INTRODUCTION TO SELINUX

Ashwini Mathur

Security Enhanced Linux

Security-Enhanced Linux (SELinux) is a security architecture integrated into the 2.6.x kernel using the *Linux Security Modules* (LSM).

It is a project of the United States National Security Agency (NSA) and the SELinux community.

SELinux Overview

SELinux provides a flexible ***Mandatory Access Control*** **(MAC)** system built into the Linux kernel.

Under standard Linux ***Discretionary Access Control*** **(DAC)**, an application or process running as a user (UID or SUID) has the user's permissions to objects such as files, sockets, and other processes.

Running a MAC kernel protects the system from malicious or flawed applications that can damage or destroy the system.

SELinux defines the **access and transition rights** of every user, application, process, and file on the system.

On a day-to-day basis, system users will be largely unaware of SELinux. Only **system administrators need to consider how strict a policy to implement for their server environment.** The policy can be as strict or as lenient as needed, and is very finely detailed. This detail gives the SELinux kernel complete, granular control over the entire system.

WHY SELinux ?

Overview

SELinux implements what's known as MAC (Mandatory Access Control). This is implemented on top of what's already present in every Linux distribution, the DAC (Discretionary Access Control).

To understand DAC, let's first consider how traditional Linux file security works.

In a traditional security model, we have three entities: User, Group, and Other (u,g,o) who can have a combination of Read, Write, and Execute (r,w,x) permissions on a file or directory. If a user jo creates a file in their home directory, that user will have read/write access to it, and so will the jo group. The "other" entity will possibly have no access to it.

Running a command like this:

```
ls -l /home/jo/
```

Now jo can change this access. jo can grant (and restrict) access to this file to other users and groups or change the owner of the file. These actions can leave critical files exposed to accounts who don't need this access.

jo can also restrict to be more secure, but that's discretionary: there's no way for the system administrator to enforce it for every single file in the system.

SELinux Modes

It's time to start playing around with SELinux, so let's begin with SELinux modes.

At any one time, SELinux can be in any of three possible modes:

- Enforcing
- Permissive
- Disabled

In **enforcing mode** SELinux will *enforce* its policy on the Linux system and make sure any unauthorized access attempts by users and processes are denied. The access denials are also written to relevant log files. We will talk about SELinux policies and audit logs later.

Permissive mode is like a semi-enabled state. SELinux doesn't apply its policy in permissive mode, so no access is denied. However any policy violation is still logged in the audit logs. It's a great way to test SELinux before enforcing it.

The **disabled mode** is self-explanatory – the system won't be running with enhanced security.

Checking SELinux Modes and Status

We can run the getenforce command to check the current SELinux mode.

```
getenforce
```

We can also run the sestatus command:

```
sestatus
```

SELinux Configuration File

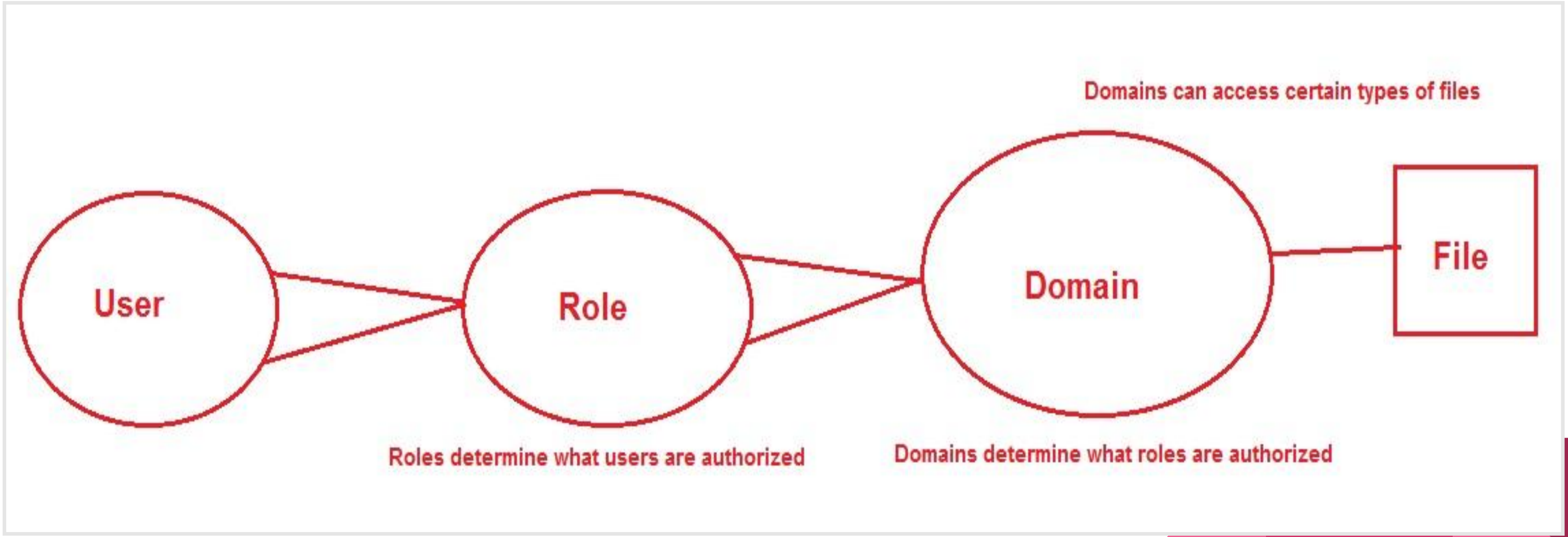
The main configuration file for SELinux is /etc/selinux/config. We can run the following command to view its contents:

cat /etc/selinux/config

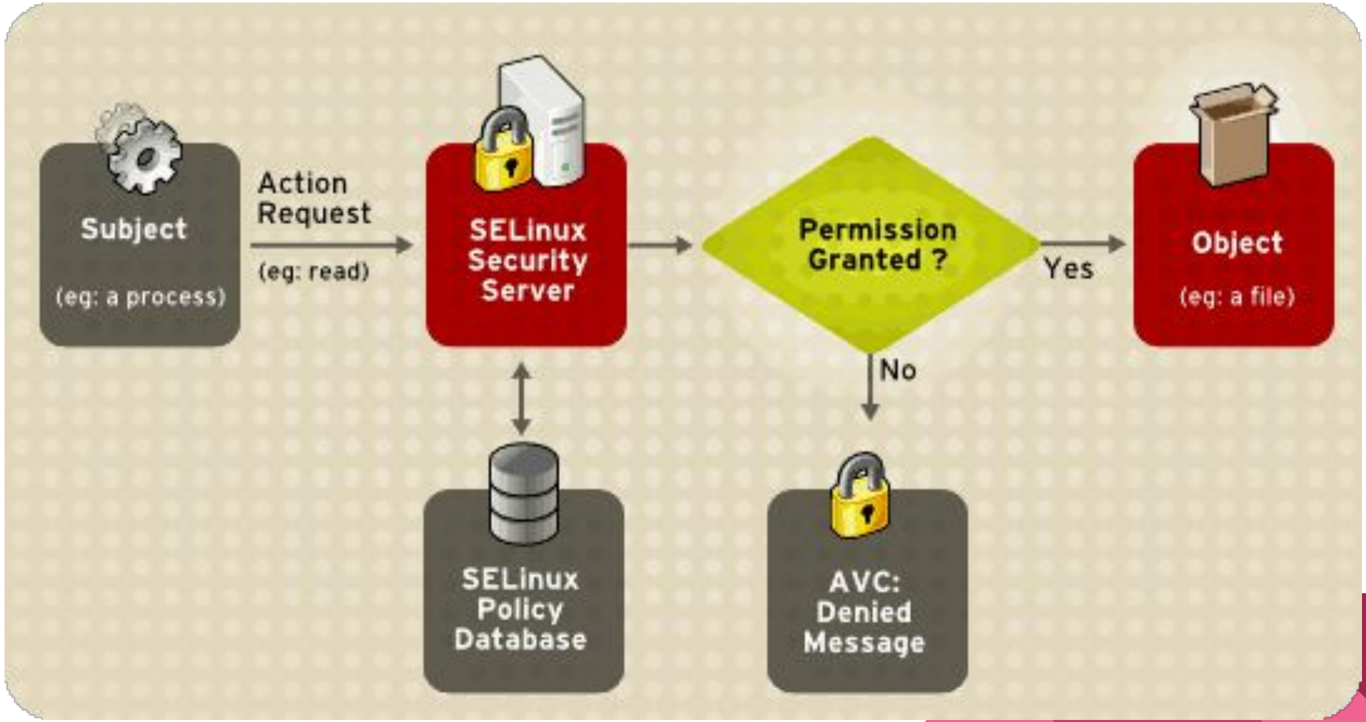
The output will look something like this:

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled
# SELINUXTYPE= can take one of these two values:
#     targeted - Targeted processes are protected,
#     minimum - Modification of targeted policy. Only selected processes are protected.
#     mls - Multi Level Security protection.
SELINUXTYPE=targeted
```

SElinux Policy


Domains can access certain types of files
User
Role
Domain
File
Roles determine what users are authorized
Domains determine what roles are authorized

The SELinux Decision Making Process



Permissive Mode

SELinux can run in ***permissive* mode**, where the (Access Vector Cache)**AVC** is checked and denials are logged, but **SELinux does not enforce the policy**. This can be useful for **troubleshooting and for developing or fine-tuning SELinux policy**.

The `/etc/selinux/` directory is the primary location for all policy files as well as the main configuration file.

The following example shows sample contents of the `/etc/selinux/` directory:

```
-rw-r--r--  1 root root  448 Sep 22 17:34 config
drwxr-xr-x  5 root root 4096 Sep 22 17:27 strict
drwxr-xr-x  5 root root 4096 Sep 22 17:28 targeted
```

The two subdirectories, `strict/` and `targeted/`, are the specific directories where the policy files of the same name (that is, `strict` and `targeted`) are contained.

SELinux Utilities

The following are some of the commonly used SELinux utilities:

- `/usr/sbin/setenforce` — Modifies in real-time the mode in which SELinux runs.
  For example:
  `setenforce 1` — SELinux runs in enforcing mode.
  `setenforce 0` — SELinux runs in permissive mode.

IPTABLES

Packet Filtering

The Linux kernel uses the **Netfilter** facility to filter packets, allowing some of them to be received by or pass through the system while stopping others.

- `filter` — The default table for handling network packets.
- `nat` — Used to alter packets that create a new connection and used for *Network Address Translation* (*NAT*).
- `mangle` — Used for specific types of packet alteration.

The built-in chains for the `filter` table are as follows:

- *INPUT* — Applies to network packets that are targeted for the host.
- *OUTPUT* — Applies to locally-generated network packets.
- *FORWARD* — Applies to network packets routed through the host.

The built-in chains for the `nat` table are as follows:

- *PREROUTING* — Alters network packets when they arrive.
- *OUTPUT* — Alters locally-generated network packets before they are sent out.
- *POSTROUTING* — Alters network packets before they are sent out.

The built-in chains for the `mangle` table are as follows:

- *INPUT* — Alters network packets targeted for the host.
- *OUTPUT* — Alters locally-generated network packets before they are sent out.
- *FORWARD* — Alters network packets routed through the host.
- *PREROUTING* — Alters incoming network packets before they are routed.
- *POSTROUTING* — Alters network packets before they are sent out.

Every network packet received by or sent from a Linux system is subject to at least one table. However, a packet may be subjected to multiple rules within each table before emerging at the end of the chain. The structure and purpose of these rules may vary,

Command Options for IPTables

Rules for filtering packets are created using the `iptables` command. The following aspects of the packet are most often used as criteria:

- *Packet Type* — Specifies the type of packets the command filters.
- *Packet Source/Destination* — Specifies which packets the command filters based on the source or destination of the packet.
- *Target* — Specifies what action is taken on packets matching the above criteria.