

Využití symbolické exekuce pro testování real-time bezpečnostně kritického softwaru

Martin Hořeňovský
Vedoucí práce: Michal Sojka

FEL ČVUT

2015

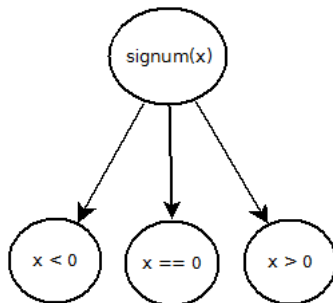
Cíl práce

- Prozkoumat možnost použití KLEE pro verifikaci bezpečnostně kritického, real-time softwaru
- Analýza eMotor softwaru pro řízení elektrických motorů
- Analýza MaCAN knihovny
- Návrh dalších vylepšení

- Dokumentace výsledků

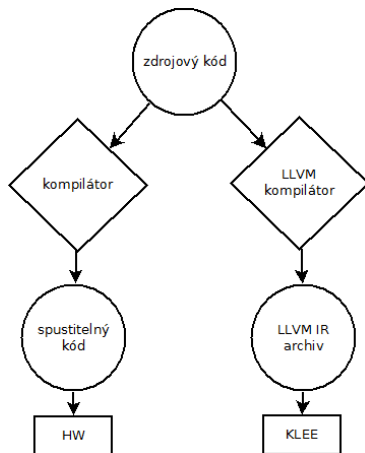
Symbolická exekuce

- Umožňuje prozkoumat všechny průchody programem
- Nepotřebuje programátorem dodané vstupy
- Umožňuje najít chyby které se projeví až za běhu
- Extrémně náročná na výpočetní zdroje
- Některé konstrukce v kódu jsou neřešitelné



KLEE

- KLEE vzniklo v roce 2008 jako nástroj pro automatické generování unit testů
- Umožňuje spustit podmnožinu jazyka C (vlákna, symbolické floaty, ASM nejsou podporovány)
- Pracuje na spustitelném programu převedeném do interní reprezentace LLVM nástrojů



MaCAN - Message Authenticated CAN

- CAN je standard pro industriální sítě
- Nemá v sobě žádné zabezpečení nebo autentifikaci
- MaCAN je zpětně kompatibilní nadstavba na CANu, umožňující podepisovat zprávy
- Knihovna implementovaná katedrou řídicí techniky

Modifikace MaCAN knihovny

- MaCAN knihovna má silně abstrahované HW závislosti
 - Většina práce tedy spočívala v doplnění KLEE "hardwaru"
- Byl vytvořen klient pro práci s knihovnou
- Vzhledem ke svému účelu používá kryptografická primitiva, která bylo potřeba obejít

Nalezitelné chyby

- Assertion violation
- Přístup k nealokované paměť
- Dělení nulou
- Přetečení integrální aritmetiky

Nalezené chyby

- Testování našlo 10 potenciálních problémů, všechny v integrální aritmetice
- 3 z nich byly reálné chyby
- 1 byla nalezena již dříve
- 1 byla opravena
- 1 je neškodná pro současné implementace HW závislostí

Evaluace

- MaCAN knihovna v testované konfiguraci má 1340 LOC, 1700 instrukcí
- Nejlepší běh pokryl 60% všech větví a 77% všech instrukcí
- Výsledky jsou z běhu dlouhého 5 dní
- Celkové výsledky byly dostatečně dobré, aby bylo KLEE přidáno mezi způsoby testování MaCAN knihovny

Problémy

- Miskonfigurace verifikovaného programu může vést k nalezení neexistujících chyb
- Defaultní heuristika vede k zaseknutí KLEE (chyba v paměťové alokaci)
- Ani při 100% pokrytí by analýza nebyla kompletní
 - Musel jsme obejít kryptografii
 - Netestoval jsem všechny možné konfigurace knihovny

Možné pokračování

- ➊ Heuristika pro prioritní pokrytí chybových stavů
- ➋ Přidání detekcí dalších chyb
- ➌ Možnost označit specifickou chybu za neškodnou

Děkuji za pozornost

Více problémů nalezeno přes dfs10

Jedná se o přehlédnutí v tabulce, dfs10 má větší pokrytí.

Path	Instrs	Time(s)	ICov(%)	BCov(%)	ICount	TSolver(%)
15dfs	4547644	416400.30	75.37	57.85	1697	99.93
10dfs	52135650	302651.30	77.61	60.76	1697	99.39
covnew	4834942	51827.84	54.10	38.66	1697	99.59

Ostatní

- ① Hloubka zpráv
 - covnew spuštění běžel s hloubkou 15 stránek.
- ② Doba běhu
 - Měření není přesné, dochází k dilataci (extrémní případ je zaseklý covnew)
 - Tabulka 4.7 používá data z nejúspěšnějšího běhu, který běžel kratší dobu
- ③ Paměťové nároky a pokrytí se s časem zvětšují