

Install Bandit – static analyzer tool for python

Command: `pip install bandit`

```
C:\WINDOWS\system32\cmd. x + v

C:\Users\aparn>pip install bandit
Collecting bandit
  Obtaining dependency information for bandit from https://files.pythonhosted.org/packages/f5/b2/327fb8e5df44eec822bb19add844d03bd2f550a1c4f21913d575cdff83cc/bandit-1.7.7-py3-none-any.whl.metadata
  Downloading bandit-1.7.7-py3-none-any.whl.metadata (5.9 kB)
Requirement already satisfied: PyYAML>=5.3.1 in c:\users\aparn\anaconda3\lib\site-packages (from bandit) (6.0)
Collecting stevedore>=1.20.0 (from bandit)
  Obtaining dependency information for stevedore>=1.20.0 from https://files.pythonhosted.org/packages/4b/68/e739fd061b0aba464bef8e8be48428b2aabbfb3f2f8f2f8ca257363ee6b2/stevedore-5.1.0-py3-none-any.whl.metadata
  Downloading stevedore-5.1.0-py3-none-any.whl.metadata (2.2 kB)
Collecting rich (from bandit)
  Obtaining dependency information for rich from https://files.pythonhosted.org/packages/be/be/1520178fa01eabe014b16e72a952b9f900631142ccd03dc36cf93e30c1ce/rich-13.7.0-py3-none-any.whl.metadata
  Downloading rich-13.7.0-py3-none-any.whl.metadata (18 kB)
Requirement already satisfied: colorama>=0.3.9 in c:\users\aparn\anaconda3\lib\site-packages (from rich->bandit) (0.4.6)
Collecting pbr!=2.1.0,>=2.0.0 (from stevedore>=1.20.0->bandit)
  Obtaining dependency information for pbr!=2.1.0,>=2.0.0 from https://files.pythonhosted.org/packages/64/dd/171c9fb653591cf265bcc89c436eec75c9bde3dec921cc236fa71e5698df/pbr-6.0.0-py2.py3-none-any.whl.metadata
  Downloading pbr-6.0.0-py2.py3-none-any.whl.metadata (1.3 kB)
Requirement already satisfied: markdown-it-py>=2.2.0 in c:\users\aparn\anaconda3\lib\site-packages (from rich->bandit) (2.2.0)
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in c:\users\aparn\anaconda3\lib\site-packages (from rich->bandit) (2.15.1)
Requirement already satisfied: mdurl<=0.1 in c:\users\aparn\anaconda3\lib\site-packages (from markdown-it-py>=2.2.0->rich->bandit) (0.1.0)

Downloading bandit-1.7.7-py3-none-any.whl (124 kB)
124.2/124.2 kB 7.1 MB/s eta 0:00:00
Downloading stevedore-5.1.0-py3-none-any.whl (49 kB)
49.6/49.6 kB 2.5 MB/s eta 0:00:00
Downloading rich-13.7.0-py3-none-any.whl (240 kB)
240.6/240.6 kB 7.4 MB/s eta 0:00:00
Downloading pbr-6.0.0-py2.py3-none-any.whl (107 kB)
107.5/107.5 kB 6.5 MB/s eta 0:00:00
Installing collected packages: pbr, stevedore, rich, bandit
Successfully installed bandit-1.7.7 pbr-6.0.0 rich-13.7.0 stevedore-5.1.0

C:\Users\aparn>
```

Syntax to run bandit on a python code

`bandit -r project_name -f html -o outputfile`

-r → perform security analysis recursively

Project\_name → name of the file

-f → type of output file, in this case it is html

-o → output file

Outputfile → name of output file

Lets perform for the folder name as ‘try’

```
PS C:\Users\aparn\OneDrive\Desktop> bandit -r try -f html -o outputfile
[main] INFO     profile include tests: None
[main] INFO     profile exclude tests: None
[main] INFO     cli include tests: None
[main] INFO     cli exclude tests: None
[main] INFO     running on Python 3.11.5
[html] INFO     HTML output written to file: outputfile
PS C:\Users\aparn\OneDrive\Desktop>
```



Bandit allows you to output results in JSON format, providing more structured information. You can use the `-f` or `--format` option to specify the output format. For example:

```
bandit -r try -f json -o bandit_results.json
```

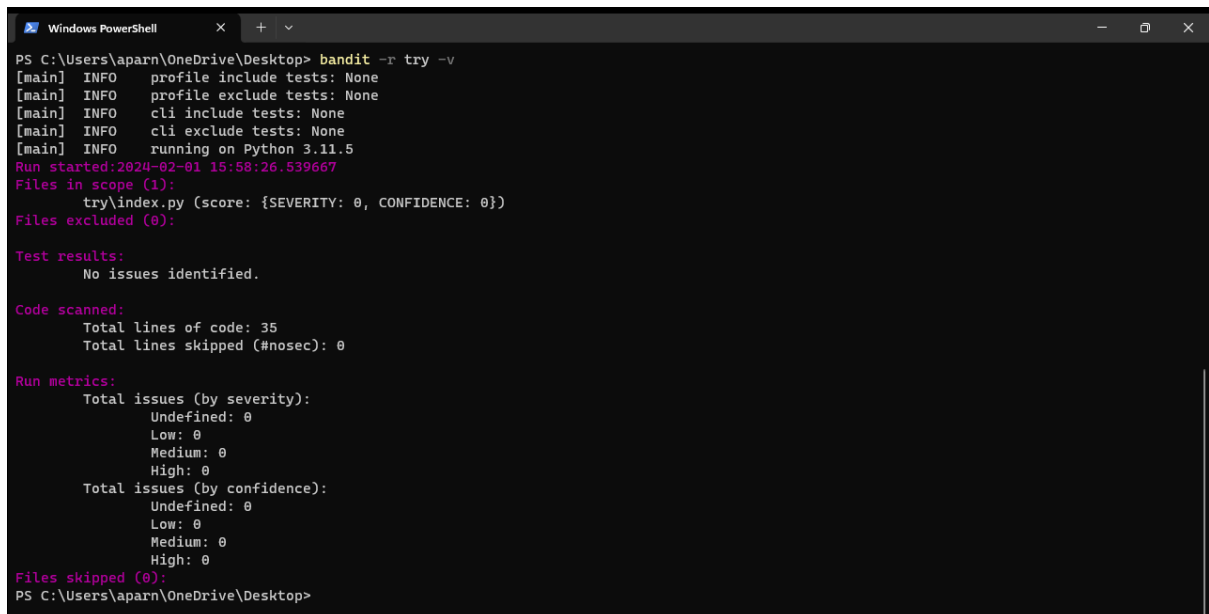
```
PS C:\Users\aparn\OneDrive\Desktop> bandit -r try -f json -o bandit_results.json
[main] INFO     profile include tests: None
[main] INFO     profile exclude tests: None
[main] INFO     cli include tests: None
[main] INFO     cli exclude tests: None
[json] INFO     JSON output written to file: bandit_results.json
PS C:\Users\aparn\OneDrive\Desktop>
```

Here is the result:-

```
{
  "errors": [],
  "generated_at": "2024-02-01T15:54:44Z",
  "metrics": {
    "_totals": {
      "CONFIDENCE.HIGH": 0,
      "CONFIDENCE.LOW": 0,
      "CONFIDENCE.MEDIUM": 0,
      "CONFIDENCE.UNDEFINED": 0,
      "SEVERITY.HIGH": 0,
      "SEVERITY.LOW": 0,
      "SEVERITY.MEDIUM": 0,
      "SEVERITY.UNDEFINED": 0,
      "loc": 35,
      "nosec": 0,
      "skipped_tests": 0
    },
    "try\\index.py": {
      "CONFIDENCE.HIGH": 0,
      "CONFIDENCE.LOW": 0,
      "CONFIDENCE.MEDIUM": 0,
      "CONFIDENCE.UNDEFINED": 0,
      "SEVERITY.HIGH": 0,
      "SEVERITY.LOW": 0,
      "SEVERITY.MEDIUM": 0,
      "SEVERITY.UNDEFINED": 0,
      "loc": 35,
      "nosec": 0,
      "skipped_tests": 0
    }
  },
  "results": []
}
```

Verbose Output: The `-v` or `--verbose` option can be used to increase verbosity, providing more information about the analysis. For example:

`bandit -r try -v`



```
PS C:\Users\aparn\OneDrive\Desktop> bandit -r try -v
[main] INFO     profile include tests: None
[main] INFO     profile exclude tests: None
[main] INFO     cli include tests: None
[main] INFO     cli exclude tests: None
[main] INFO     running on Python 3.11.5
Run started:2024-02-01 15:58:26.539667
Files in scope (1):
  try\index.py (score: {SEVERITY: 0, CONFIDENCE: 0})
Files excluded (0):

Test results:
  No issues identified.

Code scanned:
  Total lines of code: 35
  Total lines skipped (#nosec): 0

Run metrics:
  Total issues (by severity):
    Undefined: 0
    Low: 0
    Medium: 0
    High: 0
  Total issues (by confidence):
    Undefined: 0
    Low: 0
    Medium: 0
    High: 0
Files skipped (0):
PS C:\Users\aparn\OneDrive\Desktop>
```