AUBURN

UNIVERSITY

Software Quality Assurance: Project Report

Jeremy Ellison, Sophia Yuan, Ryan Harvey, Matthew Pichler

May 1, 2023

Comp 5710 Software Quality Assurance Team X

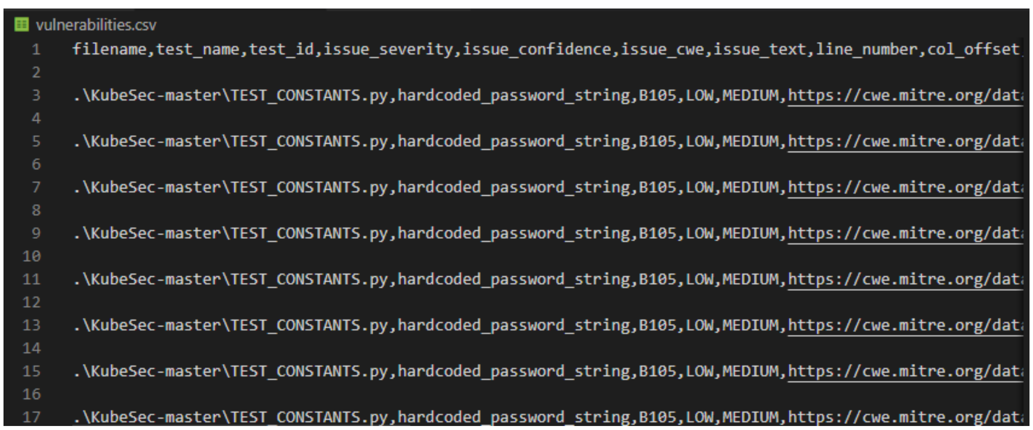
Department of Computer Science and Software Engineering

Samuel Ginn College of Engineering, Auburn University

Activities

First as a group, we met and assigned some preliminary tasks(1-5) to each of the group members. This was an effective way to start the project and gave each person something to tackle. We also assigned each group member a method to fuzz. After each group member had finished or made some progress we got back together and tackled the remaining problems as a team. We faced some issues pushing to our repository but generally we kept a good account of the versions of our code and progress.

GitHook



```
vulnerabilities.csv
filename,test_name,test_id,issue_severity,issue_confidence,issue_cwe,issue_text,line_number,col_offset

.\KubeSec-master\TEST_CONSTANTS.py,hardcoded_password_string,B105,LOW,MEDIUM,https://cwe.mitre.org/data

.\KubeSec-master\TEST_CONSTANTS.py,hardcoded_password_string,B105,LOW,MEDIUM,https://cwe.mitre.org/data

.\KubeSec-master\TEST_CONSTANTS.py,hardcoded_password_string,B105,LOW,MEDIUM,https://cwe.mitre.org/data

.\KubeSec-master\TEST_CONSTANTS.py,hardcoded_password_string,B105,LOW,MEDIUM,https://cwe.mitre.org/data

.\KubeSec-master\TEST_CONSTANTS.py,hardcoded_password_string,B105,LOW,MEDIUM,https://cwe.mitre.org/data

.\KubeSec-master\TEST_CONSTANTS.py,hardcoded_password_string,B105,LOW,MEDIUM,https://cwe.mitre.org/data

.\KubeSec-master\TEST_CONSTANTS.py,hardcoded_password_string,B105,LOW,MEDIUM,https://cwe.mitre.org/data

.\KubeSec-master\TEST_CONSTANTS.py,hardcoded_password_string,B105,LOW,MEDIUM,https://cwe.mitre.org/data
```

Fuzzing

```
$ python fuzz.py
FUZZ: isValidUserName FAILED :()
Traceback (most recent call last):
  File "C:\Users\SopY0\Desktop\SQAProject\KubeSec-master\fuzz.py", line 14, in fuzz
    result = method(*args)
             ^^^^^^^^^^^^^
TypeError: scanner.isValidUserName() argument after * must be an iterable, not NoneType
FUZZ: scanKeys FAILED :()
Traceback (most recent call last):
  File "C:\Users\SopY0\Desktop\SQAProject\KubeSec-master\fuzz.py", line 14, in fuzz
    result = method(*args)
             ^^^^^^^^^^^^^
TypeError: scanner.scanKeys() argument after * must be an iterable, not bool
FUZZ: isValidPasswordName FAILED :()
Traceback (most recent call last):
  File "C:\Users\SopY0\Desktop\SQAProject\KubeSec-master\fuzz.py", line 14, in fuzz
    result = method(*args)
             ^^^^^^^^^^^^^
TypeError: scanner.isValidPasswordName() argument after * must be an iterable, not bool
FUZZ: scanForSecrets FAILED :()
Traceback (most recent call last):
  File "C:\Users\SopY0\Desktop\SQAProject\KubeSec-master\fuzz.py", line 14, in fuzz
    result = method(*args)
             ^^^^^^^^^^^^^
TypeError: scanForSecrets() missing 1 required positional argument: 'yaml_d'
FUZZ: isValidKey FAILED :()
Traceback (most recent call last):
  File "C:\Users\SopY0\Desktop\SQAProject\KubeSec-master\fuzz.py", line 14, in fuzz
```

Forensic

```
8639    2023-04-30 20:45:20,117 Check valid user name
8640    2023-04-30 20:45:20,117 Check valid input for isValidPassword
8641    2023-04-30 20:45:20,117 Check scannned Keys
8642    2023-04-30 20:45:20,117 Check valid input for key
8643    2023-04-30 20:45:20,117 Check scan for secrets
```

Lessons Learned

Lessons learned mainly included familiarizing ourselves with Git. Setting up and becoming familiar with the Git architecture and our repository was essential for good development. While implementing a bandit git hook, we found that git will not push without first committing using the '—no-verify' tag. This allows the new code in .git/hooks/pre-commit to be pushed without being run. Committing with the hook locally is still possible and gives the correct scan output. Understanding the intent of some of the methods we fuzzed was also

important. Correctly choosing fuzzing inputs saved some time that might have been used entering in obviously correct inputs.