Model Checking and Security

Oleg Mürk and Mats Jansborg

Background

- Most nontrivial software contains bugs
- Bugs are potential security vulnerabilities
- Verify that the implementation follows its specification

Soundness and Completeness

- Soundness
  - All errors are reported
- Completeness
  - No false positives
- Together undecidable for any non-trivial properties

Model Checking

- Create a restricted model of the program which is amenable to verification
- Express the specification of the program in some temporal logic
- Algorithmically verify correctness
- Partial solution

Goals

- Survey the techniques used in model checking software
- Describe and compare a few of the most important tools
- Applications to security
- Analyze the capabilities and limitations

Specification

- Temporal logic
  - LTL
  - CTL
  - CTL*
- Büchi automata
- Properties
  - Safety
  - Liveness

Abstraction

- From program to finite model
- Slicing
- Predicate abstraction
- The objective: minimize the number of states

Model checking

- Construct the intersection of model and negated specification
- Search for acceptance cycles in the state space graph
- Combat state space explosion
  - Partial order reduction
  - State hashing

Tools: Spin

- Generic model checker
- Input language Promela
- Transition systems
- Bounded N of thread
- Bounded heap allocation
- Started in beginning 90's
- Received ACM award in 2001

Tools: Bandera & Bogor

- Platform for model checking
- Java oriented
- Input language BIR
  - Models Java constructions
  - Static type system
  - Virtual method table
  - Heap & GC
  - Threads
  - Monitors
- More expressive => Specific optimizations
  - Canonical heap representation
- Also translate to SPIN, etc

Applications

- Verifying security protocols
- Verifying interface contracts
  - MOPS – simplistic
  - SLAM – advanced
- Information flow security

Conclusions

- Model checking can check
  - Safety
  - Liveness
  - Non-properties: information flow security
- Universality => Inefficiency
- Model checking
  - SPIN, Bandera & Bogor – reactive systems
  - MOPS – simple tool
  - SLAM – advanced tool
- Specialized analysis
  - Works best on single-threaded

Conclusions

- Data Flow Analysis
  <=>
  Model Checking + Abstract Interpretation

- Static analysis
  - Data Flow Analysis
  - Abstract Interpretation
  - Model checking

- Combine & reinforce!