TwoTowers 4.0: Towards the Integration of Security Analysis and Performance Evaluation

Alessandro Aldini and Marco Bernardo
Università di Urbino "Carlo Bo"
Istituto di Scienze e Tecnologie dell'Informazione
Piazza della Repubblica 13, 61029 Urbino, Italy

Abstract

We present TwoTowers 4.0, an extended version of the software tool TwoTowers encompassing security analysis. The novelty of TwoTowers 4.0 is its capability of supporting both security analysis and performance evaluation, thus providing the means for trading the QoS of a system with the bandwidth of its illegal information leaks.

1. Introduction

The recent trends to open and distributed computing rise the problem of protecting computer systems against attacks by malicious parties, which try to violate the confidentiality and compromise the integrity of private information. Solutions to this security problem aim at minimizing the impact of the securing strategies on the system QoS. Hence, the problem of delivering an adequate QoS by minimizing the risk of information leakages is a fundamental goal of the early system design phases.

The key to achieve this goal is to integrate security analysis and performance evaluation, two activities usually carried out separately. In [2] an integrated methodology encompassing both activities is presented, which provides insights about how to trade the QoS delivered by the system and the bandwidth of its information leakages. The methodology is supported by TwoTowers 4.0, a suitably extended version of the software tool TwoTowers 3.0 [3] that encompasses a security analyzer. In the following, we describe the architecture of TwoTowers 4.0 (Sect. 2) and its new security analyzer (Sect. 3). Then we report on an application to a real case study, the NRL Pump (Sect. 4). Finally we discuss some future work (Sect. 5).

2. Tool Architecture

TwoTowers 4.0 is an automated tool for the analysis of systems described in the architectural description language ÆEmilia [4]. As shown in Fig. 1, TwoTowers 4.0 is equipped with a graphical user interface through which the user can invoke the analysis routines. The graphical user interface takes care of the integrated management of the various file types needed by the different routines.

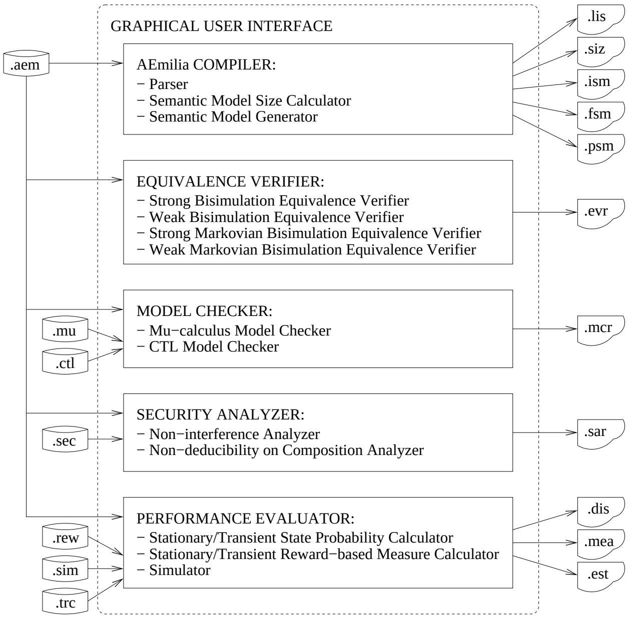
GRAPHICAL USER INTERFACE
.aem
AEmilia COMPILER:
− Parser
− Semantic Model Size Calculator
− Semantic Model Generator
.lis
.siz
.ism
.fsm
.psm
EQUIVALENCE VERIFIER:
− Strong Bisimulation Equivalence Verifier
− Weak Bisimulation Equivalence Verifier
− Strong Markovian Bisimulation Equivalence Verifier
− Weak Markovian Bisimulation Equivalence Verifier
.evr
MODEL CHECKER:
− Mu−calculus Model Checker
− CTL Model Checker
.mu
.ctl
.mcr
SECURITY ANALYZER:
− Non−interference Analyzer
− Non−deducibility on Composition Analyzer
.sec
.sar
PERFORMANCE EVALUATOR:
− Stationary/Transient State Probability Calculator
− Stationary/Transient Reward−based Measure Calculator
− Simulator
.rew
.sim
.trc
.dis
.mea
.est

Figure 1. Architecture of TwoTowers 4.0

The compiler is in charge of parsing system specifications and signalling possible lexical, syntax and static semantic errors. If a specification is correct, the compiler can generate its integrated, functional or performance semantic model, or simply report the size – in terms of number of states and transitions – of the semantic model.

The equivalence verifier checks whether two correct specifications are equivalent according to one of several different notions of equivalence. In case of non-equivalence, a distinguishing modal logic formula is reported.

The model checker verifies whether a set of μ-calculus or CTL formulas are satisfied by a correct specification. The check is executed by invoking the CWB-NC 1.2 tool [5].

The novel security analyzer verifies whether a correct specification possesses certain security properties establishing the absence of illegal information flows from high security components to low security components. In case of property violation, an illustrative modal logic formula is reported.

Finally, the performance evaluator computes the performance characteristics of correct and performance closed specifications. The computation of instant-of-time, stationary/transient performance measures specified through state and transition rewards is carried out via reward Markov chain solution or discrete event simulation.

3. Description of the Security Analyzer

The security analyzer of TwoTowers 4.0 implements a technique based on the non-interference theory [7]. Basically, such a theory assumes that sensitive information is classified into two access levels (low – corresponding to unclassified – and high – corresponding to secret), and users are assigned clearances, such that users can only access information classified at or below their clearances.

Based on the non-interference approach, two security properties can be checked by the security analyzer of TwoTowers 4.0. The first one, called strong nondeterministic non-interference [6], consists of verifying whether the view of the system behavior as observed by a low user in the absence of high user interferences is the same as that observed when the high users interact with the system. The second one, called strong non-deducibility on compositions [6], consists of verifying whether the view of the system behavior as observed by a low user is always the same before and after the execution of any interaction between the system and a high user.

In order to verify one of these two security properties, it is required to specify in an additional .sec file the action names that are high and the action names that are low with respect to the security level. Then, the verification is performed by comparing the two low views of the system based on weak bisimulation equivalence, and a modal logic formula is shown in case of security violation.

4. Application of the Security Analyzer

In [2] TwoTowers 4.0 has been applied to assess the effectiveness and the efficiency of the securing strategy implemented in the NRL Pump, a trusted device that delivers the secure replication of information from a low-security level enclave to a high-security level enclave. Even though the securing strategy implemented in the NRL Pump succeeds in mitigating most of the illegal information flows from high to low, the system suffers from an unavoidable information leakage. To assess how much information illegally flows through the NRL Pump, an Æmilia description of the NRL Pump has been provided and analyzed through the security analyzer and the performance evaluator of TwoTowers 4.0. The obtained results show the trade-off between the bandwidth of the information leakage, expressed as the number of bits leaked per unit of time, and the NRL Pump configuration parameters. Moreover, by exploiting the performance feedback provided by the integrated analysis, a strategy to reduce the information flow with a minor impact on the QoS has been inferred.

5. Future Extensions

First, we would like to extend Æmilia and TwoTowers 4.0 in order to provide support for further non-interference-based security properties that are defined to check for secrecy and integrity of data, authentication of involved parties, and non-repudiation of service.

Second, we would like to implement in TwoTowers 4.0 the probabilistic variants of the considered security properties [1].

Third, we would like to strengthen the integration of security analysis and performance evaluation, in such a way that the illustrative modal logic formula returned in case of security violation automatically determines the performance metrics affecting the bandwidth of the information leakage.

References

[1] A. Aldini, M. Bravetti, and R. Gorrieri, *A Process-algebraic Approach for the Analysis of Probabilistic Non-interference*, in *Journal of Computer Security* 12(2), 2004.

[2] A. Aldini and M. Bernardo, *An Integrated View of Security Analysis and Performance Evaluation: Trading QoS with Covert Channel Bandwidth*, to appear in Proc. of SAFECOMP'04, LNCS, 2004.

[3] M. Bernardo, *TwoTowers 3.0: Enhancing Usability*, in Proc. of MASCOTS'03, IEEE-CS Press, pp. 188–193, 2003.

[4] M. Bernardo, L. Donatiello, and P. Ciancarini, *Stochastic Process Algebra: From an Algebraic Formalism to an Architectural Description Language*, in *Performance Evaluation of Complex Systems: Techniques and Tools*, LNCS 2459:236–260, 2002.

[5] W.R. Cleaveland, T. Li, and S. Sims, *The Concurrency Workbench of the New Century - Version 1.2 - User's Manual*, `www.cs.sunysb.edu/~cwb/`, 2000.

[6] R. Focardi and R. Gorrieri, *A Classification of Security Properties*, in *Journal of Computer Security* 3:5–33, 1995.

[7] J.A. Goguen and J. Meseguer, *Security Policy and Security Models*, in Proc. of SSP'82, IEEE-CS Press, pp. 11–20, 1982.