

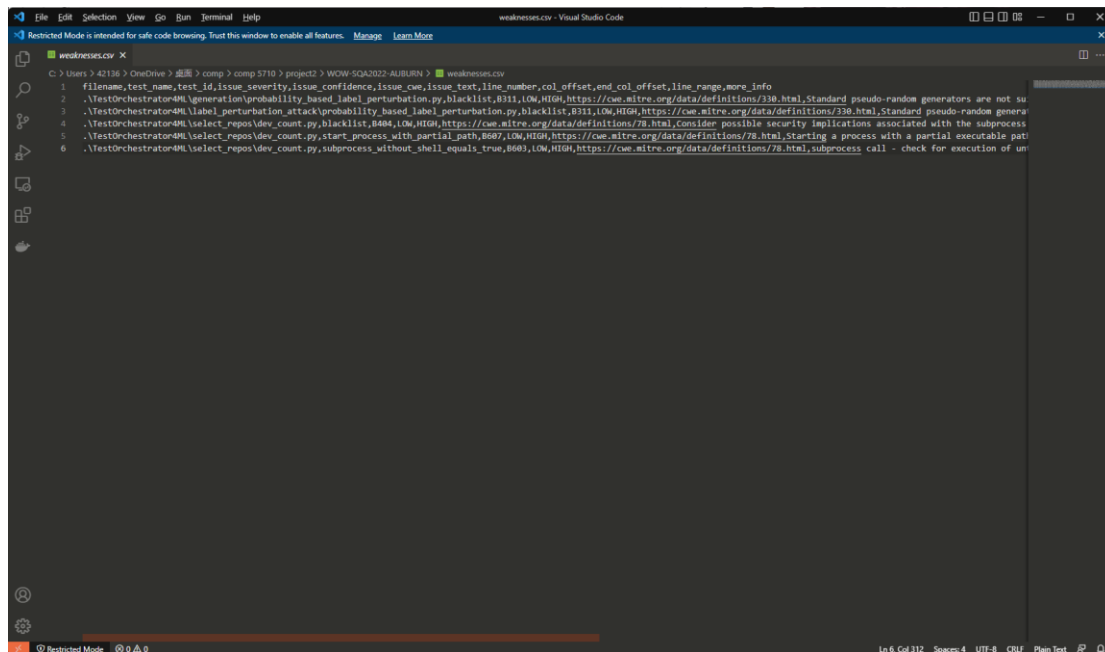
# Project Report for Team WOW

**Team name:** WOW

**Team member:** Jingming Chen

## Activities 4A:

To finish this part and detect the **security weaknesses**, I rename the pre-commit hook file and set it in the root folder. To make it been seen, I copy it and paste it outside. Also write a bandit command for writing a csv file record what it found.



```
1 filename,test_name,test_id,issue_severity,issue_cwe,issue_text,line_number,col_offset,end_col_offset,line_range,more_info
2 .\TestOrchestrator4ML\generation(probability_based_label_perturbation.py,blacklist,8311,LOW,HIGH,https://cwe.mitre.org/data/definitions/338.html,Standard pseudo-random generators are not su
3 .\TestOrchestrator4ML\label_perturbation_attack(probability_based_label_perturbation.py,blacklist,8311,LOW,HIGH,https://cwe.mitre.org/data/definitions/338.html,Standard pseudo-random genera
4 .\TestOrchestrator4ML\select_repos\dev_count.py,blacklist,8484,LOW,HIGH,https://cwe.mitre.org/data/definitions/78.html,Consider possible security implications associated with the subprocess
5 .\TestOrchestrator4ML\select_repos\dev_count.py,start_process_with_partial_path,8687,LOW,HIGH,https://cwe.mitre.org/data/definitions/78.html,Starting a process with a partial executable pat
6 .\TestOrchestrator4ML\select_repos\dev_count.py,subprocess_without_shell_equals_true,8683,LOW,HIGH,https://cwe.mitre.org/data/definitions/78.html,subprocess call - check for execution of un
```

## Activities 4B:

This step lets us use **fuzzing** to detect any 5 methods we chose. So, I choose

**generateAttack**

**predict**

**calculate\_k**

**calculate\_metrics**

**generate\_malicious\_instance**

these five methods. During my code, I use both fit and bad data to check it. Also, I create a workflow file to make sure it can be executed in the python action automatically.



The screenshot shows a GitHub repository page for `QEricQ/WOW-SQA2022-AUBURN`. The file `forensics.TEST.LOG` is selected, showing its content. The file is 57 lines long (57 sloc) and 3.67 KB. The content is a log file with timestamps and debug messages. The messages are as follows:

```
1 01-Dec-22 21:32:35:sqa-logger:DEBUG:knn.py*predict
2 01-Dec-22 21:32:35:sqa-logger:DEBUG:knn.py*predict
3 01-Dec-22 21:32:35:sqa-logger:DEBUG:knn.py*predict
4 01-Dec-22 21:32:35:sqa-logger:DEBUG:knn.py*predict
5 01-Dec-22 21:32:35:sqa-logger:DEBUG:knn.py*predict
6 01-Dec-22 21:38:17:sqa-logger:DEBUG:main.py*generateAttack
7 01-Dec-22 21:38:17:sqa-logger:DEBUG:main.py*generateAttack
8 01-Dec-22 21:38:17:sqa-logger:DEBUG:main.py*generateAttack
9 01-Dec-22 21:38:17:sqa-logger:DEBUG:main.py*generateAttack
10 01-Dec-22 21:38:17:sqa-logger:DEBUG:main.py*generateAttack
11 01-Dec-22 21:38:17:sqa-logger:DEBUG:knn.py*predict
12 01-Dec-22 21:38:17:sqa-logger:DEBUG:knn.py*predict
13 01-Dec-22 21:38:17:sqa-logger:DEBUG:knn.py*predict
14 01-Dec-22 21:38:17:sqa-logger:DEBUG:knn.py*predict
15 01-Dec-22 21:38:17:sqa-logger:DEBUG:knn.py*predict
16 01-Dec-22 21:38:17:sqa-logger:DEBUG:knn.py*calculate_k
17 01-Dec-22 21:38:17:sqa-logger:DEBUG:knn.py*calculate_k
18 01-Dec-22 21:38:17:sqa-logger:DEBUG:knn.py*calculate_k
19 01-Dec-22 21:38:17:sqa-logger:DEBUG:knn.py*calculate_k
20 01-Dec-22 21:38:17:sqa-logger:DEBUG:knn.py*calculate_k
21 01-Dec-22 21:38:17:sqa-logger:DEBUG:knn.py*calculate_metrics
22 01-Dec-22 21:38:17:sqa-logger:DEBUG:knn.py*calculate_metrics
23 01-Dec-22 21:38:17:sqa-logger:DEBUG:knn.py*calculate_metrics
24 01-Dec-22 21:38:17:sqa-logger:DEBUG:knn.py*calculate_metrics
25 01-Dec-22 21:38:17:sqa-logger:DEBUG:knn.py*calculate_metrics
26 01-Dec-22 21:38:17:sqa-logger:DEBUG:probability_based_label_perturbation.py*generate_malicious_instance
```

## Knowledge learned and Summarize:

For what I learn in these lessons, it really a lot. I love what I learned about log and csv, learning ways that create and record the output of our code. Then, both fuzzing and security weaknesses help me digging into the code I wrote, to detect the deficiency, it will be really helpful in my future studying or working. The Forensics and logging are cool, it helps us to check the method we choose to put it in, detect is there any error during runtime/testing. I still not familiar with this module. Because we just use it during workshop9 that I can use the code in it help us finish it. I will keep learning it,