Formálne metódy a informačný tok

Školiteľ:doc. RNDr. Damas Gruska, PhD.

Autor: Bc. Michal Pázmány

Cieľ

- Zameranie na problematiku dôveryhodnosti programov a prípadných únikov informácií v zdrojovom kóde
- Budeme analyzovať informačné toky v oblasti objektovo-orientovaných jazykov
- Návrh riešenia v jazyku Java

Čo to znamená?

- Sú aplikácie pre nás dôveryhodné?
- Neunikajú žiadne naše informácie?
- Chránia naše súkromie?

Príklady

Príklad explicitného zakázaného toku:

```
int {public} l;
int {secret} h;
...
l = h;
```

Príklad implicitného zakázaného toku (keď dáta pretečú nepriamo):

```
int {public} x;
boolean {secret} b;
...
if(b){
        x = 1;
}
```

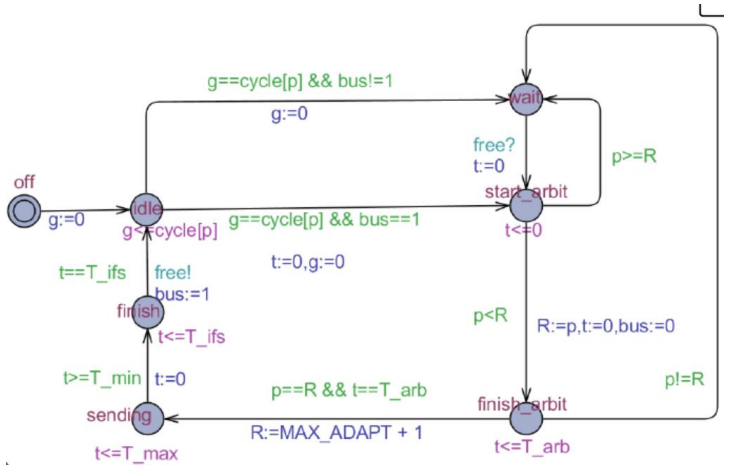
```
int {public} l;
int {secret} h;
...
if(h >= 0){
        l = 1;
}else{
        l = 3;
}
```

Riešenie

- JIF – samostatný programovací jazyk, ktorý je vlastne rozšírením jazyku Java
- Riešenie pre multi-thread aplikácie nie je rozšírené
- Pridáva statickú kontrolu informačného toku pomocou anotácií toku
  - statická kontrola informačného toku zahrňujúca objekty, subtriedy, výnimky, testy dynamických typov

Riešenie

- Pomocou časových automatov
- Nástroj UPPAAL



Existujúce riešenia

- JFlow
  - Najúspešnejšie riešenie
  - rozšíril Javu o statickú aj dynamickú analýzu informačného toku
- SPARK Examiner
  - Je formálne definovaný multiaradigmický jazyk postavený na programovacom jazyku Ada
  - Využíva sa hlavne v systémoch, kde je potrebná vysoká spoľahlivosť operácií a vysoká integrita

Zdroje

- Andrew C. Myers -JFlow: Practical Mostly-Static Information Flow Control
- John Barnes - High Integrity Software: The SPARK Approach to Safety and Security