POLITECNICO DI TORINO
Dipartimento di Automatica e Informatica

PhD in Computer and Control Engineering

XXXI cycle

Supervisor
prof. Antonio Lioy

An Expert System for Automatic Software Protection

PhD Candidate: *Leonardo Regano*

1. Introduction

Nowadays, with cyber attacks on the rise, software must be protected, to avoid risks for users and huge monetary losses for software developers: but protecting software is a difficult task restricted to few security experts.

Idea: an automated system, trained with software security expert's field experience, to enable everyone to protect their software, and also to simplify experts' work.

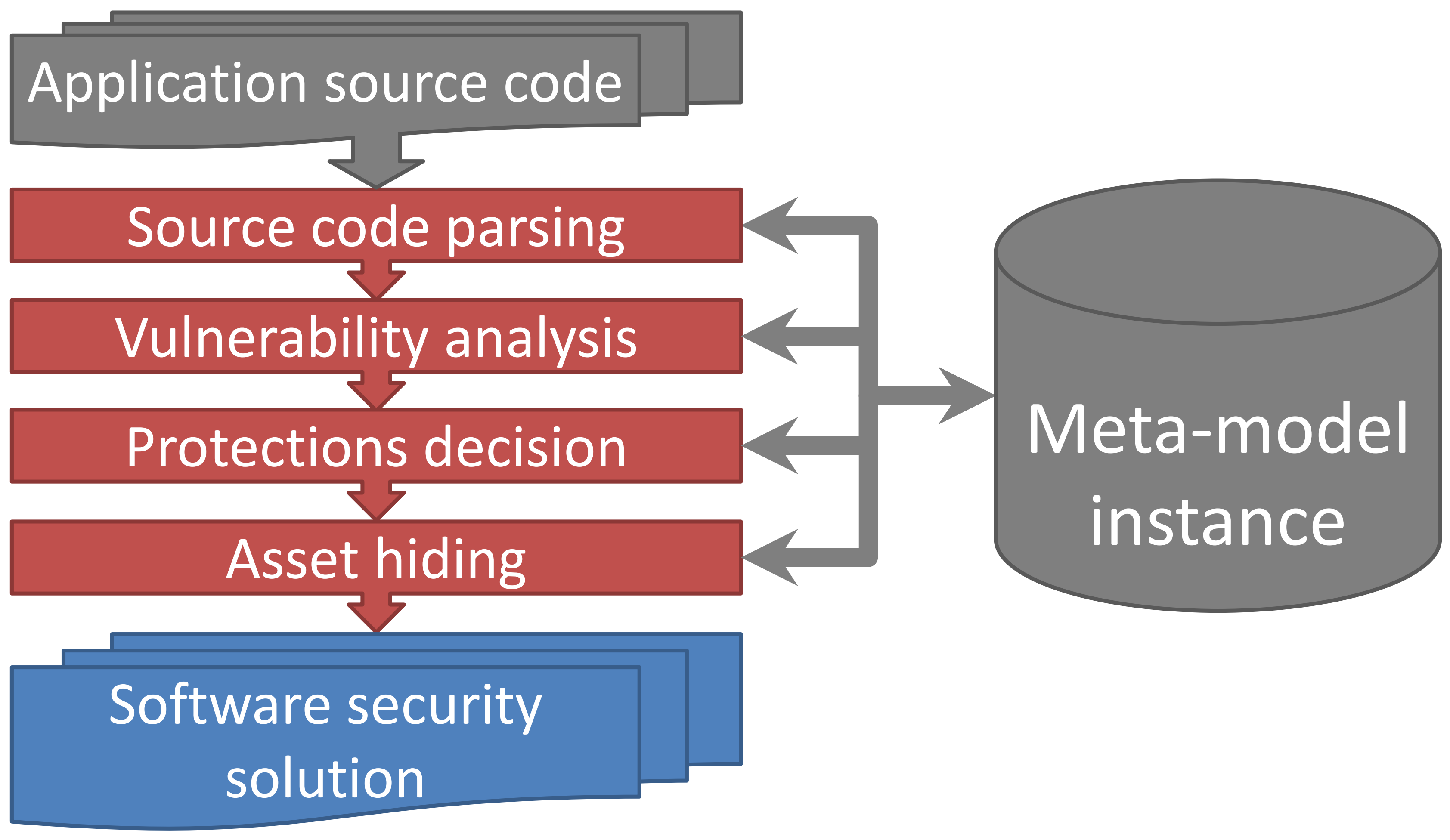
2. Software security meta-model

The need to formalize the general information about software security gathered from security experts, and also the data inferred by the expert system on the target application, lead to the definition of a comprehensive software security meta-model. Implemented as an OWL2 ontology, is able to represent information about:

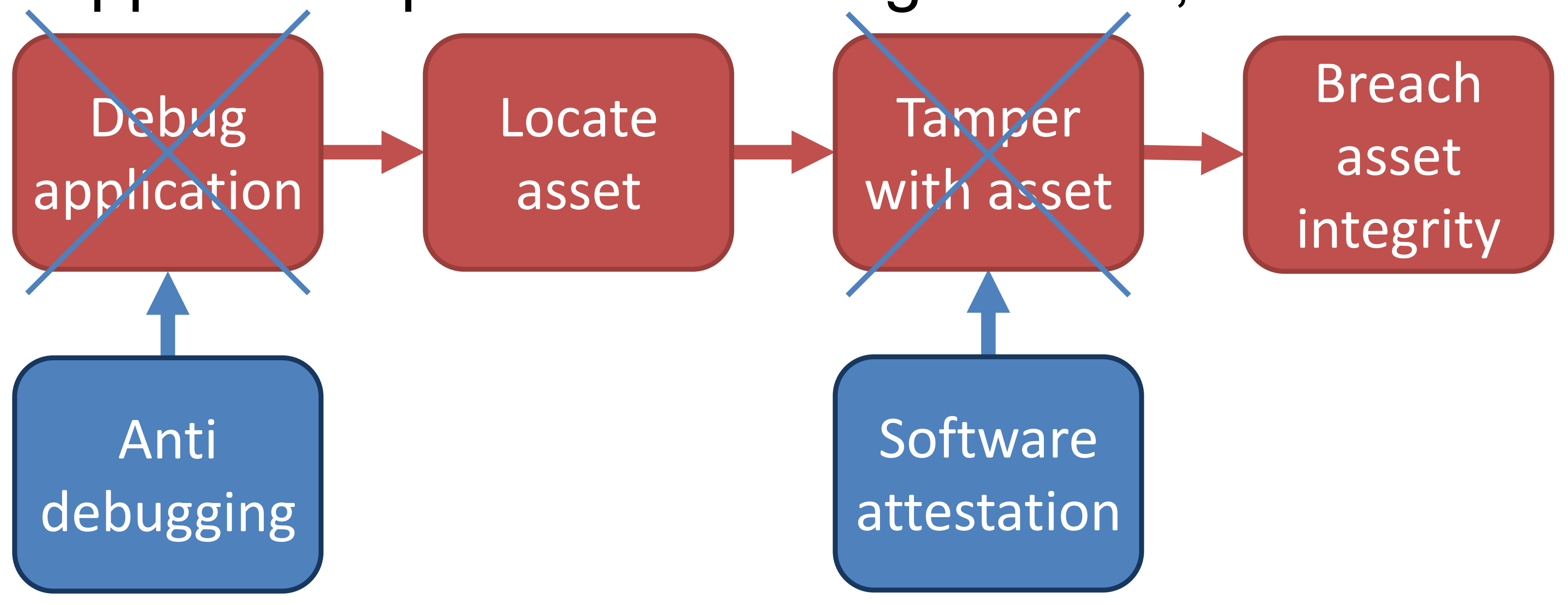
- **application structure:** variables, functions, control flow graph, call graph, etc.;
- **assets:** variables and code snippets that must be protected, with their security requirements, e.g. confidentiality, integrity;
- **attacks:** combinations of simple attacker actions (static/dynamic analysis and tampering), with the required attack tools and expertise;
- **protections:** effectiveness against attacks, synergies between protections, with tool-specific information to deploy them on assets;

3. Expert system for software protection

The system, implemented as a set of Eclipse RCP plugins, is able to analyze and protect C/C++ applications. Is publicly available on GitHub [1].



1. **source code parsing:** based on Eclipse CDT, models the application's source code structure;
2. **vulnerability analysis [2]:** finds possible attacks against assets, with a backward reasoning algorithm written in Prolog;
3. **protections decision:** in order to delay attacks, infers protections that can be applied to the assets, and combines them in a *solution,* able to protect adequately the assets with a limited application performance degradation;



4. **asset hiding [3]:** refines the solution with additional protections, applied even on non critical code areas, to slow attackers in finding and subsequently attacking the assets.

4. Conclusions

The system development started during the ASPIRE FP-7 EU project. It has been tested on several open-source applications and three industrial use-cases: a OTP generator, a software licensing scheme and a DRM video player. Experts from ASPIRE industrial partners validated the solutions inferred on the use-cases.

5. References

1. https://github.com/SPDSS/adss
2. Regano L. *et al.*, Towards Automatic Risk Analysis and Mitigation of Software Applications, WISTP 2016, DOI: 10.1007/978-3-319-45931-8_8
3. Regano L. *et al.*, Towards Optimally Hiding Protected Assets in Software Applications, QRS 2017, DOI: 10.1109/QRS.2017.47