COMPUTERS & SECURITY 47 (2014) 26–40

Available online at www.sciencedirect.com

ScienceDirect

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

Computers & Security

Evaluating and comparing the quality of access control in different operating systems

*Liang Cheng**, *Yang Zhang*, *Zhihui Han*, *Yi Deng*, *Xiaoshan Sun*, *Dengguo Feng*

Trusted Computing and Information Assurance Laboratory, Institute of Software, Chinese Academy of Sciences, China

ARTICLE INFO

Article history:
Received 28 December 2013
Received in revised form
4 April 2014
Accepted 3 May 2014
Available online 13 May 2014

Keywords:
Security measurement
Vulnerability profile
Attack surface
Access control
Operating system
Logic programming

ABSTRACT

Access control mechanisms (ACMs) have been widely used by operating systems (OSes) to protect information security. However, it is often challenging to evaluate and compare the quality of protection (QoP) of ACMs, especially when they are deployed on different OS platforms. This article presents an approach to quantitatively measure and compare the quality of ACMs, which provides useful information to support OS administrators and users to choose ACMs that fit with their security needs.

We introduce the notion of vulnerability profiles to capture the weakness of ACMs in protecting against malicious attacks, based on which vulnerability coefficients are computed as the numeric and platform-independent measurement of the QoP of ACMs. The approach combines the grey system theory and an independent vulnerability scoring system to infer complete vulnerability profiles and to calculate fair and objective vulnerability coefficients for ACMs. We implement a prototype called ACVAL based on the approach, and apply it to four mainstream ACMs. The results show that ACVAL is effective in evaluating and comparing ACMs across different OSes, a feature particularly useful to administrators of heterogeneous IT systems. To the best of our knowledge, our approach is the first to quantitative measurement and comparison of ACMs across OSes.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

Access control is one of the most widely adopted security enhancement measures in modern information systems. Till date, various access control mechanisms (ACMs) have been proposed to protect against viruses or Trojan horses. For example, Discretionary Access Control (DAC) has been used by Windows and Linux systems for years. Recent years also saw the adoption of Mandatory Access Control (MAC) and its variants in most Linux distributions and latest versions of Windows systems. These invariants include Security Enhanced Linux (SELinux) (NSA, 2001), AppArmor (N. Corp, 2002), and Mandatory Integrity Control (MIC).

However, it is often overwhelmingly complex for average users to choose one ACM that adequately addresses their security needs, a task even challenging for IT experts. A primary reason for this is the lack of measures that can help people assess and compare, in an objective and fair manner, different ACMs available to them. A more challenging task facing system administrators is that they have to ensure security in their organizations' heterogeneous IT infrastructures, in which numerous ACMs are deployed on different operating

* *Corresponding author*. Rm809 Bldg 5, 4# South Fourth Street, Zhongguancun, Beijing 100190, China. Tel.: +86 010 62661721.
E-mail addresses: chengliang@tca.iscas.ac.cn (L. Cheng), zhangyang@tca.iscas.ac.cn (Y. Zhang), hanzhihui@tca.iscas.ac.cn (Z. Han), dengyi@tca.iscas.ac.cn (Y. Deng), sunxs@tca.iscas.ac.cn (X. Sun), feng@tca.iscas.ac.cn (D. Feng).
http://dx.doi.org/10.1016/j.cose.2014.05.001
0167-4048/© 2014 Elsevier Ltd. All rights reserved.

systems (OSes). It is thus critical for them to be able to compare the quality of protection (QoP) of ACMs, even across different platforms.

Unfortunately, only few previous research efforts (such as Govindavajhala and Appel, 2006; Chen et al., 2009) provided solutions for evaluating the QoP of certain ACMs in certain OSes, and none of them is quantitative. Thus, it remains difficult for average users, who typically have no profound knowledge on information technology or security, to understand the evaluation results. Even worse, no previous research has addressed the problem of comparing ACMs across different OSes.

In this paper, we present an approach to quantitative evaluation and cross-platform comparison of the QoP of ACMs. Since different OSes have significant differences in security context and semantics, it is usually infeasible to directly compare two ACMs if they are deployed in different OSes. To assure fair comparison, our approach evaluates the QoP of arbitrary ACMs (probably deployed on different OSes) against the same types of attacks. Given a specific type of attacks and an ACM, our approach first infers a vulnerability profile from the OS protected by the ACM. A vulnerability profile articulates actions taken by the adversary to realize this type of attacks, as well as the unintended privileges he/she acquires.

Based upon the inferred vulnerability profile, our approach then computes a vulnerability coefficient of the profile, which provides a numeric yet informative indication of the protection quality of the ACM under concern. More specifically, the calculation of the vulnerability coefficient takes in account both the potential damage that actions in the vulnerability profile may cause to the system's security, and the effort required from the adversary to take these actions. Thus, a higher vulnerability coefficient means that the ACM either lows down the potential security consequences or leverages the difficulty for the adversary to succeed.

Three technical difficulties may threaten the validity of our approach. First, it is often difficult to infer a complete vulnerability profile. Our approach applies the grey system theory (Liu and Forrest, 2010) to address this difficulty, in which how attacks can succeed in an OS is formalized as properties of a grey system (i.e., it has not been fully understood). Thus, inferring the vulnerability profile is accomplished as finding a solution to the grey system, based on its properties. The second difficulty is to assure a fair comparison of ACMs running on different OSes. To address this difficulty, our approach not only makes the evaluation of ACMs as attack-oriented, but also relies upon the Grey Relational Analysis (GRA) to calculate the vulnerability coefficients of different ACMs when they face the same types of attacks. In this way, one ACM is not compared to another, but to the ideal situation in which the attacks are absolutely denied. Lastly, it is difficult to assign objective and reasonable scores to the potential damage and effort of actions involved in the inferred vulnerability profiles. We address this difficulty by using the Common Vulnerability Scoring System (CVSS), which is widely accepted by independent database of security vulnerabilities, to decide the (numeric) potential damage and effort of each encountered action. However, our approach can also be used with o any other vulnerability scoring mechanisms.

We implement a prototype tool, called ACVAL (Access Control eVALuation tool), based on our approach. ACVAL applies the logic programming language Prolog to encode information relevant to ACM evaluation and then uses its built-in inference engine to infer vulnerability profiles. The calculation of vulnerability coefficients is accomplished by programs written in various languages, depending on which OS is considered. We have applied ACVAL to four mainstream ACMs: DAC (running on Windows XP and 7), AppArmor and SELinux (on Ubuntu and Fedora) and MIC (on Windows 7). Results show that ACVAL is effective in quantitatively measuring the QoP of these ACMs.

To our knowledge, our approach is the first attempt to quantitatively evaluate and compare the QoP of ACMs in different platforms. The main contributions of our work can be summarized as follows:

- We propose a precise definition of vulnerability coefficient as a numeric and OS-independent indication of the ability of an ACM protecting against malicious attacks.
- We implement a prototype ACVAL, that bases on logic programming and GRA to automatically infer possible attack patterns that expose OSes to given attack scenarios and to quantitatively calculate the QoP of ACMs.
- We use ACVAL to study the QoP of various mainstream ACMs under two common attack scenarios and to automatically calculate the vulnerability coefficients of these ACMs.

The rest of the paper is organized as follows: Section 2 offers some background information; Section 3 presents an overview of our approach, followed by Section 4 that describes the implementation of ACVAL; Section 5 discusses the experiment of evaluating ACVAL and Section 6 compares our work with previous research. Lastly, Section 7 concludes the paper.

2. Background

DAC and MAC are two ACMs widely used in current IT systems. Their major difference lies in whether or not they allow users to change access permissions to IT assets. While DAC permits the owner to change the access settings of an IT asset, MAC enforces system-wide access control policies that cannot be overridden by any user. Modern desktop OSes often employ both of them simultaneously.

Microsoft introduced MIC as its MAC implementation since Windows Vista, based on its traditional DAC model. Windows' DAC assigns each user (or group) with a unique *Security Identifier* (SID), each process with an *Access Token*, and each IT asset (also called object) with a *security descriptor*. The Access Token of a process consists of the SID of its user, the groups that the user belongs to, and the privileges held by the user and these groups. The security descriptor of an object is a tuple of the SID of the object's owner, its discretionary access control list (DACL), and its system access control list (SACL). A DACL essentially compiles a list of access control entries (ACEs), each of which either grants or denies a set of access rights to a particular SID. On the other hand, MIC assigns an *Integrity SID* (ISID), which determines the level of access the token can achieve, to the user's access token. It also stores an ISID in

Download English Version:

https://daneshyari.com/en/article/455865

Download Persian Version:

https://daneshyari.com/article/455865

Daneshyari.com