SE4950 – Fall 2019
Project 4 – Static Analysis Lab 2 – C/C++ cppcheck/Clang/infer

Background

In the last project, we found Java issues using the SpotBugs static analyzer. In this project, we will examine issues in C/C++ code using three different static analyzers.

The first of these is cppcheck, which has a fairly long history as a open source static analyzer. To invoke it on a C/C++ source file (in this case example.c), you can simply issue a command line like the following:

```
cppcheck example.c
```

The second static analyzer we will use is the analyzer that has been developed as part of the Clang compiler development. (Clang is the C/C++ front end to the LLVM compiler backend). It watches for clang compiler invocations in a build process. So it can be used to analyze an entire project build with one command. To invoke it with the same sample file, displaying the results in a web browser, you would do the following command:

```
scan-build –V clang example.c
```

The Clang analyzer could also watch an entire build, so if you are in a directory with a Makefile, you could type:

```
scan-build –V make
```

Note that before you do this command however, you should first do a "clean" to get rid of anything that was previously compiled and built. You do this with:

```
make clean
```

The third analyzer we will use is one that was developed internally at Facebook, then released as open source called infer. Like the Clang analyzer, it can be invoked on entire builds. To use it on the same single example C source file, you would use the command:

```
infer run -- clang example.c
```

Likewise, to scan an entire build using a Makefile, you could execute the following command:

```
make clean
infer run -- make
```

Analysis Task

For each of the C source files in the /home/student/Ccode directory in the provided Virtual Machine, first look at the source and see if you can manually see any issues with the code. Then run each of the static analyzers discussed on the previous page on each of the source files. Note the errors that are detected by each of the analyzers in the CAnalysis.docx file that is attached to this project in Blackboard.

Repeat the process for the small C++ project in the /home/student/Ccode/sam directory, which uses a Makefile to build the project.

Grading

The project will be worth 25 points.