Understanding SElinux

Report-1

Introduction

Security-Enhanced Linux (SElinux) is an implementation of a mandatory access control mechanism in the Linux kernel, checking for allowed operations after standard discretionary access control rules are checked. Operating System security is the main goal of SElinux. It does not care about application security. It is a security enhancement to the Linux which can solve the problem of flawed application software and can enforce many security goals, from data confidentiality, data & application integrity to improved robustness.

The predominant type of access control we have today (in standard Linux) is called **discretionary access control (DAC)**. DAC is commonly known as *file permissions* .

In DAC, the individual users (or resource owners) specify who may or may not access the resource.In a way , users control the permissions of the files (objects, resources etc.) that they own.In standard Linux, the DAC is implemented through owner-group-world permission mode mechanism.

DAC has some inherent security flaws :-

DAC access decisions are only based on user identity and ownership, ignoring other security-relevant information such as the role of the user, the function and trustworthiness of the program, and the sensitivity and integrity of the data. Each user typically has complete discretion over their files, making it difficult to enforce a system-wide security policy. Furthermore, every program run by a user inherits all of the permissions granted to the user and is free to change access to the user's files, so minimal protection is provided against malicious software.So a process run by the root has root access to any file. It can modify even those files which it does not need in order to operate.

Security-Enhanced Linux (SELinux) adds **Mandatory Access Control (MAC)** to the Linux kernel .

A general purpose MAC architecture needs the ability to enforce an administratively-set security policy over all processes and files in the system, basing decisions on labels containing a variety of security-relevant information. MAC provides strong separation of applications

that permits the safe execution of untrustworthy applications. Its ability to limit the privileges associated with executing processes limits the scope of potential damage that can result from the exploitation of vulnerabilities in applications and system services. MAC enables information to be protected from legitimate users with limited authorization as well as from authorized users who have unwittingly executed malicious applications .

SElinux Architecture

SElinux is a linux security module that is built into the Linux kernel. It is driven by loadable policy rules. When a process tries to access a file, the operation is intercepted in the kernel by SElinux. If a policy rule allowd the operation, it continues, otherwise , it is blocked.SELinux decisions, such as allowing or disallowing access, are cached. This cache is known as the Access Vector Cache (AVC). We need to write .te file and then compile it to get the policy.VERSIONNUMBER file.

SElinux Contexts

Standard Unix uses the process's user & group ID and file's access mode , file user/group to grant or deny access.

While in SElinux, the access control attribute is called a security context. All objects (files, interprocess communication channels, sockets, network hosts etc) and subjects (processes) have a single security context associated with them. SELinux context that contains additional information, such as an SELinux user, role, type, and, optionally, a level. When running SELinux, all of this information is used to make access control decisions.SELinux contexts follow the SELinux user:role:type:level syntax:.A valid security context must have a valid user, role and type identifier. These identifiers are defined by the policy writer.

Eg: $ ls -Z file1
-rwxrw-r-- user1 group1 unconfined_u:object_r:user_home_t:s0 file1

SELlinux *user:* The SELinux user identity is an identity known to the policy that is authorized for a specific set of roles, and for a specific MLS range. Each Linux user is mapped to an SELinux user via SELinux policy.

role : Part of SELinux is the Role-Based Access Control (RBAC) security model. SELinux users are authorized for roles, and roles are authorized for domains. The role serves as an intermediary between domains and SELinux users.

type: The type is an attribute of Type Enforcement. The type defines a domain for processes, and a type for files. SELinux policy rules define how types can access each other, whether it be a domain accessing a type, or a domain accessing another domain. Access is only allowed if a specific SELinux policy rule exists that allows it.

Security Context for Processes can be viewed using **ps -eZ** command**.**

SElinux uses the security context of the process and the object that the process accesses.Then SELinux checks the policy to see whether the process type is allowed to access the object type. So the basis for access control in SElinux-permissions is the security context of the process & the object.

Allow Rule:

allow subject_t object_t : object_class { permissions }
Above rule allows subject_t type to access object_type . The level of access granted can be controlled by the permissions given in the rule.

So , in a nutshell , this is all there is . In order to secure a system , you first identify all the users , assign them roles , then label all the files and users with their security context types. Then you write allow rules to achieve the level of security you need & compile all the type definitions and allow rules in one .policy file. But it is easier said than done , as there are infinite number of security scenarios and the allow rules can get pretty complex if we want to cover each and every scenario.

Benefits of Running SELinux:

- All processes and files are labeled with a type. A type defines a domain for processes, and a type for files. Processes are separated from each other by running in their own domains, and SELinux policy rules define how processes interact with files, as well as how processes interact with each other. Access is only allowed if an SELinux policy rule exists that specifically allows it.
- Fine-grained access control. SELinux access decisions are based on all available information, such as an SELinux user, role, type, and, optionally, a level.
- SELinux policy is administratively-defined, enforced system-wide, and is not set at user discretion.
- Reduced vulnerability to privilege escalation attacks. For example, if the Apache HTTP Server is compromised, an attacker can not use that process to read files in user home directories, unless a specific SELinux policy rule was added or configured to allow such access.
- SELinux can be used to enforce data confidentiality and integrity, as well as protecting processes from untrusted inputs.

Targeted Policy:

Targeted policy is the default SELinux policy used in Red Hat Enterprise Linux. When using targeted policy, processes that are targeted run in a confined domain, and processes that are not targeted run in an unconfined domain.So the policy governs only some “targeted” deamons.

Confined Processes: Almost every service that listens on a network, such as sshd or httpd, is confined. Also, most processes that run as the Linux root user and perform tasks for users, such as the passwd application, are confined. When a process is confined, it runs in its own domain, such as the httpd process running in the httpd_t domain.

Unconfined Processes :Unconfined processes run in unconfined domains, for example, init programs run in the unconfined initrc_t domain, unconfined kernel processes run in the kernel_t domain, and unconfined Linux users run in the unconfined_t domain. For unconfined processes, SELinux policy rules are applied, but policy rules exist that allow processes running in unconfined domains almost all access. Processes running in unconfined domains fall back to using DAC rules exclusively. If an unconfined process is compromised, SELinux does not prevent an attacker from gaining access to system resources and data, but of course, DAC rules are still used. SELinux is a security enhancement on top of DAC rules - it does not replace them.

Error Messages: We can install setroubleshoot-server to view all the denial messages from the SELinux . It helps to analyze the AVC messages .

Configuration File : The /etc/selinux/config file is the main SELinux configuration file. It controls the SELinux mode and the SELinux policy to use: There are three SELinux mode which can be set in the config file .

Enforcing: SELinux policy is enforced, and SELinux denies access based on SELinux policy rules.Denial messages are logged.

Permissive : SELinux policy is not enforced. SELinux does not deny access, but denials are logged for actions that would have been denied if running SELinux in enforcing mode.

Disabled: SELinux is disabled .No rules are enforced .Only the DAC rules are used.

Use the /usr/sbin/setenforce command to change between enforcing and permissive mode. Changes made with /usr/sbin/setenforce do not persist across reboots.

So , now that we have seen the very basics of SELinux the next target for us is to work with some deamons like httpd, mysqld, sshd , samba , sendmail etc . We try to use SELinux framework to secure these deamons.