Лабораторная работа №5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Тарутина Кристина Олеговна

Содержание

[1 Цель работы 1](#_Toc163944115)

[2 Выполнение лабораторной работы 1](#_Toc163944116)

[3 Выводы 1](#_Toc163944117)

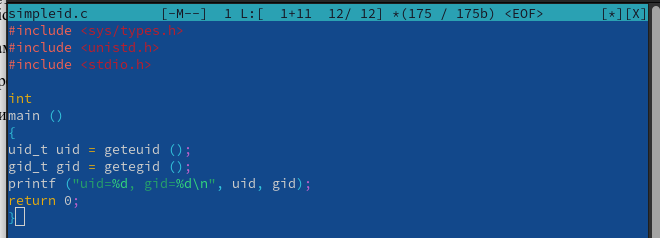
[Список литературы 6](#_Toc163944118)

# 1 Цель работы

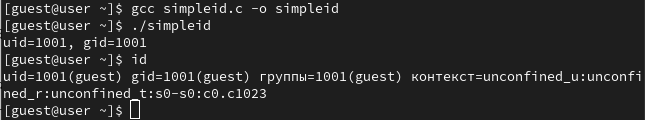
Изучение механизмов изменения идентификаторов, применения SetUID- и Sticky-битов. Получение практических навыков работы в кон- соли с дополнительными атрибутами. Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов

# 2 Выполнение лабораторной работы

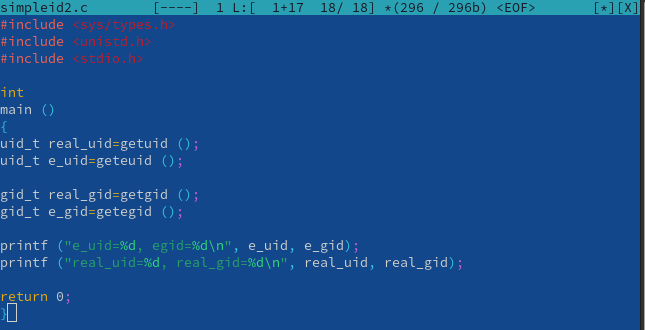
1.. Войдите в систему от имени пользователя guest. 2. Создайте программу simpleid.c(рис. [-@fig:001]).

[](https://github.com/TiRisik/study_2023-2024_infosec/blob/master/labs/lab5/report/image/image1.png){#fig:001 width=70%}

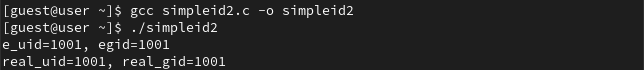
1. Скомплилируйте программу и убедитесь, что файл программы создан: gcc simpleid.c -o simpleid
2. Выполните программу simpleid: ./simpleid
3. Выполните системную программу id: id и сравните полученный вами результат с данными предыдущего пункта задания (рис. [-@fig:002]).

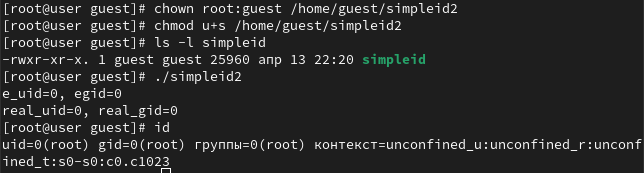
[](https://github.com/TiRisik/study_2023-2024_infosec/blob/master/labs/lab5/report/image/image2.png){#fig:002 width=70%}

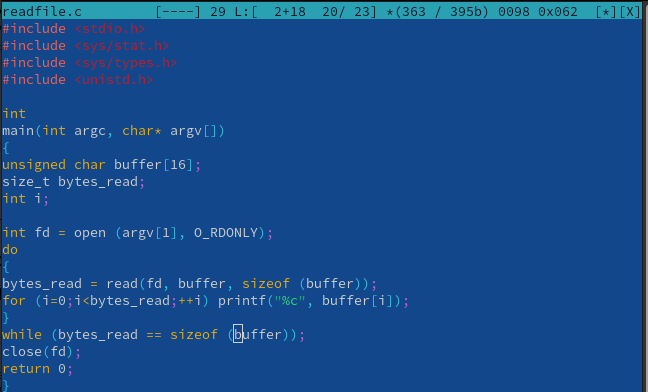
1. Усложните программу, добавив вывод действительных идентификато- ров. Получившуюся программу назовите simpleid2.c(рис. [-@fig:003]).

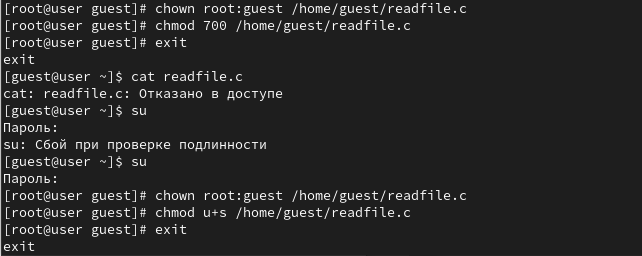
[](https://github.com/TiRisik/study_2023-2024_infosec/blob/master/labs/lab5/report/image/image3.png){#fig:003 width=70%}

1. Скомпилируйте и запустите simpleid2.c: gcc simpleid2.c -o simpleid2 ./simpleid2 (рис. [-@fig:004]).

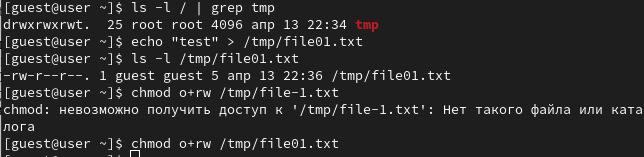
[](https://github.com/TiRisik/study_2023-2024_infosec/blob/master/labs/lab5/report/image/image4.png){#fig:004 width=70%} 8. От имени суперпользователя выполните команды chown root:guest /home/guest/simpleid2 chmod u+s /home/guest/simpleid2 9. Используйте sudo или повысьте временно свои права с помощью su. Поясните, что делают эти команды. 10. Выполните проверку правильности установки новых атрибутов и смены владельца файла simpleid2: ls -l simpleid2 11. Запустите simpleid2 и id: ./simpleid2 id Сравните результаты. 12. Проделайте тоже самое относительно SetGID-бита (рис. [-@fig:005]).

[](https://github.com/TiRisik/study_2023-2024_infosec/blob/master/labs/lab5/report/image/image5.png){#fig:005 width=70%} 13. Создайте программу readfile.c (рис. [-@fig:006]).

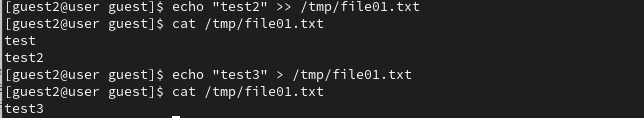
[](https://github.com/TiRisik/study_2023-2024_infosec/blob/master/labs/lab5/report/image/image6.png){#fig:006 width=70%} 15. Смените владельца у файла readfile.c (или любого другого текстового файла в системе) и измените права так, чтобы только суперпользователь (root) мог прочитать его, a guest не мог. 16. Проверьте, что пользователь guest не может прочитать файл readfile.c. 17. Смените у программы readfile владельца и установите SetU’D-бит. 18. Проверьте, может ли программа readfile прочитать файл readfile.c? (рис. [-@fig:007]).

[](https://github.com/TiRisik/study_2023-2024_infosec/blob/master/labs/lab5/report/image/image7.png){#fig:007 width=70%})

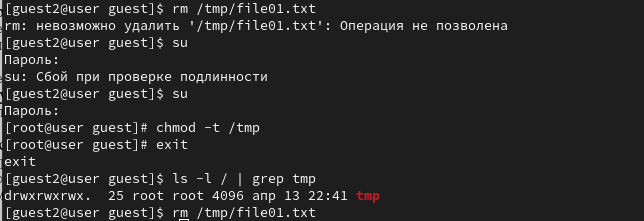
1. Выясните, установлен ли атрибут Sticky на директории /tmp, для чего выполните команду ls -l / | grep tmp
2. От имени пользователя guest создайте файл file01.txt в директории /tmp со словом test: echo "test" > /tmp/file01.txt
3. Просмотрите атрибуты у только что созданного файла и разрешите чте- ние и запись для категории пользователей «все остальные»: ls -l /tmp/file01.txt chmod o+rw /tmp/file01.txt ls -l /tmp/file01.txt (рис. [-@fig:008]).

[](https://github.com/TiRisik/study_2023-2024_infosec/blob/master/labs/lab5/report/image/image8.png){#fig:008 width=70%})

1. От пользователя guest2 (не являющегося владельцем) попробуйте про- читать файл /tmp/file01.txt: cat /tmp/file01.txt
2. От пользователя guest2 попробуйте дозаписать в файл /tmp/file01.txt слово test2 командой echo "test2" > /tmp/file01.txt Удалось ли вам выполнить операцию?
3. Проверьте содержимое файла командой cat /tmp/file01.txt
4. От пользователя guest2 попробуйте записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию ко- мандой echo "test3" > /tmp/file01.txt Удалось ли вам выполнить операцию?
5. Проверьте содержимое файла командой cat /tmp/file01.txt (рис. [-@fig:009]).

[](https://github.com/TiRisik/study_2023-2024_infosec/blob/master/labs/lab5/report/image/image9.png){#fig:009 width=70%})

1. От пользователя guest2 попробуйте удалить файл /tmp/file01.txt ко- мандой rm /tmp/fileOl.txt Удалось ли вам удалить файл?
2. Повысьте свои права до суперпользователя следующей командой su - и выполните после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp: chmod -t /tmp
3. Покиньте режим суперпользователя командой exit
4. От пользователя guest2 проверьте, что атрибута t у директории /tmp нет: ls -l / | grep tmp
5. Повторите предыдущие шаги. Какие наблюдаются изменения? (рис. [-@fig:010]).

[](https://github.com/TiRisik/study_2023-2024_infosec/blob/master/labs/lab5/report/image/image10.png){#fig:010 width=70%})

1. Повысьте свои права до суперпользователя и верните атрибут t на ди- ректорию /tmp: su - chmod +t /tmp exi (рис. [-@fig:011]).

[](https://github.com/TiRisik/study_2023-2024_infosec/blob/master/labs/lab5/report/image/image11.png){#fig:011 width=70%})

# 3 Выводы

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

# Список литературы