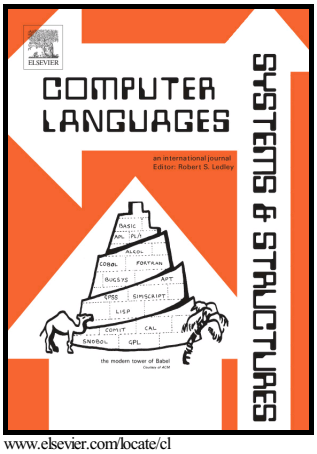


www.elsevier.com/locate/cl

# Model-based Analysis of Java EE Web Security Misconfigurations

Salvador Martínez, Valerio Cosentino, Jordi Cabot

PII: S1477-8424(16)30134-8
DOI: http://dx.doi.org/10.1016/j.cl.2017.02.001
Reference: COMLAN246

To appear in: *Computer Language*

Received date: 10 October 2016
Revised date: 27 January 2017
Accepted date: 3 February 2017

Cite this article as: Salvador Martínez, Valerio Cosentino and Jordi Cabot Model-based Analysis of Java EE Web Security Misconfigurations, *Compute Language,* http://dx.doi.org/10.1016/j.cl.2017.02.001

# Model-based Analysis of Java EE Web Security Misconfigurations

Salvador Martínez[a,b,*], Valerio Cosentino[a,d], Jordi Cabot[c,d]

[a] *AtlanMod team (Inria, Mines Nantes, LINA), Nantes, France*
[b] *CEA, LIST, Laboratory of Model Driven Engineering for Embedded Systems, Gif-sur-Yvette, France*
[c] *ICREA, Barcelona, Spain*
[d] *UOC, Barcelona, Spain*

**Abstract**

The Java EE framework, a popular technology of choice for the development of web applications, provides developers with the means to define access-control policies to protect application resources from unauthorized disclosures and manipulations. Unfortunately, the definition and manipulation of such security policies remains a complex and error prone task, requiring expert-level knowledge on the syntax and semantics of the Java EE access-control mechanisms. Thus, misconfigurations that may lead to unintentional security and/or availability problems can be easily introduced. In response to this problem, we present a (model-based) reverse engineering approach that automatically evaluates a set of security properties on reverse engineered Java EE security configurations, helping to detect the presence of anomalies. We evaluate the efficacy and pertinence of our approach by applying our prototype tool on a sample of real Java EE applications extracted from GitHub.

*Keywords:* Model-Driven Engineering, Security, Reverse-engineering

## 1. Introduction

Java EE is a popular technology of choice for the development of dynamic web applications (serving also as the basis for other less general purpose frameworks) that expose distributed information and services to remote users. In this scenario, security is a main concern [1], as the web resources that constitute the web application can be potentially accessed by many users over untrusted networks. As a consequence, the Java EE framework provides developers with the means to specify access-control policies in order to assure the confidentiality and integrity properties of the resources exposed by web applications.

Unfortunately, despite the availability of these security mechanisms, implementing security configurations remains a complex and error prone activity where high expertise

*Corresponding author
*Email address:* salva.martinez@mines-nantes.fr (Salvador Martínez)