Opaque properties and SMT-solvers

Alexandre Gonzalvez[1,2], Olivier Decourbe[2], Sebastien Josse[3], Caroline Fontaine[4], Axel Legay[5]

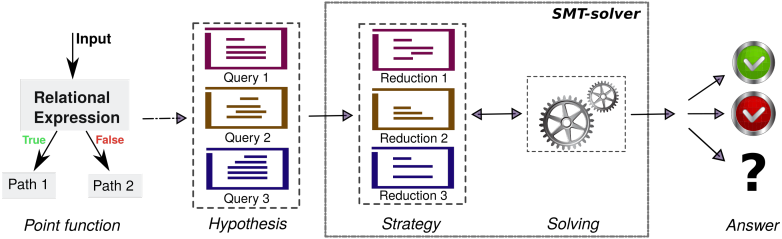
1: IMT Atlantique, 2: Inria, 3: DGA, 4: CNRS & LSV, 5: UCLouvain

December, 2018

Motivations

- **Problem** : Cyber threat analysts want to (partially) deobfuscate a family of malware that use an anti-tampering mechanism based on *opaque properties*, in order to obtain at least a behavioural signature.
- *Opaque properties* = the desire to increase the time of analysis of a code or a binary performed by a human or a machine or both

Context



Abstraction can be realized with a framework and sent to an SMT (Satisfiability Modulo Theories) solver, which checks satisfiability of given hypothesis in regards to some background theory, and approximations.

Problem analysis

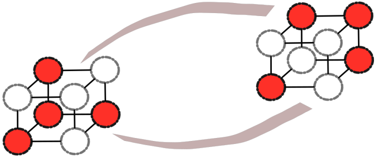


Figure: Knowledge: structure, valuation and model

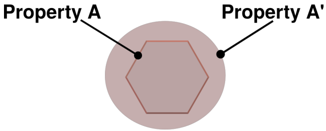


Figure: Hypothesis and approximation

An attacker point of view against opacity properties:

- To explain why opacity properties have a negative impact in the learning process made by an SMT solver

Experience

In the aim to reduce the time for the analysis i.e. *the number of steps to learn a concept* composed with opacity properties, hypothesis need to be rewritten and adapted for each opacity property.

Experience

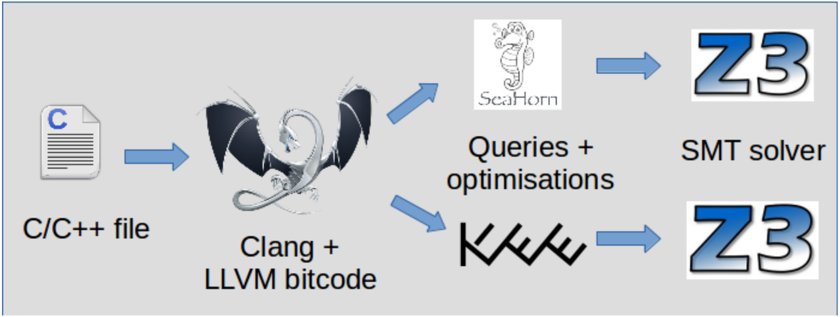


Figure: Simplified architecture of Seahorn (Gurfinkel et al.), and Simplified architecture of KLEE (Cadar et al.)

Experience

The APartow hash function (Aphash) composed with a constant expression with free variables $(x * (x + 1)\%2)$:

	Aphash	
Solver	Input size (char)	Time (sec)
Seahorn - Z3	5	UNSAT
	10	UNSAT
	15	UNSAT
KLEE - Z3	5	952
	10	184
	15	TO
Our solution	5	47
	10	55
	15	80

TO = 20 min = 1200 sec

Conclusion

- A "pre-processing" step for queries can reduce the impact of one opaque property
- Future work: To automatize this pre-processing step for some opaque properties

Thank you for your attention!

contact : alexandre.gonzalvez@inria.fr