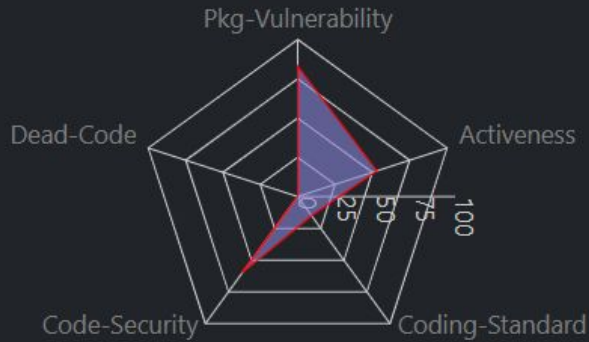




Debt Detective

TEAM-5
Release 2

Recap



- Package Vulnerability and Security
- Community Activeness and Maintainance
- Coding Quality and Standard
- Code Vulnerability and Security
- Depreciated, Dead and Outdated code

Package Vulnerability, Activeness

Elaboration on Vulnerabilities

numpy

CVE-2021-41495

Numpy 1.22.2 includes a fix for CVE-2021-41495: Null Pointer Dereference vulnerability exists in numpy.sort in NumPy in the PyArray_DescrNew function due to missing return-value validation, which allows attackers to conduct DoS attacks by repetitively creating sort arrays. NOTE: While correct that validation is missing, an error can only occur due to an exhaustion of memory. If the user can exhaust memory, they are already privileged. Further, it should be practically impossible to construct an attack which can target the memory exhaustion to occur at exactly this place.

<https://github.com/numpy/numpy/issues/19038>

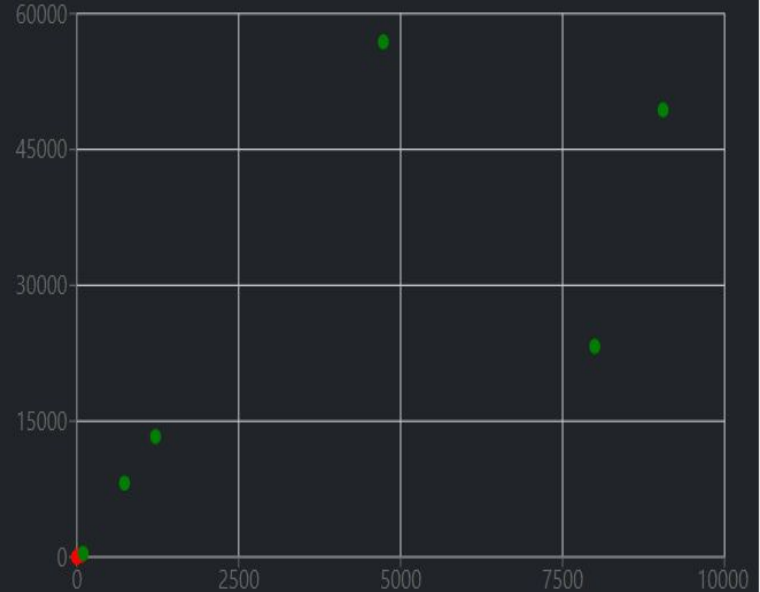
numpy

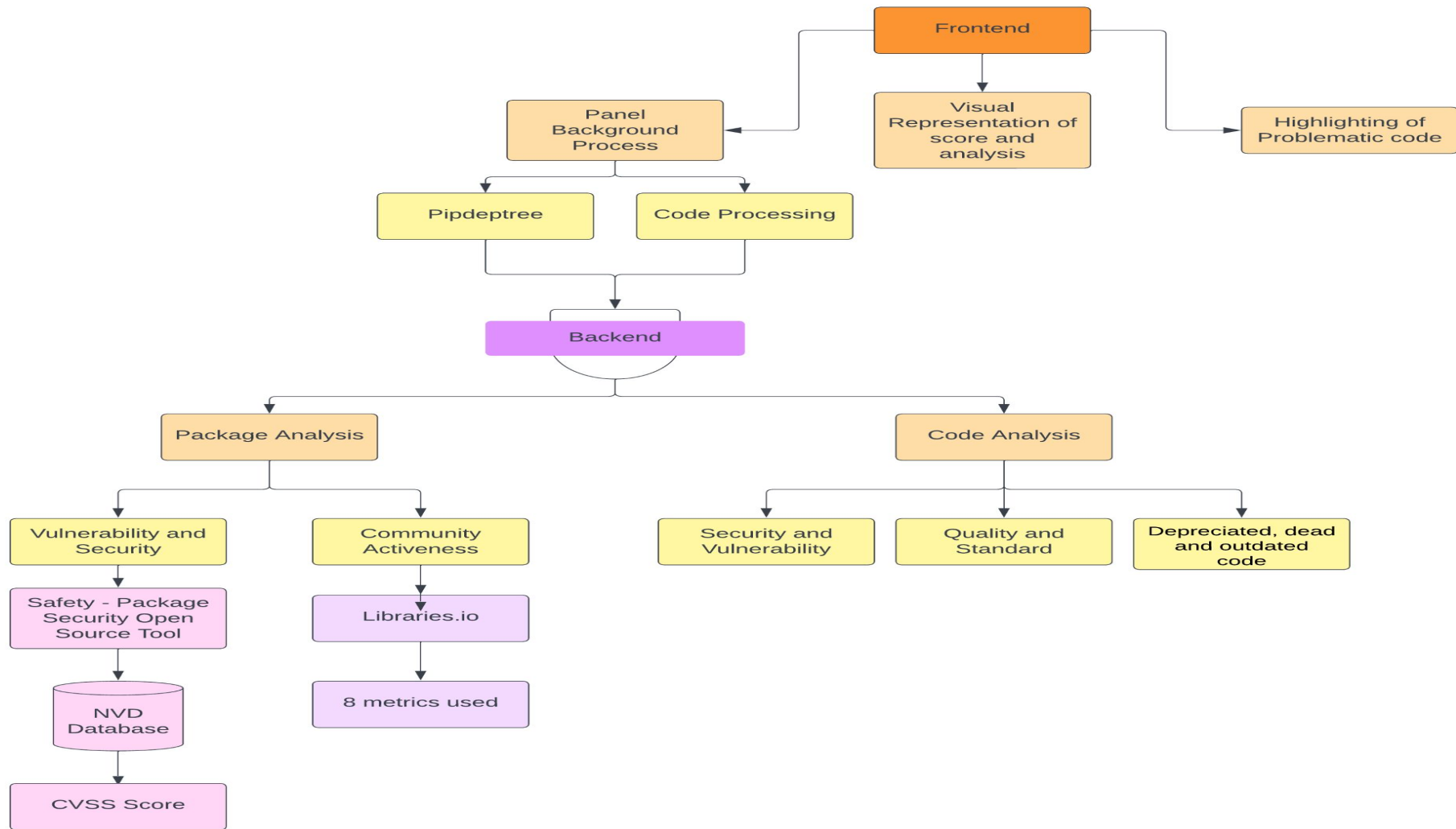
CVE-2021-34141

Numpy 1.22.0 includes a fix for CVE-2021-34141: An incomplete string comparison in the numpy.core component in NumPy before 1.22.0 allows attackers to trigger slightly incorrect copying by constructing specific string objects. NOTE: the vendor states that this reported code behavior is "completely harmless."

<https://github.com/numpy/numpy/issues/18993>

Community Activeness and Maintainance





RELEASE 1 COMMENTS

“Naming convention could have been improved, example- currScore is not a valid name”

```
def calculateLibrariesIOScore(pythonDict,name,versionsArray):  
    currScore = 0  
    (variable) currScore: Any  
    Variable name "currScore" doesn't conform to  
    snake_case naming style  
    /sourcerank?api_key=7b7f69d0b46f645c7cfc7c6231db6ae6".format(source="Pypi",name=name)  
    currScore += dict["readme_present"];  
    #multiple versions  
    if len(versionsArray) >1:  
        currScore+=1  
    #recent release
```

“Technical Debt analyzer in GH”

“Github repo link?”

- Rather than the conventional method of analysing the code quality and standard along with the issues associated with security at the end of the lifecycle it is more useful for the developer to tackle the potential issues on the go giving him more control over how the product would function in the long term

“Provide individual metrics score and provide visualization”

Sub-parameters and Scores for each package

FASTAPI
PYDANTIC
TYPINGEXTENSIONS
STARLETTE
ANYIO
IDNA
SNIFFIO
NUMPY
RATELIMIT
REQUESTS
TYPING

FASTAPI

- Age of Repo: 1
(The repository is more than 6 months old)
- Contributors: 2/10
- Dependents: 22
- Multiple Versions : ❌
- README: ✔️
- Any Recent Releases : ✔️
- Stars and Forks: 29

PYDANTIC

TYPINGEXTENSIONS

STARLETTE

Package Vulnerability and Security

TYPING	CVE-2021-41495
---------------	-----------------------

TYPING	CVE-2021-34141
---------------	-----------------------

TYPING	CVE-2021-41496
---------------	-----------------------

TYPING

CVE-2021-41495

Numpy 1.22.2 includes a fix for CVE-2021-41495: Null Pointer Dereference vulnerability exists in `numpy.sort` in NumPy in the `PyArray_DescrNew` function due to missing return-value validation, which allows attackers to conduct DoS attacks by repetitively creating sort arrays. NOTE: While correct that validation is missing, an error can only occur due to an exhaustion of memory. If the user can exhaust memory, they are already privileged. Further, it should be practically impossible to construct an attack which can target the memory exhaustion to occur at exactly this place.
<https://github.com/numpy/numpy/issues/19038>

TYPING

CVE-2021-34141

SCORES

Package Vulnerability

- [pipdeptree](#)
- Security

Exploitability Score	<ul style="list-style-type: none">• Availability Impact
	<ul style="list-style-type: none">• Integrity Impact
	<ul style="list-style-type: none">• Confidentiality Impact
Impact Score	The potential impact of a security vulnerability on the target system

Community Activeness

- Libraries.io

Age of repository	0/1 (at least 6 months)
Contributors	$\log_2(\text{number of contributors})$
Dependents	$2 * \log_2(\text{dependent_packages})$
Multiple versions	0/1
Readme	0/1
Recent Release	0/1
Stars and Forks	$\log_2(\text{stars}) + \log_2(\text{forks})$

Code Security

- Bandit

ISSUE SEVERITY/CONFIDENCE	SCORE
LOW	1
MEDIUM	3
HIGH	5

<https://docs.google.com/document/d/1PKPCAuprJ8-26dom4fq2NGJznpjUHnjlFXfM4F888KhU/edit>

Results

Coding Standard

Consider using enumerate instead of iterating with range and len

```
for i in range(len(packages_arr)):  
    getresults(packages_arr[i])
```

Missing function or method docstring

```
async def osv():
```

Try, Except, Continue detected. debt-detective(Confidence: HIGH)

No exception type(s) specified

[View Problem \(Alt+F8\)](#) No quick fixes available

```
except:
```

```
    continue;
```

[n]  See Real World Examples From GitHub

pa

Formatting a regular string which could be a f-string

except

re

[Follow link \(ctrl + click\)](#)

```
url = "https://libraries.io/api/{package}/{name}?api=7b7f69d0b46f645c7cfc7c6231db6ae6?".format(package="Pypi",name=name)
```


(class) LoginManager

This object is used to hold the settings used for logging in. Instances of `LoginManager` are **not** bound to specific apps, so you can create one in the main body of your code and then bind it to your app in a factory function.

Too many instance attributes (16/7)

```
class LoginManager:
```

Too many arguments (7/5)

```
def create_form(
    self,
    label_columns=None,
    inc_columns=None,
    description_columns=None,
    validators_columns=None,
    extra_fields=None,
    filter_rel_fields=None,
):
```

(variable) packages_arr: list[str]

@app
def
Redefining name 'packages_arr' from outer scope (line 23)

```
packages_arr = ["numpy", "pandas", "tensorflow", "requests"]
```

```
if widget:
    return field(
        self.label,
        description=self.description,
        validators=self.validators,
        widget=widget())
```

Unnecessary "else" after "return", remove the "else" and de-indent the code inside it

```
else:
    return field(
        self.label,
        description=self.description,
        validators=self.validators,
        default=self.default,
    )
```

```
log.error("Column %s Type not supported" % self.colname)
```

(variable) frame_size_x: Literal[720]

Constant name "frame_size_x" doesn't conform to UPPER_CASE naming style

frame_size_x = 720

Security

Use of unsafe yaml load. Allows instantiation of arbitrary objects. Consider yaml.safe_load(). debt-detective(Confidence: HIGH)

b = (module) yaml

n_va

k =  [See Real World Examples From GitHub](#)

ystr [View Problem \(Alt+F8\)](#) No quick fixes available

y = yaml.load(ystr)

main.py 1, M

requirements.txt M

utils.py

stackoverflow.py

_utils.py

main.py > ...

```
326 input
327 with open("bandit_output.json", "r") as f:
328     input_data = f.read()
329     python_dict_bandit= json.loads(input_data)
330     SECURITY_ARRAY = []
331     total_security_score = 0
332     code_security_score = 0
```

Using open without explicitly specifying an encoding

PROBLEMS

13

OUTPUT

DEBUG CONSOLE

TERMINAL

main.py 13

- ⚠ Audit url open for permitted schemes. Allowing use of file:/ or custom schemes is often unexpected. debt-detective(Confidence: HIGH) [Ln 215, Col 20]
- ① Consider possible security implications associated with the subprocess module. debt-detective(Confidence: HIGH) [Ln 11, Col 1]
- ① Starting a process with a partial executable path debt-detective(Confidence: HIGH) [Ln 99, Col 13]
- ① subprocess call with shell=True seems safe, but may be changed in the future, consider rewriting without shell debt-detective(Confidence: HIGH) [Ln 99, Col 13]
- ① Try, Except, Continue detected. debt-detective(Confidence: HIGH) [Ln 130, Col 13]
- ① Starting a process with a partial executable path debt-detective(Confidence: HIGH) [Ln 264, Col 9]
- ① subprocess call with shell=True seems safe, but may be changed in the future, consider rewriting without shell debt-detective(Confidence: HIGH) [Ln 264, Col 9]
- ① Starting a process with a partial executable path debt-detective(Confidence: HIGH) [Ln 278, Col 9]
- ① subprocess call with shell=True seems safe, but may be changed in the future, consider rewriting without shell debt-detective(Confidence: HIGH) [Ln 278, Col 9]
- ① Starting a process with a partial executable path debt-detective(Confidence: HIGH) [Ln 295, Col 9]
- ① subprocess call with shell=True seems safe, but may be changed in the future, consider rewriting without shell debt-detective(Confidence: HIGH) [Ln 295, Col 9]
- ① Starting a process with a partial executable path debt-detective(Confidence: HIGH) [Ln 323, Col 9]
- ① subprocess call with shell=True seems safe, but may be changed in the future, consider rewriting without shell debt-detective(Confidence: HIGH) [Ln 323, Col 9]

Standard pseudo-random generators are not suitable for security/cryptographic purposes. debt-detective(Confidence: HIGH)

(variable) randrange: (start: int, stop: int | None = None, step: int = 1) -> int

Choose a random item from range(start, stop[, step]).

This fixes the problem with randint() which includes the endpoint; in Python this is usually not what you want.



[See Real World Examples From GitHub](#)

[View Problem \(Alt+F8\)](#) No quick fixes available

```
random.randrange(1, (frame_size_x//10)) * 10, random.randrange(1, (frame_size_y//10)) * 10]
```



```
import logging
import traceback
from fastapi import FastAPI, Request
from typing import Union
from fastapi import Request
```

Consider possible security implications associated with the subprocess module. debt-detective(Confidence: HIGH)

standard import "import subprocess" should be placed
before "from fastapi import FastAPI, Request"

[View Problem \(Alt+F8\)](#) No quick fixes available

```
import subprocess
import math
from collections import OrderedDict
import numpy as np
from pylint.lint import Run
import os
import re
```

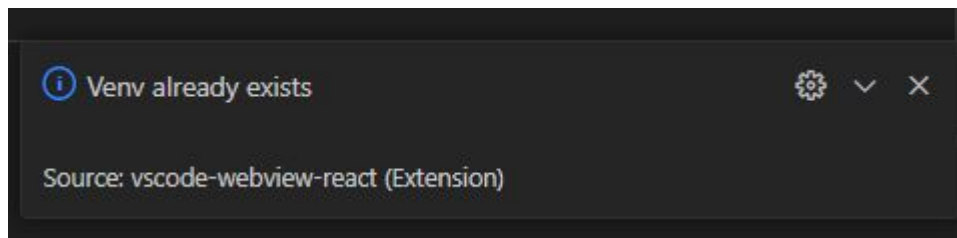
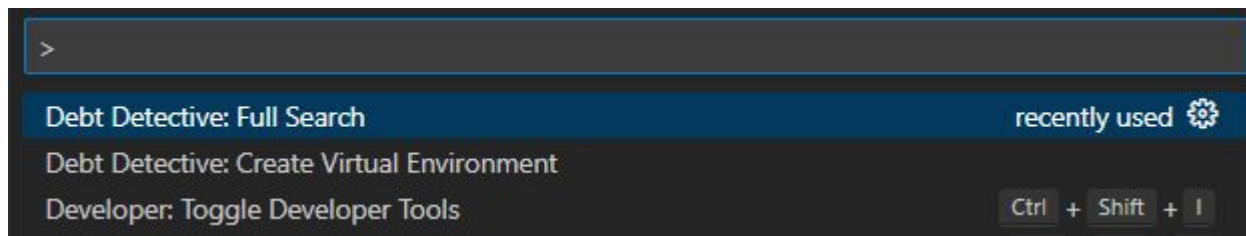

Extra Features

Dead Code and unused variables

```
Unused numpy imported as np  
unused import np (90% confidence) Dict  
import numpy as np
```

```
unused variable python_dict_linter (60% confidence)  
python_dict_linter= json.loads(input_data)  
try:
```

Venv Helper



requirements.txt

main.py

requirements.txt

1 tensorflow==2.10.0

2 numpy==1.24.2

3 pandas==1.5.3

4 requests==2.28.2

Create Environment...

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

Collecting markdown>=2.6.8
Downloading Markdown-3.4.3-py3-none-any.whl (93 kB)
93.9/93.9 KB 5.6 MB/s eta 0:00:00

Collecting werkzeug==1.0.1
Using cached Werkzeug-2.2.3-py3-none-any.whl (233 kB)

Collecting tensorboard-plugin-wit==1.6.0
Using cached tensorboard_plugin_wit-1.8.1-py3-none-any.whl (781 kB)

Collecting tensorboard-data-server<0.7.0,>=0.6.0
Using cached tensorboard_data_server-0.6.1-py3-none-any.whl (2.4 kB)

Collecting google-auth-oauthlib<0.5,>=0.4.1
Using cached google_auth_oauthlib-0.4.6-py2.py3-none-any.whl (18 kB)

Collecting pyasn1-modules>=0.2.1
Downloading pyasn1_modules-0.3.0-py2.py3-none-any.whl (181 kB)
181.3/181.3 KB 5.5 MB/s eta 0:00:00

Collecting rsa<5,>=3.1.4
Using cached rsa-4.9-py3-none-any.whl (34 kB)

Collecting cachetools<6.0,>=2.0.0
Using cached cachetools-5.3.0-py3-none-any.whl (9.3 kB)

Collecting requests-oauthlib>=0.7.0
Using cached requests_oauthlib-1.3.1-py2.py3-none-any.whl (23 kB)

Collecting importlib-metadata>=4.4
Downloading importlib_metadata-6.6.0-py3-none-any.whl (22 kB)

Collecting MarkupSafe>=2.1.1
Using cached MarkupSafe-2.1.2-cp39-cp39-win_amd64.whl (16 kB)

Collecting zipp>=0.5
Using cached zipp-3.15.0-py3-none-any.whl (6.8 kB)

Collecting pyasn1<0.6.0,>=0.4.6
Downloading pyasn1-0.5.0-py2.py3-none-any.whl (83 kB)
83.9/83.9 KB 1.6 MB/s eta 0:00:00

Collecting oauthlib>=3.0.0
Using cached oauthlib-3.2.2-py3-none-any.whl (151 kB)

Installing collected packages: tensorboard-plugin-wit, pytz, libclang, keras, flatbuffers, zipp, wrapt, wheel, urllib3, typi
s-filestorage, tensorflow-estimator, tensorboard-data-server, six, pyasn1, protobuf, packaging, oauthlib, numpy, MarkupSafe, idna, grpcio, gast, charset-normalizer, ce
rtifi, cachetools, absl-py, werkzeug, rsa, requests, python-dateutil, pyasn1-modules, opt-einsum, keras-preprocessing, import
lib-metadata, google-auth, google-auth-oauthlib, pandas, markdown, google-auth-oauthlib, tensorboard, tensorflow

pwsh

Venv Task

We noticed a new environment has been created. Do you want to select it for the workspace folder?

Source: Python (Extension)

Yes

No

Do not show again

Installing Venv

Version conflicts Detection

```
fastapi==0.92.0
```

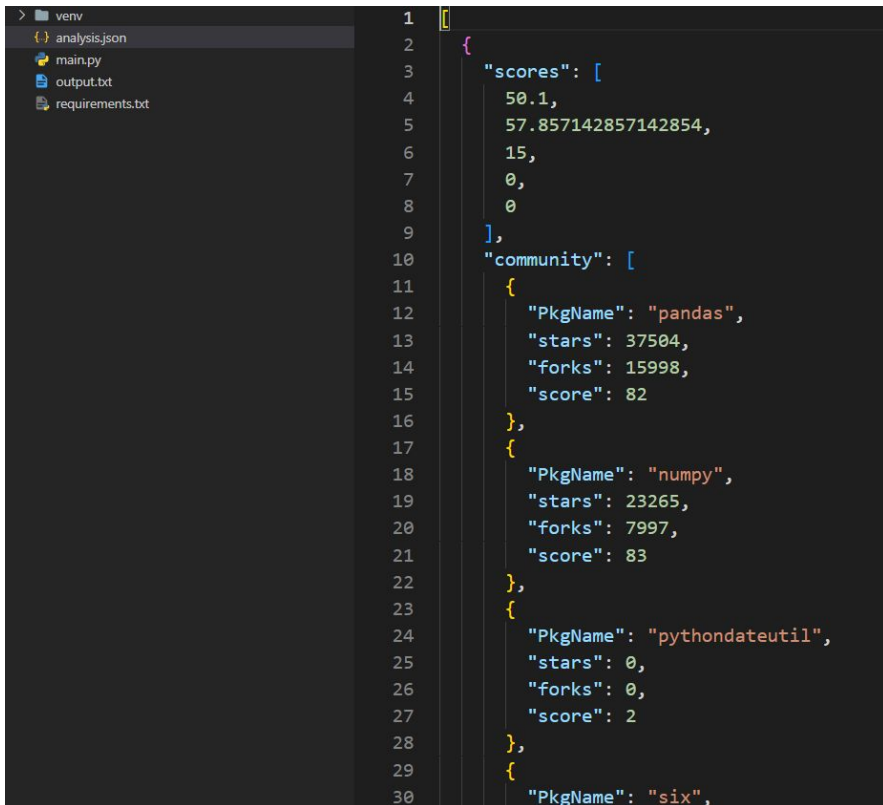
Package anyio is outdated. Required version: >= 3.4.0, version in txt file: 3.1.2 debt-detective(debt-detective)

[View Problem \(Alt+F8\)](#) No quick fixes available

```
anyio==3.1.2
```

```
idna==3.4
```

Documentation



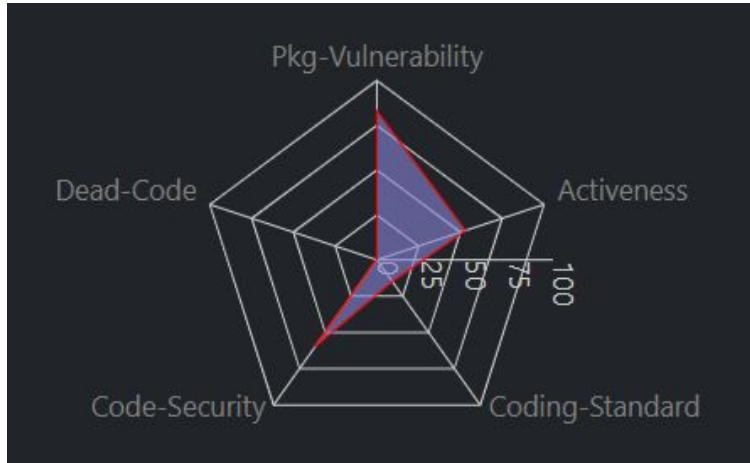
```
1  [
2  {
3    "scores": [
4      50.1,
5      57.857142857142854,
6      15,
7      0,
8      0
9    ],
10   "community": [
11     {
12       "PkgName": "pandas",
13       "stars": 37504,
14       "forks": 15998,
15       "score": 82
16     },
17     {
18       "PkgName": "numpy",
19       "stars": 23265,
20       "forks": 7997,
21       "score": 83
22     },
23     {
24       "PkgName": "pythondateutil",
25       "stars": 0,
26       "forks": 0,
27       "score": 2
28     },
29     {
30       "PkgName": "six",
```

After analysing the code the results are displayed in UI of vscode apart from that they are completely captured in a file named analysis.json

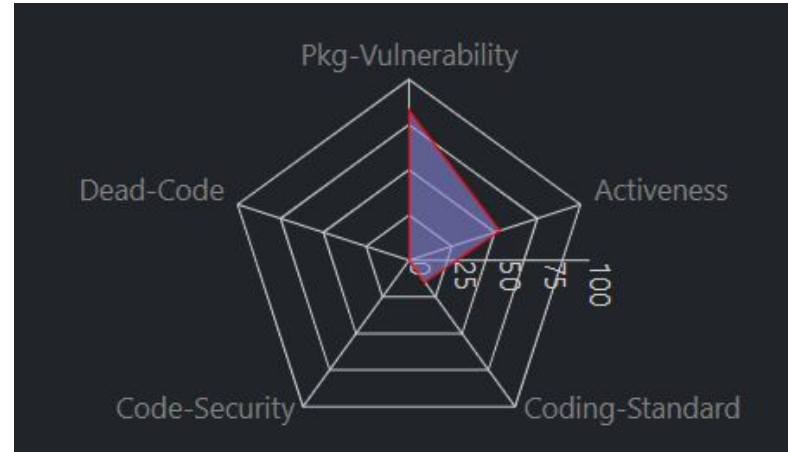
This helps developers to look back into it later without running the tool again and again

Also they can build their own things on top of this json file!

Code Entropy Graph



Before



After