SYMBOLIC ANALYSIS FOR BUFFER OVERFLOW

Surinder Kumar Jain, Supervisor : Bernhard Scholz

September 4, 2009

School of Information Technology, University of Sydney

(suri@it.usyd.edu.au and scholz@it.usyd.edu.au)

Abstract

We do Symbolic Analysis by formulating an intermediate representation of a program with non-deterministic semantics. Using program analysis techniques, we enumerate all acyclic paths with in program loops. Semantics of Program loops are formulated as and solved as symbolic recurrence system for Closed form and for exit condition as loop invariants. A computer algebra system (CAS) is used to automate the solution of recurrence system. Acyclic paths in the program and in program loops are pruned by reasoning contradictions in path conditions of these acyclic paths.

We extend non-deterministic semantics developed for simple structured language to general control flow graphs, both reducible as well as irreducible. A demand driven technique is used to analyse only relevant slices of the program so that it can be performed efficeintly and without state explosion problem.

A prototype tool using this techniques is developed to determine program security vulnerability caused by buffer overflows. Our technique can also be used for other analysis like Worst case execution time, proving safety invariants, progress invariants for fair termination etc.