



Doctoral Dissertation
Doctoral Program in Computer and Control Engineering (31st cycle)

# An Expert System for Automatic Software Protection

**Leonardo Regano**
* * * * * *

**Supervisors**
Prof. Antonio Lioy, Supervisor
Cataldo Basile, Ph.D., Co-supervisor

**Doctoral Examination Committee:**
Bart Coppens, Ph.D., Referee, Ghent University
Prof. Claudia Raibulet, Referee, Università degli Studi di Milano-Bicocca
Prof. Bjorn De Sutter, Ghent University
Prof. Stefano Paraboschi, Università degli Studi di Bergamo
Prof. Riccardo Sisto, Politecnico di Torino

Politecnico di Torino
17 July 2019

I hereby declare that, the contents and organisation of this dissertation constitute my own original work and does not compromise in any way the rights of third parties, including those relating to the security of personal data.

.....

Leonardo Regano
Turin, 17 July 2019

# Abstract

In the Information Age, software is omnipresent in almost everyone's life. A plethora of services is offered in a digital manner, for example with e-government portals, e-banking mobile apps, and e-commerce websites. Indeed, users trust the software enabling such services with their personal information, like credit card numbers or e-banking credentials. Thus, software must be protected with care, in order to mitigate possible threats against such valuable data. Furthermore, software companies have to protect the assets in their applications, such as intellectual property of algorithms, methods preventing unauthorized application distribution, and users' personal data. Failing to do so may deeply damage software companies' finances, due to lost application sales, and reputation, if users' personal data is leaked.

However, protecting applications is a cumbersome task, reserved to few expert practitioners of this field, due to the rising complexity of applications, and the availability of numerous protection techniques. Each of the latter has strengths and weakness, and their effectiveness in safeguarding the application assets depend on numerous factors, such as the structure of protected code, the tools employed to apply such protections along with their configuration parameters, and the expected skills of a possible attacker that may be interested in breaching the application assets.

In this thesis, an expert system for automating the protection of applications is presented. To the best of the author's knowledge, it is the first application of the expert system paradigm to this challenging problem. Mimicking the decision process of a software security expert, the system, given the source code of an application that must be protected, is able to produce a binary of the application, hardened with the protection technique most suitable to defer, for the longest time possible, an attacker aiming to breach the application assets. Apart from the program source code, the system requires from the user only a list of the assets that must be protected, with each of them associated with one or more high-level security requirements (e.g., confidentiality, integrity), which must be safeguarded against possible attacks. The system has been developed during the EC-funded ASPIRE project, whose objective was to develop a set of protection techniques for Android applications, along with automated tools to deploy them on threatened code. The system is not only able to decide the generic protection techniques that must be applied to the program code, but also the specific parameters to drive such tools, thus providing a comprehensive protection solution specifically tailored for the targeted application. The system is based on the formalization of the mental decision processes and background knowledge of software security experts involved in the aforementioned project.

This thesis advances the state of the art in the field of software security with the following contributions: (1) a meta-model for software security, able to formalize all the related concepts, such as characteristics of the application and of the components of its code, attacks that can be mounted against the application, and protection techniques that can be used to mitigate them; (2) a risk assessment methodology for software, with a formalization of attacks against applications assets, consisting in the identification of simple attack tasks, which can be then chained incomplete attacks that can be carried out to successfully breach the application assets; (3) a risk mitigation strategy, based on a game-theoretic approach, able to infer the protections best able to defer possible attacks against the application; (4) a set of asset hiding strategies, devised to increase the effort needed by an attacker to locate assets in the application binary; (5) a complete and automated workflow for software protection, implementing the aforementioned risk management processes in a fully-fledged expert system.