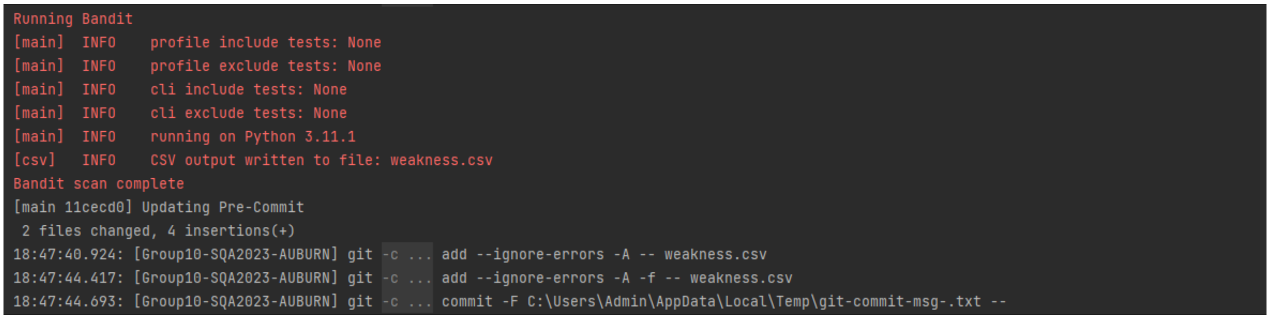
Group: Group 10 (Umar Mohammed and Pascal Onyeka)

GitHub Repository link: https://github.com/paschal27/Group10-SQA2023-AUBURN

Date: 4/26/2023

Report

1. **Security Weakness:** Pre-commit was created with bandit integration in a way that it reports on all Python files in the GitHub repository the weaknesses found. For Bandit to run, Pre-Commit must be put in .git/hooks. The output file is named “weakness.csv” in the home directory of the repository.



```
Running Bandit
[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 3.11.1
[csv]   INFO    CSV output written to file: weakness.csv
Bandit scan complete
[main 11cecd0] Updating Pre-Commit
 2 files changed, 4 insertions(+)
18:47:40.924: [Group10-SQA2023-AUBURN] git -c ... add --ignore-errors -A -- weakness.csv
18:47:44.417: [Group10-SQA2023-AUBURN] git -c ... add --ignore-errors -A -f -- weakness.csv
18:47:44.693: [Group10-SQA2023-AUBURN] git -c ... commit -F C:\Users\Admin\AppData\Local\Temp\git-commit-msg-.txt --
```

Example:

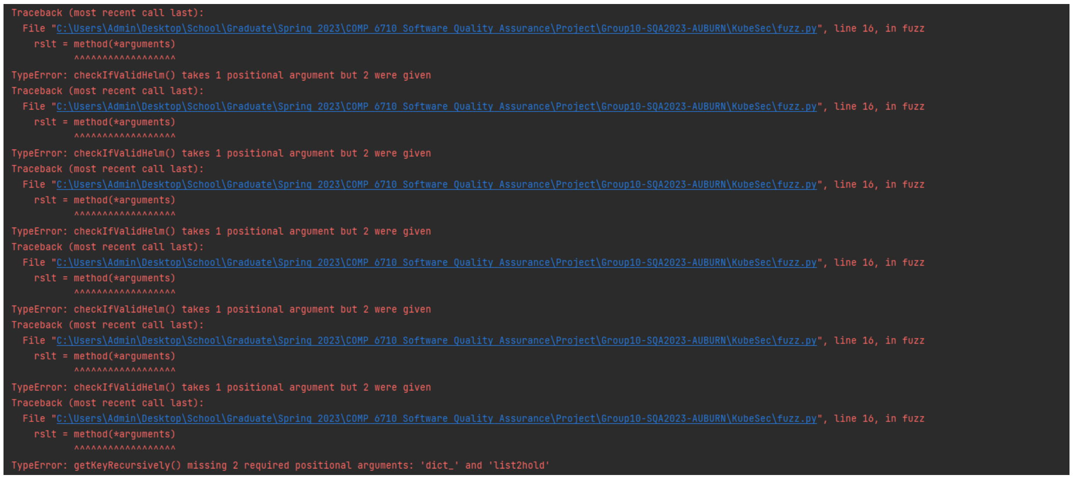
filename	**test_name**	**test_id**	**issue_severity**	**issue_confidence**	**issue_cwe**	**issue_text**	**line_number**	**col_offset**	**line_range**	**more_info**
KubeSec/TEST_CONSTANTS.py	hardcoded_password_string	B105	LOW	MEDIUM	https://cwe.mitre.org/data/definitions/259.html	Possible hardcoded password: 'TEST_ARTIFACTS/helm.values.yaml'	8	22	[8]	https://bandit.readthedocs.io/en/1.7.4/plugins/b105_hardcoded_password_string.html

2. **Fuzzing**: Five methods listed below were fuzzed from the “parser.py” file in the project’s home directory.

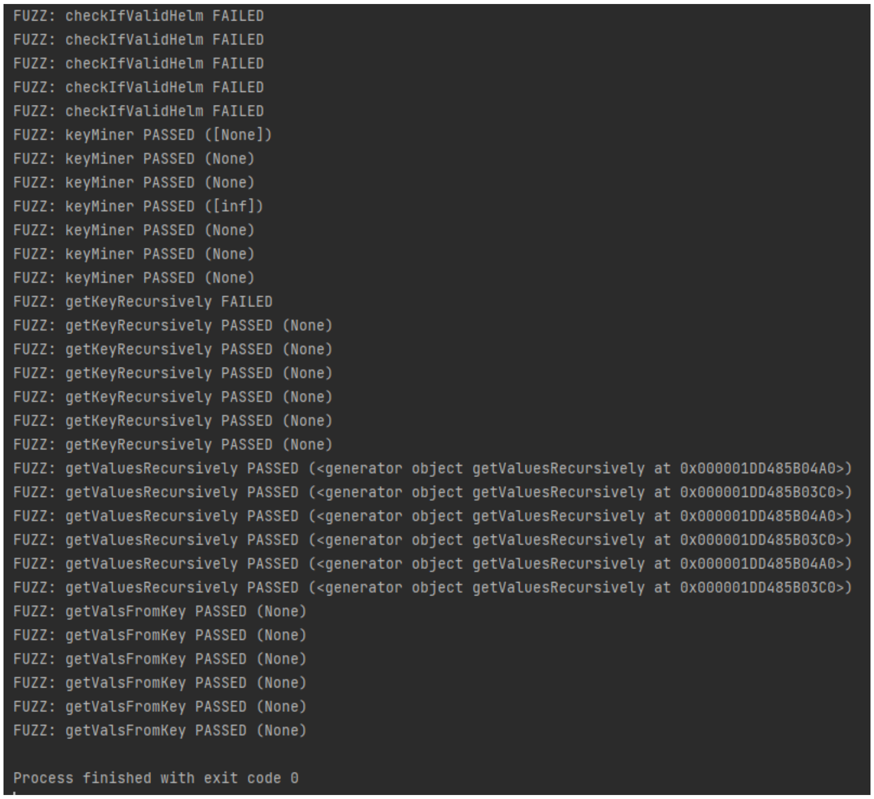
    a. checkIfValidHelm : TypeError: checkIfValidHelm() takes 1 positional argument but 2 were given
    b. keyMiner

c. getKeyRecursively
d. getValuesRecursively
e. GetValsFromKey

Output:



```
Traceback (most recent call last):
  File "C:\Users\Admin\Desktop\School\Graduate\Spring 2023\COMP 6710 Software Quality Assurance\Project\Group10-SQA2023-AUBURN\KubeSec\fuzz.py", line 16, in fuzz
    rslt = method(*arguments)
           ^^^^^^^^^^^^^^^^^^
TypeError: checkIfValidHelm() takes 1 positional argument but 2 were given
Traceback (most recent call last):
  File "C:\Users\Admin\Desktop\School\Graduate\Spring 2023\COMP 6710 Software Quality Assurance\Project\Group10-SQA2023-AUBURN\KubeSec\fuzz.py", line 16, in fuzz
    rslt = method(*arguments)
           ^^^^^^^^^^^^^^^^^^
TypeError: checkIfValidHelm() takes 1 positional argument but 2 were given
Traceback (most recent call last):
  File "C:\Users\Admin\Desktop\School\Graduate\Spring 2023\COMP 6710 Software Quality Assurance\Project\Group10-SQA2023-AUBURN\KubeSec\fuzz.py", line 16, in fuzz
    rslt = method(*arguments)
           ^^^^^^^^^^^^^^^^^^
TypeError: checkIfValidHelm() takes 1 positional argument but 2 were given
Traceback (most recent call last):
  File "C:\Users\Admin\Desktop\School\Graduate\Spring 2023\COMP 6710 Software Quality Assurance\Project\Group10-SQA2023-AUBURN\KubeSec\fuzz.py", line 16, in fuzz
    rslt = method(*arguments)
           ^^^^^^^^^^^^^^^^^^
TypeError: checkIfValidHelm() takes 1 positional argument but 2 were given
Traceback (most recent call last):
  File "C:\Users\Admin\Desktop\School\Graduate\Spring 2023\COMP 6710 Software Quality Assurance\Project\Group10-SQA2023-AUBURN\KubeSec\fuzz.py", line 16, in fuzz
    rslt = method(*arguments)
           ^^^^^^^^^^^^^^^^^^
TypeError: checkIfValidHelm() takes 1 positional argument but 2 were given
Traceback (most recent call last):
  File "C:\Users\Admin\Desktop\School\Graduate\Spring 2023\COMP 6710 Software Quality Assurance\Project\Group10-SQA2023-AUBURN\KubeSec\fuzz.py", line 16, in fuzz
    rslt = method(*arguments)
           ^^^^^^^^^^^^^^^^^^
TypeError: getKeyRecursively() missing 2 required positional arguments: 'dict_' and 'list2hold'
```



```
FUZZ: checkIfValidHelm FAILED
FUZZ: checkIfValidHelm FAILED
FUZZ: checkIfValidHelm FAILED
FUZZ: checkIfValidHelm FAILED
FUZZ: checkIfValidHelm FAILED
FUZZ: keyMiner PASSED ([None])
FUZZ: keyMiner PASSED (None)
FUZZ: keyMiner PASSED (None)
FUZZ: keyMiner PASSED ([inf])
FUZZ: keyMiner PASSED (None)
FUZZ: keyMiner PASSED (None)
FUZZ: keyMiner PASSED (None)
FUZZ: getKeyRecursively FAILED
FUZZ: getKeyRecursively PASSED (None)
FUZZ: getKeyRecursively PASSED (None)
FUZZ: getKeyRecursively PASSED (None)
FUZZ: getKeyRecursively PASSED (None)
FUZZ: getKeyRecursively PASSED (None)
FUZZ: getKeyRecursively PASSED (None)
FUZZ: getValuesRecursively PASSED (<generator object getValuesRecursively at 0x000001DD485B04A0>)
FUZZ: getValuesRecursively PASSED (<generator object getValuesRecursively at 0x000001DD485B03C0>)
FUZZ: getValuesRecursively PASSED (<generator object getValuesRecursively at 0x000001DD485B04A0>)
FUZZ: getValuesRecursively PASSED (<generator object getValuesRecursively at 0x000001DD485B03C0>)
FUZZ: getValuesRecursively PASSED (<generator object getValuesRecursively at 0x000001DD485B04A0>)
FUZZ: getValuesRecursively PASSED (<generator object getValuesRecursively at 0x000001DD485B03C0>)
FUZZ: getValsFromKey PASSED (None)
FUZZ: getValsFromKey PASSED (None)
FUZZ: getValsFromKey PASSED (None)
FUZZ: getValsFromKey PASSED (None)
FUZZ: getValsFromKey PASSED (None)
FUZZ: getValsFromKey PASSED (None)

Process finished with exit code 0
```

3. **Forensics Integration** – Forensics was integrated into 5 different functions as required by the project specification. A logging file was created which controlled the file format and log output format. The log output contained date, time and the message gotten corresponding to each function.

Conclusion

We learned how to implement git hooks and perform analysis on any file or changes made to the repository. Also, we improved our knowledge of fuzzing and logging. Understanding how to fuzz and perform logging on functions.