Finding JavaScript Vulnerabilities With Concolic Execution Using Switched MUS Reduction on Quivers

Colaresi, Landon (School: Pittsburgh Allderdice High School)

Concolic execution is a popular automatic vulnerability detection method that relies on dynamically enumerating all possible paths using an SMT solver. However, the most popular programming language today, JavaScript, is traditionally difficult to concolically execute as a result of event-based control flow. This project presents an extension to the traditional concolic execution algorithm that allows it to efficiently reason about event-based control flow, known as Switched MUS Reduction on Quivers (SiMReQ). A set of known transitions between events is dynamically built up, forming a data structure known as a quiver, and walks through the quiver that end in failure are continuously generated, forming potential failure modes for the program. However, because the quiver representation attempts to be sound, these walks are highly overconstrained. Thus, they are iteratively relaxed by generating Minimal Unsatisfiable Subsets (MUSes), which are a existing formalization of the conflicts in a particular unsatisfiable set of constraints. This creates a cycle in which concolic execution updates the quiver, which leads to new potential failure modes, which are relaxed to yield new inputs for concolic execution to evaluate. Due to the lack of a stable JavaScript concolic execution engine to work with and the complexity of creating one from scratch, two variants of the SiMReQ algorithm were tested against traditional concolic execution on synthetic, randomly-generated programs. On average, the simplest form of SiMReQ found ~32.4\% more bugs, and the more advanced (JIT DSE) variant found ~92.4\% more bugs.

Awards Won:
Third Award of $1,000
National Security Agency Research Directorate : First Place Award “Cybersecurity”