Making Advanced Software Protection Tools Usable for Non-Experts

(Invited Paper)

Bjorn De Sutter*
*Computer Systems Lab
Ghent University, Belgium
bjorn.desutter@elis.ugent.be

***Abstract*—We present the EU FP7 ASPIRE project, with a focus on its design of a decision support system that will enable non-expert users to configure a complex software protection tool chain to protect the assets in their software.**

***Index Terms*—decision support system, protection metrics**

I. Introduction

ASPIRE (https://www.aspire-fp7.eu) stands for Advanced Software Protection: Integration, Research, and Exploitation. In this EU FP7 project, three market leaders in security ICT solutions and four academic institutions join forces to protect the assets of service, software and content providers. The core objective of ASPIRE is to develop an integrated software security framework that allows developers to add effective software protection to applications automatically. The goal is to establish trustworthy execution of software on mobile client devices that lack generic and open security hardware elements to be exploited, but that have a (persistent or occasional) network connection to a trusted entity at their disposal. With our solutions, we want mobile software security to become

- trustworthy by leveraging on the available network connection and developing a layered security approach of strong protections;
- measurable by developing practical metrics based on validated attack and protection models;
- cheaper by integrating support for the protections into an industrial-strength ASPIRE Framework;
- more valuable by enabling shorter times-to-market;
- more productive by being more widely applicable.

A single monolithic protection technique that mitigates all threats on mobile applications is impossible to design. There are simply too many different threats to consider, as well as too many attack methods and tools that need to be mitigated. So instead, a series of techniques needs to be deployed, each with a specific purpose. The approach we therefore conceived for this project is the layered software security approach, where several lines of defense are composed. In the ASPIRE project, we are developing, and will demonstrate and validate a tool chain based on compiler tools with which one can deploy a wide range of concrete protection techniques.

II. Decision Support

A key challenge then is to enable non-expert users to use these tools effectively and efficiently. We aim to do so by developing a decision support system. The main idea is that the programmer annotates the assets he wants to protect, and that the decision support system assists the developer in selecting the protections to apply. This system then instructs the ASPIRE tool chain to implement the protection, discharging the programmers from manually selecting the protections. The decision support system will contain expert knowledge to make such decisions, and its task of course includes estimating the strength of all applicable and potentially useful protection techniques and their potentially useful configurations.

This keynote will focus on a unified framework for measuring and estimating the strength of protections to support automatic selection of "golden combination" of protections given a set of assets and software to be protected, as well as constraints on, e.g., acceptable execution slowdown or network bandwidth requirements.

This framework supports the automated modeling of a wide range of attacks on the relevant assets, and includes different types of software metrics to estimate and evaluate the effort needed to perform different attack steps [1].

The metrics design starts from an analysis of all relevant aspects contributing to attack effort, such as attack goals, attacker capabilities, available tools and techniques, software features and attack actions to be performed. We conjecture that the attack effort can be computed on the basis of several code complexity metrics, many of which need to be combined to cover a wide range of attack methods. These metrics are to be computed on static as well as dynamic program representations, such as graphs and traces. We propose a unified framework to adapt the computation of the metrics to the relevant features of individual attack steps to cover potency, resilience and stealth, the three important aspects of software protections [2].

Acknowledgment

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement number 609734.

References

[1] B. De Sutter, "ASPIRE deliverable D4.02: Preliminary Complexity Metrics," available on the project website.
[2] C. Collberg and J. Nagra, *Surreptitious Software*. Addison-Wesley Professional, 2009.