SE Linux Security Components

Paras Garg *(Student)*, *Stevens Institute of Technology*

Abstract— What the SE Linux required for any process or file to execute, how these required components have the impact on the SE Linux decision making and why does SE Linux need these requirements for enabling enhanced security in the Linux system.

Index Terms— Context, Label, Linux, Policy, SE Linux, Security.

INTRODUCTION

THE concept of this document is to provide the detailed information about SE Linux Security Components. It will be covering all the important components required by SE Linux to maintain the integrity and security within the Linux system that Linux itself can not preserve.

The document will provide the comprehensive view of the features and the strategies adopted by SE Linux that enhance the capability of the Linux system to defend itself from the exploitation and controls the access to the resources. How the nature and behavior of the resource get evaluated in the Linux system in the presence of SE Linux.

It also comprehends the functionality of the SE Linux that how the resource executes in the SE Linux, how SE Linux decides which process request is to be approved or to be disapproved, how it marks the difference in trusted and non trusted processes, how it enables the sensitivity level inside the system, and how it changes the security measures inside the Linux system.

SE LINUX CONTEXT

The first step in the SE Linux security is to put labels on each [1]entity in the Linux system. The label is like a process or a file attribute which shows the context of the resource. The label could be any attribute such as owner, group, or date of creation of a process or a file.

Definition

All the processes or files are labeled such a way that represents security relevant information. This information is called SE Linux Context. SE Linux context can also be interpreted as the collection of security relevant information.

[1]The entity could be anything in a Linux system such as files, processes, sockets, network interfaces, etc.

• *Paras Garg is the Student with the Department of Computer Science, Stevens Institute of Technology, 1 Castle Point Terrace, Hoboken, NJ 07030. E-mail: pgarg2@ stevens.edu.*

Need

SE Linux Context or Label is the most important aspect for maintaining the SE Linux system, as all the SE Linux policy decisions or access control decisions are based on the label of the resource.

Fig. 1

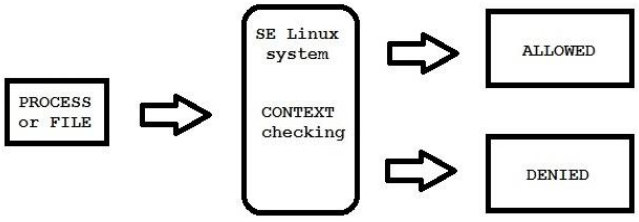


Fig. 1 shows, how a process or a file is allowed or rejected by SE Linux on the basis of context which a process or a file has.

SE Linux doesn't care how the how the process is called, or which process id it has, or under which account the process run. All it wants to know is what the context of the triggered process.

Elements

SE Linux contains additional information about a process or a file

1. User, which represents the SE Linux user.
2. Role, which represents the SE Linux role.
3. Type, which represents the SE Linux type.
4. Level, represents the sensitivity.

Syntactically, all these elements are separated by colon ':' in the given Linux command.

USER **:** ROLE **:** TYPE **:** LEVEL

Example, in the following commands the SE Linux context which is used in a process to retrieve the information about the current user.

```
$ id -Z
unconfined_u:unconfined_r:unconfined_t:s0
```

SE LINUX USER

SE Linux user is different from the Linux user. Unlike the Linux user information which can change while working on the system, the SE Linux user remains the same even when the Linux user itself has changed. Because of this unchangeable state of the SE Linux user specifies the access controls can be implemented to ensure that users cannot work around the (limited) set of permissions granted to them, even when they get privileged access.

But the most important feature of SE Linux user is that it defines restrictions on the role which is allowed to the user. Once a user is assigned as a SE Linux user, it is not allowed to change the role that it isn't meant to be in. This is the role-based access control implementation of SE Linux.

By convention SE Linux user is defined by a "_u" suffix as shown in the commands below, although it is not mandatory.

```
$ id -Z
unconfined_u:unconfined_r:unconfined_t:s0
```

SE LINUX ROLE

SE Linux role ensures that role-based control is enabled in the SE Linux system as this control is vital in order to keep the system secure, especially from the malicious user attempts.

SE Linux roles are basically used to define which type of process user can be in that implies SE Linux role defines what a user can do or what cannot do.

Typically, there are five types of roles in any SE Linux enabled system, but can also supports more and by convention SE Linux role is defined by a "_r" suffix as shown in the commands below.

```
$ id -Z
unconfined_u:unconfined_r:unconfined_t:s0
```

The types of roles which are present in SE Linux enabled system are–

1. user_r – This role is for restricted users, and allowed to have processes with types specific to end user applications. Those users use to switch Linux users are not allowed.
2. staff_r – This role is for non-critical tasks. This role is best for the operators as tasks are restricted to some applications as restricted user.
3. sysadm_r – This role is for system administration tasks. The role is best for system administrators, which are allowed to perform various system administration tasks and no end user applications are allowed to keep the system free from infections as end user applications are considered as untrusted and vulnerable.
4. syster_r – This role is for background processes and allows various daemon and system processes. No other types of applications such as end user applications and administration applications are allowed.
5. unconfined_r – This role is for unconfined tasks, best suited for the end users. This role allow a limited number of applications and also allows some privileged applications which are necessary to operate other application, but has to be in more or less unconfined manner. This privileged access is only allowed to system administrators those who wanted to protect certain applications by keeping other system applications or operations almost untouched.

SE LINUX TYPE

SE Linux type is the most crucial and important element among all four elements. Approximately over 99 percentages of policy rules consist of rules based on the interaction between the two types which even do not mention about other elements (user, role and sensitivity).

SE Linux type is essential because it enforces the mandatory access control system as SE Linux type defines fine-grained access control of that process (called the domain) with respect to itself and other types such as processes, files, sockets, network interfaces, etc. And this fine-grained access control is most likely responsible for rejecting the access attempts by the processes.

SE Linux type enforcement enables the SE Linux to control what an application is allowed to do which is based on how the application got executed on the first place. For example, suppose a web server is executed by the system using init function and the same web server is launched by the end user interactively. Even though the executed process binary and the path is same, the process executed by the system is considered as trusted while on the other hand the process executed by the user is considered as normal behavior. And based on this considered behavior system would be allowed to do anything what it wanted to do and user would have the restricted set of roles which it can perform.

By convention SE Linux type is defined by a "_t" suffix as shown in the commands below, although it is not mandatory.

```
$ id -Z
unconfined_u:unconfined_r:unconfined_t:s0
```

SE LINUX LEVEL

SE Linux level is also called the sensitivity label, it represent the sensitivity of the resource. The sensitivity label is an optional element of the SE Linux context and in some of the SE Linux enables the system to have this element set disable by default.

SE Linux level is needed for Multi Level Security support within the SE Linux. These sensitivity labels are used for classification of resources and restriction of access to resources based on a security clearance.

In the most documentation of any organization the sensitivity label is marked as public, internal, confiden-

tial, strictly confidential, or regulatory but the syntax of SE Linux level does not allow to store values in this classification, instead of this sensitivity label accepts the numbers ranging from '0' as the lowest confidentiality to any number which a system administrator wants to assign as the highest confidentiality value. This highest value configured with the SE Linux policies when the policies were built.

These sensitivity labels consist of two parts: a confidentiality value (prefix with "s") and a category value (prefix with "c") as shown in command lines below.

```
$ id -Z
user_u:user_r:user_t:s0.s6:c1.c6
```

In the above command lines 's0. s6' represents the confidentiality value of label as the first part 's0' shows the current sensitivity level, whereas the second part 's6' shows the sensitivity clearance level and if this part is not present then 0 is considered to be as sensitivity clearance level. And 'c1.c6' represents the category value, it shows the category is set from 0 through 6 and if this section is not present in the command line then the category value is considered as 0.

The parts of sensitivity label:

1. Confidentiality Value – It defines a process must have minimum access control permit to access the resource else the attempt would be denied.
2. Category Value – The idea behind this part is to enforce multi-tenancy within the Linux system by assigning same tags to processes and resources having a similar tenant. This tagging allows the similar tagged processes to access the targeted resource when the tag of process and resource matches. And if the proper tagging is not assigned or matched the process are not allowed to access the targeted resources or processes.

SE POLICY

Policy is the set of rules that guide the SE Linux security engine. It defines types for file objects and domains for processes, uses roles to limit the domains that can be entered, and has user identities to specify the roles that can be attained.

The policy specifies the rules in that environment. It is written in a simple language created specifically for writing security policy. Policy writers use m4 macros to capture common sets of low-level rules. There are a number of m4 macros defined in the existing policy, which assist greatly in writing new policy. These rules are preprocessed into many additional rules as part of building *policy.conf,* which is compiled into the binary policy.

SE LINUX SECURITY

Label Based Access Control (LBAC)

LBAC is configured by the system administrator by creating a security label components. This security label contains the data which is required to match with the process which attempt to access the resources. Insufficient or mismatch of data will reject the access of resources.

Role Based Access Control (RBAC)

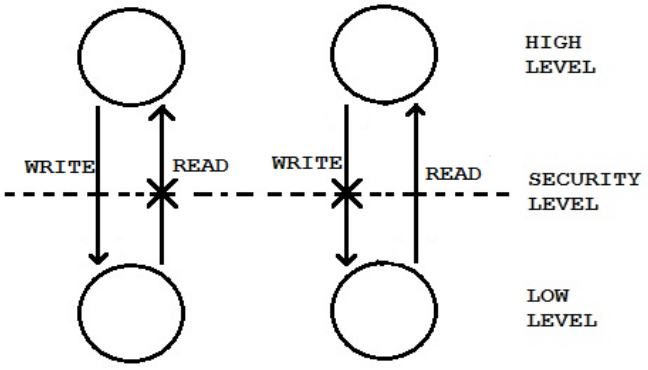
RBAC enforcement enables the SE Linux to control what an application is allowed to do which is based on how the application got executed on the first place.

User Based Access Control (UBAC)

UBAC prevent the user of SE Linux from changing the information while compiling and do not allow to access to the resources or files of the different SE Linux user.

Multi Level Security (MLS)

MLS enforce the Bell-LaPadula configuration in SE Linux that is based on security clearance level. It states that a process can neither read anything with a higher confidentiality level nor write anything to any resource with a lower confidentiality level.



Multi Category Security (MCS)

MCS allows users to label process or files with categories and allow all the processes to access the resource which has the same assigned category. If category mismatch or doesn't assign properly, then no access to resources is granted.

REFERENCES

[1] Red Hat SELinux Guide (2005), Red Hat, Inc. Retrieved from https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/4/html/SELinux_Guide/

[2] Robert Krátký and Barbora Ančincová, Red Hat SELinux User Guide (2016), Red Hat, Inc. Retrieved from https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Security-Enhanced_Linux/

[3] Sadequl Hussain (2014, September 25), DigitalOcean. Retrieved from https://www.digitalocean.com/community/tutorials/an-introduction-to-selinux-on-centos-7-part-2-files-and-processes#selinux-for-processes-and-files/

[4] Sven Vermeulen (2013), "*SELinux System Administration*". Retrieved from https://www.safaribooksonline.com/library/view/selinux-system-administration/9781783283170/

[5] IBM Knowledge Center, IBM. Retrieved from https://www.ibm.com/support/knowledgecenter

[6] Gentoo Foundation, Inc. Retrieved from https://wiki.gentoo.org/wiki/SELinux/Tutorials/SELinux_Multi-Level_Security