**信息系统安全大作业**

|  |  |
| --- | --- |
| 学年学期： | **2019 -2020 学年◼春🞏秋学期** |
| 课程名称： | **信息系统安全** |
| 姓 名： |  |
| 学院和专业： | **网络空间安全与信息法学院**  **信息安全卓越班** |
| 班 级： |  |
| 教 师： |  |
| 联系电话： |  |

基于SELinux的强制访问研究

摘 要：针对SELinux不同模式下的规则进行强制访问。在Linux中利用SELinux进行强制访问控制实验，创建不同角色，制定相关策略，并分别绑定不同用户，以达到切换不同用户产生的访问效果不同。

关键词：SELinux；强制访问；

**Research on Mandatory Access Based on SELinux**

1(Chongqing University of Posts and Telecommunications, Chongqing 400065,China)

1. Mandatory access to the rules in different modes of SELinux. Use SELinux in Linux to conduct mandatory access control experiments, create different roles, formulate relevant strategies, and bind different users separately to achieve different access effects when switching between different users.
2. SELinux ; Mandatory access control；

# 引　言

对于SELinux的应用，它带给Linux的主要价值是为其提供了一个灵活的，可配置的MAC机制。

Security-Enhanced Linux (SELinux)由以下两部分组成：Kernel SELinux模块(/kernel/security/selinux)、用户态工具。SELinux是一个安全体系结构，它通过LSM(Linux Security Modules)框架被集成到Linux Kernel 2.6.x中。它是NSA (United States National Security Agency)和SELinux社区的联合项目。

SELinux提供了一种灵活的强制访问控制(MAC)系统，且内嵌于Linux Kernel中。SELinux定义了系统中每个用户、进程、应用和文件的访问和转变的权限，然后它使用一个安全策略来控制这些实体(用户、进程、应用和文件)之间的交互，安全策略指定如何严格或宽松地进行检查。

SELinux对系统用户(system users)是透明的，只有系统管理员需要考虑在他的服务器中如何制定严格的策略。策略可以根据需要是严格的或宽松的。

只有同时满足了标准Linux访问控制和SELinux访问控制时，主体才能访问客体。[1]

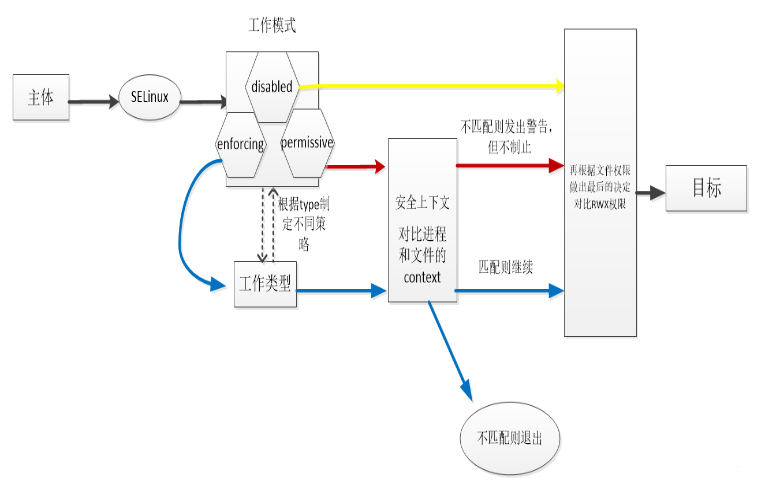


图1 SELinux工作原理

Fig.1 SELinux working principle

# SELinux介 绍

## SELinux概述

SELinux是一种基于域-类型模型的强制访问控制安全系统，它由NSA编写并设计成内核模块包含到内核中，相应的某些安全相关的应用也被打了SELinux补丁，最后还有一个相应的安全策略。在某种程度上它可以被看作是与标准权限系统并行的权限系统。在常规模式中，以用户身份运行进程，并且系统上的文件和其他资源都设置了权限标签，若要访问文件，你必须具有普通访问权限和SELinux访问权限，即使以超级用户身份root运行进程也有可能拒绝访问。

### SELinux的作用和权限管理

SELinux主要作用就是最大限度地减小系统中服务进程可访问的资源。通过对进程和文件资源采用MAC（强制访问方式）为Linux系统提供安全性。

在没有使用SELinux的操作系统中，决定一个资源能否被访问的因素是某个资源是否拥有对应用户的权限；只要访问这个资源的进程符合这个条件就能被访问，root用户不受任何限制，所以系统上任何资源都可以无限制地访问。（DAC）

在使用了SELinux的操作系统中，决定一个资源能否被访问的因素除了用户权限外，还需判断每一类进程是否拥有对某一类资源的访问权。即使进程是以root身份运行的，也需要判断这个进程的类型以及允许访问的资源类型才能决定是否允许访问某个资源；或以root身份运行的服务进程，一般也只能访问到它所需要的资源，即使程序出了漏洞，影响范围也只有在其允许的访问资源内，大大增强安全性。（MAC）

### SELinux三种状态

⑴ enfirocing------强制模式

代表SELinux运作中，且已经正确的开始限制domain/type了。

⑵ permissive------宽容模式

代表SELinux运作中，但仅会有警告讯息并不会实际限制domain/type的存取，这种模式可以用来作为SELinux的debug之用。

⑶ disabled------关闭模式

SELinux并没有实际运作。

### SELinux安全上下文

安全上下文是一个简单的、一致的访问控制属性，在SELinux中，类型标识符是安全上下文的主要组成部分，一个进程的类型通常被成为一个域（domain），“域”和“域类型”意思都一样，都是安全上下文中的“type”。

cp会重新生成安全上下文（变为目标目录相同）；mv不会改变。

安全上下文格式：

⑴ USER

类似Linux系统中的UID，提供身份识别，用来记录身份。常见的user有user\_u（普通用户登录系统后的预设）；system\_u（开机过程中系统进程的预设）；root（登录后的预设）；undefined\_u（非限制，没有定义）。

⑵ ROLE

object\_r（文件、目录和设备的role）；system\_r（程序的role）；用户的role；

⑶ TYPE

用来将主体（subject）和客体（object）划分为不同的组，给每个主体和系统中的客体定义了一个类型，为进程运行提供最低的权限环境。

当一个类型与执行中的进程相关时，其type也称为domain。

### 布尔值

SELinux布尔值是更改SELinux策略行为的开关，可以启用或禁用的规则。安全管理员可以使SELinux布尔值来调整策略，以有选择地进行调整许多软件包都具有man page \*——selinux（8），其中详细说明了所使用的一些布尔值；man -k ‘——selinux’可以轻松地找到这些手册。getsebool用于显示布尔值，setsebool用于修改布尔值，setsebool -P修改SELinux策略，以永久保留修改。

### SELinux的运行机制

当一个subject(如: 一个应用)试图访问一个object(如：一个文件)，Kernel中的策略执行服务器将检查AVC (Access Vector Cache), 在AVC中，subject和object的权限被缓存(cached)。如果基于AVC中的数据不能做出决定，则请求安全服务器，安全服务器在一个矩阵中查找“应用+文件”的安全环境。然后根据查询结果允许或拒绝访问，拒绝消息细节位于/var/log/messages中。[1]

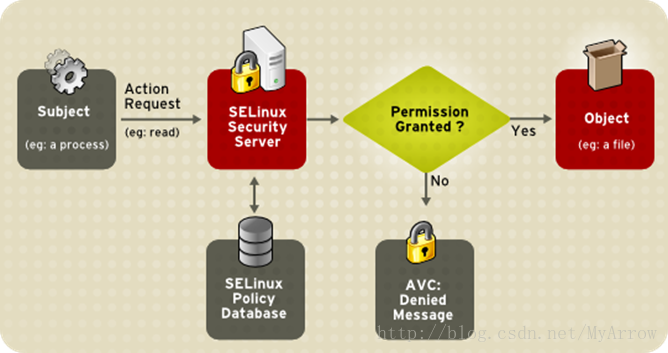


图2 SELinux决策过程

Fig.2 SELinux decision process

### SELinux日志管理

SELinux阻止的进程的日志记录存放在/var/log/audit/audit.log文件中，但是该文件中的内容用户阅读体验很差，可以去cat /var/log/audit/audit.log阅读，因此系统为我们提供了sealert工具，帮助我们整理该日志文件，sealert工具处理日志文件的时候需要花费一点时间，请耐心等待。另外SELinux的日志功能需要auditd.service服务的开启。

# 强制访问控制实验

### 实验思路

为了解决标准的“用户-组-其他/读-写-执行”权限以及访问控制列表[1]的限制以及加强安全机制，美国国家安全局（NSA）设计出一个灵活的强制访问控制Mandatory Access Control（MAC）方法 SELinux（Security Enhanced Linux 的缩写），来限制标准的权限之外的种种权限，在仍然允许对这个控制模型后续修改的情况下，让进程尽可能以最小权限访问或在系统对象（如文件，文件夹，网络端口等）上执行其他操作。

本实验基于SELinux环境，为了进一步研究类型强制，我们以管理密码的程序（即passwd）为例进行说明，在 Linux中，passwd 程序是可信任的，修改存储经过加密的密码的影子密码文件（/etc/shadow），passwd 程序执行它自己内部的安全策略，允许普通用户修改属于他们自己的密码，同时允许root修改所有密码。为了执行这个受信任的作业，passwd程序需要有移动和重新创建shadow 文件的能力，在标准 Linux 中，它有这个特权，因为passwd程序可执行文件在执行时被加上了 setuid 位，它作为 root 用户（它能访问所有文件）允许，然而，许多程序都可以作为 root 允许（实际上，所有程序都有可能作为 root 允许）。这就意味着任何程序（当以 root 身份运行时）都有可能能够修改 shadow 文件。类型强制使我们能做的事情是确保只有 passwd 程序（或类似的受信任的程序）可以访问 shadow 文件，不管运行程序的用户是谁。

所以我们的实验基本思路在于，创建一个策略文件，创建一个角色名为user\_r可以访问passwd程序，一个角色名为restrict\_r不可以访问passwd程序，并相应创建SELinux用户guest1和guest2，将guest1和user\_r绑定，将guest2和restrict\_r绑定。完成角色和SELinux用户的绑定之后，将相应的Linux登录用户和SELinux用户绑定，以达到切换用户的效果。最后，分别使用两个用户访问passwd程序来观察实验结果，得出结论。

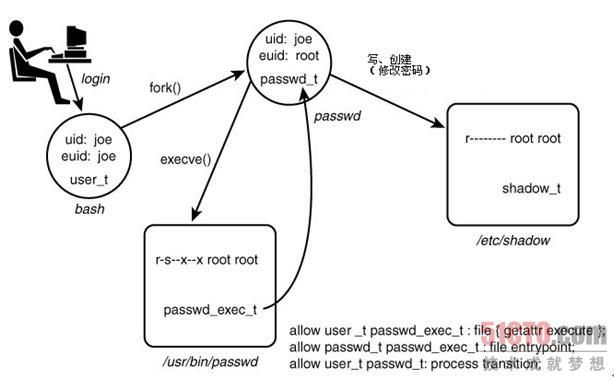


图3 SELinux实验域转变

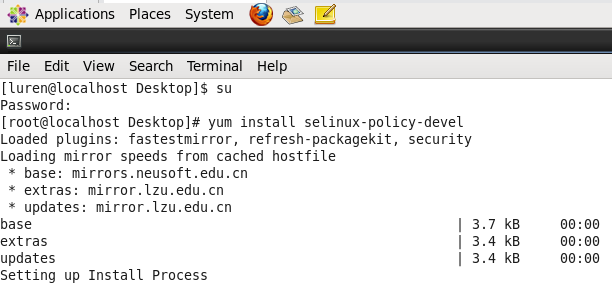
Fig.3 SELinux experimental domain transformation

1. 实验操作

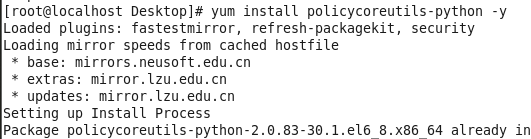
### 安装准备

(1)虚拟机版本CentOS6

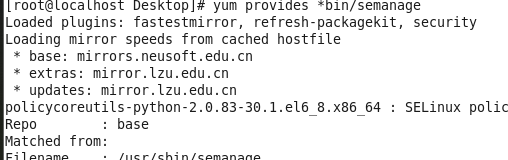
(2)安装SELinux



(3)安装policycoreutils-python**软件包**



(4)安装semanage

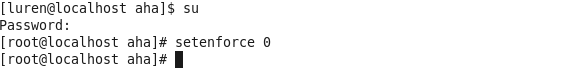


1. 策略编写

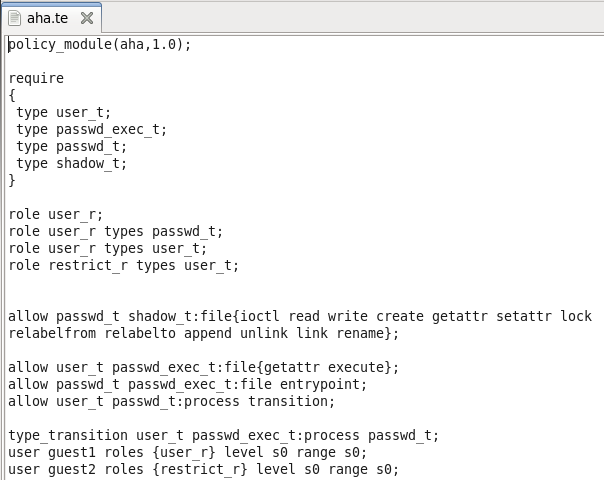
创建Linux用户luren，登录后获取超级用户权限并开始操作。查看SELinux的状态为强制状态。



更改为permissive状态。



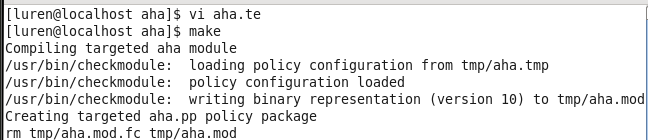
* 1. 创建新文件夹aha。
  2. 在文件夹aha目录下打开终端并创建空策略文件aha.te。
  3. 编辑aha.te策略文件，创建角色user\_r 和restrict\_r，设定user\_r可以访问passwd程序，另一个则不能。创建两个SELinux用户guest1和guest2。用guest1 绑定user\_r，用guest12绑定restrict\_r。



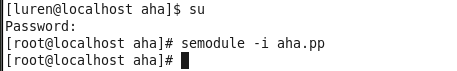
* 1. 在/usr/share/selinux/devel目录下寻找到Makefile文件，并将其复制到文件夹aha下。



* 1. 进行make操作，产生文件aha.pp。

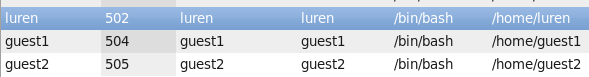


* 1. 挂载aha.pp文件。



1. 用户创建及绑定

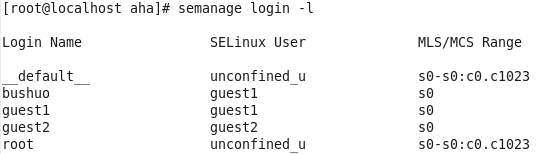
创建两个Linux用户guest1和guest2。



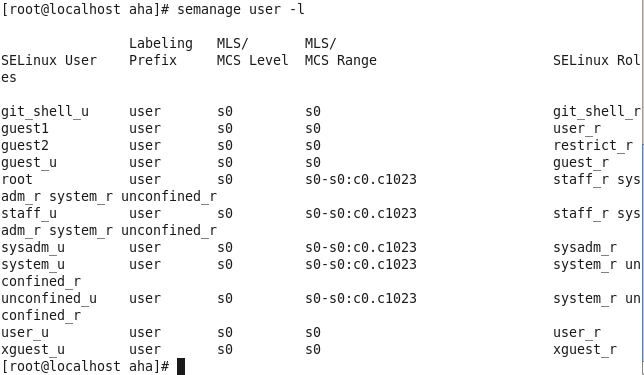
将两个Linux用户和两个SELinux用户进行绑定操作。



查看Linux用户绑定结果，绑定成功。



查看SELinux用户和角色的绑定结果，绑定成功。

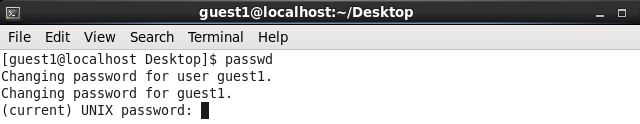


1. 实验结果验证

将SELinux状态转化为enforcing。



切换用户guest1，passwd成功。



切换用户guest2，passwd失败。



# 总 结

通过本次学习实践，我了解到Selinux 总是针对服务和进程出现的，要学好Selinux就不要禁用它，这样在做练习时，才会遇上相关的问题，才能看到Selinux给你解决问题的提示，也只有这样才能学好Selinux。