Hybrid Static-Dynamic Attacks against Software Protection Mechanisms

Matias Madou Bertrand Anckaert Bjorn De Sutter Koen De Bosschere

Electronics and Information Systems Department
Ghent University
Sint-Pietersnieuwstraat 41
9000 Ghent, Belgium
{*mmadou,banckaer,brdsutte,kdb*}*@elis.UGent.be*

ABSTRACT

Advances in reverse engineering and program analyses have made software extremely vulnerable to malicious host attacks. These attacks typically take the form of intellectual property violations, against which the software needs to be protected. The intellectual property that needs to be protected can take on different forms. The software might, e.g., consist itself of proprietary algorithms and datastructures or it could provide controlled access to copyrighted material. Therefore, in recent years, a number of techniques have been explored to protect software. Many of these techniques provide a reasonable level of security against static-only attacks. Many of them however fail to address the problem of dynamic or hybrid static-dynamic attacks. While this type of attack is already commonly used by black-hats, this is one of the first scientific papers to discuss the potential of these attacks through which an attacker can analyze, control and modify a program extensively. The concepts are illustrated through a case study of a recently proposed algorithm for software watermarking [6].

Categories and Subject Descriptors

D.2.0 [**Software Engineering**]: General—*protection mechanisms*; K.4.1 [**Computers and Society**]: Public Policy Issues—*Intellectual property rights*; K.4.4 [**Computers and Society**]: Electronic Commerce—*Intellectual property; Security*; K.5.1 [**Legal Aspects Of Computing**]: Hardware/Software Protection—*copyrights; proprietary rights*

General Terms

Economics, Security

Keywords

Intellectual Property, Software Protection, Attacks, Obfuscation, Watermarking, Tamper-resistance

DRM'05, November 7, 2005, Alexandria, Virginia, USA.

1. INTRODUCTION

The intellectual property contained within and protected by software is enormous. The cost of software piracy alone is estimated at 29 billion dollars for the year 2003 [18]. Besides the copying of the entire application, software developers are also faced with the threat that valuable parts of their application could be included in a competitor's code [1]. Furthermore, software may provide a controlled access to other copyrighted material [37]. If the software is tampered with, illegal access to this material may be obtained.

This paper focuses on pure software-based mechanisms for self-defense. We do not consider approaches that require changes to the hardware, operating system, or any other part of the computing system, other than a program itself.

A number of approaches to protect software intellectual property have recently gained increased interest [9, 26]. Software watermarking [8, 30, 33] has been proposed as a defense against software piracy, in which the embedding of a copyright notice in the software allows to prove ownership of the code. Software fingerprinting is a related technique that embeds a unique message into each distributed copy to facilitate the tracking and prosecution of copyright infringers. Another technique, obfuscation [10, 21, 35], attempts to transform an original program into a program that is harder to understand. Any attack against software requires an (at least partial) understanding of the program. Obfuscation is a technique to prevent the attacker from acquiring such an understanding, thereby making attacks more difficult. Tamper resistance [2, 3, 16] is a technique that is targeted specifically at making the program unmodifiable. As such, obfuscation and tamper-resistance can be used to reinforce other techniques.

In recent years, many techniques for watermarking, obfuscation and tamper-resistance have been published. However, little research seems to have been done into the vulnerability of these methods. While most of the existing techniques are fairly robust against static attacks, many fail in the presence of dynamic or hybrid static-dynamic attacks. Although the authors of the aforementioned techniques acknowledge that it is not possible to protect software against all conceivable attacks, they do claim to make possible attacks "expensive enough" –in time, effort, or resources– that for most attackers, an attack it is not worthwhile. The hybrid static-dynamic attacks proposed in this paper refute this claim, however, as they can be implemented without significant requirements as to computation time, effort or resources.