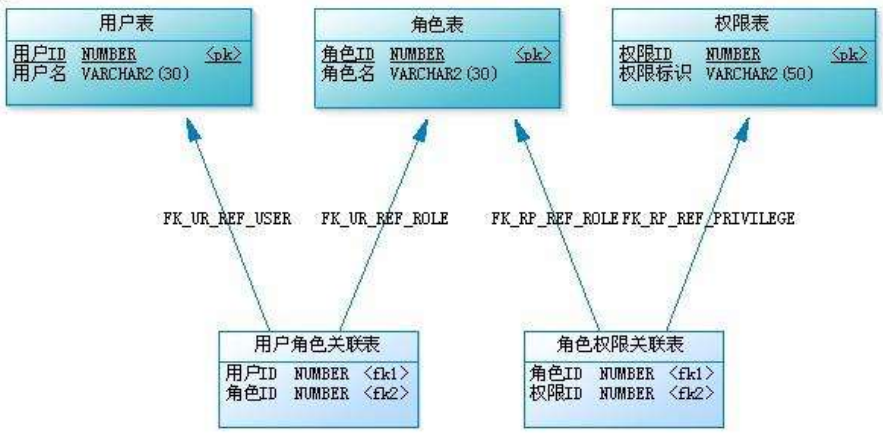
RBAC 访问控制

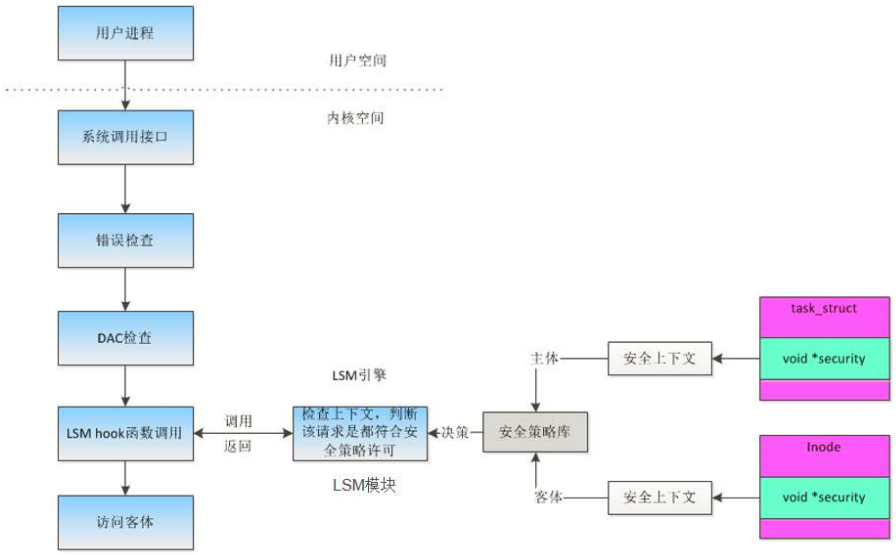
1. Introduction

基于角色的访问控制 RBAC 的基本思想是以角色作为访问控制的主体，通过授权给用访问控制权限。用户不是自始至终以同样的注册身份和劝降访问系统，而是以一定的角色访问，不同的角色被赋予不同的访问权限，系统的访问控制机制只看到角色而看不到用户。用户以什么样的角色对资源进行访问，决定了用户拥有服让权限以及可执行的操作。



2. Methodology

Linux LSM 通过劫持对内核对象的访问进行仲裁，实现通用的强制访问控制框架。每个安全机制只需要按照 LSM 的接口实现具体函数即可，可以自定义安全策略。因此可以实现一个 LSM 安全模块，使得 Linux 具备简单的 RBAC 安全功能。



Linux 安全模块(LSM)目前作为一个 Linux 内核补丁的形式实现。其本身不提供任何具体的安全策略，而是提供了一个通用的基础体系给安全模块，由安全模块来实现具体的安全策略。其主要在五个方面对 Linux 内核进行了修改：(1)在特定的内核数据结构中加入了安全域；(2)在内核源代码中不同的关键点插入了对安全钩子函数的调用；(3)加入了一个通用的安全系统调用；(4)提供了函数允许内核模块注册为安全模块或者注销；(5)将 capabilities 逻辑的大部分移植为一个可选的安全模块。

编写一个基于 LSM 的安全模块的基本流程：(1)确定需要 hook 的函数；(2)对 hook 函数进行填充，添加自己的逻辑(安全检查)；(3)添加到在 security_hook_list 的数据结构里；(4)对这个有注册逻辑的函数进行注册。本次实验中只针对只针对文件创建与文件重命名这 2 个操作提供访问控制，部分思路如下：

```c
int gmlsm_inode_create (struct inode *dir, struct dentry *dentry, umode_t mode)
{
    int uid = current->real_cred->uid.val ;
    printk ("GomoLSM: call [inode_create] by uid: %d\n", uid) ;

    return user_permission (uid, 0) ;
}

int gmlsm_inode_rename (struct inode *old_inode, struct dentry *old_dentry,
            struct inode *new_inode, struct dentry *new_dentry)
{
    int uid = current->real_cred->uid.val ;
    printk ("GomoLSM: call [inode_rename] by uid: %d\n", uid) ;

    return user_permission (uid, 1) ;
}

static struct security_hook_list gmlsm_hooks[] = {
    LSM_HOOK_INIT(inode_rename,gmlsm_inode_rename),
    LSM_HOOK_INIT(inode_create,gmlsm_inode_create),
};

void __init gmlsm_add_hooks(void)
{
    pr_info("GomoLSM: LSM LOADED.\n");
    security_add_hooks(gmlsm_hooks, ARRAY_SIZE(gmlsm_hooks));
}

static __init int gmlsm_init(void){
    gmlsm_add_hooks();
    return 0;
}

security_initcall(gmlsm_init);
```

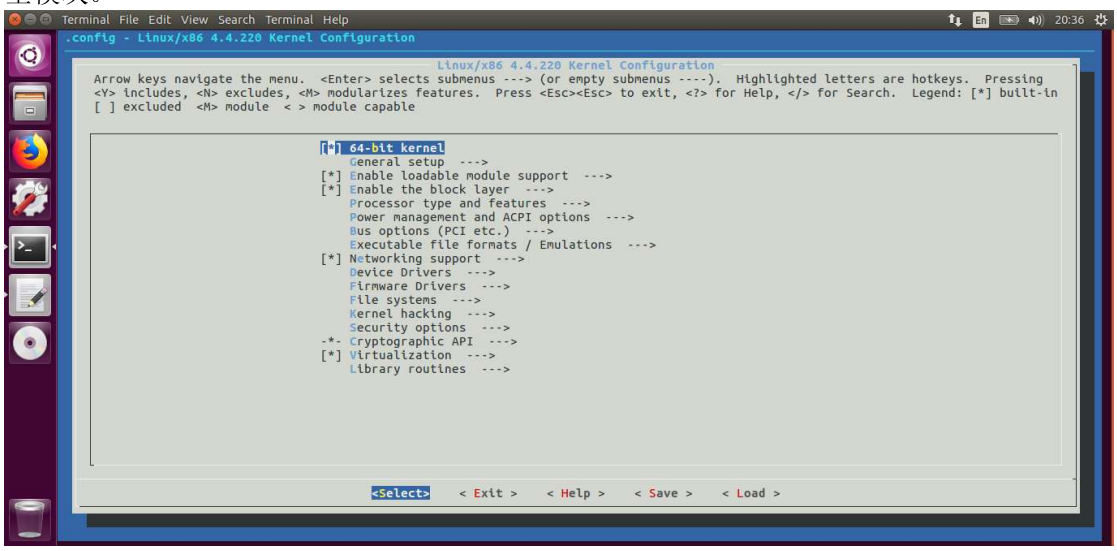
3. Implementation details

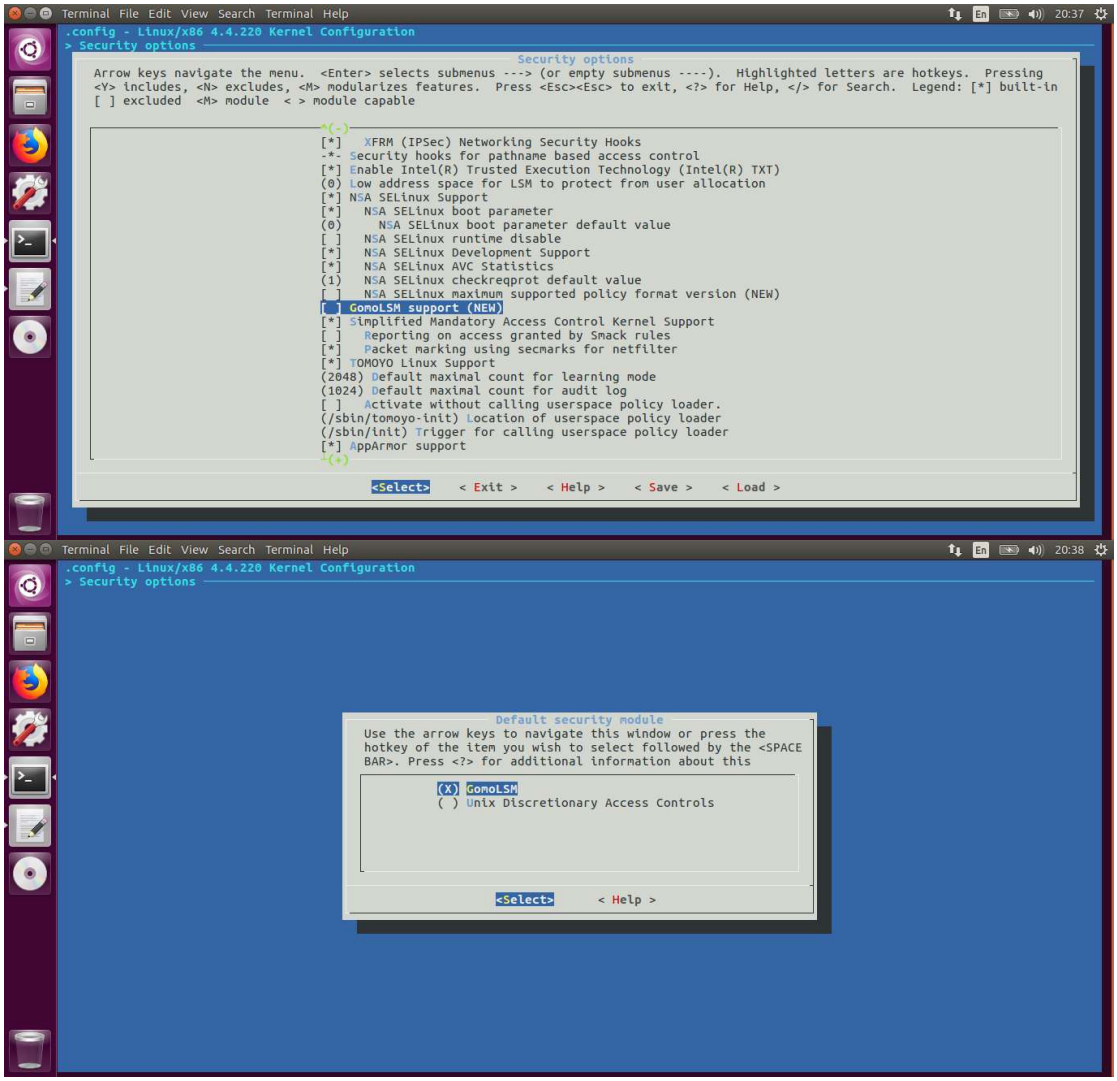
3.1 编译 LSM 安全模块

自定义的 LSM 安全模块具体源码见附件中的 GomoLSM.c 文件，主要包括 get_role、role_permission、user_permission 函数以及模块开关函数和相关 hook 函数的操作。在内核 2.6.x 后，LSM 模块不再运行动态加载到内核，而需要编写 Makefile 以及 Kconfig 文件将模块编译进内核，编写格式可以参照 SElinux 或者 yama 的格式进行编写。

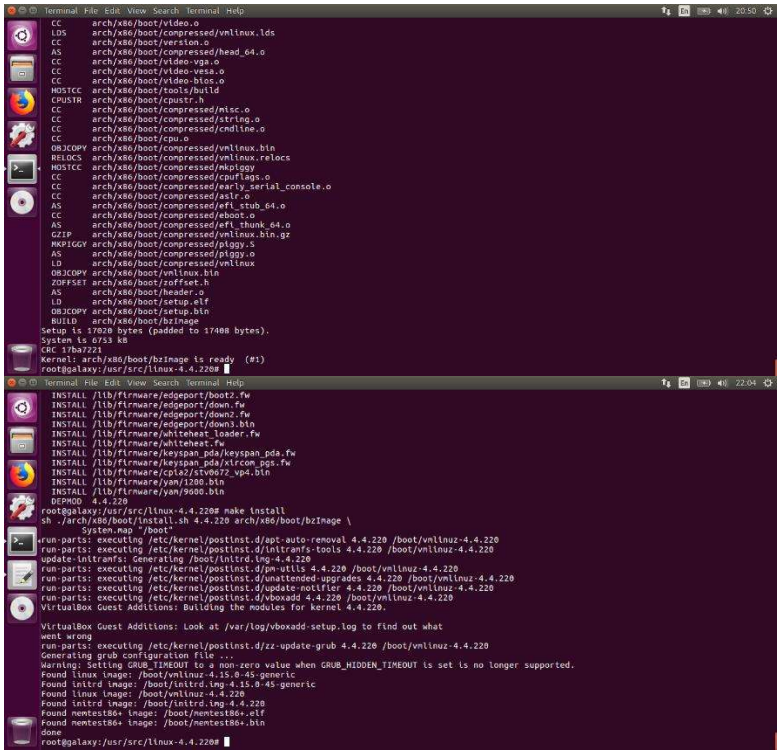
本次实验在 VirtualBox 虚拟机上进行，Ubuntu 版本为 16.04LTS，使用的内核为 linux-4.4.220，编译过程大概如下：

(1)将/boot/config -’uname’ -r 拷贝到 linux 内核源码根目录中，并重命名为.config，然后 make menuconfig 即可进行配置编译选项，对 LSM 而言，进入 Security options，将该模块选上，而将 SELinux、TOMOYO、Yama 等其他安全模块都取消，再将该模块选为默认安全模块。





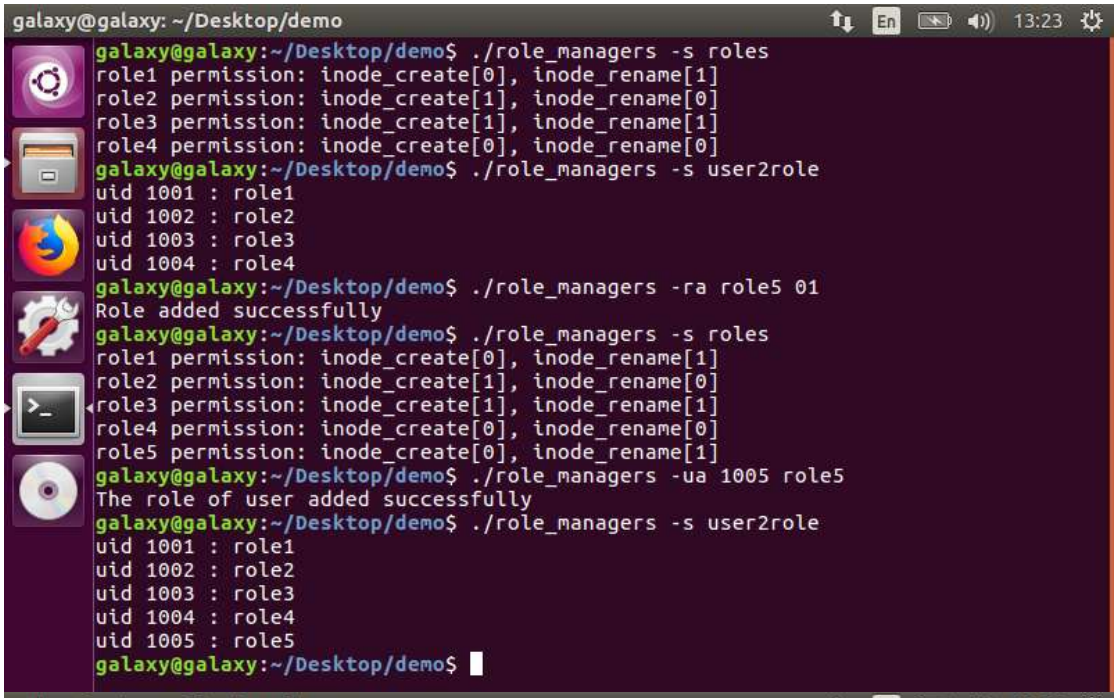
(2)接着进行内核编译安装，修改 grub 后重启，新的安全模块即可生效。

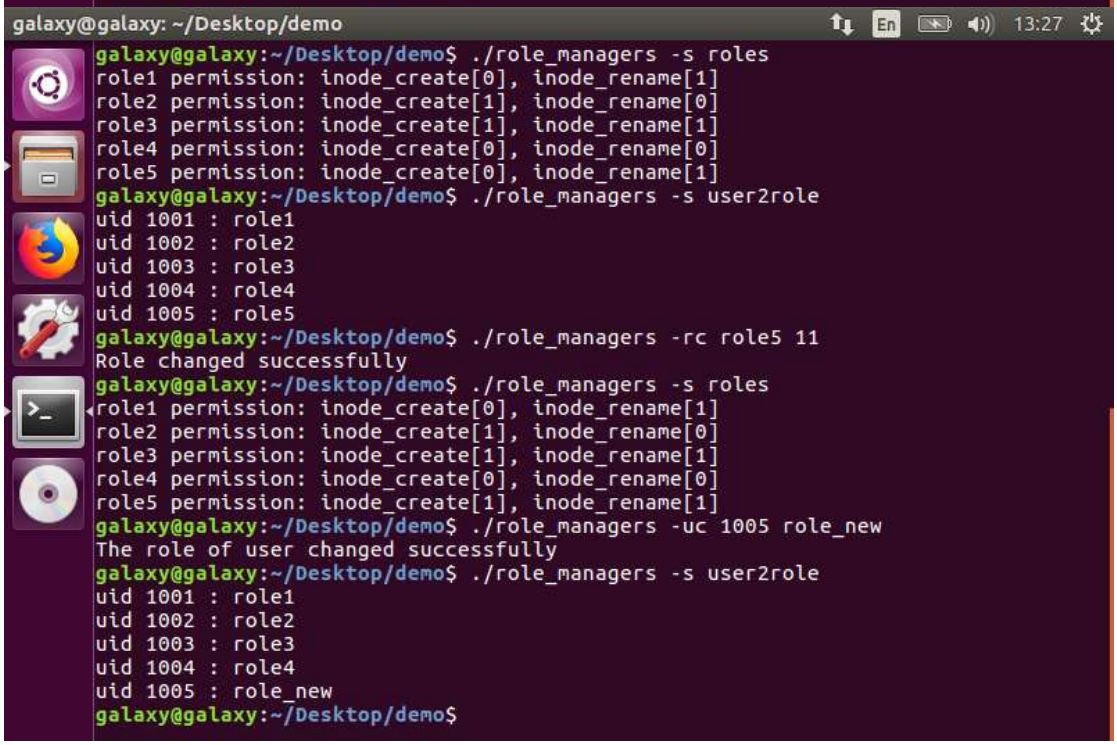


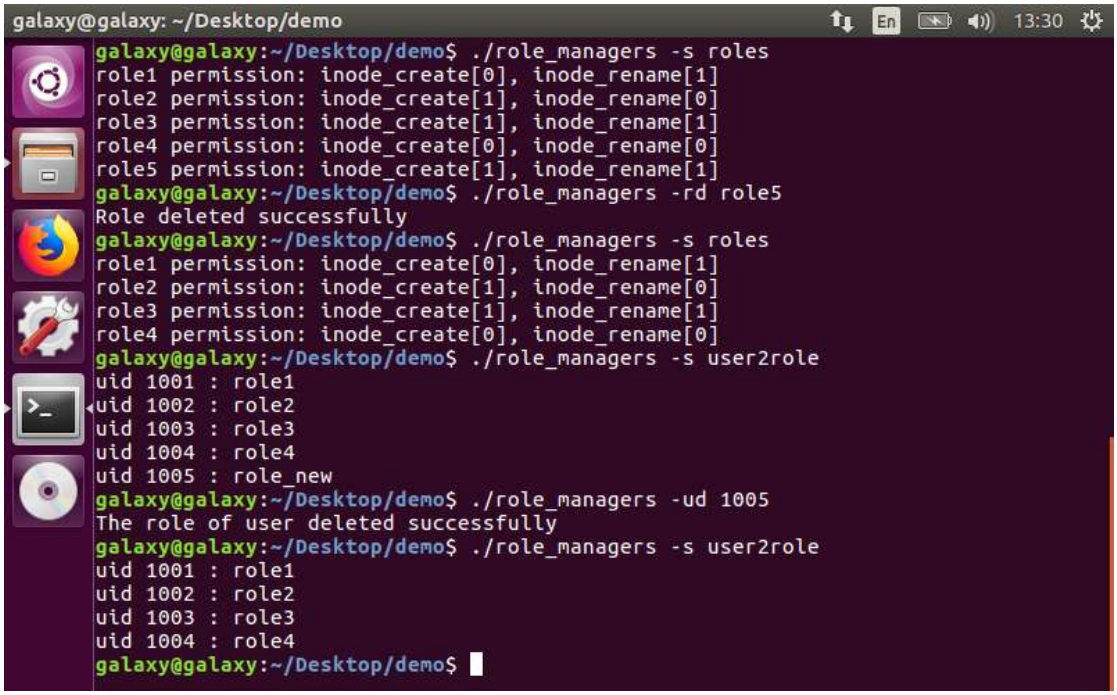
3.2 角色和权限管理

role_manager.c 文件用于对角色和权限进行简单的管理操作，主要包括创建、删除、修改角色、权限，以及 LSM 的模块开关。通过 UID 实现用户与角色、角色与权限之间的关联，在用户进行相关访问操作时，系统的 LSM 安全模块会根据当前用户的角色，以及自定义的安全策略对用户行为进行访问控制。

角色管理程序的设计具体见附件，控制权限只针对文件创建与文件重命名这 2 个操作提供访问控制，为 0 表示不允许，为 1 表示允许；以及模块启用与关闭，如果为 Disable，该模块不干预访问控制，当为 Enable 时才干预访问控制。大致的创建、删除、修改功能如下：



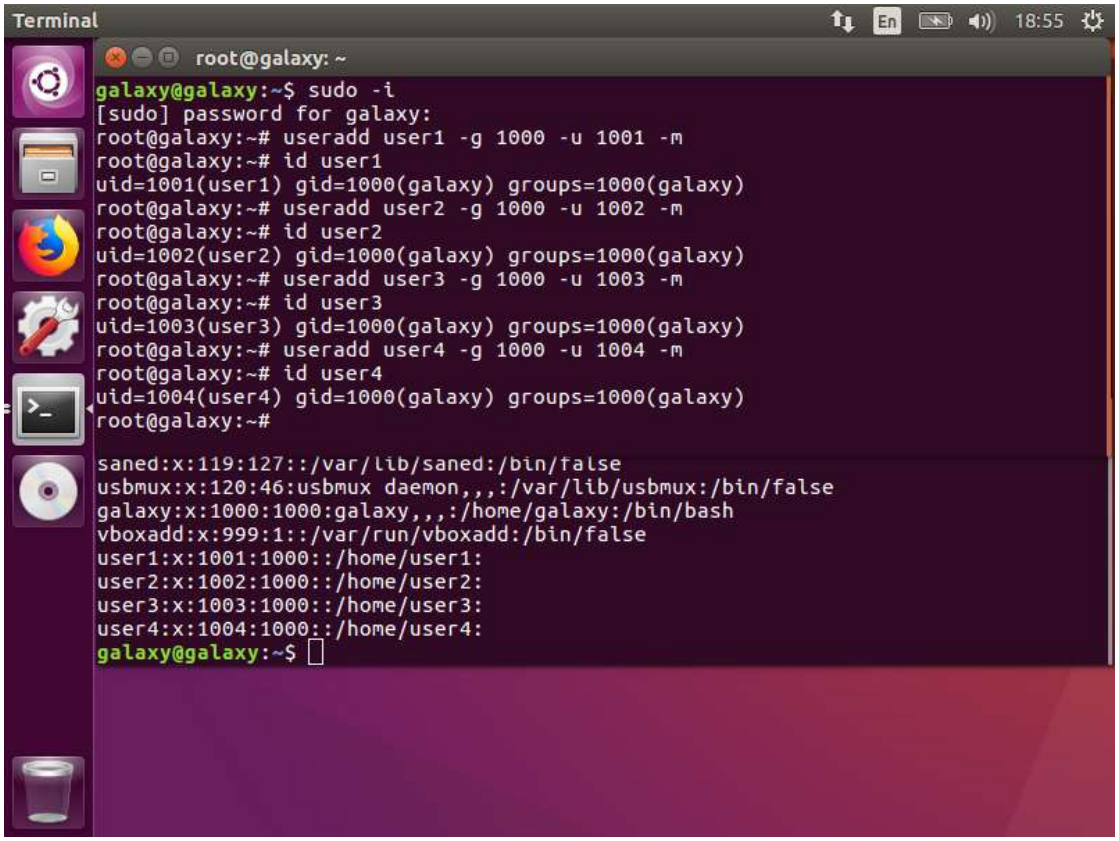




3.3 用户访问控制

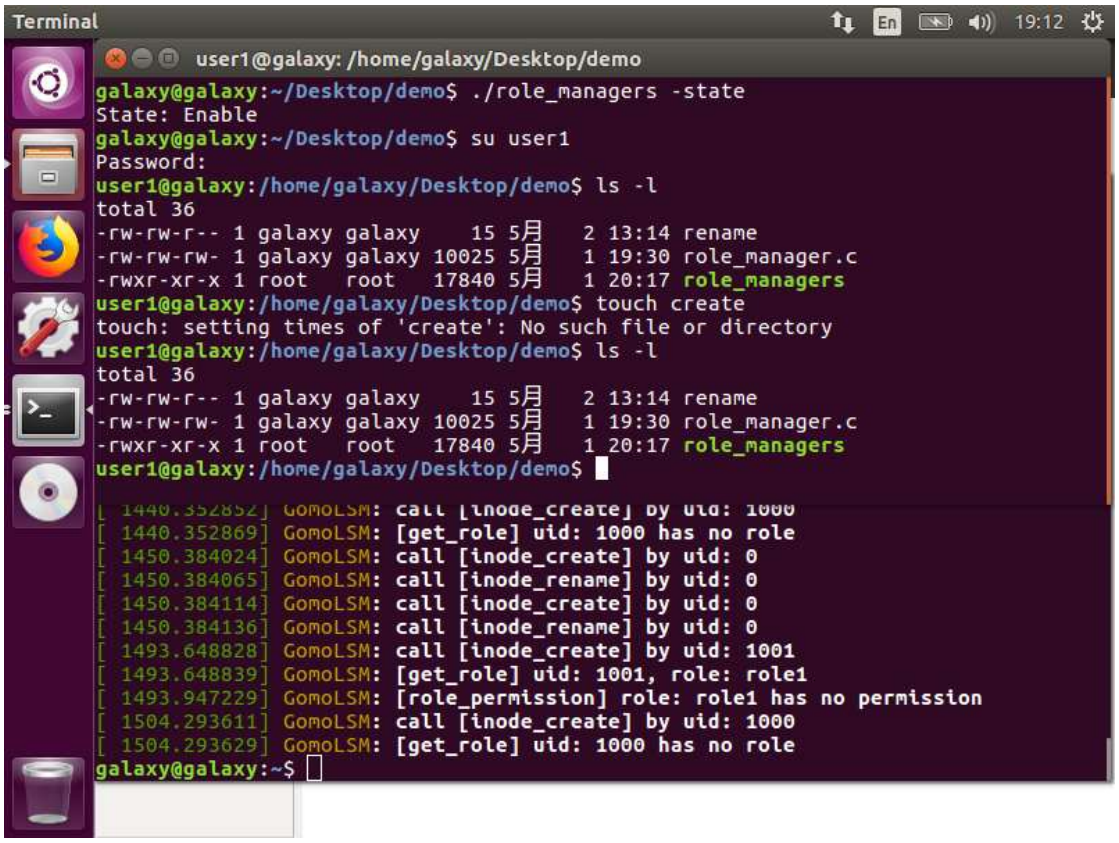
LSM 安全模块以及角色管理上文已经阐述，接下来创建不同的用户以验证 LSM 安全模块是否起到对基本访问控制的作用。

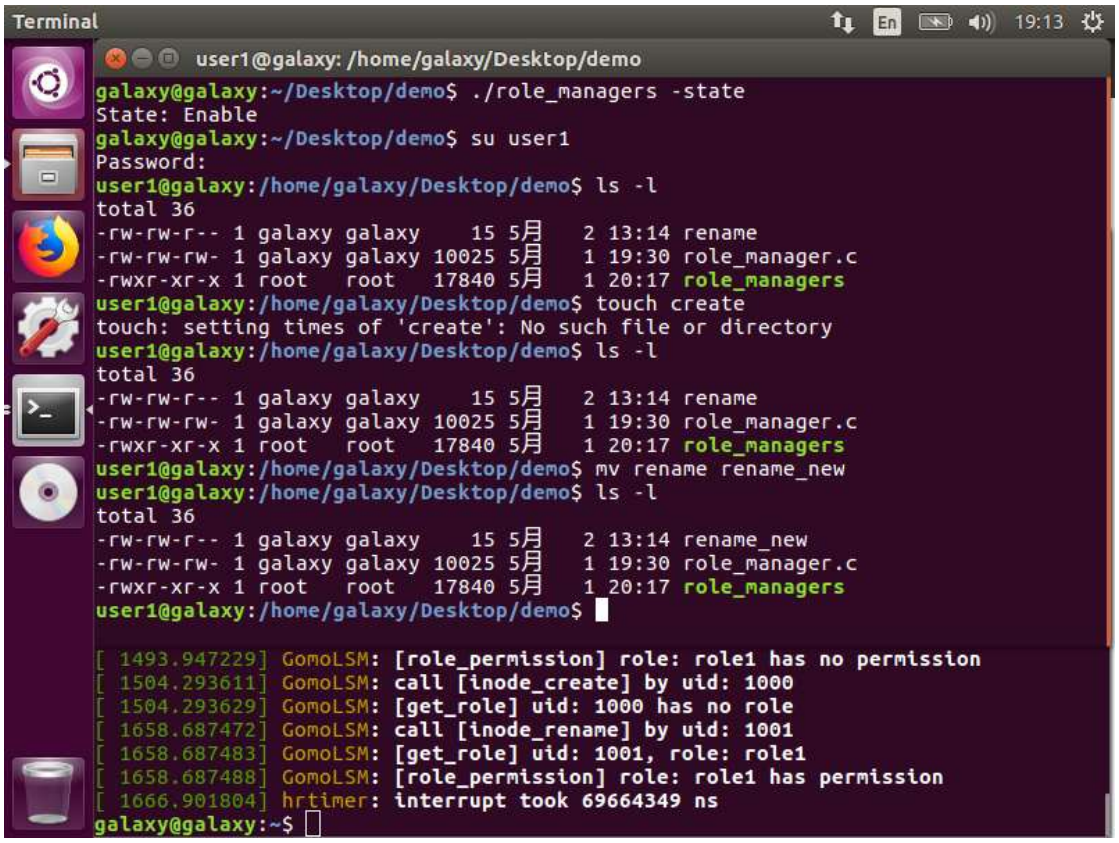
实验中添加了 user1-4 共个用户，依次对应 role1-4 共 4 个角色。其中 4 个角色之前已添加，添加用户过程大致如下：



role1-4 的权限依次为 01、10、11、00(对应文件创建与文件重命名权限)，role1-4 与 user1-4 的关联 UID 依次为 1001-1004。要创建的文件为 create，要重命名的文件为 rename。验证过程大致如下：

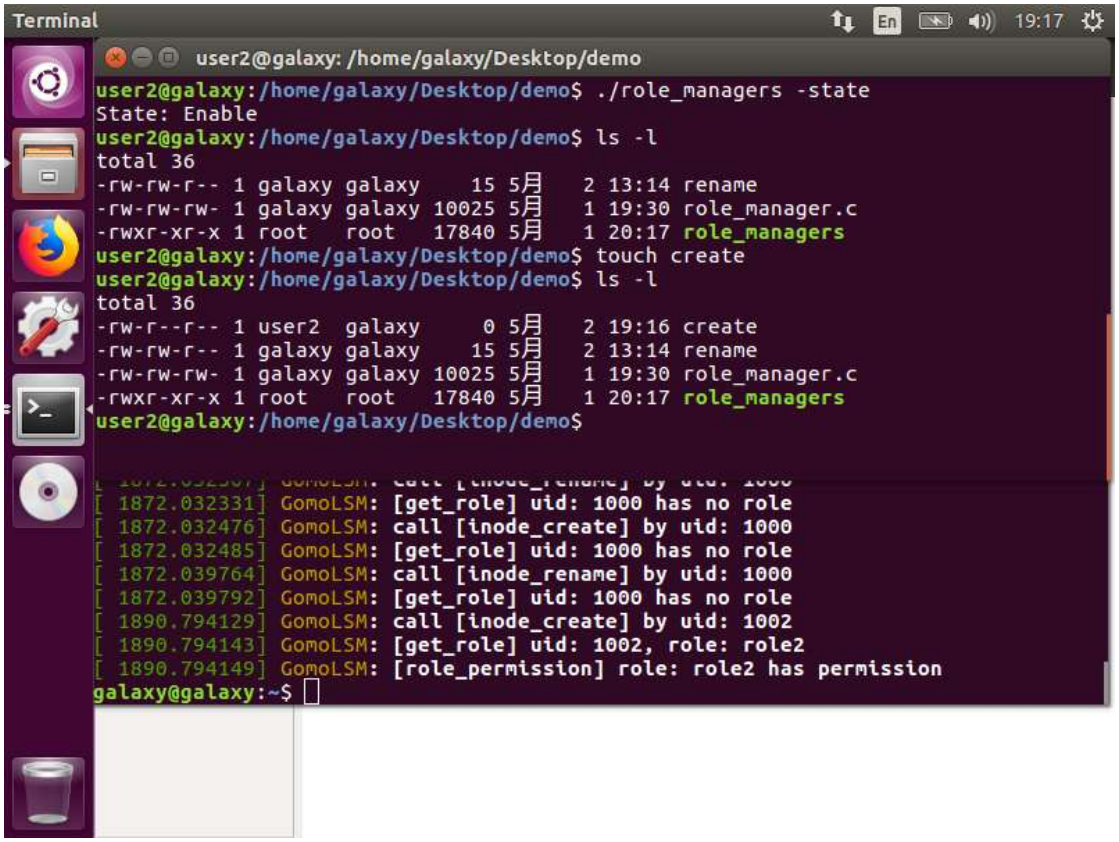
(1)用户 user1 对应的角色为 role1，没有创建文件权限而拥有文件重命名权限，验证结果如下：

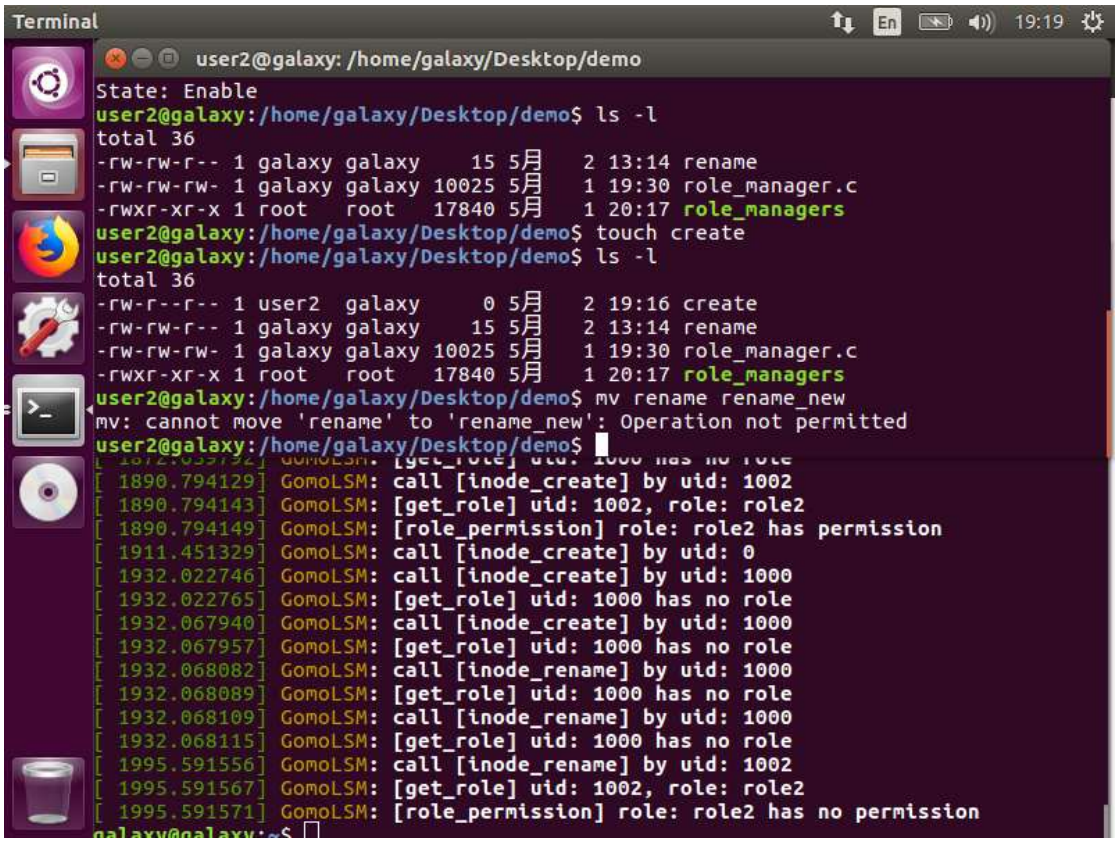




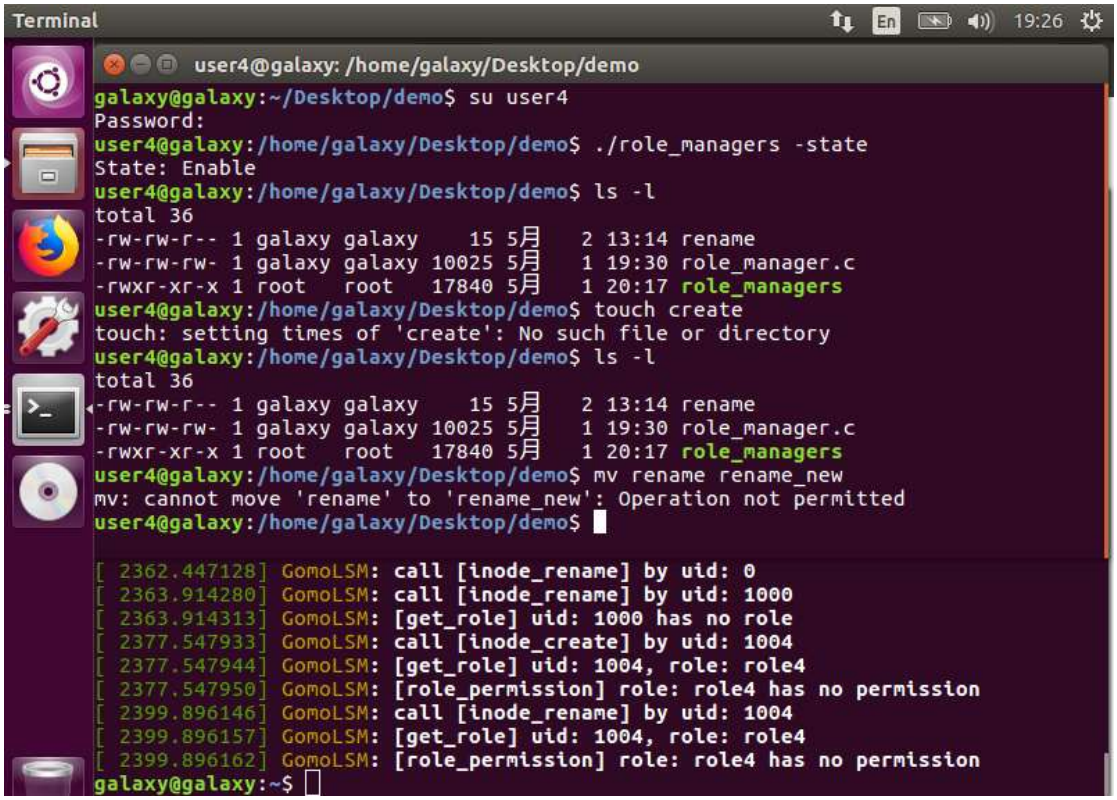
(2)用户 user2 对应的角色为 role2，拥有创建文件权限而没有文件重命名权限，验证

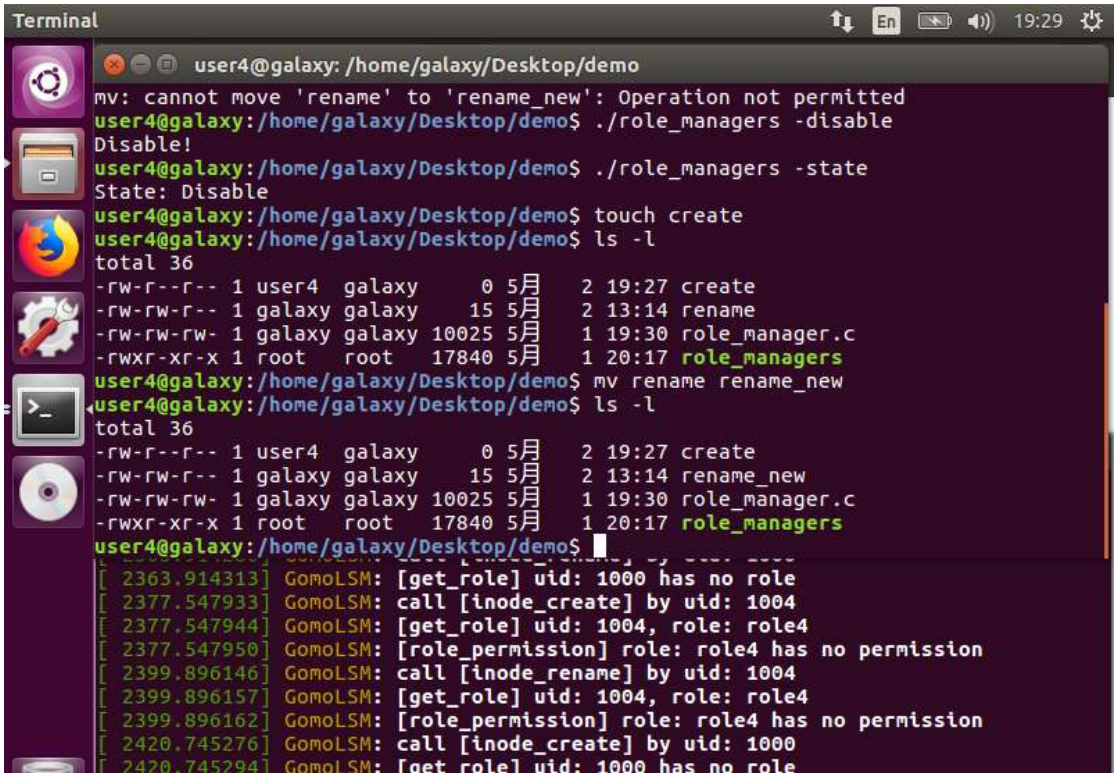
结果如下：





(3)用户 user4 对应的角色为 role4，没有创建文件权限而没有文件重命名权限，关闭 LSM 安全模块后，模块停止运行，2 种权限不受限制，验证结果如下：


Terminal
En
19:26
user4@galaxy: /home/galaxy/Desktop/demo
galaxy@galaxy:~/Desktop/demo$ su user4
Password:
user4@galaxy:/home/galaxy/Desktop/demo$ ./role_managers -state
State: Enable
user4@galaxy:/home/galaxy/Desktop/demo$ ls -l
total 36
-rw-rw-r-- 1 galaxy galaxy   15 5月   2 13:14 rename
-rw-rw-rw- 1 galaxy galaxy 10025 5月   1 19:30 role_manager.c
-rwxr-xr-x 1 root   root   17840 5月   1 20:17 role_managers
user4@galaxy:/home/galaxy/Desktop/demo$ touch create
touch: setting times of 'create': No such file or directory
user4@galaxy:/home/galaxy/Desktop/demo$ ls -l
total 36
-rw-rw-r-- 1 galaxy galaxy   15 5月   2 13:14 rename
-rw-rw-rw- 1 galaxy galaxy 10025 5月   1 19:30 role_manager.c
-rwxr-xr-x 1 root   root   17840 5月   1 20:17 role_managers
user4@galaxy:/home/galaxy/Desktop/demo$ mv rename rename_new
mv: cannot move 'rename' to 'rename_new': Operation not permitted
user4@galaxy:/home/galaxy/Desktop/demo$
[2362.447128] GomoLSM: call [inode_rename] by uid: 0
[2363.914280] GomoLSM: call [inode_rename] by uid: 1000
[2363.914313] GomoLSM: [get_role] uid: 1000 has no role
[2377.547933] GomoLSM: call [inode_create] by uid: 1004
[2377.547944] GomoLSM: [get_role] uid: 1004, role: role4
[2377.547950] GomoLSM: [role_permission] role: role4 has no permission
[2399.896146] GomoLSM: call [inode_rename] by uid: 1004
[2399.896157] GomoLSM: [get_role] uid: 1004, role: role4
[2399.896162] GomoLSM: [role_permission] role: role4 has no permission
galaxy@galaxy:~$


Terminal
En
19:29
user4@galaxy: /home/galaxy/Desktop/demo
mv: cannot move 'rename' to 'rename_new': Operation not permitted
user4@galaxy:/home/galaxy/Desktop/demo$ ./role_managers -disable
Disable!
user4@galaxy:/home/galaxy/Desktop/demo$ ./role_managers -state
State: Disable
user4@galaxy:/home/galaxy/Desktop/demo$ touch create
user4@galaxy:/home/galaxy/Desktop/demo$ ls -l
total 36
-rw-r--r-- 1 user4 galaxy   0 5月   2 19:27 create
-rw-rw-r-- 1 galaxy galaxy   15 5月   2 13:14 rename
-rw-rw-rw- 1 galaxy galaxy 10025 5月   1 19:30 role_manager.c
-rwxr-xr-x 1 root   root   17840 5月   1 20:17 role_managers
user4@galaxy:/home/galaxy/Desktop/demo$ mv rename rename_new
user4@galaxy:/home/galaxy/Desktop/demo$ ls -l
total 36
-rw-r--r-- 1 user4 galaxy   0 5月   2 19:27 create
-rw-rw-r-- 1 galaxy galaxy   15 5月   2 13:14 rename_new
-rw-rw-rw- 1 galaxy galaxy 10025 5月   1 19:30 role_manager.c
-rwxr-xr-x 1 root   root   17840 5月   1 20:17 role_managers
user4@galaxy:/home/galaxy/Desktop/demo$
[2363.914313] GomoLSM: [get_role] uid: 1000 has no role
[2377.547933] GomoLSM: call [inode_create] by uid: 1004
[2377.547944] GomoLSM: [get_role] uid: 1004, role: role4
[2377.547950] GomoLSM: [role_permission] role: role4 has no permission
[2399.896146] GomoLSM: call [inode_rename] by uid: 1004
[2399.896157] GomoLSM: [get_role] uid: 1004, role: role4
[2399.896162] GomoLSM: [role_permission] role: role4 has no permission
[2420.745276] GomoLSM: call [inode_create] by uid: 1000