Project in advanced programming - Outline

Edo Cohen	Tzafrir Rehan	Gai Shaked
039374814	039811880	036567055
sedoc@t2	tzafrir@cs	gai@tx

November 1, 2010

We intend to write a static Buffer Overrun Analyzer (boa). Boa receive a C program as an input, and determine whether a buffer overrun is possible during an execution of the code. Boa will be implemented in two stages, the first will be context and flow insensitive analysis, which will always warn about any possible buffer ovverun under if -

- no pointer manipulation regarding buffers

This relatively humble condition will ensure our solution is sound (i.e. boa will always warn if buffer overrun is feasible) but the context and flow insensitivity also prone to lots of false alarms. Thus we intend to test limited flow and context sensitivity, which we hope will be able to reduce the flase positives without letting any possible false negative result. For this secondphase we will also require that the input C code will not include -

- concurrency
- goto statements

The implementation will be based on Clang [1] as the static analysis front end. Clang API will be used to generate integer linear programming constraints for each buffer and integer in the code, constriants which will model the maximal and minimal used (and allocated) index each buffer. Finally we will use GLPK [2][3] to solve the integer linear programming problem inflected by the constriants, and the solution will determine whether buffer overrun is possible.

[1] http://clang.llvm.org/
[2] http://en.wikipedia.org/wiki/GNU_Linear_Programming_Kit
[3] http://www.gnu.org/software/glpk/