Strings Enhanced Symbolic Execution

Treating Strings as ADTs in a KLEE/Z3 framework

May 9, 2018

Introduction

- str.klee is gradually becoming more stable.
- A multitude of artificial examples supports str.klee
- The following four (previously known) CVEs are discovered:

CVE	str.klee discovery time (minutes)	vanilla.klee discovery time (minutes)
dnstracer CVE-2017-9430	31	33
libtiff CVE-2006-2656	34	29
gzip CVE-2001-1228	30	–
mp3info CVE-2006-2465	56	–

Main Opportunities

- C to C transformations, like this example
- C to Z3 transformations, like these examples:
  - while (tmp[0] == 'A' || tmp[1] == 'B') {tmp++;}
  - translated to Z3
  - do { *p = 0; p++; i++; } while (i < len);
  - translated to Z3
- C to C under approximations of execution flow:
  - for (i = 0; i < len; i++)
    if ('A' ≤ p[i] ≤ 'Z') { p[i] = p[i] + 32; }
  - replacing with: nop+string constraints will help here
  - if (fiLong ≠ NULL) {
    fi = strrchr (fiLong, '/');
    if (fi == NULL) { fi = strrchr (fiLong, '\\'); }
    if (fi ≠ NULL) { fi++; }
    if (fi == NULL) { fi = fiLong; }}
  - Replacing with: fi=fiLong+string constraints will help.
- Improve reads/writes with caching / reduction of versions
- Context aware sorts in SE