \______/            |
|                                                                              |
+==============================================================================+
| REPORT                                                                       |
| checked 18 packages, using free DB (updated once a month)                    |
+============================+===========+==========================+==========+
| package                    | installed | affected                 | ID       |
+============================+===========+==========================+==========+
| aiohttp-jinja2             | 1.1.0     | <1.1.1                   | 37095    |
| aiohttp-jinja2             | 1.1.0     | <1.1.1                   | 44431    |
| aiohttp-jinja2             | 1.1.0     | <1.1.1                   | 44432    |
| aiohttp                    | 3.5.3     | <3.7.4                   | 39659    |
| aiohttp                    | 3.5.3     | <3.8.0                   | 42692    |
| jinja2                     | 2.10      | <2.11.3                  | 39525    |
| pyyaml                     | 3.13      | <4                       | 36333    |
| pyyaml                     | 3.13      | <5.3.1                   | 38100    |
| pyyaml                     | 3.13      | <5.4                     | 39611    |
+==============================================================================+
```

DVPWA: SCA



```
aiohttp-jinja2==1.1.0
aiohttp-session==2.7.0
aiohttp==3.5.3
aiopg==0.15.0
aioredis==1.2.0
async-timeout==3.0.1       # via aiohttp, aioredis
attrs==18.2.0              # via aiohttp
chardet==3.0.4             # via aiohttp
hiredis==0.3.1             # via aioredis
idna==2.8                  # via yarl
jinja2==2.10               # via aiohttp-jinja2
markupsafe==1.1.0          # via jinja2
multidict==4.5.2           # via aiohttp, yarl
psycopg2==2.7.6.1          # via aiopg
pyyaml==3.13
trafaret-config==2.0.2
trafaret==1.2.0
yarl==1.3.0                # via aiohttp
```



```
aiohttp-jinja2==1.1.1
aiohttp-session==2.7.0
aiohttp==3.8.0
aiopg==0.15.0
aioredis==1.2.0
async-timeout==4.0.2       # via aiohttp, aioredis
attrs==18.2.0              # via aiohttp
chardet==3.0.4             # via aiohttp
hiredis==0.3.1             # via aioredis
idna==2.8                  # via yarl
jinja2==2.11.3               # via aiohttp-jinja2
markupsafe==1.1.0          # via jinja2
multidict==4.5.2           # via aiohttp, yarl
psycopg2==2.7.6.1          # via aiopg
pyyaml==5.4
trafaret-config==2.0.2
trafaret==1.2.0
yarl==1.3.0                # via aiohttp
whispers==1.5.3
bandit==1.7.4
lxml
```

On the pipeline: SAST → Bandit, by PyCQA

PyCQA / **bandit** Public

- ➔ Bandit is a tool that can be used during development or afterward. It is also used to analyze existing projects and find possible flaws.
- ➔ Finds common security issues in Python code by processing files, builds an AST from them, runs appropriate plugins against the AST nodes and produces a report from results obtained.

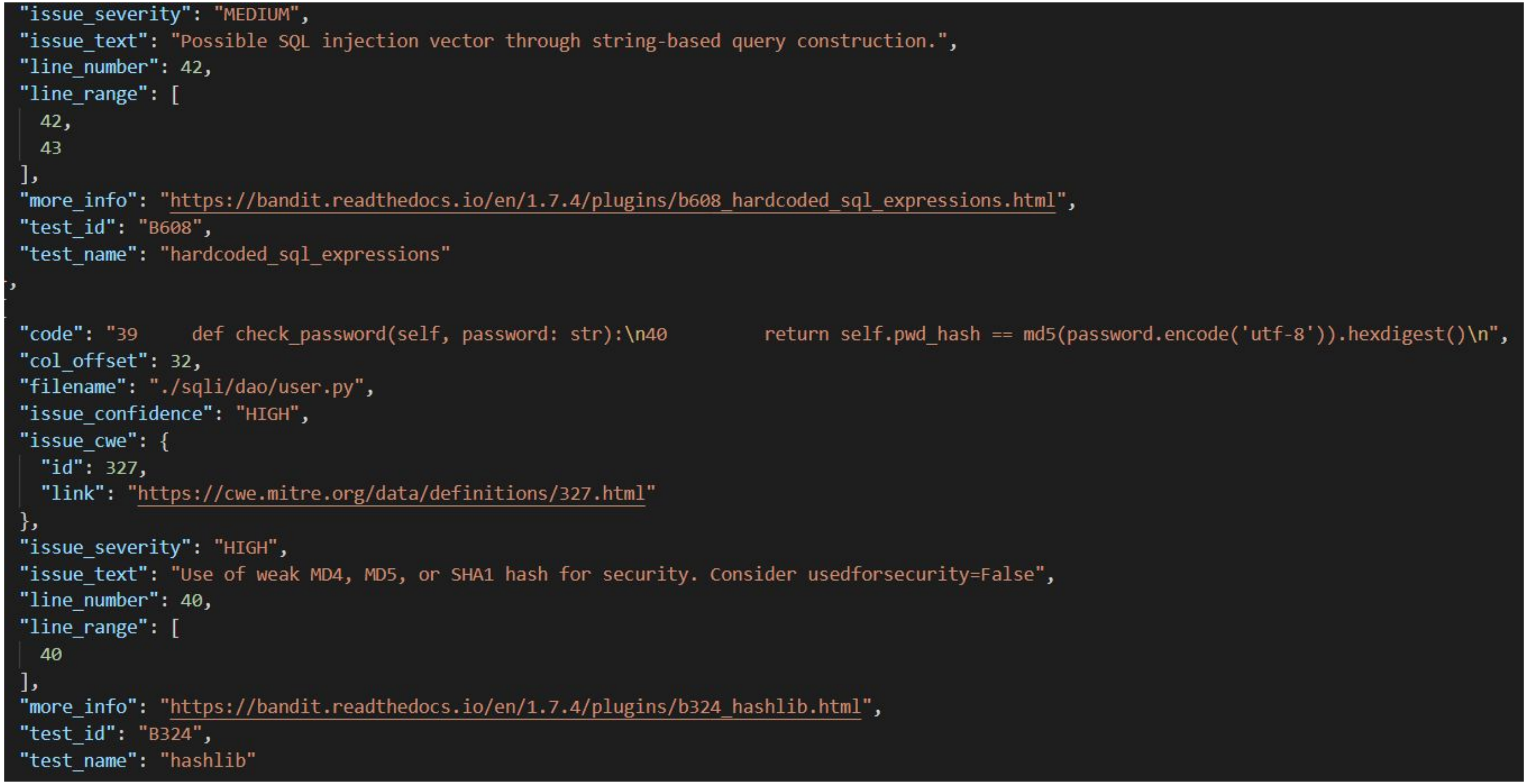
PROS

- ➔ Supported by a large company
- ➔ Ships with 68 security checks, and supports custom rules and overriding plugin configurations
- ➔ Github actions provided by community

CONS

- ➔ Multithreading not supported
- ➔ Only Python is supported

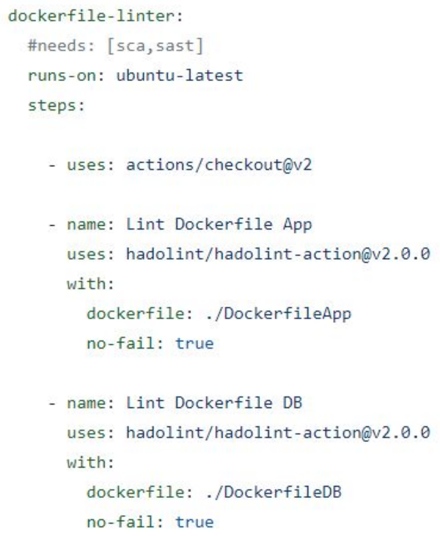
DVPWA: Bandit



```
 "issue_severity": "MEDIUM",
 "issue_text": "Possible SQL injection vector through string-based query construction.",
 "line_number": 42,
 "line_range": [
   42,
   43
 ],
 "more_info": "https://bandit.readthedocs.io/en/1.7.4/plugins/b608_hardcoded_sql_expressions.html",
 "test_id": "B608",
 "test_name": "hardcoded_sql_expressions"
},
{
 "code": "39     def check_password(self, password: str):\n40         return self.pwd_hash == md5(password.encode('utf-8')).hexdigest()\n",
 "col_offset": 32,
 "filename": "./sqli/dao/user.py",
 "issue_confidence": "HIGH",
 "issue_cwe": {
   "id": 327,
   "link": "https://cwe.mitre.org/data/definitions/327.html"
 },
 "issue_severity": "HIGH",
 "issue_text": "Use of weak MD4, MD5, or SHA1 hash for security. Consider usedforsecurity=False",
 "line_number": 40,
 "line_range": [
   40
 ],
 "more_info": "https://bandit.readthedocs.io/en/1.7.4/plugins/b324_hashlib.html",
 "test_id": "B324",
 "test_name": "hashlib"
```

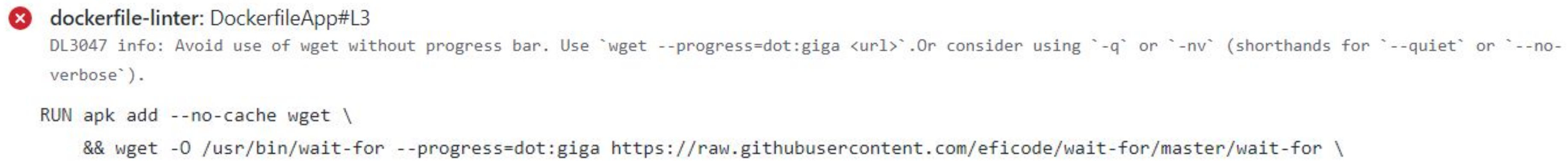
Dockerfile-linter - Introduction

Hadolint Action is a Dockerfile linter that helps building best practice Docker images



```yaml
dockerfile-linter:
  #needs: [sca,sast]
  runs-on: ubuntu-latest
  steps:

    - uses: actions/checkout@v2

    - name: Lint Dockerfile App
      uses: hadolint/hadolint-action@v2.0.0
      with:
        dockerfile: ./DockerfileApp
        no-fail: true

    - name: Lint Dockerfile DB
      uses: hadolint/hadolint-action@v2.0.0
      with:
        dockerfile: ./DockerfileDB
        no-fail: true
```

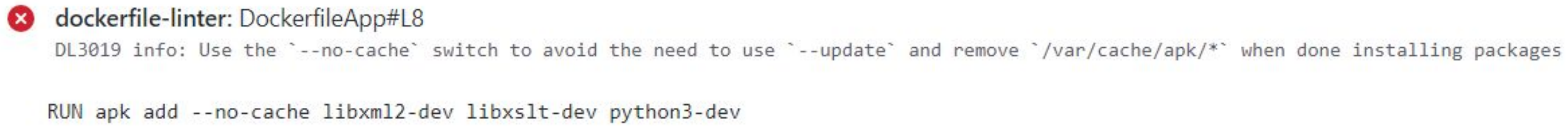
Dockerfile-linter - Improvement



dockerfile-linter: DockerfileApp#L3

DL3047 info: Avoid use of wget without progress bar. Use `wget --progress=dot:giga <url>`.Or consider using `-q` or `-nv` (shorthands for `--quiet` or `--no-verbose`).

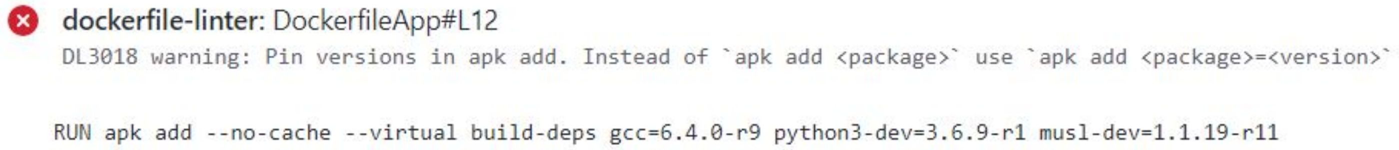
```
RUN apk add --no-cache wget \
    && wget -O /usr/bin/wait-for --progress=dot:giga https://raw.githubusercontent.com/eficode/wait-for/master/wait-for \
```



dockerfile-linter: DockerfileApp#L8

DL3019 info: Use the `--no-cache` switch to avoid the need to use `--update` and remove `/var/cache/apk/*` when done installing packages

```
RUN apk add --no-cache libxml12-dev libxslt-dev python3-dev
```

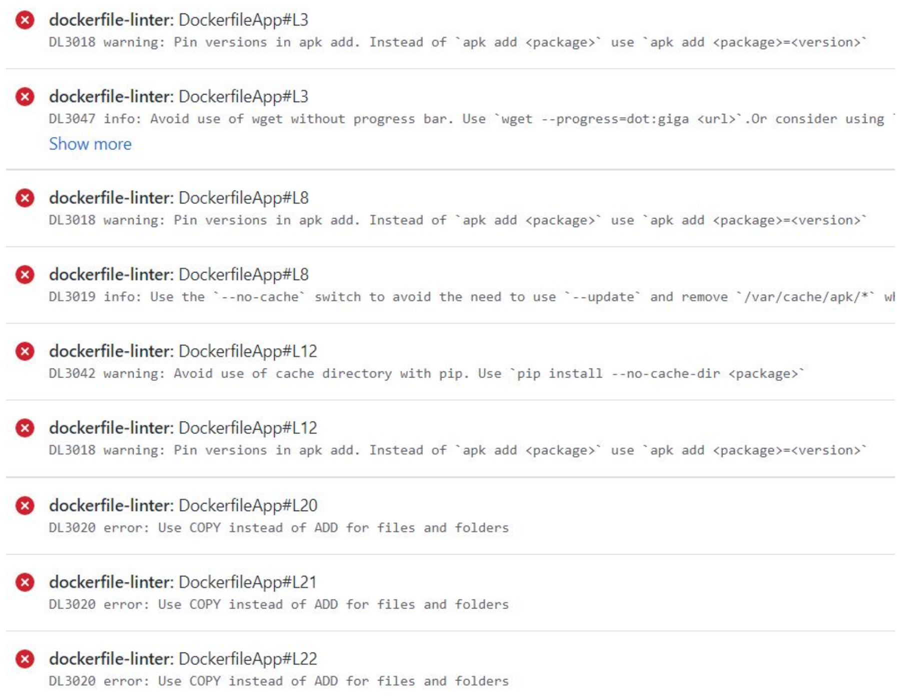


dockerfile-linter: DockerfileApp#L12

DL3018 warning: Pin versions in apk add. Instead of `apk add <package>` use `apk add <package>=<version>`

```
RUN apk add --no-cache --virtual build-deps gcc=6.4.0-r9 python3-dev=3.6.9-r1 musl-dev=1.1.19-r11
```

Dockerfile-linter - Result

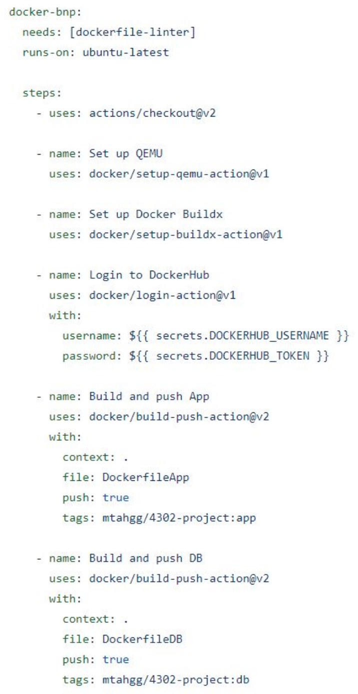


```
dockerfile-linter: DockerfileApp#L3
DL3018 warning: Pin versions in apk add. Instead of `apk add <package>` use `apk add <package>=<version>`

dockerfile-linter: DockerfileApp#L3
DL3047 info: Avoid use of wget without progress bar. Use `wget --progress=dot:giga <url>`.Or consider using `
Show more

dockerfile-linter: DockerfileApp#L8
DL3018 warning: Pin versions in apk add. Instead of `apk add <package>` use `apk add <package>=<version>`

dockerfile-linter: DockerfileApp#L8
DL3019 info: Use the `--no-cache` switch to avoid the need to use `--update` and remove `/var/cache/apk/*` wl

dockerfile-linter: DockerfileApp#L12
DL3042 warning: Avoid use of cache directory with pip. Use `pip install --no-cache-dir <package>`

dockerfile-linter: DockerfileApp#L12
DL3018 warning: Pin versions in apk add. Instead of `apk add <package>` use `apk add <package>=<version>`

dockerfile-linter: DockerfileApp#L20
DL3020 error: Use COPY instead of ADD for files and folders

dockerfile-linter: DockerfileApp#L21
DL3020 error: Use COPY instead of ADD for files and folders

dockerfile-linter: DockerfileApp#L22
DL3020 error: Use COPY instead of ADD for files and folders
```

$\Rightarrow$ None

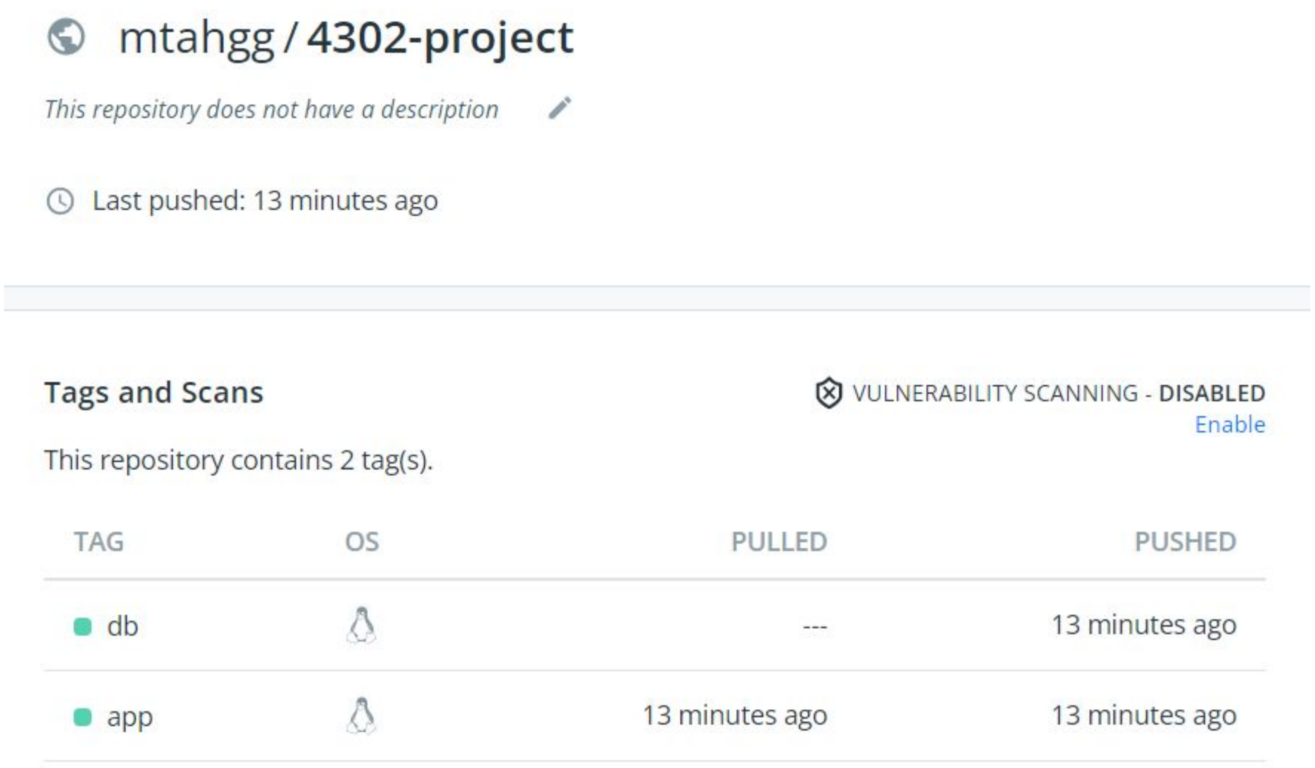
Docker-bnp - Introduction

GitHub Action to build and push Docker images



```yaml
docker-bnp:
  needs: [dockerfile-linter]
  runs-on: ubuntu-latest

  steps:
    - uses: actions/checkout@v2

    - name: Set up QEMU
      uses: docker/setup-qemu-action@v1

    - name: Set up Docker Buildx
      uses: docker/setup-buildx-action@v1

    - name: Login to DockerHub
      uses: docker/login-action@v1
      with:
        username: ${{ secrets.DOCKERHUB_USERNAME }}
        password: ${{ secrets.DOCKERHUB_TOKEN }}

    - name: Build and push App
      uses: docker/build-push-action@v2
      with:
        context: .
        file: DockerfileApp
        push: true
        tags: mtahgg/4302-project:app

    - name: Build and push DB
      uses: docker/build-push-action@v2
      with:
        context: .
        file: DockerfileDB
        push: true
        tags: mtahgg/4302-project:db
```

Docker-bnp - Result



mtahgg / **4302-project**

This repository does not have a description

Last pushed: 13 minutes ago

Tags and Scans

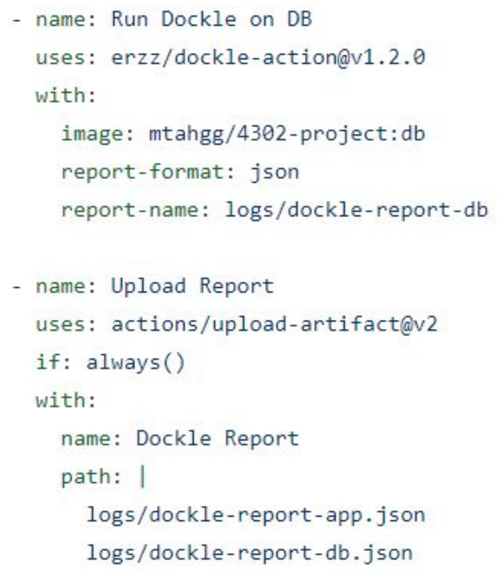
VULNERABILITY SCANNING - **DISABLED**
Enable

This repository contains 2 tag(s).

TAG	OS	PULLED	PUSHED
🟩 db	🐧	---	13 minutes ago
🟩 app	🐧	13 minutes ago	13 minutes ago

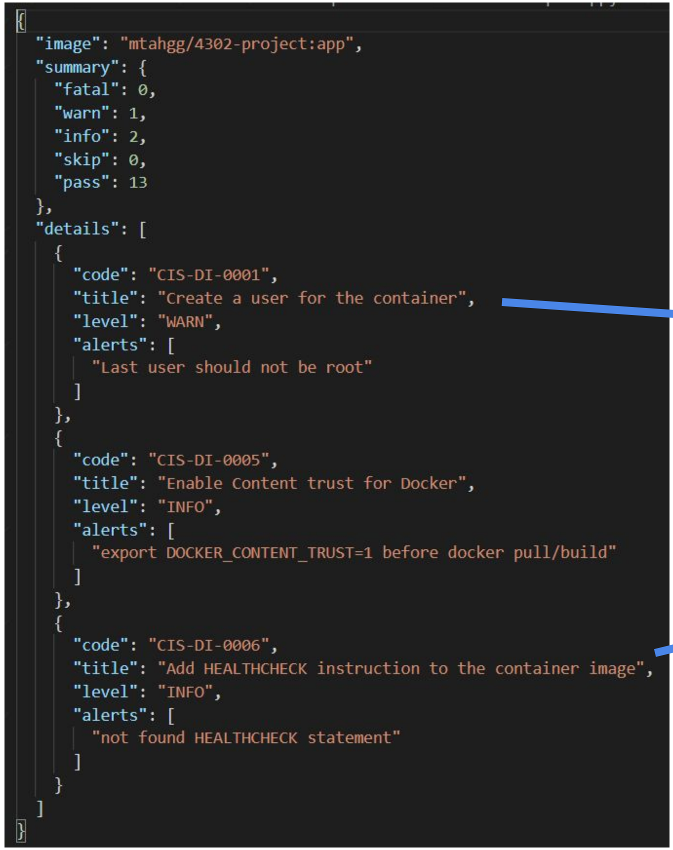
Container-scan - Introduction

Dockle action executes the excellent Dockle linter for containers that will run numerous checks on an image for Best Practices and against CIS benchmarks.

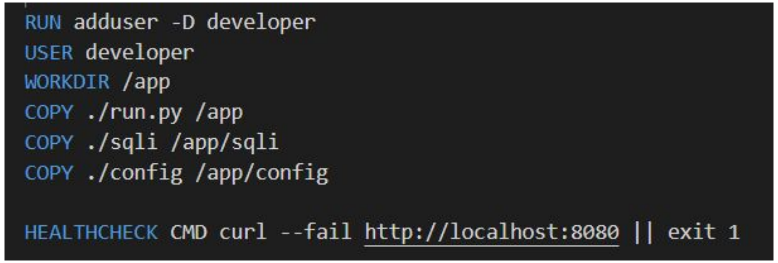


```yaml
- name: Run Dockle on DB
  uses: erzz/dockle-action@v1.2.0
  with:
    image: mtahgg/4302-project:db
    report-format: json
    report-name: logs/dockle-report-db

- name: Upload Report
  uses: actions/upload-artifact@v2
  if: always()
  with:
    name: Dockle Report
    path: |
      logs/dockle-report-app.json
      logs/dockle-report-db.json
```

Container-scan - Improvement

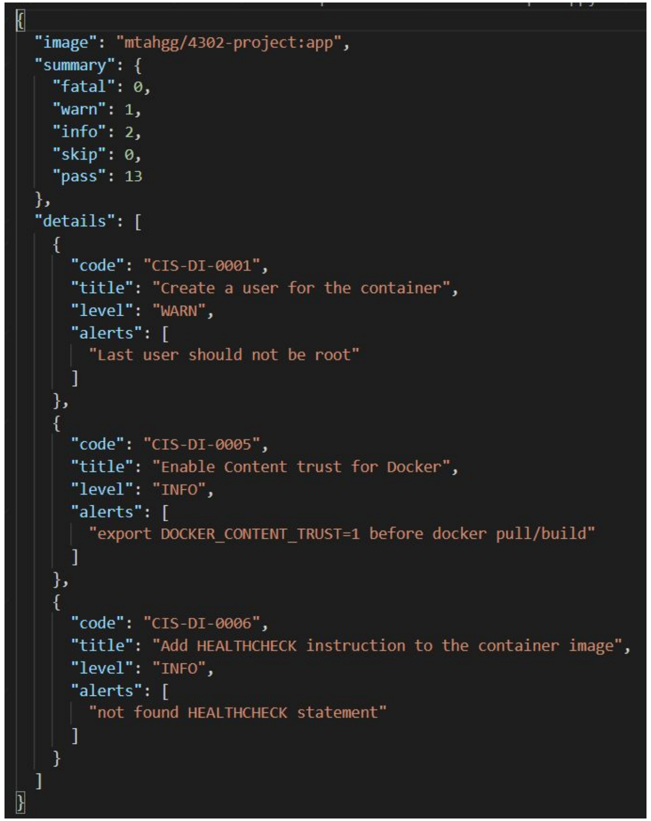


```
{
  "image": "mtahgg/4302-project:app",
  "summary": {
    "fatal": 0,
    "warn": 1,
    "info": 2,
    "skip": 0,
    "pass": 13
  },
  "details": [
    {
      "code": "CIS-DI-0001",
      "title": "Create a user for the container",
      "level": "WARN",
      "alerts": [
        "Last user should not be root"
      ]
    },
    {
      "code": "CIS-DI-0005",
      "title": "Enable Content trust for Docker",
      "level": "INFO",
      "alerts": [
        "export DOCKER_CONTENT_TRUST=1 before docker pull/build"
      ]
    },
    {
      "code": "CIS-DI-0006",
      "title": "Add HEALTHCHECK instruction to the container image",
      "level": "INFO",
      "alerts": [
        "not found HEALTHCHECK statement"
      ]
    }
  ]
}
```

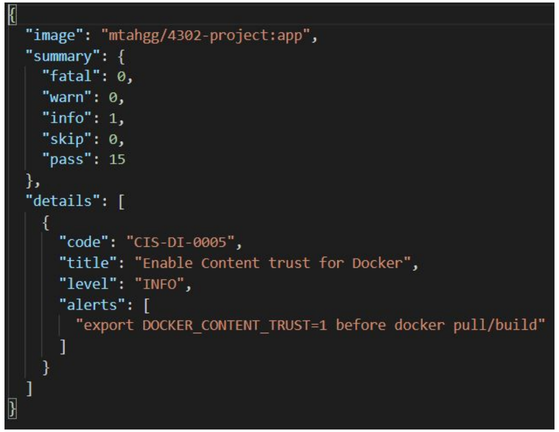


```
RUN adduser -D developer
USER developer
WORKDIR /app
COPY ./run.py /app
COPY ./sqli /app/sqli
COPY ./config /app/config

HEALTHCHECK CMD curl --fail http://localhost:8080 || exit 1
```

Container-scan - Result



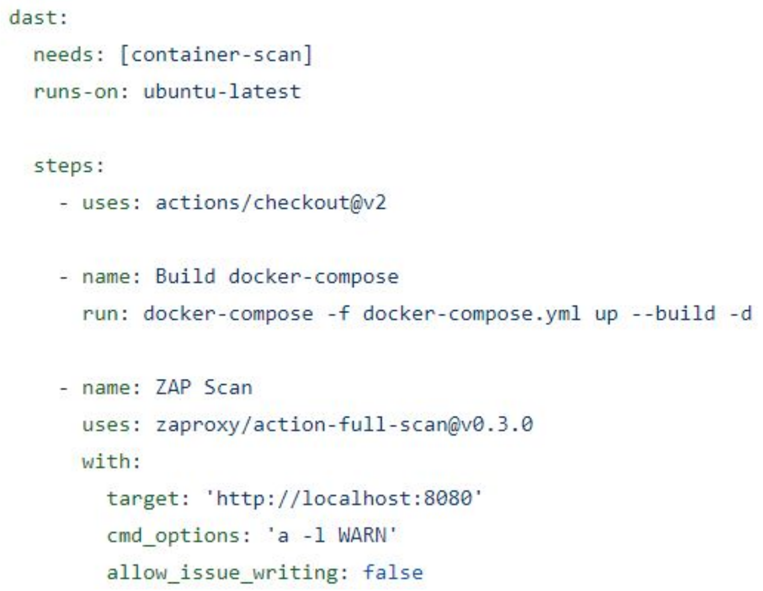
```
{
  "image": "mtahgg/4302-project:app",
  "summary": {
    "fatal": 0,
    "warn": 1,
    "info": 2,
    "skip": 0,
    "pass": 13
  },
  "details": [
    {
      "code": "CIS-DI-0001",
      "title": "Create a user for the container",
      "level": "WARN",
      "alerts": [
        "Last user should not be root"
      ]
    },
    {
      "code": "CIS-DI-0005",
      "title": "Enable Content trust for Docker",
      "level": "INFO",
      "alerts": [
        "export DOCKER_CONTENT_TRUST=1 before docker pull/build"
      ]
    },
    {
      "code": "CIS-DI-0006",
      "title": "Add HEALTHCHECK instruction to the container image",
      "level": "INFO",
      "alerts": [
        "not found HEALTHCHECK statement"
      ]
    }
  ]
}
```



```
{
  "image": "mtahgg/4302-project:app",
  "summary": {
    "fatal": 0,
    "warn": 0,
    "info": 1,
    "skip": 0,
    "pass": 15
  },
  "details": [
    {
      "code": "CIS-DI-0005",
      "title": "Enable Content trust for Docker",
      "level": "INFO",
      "alerts": [
        "export DOCKER_CONTENT_TRUST=1 before docker pull/build"
      ]
    }
  ]
}
```

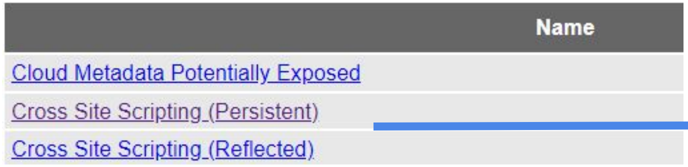
Dast - Introduction

OWASP ZAP Full Scan is a GitHub Action to perform Dynamic Application Security Testing (DAST). The application is built locally for ZAP scan.

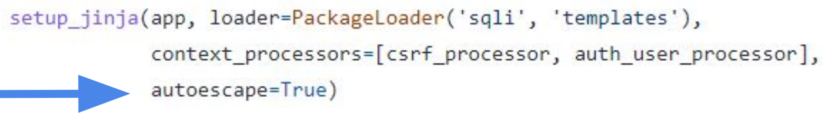


```
dast:
  needs: [container-scan]
  runs-on: ubuntu-latest

  steps:
    - uses: actions/checkout@v2

    - name: Build docker-compose
      run: docker-compose -f docker-compose.yml up --build -d

    - name: ZAP Scan
      uses: zaproxy/action-full-scan@v0.3.0
      with:
        target: 'http://localhost:8080'
        cmd_options: 'a -l WARN'
        allow_issue_writing: false
```

Dast - Improvement

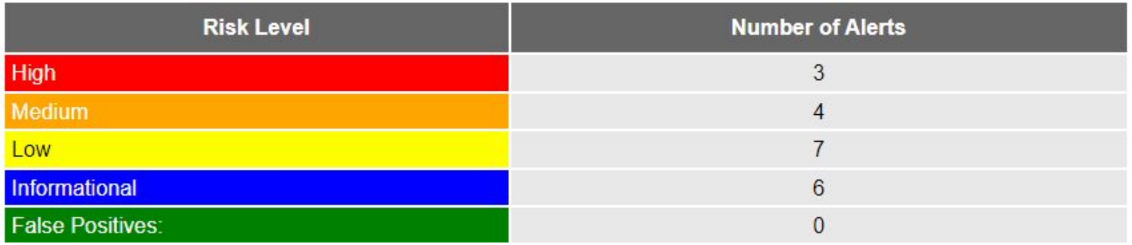


Name
Cloud Metadata Potentially Exposed
Cross Site Scripting (Persistent)
Cross Site Scripting (Reflected)

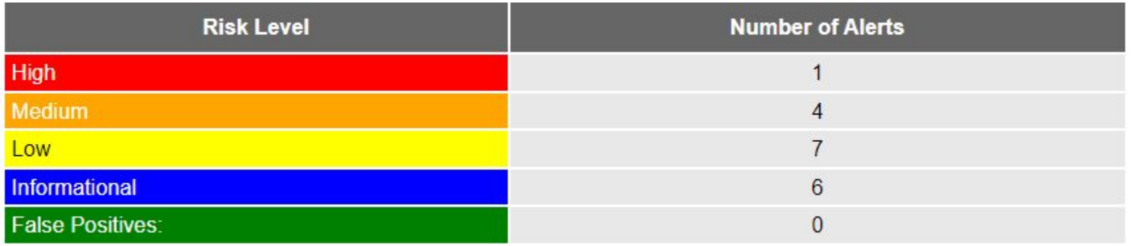


```
setup_jinja(app, loader=PackageLoader('sqli', 'templates'),
            context_processors=[csrf_processor, auth_user_processor],
            autoescape=True)
```

Dast - Result



Risk Level	Number of Alerts
High	3
Medium	4
Low	7
Informational	6
False Positives:	0



Risk Level	Number of Alerts
High	1
Medium	4
Low	7
Informational	6
False Positives:	0

What we learned

- ➔ Learning to solve issues/errors not encountered during lessons
- ➔ MANY errors every step of the way
- ➔ Each step is critical which could create vulnerability
- ➔ Not every non-critical vulnerability have to be fixed

END.