Available online at www.sciencedirect.com

SciVerse ScienceDirect

journal homepage: www.elsevier.com/locate/cosrev

Survey

Which security policies are enforceable by runtime monitors? A survey

Raphaël Khoury[*,1], *Nadia Tawbi*

Département d'informatique et de génie logiciel, Université Laval, Québec, Canada

ARTICLE INFO

Article history:
Received 11 January 2011
Received in revised form
4 January 2012
Accepted 4 January 2012

Keywords:
Computer security
Runtime monitoring
Dynamic analysis
Security policies

ABSTRACT

Runtime monitoring is a widely used approach to ensure code safety. Several implementations of formal monitors have been proposed in the literature, and these differ with respect to the set of security policies that they are capable of enforcing. In this survey, we examine the evolution of knowledge regarding the issue of precisely which security policies monitors are capable of enforcing. We identify three stages in this evolution. In the first stage, we discuss initial limits on the set of enforceable properties and various ways in which this set can be extended. The second stage presents studies that identify constraints to the enforcement power of monitors. In the third stage, we present a final series of studies that suggest various alternative definitions of enforcement, which specify both the set of properties the monitors can enforce as well as the manner by which this enforcement is provided.

© 2012 Elsevier Inc. All rights reserved.

Contents

* *Corresponding author.* Tel.: +1 418 653 0177.
E-mail addresses: raphael.khoury.1@ulaval.ca (R. Khoury), nadia.tawbi@ift.ulaval.ca (N. Tawbi).
[1] Present address: Defence Research and Development Canada, Canadian Department of National Defence, Valcartier, G1W 3J7, Québec, Canada.

1574-0137/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
doi:10.1016/j.cosrev.2012.01.001

1. Introduction

As security concerns become increasingly central in our everyday interaction with complex systems, one solution that is rapidly gaining wide acceptance to address these concerns is run time monitoring, an approach to code safety that permits the execution of untrusted code by observing its execution and reacting as needed to avoid a violation of a security policy. This enforcement method has accordingly been the subject of a number of academic studies. Although these studies have explored a number of different topics related to the enforcement of security policies by monitors, one question seems to recur frequently in most if not all the works: exactly which set of properties are *monitorable*, in the sense that they are enforceable by monitors operating under different sets of constraints[2]? These are the policies for which a given monitor is capable of ensuring that a violation does not occur while operating under certain limiting constraints. This seemingly straightforward question does not admit a simple answer and has been the topic of numerous papers with conclusions that sometimes seem in conflict with each other. Yet determining in what context a given security policy becomes enforceable is central to the selection of the enforcement mechanism and even to the design of the security policy itself.

In this survey, we examine the various answers that have been proposed in the literature. As we will show, the set of properties enforceable by monitors is highly contingent on a number of factors, namely the availability of data about the target program's possible behavior, the means at the disposal of the monitor to react to a potential violation, memory and computability constraints and the definition of *enforcement* being used. A precise knowledge of these issues is essential to adequately select the best enforcement mechanism that can guarantee the respect of a given security policy. Furthermore, when the desired security policy falls outside the range of policies enforceable by the preferred enforcement mechanism, the research discussed in this survey can suggest ways to extend the mechanism's limits.

In our view, the evolution of knowledge with respect to the limits of the enforcement power of monitors can be subdivided into three stages of development. In the first stage, studies delineated the set of enforceable properties and suggested various ways this set could be extended. Researchers in this stage proposed increasingly powerful conceptual models of monitors culminating with the edit automaton, a very flexible model that can be used to capture the behavior of any runtime enforcement mechanism. Second stage studies, in contrast, dealt with constraints which limit the enforcement power of monitors. Research in this stage introduced memory and computability constraints into the models that had been previously proposed. Third stage studies take issue with the definition of enforcement used in the works of the previous two stages and propose alternative definitions to constrain the monitor's behavior further and capture restrictions on the monitors behavior when it is faced with a possible violation of the security policy. These latter studies ask not only *if* a given policy can be enforced by monitor, but also *how* this enforcement would take place, adding to the enforcement paradigm important limits on its permissible behavior, which are necessary to provide meaningful enforcement but where absent from previous models.

We begin in Section 2 by rigorously defining the notions of executions, security policy, and monitor that we will manipulate. In Section 3, we examine the first stage of studies, which show how these definitions allow us to state the limitations of a rudimentary monitor. From these limitations, the set of security properties enforceable by this monitor can be deduced. We then examine various ways to enhance the power of monitors by giving in each case a new formal definition of the extended monitor and identify the most promising model. We next turn to the enforcement power of those monitors found to be the most powerful and give a lower bound to the set of properties they can enforce. In Section 4 we examine the second stage of studies that focus on how computational and memory constraints can affect the set of enforceable properties. Finally, in Section 5, we revisit the notion of enforcement and propose alternative definitions put forth in the third stage of studies. Concluding remarks are given in Section 6.

2. Security policies and security properties

The first step in our analysis is to define a theoretical framework that can accommodate the various concepts we elaborate, such as executions, security policies, and monitors. To this end we adopt the formal notation devised in [3] which is widely used in the field.

An execution σ of a program, or trace, is modeled as a finite or infinite sequence of atomic program actions:

$$\sigma = a_0, a_1, a_2, \ldots.$$

We let a range over a finite or countably infinite set of atomic actions Σ. The empty sequence is noted ϵ, the set of all finite length sequences is noted Σ^*, that of all infinite length sequences is noted Σ^ω, and the set of all possible sequences

[2] In this study, we are uniquely interested with *runtime enforcement*, which refers to techniques that prevent violations of the security property. Monitors can additionally be used for runtime verification, which simply aims to detect the violation. An examination of the set of properties that are monitorable in the latter case falls outside the scope of this study. The interested reader is referred to [1,2].

Download English Version:

https://daneshyari.com/en/article/471094

Download Persian Version:

https://daneshyari.com/article/471094

Daneshyari.com