Selective Symbolic Execution

Vitaly Chipounov, Vlad Georgescu, etc. *EPFL*

Rate: ★★★★☆

September 12, 2013

This paper discuss the S2E method very clear in a relative higher level. In the paper, it discuss the symbolic execution as following: the symbolic engine will represent the programs memory in an engine-specific data structure and, when the execution reaches a branch whose condition involves symbolic values, the engine forks the program states, such as each path has its own private program state.

Challenges

One of the challenge with SE is that it has to deal with the programs interaction with the environment. This challenge also prevent SE from being efficiently executed.

S2E is a virtual execution platform that allow user to specify a scope of interest within a systems execution space, and then focuses CPU and memory resources on symbolically executing that scope only. This is a great idea to make SE to run in real system. By doing so, S2E pruning parts of the execution tree that the developer would not even look at once execution completed.

Use cases

The use case for this is broad, including (1) Testing in complex environment, (2) Fine-grain module testing, (3) Data-driven testing, (4) Hybrid-input testing, (5) Reproducing user-reported bugs, (6) Dynamic failure analysis, (7) Failure avoidance, (8) Reverse engineering programs.

For (1), Think about symbolic execute Firefox, When the input to FF is symbolic, FF also call system libs with symbolic, how to ensure that the return value to FF is vilad? This means that the symbolic to concrete conversion from FF to environment is one way.

For (2), Think about doing unit testing for one of FFs properties. For (3), Think about during code development, you want to change a data structure that has been extensively touched throughout the code, how could you ensure the modified data structure is works fine after the modification. For (4), This is similar to the (1), which is to ensure quickly reach some conner case. For (5), The metaphor in the paper is great, which says: The details of the bug report define an envelope of executions, and S2E searches for the culprit path only within the envelope. For (6) and (7), they are involving some special scenario, refer to the papepr