AIN SHAMS UNIVERSITY
FACULTY OF COMPUTER
& INFORMATION SCIENCES
Abbassia, Cairo, Egypt.

Developing a Software Development and Analysis Framework using Runtime Monitoring

A thesis submitted to the Department of Information Systems Faculty of Computer and Information Sciences Ain Shams University, Egypt

In partial fulfillment of the requirements for the degree of Doctor of Philosophy, PhD, of Computer and Information Sciences

BY

Soha M Nabil AbdelHamid Hussein
Assistant lecturer in the Department of Information Systems Faculty of Computer & Information Sciences Ain Shams University, Egypt

Under the supervision of

Prof. Dr. Mohamed Essam Khalifa
Ex-Dean of the Faculty of Computer & Information Sciences, Ain Shams University, Egypt

Prof. Dr. Mohamed Hashem Abd El Aziz
Vise Dean for Student Affairs Faculty of Computer & Information Sciences, Ain Shams University,Egypt

Prof. Dr. Grigore Roşu
Associate Professor in the Department of Computer Science, University of Illinois at Urbana-Champaign, USA

Acknowledgments

This work would not have been completed without the support of many people to whom I wish to express my deep gratitude.

My deepest gratitude goes to my supervisors, Prof. Mohamed Essam Khalifa and Prof. Mohamed Hashim, who were abundantly supportive and helpful, and who were always there for me with their guidance treating me like a daughter.

I will also always be in debit to my supervisor Prof. Grigore Roşu for having me for 2 years in UIUC where he has continuously guided me with his knowledge and with his valuable support, not to mention the generous hospitality of his and his family throughout my visit. Also, I would like to thank my colleagues in the MOP team, especially Patrick Meredith and Dongyun Jin for their cooperation and for their notes on some parts of this work.

I would also like to convey thanks to the Faculty of Computer and Information Systems and especially the department of Information Systems, for supporting my joint research grant during my 2 years visit to the United States.

I also want to thank Prof. Dr. Mohamed Kouta who passed away a couple of months before I returned to Egypt. I consider him my God Father, I wish he lived to see that moment. Goes to him my deep gratitude and may his soul rest in peace.

Also, i would like to express my love and gratitude to my beloved parents, Prof. Mohamed Nabil and Prof. Asmaa AbdelMonium, for being there for me all the way through, and for giving me a model to look at on both the academic and personal levels.

Also I would like to thank my brothers, Dr. Haitham Hussein and Eng. Shady Hussein, for being the best brothers one could ever dream of having.

A special thanks goes to my daughter, Fairouz Otefy, for being there with me during my research and for being the joy of my life. I believe that she was the motivation for me to make it through. I also would like to thank my husband, Eng. Essam Otefy, for doing the best he could.

Finally, I would like to thank God for all his blessings, generosity, forgiveness, for giving me chances, and for being there for me when it was absolutely hard to go on.

Abstract

For a long time, *runtime verification RV* has been used to formalize system's properties for testing and verification, but (as far as our knowledge is) it hasn't been used to enforce security properties about systems. On the other hand there exists tailored runtime security enforcement systems (usually called *execution monitor*-based systems abbreviated as EM-based systems) that are specialized to specify and enforce only security properties during runtime.

In this thesis we do the following:

First, since runtime verification and execution monitors are completely two separate research disciplines, we carefully try to understand features where both disciplines overlap and where they do not. We go one step further by actually using an RV system to do the job of an EM system. This required the addition of a new feature (around join point) to our RV system as well as providing ways for policy composition and ensuring correctness on the generated monitored code. Our point here is that with slight additions on RV-based systems, RV-based systems can be generic enough to include features of any EM-based system, while still enjoying other features that comes with RV-based systems.

Second, we present a formal model that models specification monitors (which are used to enforce system's properties) as *finite state transducers*. Our formal model intends to solve problem of conflicts among different monitors that are co-existing at the same time. We show how the formal model can be used to detect potential conflicts among different monitors and to resolve them.

Third, we extend JavaMOP language (the main generic runtime verification and monitoring system used throughout the thesis) to support extending of specifications from one another; where we supported the addition of new syntax to the JavaMOP language, created a parser with its extended abstract syntax tree. Such an extension allows JavaMOP users to overload and/or override elements of a super specification, i.e., local variables, events properties and handlers, written in JavaMOP. Also we introduced 2 new pointcuts to JavaMOP, *event pointcut* and the *handler pointcut*, to allow specifications to pointcut other specification's events as well as their handlers.

Finally, we measure the performance overhead of the instrumented programs using execution-monitors based systems and those instrumented using a runtime verification and monitoring system via specialized programs and also using the DaCapo benchmark.

Generally JavaMOP showed relatively low performance overhead of the instrumented program as opposed to other EM-based systems. That is in addition to

its expressive power in expressing different complex security specifications and the ability to resolve potential conflicts among them after extending the JavaMOP language.

Keywords: Runtime Verification and Monitoring, MOP, Execution Monitors, Inlined Reference Monitors, Security Access Control Policies, Policies Conflicts Detection and Resolution, Finite State Transducers, Extending Specifications.

Contents

List of Figures

List of Tables

Listings