A Novel Software Protection Approach for Code Obfuscation to Enhance Software Security

Pratiksha Gautam, Jaypee University of Information Technology, Department of Computer Science, and Engineering, Waknaghat, India
Hemraj Saini, Jaypee University of Information Technology, Department of Computer Science, and Engineering, Waknaghat, India

ABSTRACT
Over the past few decades ago, software developers analyzed robustly several forms of software protection against illegal copying or piracy. With the expansion in digital technology, the risk of illegal copying of software also amplifies. The increasing piracy rate has posed a serious threat to software developers leading to the development of various software protection techniques. However, various techniques have been proposed for copyright protection such as software watermarking, obfuscation, tamper-proofing and diversity. The code transformation (obfuscation) is a method of transforming a program into a form which is more complicated for an adversary to understand or change the original code from an illegitimate process of reverse engineering. None of the current code obfuscation approaches provide resistance from reverse engineering attacks. The reverse engineering threat occurs due to the unconfined software code to the user. Malicious reverse engineering of software codes can be harder by exertion of code transformation on software programs. To address this, we acquaint a peculiar code transformation approach for software protection. The proposed approach is used semantically equivalent to code clone within the source code to protect logical part of program text. We have successfully implement our approach using open source java project Gantt project system and open source java obfuscator’s tools. In this paper, we present our approach and demonstrate it with an illustration. The intent of this method is to prevent static analysis attack and make dynamic attack compact for an adversary. This makes it worthwhile against reverse engineering attacks.

KEYWORDS
Code Clone, Obfuscation, Reverse Engineering, Software Security, Tampering, Watermarking

INTRODUCTION
Software piracy and its unauthorized adaptations is one of the emerging concerns in computer security. However, substantial endeavors have been made to enhance the security of the computer systems. Software security plays a vital role in IT industry as its breach leads to huge financial losses. There are several security attacks such as piracy, reverse engineering and tampering which exploits...
inadequately protected software. Moreover, several techniques have been proposed for software protection from various attacks. Hence, it is essential to develop an approach which protects software from threat analysis and unauthorized modifications. Software security is a concept of engineering software in such a way that it continues to function correctly under malicious attack. In addition to this, stenography, which is a branch of cryptography, is another significant facet of software protection and it explores how to transmit data surreptitiously. Software security errors can be assorted as coding errors in source code and copy right protection of software code. The intellectual property protection is a two-player game between two adversaries: A (software developer) whose aim is to keep program secure from assault, and B (reverse engineer) who analyzes the code and adapt it into a form which is easy to peruse and get it. The adaptation which is accomplished by B is not necessarily similar to A’s source code. It is essential that the reverse engineered code should be comprehensible by B. However, it may not be possible for A to protect the whole program from B (Collberg et al., 1997). The intellectual property attacks (Collberg & Thomborson, 2002) can be categorized as 1) software piracy (unauthorized distribution, replicate and resale of software without legal rights), 2) reverse engineering (analyzing software) and 3) tampering (unauthorized modifications/altering). The protection methods against these attacks are watermarking, obfuscation and tamper-proofing. The obfuscating transformation makes the program harder without changing its functionality for an adversary. Obfuscation based attacks can be characterized as static analysis, dynamic analysis and code clone attacks (Kulkarni & Metta, 2014).

Obfuscation means an efficient semantic-preserving transformation of a program P into a form which thwart comprehensibility of the code and produce a new program T(P'). An obfuscated program is semantically equivalent as original program P (Barak et al., 2012; Varnovsky et al., 2003; Lynn et al., 2004; Goldwasser et al., 2007). Obfuscation can be categorized into two types as: 1) static obfuscation in which transformed code remains persist at run time, and 2) dynamic obfuscation, to obfuscate programs constantly at runtime, keeping them in constant transmutation which tries to impede dynamic analysis (Collberg & Nagra, 2009; Madou et al., 2005).

In this paper, we have proposed a novel obfuscation approach for software protection as well as a comprehensive review on code obfuscation. We will start with the basic introduction of obfuscation after that we will classify the techniques and tools in two different ways. Foremost, the basic concept of obfuscation, property and their pros, cons and subsequent proposed approach. The remaining of the paper is organized as follows. In section-II, we entail the obfuscation concepts, their property and the taxonomy of obfuscation. The section-III is related to several reverse engineering attacks and their protection approaches. Section-IV puts its eyes on classification of obfuscation. Section-V is depicting the detailed literature with identified research gaps about the topic. The proposed method and its implementation is explored in section-VI. The results are discussed in section-VII. Finally, the section-VIII concludes the paper.

**MERITS AND PROPERTY OF OBFUSCATION**

An obfuscation transformation can be evaluated on the basis of some properties which are presented by Collberg et al. (Collberg et al., 1997; Gregory, 2002; Kulkarni & Metta, 2014). In Figure 1, we have discussed several properties of obfuscation.

Figure 1 (a) shows various obfuscation properties such as: potency is the extent to which the transformed program obscured to the reader. Resilience, the amplitude to which a transformation T can resist an automated de-obfuscation. Cost, performance overhead added due to transformation (Kulkarni & Metta, 2014; Gregory, 2002). Stealth, in which transformed code can be characterized
Related Content

Modeling and Verification of Cooperation Incentive Mechanisms in User-Centric Wireless Communications
www.igi-global.com/chapter/modeling-and-verification-of-cooperation-incentive-mechanisms-in-user-centric-wireless-communications/86323?camid=4v1a

A Study of Recursive Techniques for Robust Identification of Time-Varying Electrical Equivalent Circuit Models of Li-Ion Batteries
www.igi-global.com/article/a-study-of-recursive-techniques-for-robust-identification-of-time-varying-electrical-equivalent-circuit-models-of-li-ion-batteries/196259?camid=4v1a
Security Model of Internet of Things Based on Binary Wavelet and Sparse Neural Network

Towards A Virtual Machine Migration Algorithm Based On Multi-Objective Optimization
www.igi-global.com/article/towards-a-virtual-machine-migration-algorithm-based-on-multi-objective-optimization/188625?camid=4v1a