Static Analysis using splint

Author : Vikas Nagpal (https://github.com/vikasnagpaliitd)
Version 1.0

Background

Imagine the below code

*int *ptr;*
*printf("%d", *ptr);*

A simple code review shows we are accessing an invalid ptr (which has not been malloc'ed or pointed to a valid memory area).

Static Analysis tools, analyze the code in similar (but much more sophisticated) ways, and warn us of potential bugs in code.

They are called "static" analyzers, as they analyze the stationary (not running) code.This is in contrast with dynamic tools like valgrind, which run the executable and watch what is happening.

Installing splint

$ sudo apt install splint

Examples of caught errors

Some (not all) examples which splint can catch are given below

1. Variable used before definition
2. Test expression for if is assignment expression
3. Test expression type is not boolean or int. (Use -predboolint to inhibit warning)
4. Function exported but not used outside file
5. Stack-allocated storage local reachable from return value

6. many more…

Example code for demo

Demo code is available at https://github.com/vikasnagpaliitd/linux-prog-tools

References

1. Splint.org
2. Splint Manual
3. Splint (programming tool) - Wikipedia
4. Youtube video : https://youtu.be/941R_i9f5E8