EasyChair Preprint

№ 9375

Virtual Machine Constraints of Storage Capacity of CPU

Aiman Haqanni and Fatima Tahir

December 11, 2022

Virtual Machine Constraints of Storage Capacity of CPU

Aiman Haqanni, Fatima Tahir

Abstract

Currently, customers can express basic business needs, such as deploying applications and hosting virtual machines (VM), with constraints on storage capacity, memory, and CPU speed. However, the architectures of information systems are becoming more and more complex. Customers' cloud computing professionals must be able to customize service models and demand more complex business needs. In addition, customers are interested much to safety criteria. They wish to express fine security needs to protect their information systems, applications, and platforms that tend to be complex and highly personalized.

Keywords: Cloud Computing, Virtual Machine

1. Introduction

First, the proposed solution must make it possible to express the security needs of a software architecture hosted in a cloud infrastructure[1]. As the Cloud is a heterogeneous environment, the solution must be system independent to which the policy applies. Moreover, to cover a wide spectrum of techniques and security needs, we propose to reuse existing security mechanisms since there are currently many of them, each specializing in applying a subset of properties[2]. In addition, the various secured machines do not necessarily have the exact security mechanisms: the solution must abstract the mechanisms so that security needs can be expressed without depending on them explicitly. The second objective of this thesis concerns updating the protection throughout machine life. Indeed, the solution must be able to detect changes producing on the system and harming the proper application of the needs (unavailability of a mechanism, problem when

applying a property, etc.). When such an event is detected, the proposed solution must also be able to react in order to continue to meet security needs. Finally, the solution must propose an evaluation method the quality of the application, in order to be able to easily compare the application of the same policy on different machines or on a single machine over time.

The thesis provides an implementation of this architecture. This implementation is able to interpret all of the proposed language in order to apply and ensure a policy. It can also perform the automatic reconfiguration and evaluation phases. It has been tested as part of the Seed4C project on several industrial use cases as well as only in an experiment presented in this document.

II. Related Work

Security properties are the basis for expressing security needs. The set of security properties is commonly seen as a set derived from three main properties: confidentiality, integrity and avail ability CIA: Confidentiality, Integrity, Availability)[3]. The exact interpretation of what these three properties imply varies according to the context of use, but their definition and application are part essential part of the security evaluation criteria, at the European level and international . Several definitions of these properties exist in the literature we resent here a synthesis.

Historical models

Some security properties, previously defined, can be applied by access control mechanisms. In this section, we therefore detail models of historical access

control, which introduced the concepts subsequently taken up by various security policy templates. An access control system is usually modeled using the following three elements: – a set of subjects which are the active entities of the system (for example, the process) ; – a set of objects which are the passive entities of the system, on which the subjects can perform actions (files, sockets, etc.); – a set of permissions that represent the actions allowed between a subject and an object (reading, writing, etc.), or between two subjects (sending a signal). Discretionary Access Control Discretionary Access Control (DAC) is the historical model present by default on the majority of Operating systems. In this model, the management of access rights to a resource is left to the discretion of the owner of this resource. For example, under Unix, the owner of a file can set read, write and execute permissions for himself, for members of the group owning the file, and for all others system users. An access control model can be represented as a matrix, where a line represents a topic, a column represents an object or topic, and each item in the matrix represents a set of subject permissions on the object (or on the second subject). This model was formalized by using the Capability Lists and Access Control Lists (ACLs). He proposes therefore to indicate, in a matrix A, the set D of protection domains (representing program execution contexts, i.e. subjects) on the lines, and the set X of objects on the columns. Lampson therefore defines the lists of capabilities which establish the permissions of a domain d on the set of objects o of the system. It is therefore the set of actions authorized for each domain.

Conclusion:

Cloud computing is an increasingly heterogeneous and dynamic type of environment.

additionally used. Different service and deployment models exist and allow to meet a variety of user needs. The combination of an environment heterogeneous and numerous user applications makes security an essential point but complex to address. Defining security needs can indeed be a task difficult, especially since the user of the service does not necessarily know these needs. However, we have seen that there are risk analysis methods that allow a user to determine his security needs. Thus, in the remainder of this document, we will consider that the user of the service is able to establish the list of his needs, if necessary using one of these methods.

Reference:

[1] I. Foster, Y. Zhao, I. Raicu, and S. Lu, "Cloud computing and grid computing 360-degree compared," in *Grid Computing Environments Workshop, 2008. GCE'08*, 2008: Ieee, pp. 1-10.

[2] C. Binnig, D. Kossmann, T. Kraska, and S. Loesing, "How is the weather tomorrow?: towards a benchmark for the cloud," in *Proceedings of the Second International Workshop on Testing Database Systems*, 2009: ACM, p. 9.

[3] R. Chow *et al.*, "Controlling data in the cloud: outsourcing computation without outsourcing control," in *Proceedings of the 2009 ACM workshop on Cloud computing security*, 2009: ACM, pp. 85-90.