AUBURN

UNIVERSITY

Project for Software Quality Assurance (COMP 5710/6710)

Team Members: Mary Mitchell and Adia Foster

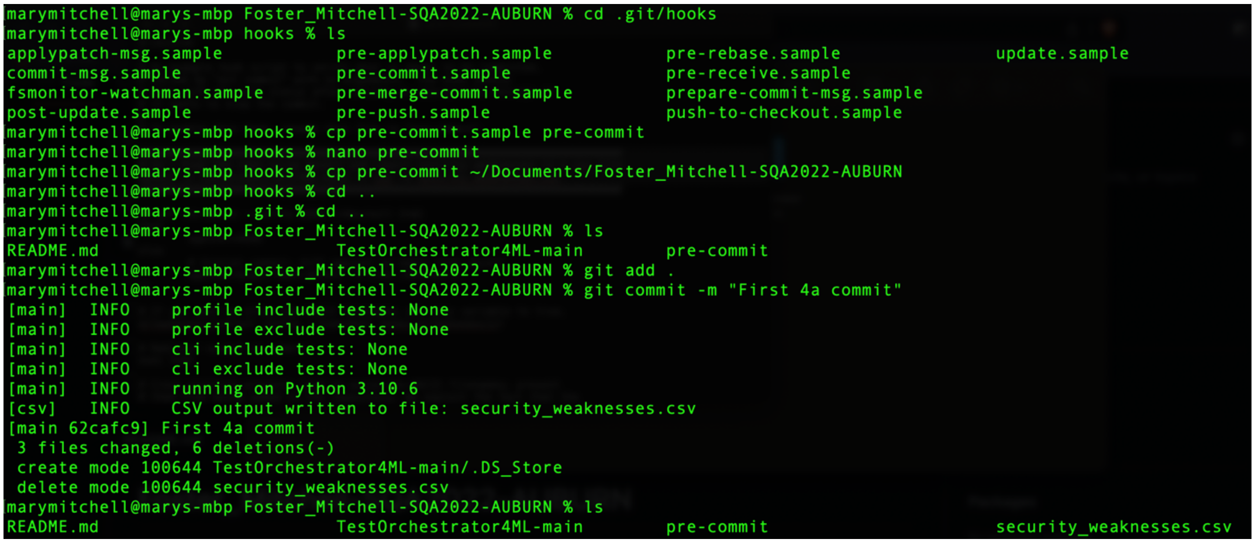
Team Name: Foster_Mitchell

1 December 2022

Table of Contents

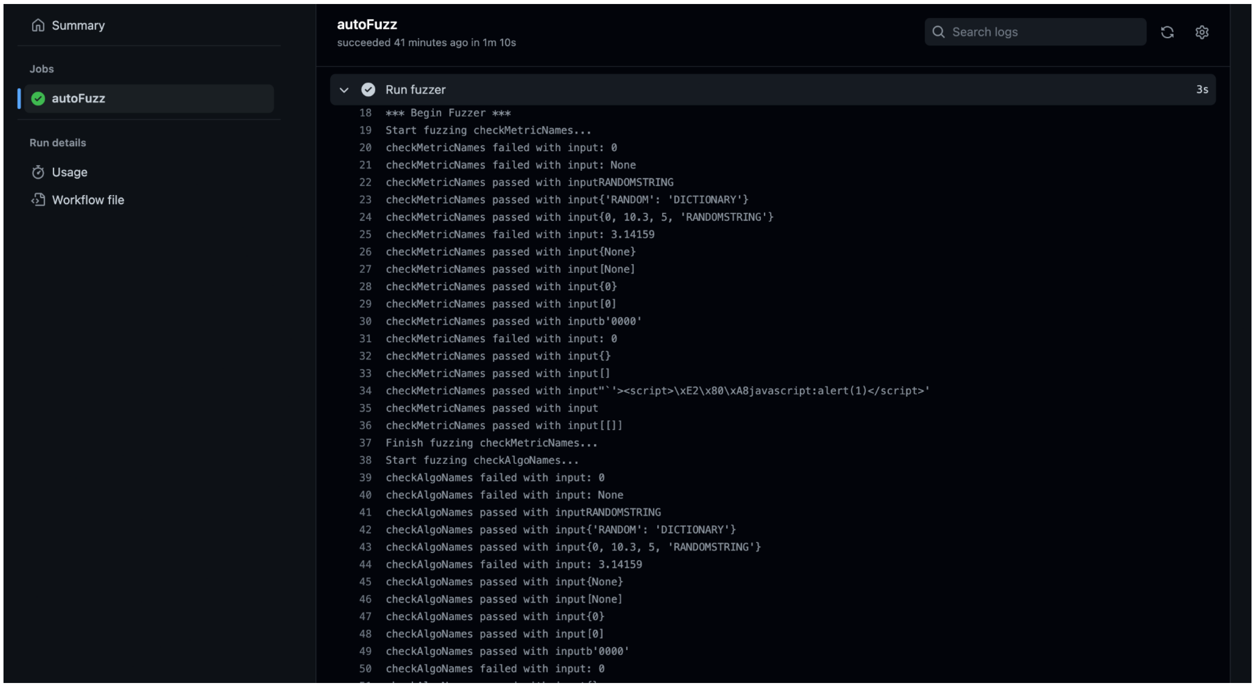
List of Figures

Figure 1: Example of making the git hook



```
marymitchell@marys-mbp Foster_Mitchell-SQA2022-AUBURN % cd .git/hooks
marymitchell@marys-mbp hooks % ls
applypatch-msg.sample           pre-applypatch.sample           pre-rebase.sample               update.sample
commit-msg.sample               pre-commit.sample               pre-receive.sample
fsmonitor-watchman.sample       pre-merge-commit.sample         prepare-commit-msg.sample
post-update.sample              pre-push.sample                 push-to-checkout.sample
marymitchell@marys-mbp hooks % cp pre-commit.sample pre-commit
marymitchell@marys-mbp hooks % nano pre-commit
marymitchell@marys-mbp hooks % cp pre-commit ~/Documents/Foster_Mitchell-SQA2022-AUBURN
marymitchell@marys-mbp hooks % cd ..
marymitchell@marys-mbp .git % cd ..
marymitchell@marys-mbp Foster_Mitchell-SQA2022-AUBURN % ls
README.md                       TestOrchestrator4ML-main        pre-commit
marymitchell@marys-mbp Foster_Mitchell-SQA2022-AUBURN % git add .
marymitchell@marys-mbp Foster_Mitchell-SQA2022-AUBURN % git commit -m "First 4a 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 3.10.6
[csv]   INFO    CSV output written to file: security_weaknesses.csv
[main 62cafc9] First 4a commit
 3 files changed, 6 deletions(-)
 create mode 100644 TestOrchestrator4ML-main/.DS_Store
 delete mode 100644 security_weaknesses.csv
marymitchell@marys-mbp Foster_Mitchell-SQA2022-AUBURN % ls
README.md                       TestOrchestrator4ML-main        pre-commit                      security_weaknesses.csv
```

Figure 2: Output of ‘fuzz.py’ in GitHub actions

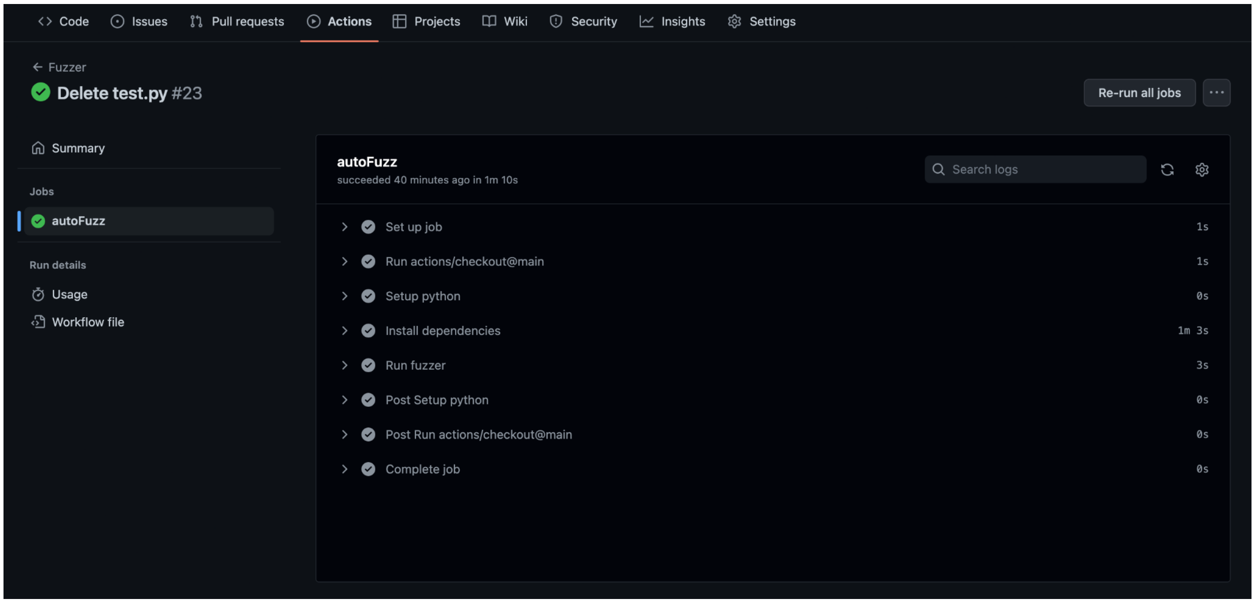


autoFuzz

succeeded 41 minutes ago in 1m 10s

```
*** Begin Fuzzer ***
Start fuzzing checkMetricNames...
checkMetricNames failed with input: 0
checkMetricNames failed with input: None
checkMetricNames passed with inputRANDOMSTRING
checkMetricNames passed with input{'RANDOM': 'DICTIONARY'}
checkMetricNames passed with input{0, 10.3, 5, 'RANDOMSTRING'}
checkMetricNames failed with input: 3.14159
checkMetricNames passed with input{None}
checkMetricNames passed with input[None]
checkMetricNames passed with input{0}
checkMetricNames passed with input[0]
checkMetricNames passed with inputb'0000'
checkMetricNames failed with input: 0
checkMetricNames passed with input{}
checkMetricNames passed with input[]
checkMetricNames passed with input"`'><script>\xE2\x80\xA8javascript:alert(1)</script>'
checkMetricNames passed with input
checkMetricNames passed with input[[]]
Finish fuzzing checkMetricNames...
Start fuzzing checkAlgoNames...
checkAlgoNames failed with input: 0
checkAlgoNames failed with input: None
checkAlgoNames passed with inputRANDOMSTRING
checkAlgoNames passed with input{'RANDOM': 'DICTIONARY'}
checkAlgoNames passed with input{0, 10.3, 5, 'RANDOMSTRING'}
checkAlgoNames failed with input: 3.14159
checkAlgoNames passed with input{None}
checkAlgoNames passed with input[None]
checkAlgoNames passed with input{0}
checkAlgoNames passed with input[0]
checkAlgoNames passed with inputb'0000'
checkAlgoNames failed with input: 0
```

Figure 3: Example of a successful GitHub action



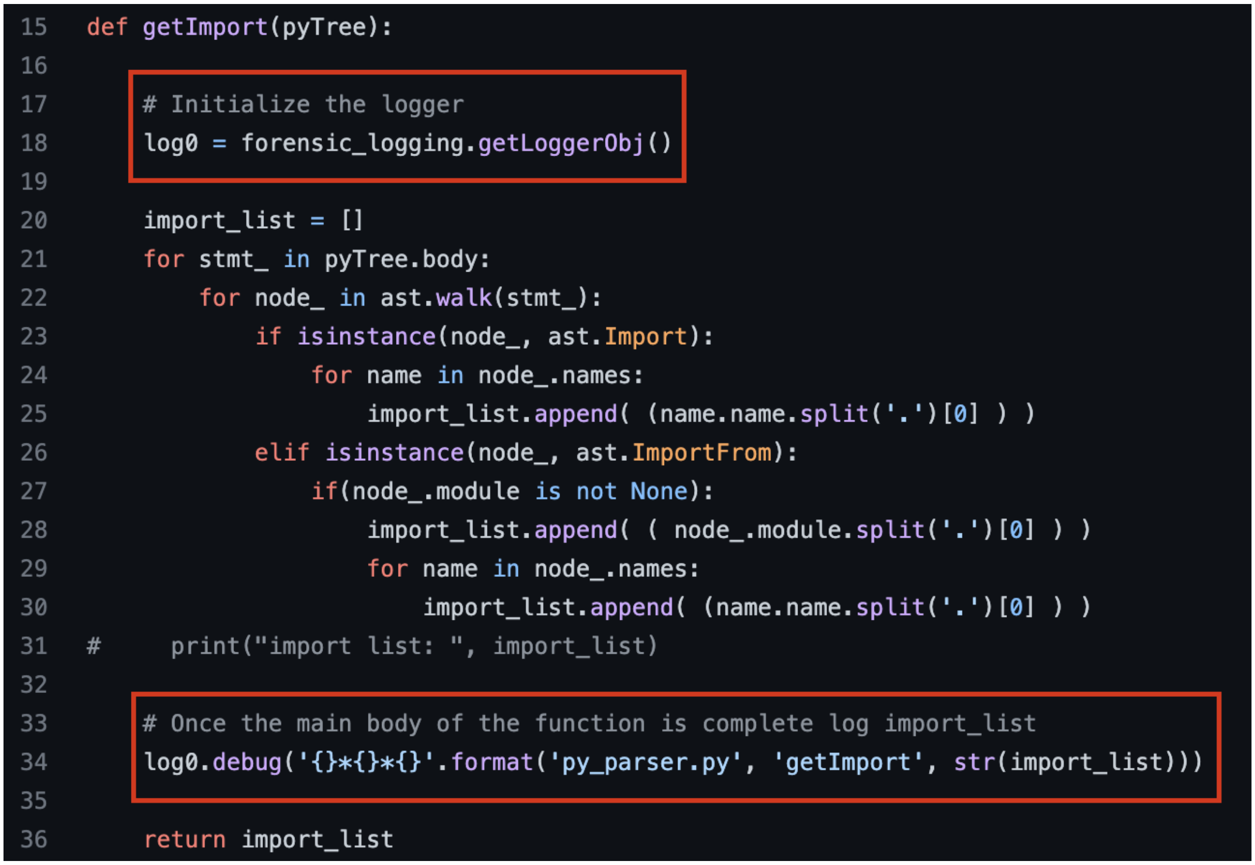
Delete test.py #23

autoFuzz

autoFuzz
succeeded 40 minutes ago in 1m 10s

```
Set up job                           1s
Run actions/checkout@main            1s
Setup python                         0s
Install dependencies              1m 3s
Run fuzzer                           3s
Post Setup python                    0s
Post Run actions/checkout@main       0s
Complete job                         0s
```

Figure 4: Logging commands



```python
def getImport(pyTree):

    # Initialize the logger
    log0 = forensic_logging.getLoggerObj()

    import_list = []
    for stmt_ in pyTree.body:
        for node_ in ast.walk(stmt_):
            if isinstance(node_, ast.Import):
                for name in node_.names:
                    import_list.append( (name.name.split('.')[0] ) )
            elif isinstance(node_, ast.ImportFrom):
                if(node_.module is not None):
                    import_list.append( ( node_.module.split('.')[0] ) )
                    for name in node_.names:
                        import_list.append( (name.name.split('.')[0] ) )
#     print("import list: ", import_list)

    # Once the main body of the function is complete log import_list
    log0.debug('{}*{}*{}'.format('py_parser.py', 'getImport', str(import_list)))

    return import_list
```

List of Tables

Table 1: Output of security weaknesses from git hook

filename	test_name	test_id	issue_severity	issue_confidence	issue_cwe	issue_text	line_number	col_offset	line_range	more_info
TestOrchestrator4ML-main/generation/probability_based_label_perturbation.py	blacklist	B311	LOW	HIGH	https://cwe.mitre.org/data/definitions/330.html	Standard pseudo-random generators are not suitable for security/cryptographic purposes.	28	40	[28]	https://bandit.readthedocs.io/en/1.7.4/blacklists/blacklist_calls.html#b311-random
TestOrchestrator4ML-main/label_perturbation_attack/probability_based_label_perturbation.py	blacklist	B311	LOW	HIGH	https://cwe.mitre.org/data/definitions/330.html	Standard pseudo-random generators are not suitable for security/cryptographic purposes.	28	40	[28]	https://bandit.readthedocs.io/en/1.7.4/blacklists/blacklist_calls.html#b311-random
TestOrchestrator4ML-main/select_repos/dev_count.py	blacklist	B404	LOW	HIGH	https://cwe.mitre.org/data/definitions/78.html	Consider possible security implications associated with the subprocess module.	7	0	[7]	https://bandit.readthedocs.io/en/1.7.4/blacklists/blacklist_imports.html#b404-import-subprocess
TestOrchestrator4ML-main/select_repos/dev_count.py	start_process_with_partial_path	B607	LOW	HIGH	https://cwe.mitre.org/data/definitions/78.html	Starting a process with a partial executable path	26	24	[26]	https://bandit.readthedocs.io/en/1.7.4/plugins/b607_start_process_with_partial_path.html
TestOrchestrator4ML-main/select_repos/dev_count.py	subprocess_without_shell_equals_true	B603	LOW	HIGH	https://cwe.mitre.org/data/definitions/78.html	subprocess call - check for execution of untrusted input.	26	24	[26]	https://bandit.readthedocs.io/en/1.7.4/plugins/b603_subprocess_without_shell_equals_true.html

1 Part 4a – Git Hook

The first main task in this project was to create a git hook that would run anytime a file in the project repository was changed and committed. The job of this git hook was to scan the TestOrchestrator4ML repository and report any security weaknesses in a csv file. We did this by first going to the hooks directory contained in the .git directory of the repository. Once we were there, we copied the provided pre-commit.sample file to a file named pre-commit. Next we opened the pre-commit file and added the following command to find the security weaknesses: 'bandit -r TestOrchestrator4ML-main -f csv -o security_weaknesses.csv'. We used the 'r' flag to recursively search the TestOrchestrator4ML directory, the 'f' flag to format the output as csv, and the 'o' flag to specify the output file. This process was similar to that of workshop 4, but instead of running cppcheck we used bandit - the tool we learned about in workshop 1. To use this hook, simply clone our repository, copy the pre-commit file to the .git/hooks directory, and make a commit. An example of this process can be found in Figure 1 and the resulting output csv contents can be found in Table 1.

2 Part 4b – Fuzz.py

The second portion of this assignment was to create a fuzzer that would fuzz five methods from the TestOrchestrator4ML repository and automatically report the results through GitHub actions. The methods we chose were 'checkMetricNames' from the 'py_parser' file in the 'detection' directory, the 'checkAlgoNames' method from the 'py_parser' file in the 'generation' directory, the 'generateAttack' method from the 'main' file in the 'generation' directory, the 'runs' method from the 'cliffsDelta' file in the 'label_perturbation_attack' directory, and the 'euc_dist' method from the 'knn' file in the 'label_perturbation_attack' directory. For this task, we used our knowledge of fuzzing from workshop 5 and began by compiling various inputs that we thought might produce interesting results when run on the methods. For example, empty lists and dictionaries, strings, integers, floating point numbers, and 'None' type values were among the chosen inputs. We then created a method for each of the functions we would be fuzzing and initialized an assortment of our input values in each method. These values were then tested on our chosen functions by using a for-loop containing a try-except block. The method would use the for-loop to iterate through our list of input values and try running them on the function being fuzzed. If the input was handled without error, then the execution remained in the try section of the code and would report the input as having passed the fuzzer. Otherwise, if the input created an error within the method, the code would jump to the exception portion of the code and report that the input had caused an error. Once all five of our chosen methods had been through this process, our program would indicate that it was finished and conclude its execution. A portion of this output can be viewed in Figure 2 and the full output is available in Appendix A.

The next step for this part of the project was to create a GitHub action that would run our ‘fuzz.py’ program automatically. We did this by going to the ‘Actions’ tab on GitHub and clicking the option to create a new workflow. This initialized our ‘main.yml’ file which is contained in the .github/workflows directory. Once the file was created, we decided that we wanted the primary method of triggering our GitHub action to be when something was pushed. Next, we made sure that all of the necessary dependencies to execute the fuzzer were run in the action. This mostly included using pip3 to install various libraries used in the project. Finally, we specified the working directory as TestOrchestrator4ML and used the command ‘python3 fuzz.py’ to execute the fuzzer. An example of a successful GitHub action using this method is viewable in Figure 3.

3 Part 4c – Forensics

The final task of this project was to integrate forensics into the repository by modifying five methods. We used the code and knowledge we got from completing workshop 9 to log our methods. First we created a file named ‘forensic_logging.py’ and created a getLoggerObj method within it. This method used the logging python library and would be called within the five methods to log various attributes. The logging level we set this method to was ‘INFO’. Next we chose five methods within the ‘generation’ directory. The first two methods were ‘getImport’ and ‘getFunctionAssignments’ from the ‘py_parser’ file. The third method was ‘calculate_k’ in the ‘attack_model’ file. The fourth method was ‘generate_malicious_instance’ from the ‘probability_based_label_perturbation’ file. Lastly, the fifth method was ‘random_label_perturbation’ from the ‘random_label_perturbation’ file. Figure 4 demonstrates the two primary lines of code we used in each of the methods to integrate forensics into the project.

4 Review

While most of the activities we completed for the project were quite familiar to us from the workshops we did throughout the semester, we were still able to learn a few things during this process. The most challenging thing we found was the GitHub action. While we had a little experience with them already, becoming familiar with the mechanics of the workflow in order to get our fuzzer to run automatically took a bit of trial and error. To view our repository, simply visit the following link: https://github.com/merrymitch/Foster_Mitchell-SQA2022-AUBURN.git. Overall, this project helped to reinforce the techniques we have learned during the duration of this course.

Appendix A: Full Output of fuzz.py

```
*** Begin Fuzzer ***
Start fuzzing checkMetricNames...
checkMetricNames failed with input: 0
checkMetricNames failed with input: None
checkMetricNames passed with inputRANDOMSTRING
checkMetricNames passed with input{'RANDOM': 'DICTIONARY'}
checkMetricNames passed with input{0, 10.3, 5, 'RANDOMSTRING'}
checkMetricNames failed with input: 3.14159
checkMetricNames passed with input{None}
checkMetricNames passed with input[None]
checkMetricNames passed with input{0}
checkMetricNames passed with input[0]
checkMetricNames passed with inputb'0000'
checkMetricNames failed with input: 0
checkMetricNames passed with input{}
checkMetricNames passed with input[]
checkMetricNames passed with input"`'><script>\xE2\x80\xA8javascript:alert(1)</script>'
checkMetricNames passed with input
checkMetricNames passed with input[[]]
Finish fuzzing checkMetricNames...
Start fuzzing checkAlgoNames...
checkAlgoNames failed with input: 0
checkAlgoNames failed with input: None
checkAlgoNames passed with inputRANDOMSTRING
checkAlgoNames passed with input{'RANDOM': 'DICTIONARY'}
checkAlgoNames passed with input{0, 10.3, 5, 'RANDOMSTRING'}
checkAlgoNames failed with input: 3.14159
checkAlgoNames passed with input{None}
checkAlgoNames passed with input[None]
checkAlgoNames passed with input{0}
checkAlgoNames passed with input[0]
checkAlgoNames passed with inputb'0000'
checkAlgoNames failed with input: 0
checkAlgoNames passed with input{}
checkAlgoNames passed with input[]
checkAlgoNames passed with input"`'><script>\xE2\x80\xA8javascript:alert(1)</script>'
checkAlgoNames passed with input
checkAlgoNames passed with input[[]]
Finish fuzzing checkAlgoNames...
Start fuzzing generateAttack...
generateAttack failed with inputs:
/home/runner/work/Foster_Mitchell-SQA2022-AUBURN/Foster_Mitchell-SQA2022-AUBURN/TestOrchestrator4ML
-main and 3.14159
generateAttack failed with inputs:
/home/runner/work/Foster_Mitchell-SQA2022-AUBURN/Foster_Mitchell-SQA2022-AUBURN/TestOrchestrator4ML
-main and 0
generateAttack failed with inputs:
/home/runner/work/Foster_Mitchell-SQA2022-AUBURN/Foster_Mitchell-SQA2022-AUBURN/TestOrchestrator4ML
-main and 22
```

```
generateAttack failed with inputs:
/home/runner/work/Foster_Mitchell-SQA2022-AUBURN/Foster_Mitchell-SQA2022-AUBURN/TestOrchestrator4ML-main and None
generateAttack failed with inputs:
/home/runner/work/Foster_Mitchell-SQA2022-AUBURN/Foster_Mitchell-SQA2022-AUBURN/TestOrchestrator4ML-main and RANDOMSTRING
generateAttack failed with inputs:
/home/runner/work/Foster_Mitchell-SQA2022-AUBURN/Foster_Mitchell-SQA2022-AUBURN/TestOrchestrator4ML-main and -8.9
generateAttack failed with inputs:
/home/runner/work/Foster_Mitchell-SQA2022-AUBURN/Foster_Mitchell-SQA2022-AUBURN/TestOrchestrator4ML-main and {}
generateAttack failed with inputs:
/home/runner/work/Foster_Mitchell-SQA2022-AUBURN/Foster_Mitchell-SQA2022-AUBURN/TestOrchestrator4ML-main and []
generateAttack passed with inputs: THIS/IS/WHAT/A/PATH/LOOKS/LIKE and 3.14159
generateAttack passed with inputs: THIS/IS/WHAT/A/PATH/LOOKS/LIKE and 0
generateAttack passed with inputs: THIS/IS/WHAT/A/PATH/LOOKS/LIKE and 22
generateAttack passed with inputs: THIS/IS/WHAT/A/PATH/LOOKS/LIKE and None
generateAttack passed with inputs: THIS/IS/WHAT/A/PATH/LOOKS/LIKE and RANDOMSTRING
generateAttack passed with inputs: THIS/IS/WHAT/A/PATH/LOOKS/LIKE and -8.9
generateAttack passed with inputs: THIS/IS/WHAT/A/PATH/LOOKS/LIKE and {}
generateAttack passed with inputs: THIS/IS/WHAT/A/PATH/LOOKS/LIKE and []
Finish fuzzing generateAttack...
Start fuzzing runs...
runs passed with input0
runs passed with inputNone
runs passed with inputRANDOMSTRING
runs passed with input{'RANDOM': 'DICTIONARY'}
runs passed with input{0, 10.3, 5, 'RANDOMSTRING'}
runs passed with input3.14159
runs passed with input{None}
runs passed with input[None]
runs passed with input{0}
runs passed with input[0]
runs passed with inputb'0000'
runs passed with input0
runs passed with input{}
runs passed with input[]
runs passed with input"`'><script>\xE2\x80\xA8javascript:alert(1)</script>'
runs passed with input
runs passed with input[[]]
Finish fuzzing runs...
Start fuzzing euc_dist...
euc_dist passed with inputs: 0 and 0
euc_dist passed with inputs: 0 and 2
euc_dist passed with inputs: 0 and 1
euc_dist passed with inputs: 0 and -4
euc_dist failed with inputs: 0 and None
euc_dist passed with inputs: 0 and 43
euc_dist failed with inputs: 0 and RANDOMSTRING2
euc_dist failed with inputs: 0 and {None}
euc_dist failed with inputs: 0 and [0]
euc_dist passed with inputs: 0 and 1e-06
euc_dist passed with inputs: 1 and 0
euc_dist passed with inputs: 1 and 2
euc_dist passed with inputs: 1 and 1
euc_dist passed with inputs: 1 and -4
euc_dist failed with inputs: 1 and None
euc_dist passed with inputs: 1 and 43
euc_dist failed with inputs: 1 and RANDOMSTRING2
euc_dist failed with inputs: 1 and {None}
euc_dist failed with inputs: 1 and [0]
euc_dist passed with inputs: 1 and 1e-06
euc_dist passed with inputs: 2 and 0
euc_dist passed with inputs: 2 and 2
euc_dist passed with inputs: 2 and 1
euc_dist passed with inputs: 2 and -4
euc_dist failed with inputs: 2 and None
euc_dist passed with inputs: 2 and 43
euc_dist failed with inputs: 2 and RANDOMSTRING2
euc_dist failed with inputs: 2 and {None}
euc_dist failed with inputs: 2 and [0]
euc_dist passed with inputs: 2 and 1e-06
euc_dist passed with inputs: -1 and 0
euc_dist passed with inputs: -1 and 2
euc_dist passed with inputs: -1 and 1
euc_dist passed with inputs: -1 and -4
euc_dist failed with inputs: -1 and None
euc_dist passed with inputs: -1 and 43
euc_dist failed with inputs: -1 and RANDOMSTRING2
euc_dist failed with inputs: -1 and {None}
euc_dist failed with inputs: -1 and [0]
euc_dist passed with inputs: -1 and 1e-06
euc_dist failed with inputs: None and 0
euc_dist failed with inputs: None and 2
euc_dist failed with inputs: None and 1
euc_dist failed with inputs: None and -4
euc_dist failed with inputs: None and None
euc_dist failed with inputs: None and 43
euc_dist failed with inputs: None and RANDOMSTRING2
euc_dist failed with inputs: None and {None}
euc_dist failed with inputs: None and [0]
euc_dist failed with inputs: None and 1e-06
euc_dist passed with inputs: 3.14159 and 0
euc_dist passed with inputs: 3.14159 and 2
euc_dist passed with inputs: 3.14159 and 1
euc_dist passed with inputs: 3.14159 and -4
euc_dist failed with inputs: 3.14159 and None
euc_dist passed with inputs: 3.14159 and 43
euc_dist failed with inputs: 3.14159 and RANDOMSTRING2
euc_dist failed with inputs: 3.14159 and {None}
euc_dist failed with inputs: 3.14159 and [0]
euc_dist passed with inputs: 3.14159 and 1e-06
euc_dist failed with inputs: RANDOMSTRING1 and 0
euc_dist failed with inputs: RANDOMSTRING1 and 2
euc_dist failed with inputs: RANDOMSTRING1 and 1
euc_dist failed with inputs: RANDOMSTRING1 and -4
euc_dist failed with inputs: RANDOMSTRING1 and None
euc_dist failed with inputs: RANDOMSTRING1 and 43
euc_dist failed with inputs: RANDOMSTRING1 and RANDOMSTRING2
euc_dist failed with inputs: RANDOMSTRING1 and {None}
euc_dist failed with inputs: RANDOMSTRING1 and [0]
euc_dist failed with inputs: RANDOMSTRING1 and 1e-06
euc_dist failed with inputs: {'RANDOM': 'DICTIONARY'} and 0
euc_dist failed with inputs: {'RANDOM': 'DICTIONARY'} and 2
euc_dist failed with inputs: {'RANDOM': 'DICTIONARY'} and 1
euc_dist failed with inputs: {'RANDOM': 'DICTIONARY'} and -4
euc_dist failed with inputs: {'RANDOM': 'DICTIONARY'} and None
euc_dist failed with inputs: {'RANDOM': 'DICTIONARY'} and 43
euc_dist failed with inputs: {'RANDOM': 'DICTIONARY'} and RANDOMSTRING2
euc_dist failed with inputs: {'RANDOM': 'DICTIONARY'} and {None}
euc_dist failed with inputs: {'RANDOM': 'DICTIONARY'} and [0]
euc_dist failed with inputs: {'RANDOM': 'DICTIONARY'} and 1e-06
euc_dist failed with inputs: [] and 0
euc_dist failed with inputs: [] and 2
euc_dist failed with inputs: [] and 1
euc_dist failed with inputs: [] and -4
euc_dist failed with inputs: [] and None
euc_dist failed with inputs: [] and 43
euc_dist failed with inputs: [] and RANDOMSTRING2
euc_dist failed with inputs: [] and {None}
euc_dist failed with inputs: [] and [0]
euc_dist failed with inputs: [] and 1e-06
euc_dist failed with inputs: {} and 0
euc_dist failed with inputs: {} and 2
euc_dist failed with inputs: {} and 1
euc_dist failed with inputs: {} and -4
euc_dist failed with inputs: {} and None
euc_dist failed with inputs: {} and 43
euc_dist failed with inputs: {} and RANDOMSTRING2
euc_dist failed with inputs: {} and {None}
euc_dist failed with inputs: {} and [0]
euc_dist failed with inputs: {} and 1e-06
Finish fuzzing euc_dist...
*** End Fuzzer ***
```