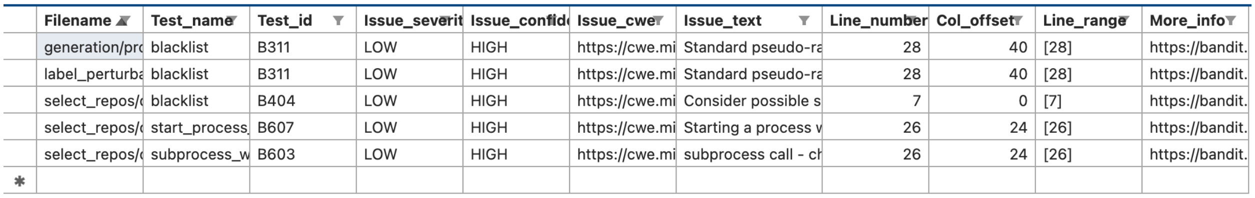
COMP 5710 - SQA Project

Lindsey Rafalsky, Hunter Westerlund, Tiffany Wu

Git Hook

We created a git hook that runs each time a python file is committed to the repository. The CSV output is shown below:



	Filename	Test_name	Test_id	Issue_severit	Issue_confid	Issue_cwe	Issue_text	Line_number	Col_offset	Line_range	More_info
	generation/pr	blacklist	B311	LOW	HIGH	https://cwe.mi	Standard pseudo-r	28	40	[28]	https://bandit.
	label_perturb	blacklist	B311	LOW	HIGH	https://cwe.mi	Standard pseudo-r	28	40	[28]	https://bandit.
	select_repos/	blacklist	B404	LOW	HIGH	https://cwe.mi	Consider possible s	7	0	[7]	https://bandit.
	select_repos/	start_process_	B607	LOW	HIGH	https://cwe.mi	Starting a process	26	24	[26]	https://bandit.
	select_repos/	subprocess_w	B603	LOW	HIGH	https://cwe.mi	subprocess call - c	26	24	[26]	https://bandit.
*											

We added this pre-commit file to the repository as well in the main folder since it would not be available on Github.

Fuzzing

We created a fuzz.py file that will automatically fuzz 5 Python methods:

1. checkTestFile from detect_test.
2. calculate_k from generation attack_model.
3. perform_inference from generation attack_model.
4. label_flip_perturbation from generation loss_based_label_perturbation.
5. generate_malicious_instace from generation probability_based_label_perturbation.

We reported bugs discovered by the fuzz.py file. These bugs included TypeError, ValueError, and AttributeError. The bugs are reported in the file named fuzzing_bugs.txt.

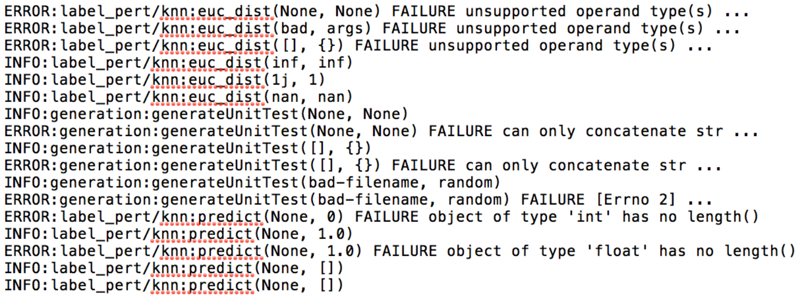
We executed fuzz.py from GitHub actions. Example below:

```
Fuzz: generate_malicious_instance FAIL
File "/home/runner/work/TEAM2000-SQA2022-AUBURN/TEAM2000-
SQA2022-AUBURN/fuzz.py", line 11, in fuzz
result = method(*arguments)
File "/home/runner/work/TEAM2000-SQA2022-AUBURN/TEAM2000-
SQA2022-AUBURN/detect_test.py", line 31, in checkTestFile
if(not (repo in repo_test_dict)):
UnboundLocalError: local variable 'repo' referenced before
assignment
```

```
Traceback (most recent call last):
File "/home/runner/work/TEAM2000-SQA2022-AUBURN/TEAM2000-SQA2022-AUBURN/fuzz.py", line 11, in fuzz
result = method(*arguments)
File "/home/runner/work/TEAM2000-SQA2022-AUBURN/TEAM2000-SQA2022-AUBURN/generation/attack_model.py", line 26, in calculate_k
model.fit(X_train, y_train)
File "/opt/hostedtoolcache/Python/3.10.8/x64/lib/python3.10/site-packages/sklearn/neighbors/_classification.py", line 207, in fit
return self._fit(X, y)
File "/opt/hostedtoolcache/Python/3.10.8/x64/lib/python3.10/site-packages/sklearn/neighbors/_base.py", line 407, in _fit
X, y = self._validate_data(
File "/opt/hostedtoolcache/Python/3.10.8/x64/lib/python3.10/site-packages/sklearn/base.py", line 563, in _validate_data
raise ValueError(
ValueError: This KNeighborsClassifier estimator requires y to be passed, but the target y is None.
```

Forensics

We did forensics on the following methods: euc_dist, generateUnitTest, predict, call_prob, and call_loss. Here is a log of the forensics.



```
ERROR:label_pert/knn:euc_dist(None, None) FAILURE unsupported operand type(s) ...
ERROR:label_pert/knn:euc_dist(bad, args) FAILURE unsupported operand type(s) ...
ERROR:label_pert/knn:euc_dist([], {}) FAILURE unsupported operand type(s) ...
INFO:label_pert/knn:euc_dist(inf, inf)
INFO:label_pert/knn:euc_dist(1j, 1)
INFO:label_pert/knn:euc_dist(nan, nan)
INFO:generation:generateUnitTest(None, None)
ERROR:generation:generateUnitTest(None, None) FAILURE can only concatenate str ...
INFO:generation:generateUnitTest([], {})
ERROR:generation:generateUnitTest([], {}) FAILURE can only concatenate str ...
INFO:generation:generateUnitTest(bad-filename, random)
ERROR:generation:generateUnitTest(bad-filename, random) FAILURE [Errno 2] ...
ERROR:label_pert/knn:predict(None, 0) FAILURE object of type 'int' has no length()
INFO:label_pert/knn:predict(None, 1.0)
ERROR:label_pert/knn:predict(None, 1.0) FAILURE object of type 'float' has no length()
INFO:label_pert/knn:predict(None, [])
INFO:label_pert/knn:predict(None, [])
```

Lessons Learned

We learned more about the ways that git hooks and git in general can be used to help streamline the software process and help mitigate vulnerabilities and bugs in the code. We also learned more about Github actions and the ways that it can help with ensuring that each commit to the repository can be analyzed for issues.