

Лабораторная работа №5

Валиева Найля Разимовна - студентка группы НКНбд-01-18

13.11.2021

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

- Помимо прав администратора в некоторых случаях требуются средства разработки приложений. Для разграничения их прав нужно использовать дополнительные атрибуты.

- Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов. Получение практических навыков работы в консоли с дополнительными атрибутами. Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

- Подготовить лабораторный стенд
- Рассмотреть компиляцию программ
- Создать программы
- Исследовать Sticky-бит

Результаты выполнения лабораторной работы

- Подготовила лабораторный стенд (рис - @fig:001).

```
Зависимости установлены:
  cloog-ppl.i686 0:0.15.7-1.2.el6          cpp.i686 0:4.4.7-23.el6
  mpfr.i686 0:2.4.1-6.el6                 ppl.i686 0:0.10.2-11.el6

Готово!
[root@nrvalieva Рабочий стол]# gcc -v
Используются внутренние спецификации.
Целевая архитектура: i686-redhat-linux
Параметры конфигурации: ../configure --prefix=/usr --mandir=/usr/share/man --inf
odir=/usr/share/info --with-bugurl=http://bugzilla.redhat.com/bugzilla --enable-
bootstrap --enable-shared --enable-threads=posix --enable-checking=release --wit
h-system-zlib --enable-_cxa_atexit --disable-libunwind-exceptions --enable-gnu-
unique-object --enable-languages=c,c++,objc,obj-c++,java,fortran,ada --enable-ja
va-awt=gtk --disable-dssi --with-java-home=/usr/lib/jvm/java-1.5.0-gcj-1.5.0.0/j
re --enable-libgcj-multifile --enable-java-maintainer-mode --with-ecj-jar=/usr/s
hare/java/eclipse-ecj.jar --disable-libjava-multilib --with-ppl --with-cloog --w
ith-tune=generic --with-arch=i686 --build=i686-redhat-linux
Модель многократности: posix
gcc версия 4.4.7 20120313 (Red Hat 4.4.7-23) (GCC)
[root@nrvalieva Рабочий стол]# setenforce 0
[root@nrvalieva Рабочий стол]# getenforce
Permissive
[root@nrvalieva Рабочий стол]#
```

Рис. 1: Установка компилятора gcc

- Рассмотрела компиляцию программ (рис @fig:002).

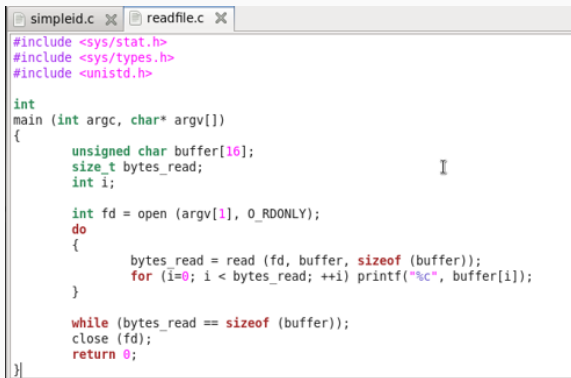
```
[root@nrvalieva Рабочий стол]# whereis gcc
gcc: /usr/bin/gcc /usr/lib/gcc /usr/libexec/gcc /usr/share/man/man1/gcc.1.gz
[root@nrvalieva Рабочий стол]# whereis g++
g++:
[root@nrvalieva Рабочий стол]#
```

Рис. 2: Проверка названий компиляторов

- Провела работу с программами (рис @fig:003, рис @fig:004, рис @fig:005).

```
[root@nrvalieva Рабочий стол]# chown root:guest simpleid2
[root@nrvalieva Рабочий стол]# ls
simpleid simpleid2 simpleid2.c simpleid.c~
[root@nrvalieva Рабочий стол]# chmod u+s simpleid2
[root@nrvalieva Рабочий стол]# ls -l
итого 24
-rwxrwxr-x. 1 guest guest 4890 Ноя 13 21:17 simpleid
-rwsrwxr-x. 1 root  guest 4971 Ноя 13 21:21 simpleid2
-rw-rw-r--. 1 guest guest  315 Ноя 13 21:20 simpleid2.c
-rw-rw-r--. 1 guest guest  180 Ноя 13 21:17 simpleid.c~
[root@nrvalieva Рабочий стол]# █
```

Рис. 3: Смена владельца и атрибутов от имени суперпользователя



```
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

int
main (int argc, char* argv[])
{
    unsigned char buffer[16];
    size_t bytes_read;
    int i;

    int fd = open (argv[1], O_RDONLY);
    do
    {
        bytes_read = read (fd, buffer, sizeof (buffer));
        for (i=0; i < bytes_read; ++i) printf("%c", buffer[i]);
    }

    while (bytes_read == sizeof (buffer));
    close (fd);
    return 0;
}
```

Рис. 4: Создание программы readfile.c

```
[root@nrvalieva Рабочий стол]# ./readfile readfile.c
#include <fcntl.h>
#include <stdio.h>
#include <sys/stat.h>
#include <sys/types.h>
#include <unistd.h>

int
main (int argc, char* argv[])
{
    unsigned char buffer[16];
    size_t bytes_read;
    int i;

    int fd = open (argv[1], O_RDONLY);
    do
    {
        bytes_read = read (fd, buffer, sizeof (buffer));
        for (i=0; i < bytes_read; ++i) printf("%c", buffer[i]);
    }

    while (bytes_read == sizeof (buffer));
    close (fd);
}
```

Рис. 5: Проверка чтения файла

- Исследовала Sticky-бит (рис @fig:006, рис @fig:007, рис @fig:008).

```
guest@nrvalieva /]$ echo "test" > /tmp/file01.txt
```

Рис. 6: Создание файла и внесение записи в него

```
[guest2@nrvalieva ~]$ rm /tmp/file01.txt  
rm: невозможно удалить «/tmp/file01.txt»: Операция не допускается
```

Рис. 7: Попытка удаления файла от имени пользователя **guest2**

```
[guest2@nrvalieva ~]$ ls -l / | grep tmp
drwxrwxrwx. 24 root root 4096 Ноя 13 21:37 tmp
[guest2@nrvalieva ~]$ echo "test" > /tmp/file01.txt
[guest2@nrvalieva ~]$ cat /tmp/file01.txt
test
[guest2@nrvalieva ~]$ echo "test2" >> /tmp/file01.txt
[guest2@nrvalieva ~]$ cat /tmp/file01.txt
test
test2
[guest2@nrvalieva ~]$ echo "test3" > /tmp/file01.txt
[guest2@nrvalieva ~]$ cat /tmp/file01.txt
test3
[guest2@nrvalieva ~]$ rm /tmp/file01.txt
[guest2@nrvalieva ~]$ █
```

Рис. 8: Проведение различных операций после удаления атрибута t

Таким образом, я изучила механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получила практические навыки работы в консоли с дополнительными атрибутами. Рассмотрела работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.