

An Architecture and for security policy management on homogeneous networks

Pedro Chávez Lugo, Juan José Flores Romero
Postgrado de Ingeniería Eléctrica
Universidad Michoacana
Morelia, Michoacan, México
pedro@lsc.fie.umich.mx, juanf@umich.mx

Juan Manuel García García
Departamento de Sistemas Computacionales
Instituto Tecnológico de Morelia
Morelia, Michoacan, México
jmgarcia@itmorelia.edu.mx

Abstract

The security policy on operating systems limit the interaction between subjects and objects by means of access control mechanism. Many operating systems use one or more access control models and the sysadmins face complex issues in order to implement and customize the security policy for homogeneous and heterogenous systems. A secure system need right combination of security and functionality with easy of set-up and use. This work propose an architecture and its protocols to easy and secure policy management in homogeneous systems.

1 Introduction

The computer systems need a security policy (set of rules) to keep on resources integrity, confidentiality and availability. Many operating systems use one or more access control models and the sysadmins face complex issues in order to implement and customize the security policy for homogeneous and heterogenous systems. A friendly and secure system need the right combination of security and functionality with easy of set-up and use.

Security Enhanced Linux (SELinux) is a secure operating which use three differences access control models: Multilevel Security (MLS) [BL73], Type Enforcement (TE)¹ [Com06] and a kind of Role Based Access Control (RBAC) [FK92]. Some tools has been development to manipulate and analyze SELinux Policy [Nak05] [May05] [HGH⁺05]. OS Solaris² 10 and SecureOS³ are others example of secure operating systems.

If the policy change implies to reply it in all hosts. In homogeneous systems only is necessary to update the policy in all hosts but in heterogenous systems implies to update the policies in the correct host.

¹Type Enforcement is a Trademark of Secure Computing

²Solaris is a Trademark of Sun Microsystems

³SecureOS is a Register mark of Secure Computing

Secure operating systems uses a local security policy and then on a distributed environment an architecture to easy and secure policy update to all hosts is required. This work propose an architecture and protocols to easy and secure policy management in homogeneous systems.

2 Architecture

For easy and secure management of security policy in homogeneous systems we propose to divide the policy in two parts: System's policy to operate OS and user's policy. The user's policy must be divided to obtain granularity. The granularity must be small to guarantee correct relationships.

In this work we use SELinux as operating system. In our architecture the security server is used by administrator for write, compile and distribute the policy to clients. In SELinux their formal model contains roles, domains, types and levels. Roles to access domains, domain can transition to others domains, domains to access types (See Figure 1). Types are associate to objects like sockets, files, directories, pipes, file systems, etc.

If a user request a role is necessary to know the relationship between domains, domain transitions and types.

3 Protocols

The *Authentication, new role, remote access* protocols are proposed for the easy and secure manage policy. These protocols are inspired by the kerberos V authentication protocol [oT07]. The policy server is denoted by T and clients by A and B .

1. Notation.

E is a symmetric encryption algorithm, k is the session key chosen by T , to be shared by A and B .

L indicates a validity period (lifetime) and N is a maximum execution number to be made by the user.

u user name, $password$ password for u , $c.s$ SELinux

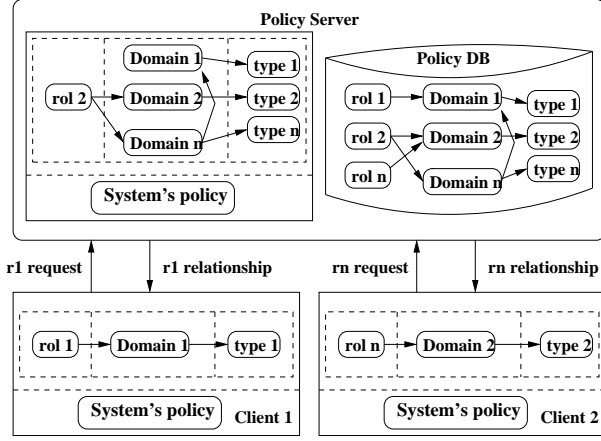


Figure 1. Architecture

security context for u , r role name, rr role relationship, $ticket$ and $idticket$ ticket identifier.

2. One-time setup. A and T share a ket K_{AT} ; similarly, B and T share K_{BT} .

Each role has defined their own L and N parameters. With N parameter tries to limit the user/role executions to protect the system by this misuse.

3.1 Authentication protocol

A user must require a role to operate the computer system. Policy server verify if the user actually can obtain the role requested to send client parameters for session.

- Protocol messages

$$A \rightarrow T: A, E_{K_{AT}}(u, r, password) \quad (1)$$

$$A \leftarrow T: ticket_A = E_{K_{AT}}(rr, c_s, L, N, idticket) \quad (2)$$

3.2 New role protocol

If a user try to obtain a new role:

- Protocol messages

$$A \rightarrow T: A, E_{K_{AT}}(r, idticket) \quad (1)$$

$$A \leftarrow T: E_{K_{AT}}(rr, c_s) \quad (2)$$

3.3 Remote access protocol

If a role try to obtain access to B client:

- Protocol messages

$$A \rightarrow T: A, E_{K_{AT}}(B, idticket) \quad (1)$$

$$A \leftarrow T: E_{K_{AT}}(k), E_{K_{BT}}(k, u, r, idticket) \quad (2)$$

$$A \rightarrow B: A, E_{K_{BT}}(k, u, r, idticket) \quad (3)$$

$$B \rightarrow T: B, E_{K_{BT}}(u, r, idticket) \quad (4)$$

$$B \leftarrow T: ticket_B = E_{K_{BT}}(rr, c_s, L, N, idticket) \quad (5)$$

$$A \leftarrow B: E_k(\text{messages})$$

3.4 Reply relationships

When the policy change in policy server all the clients must be updated with the respective relationship to guarantee consistency. The server policy contains a list of clients and roles for reply the right relationships.

4 Conclusions

Is necessary to extend the use of secure operating systems in computer networks and develop the correct schemas to obtain right combination of security and functionality with easy of set-up and use. Divide the policy to obtain granularity can provide us with easy policy management for quickly distribution. Is necessary the use of some protocols to guarantee the integrity and confidentiality policy in policy distribution.

References

- [BL73] D. Elliot Bell and Leonard LaPadula. Secure computer systems: Mathematical foundations. Technical report, 1973. MITRE Technical Report 2547, Volume I.
- [Com06] Secure Computing. Type enforcement technology, 2006. <http://www.securecomputing.com>.
- [FK92] David Ferraiolo and Richard Khun. Role-based access control. 1992. Proceedings 15th National Computer Security Conference.
- [HGH⁺05] Amy L. Herzog, Joshua D. Guttman, David R. Harris, Jonh D. Ramsdell, Ariel E. Segall, and Brian T. Sniffen. Policy analysis and generation work at mitre. 2005. Security Enhanced Linux Symposium.
- [May05] Frank Mayer. Setools package: Tools for understanding selinux policies. 2005. Security Enhanced Linux Symposium.
- [Nak05] Yuichi Nakamura. Simplifying policy management with selinux policy editor. 2005. Security Enhanced Linux Symposium.
- [oT07] Massachusetts Institute of Technology. Kerberos: The network authentication protocol, 2007. <http://web.mit.edu/kerberos/>.