

What is SELinux (Security Enhanced Linux) and Type Enforcement?

by Gregg Zepp



Although often mistakenly considered a complete security OS solution, it is not. What it does is help add security functionality to a Linux distribution through the use of security modules in the kernel. It is not a distribution, albeit specific distributions exist because of its presence and security features, but rather, it is a set of security modifications that be applied to the OS kernel to restrict applications and user access. [1] It consists of a library (libseline), and userland utilities for compiling policy (checkpolicy), and loading policy (policycoreutils), and other user programs. It's main purpose is to provide an implementation of Mandatory Access Controls (MAC) using type enforcement and role-based access control (RBAC) within a Linux OS. [2]

There is an identity associated with every process on the Linux system. When a user presents credentials and logs on, the identity portion of the security context related to the login shell will reflect the user's identity. The RBAC policy used in SELinux defines allowable user actions for a particular role using a "type" policy. The RBAC policy specifies domains that can be entered by roles and transfers the assignment of permissions to the type configuration. [3]

The type enforcement model is important to SELinux. A "type" is a way of classifying an application or resource. Type enforcement is the enforcement of access control on that type. All files, processes, network resources, etc, on an SELinux system has a label, and one of the components of that label is the "type". For example, the files in one's Linux home directory are probably labeled *user_home_t*. *user_home_t* is the type and means the policy should treat all those files as your home directory files. Running applications also have labels. For example, one's web browser may be running as *firefox_t* on a system. [4] Type enforcement simply allows one to specify what application label can access what resource label. Another example could be network services being confined to a particular port and the Apache web server being restricted to only 80 by default. [5] Lastly, since MAC is a central function to SELinux and implements a centralized policy determining which software/mechanisms/users can access what resources, it would be helpful to give an overview of MAC.

What are Mandatory Access Controls (MAC)? When a system is operating under the MAC model, users and data owners do not have as much freedom to determine who can access files and who cannot. The OS makes the final decision and can override the user's wishes. The OS's security is often set by system policies, which are set by system owners and administrators and users cannot modify them. Although not a specific example of MAC policies, but an example of an OS's security policies would be, for example, in a Windows system can be found at: START-> Programs -> Administrative Tools -> Local Security Policy and the system's Group Policies by typing in "Gpedit.msc" at the Run command off the START menu. [6]

These policies indicate which subject has access to which object. This access control model can increase the level of security, because it is based on a policy that does not allow any operation not explicitly authorized by the system owners/administrators. The MAC model is developed for and implemented in systems in which confidentiality has the highest priority, such as in the military. Users receive a clearance label and objects receive a classification label, also referred to as security levels. [7] The labels may often have the same names.

When users are given a security clearance, such as Secret, Top Secret, Confidential, etc, users have been cleared to handle data labeled the same when they have a need to know. When a system makes a decision about fulfilling a request to access an object, it is based on the clearance level of the subject, the classification label attached to the object, and the security policy/policies of the system. Security labels are attached to all files, directories, and devices. Therefore, for example, a user may have a security clearance of Secret, and the data the user requests may have a security label with the classification of Top Secret. In this situation, the user will be denied access to the file/directory/system because their clearance level is not equal to nor higher than the classification of the file/directory/system they are trying to access. [8]

MAC is used in places where information classification and confidentiality is of most importance, such as military institutions. MAC cannot be simply turned on/off, unless the OS distribution one is using has the feature built in. Special distributions exist where this feature has been implemented.

The NSA (National Security Agency) developed SELinux initially. The project website can be found: www.nsa.gov/research/selinux/index.shtml

Since then, the NSA has partnered with Red Hat to continue development and carry out integration of SELinux into Fedora and Red Hat Enterprise Linux. [9] It is not specific to Red Hat however and other Linux distributions and other operating systems have adopted SELinux and similar frameworks. Some of these supporting distributions are: Gentoo, Debian, Ubuntu, SUSE, Slackware, and Solaris's Trusted Solaris OS. [10]

-
1. SELinux Project FAQ: selinuxproject.org/page/FAQ
 2. Gentoo's SELinux page: www.gentoo.org/proj/en/hardened/selinux/
 3. NSA's SELinux site: <http://www.nsa.gov/research/files/selinux/papers/ottawa01/node3.shtml>
 4. SELinux Project FAQ: selinuxproject.org/page/FAQ
 5. Fedora Project FAQ: fedoraproject.org/wiki/SELinux/FAQ
 6. Windows Local Security Policy: technet.microsoft.com/en-us/library/dd277395.aspx
 7. CompTIA Security+ TechNotes: www.techexams.net/technotes/securityplus/mac_dac_rbac.shtml
 8. Harris, Shon; *CISSP Certification All-in-One Exam Guide*, 4th ed; McGraw-Hill Osborne Media, 2007. p.212
 9. Fedora Project FAQ: fedoraproject.org/wiki/SELinux/FAQ
 10. Wiki: en.wikipedia.org/wiki/Security-Enhanced_Linux#Other_systems