SYM-EXE && AEG

Rafael

SYMBOLIC EXECUTION

- CSPs (Constraint Solving Problem)
- Traditional Symbolic Execution
- Modern Symbolic Execution
- Challenges
- Related papers
  Still love [Symbolic execution for software testing: three decades later]

STORY

- 2012-2017, DoD found vulnerabilities in weapon system
- DoD started to develop tools for Binary analysis and Symbolic Execution
- In 2016, DARPRA hosted CyberGrandChallenge
  - Created automatic defense system
  - CMU ForAllSecure won first prize on CGC

SYMBOLIC EXECUTION

- Is a kind of static analysis techniques
- Replace actual value with **symbol** to simulate the execution
- Be used for software testing; be used for bug finding
- Get possible input for specific condition
  e.g. Given constraints for vulnerabilities, generate the possible input
- Working
  - execution tree consists of execution path
  - **symbol state**: initialized as empty mapping
  - **symbolic path constraint**: initialized as true
  - branches → add condition to path constraint -> create symbol execution instance with PC
    both branches can be run at the same time
  - Done: PC solved by constraint solver and generate concrete input
- Challenge: infeasible when constraint solver cannot work efficiently

MODERN SYMBOLIC EXECUTION

- Mix and Strike balance between Concrete and Symbolic Execution
- Concolic Testing (DFS)
  - Maintain concrete state and symbolic state
  - Cover all execution path with reversing constraint
  - Given random input
    -> run path constraint
    ->… done
    -> reverse constraint
    -> generat new input and run new path constraint
- Execution-Generated Testing (EGT)
  - Concolic Testing Optimization

CHALLENGES

- Path Explosion
- Constraint Solving
  - Irrelevant Constraint Elimination
  - Reuse previous results
- Memory Modeling
  - Precision is important
  - Too high, computation complexity would also be high
  - Too low too general, lose cases

AUTOMATIC EXPLOIT GENERATION

1. Bug-Find
   preconditioned symbolic execution
   path prioritization: Heuristics
2. Scenerio Analysis
   EIP status, memory alignment
3. Construction of Constraint Condition
4. Constraint Solving, Exploit generation