AUBURN

UNIVERSITY

Software Quality Assurance (COMP 6710)

Prof: Akond Rahman, PhD

Project Report

Date: April 27th, 2023

Done By:

Logan Smith - lws0022

Satya Sai Teja Charla - szc2039

Table of Contents

Summary:

The objective of this project is to integrate software quality assurance activities into an existing Python project. Whatever we learned from our workshops will be integrated in the project by apply the following activities related to software quality assurance:

1. Create a Git Hook that will run and report all security weaknesses in the project in a CSV file whenever a Python file is changed and committed.
2. Create a fuzz.py file that will automatically fuzz 5 Python methods of your choice. Report any bugs you discovered by the fuzz.py file. fuzz.py will be automatically executed from GitHub actions.
3. Integrate forensics by modifying 5 Python methods of your choice.

Project for Software Quality Assurance (CSC 5710/6710)

TeamLS-SQA2023-Auburn

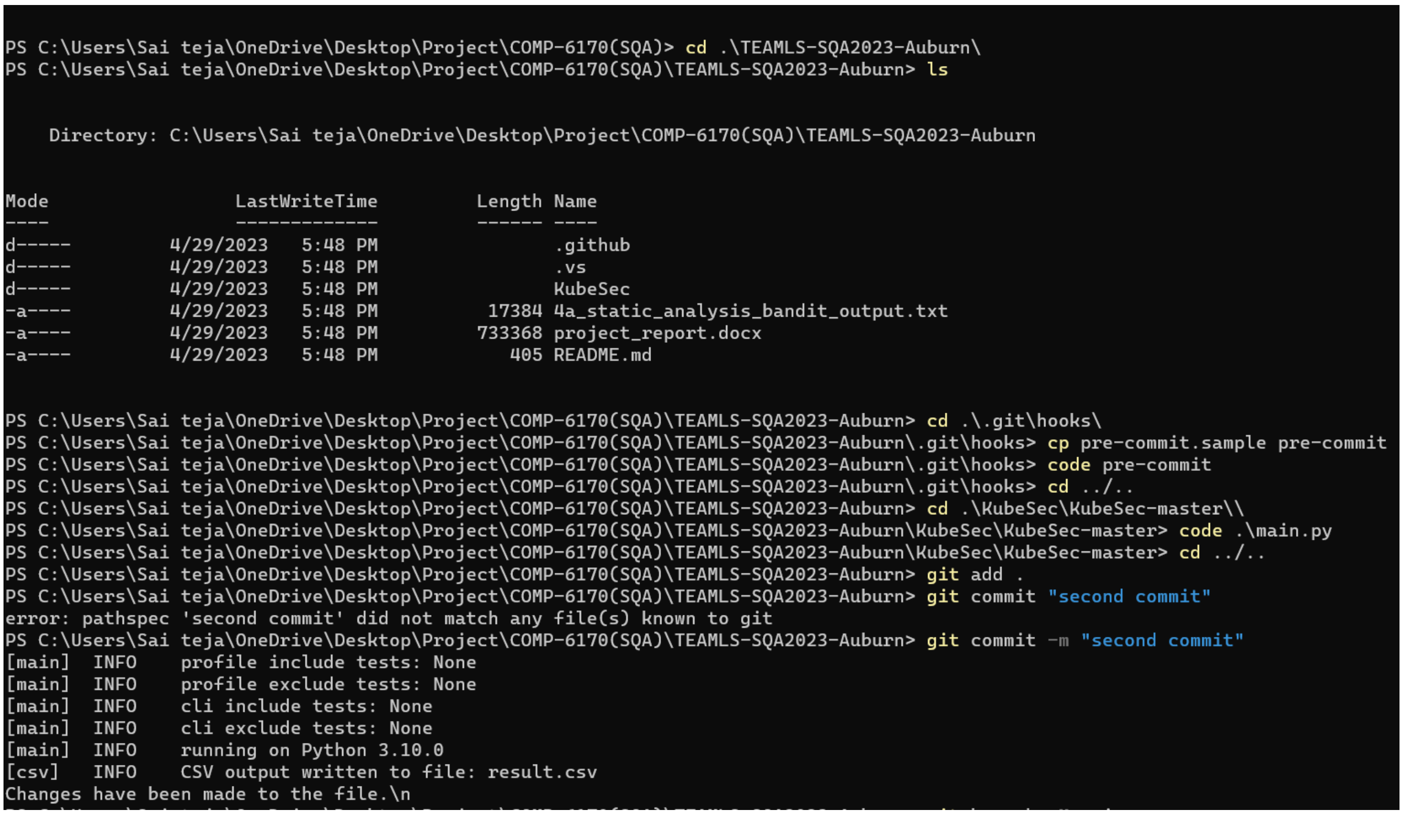
1. Static Analysis:

After creating a git-repo, I've cloned the repository onto my machine and made some changes to

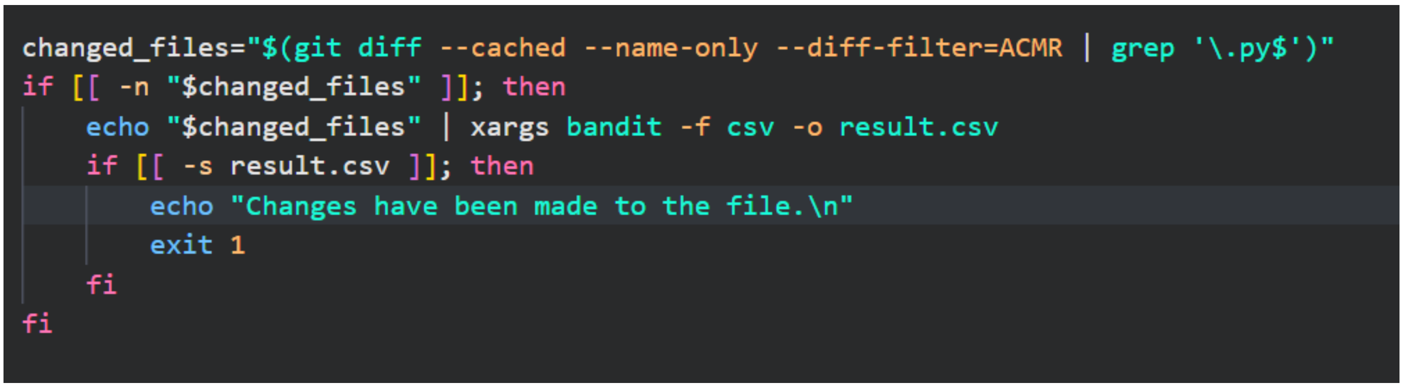
'./git/hooks/pre-commit.sample' file by copying the contents in that file and created a new file named '*pre-commit'* after that I've modified the main.py file in the same repository such that to see the effects of the modified *pre-commit* hook.

Finally, I ran bandit -r command to see any security weaknesses in the provided file and recorded the output.

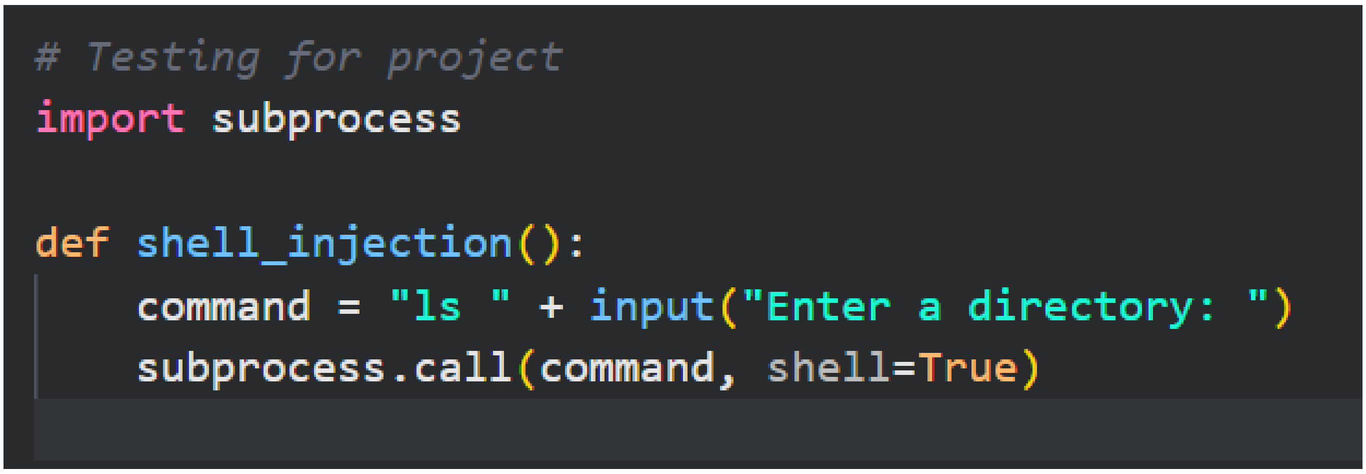
Down below are the screenshot taken during the execution of this segment:



```
PS C:\Users\Sai teja\OneDrive\Desktop\Project\COMP-6170(SQA)> cd .\TEAMLS-SQA2023-Auburn\
PS C:\Users\Sai teja\OneDrive\Desktop\Project\COMP-6170(SQA)\TEAMLS-SQA2023-Auburn> ls


    Directory: C:\Users\Sai teja\OneDrive\Desktop\Project\COMP-6170(SQA)\TEAMLS-SQA2023-Auburn


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----         4/29/2023   5:48 PM                .github
d-----         4/29/2023   5:48 PM                .vs
d-----         4/29/2023   5:48 PM                KubeSec
-a----         4/29/2023   5:48 PM          17384 4a_static_analysis_bandit_output.txt
-a----         4/29/2023   5:48 PM         733368 project_report.docx
-a----         4/29/2023   5:48 PM            405 README.md


PS C:\Users\Sai teja\OneDrive\Desktop\Project\COMP-6170(SQA)\TEAMLS-SQA2023-Auburn> cd .\.git\hooks\
PS C:\Users\Sai teja\OneDrive\Desktop\Project\COMP-6170(SQA)\TEAMLS-SQA2023-Auburn\.git\hooks> cp pre-commit.sample pre-commit
PS C:\Users\Sai teja\OneDrive\Desktop\Project\COMP-6170(SQA)\TEAMLS-SQA2023-Auburn\.git\hooks> code pre-commit
PS C:\Users\Sai teja\OneDrive\Desktop\Project\COMP-6170(SQA)\TEAMLS-SQA2023-Auburn\.git\hooks> cd ../..
PS C:\Users\Sai teja\OneDrive\Desktop\Project\COMP-6170(SQA)\TEAMLS-SQA2023-Auburn> cd .\KubeSec\KubeSec-master\\
PS C:\Users\Sai teja\OneDrive\Desktop\Project\COMP-6170(SQA)\TEAMLS-SQA2023-Auburn\KubeSec\KubeSec-master> code .\main.py
PS C:\Users\Sai teja\OneDrive\Desktop\Project\COMP-6170(SQA)\TEAMLS-SQA2023-Auburn\KubeSec\KubeSec-master> cd ../..
PS C:\Users\Sai teja\OneDrive\Desktop\Project\COMP-6170(SQA)\TEAMLS-SQA2023-Auburn> git add .
PS C:\Users\Sai teja\OneDrive\Desktop\Project\COMP-6170(SQA)\TEAMLS-SQA2023-Auburn> git commit "second commit"
error: pathspec 'second commit' did not match any file(s) known to git
PS C:\Users\Sai teja\OneDrive\Desktop\Project\COMP-6170(SQA)\TEAMLS-SQA2023-Auburn> git commit -m "second commit"
[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 3.10.0
[csv]   INFO    CSV output written to file: result.csv
Changes have been made to the file.\n
```

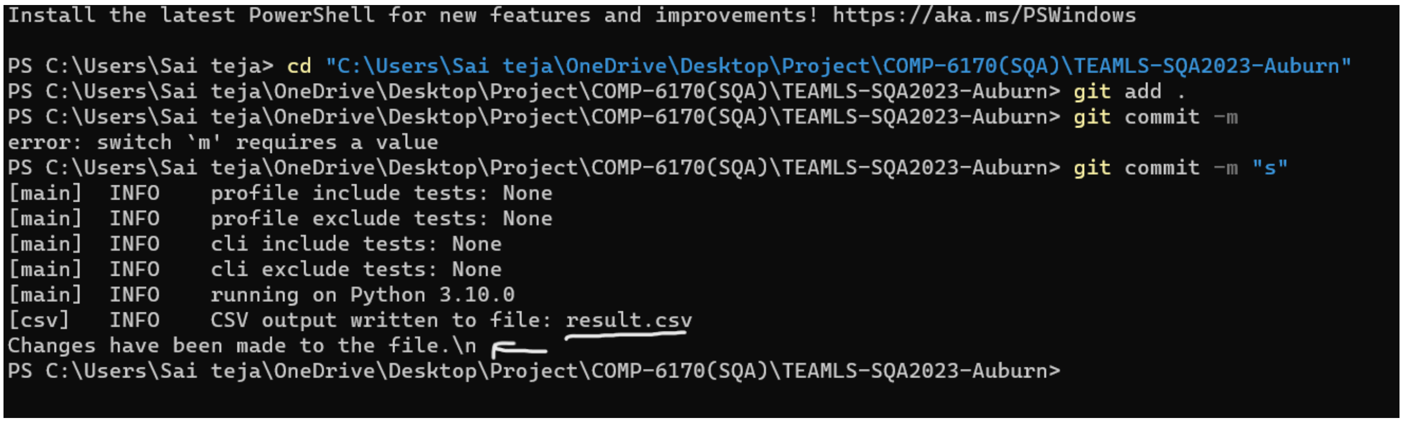


```
changed_files="$(git diff --cached --name-only --diff-filter=ACMR | grep '\.py$')"
if [[ -n "$changed_files" ]]; then
    echo "$changed_files" | xargs bandit -f csv -o result.csv
    if [[ -s result.csv ]]; then
        echo "Changes have been made to the file.\n"
        exit 1
    fi
fi
```



```python
# Testing for project
import subprocess

def shell_injection():
    command = "ls " + input("Enter a directory: ")
    subprocess.call(command, shell=True)
```

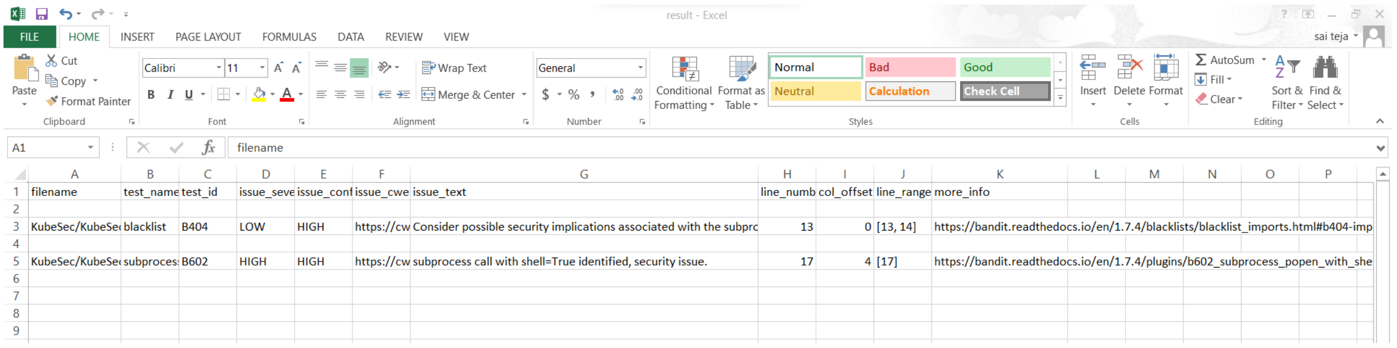
Now, when we use 'commit' on the terminal.



```
Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Sai teja> cd "C:\Users\Sai teja\OneDrive\Desktop\Project\COMP-6170(SQA)\TEAMLS-SQA2023-Auburn"
PS C:\Users\Sai teja\OneDrive\Desktop\Project\COMP-6170(SQA)\TEAMLS-SQA2023-Auburn> git add .
PS C:\Users\Sai teja\OneDrive\Desktop\Project\COMP-6170(SQA)\TEAMLS-SQA2023-Auburn> git commit -m
error: switch `m' requires a value
PS C:\Users\Sai teja\OneDrive\Desktop\Project\COMP-6170(SQA)\TEAMLS-SQA2023-Auburn> git commit -m "s"
[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 3.10.0
[csv]   INFO    CSV output written to file: result.csv
Changes have been made to the file.\n
PS C:\Users\Sai teja\OneDrive\Desktop\Project\COMP-6170(SQA)\TEAMLS-SQA2023-Auburn>
```

You can see that 'result.csv' has been created and it throws out the warning if any changes have been made to the files.

In that result.csv file, It records all the security weaknesses with filename, test_id, issues ..etc.

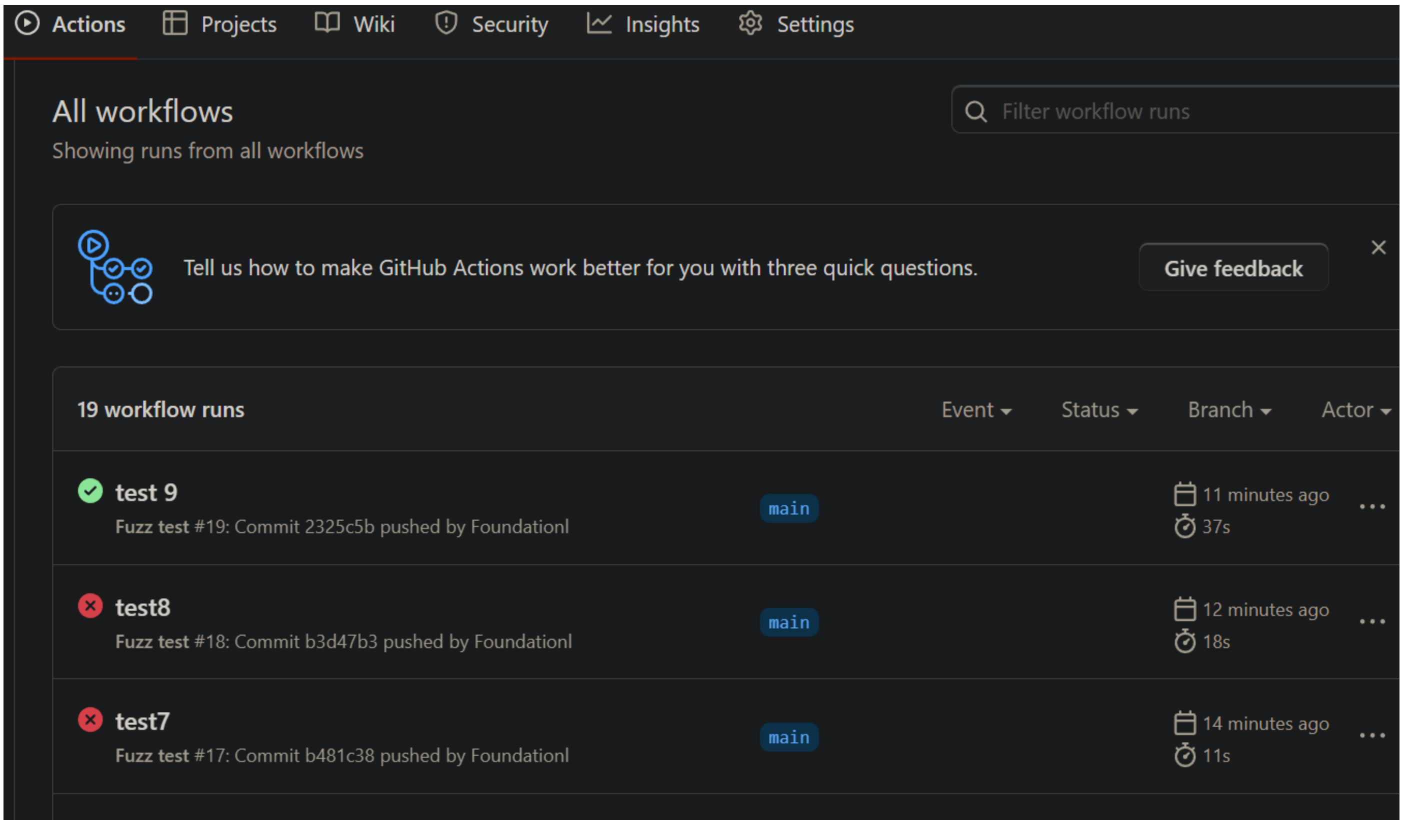


	A	B	C	D	E	F	G	H	I	J	K
1	filename	test_name	test_id	issue_seve	issue_conf	issue_cwe	issue_text	line_numb	col_offset	line_range	more_info
2											
3	KubeSec/KubeSe	blacklist	B404	LOW	HIGH	https://cw	Consider possible security implications associated with the subprc	13	0	[13, 14]	https://bandit.readthedocs.io/en/1.7.4/blacklists/blacklist_imports.html#b404-imp
4											
5	KubeSec/KubeSe	subproces	B602	HIGH	HIGH	https://cw	subprocess call with shell=True identified, security issue.	17	4	[17]	https://bandit.readthedocs.io/en/1.7.4/plugins/b602_subprocess_popen_with_she

I've tried pushing the pre-commit file but it didn't work, but I'll be uploading the result.csv file and pre-commit file on canvas if needed.

1. Fuzzing:

After creating a git-repo, I've cloned the repository onto my machine and made some changes. First, I had to add a ***'.github/workflow/main.yml'*** file so that on actions like pushing, the fuzzing will run on the 5 chosen functions and print a report. After several test and review iterations I was able to print a report from workflows.



All workflows

Showing runs from all workflows

Tell us how to make GitHub Actions work better for you with three quick questions.

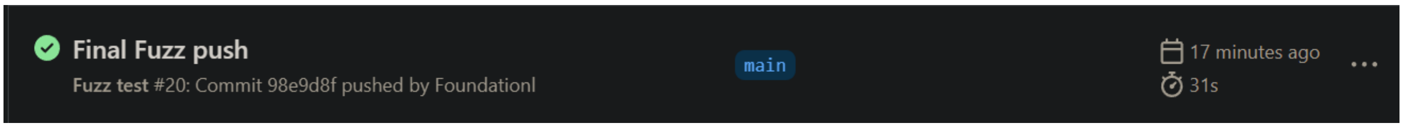
19 workflow runs

test 9
Fuzz test #19: Commit 2325c5b pushed by Foundationl — main — 11 minutes ago — 37s

test8
Fuzz test #18: Commit b3d47b3 pushed by Foundationl — main — 12 minutes ago — 18s

test7
Fuzz test #17: Commit b481c38 pushed by Foundationl — main — 14 minutes ago — 11s

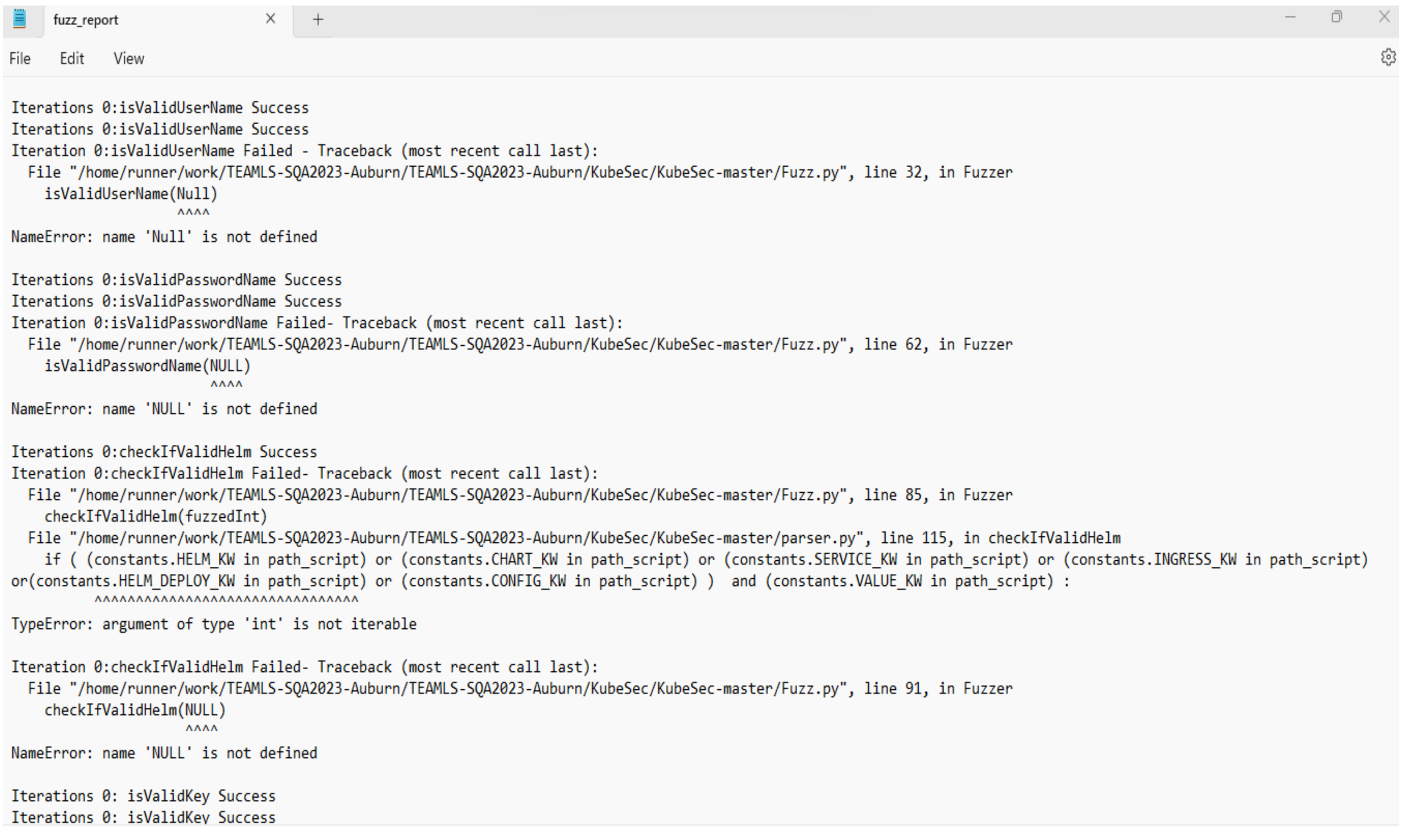
I now was ready to set up my Fuzzing function. This involved finding 5 methods throughout the zip to test. These chosen methods where {Class Scanner [Function isValidUserName, isValidPasswordName, isValidKey], Class Parser [Function keyMiner, checkIfValidHelm]}. The inputs chosen where a random generated int, a random generated string of fixed size, and NULL. The Fuzz.py will test those 5 methods with these inputs. The fuzzing function would then print successful and unsuccessful tests in a report. The unsuccessful test will also display the error associated with the failure. When pushing to GitHub you can go to actions and find under workflows the latest commit and view report, for reference the final working Fuzz push.



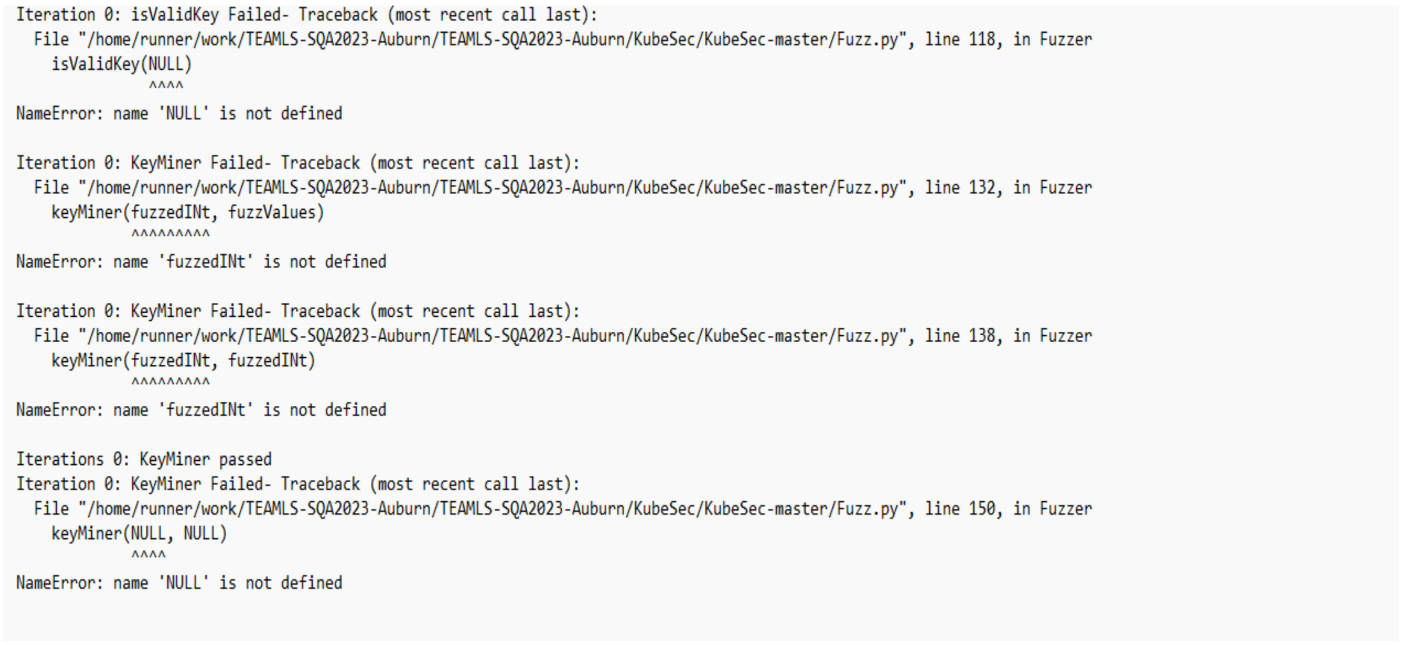
Final Fuzz push
Fuzz test #20: Commit 98e9d8f pushed by Foundationl — main — 17 minutes ago — 31s

In this workflow you will be able to see the fuzz report which prints the tests after first iteration.

Fuzz.py output:



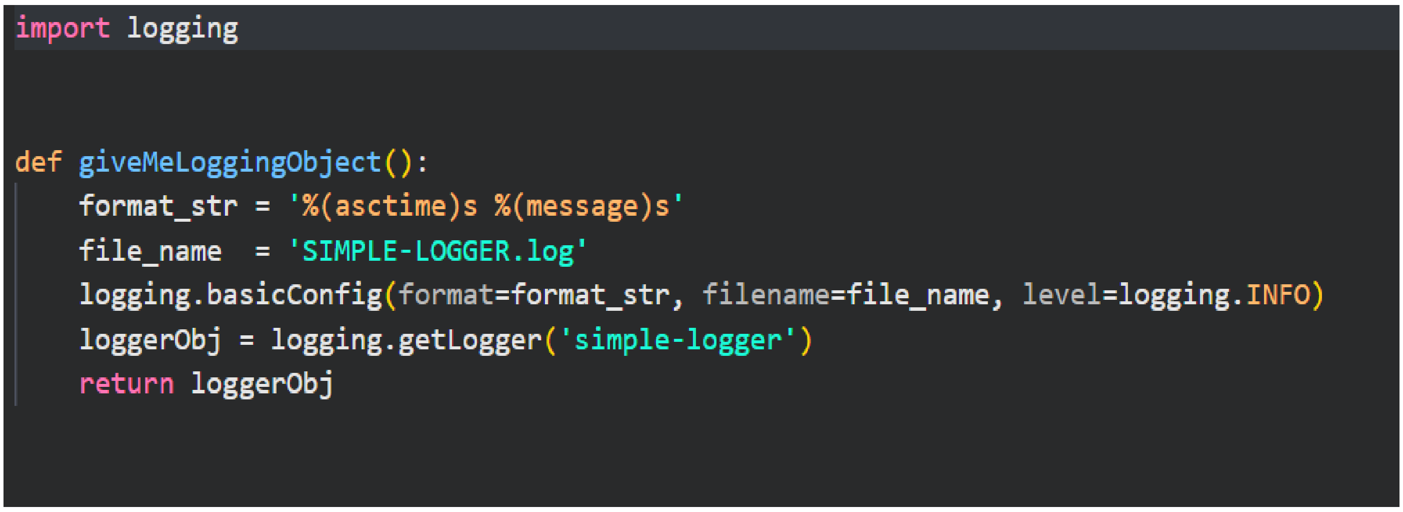
```
Iterations 0:isValidUserName Success
Iterations 0:isValidUserName Success
Iteration 0:isValidUserName Failed - Traceback (most recent call last):
  File "/home/runner/work/TEAMLS-SQA2023-Auburn/TEAMLS-SQA2023-Auburn/KubeSec/KubeSec-master/Fuzz.py", line 32, in Fuzzer
    isValidUserName(Null)
                    ^^^^
NameError: name 'Null' is not defined

Iterations 0:isValidPasswordName Success
Iterations 0:isValidPasswordName Success
Iteration 0:isValidPasswordName Failed- Traceback (most recent call last):
  File "/home/runner/work/TEAMLS-SQA2023-Auburn/TEAMLS-SQA2023-Auburn/KubeSec/KubeSec-master/Fuzz.py", line 62, in Fuzzer
    isValidPasswordName(NULL)
                        ^^^^
NameError: name 'NULL' is not defined

Iterations 0:checkIfValidHelm Success
Iteration 0:checkIfValidHelm Failed- Traceback (most recent call last):
  File "/home/runner/work/TEAMLS-SQA2023-Auburn/TEAMLS-SQA2023-Auburn/KubeSec/KubeSec-master/Fuzz.py", line 85, in Fuzzer
    checkIfValidHelm(fuzzedInt)
  File "/home/runner/work/TEAMLS-SQA2023-Auburn/TEAMLS-SQA2023-Auburn/KubeSec/KubeSec-master/parser.py", line 115, in checkIfValidHelm
    if ( (constants.HELM_KW in path_script) or (constants.CHART_KW in path_script) or (constants.SERVICE_KW in path_script) or (constants.INGRESS_KW in path_script)
or(constants.HELM_DEPLOY_KW in path_script) or (constants.CONFIG_KW in path_script) )  and (constants.VALUE_KW in path_script) :
          ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
TypeError: argument of type 'int' is not iterable

Iteration 0:checkIfValidHelm Failed- Traceback (most recent call last):
  File "/home/runner/work/TEAMLS-SQA2023-Auburn/TEAMLS-SQA2023-Auburn/KubeSec/KubeSec-master/Fuzz.py", line 91, in Fuzzer
    checkIfValidHelm(NULL)
                     ^^^^
NameError: name 'NULL' is not defined

Iterations 0: isValidKey Success
Iterations 0: isValidKey Success
```



```
Iteration 0: isValidKey Failed- Traceback (most recent call last):
  File "/home/runner/work/TEAMLS-SQA2023-Auburn/TEAMLS-SQA2023-Auburn/KubeSec/KubeSec-master/Fuzz.py", line 118, in Fuzzer
    isValidKey(NULL)
               ^^^^
NameError: name 'NULL' is not defined

Iteration 0: KeyMiner Failed- Traceback (most recent call last):
  File "/home/runner/work/TEAMLS-SQA2023-Auburn/TEAMLS-SQA2023-Auburn/KubeSec/KubeSec-master/Fuzz.py", line 132, in Fuzzer
    keyMiner(fuzzedINt, fuzzValues)
             ^^^^^^^^^
NameError: name 'fuzzedINt' is not defined

Iteration 0: KeyMiner Failed- Traceback (most recent call last):
  File "/home/runner/work/TEAMLS-SQA2023-Auburn/TEAMLS-SQA2023-Auburn/KubeSec/KubeSec-master/Fuzz.py", line 138, in Fuzzer
    keyMiner(fuzzedINt, fuzzedINt)
             ^^^^^^^^^
NameError: name 'fuzzedINt' is not defined

Iterations 0: KeyMiner passed
Iteration 0: KeyMiner Failed- Traceback (most recent call last):
  File "/home/runner/work/TEAMLS-SQA2023-Auburn/TEAMLS-SQA2023-Auburn/KubeSec/KubeSec-master/Fuzz.py", line 150, in Fuzzer
    keyMiner(NULL, NULL)
             ^^^^
NameError: name 'NULL' is not defined
```

3. Software Forensics:

Created a simple python file which will log whenever a particular python method is used.

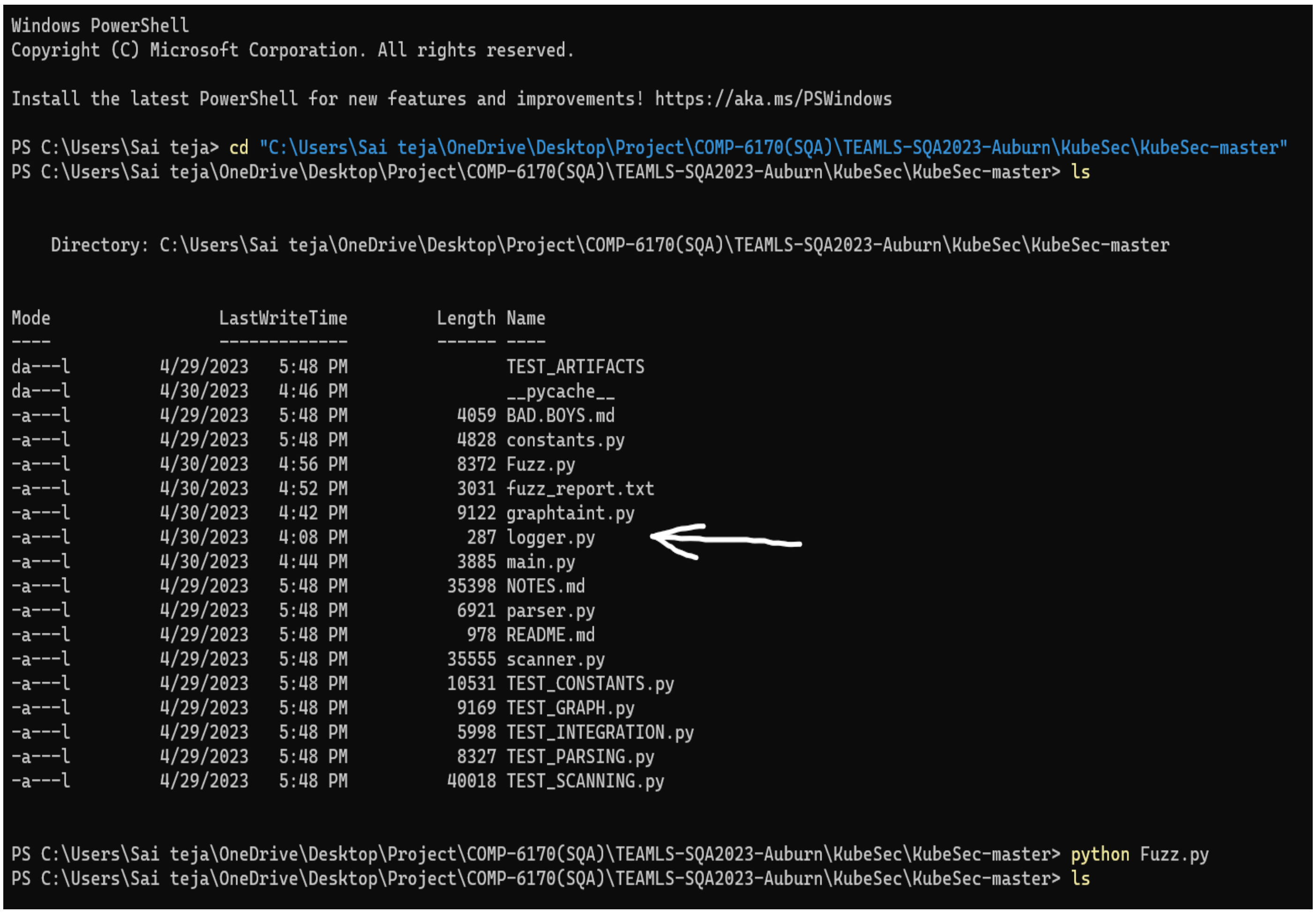


```
import logging


def giveMeLoggingObject():
    format_str = '%(asctime)s %(message)s'
    file_name  = 'SIMPLE-LOGGER.log'
    logging.basicConfig(format=format_str, filename=file_name, level=logging.INFO)
    loggerObj = logging.getLogger('simple-logger')
    return loggerObj
```

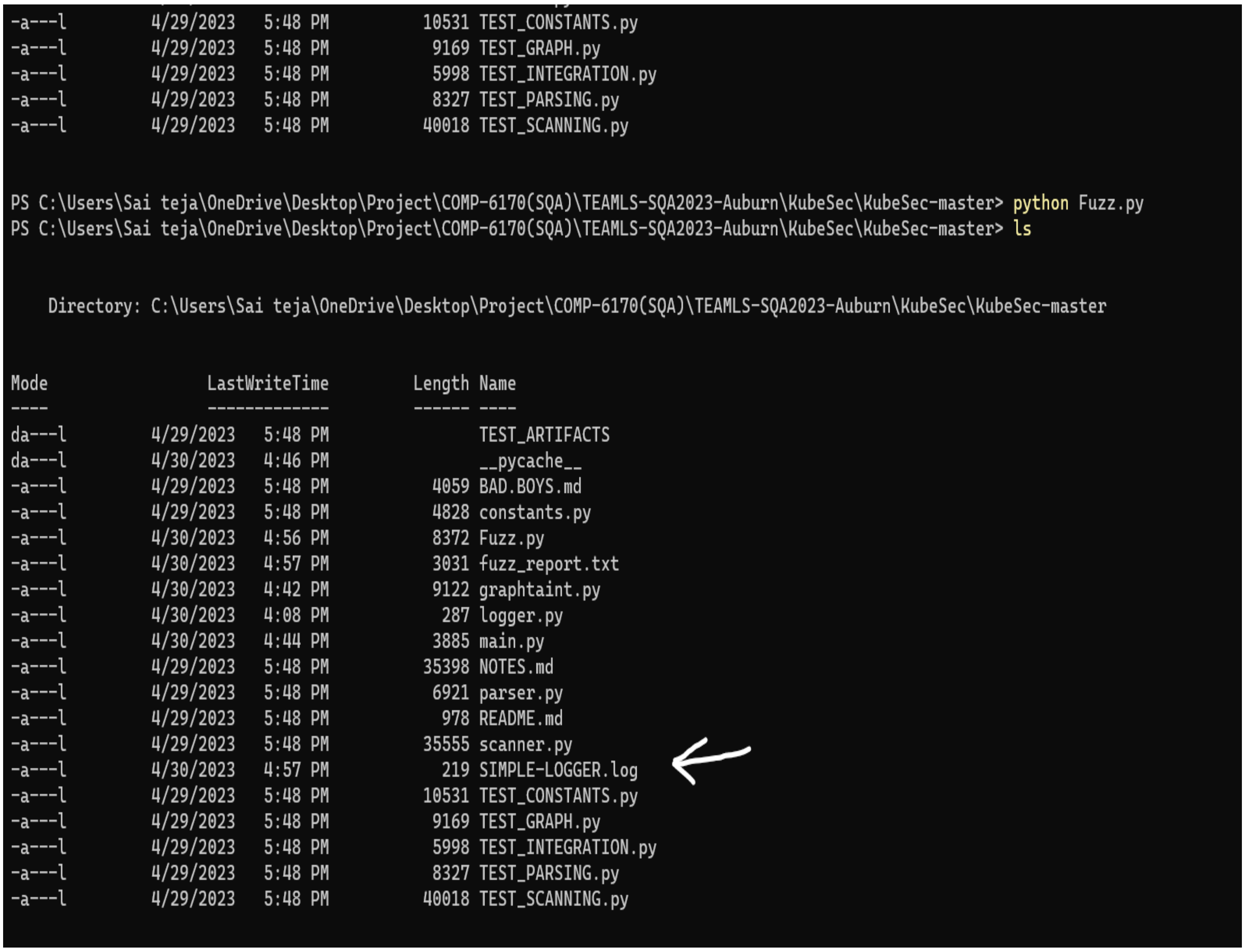
After importing this file onto Fuzz.py file, to actually record it, first we need to use the following snippet of code.

```
logObj = logger.giveMeLoggingObject()
logObj.info( <Add the comment> )
```

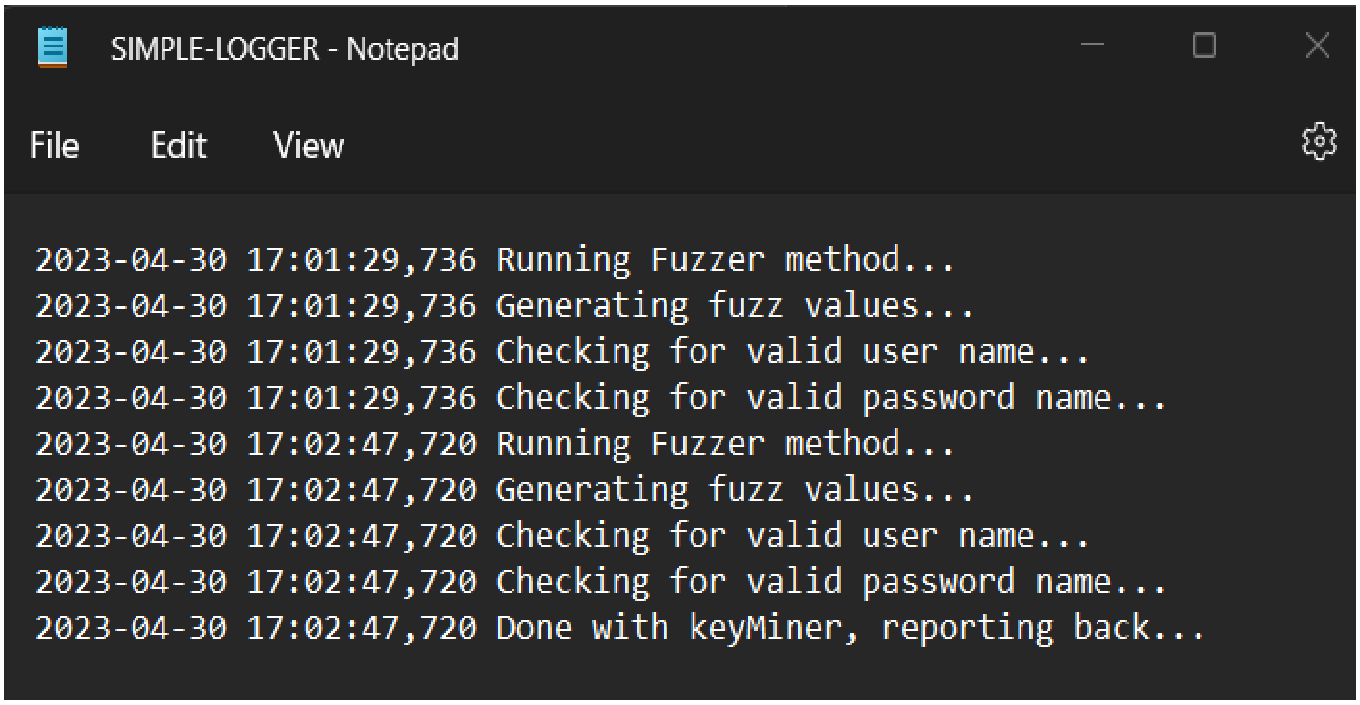
After running that particular file, it will automatically generate a log file in the current directory.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Sai teja> cd "C:\Users\Sai teja\OneDrive\Desktop\Project\COMP-6170(SQA)\TEAMLS-SQA2023-Auburn\KubeSec\KubeSec-master"
PS C:\Users\Sai teja\OneDrive\Desktop\Project\COMP-6170(SQA)\TEAMLS-SQA2023-Auburn\KubeSec\KubeSec-master> ls


    Directory: C:\Users\Sai teja\OneDrive\Desktop\Project\COMP-6170(SQA)\TEAMLS-SQA2023-Auburn\KubeSec\KubeSec-master


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
da---l         4/29/2023   5:48 PM                TEST_ARTIFACTS
da---l         4/30/2023   4:46 PM                __pycache__
-a---l         4/29/2023   5:48 PM           4059 BAD.BOYS.md
-a---l         4/29/2023   5:48 PM           4828 constants.py
-a---l         4/30/2023   4:56 PM           8372 Fuzz.py
-a---l         4/30/2023   4:52 PM           3031 fuzz_report.txt
-a---l         4/30/2023   4:42 PM           9122 graphtaint.py
-a---l         4/30/2023   4:08 PM            287 logger.py
-a---l         4/30/2023   4:44 PM           3885 main.py
-a---l         4/29/2023   5:48 PM          35398 NOTES.md
-a---l         4/29/2023   5:48 PM           6921 parser.py
-a---l         4/29/2023   5:48 PM            978 README.md
-a---l         4/29/2023   5:48 PM          35555 scanner.py
-a---l         4/29/2023   5:48 PM          10531 TEST_CONSTANTS.py
-a---l         4/29/2023   5:48 PM           9169 TEST_GRAPH.py
-a---l         4/29/2023   5:48 PM           5998 TEST_INTEGRATION.py
-a---l         4/29/2023   5:48 PM           8327 TEST_PARSING.py
-a---l         4/29/2023   5:48 PM          40018 TEST_SCANNING.py


PS C:\Users\Sai teja\OneDrive\Desktop\Project\COMP-6170(SQA)\TEAMLS-SQA2023-Auburn\KubeSec\KubeSec-master> python Fuzz.py
PS C:\Users\Sai teja\OneDrive\Desktop\Project\COMP-6170(SQA)\TEAMLS-SQA2023-Auburn\KubeSec\KubeSec-master> ls
```



```
-a---l         4/29/2023   5:48 PM          10531 TEST_CONSTANTS.py
-a---l         4/29/2023   5:48 PM           9169 TEST_GRAPH.py
-a---l         4/29/2023   5:48 PM           5998 TEST_INTEGRATION.py
-a---l         4/29/2023   5:48 PM           8327 TEST_PARSING.py
-a---l         4/29/2023   5:48 PM          40018 TEST_SCANNING.py


PS C:\Users\Sai teja\OneDrive\Desktop\Project\COMP-6170(SQA)\TEAMLS-SQA2023-Auburn\KubeSec\KubeSec-master> python Fuzz.py
PS C:\Users\Sai teja\OneDrive\Desktop\Project\COMP-6170(SQA)\TEAMLS-SQA2023-Auburn\KubeSec\KubeSec-master> ls


    Directory: C:\Users\Sai teja\OneDrive\Desktop\Project\COMP-6170(SQA)\TEAMLS-SQA2023-Auburn\KubeSec\KubeSec-master


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
da---l         4/29/2023   5:48 PM                TEST_ARTIFACTS
da---l         4/30/2023   4:46 PM                __pycache__
-a---l         4/29/2023   5:48 PM           4059 BAD.BOYS.md
-a---l         4/29/2023   5:48 PM           4828 constants.py
-a---l         4/30/2023   4:56 PM           8372 Fuzz.py
-a---l         4/30/2023   4:57 PM           3031 fuzz_report.txt
-a---l         4/30/2023   4:42 PM           9122 graphtaint.py
-a---l         4/30/2023   4:08 PM            287 logger.py
-a---l         4/30/2023   4:44 PM           3885 main.py
-a---l         4/29/2023   5:48 PM          35398 NOTES.md
-a---l         4/29/2023   5:48 PM           6921 parser.py
-a---l         4/29/2023   5:48 PM            978 README.md
-a---l         4/29/2023   5:48 PM          35555 scanner.py
-a---l         4/30/2023   4:57 PM            219 SIMPLE-LOGGER.log
-a---l         4/29/2023   5:48 PM          10531 TEST_CONSTANTS.py
-a---l         4/29/2023   5:48 PM           9169 TEST_GRAPH.py
-a---l         4/29/2023   5:48 PM           5998 TEST_INTEGRATION.py
-a---l         4/29/2023   5:48 PM           8327 TEST_PARSING.py
-a---l         4/29/2023   5:48 PM          40018 TEST_SCANNING.py
```



```
2023-04-30 17:01:29,736 Running Fuzzer method...
2023-04-30 17:01:29,736 Generating fuzz values...
2023-04-30 17:01:29,736 Checking for valid user name...
2023-04-30 17:01:29,736 Checking for valid password name...
2023-04-30 17:02:47,720 Running Fuzzer method...
2023-04-30 17:02:47,720 Generating fuzz values...
2023-04-30 17:02:47,720 Checking for valid user name...
2023-04-30 17:02:47,720 Checking for valid password name...
2023-04-30 17:02:47,720 Done with keyMiner, reporting back...
```

Conclusion:

To summarize, my teammate and I have integrated software quality assurance activities into an existing Python project.

Static Analysis: Created a git hook which will report any security weaknesses detected from the files. If any changes are made to the existing files, all the security weaknesses in it will be reported in a CSV file.

Fuzzing: Created a python file called ‘Fuzz.py’ which will run on the 5 chosen functions and print a report. After several test and review iterations we were able to print a report from workflows.

Forensics: This last bit of project was easier to implement compared to others, but it is an important one.

Whatever we learned from our workshops has been integrated in this project.

References:

Canvas files -	https://auburn.instructure.com/courses/1478245/files
StackOverFlow -	https://stackoverflow.com/