Лабораторная работа № 5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Кекишева Анастасия Дмитриевна

Содержание

# Цель работы

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.

# Задание

1. Выполнить последовательность дейсвий, указанных в лабораторной работе [@bib1], создавая программы и работая с битами (SetUID, SetGID, Sticky-бит), чтобы изучить влияние дополнительных атрибутов.

# Теоретическое введение

Рассмотрим некоторые команды, которые пригодяться нам в данной лабораторной.

* chown [ПАРАМЕТР]… [ВЛАДЕЛЕЦ][:[ГРУППА]] ФАЙЛ… Эта команда позволяет сменить владельца и группу указанного ФАЙЛА на ВЛАДЕЛЬЦА и/или ГРУППУ [@bib3].
* gcc [ИМЯ\_ФАЙЛА].c -o [ИМЯ\_ПРОГРАММЫ] Это команда поможет нам конвертировать файлы [@bib1].

Рассмотрим биты, с которыми мы будем работать.

Setuid – это бит разрешения, который позволяет пользователю запускать исполняемый файл с правами владельца этого файла. Другими словами, использование этого бита позволяет нам поднять привилегии пользователя в случае, если это необходимо. Классический пример использования этого бита в операционной системе это команда sudo [@bib2].

root@ruvds-hrc [~]# which sudo /usr/bin/sudo root@ruvds-hrc [~]# ls -l /usr/bin/sudo -rwsr-xr-x 1 root root 125308 Feb 20 14:15 /usr/bin/sudo

Как мы видим на месте, где обычно установлен классический бит x (на исполнение), у нас выставлен специальный бит s. Это позволяет обычному пользователю системы выполнять команды с повышенными привилегиями без необходимости входа в систему как root, разумеется зная пароль пользователя root. Установка бита setuid не представляет сложности. Для этого используется команда:

root@ruvds-hrc [~]# chmod u+s

Аналогично setuid, бит setgid выставляется с помощью команды chmod g + s.

-rwxr-sr-x 1 root root 125308 Feb 20 14:15 /usr/bin/sudo

Последний специальный бит разрешения – это Sticky Bit . В случае, если этот бит установлен для папки, то файлы в этой папке могут быть удалены только их владельцем. Пример использования этого бита в операционной системе это системная папка /tmp . Эта папка разрешена на запись любому пользователю, но удалять файлы в ней могут только пользователи, являющиеся владельцами этих файлов [@bib2].

root@ruvds-hrc [~]# ls -ld /tmp drwxrwxrwt 8 root root 4096 Mar 25 10:22 /tmp

Символ «t» указывает, что на папку установлен Sticky Bit.

# Выполнение лабораторной работы

## Изучение механики SetUID

1. Вошла в систему от имени пользователя guest1. И создайте программу simpleid.c, прежде создав файл, затем записав в него код программы (рис. @fig:001).

|  |
| --- |
| Создание программы simpleid.c |

Создание программы simpleid.c

1. Скомпилировала программу командой gcc simpleid2.c -o simpleid2 и запустила simpleid2.c (рис. @fig:002).

|  |
| --- |
| Компиляция и запуск программы |

Компиляция и запуск программы

1. Выполнила системную программу id. Сравнивая результаты выполнения команды ./simpleid и id, можно сказать что программа вывела только групповое и личное id, в id добавилость ещё одно групповое id (рис. @fig:002).
2. Усложнила программу, добавив вывод действительных идентификаторов (рис. @fig:003).  
   Создав программу simpleid2.c, простотрела её, командой cat искомпилировала её (рис. @fig:004).

|  |
| --- |
| Написание программы simpleid2.c |

Написание программы simpleid2.c

|  |
| --- |
| Компиляция и запуск программы simpleid2 |

Компиляция и запуск программы simpleid2

1. От имени суперпользователя выполнила команды:

* chown root:guest1 /home/guest1/simpleid2, которая поменяла владельца программы на root и сделала так, что этот файл принадлежит группе guest1.
* chmod u+s /home/guest/simpleid2 назначила права доступа, которые значат, что пользователь выполняет файл с разрешениями владельца файла (рис. @fig:005).

|  |
| --- |
| Компиляция и запуск программы |

Компиляция и запуск программы

1. Далее выполнила проверку правильности установки новых атрибутов и смены владельца файла simpleid2 командой ls -l simpleid2 и запустила simpleid2 и id: Результаты сравнения: помимо индетифткаторов юзера и группы, программа вывела нам также текущие индетификаторы (рис. @fig:005).
2. Проделала тоже самое относительно SetGID-бита – для этого выполнила команду с атрибутом g: chmod g+s /home/guest/simpleid2 (рис. @fig:006).

|  |
| --- |
| Компиляция и запуск программы |

Компиляция и запуск программы

1. Создала программу readfile.c и откомпиилировала её (рис. @fig:007).

|  |
| --- |
| Создание и компиляция программы readfile.c |

Создание и компиляция программы readfile.c

1. Смените владельца у файла readfile.c на root, настроила также и группу root, измените права так, чтобы только суперпользователь (root) мог прочитать его, a guest1 и другие не могли. Смените у программы readfile владельца и установите SetU’D-бит. Проверила, что пользователь guest1 не может прочитать файл readfile.c. А также проверила, может ли программа readfile прочитать файл readfile.c – да может (рис. @fig:008).

|  |
| --- |
| Настройка прав для файла readfile.c |

Настройка прав для файла readfile.c

1. Проверила, что программа readfile может прочитать файл /etc/shadow (рис. @fig:009).

|  |
| --- |
| Чтение файла /etc/shadow с помощью программы readfile |

Чтение файла /etc/shadow с помощью программы readfile

## Исследование Sticky-бита

1. Выяснила, установлен ли атрибут Sticky на директории /tmp, для чего выполнила команду ls -l / | grep tmp (рис. @fig:010).
2. От имени пользователя guest1 создала файл file01.txt в директории /tmp со словом test: echo “test” > /tmp/file01.txt (рис. @fig:010).

|  |
| --- |
| Проверка атрибута sticky и создание файла |

Проверка атрибута sticky и создание файла

1. Просмотрела атрибуты (рис. @fig:011) у только что созданного файла и разрешила чтение и запись для категории пользователей «все остальные»:

* ls -l /tmp/file01.txt
* chmod o+rw /tmp/file01.txt
* ls -l /tmp/file01.txt

|  |
| --- |
| Добавление прав остальным пользователям на чтение и запись |

Добавление прав остальным пользователям на чтение и запись

1. От пользователя guest2 (не являющегося владельцем) попробовала прочитать файл /tmp/file01.txt и дозаписать в него. Дозаписать удалось, но там остался лишь новый текст, старого test не было (рис. @fig:012).

|  |
| --- |
| Проверка атрибута sticky и создание файла |

Проверка атрибута sticky и создание файла

1. От пользователя guest2 попробовала записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой echo “test3” > /tmp/file01.txt – операция была выполнена успешно (рис. @fig:012).
2. Проверьте содержимое файла и от пользователя guest2 попробовала удалить файл – файл удалить не удалось, нет прав на это. Далее повысила свои права до суперпользователя следующей командой su - и сняла атрибут t (Sticky-бит) с директории /tmp: chmod -t /tmp. Покинула режим суперпользователя командой exit. От пользователя guest2 проверила, что атрибута t у директории /tmp нет: ls -l / | grep tmp (рис. @fig:013).
3. Повторите предыдущие шаги – в этот раз смогла удалить файл (рис. @fig:013).  
   После выполнения снова довавила sticky-бит.

|  |
| --- |
| Шаги без sticky-бита |

Шаги без sticky-бита

# Выводы

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

# Список литературы