Department of Computer Science
The University of Auckland
New Zealand

Design and Evaluation of Software Obfuscations

Anirban Majumdar

2008

Supervisors:

Prof. Clark D. Thomborson

Dr. Stephen J. Drape

A thesis submitted in partial fulfilment of
the requirements of Doctor of Philosophy
in Computer Science

The University of Auckland

Thesis Consent Form

This thesis may be consulted for the purpose of research or private study provided that due acknowledgement is made where appropriate and that the author's permission is obtained before any material from the thesis is published.

I agree that the University of Auckland Library may make a copy of this thesis for supply to the collection of another prescribed library on request from that Library; and

1. I agree that this thesis may be photocopied for supply to any person in accordance with the provisions of Section 56 of the Copyright Act 1994.

   Or

2. ~~This thesis may not be photocopied other than to supply a copy for the collection of another prescribed library.~~

   (*Strike out 1 or 2*)

Signed: .....

Date: .....

Created: 5 July 2001

Last updated: 30 September 2007

Abstract

Software obfuscation is a protection technique for making code unintelligible to automated program comprehension and analysis tools. It works by performing semantic preserving transformations such that the difficulty of automatically extracting the computational logic out of code is increased. Obfuscating transforms in existing literature have been designed with the ambitious goal of being resilient against all possible reverse engineering attacks. Even though some of the constructions are based on intractable computational problems, we do not know, in practice, how to generate hard instances of obfuscated problems such that all forms of program analyses would fail.

In this thesis, we address the problem of software protection by developing a weaker notion of obfuscation under which it is not required to guarantee an absolute blackbox security. Using this notion, we develop provably-correct obfuscating transforms using dependencies existing within program structures and indeterminacies in communication characteristics between programs in a distributed computing environment. We show how several well known static analysis tools can be used for reverse engineering obfuscating transforms that derive resilience from computationally hard problems. In particular, we restrict ourselves to one common and potent static analysis tool, the static slicer, and use it as our attack tool. We show the use of derived software engineering metrics to indicate the degree of success or failure of a slicer attack on a piece of obfuscated code. We address the issue of proving correctness of obfuscating transforms by adapting existing proof techniques for functional program refinement and communicating sequential processes.

The results of this thesis could be used for future work in two ways: first, future researchers may extend our proposed techniques to design obfuscations using a wider range of dependencies that exist between dynamic program structures. Our restricted attack model using one static analysis tool can also be relaxed and obfuscations capable of withstanding a broader class of static and dynamic analysis attacks could be developed based on the same principles. Secondly, our obfuscatory strength evaluation techniques could guide anti-malware researchers in the development of tools to detect obfuscated strains of polymorphic viruses.

Acknowledgements

This thesis has benefitted immensely from the wisdom and advice of many individuals. This is a good opportunity to thank them all.

First and foremost, I am deeply indebted to my supervisors Professor Clark Thomborson and Dr. Stephen Drape for mentoring me towards a completion in the minimum allowable time. Over the last three years, Professor Thomborson has given me both the freedom to pursue my own research interests and the insightful guidance that I have needed. Through him I learnt the art of lateral thinking and the use of rigour in scientific research. I am grateful to him for funding my research through the New Zealand government's New Economy Research Foundation fund on "Securing Software for New Zealand". I am extremely fortunate to have had Dr. Drape as my co-supervisor over the period of last eighteen months. Dr. Drape has not only been an excellent thesis advisor but also a valued friend. During the troubled times of funding uncertainty, he has taught me to persevere and remain focussed on my research. Our mutual admiration for progressive rock and Ricky Gervais resulted in some memorable time spent at rock concerts and in front screens appreciating countless episodes of The Office. Dr. Drape has been instrumental in introducing me to programming language theory and I am particularly indebted for his help in adapting a framework for proving correctness of imperative transforms. I am thankful to Professor Thomborson for his help in shaping up the final thesis draft and to Dr. Drape for taking time off of his busy schedule at Oxford University to proofread my thesis drafts.

I have benefitted immensely from interactions with the former members of the Secure Systems Group. I grateful to Dr. Antoine Monsifrot, of Thomson R & D France, for his collaborative efforts on static analysis experiments with BDDBDDB and Indus. Dr. Jasvir Nagra, of University of Trento, has been my peer mentor all these years and have enlightened me with his mastery of two dozen programming languages and analytical skills. Michael Stay, of Google Mountain View Labs, fascinated me with his alternative view of distributed opaque predicates using the cryptographic theory of multi-party computations. I have thoroughly enjoyed the intriguing discussions on computer forensics and technorealism with Dr. Simson Garfinkel of Harvard University. I am thankful to my former mentors Professor Hon Fung Li of Concordia University, Canada, Professor Subhansu Bandopadhyay and Dr. Nabendu Chaki of Calcutta University, India for their unvarying impetus that catalysed my yearning to take up research in computer science as my professional career. I am thankful to my school and university friends in India and abroad for providing me with encouragement and entertainment right from the beginning. I am thankful to Dr. David Pearce, my oral examiner, for his help in improving the overall quality of the thesis with his detailed suggestions.

My greatest gratitude goes to my parents for their unconditional support during the course of my PhD. My wonderful Ma has nurtured me with love and care all my life.

She has been the source of limitless encouragement during these difficult years. My dad motivated my inquisitiveness from very early on and supported my career decisions no matter where they took me. Being a scientist himself, my dad provided immensely valuable suggestions at critical junctures of my research and thesis writing. I am thankful to my sister for helping me choose a career in computer science over medicine right after my high school days. I thank Sikharini Majumdar for her encouraging phone conversations and for generously sharing the plight of fellow PhD students across the Pacific. She made me feel that I am not alone, after all. Finally, I am thankful to my wife Raka; her unwavering patience and encouragement during these trying times have made it all worth it. I could not have envisioned this personal journey without her. It is to Raka that I dedicate this thesis.

Contents

List of Figures

List of Tables