Assignment 4 + Bonus Project

Yulei Sui

University of Technology Sydney, Australia

Assignment 4: Quiz + A Coding Task

- A quiz (10 points)
  - Taint Analysis
  - C++ file writing and reading

Assignment 4: Quiz + A Coding Task

- A quiz (10 points)
  - Taint Analysis
  - C++ file writing and reading
- One coding task (20 points)
  - Implement method `readSrcSnkFromFile` in `Assignment-4.cpp` using C++ file reading to configure sources and sinks.
  - Implement method `printICFGPath` to collect the tainted ICFG paths and add each path (a sequence of node IDs) as a string into `std::set<std::string> paths` similar to Assignment 2
  - Implement method `aliasCheck` to check aliases of the variables at source and sink.

Assignment 4: Coding Task

- Code template and specification: `https://github.com/SVF-tools/Teaching-Software-Analysis/wiki/Assignment-4`
- Make sure your previous implementations in `Assignment-2.cpp` and `Assignment-3.cpp` are in place.
  - Class `TaintGraphTraversal` in Assignment 4 is a **child class** of ‘ICFGTraversal‘. `TaintGraphTraversal` will use the `DFS` method implemented in Assignment 2 for **control-flow traversal**.
  - Andersen’s analysis implemented in Assignment 3 will also be used for **checking aliases** between two pointers.

A Bonus Task of This Subject

- One bonus project with 5 bonus points (The total marks of this subject will be capped at 100).
    - Dump the taint program paths into a text file
    - Implement a VSCode extension to annotate and visualize the tainted paths from a source to a sink.
- Please submit (1) **a demo video** in mp4 format (which is recommended) **or** (2) a **VSCode Extension** (.vsix fomrat) with proper documentations.

C++ File Reading

Implement method `readSrcSnkFormFile` in `Assignment-4.cpp` to parse the two lines from `SrcSnk.txt` in the form of

```
source -> { source src set getname update getchar tgetstr }
sink -> { sink mysql_query system require chmod broadcast }
```

Please refer to the following links (among many others) for C++ file reading:

- https://www.tutorialspoint.com/cplusplus/cpp_files_streams.htm
- https://www.cplusplus.com/doc/tutorial/files/
- https://linuxhint.com/cplusplus_read_write/
- https://opensource.com/article/21/3/ccc-input-output

Visualizing Tainted Paths (5-mark bonus task)

This task is optional and there is no uniform answer! Some hits as below but you are also encouraged to design and implement your own approach.

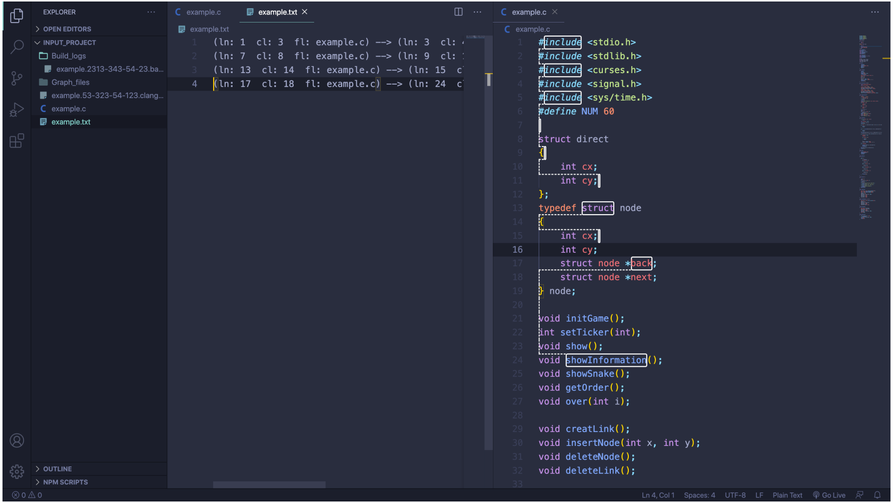
- Output taint paths into a text file in the following format for example, '{ ln: number cl: number, fl: name } → { ln: number, cl: number, fl: name } → { ln: number, cl: number, fl: name }'.
- Create a VSCode extension to read the text file
- Annotate the target source file (e.g., example.c) based on the taint paths reading from the text file.

Two VSCode extension examples (note that they are just general examples for references and are **NOT** the solution to the task):

`https://github.com/akshatsinghkaushik/vscode-extension-example`

`https://github.com/spcidealacm/codepointer_js`

VSCode Extension Demo (feel free to design yours)



```
(ln: 1  cl: 3  fl: example.c) --> (ln: 3  cl:
(ln: 7  cl: 8  fl: example.c) --> (ln: 9  cl:
(ln: 13  cl: 14  fl: example.c) --> (ln: 15  c
(ln: 17  cl: 18  fl: example.c) --> (ln: 24  c
```

```c
#include <stdio.h>
#include <stdlib.h>
#include <curses.h>
#include <signal.h>
#include <sys/time.h>
#define NUM 60

struct direct
{
    int cx;
    int cy;
};
typedef struct node
{
    int cx;
    int cy;
    struct node *back;
    struct node *next;
} node;

void initGame();
int setTicker(int);
void show();
void showInformation();
void showSnake();
void getOrder();
void over(int i);

void creatLink();
void insertNode(int x, int y);
void deleteNode();
void deleteLink();
```