2020 IEEE Conference on Communications and Network Security (CNS)

SYMBION: Interleaving Symbolic with Concrete Execution

Fabio Gritti, Lorenzo Fontana, Eric Gustafson, Fabio Pagani, Andrea Continella,
Christopher Kruegel, and Giovanni Vigna

University of California, Santa Barbara
{degrigis, lfontana, edg, pagani, conand, chris, vigna}@cs.ucsb.edu

***Abstract*—Symbolic execution is a powerful technique for exploring programs and generating inputs that drive them into specific states. However, symbolic execution is also known to suffer from severe limitations, which prevent its application to real-world software. For example, symbolically executing programs requires modeling their interactions with the surrounding environment (e.g., libraries, operating systems). Unfortunately, models are usually created manually, introducing considerable approximations of the programs behaviors and significant imprecision in the analysis. In addition, as the complexity of the system under analysis grows, additional models are needed, making this process unsustainable. For these reasons, in this paper we propose**

routines. These routines, known as *models*, emulate the effects of a specific environment interaction (e.g., syscalls or calls to third-party libraries) on the program's (symbolic) state. This approximation represents another distinct problem for symbolic execution since models must be developed for every interaction the program has with the environment.

For these reasons, the task of symbolically executing a specific portion of an application's code can be quite challenging. In fact, symbolically executing the program from its entry point to the interesting part is often infeasible or