LinuxTesting.org

Analysis Strategies for Variability Bugs in the Linux Kernel

Vadim Mutilin

ISP RAS

Ivannikov Institute for System Programming of the RAS

Linux as a Product Line

- Feature model – Kconfig (>10 000 options)
  - depends, if condition, visibility, default values
  - nonboolean types string, int, tristate
  - *select – cross hierarchy dependencies
  - *tree structure traversal order in .config
- Variability mapping to the code
  - Kbuild – conditions in Makefiles
  - #ifdef conditions

Variability Bugs
(appears not in all configurations)

- Inconsistencies in feature model
- Preprocessor errors
- Compile errors
- Module link errors
- Kernel link errors
- Semantic errors
    - General (e.g. memory safety, data races)
    - Specific

Analysis Strategies

- Predefined configuration
  - allmodconfig, allyesconfig ...
- Sampling
  - #ifdef block-coverage, most enabled-disabled, one-enabled, one-disabled, pair-wise, three-wise ...
- Variability-aware analysis
  - Undertaker, TypeChef-based

Undertaker

- Properties
  - Dead-code analysis of #ifdef blocks
  - Checking of feature model
- Abilities
  - Translation of feature model into a boolean formula
  - Parsing of preprocessor directives

TypeChef

- Properties
  - Type checks
  - Live variable analysis
    [J.Liebig, A.Rhein, C.Kästner, S.Apel, J.Dörre, C.Lengauer. Scalable Analysis of Variable Software]
- Abilities
  - Variability-aware parsing (constructs a syntax tree with variability choices)

Variability-aware Software Model-Checking

- Rewriting (variability encoding)
  - A.Iosif-Lazar, J.Melo, A.Dimovski, C.Brabrand, A.Wąsowski. Effective Analysis of C Programs by Rewriting Variability, 2017
  - S.Apel, H.Speidel, P.Wendler, A.Rhein, D.Beyer. Detection of Feature Interactions using Feature-Aware Verification, 2011
- Abstract model-checking (not software)

Variability Bugs in Linux

- 17 bugs in drivers out of 42 from the paper
  [I.Abal, C.Brabrand, A.Wasowski. 42 Variability Bugs in the Linux Kernel: A Qualitative Analysis]
- 11 appear in x86_64 (6 require other architectures)
- Properties
  - Null pointer dereference – 5
  - Buffer overflow – 3
  - Compile errors – 3
  - Kernel link errors – 2
  - Specific – 3 (Domain rule – 2)

Variability Bugs in Linux (2)

- Variability conditions
  (only key options – missing dependencies for feature model and build)

Condition	Qty	
a	1	2
!a	1	
a && b	5	11
a && !b	6	
a && b && c	2	4
a && b && !c	2	

Variability Bugs in Linux (3)

- Variability conditions in
  - Compilation unit – 11
    - single C – 6
    - headers – 5
  - Multimodule – 6

Predefined Configurations

- Good enough #ifdef block coverage (78%)
- Valid reproducible .config files
- Misses negative options (!option)
- 6 from 17 bugs appear in all[yes/mod]config x86_64

Sampling

- Can reuse existing analyses without modifications
- Heuristics (without – too many configurations
  [F. Medeiros, C.Kästner, M.Ribeiro, R.Gheyi, S.Apel.
  A Comparison of 10 Sampling Algorithms for Configurable Systems])
  - No feature model
  - No build dependencies
  - Only C files (without headers)
- No valid .config files

Results for TypeChef

- Searching for type errors in a single compilation unit

Total			17
Do not check the corresponding property			12
Not a single compilation unit			2
Potentially may find			3
	Found		0
	Found with hints		1
	Found in examples		3

Results for Klever

- Domain specific rules, memory safety, data races
- Manually prepared configuration

Total			17
Not x86_64			6
Compile&link errors			3
Potentially may find			8
	Need to fix/update Environment Model Generator		4
	Need to write specification for the environment		2
	Need to write rule specification		1
	Bug #9233		1
	Found		0

Main Challenges

- Get valid reproducible .config files
- Scalable on-demand variability-aware parsing (by hints from analysis?)
- Variability-aware model checking
  - without rewriting?
  - sample-based?

LinuxTesting.org

Thank you!

Vadim Mutilin
http://linuxtesting.org/project/ldv

ISP RAS
Ivannikov Institute for System Programming of the RAS