Protection of Components Based on a Smart-Card Enhanced Security Module

Joaquín García-Alfaro[1], Sergio Castillo[1], Jordi Castellà-Roca[2], Guillermo Navarro[1], and Joan Borrell[1]

[1] DEIC-UAB, 08193 Bellaterra (Catalonia), Spain
{jgarcia,scastillo,gnavarro,jborrell}@deic.uab.es
[2] DEiM-ETSE-URV, 43007 Tarragona (Catalonia), Spain
jordi.castella@urv.net

Abstract. We present in this paper the use of a security mechanism to handle the protection of network security components, such as *Firewalls* and *Intrusion Detection Systems*. Our approach consists of a kernel-based access control method which intercepts and cancels forbidden system calls launched by a potential remote attacker. This way, even if the attacker gains administration permissions, she will not achieve her purpose. To solve the administration constraints of our approach, we use a smart-card based authentication mechanism for ensuring the administrator's identity. Through the use of a cryptographic protocol, the protection mechanism verifies administrator's actions before holding her the indispensable privileges to manipulate a component. Otherwise, the access control enforcement will come to its normal operation. We also show in this paper an overview of the implementation of this mechanism on a research prototype, developed for GNU/Linux systems, over the *Linux Security Modules* (LSM) framework.

1 Introduction

The protection of network security components, such as *Firewalls* and *Intrusion Detection Systems*, is a serious and important problem which must be solved. Otherwise, whenever a remote adversary manages to compromise the security of these components, she may obtain the control of the system itself. Contrary to many other elements of a network, security components are almost always working with special privileges to properly execute their tasks [6]. This situation is very likely to lead remote attackers to acquire these privileges in an unauthorized manner. For instance, the existence of programming errors within the code of these components, the illicit manipulation of their related resources (such as processes, filesystem, and so on), or even the increase of privileges though operating system's errors, are just a few examples regarding means in which a remote adversary can bypass traditional security policy controls.

In [4,5] we presented an enhanced protection module integrated into the kernel of an attack prevention system intended to intercept and cancel forbidden system calls launched by a remote attacker. Specifically, the mechanism presented in [4,5] prevents a privilege escalation attack on the prevention system itself – through an enhanced access control scheme which handles the protection of the system's elements. This strategy introduces, however, some administration constraints, since the administrators are not

J. Lopez (Ed.): CRITIS 2006, LNCS 4347, pp. 128–139, 2006.

able to throw system calls which may suppose a threat to the protected system. To solve these constraints, we present in this paper an extended version of our approach which includes a smart-card based authentication mechanism, which acts as a reinforcement of the kernel-based access control. The objective of this complementary mechanism is twofold. First, it holds to the administrator the indispensable privileges to carry management and configuration activities just when she verifies her identity through a two-factor authentication mechanism. Second, it allows us to avoid those attacks focused on getting the rights of the administrative entity, such as dictionary-based attacks or buffer overflows.

The rest of this paper is organized as follows. Section 2 summarizes some related works. Section 3 shows an overview of our protection strategy. Section 4 takes a closer look at the development of the proposed mechanism. Section 5 presents our smart-card based authentication protocol intended to solve the administration constraints introduced by the protection mechanism. An evaluation concerning the efficiency of our proposal is then presented in Section 6. Finally, Section 7 closes the paper with a list of conclusions.

2 Related Work

There are two main approaches to safely execute processes with special privileges on modern operating systems. A first approach, as the one presented in this paper, is to apply a kernel-based access control to the outcoming system calls. A second approach is the creation of restricted environments, in which the processes will be executed and controlled outside the trusted system space.

Regarding the first approach, the proposals closest to ours are the protection mechanisms presented in [9] and [11] for the creation of enhanced access control mechanisms integrated in the kernel of the GNU/Linux operating system. The main goal behind these two proposals is to reinforce the complete system by controlling the system calls and ensuring which process or user does the system call and against what it will be done. The ability to control the access to the resources allows to protect the security components and to avoid that nobody (including an attacker with administrator privileges) can disable them.

Nevertheless, both approaches differ from ours in a number of ways. First, and to our best knowledge, neither [9] nor [11] do not address the management of administration constraints, as our proposal does through the two-factor authentication mechanism we present in Section 5. Second, our approach, entirely based on the *Linux Security Modules* (LSM) framework [13], guarantees the compatibility with previous applications and kernel modules without the necessity of modifications. However, both [9] and [11] require the rewriting of some features of the original Linux kernel to properly work. This situation may force to recompile existing code and/or modules in order to obtain the new security features. Although it exists a LSM-based prototype for the approach presented in [9], it does not seem to be actively maintained for the current Linux-2.6 kernel series.