Formally Specifying Linux Protection

Osama A. Rayis

Sudan University of Science & Technology

rayis@sustech.edu

Abstract

Authorization and protection deal with the problem of the control of access to resources. A key aspect of modern computing systems is resource sharing, so a need arose to govern access to these resources only to authorized users. In multi-user operating systems (such as Linux) authorization is of great interest. Computer security and authorization as a subset is characterized by the fact that a security fault or hole can be very costly. It is of great interest therefore to formalize and reason about security. Z notation is a powerful well-known formal notation based on set theory and predicate calculus which provides both abstraction and formalism. This work reports a formal expression in the Z notation for the basic protection (authorization) system of the Linux operating system.

Key words:

Authorization, Linux Security, Access Control, Protection, Z notation, Formal Modeling, Reasoning.

I- INTRODUCTION

Modern operating systems like Linux came with many capabilities like multiple user support, remote access, networking, resource sharing and many other useful functions. Despite of its usefulness such facilities give rise to the risks of unauthorized access, thus processes and resources in operating systems must be protected (Johnson & Troan 2005), (Silberschatz *et al* 2002). Protection is an important quality for Linux to have and this quality should be rendered proven formally so that the operating system can be trusted. Proving system conformance by formal means (if feasible at all) is expensive and rarely cost-effective; one area in which it is cost-effective is computer security (Mclean 1990).

Authorization can be defined as the problem governing subject's accesses to objects according to some rights theses subjects have permission to perform on the objects, where subjects in this context can be any computing element. Lampson (1971) gave a model for authorization known now as access matrix model which was further refined and improved in (Graham & Denning 1972) and in (Harrison, *et al* 1976). The model expressed here follows Harrison *et al* (1976) model. The security kernel mechanism introduced by Schell is based on defining a small subset of the system to be responsible for the system's security and this subset would monitor all accesses, would be correct and it would be isolated and tamper-proof. The lattice security model which extends the access matrix model with classification, clearances and rules is found in the Bell and LaPPadula (1973) famous article. Denning (1976) introduced the information flow model which is based on the lattice security model but requires that the flow of information is subject to the flow of relation among security classes.

Again in the 90th due to the large implementation of networking and distribution, authorization had gained lot of interest in many specialized area. In 1990 Glenn *et al* extended the distributed model given by Akyildiz et al (1989) and reported a formal model in centralized, parallel and

T. Sobh et al. (eds.), *Novel Algorithms and Techniques in Telecommunications and Networking,*
DOI 10.1007/978-90-481-3662-9_12,

distributed systems. Glenn *et al* reduced the problem of proving security for concurrent model to proving security for sequential model. Protection in embedded systems was studied in (Rayis 1996). Glasgow *et al* (1992) presented a logic for reasoning about security which is based on a modal logic framework.

Time, roles, constraints, events were motivation to extend classical models to new application areas as in distributed databases and workflow management systems by Sandhu *et al* (1994, 1996), Atluri and Huang (1996), Elisa Bertino *et al* (1996a, 1996b, 1997a and 1997b), (Rayis 1997), (Tomur & Erten, 2006), (Kwon & Moon, 2007) and (Peleg *et al*. 2008). Thorough surveys on the subject of authorization can be found in (Denning 1982), (Sandhu *et al.* 1994), (Sandhu *et al.* 1996), (Goscinski 1991), (Pfleeger 1989), (Boyd 1993), (Leiss 1982), (Landwehr 1981) (Stallings, 2007) and (Stallings & Brown , 2007). Computer Security is characterized by the fact that a single mistake can cost billions. Authorization deals with threats and risks and involves requirements that are considered of supreme importance, thus high level of assurance is needed, and testing alone is insufficient to establish the required level of confidence. Z notation developed at Oxford is a powerful tool for formal expression and reasoning. It is intended to be used here because we think that the development of security systems should follow a process models as information systems development and use similar tools and methodologies.

The Z language is a powerful formal tool for specification and it is proposed to be used to specify the above mentioned problem formally. Snekkenes in [29] had expresses some authentication procedures of X.509 (which is a joint ISO and CCITT specifications) in Z. There he noted the benefits of Z to make compact specifications through the reuse of schemas. Through the use of Z he reported also some weaknesses of the X.509. Boyd (1993) had also specified the authentication in Z, he developed some secure communication architectures using Z. Boswell (1995) presented a formal development of security policy model in Z for the NATO Air Command Control System. He described Mandatory and discretionary access control rules and integrity control. He concluded the capacity of Z to be used in such field, the modularity provided by it and the help it provides for informal validation. Further information about Z notation is found in many references one of them is (Potter *et al.* 1996).

The system to be modeled is a standard Linux protection system. A simplified description of a Linux protection system consists of a finite number of objects of different types. The n types are processes and files and they are related by rights which some object (process) may have over some other object (file). The number of objects is not fixed but is finite. Rights may be "*read*", "*write*", "*execute*", "*delete*", "*update*" and "*own*". A Z specification of the system will first define given sets, definitions and initial conditions. The second thing to be specified is the system state or the configuration. Third is the modeling of primitive operations and last is modeling commands.

II- THE PROTECTION SYSTEM

The model adapted to Linux core protection system presented here was developed using features described in (Johnson & Troan 2005), (Silberschatz et al 2002) (Torvalds, 2008) and the model originally developed by Harrison, et al (1976) was used as a core model. Yet new extensions to model extra security features in Linux can be augmented. An example is the support of role based security authorization, where the model presented in (Sandhu et al. 1996) can be of much help.