GenISIS: un outil de recherche d’attaques d’initié en Systèmes d’Information

Authors: Amira RADHOUANI
Akram IDANI
Yves LEDRU
Narjes BEN RAJEB

Laboratoire d’Informatique de Grenoble
Information System security includes = Protection against external intruders + Insider attacks.

Great!!! thanks to my access, i'll transfer all to my account

They are not smart enough these IT!
1. Introduction
   - Illustration example

2. Malicious behavior

3. Extraction of malicious behaviors
   - Extraction of malicious behaviors from B Specification

4. Conclusion
   - Constraint-solving based approach
   - GenISIS tool
Introduction

1. Illustration example
2. Dynamic analysis
Dynamic analysis searches for sequences of actions **modifying the state** and **breaking the authorization constraint**.
Malicious behavior
A malicious behaviour executed by a user $u$, regarding authorization constraints, is an observable secure behaviour $Q$ with $m$ steps such that:

- user $u$ is malicious and would like to run $\text{op}_m$ by misusing his roles $R_u$.
- $\text{val}_0$ : is an initial state where $(u, R_u, c_m) \models \text{false}$
- for every step $i$ ($i \in 1..m$) premise $(u, R_u, c_i) \models \text{true}$

[A. Radhouani et al., Trans. Petri Nets and Other Models of Concurrency 10: 131-152 (2015)]
Extraction of malicious behaviors

1. Extraction of malicious behaviors from B Specification
2. Proof based approach
3. Constraint solving based approach
4. GenISIS Tool
Symmetric transition system
Symbolic proof

Proof obligations on reachability properties:
- Having $E$ and $F$, 2 disjoint state predicates
- And $op(x_1,x_2,...,x_n)$ is an operation of the IS.

- Enabledness: $\exists x_1,\ldots,x_n, \text{var} . P_I \land \text{Pre}(op)$
- Reachability: $\exists x_1,\ldots,x_n, \text{var} . P_I \land \text{Pre}(op) \Rightarrow \neg [\text{Action}(op)] \neg P_F$

$\exists x_1,\ldots,x_n, \text{var} . P_I \land \text{Pre}(op) \land \neg [\text{Action}(op)] \neg P_F$
EXTRACTION OF MALICIOUS BEHAVIORS FROM B SPECIFICATION

\[ c_m \land \neg Pre(o_i) \land \neg e_m \lor \neg Pre(op_m) \land Pre(o_i) \]

\[ c_m \land Pre(op_m) \]

[Diagram with logical expressions and symbols]
EXTRACTION OF MALICIOUS BEHAVIORS FROM B SPECIFICATION

\[ \neg c_m \lor \neg \text{Pre}(op_m) \]

\[ \land \neg \text{Pre}(o_i) \]

\[ \land \text{Pre}(o_i) \]

\[ c_m \land \text{Pre}(op_m) \]
EXTRACTION OF MALICIOUS BEHAVIORS FROM B SPECIFICATION

\[ \neg c_m \lor \neg Pre(op_m) \]

\[ \land \neg Pre(o_{i-1}) \]

\[ \land Pre(o_{i-1}) \]

\[ \land Pre(o_i) \]

\[ c_m \land Pre(op_m) \]
$\neg c_m \lor \neg Pre(op_m) \\
\land \neg Pre(o_{i-1})$
EXTRACTION OF MALICIOUS BEHAVIORS FROM B SPECIFICATION

\[
\neg c_m \lor \neg Pre(op_m) \\
\land \neg Pre(o_{i-1}) \\
\land \neg Pre(o_{i-1}) \\
\land Pre(o_{i-1}) \\
\land Pre(o_i) \\
c_m \land Pre(op_m)
\]
EXTRACTION OF MALICIOUS BEHAVIORS FROM B SPECIFICATION

\[ \neg c_m \lor \neg Pre(op_m) \]
\[ \land \neg Pre(o_i) \]

\[ \land \neg Pre(o_{i-2}) \]

\[ \land \neg Pre(o_{i-1}) \]
\[ \land Pre(o_{i-2}) \]

\[ \land Pre(o_{i-1}) \]

\[ \land Pre(o_i) \]

\[ c_m \land Pre(op_m) \]

\[ Q \triangleq init ; \quad ; \quad ; \quad ; \quad op_m \]
First step: Use of a prover (AtelierB) to extract symbolic operations.

Second step: Use a model checker (ProB) to find operation valuations after eliminating operations which don’t appear in the first step.

AtelierB fails to discharge automatically PO when the proof becomes huge.

In our example:
- First iteration: 3 extra operations are kept.
- Second iteration: automatic proof fails for all operations.

Unable to extract scenarios that involve the same operation several times.

Constraint solving problem:

\[ \{x_1, \ldots, x_n \mid \exists \text{var. } P_I \land Pre(op) \land \neg [Action(op)] \neg P_F \} \]

- Allows to valuate operation parameters.
- Simplifies the proof.
- Allows to extract scenarios which involves the same operation several times (the same operation with different valuations).
GENISIS TOOL

-Generator of Insider Scenarios from an Information System-

In

GenISIS Process

Out

JBtools

BoB

Xtext

ProB

Constraint solving problems

Constraint solver

SpeCS

CCL

Javaapi

$op_m$

$op_m$

$op_m$
Conclusion
GenISIS was able to extract 9 scenarios.
- 2 real attacks: allowed in the security model.
- 7 fake attacks: not allowed in the security model.

A model-checker (i.e ProB) extracted the same attacks after exploring more than 1500 states and 36000 transitions.

GenISIS was successfully tested on 5 case studies.

Try it, it is available on open source in: http://genisisforge.imag.fr/
Thanks for your attention