Отчет по лабораторной работе 5

Дисциплина: Информационная безопасность

Савченков Дмитрий Андреевич, НПИбд-02-18

Содержание

[Цель работы 1](#_Toc87630280)

[Выполнение лабораторной работы 1](#_Toc87630281)

[Выводы 9](#_Toc87630282)

[Список литературы 9](#_Toc87630283)

# 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.

# Выполнение лабораторной работы

1. Создание программы

1.1. Вошел в систему от имени пользователя guest.

1.2. Создал программу *simpleid.c* по шаблону из методички. (рис. 1)

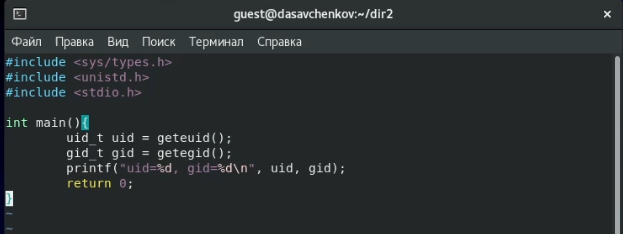


Figure 1: Программа simpleid.c

1.3. Скомплилировал программу и убедилась, что файл программы создан: gcc simpleid.c -o simpl. (рис. 2)

1.4. Выполнил программу simpleid: ./simpl. (рис. 2)

1.5. Выполнил системную программу id: id. (рис. 2) Полученный мной результат совпадает с данными предыдущего пункта задания.

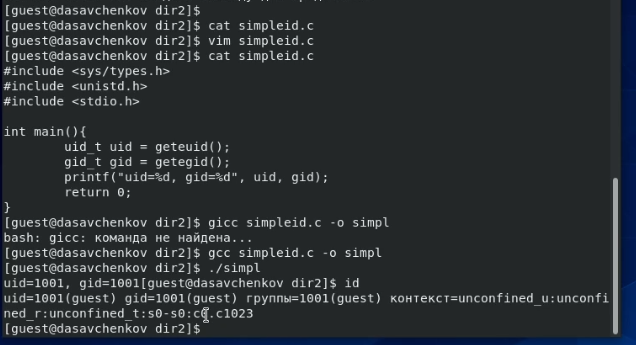


Figure 2: Компиляция и выполнение программы simpleid

1.6. Усложнил программу, добавив вывод действительных идентификаторов согласно шаблону из методички. Для получившейся программы оставил название simpleid.c. (рис. 3)

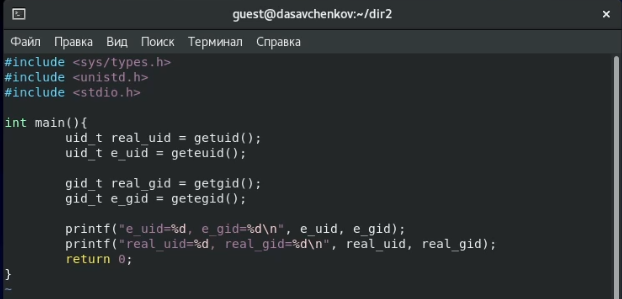


Figure 3: Программа simpleid2.c

1.7. Скомпилировал и запустил simpleid.c: gcc simpleid.c -o simpl и ./simpl. (рис. 4)

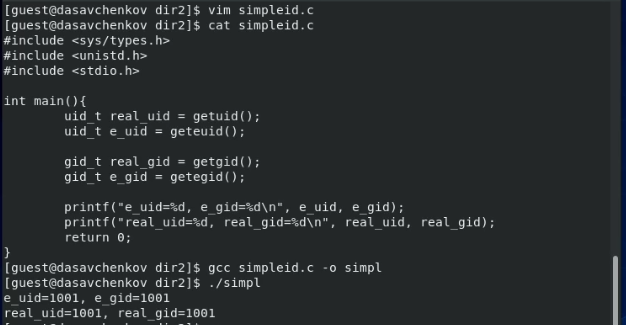


Figure 4: Компиляция и выполнение программы simpleid2

1.8. От имени суперпользователя выполнил команды: chown root:guest /home/guest/simpl и chmod u+s /home/guest/simpl. (рис. 5)

1.9. Повысил временно свои права с помощью su. (рис. 5) Первая команда меняет владельца файла, а вторая добавляет SetUID-бит.

1.10. Выполнил проверку правильности установки новых атрибутов и смены владельца файла simpl: ls -l simpl. (рис. 5)

1.11. Запустил simpl и id: ./simpl и id. (рис. 5)

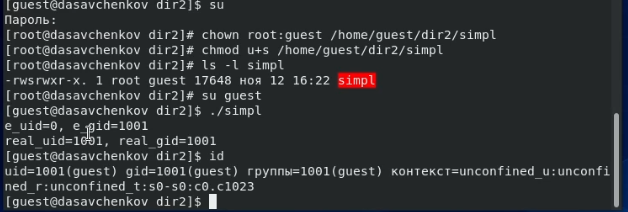


Figure 5: Смена пользователя. Установка SetUID-бита. Выполнение программы simpl

1.12. Проделал то же самое относительно SetGID-бита. (рис. 6)

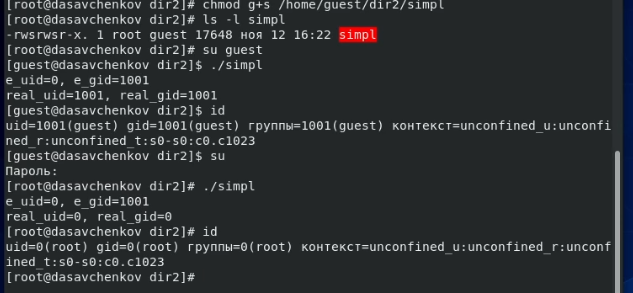


Figure 6: Установка SetGID-бита. Выполнение программы simpl

1.13. Создал программу readfile.c по шаблону из методички. (рис. 7)

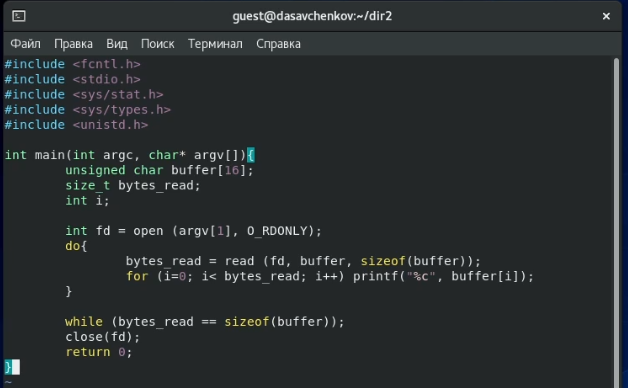


Figure 7: Программа readfile.c

1.14. Откомпилировал её: gcc readfile.c -o readfile. (рис. 8)

1.15. Сменил владельца у файла readfile.c и изменил права так, чтобы только суперпользователь (root) мог прочитать его, a guest не мог. (рис. 8)

Figure 8: Работа с программой readfile.c

Figure 8: Работа с программой readfile.c

1.16. Проверил, что пользователь guest не может прочитать файл readfile.c. (рис. 9)

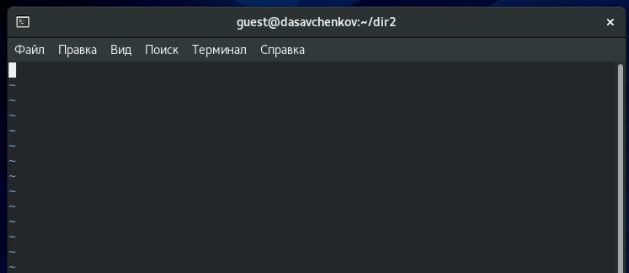


Figure 9: Запрет на чтение программы readfile.c для guest

1.17. Сменил у программы readfile владельца (рис. 8) и установил SetUID-бит (рис. 10).

Figure 10: Установка SetUID-бита на программу readfile

Figure 10: Установка SetUID-бита на программу readfile

1.18. Проверил, может ли программа readfile прочитать файл *readfile.c*. (рис. 11)

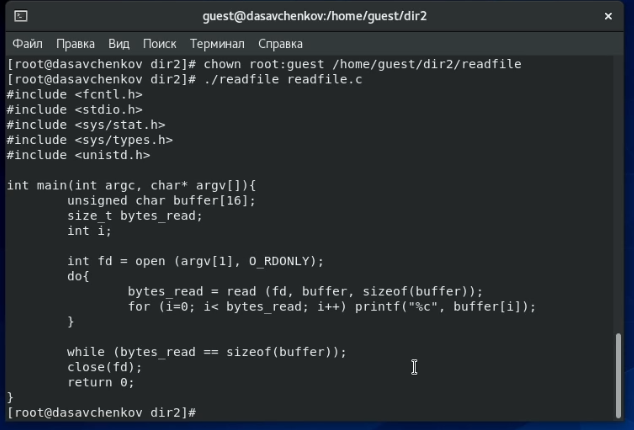


Figure 11: Программа readfile читает readfile.c

1.19. Проверил, может ли программа readfile прочитать файл */etc/shadow*. (рис. 12)

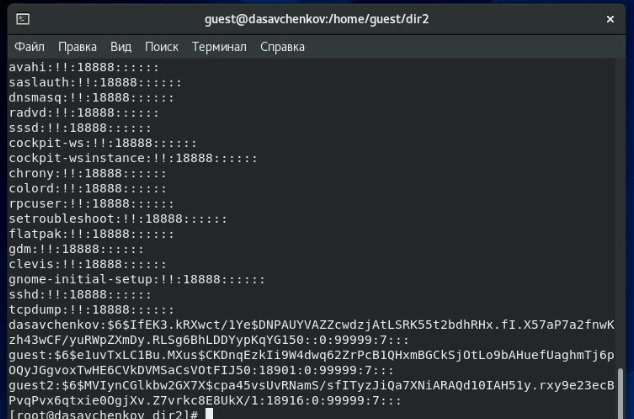


Figure 12: Программа readfile читает /etc/shadow

1. Исследование Sticky-бита

2.1. Выяснил, установлен ли атрибут Sticky на директории /tmp, для чего выполнил команду: ls -l / | grep tmp. (рис. 13)

2.2. От имени пользователя guest создал файл file01.txt в директории /tmp со словом test: echo "test" > /tmp/file01.txt. (рис. 13)

2.3. Просмотрел атрибуты у только что созданного файла и разрешил чтение и запись для категории пользователей «все остальные»: ls -l /tmp/file01.txt, chmod o+rw /tmp/file01.txt и ls -l /tmp/file01.txt. (рис. 13)

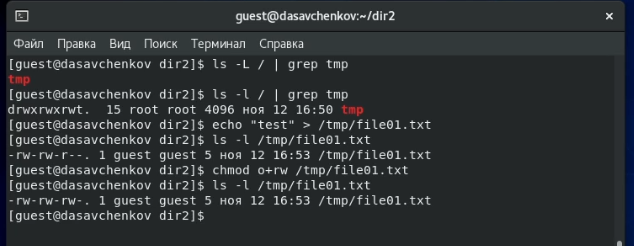


Figure 13: Исследование Sticky-бита от имени guest

2.4. От пользователя guest2 (не являющегося владельцем) попробовал прочитать файл /tmp/file01.txt: cat /tmp/file01.txt. (рис. 14)

2.5. От пользователя guest2 попробовал дозаписать в файл */tmp/file01.txt* слово test2 командой: echo "test2" >> /tmp/file01.txt. (рис. 14) Операция прошла успешно.

2.6. Проверил содержимое файла командой: cat /tmp/file01.txt. (рис. 14)

2.7. От пользователя guest2 попробовал записать в файл */tmp/file01.txt* слово test3, стерев при этом всю имеющуюся в файле информацию командой: echo "test3" > /tmp/file01.txt. (рис. 14) Операция прошла успешно.

2.8. Проверил содержимое файла командой: cat /tmp/file01.txt. (рис. 14)

2.9. От пользователя guest2 попробовал удалить файл */tmp/file01.txt* командой: rm /tmp/fileOl.txt. (рис. 14) Операция была не позволена.

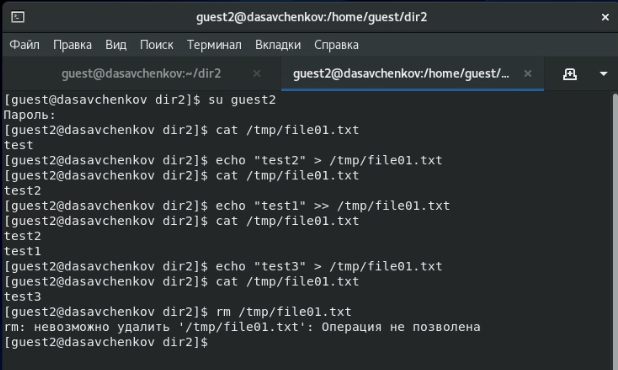


Figure 14: Работа с file01.txt от имени guest2 при наличии Sticky-бита

2.10. Повысил свои права до суперпользователя следующей командой: su -, и выполнил после этого команду, снимающую атрибут t (Sticky-бит) с директории */tmp*: chmod -t /tmp. (рис. 15)

2.11. Покинул режим суперпользователя командой: exit. (рис. 15)

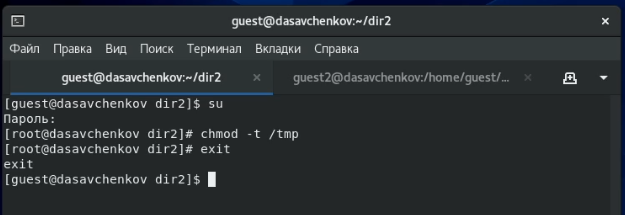


Figure 15: Снятие Sticky-бита с /tmp

2.12. От пользователя guest2 проверил, что атрибута t у директории */tmp* нет: ls -l / | grep tmp. (рис. 16)

2.13. Повторил предыдущие шаги. (рис. 16) Теперь удалось удалить файл.

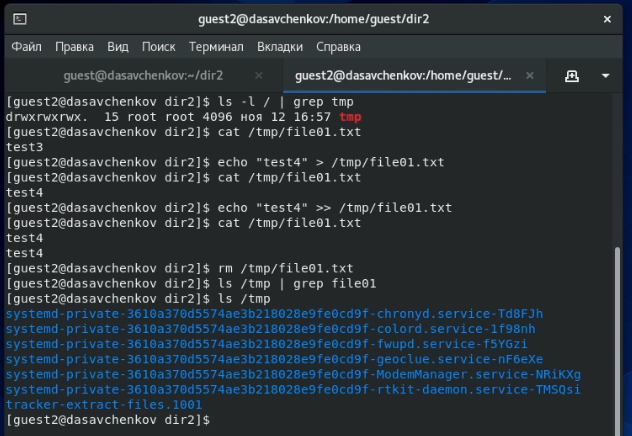


Figure 16: Работа с file01.txt от имени guest2 без Sticky-бита

2.14. Да, мне удалось удалить файл от имени пользователя, не являющегося его владельцем.

2.15. Повысил свои права до суперпользователя и вернул атрибут t на директорию /tmp: su -, chmod +t /tmp и exit. (рис. 17)

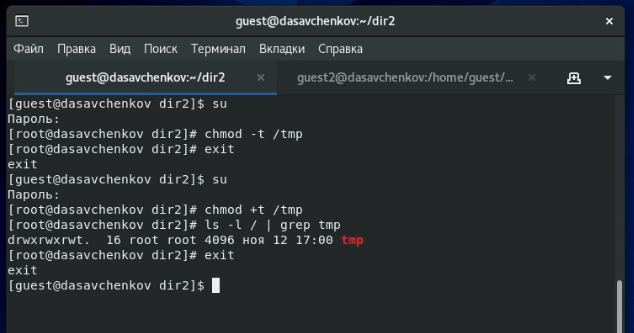


Figure 17: Возвращение Sticky-бита на /tmp

# Выводы

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

# Список литературы

1. Кулябов Д. С., Королькова А. В., Геворкян М. Н. Информационная безопасность компьютерных сетей. Лабораторная работа № 5. Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов