Lab: Information Flow Tracking

(Week 3)

Yulei Sui

School of Computer Science and Engineering University of New South Wales, Australia

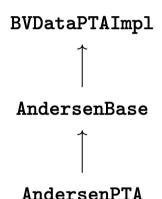
Assignment-1

- Assignment-1 (20 points)
 - readSrcSnkFromFile and reachability: Implement a context-sensitive graph traversal on a CodeGraph (i.e., ICFG) and collect feasible paths from a source node to a sink node on SVF's ICFG.
 - solveWorklist: Implement field-sensitive Andersen's inclusion-based constraint solving for points-to analysis on SVF's ConstraintGraph
 - aliasCheck: Implement taint analysis in class ICFGTraversal. Checking
 aliases of the two variables at source and sink. Two variables are aliases if their
 points-to sets have at least one overlapping element.

Assignment-1

- Assignment-1 (20 points)
 - readSrcSnkFromFile and reachability: Implement a context-sensitive graph traversal on a CodeGraph (i.e., ICFG) and collect feasible paths from a source node to a sink node on SVF's ICFG.
 - solveWorklist: Implement field-sensitive Andersen's inclusion-based constraint solving for points-to analysis on SVF's ConstraintGraph
 - aliasCheck: Implement taint analysis in class ICFGTraversal. Checking
 aliases of the two variables at source and sink. Two variables are aliases if their
 points-to sets have at least one overlapping element.
 - Specification and code template: https: //github.com/SVF-tools/Software-Security-Analysis/wiki/Assignment-1
 - SVF APIs for control- and data-flow analysis https: //github.com/SVF-tools/Software-Security-Analysis/wiki/SVF-CPP-API

Assignment Structure



 You will be working on AndersenPTA's solveWorklist method.

Assignment Structure

BVDataPTAImpl



AndersenBase



AndersenPTA

- You will be working on AndersenPTA'S solveWorklist method.
- Constraint graph is the field consCG.

Assignment Structure

BVDataPTAImpl AndersenBase AndersenPTA

- You will be working on AndersenPTA'S solveWorklist method.
- Constraint graph is the field consCG.
- More APIs about points-to operations and constraint graph are here:

```
https://github.com/SVF-tools/
Software-Security-Analysis/wiki/
SVF-CPP-API
```

APIs for Implementing Andersen's analysis

```
::getPts(NodeID ptr)
                                                                      //get points-to set of ptr
   SVF:: AndersenBase
                                  ::addPts(NodeID ptr, NodeID obi)
                                                                      // add obj to point-to set of object ptr
                                  ::unionPts(NodeTD ntr. NodeTD ntr)
                                                                      // union two point-to sets
                                  :: pushIntoWorklist(NodeID id)
                                                                     // push the node to worklist
                                  ::popFromWorklist()
                                                                      // pop a node from the worklist
                                  ::isInWorklist(NodeID id)
                                                                      // return true if the node in the worklist
                                  ::isWorklistEmptv()
                                                                      // return true if the worklist is empty
    SVF:: AndersenPTA
                                  ::addCopyEdge(NodeID src. NodeID dst) // add a copy edge from src to dst
                                  ::getConstraintNode(nodeId id)
                                                                    //get the node based on its id
SVF::ConstraintGraph
                                   :: dump()
                                                                    // dump the ConsG
                                                                 // get incoming store edges of the node
                                  ::getStoreInEdge()
                                                                 //get outgoing store edges of the node
                                   ::getStoreOutEdge()
SVF::ConstraintNode
                                   ::getDirectOutEdge()
                                                                 // get outgoing copy edges of the node
                                  ::getDirectInEdge()
                                                                 // get incoming copy edges of the node
```

```
https://github.com/SVF-tools/Software-Security-Analysis/wiki/SVF-CPP-API#worklist-operations
https://github.com/SVF-tools/Software-Security-Analysis/wiki/SVF-CPP-API#points-to-set-operations
https://github.com/SVF-tools/Software-Security-Analysis/wiki/SVF-CPP-API#alias-relations
https://github.com/SVF-tools/Software-Security-Analysis/wiki/SVF-CPP-API#constraintgraph-constraintnode-and-constraintedge
```

C++ File Reading

Implement method readSrcSnkFormFile in Assignment-1.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