

PB173 – Ovladače jádra – Linux

X. b.

Jiří Slabý

ITI, Fakulta Informatiky

30. 11. 2010

Hledání chyb

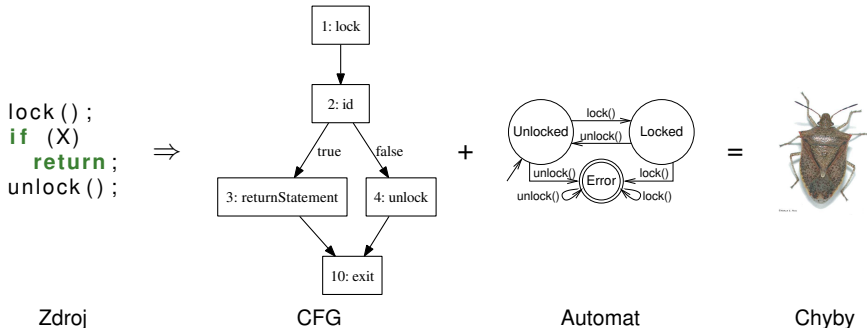
- Online
 - Testování jádra
- Offline
 - Statická kontrola
 - Model-checking

Online (jádro musí běžet)

- Automatické testy
 - Kompilace různých konfigurací jádra
 - Bootování systému
 - <http://kisskb.ellerman.id.au/kisskb/branch/9/>
- Unit-testy
 - Linux Test Project (LTP)
 - Pomocí systémových volání se zkouší co nejvíce cest v ovladači
 - Všechny `ioctl` s různými argumenty
 - Nesprávné čtení přes `read` atp.
 - Domácí úkol

Jednoduchá analýza toku – offline (stačí kód)

- Relativně rychlá, mnoho falešných hlášení (FP)
- Existují (drahé \$) nástroje s minimem FP
- Mnoho nástrojů
 - SPARSE, SMATCH, COVERITY, STANSE (ITI)



Zdroj

CFG

Automat

Chyby

Tabulka: Postup základní statické analýzy

Použití SPARSE na pb173/10

- 1 make C=2
- 2 Opravit nahlášené chyby

Symbolická exekuce – offline

- Pomalejší (někdy výrazně), minimum FP
- Jen pár nástrojů
 - KLEE (kontroloval vybrané souborové systémy)

$$\begin{array}{l} a \neq b; \\ \text{return } a; \end{array} \Rightarrow \begin{array}{l} \%0 = \text{sdiv } \%a, \%b \\ \text{ret } \%0 \end{array} + \begin{array}{l} (\varphi \wedge b = 0) \\ \vee \\ (\psi \wedge b \neq 0) \\ \Downarrow \\ b = 0? \end{array} = \text{Chyba}$$

Zdroj

Low-level kód

Řešení podmínky
cesty

Chyby

Tabulka: Postup rozšířené statické analýzy

Offline

- Matematicky se dokazuje, co může nastat a co ne
- Trpí explozí cest
- Jen pár nástrojů
 - SPIN
- *Vstup*: model chování, kód
- *Výstup*: za jakých podmínek je model splněn

- S/G DMA, MSI či jiné HW chuťovky
 - Nemáme HW, pouze teoreticky
- Základy HW (paging, KBD, VGA/VESA, PIC, TIMER, PCI BIOS)
 - Spustit qemu bez OS a pracovat s (emulovaným) HW
- Opravdové ladění chyb
 - Odladit kód s úmyslně přidanými chybami
 - K dispozici bude Oops
- Síťovky (network stack)
 - Vyrobit virtuální síťovku
- Opakování jakéhokoliv tématu ze cvičení
 - Např. víc do hloubky
- Práce s GITem
- Perf
- ...