Install Both

```
pip install bandit safety
```

Created Insecure File

```
#filename is insecure_code.py

#hardcoded token/password in code
token = ";alkjdsalfjieajlh"

#manually working with files in temp directory is not safe

temp_dir = "/tmp"
```

Executed Bandit

```
#make sure you saved insecure_code.py before executing this command

bandit insecure_code.py
```

```
Run started:2024-10-27 18:20:55.872915

Test results:
>> Issue: [B105:hardcoded_password_string] Possible hardcoded password:
 ';alkjdsalfjieajlh'
   Severity: Low   Confidence: Medium
   CWE: CWE-259 (https://cwe.mitre.org/data/definitions/259.html)
   More Info: https://bandit.readthedocs.io/en/1.7.10/plugins/b105_hard
coded_password_string.html
   Location: ./insecure_code.py:2:8
1       #hardcoded token/password in code
2       token = ";alkjdsalfjieajlh"
3

--------------------------------------------------
>> Issue: [B108:hardcoded_tmp_directory] Probable insecure usage of tem
p file/directory.
   Severity: Medium   Confidence: Medium
   CWE: CWE-377 (https://cwe.mitre.org/data/definitions/377.html)
   More Info: https://bandit.readthedocs.io/en/1.7.10/plugins/b108_hard
coded_tmp_directory.html
   Location: ./insecure_code.py:5:11
4       #manually working with files in temp directory is not safe
5       temp_dir = "/tmp"
```

Executing Bandit with Options

```
#This doesn't change any metrics or anything, it only changes what it
reports to you.

#scanning for low severity
bandit insecure_code.py -l

#scanning for medium severity
bandit insecure_code.py -ll

#scanning for high severity
bandit insecure_code.py -lll

#scanning an entire directory
bandit -r .
```

NOSEC

```
#If you add the bandit code and prepend it with "nosec" as shown below, it
will remove that scanning result from the metrics.


#hardcoded token/password in code
token = ";alkjdsalfjieajlh" # nosec: B105

#manually working with files in temp directory is not safe

temp_dir = "/tmp"
```

Ignoring Something Throughout Entire Program Instead of One by One with nosec

```
bandit insecure_code.py -s B105
```

Safety

```
#checks your project's dependencies for known security vulnerabilities

#to demonstrate its usage and value, I've installed a version of numpy that
has a known security vulnerability.

pip install numpy==1.22.1

safety check

#to get more details
safety check --full-report
```

```
-> Vulnerability found in numpy version 1.22.1
   Vulnerability ID: 44715
   Affected spec: <1.22.2
   ADVISORY: Numpy 1.22.2  includes a fix for
   CVE-2021-41495: Null Pointer Dereference vulnerability exists...
   CVE-2021-41495
   For more information about this vulnerability, visit
   https://data.safetycli.com/v/44715/97c
   To ignore this vulnerability, use PyUp vulnerability id 44715 in
   safety's ignore command-line argument or add the ignore to your
   safety policy file.
```

Full-Report Details

```
-> Vulnerability found in numpy version 1.22.1
   Vulnerability ID: 44715
   Affected spec: <1.22.2
   ADVISORY: 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.NOTE2: The specs we
   include in this advisory differ from the publicly available on
   other sources. For example, the advisory posted by the NVD
   indicate that versions up to and including 1.19.0 are affected.
   However, research by Safety CLI Cybersecurity confirms that the
   vulnerability remained unaddressed until version 1.22.2.
   CVE-2021-41495
   For more information about this vulnerability, visit
   https://data.safetycli.com/v/44715/97c
   To ignore this vulnerability, use PyUp vulnerability id 44715 in
   safety's ignore command-line argument or add the ignore to your
   safety policy file.
```