Towards Usable Application-Oriented Access Controls: Qualitative Results from a Usability Study of SELinux, AppArmor and FBAC-LSM

Z. Cliffe Schreuders, Leeds Metropolitan University, UK
Tanya Jane McGill, Murdoch University, Australia
Christian Payne, Murdoch University, Australia

ABSTRACT

A number of security mechanisms are available for improving the security of systems by restricting the actions of individual programs to activities that are authorised. However, configuring these systems to enforce end users’ own security goals is often beyond their expertise. Little research has investigated the usability issues associated with application-oriented access controls. This paper presents the results of a qualitative analysis of user perceptions of the usability of three application-oriented security systems: SELinux, AppArmor, and FBAC-LSM. Qualitative analysis identified a number of factors that affect the usability of application-restriction mechanisms. These themes are used to compare the usability of the three systems studied, and it is proposed that these factors can be used to inform the design of new systems and development of existing ones. Changes to the three security systems are also proposed to address or mitigate specific usability issues that were identified.

Keywords: AppArmor, Application-Oriented Access Controls, FBAC-LSM, Functionality-Based Application Confinement (FBAC), HCI Security (HCISec), Sandboxing, Selinux, Usability

INTRODUCTION

End users face serious security risks related to processes maliciously misusing users’ authority. One of the largest threats to end users is flaws in applications such as PDF readers, media players, web browsers and email clients (Dhamankar, Dausin, Eisenbarth, & King, 2009). These vulnerabilities can inadvertently allow remote attackers to subvert the behaviour of programs in order to carry out malicious actions. Trojan horses, where malware poses as legitimate software and carries out malicious activities, are also a significant threat.

DOI: 10.4018/jisp.2012010104
Linux, like most operating systems, typically allows applications to act with all the authority of a user. The Linux discretionary access control (DAC) mechanism authorises processes to run with the full authority of the associated user, regardless of the trustworthi-
ness of programs. In the current threat climate this approach is inadequate as a sole access control measure; basing security decisions on the identity of the user does not protect against processes which act maliciously due to software vulnerabilities or malware.

The Linux Security Module (LSM) framework provides a means for security extensions to be incorporated into the Linux kernel (Wright, Cowan, Smalley, Morris, & Kroah-Hartman, 2002). Many of the LSMs that have been de-
veloped can address threats posed by malicious code, by restricting specific processes to autho-
risied actions. Examples of LSMs that can place restrictions on the activities of processes include SELinux (Smalley, Vance, & Salamon, 2001), AppArmor (previously known as SubDomain) (Cowan et al., 2000), TOMOYO (Harada, Ho-
rie, & Tanaka, 2004), and SMACK (Schaufler, 2008). However, as is typical for this class of security mechanism (DeWitt & Kuljis, 2006), these systems face usability challenges that can limit the practical benefit to end users.

A new model, known as the functionality-based application confinement (FBAC) model, has been designed to meet end user usability goals (Schreuders & Payne, 2008a). The model incorporates policy abstractions, known as functionali-
ties, that can model the privileges authorised to processes based on the high level features applications provide (Schreuders & Payne, 2008b). A Linux implementation of the FBAC model has been developed, known as FBAC-LSM (FBAC-LSM is free open source software available at: http://schreuders.org/ FBAC-LSM). The implementation also lever-
ges automation techniques, which the FBAC model is naturally suited to.

A study has been conducted to compare the usability of three different approaches to application restrictions: FBAC-LSM, and two of the most widely deployed Linux security ex-
tensions, SELinux and AppArmor (Schreuders, McGill, & Payne, 2011). The results showed that the functionality-based mechanism enabled end users to effectively control the privileges of their applications with far greater success than the widely used alternatives. In particular, policies created using FBAC-LSM were more likely to be enforced and exhibited significantly lower risk exposure, while not interfering with the ability of the application to perform its intended task (Schreuders et al., 2011). In order to further explore and understand the reasons for the usability differences between the three security systems, this paper presents the results of qualitative analysis of the participant feedback for each of the security systems. The qualitative analysis identified a number of emergent themes in participants’ comments. These themes indicate a number of factors that affect the usability of application-restriction mechanisms, and are likely to be responsible for the usability differences between the security systems studied. These results are then discussed and used to compare the usability of the three systems studied. The paper also proposes changes to all three systems to address or mitigate specific usability issues that were identified throughout the study.

BACKGROUND

To reduce the threat of the problems posed by the limitations of user-oriented access control, application confinement has become an active area of research (Berman, Bourassa, & Selberg, 1995; Cowan et al., 2000; Goldberg, Wagner, Thomas, & Brewer, 1996; Hallyn & Kearns, 2000; Harada et al., 2004; Ott, 2002; Provos, 2002; Smalley et al., 2001). Application-orient-
ed access controls provide restrictions based on the identity of the application or process rather than just the identity of the user. Application confinement can limit the ability of applications to access resources outside of those they require to perform legitimately, thus restricting the damage malware or exploited vulnerabilities can cause, and can limit software to actions that...
Effects of Digital Convergence on Social Engineering Attack Channels
www.igi-global.com/chapter/effects-digital-convergence-social-engineering/29050?camid=4v1a

Optimized Packet Filtering Honeypot with Snooping Agents in Intrusion Detection System for WLAN
www.igi-global.com/article/optimized-packet-filtering-honeypot-with-snooping-agents-in-intrusion-detection-system-for-wlan/190856?camid=4v1a