ICSE2023
45th International Conference on Software Engineering
Melbourne Convention and Exhibition Centre
14-20 May 2023

Boosting Symbolic Execution for Heap-based Vulnerability Detection and Exploit Generation

Haoxin TU

SMU
SINGAPORE MANAGEMENT UNIVERSITY

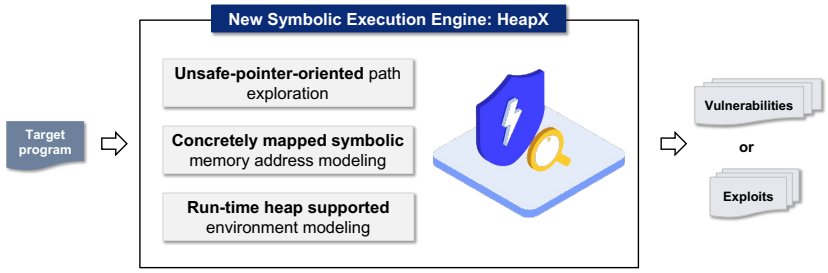
INTRODUCTION

- Detecting heap-based vulnerabilities (e.g., UAF) and demonstrating their severity via generating exploits for them are of critical importance.
- Symbolic execution-based approaches have shown their potential in the above tasks. However, they still have following fundamental limitations:
  - Path exploration (not vulnerability-oriented)
  - Memory modeling (concrete modeling of heap addresses)
  - Environment modeling (no native environment support for heap allocation)
- **Objective:** we aim to design and implement a boosted symbolic execution engine named HeapX to facilitate the automatic detection and exploitation of heap-based vulnerabilities.

SYSTEM DESIGN

- **Overview:** a new path exploration strategy, a new memory model, and a new environment modeling solution are expected to be designed in HeapX.

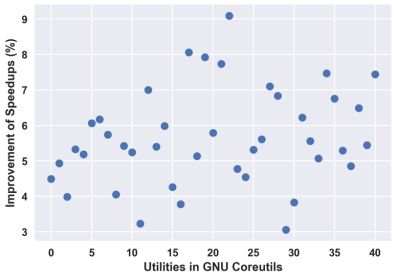
- **Key Insights**
  - ➢ Path searching towards the ones that are more likely to be vulnerable
  - ➢ Memory addresses from heap allocation are dynamically determined
  - ➢ Native heap address is an important requirement for exploit generation and verification



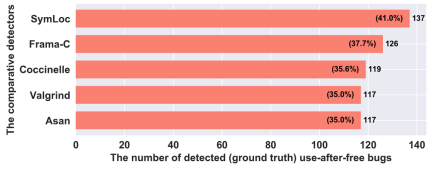
PRELIMINARY RESULTS

Evaluation Criteria
- ➢ Performance
- ➢ Code coverage
- ➢ The number of vulnerabilities
- ➢ The number of exploits

For sub-solution 1: FastKLEE [1]



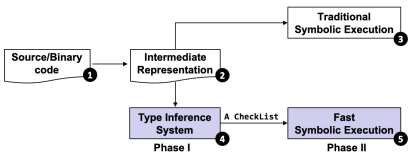
For sub-solution 2: SymLoc [2]



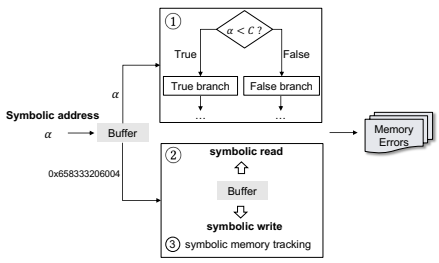
For sub-solution 3: Ongoing …

PROPOSED SOLUTIONS

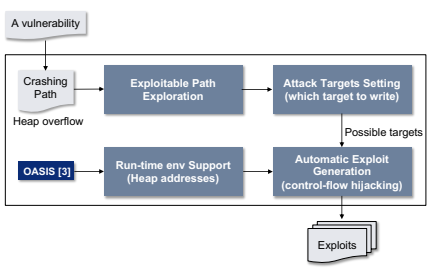
Sub-solution 1: FastKLEE [1]
- ➢ Reduce unnecessary bound-checks on safe pointers



Sub-solution 2: SymLoc [2]
- ➢ Symbolize heap memory addresses
- ➢ Support efficient symbolic read/write
- ➢ Track the uses of symbolic addresses



Sub-solution 3: HeapExp
- ➢ Explore exploitable paths
- ➢ Support native heap environment



FUTURE WORK

- ➢ Extend FastKLEE for unsafe-pointer-oriented path exploration
- ➢ Empirical study to learn existing exploit patterns for CVEs
- ➢ Design new algorithms for Automatic Exploit Generation
- ➢ Integration of all sub-solutions into one HeapX system

CONTRACT

Email: haoxintu.2020@phdcs.smu.edu.sg
GitHub: https://github.com/haoxintu
Twitter: @tuhaoxin

ACKNOWLEDGEMENT & REFERENCES

This research/project is supported by the National Research Foundation, Singapore and the National Satellite of Excellence in Trustworthy Software Systems (NSoE-TSS) award number NSOE-TSS2019-04.

[1] Haoxin Tu, Lingxiao Jiang, Xuhua Ding, and He Jiang. "FastKLEE: faster symbolic execution via reducing redundant bound checking of type-safe pointers." In ESEC/FSE, pp. 1741-1745. 2022.
[2] Haoxin Tu, Lingxiao Jiang, Jiaqi Hong, Xuhua Ding, and He Jiang, "Concretely Mapped Symbolic Memory Locations for Memory Error Detection" (Major Revision on TSE)
[3] Hong, Jiaqi, and Xuhua Ding. "A novel dynamic analysis infrastructure to instrument untrusted execution flow across user-kernel spaces." In IEEE S&P, pp. 1902-1918. IEEE, 2021.

SCHOOL OF COMPUTING AND INFORMATION SYSTEMS