Отчёт по лабораторной работе №5

Дисциплина: Информационная безопасность

Евдокимов Иван Андреевич

Содержание

# Техническое оснащение:

* Персональный компьютер с операционной системой Windows 10;
* OBS Studio, использующийся для записи скринкаста лабораторной работы;
* Приложение Visual Studio Code для редактирования файлов формата *md*, а также для конвертации файлов отчётов и презентаций;

# Цель работы:

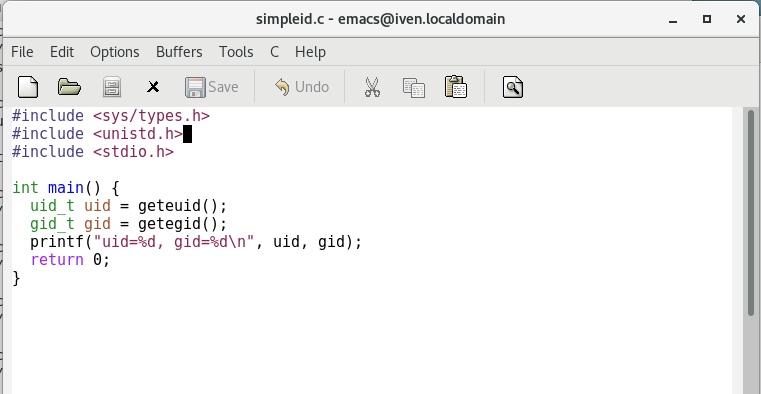
Целью данной лабораторной работы является изучить механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получение практических навыков работы в консоли с дополнительными атрибутами. Рассмотрение работы механизма смены идентификатора процессов пользователей, а также влияние бита Sticky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

# Выполнение лабораторной работы

1. Вошёл в систему от имени пользователя guest и создал программу simpleid.c.

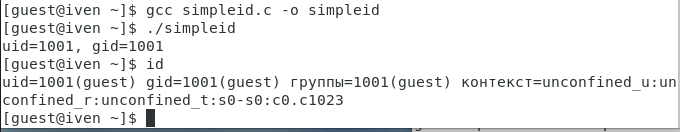
Вошёл в систему

Вошёл в систему



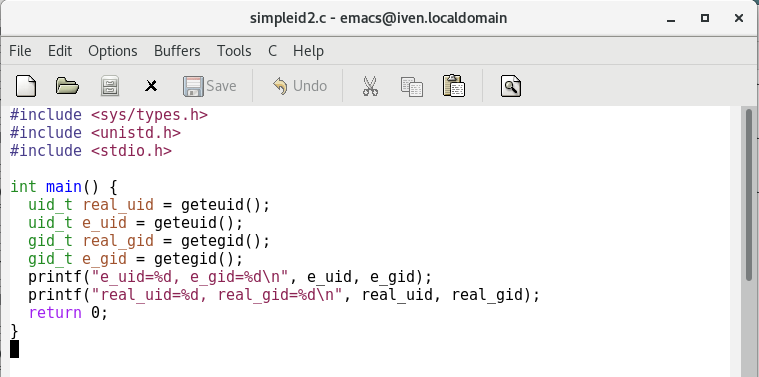
Программа simpleid.c

1. Скомпилировал программу и убедился, что файл программы создан. Выполнил программу simpleid. Выполнил системную программу id. В отличие от команды id, моя программа не выводит контекст и все группы, в которые пользователь.



Выполнил системную программу id

1. Усложнил программу, добавив вывод действительных идентификаторов.



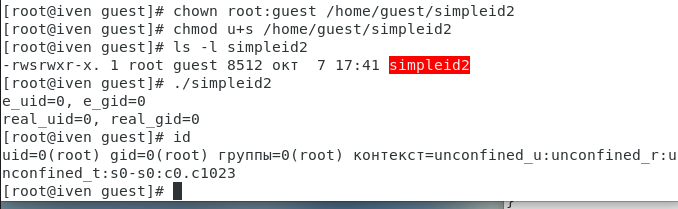
Усложнил программу

1. Получившуюся программу назвал simpleid2.c. Скомпилировал и запустил simpleid2.c.

|  |
| --- |
| Скомпилировал и запустил simpleid2.c |

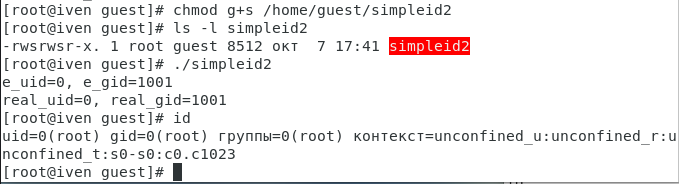
Скомпилировал и запустил simpleid2.c

1. От имени суперпользователя выполнил команды chown root:guest /home/guest/simpleid2 и chmod u+s /home/guest/simpleid2. Первая команда меняет владельца файла simpleid2 на группу guest. Вторая команда меняет права доступа к файлу simpleid2 для пользователя и установленные атрибуты SUID или SGID позволяют запускать файл на выполнение с правами владельца файла или группы соответственно. Выполнила проверку правильности установки новых атрибутов и смены владельца файла simpleid2. Запустил simpleid2 и id. Сравнил результаты.

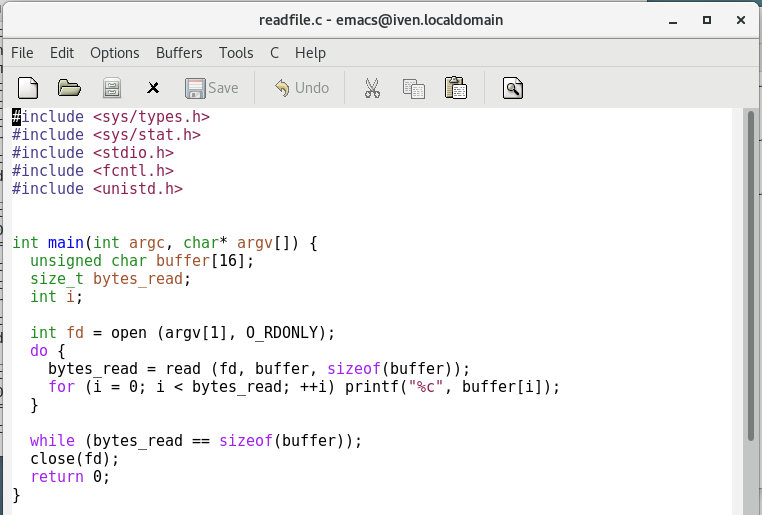


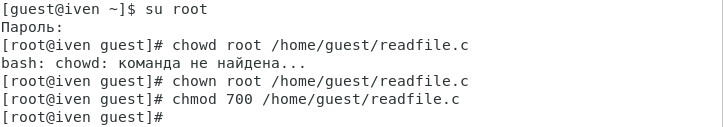
Выполнил команды chown root:guest

1. Проделал тоже самое относительно SetGID-бита. Значения вывода обеих программ совпадают, только в отличие от предыдущего пункта значение e\_gid = 1002. (@fig:007)



SetGID-бит

1. Создал программу readfile.c 
2. Откомпилировали программу readfile.c: **gcc readfile.c -o readfile**. Сменили владельца у файла readfile.c и изменили права так, чтобы только суперпользователь (root) мог прочитать его, a guest не мог. (@fig:009)



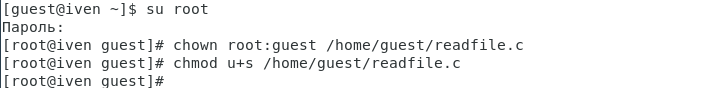
Смена владельца и изменение прав программы readfile.c

1. Проверили, что пользователь guest не может прочитать файл readfile.c.(@fig:010)



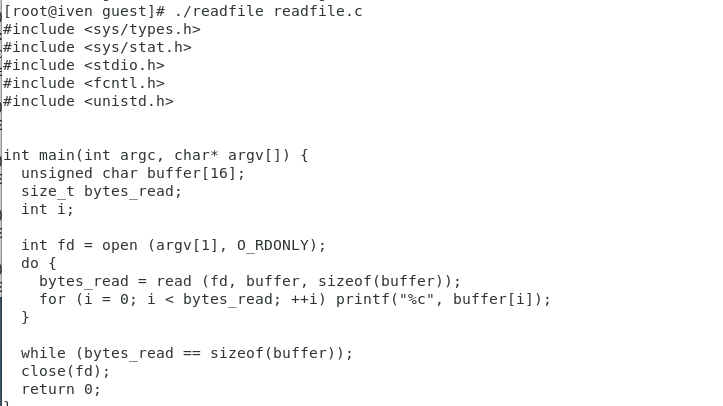
Проверка возможности чтения файла readfile.c пользователем guest

1. Сменили у программы readfile владельца и установили SetU’D-бит. (@fig:011)



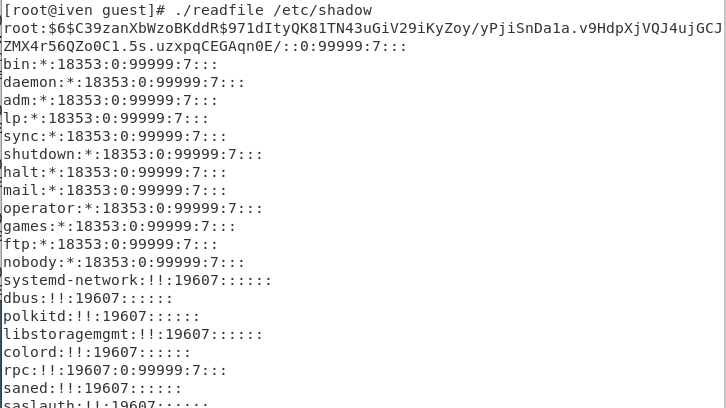
Смена у программы readfile владельца и установка SetU’D-бита

1. Проверяю, может ли программа readfile прочитать файл readfile.c. Да, может. (@fig:012)



Чтение программой readfile файла readfile.с

1. Проверяю, может ли программа readfile прочитать файл /etc/shadow. Да, может. (@fig:013)



Чтение программой readfile файла readfile.с

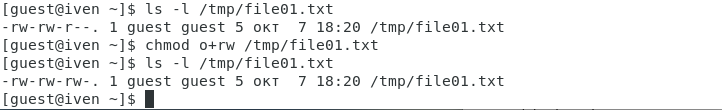
## Исследование Sticky-бита

1. Выяснили, что установлен атрибут Sticky на директории /tmp с помощью команды **ls -l / | grep tmp**. От имени пользователя guest создали файл file01.txt в директории /tmp со словом test. (@fig:014)



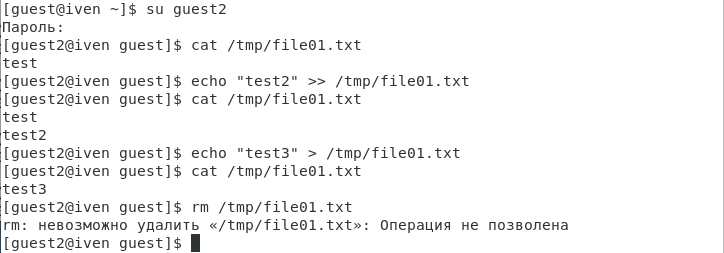
Выполнение команды ls -l / | grep tmp и создание файла file01.txt

1. Просмотрели атрибуты у только что созданного файла и разрешили чтение и запись для категории пользователей «все остальные». (@fig:015)



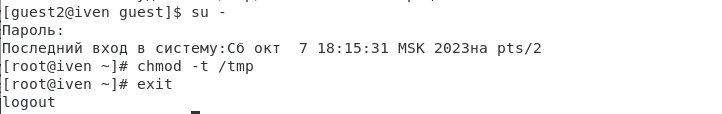
Арибуты файла file01.txt

1. От пользователя guest2 (не являющегося владельцем) попробовали прочитать, дозаписать, записать, удалить файл /tmp/file01.txt. Выполнено все, кроме удаления файла. (@fig:016)



Чтение, дозапись, запись, удаление файл /tmp/file01.txt

1. Повысили свои права до суперпользователя командой **su -** и выполнили после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp: **chmod -t /tmp**. Покинули режим суперпользователя командой **exit**.(@fig:017)



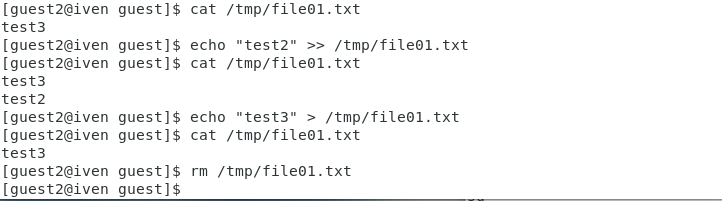
Сняли Sticky-бит с директории /tmp

1. От пользователя guest2 проверили, что атрибута t у директории /tmp нет. (@fig:018)

Отсутсвие атрибута t у директории /tmp

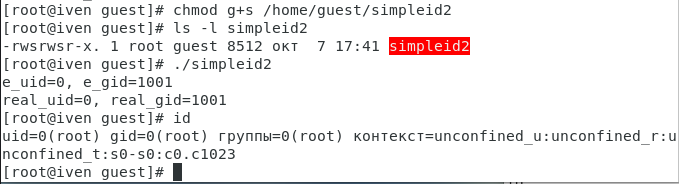
Отсутсвие атрибута t у директории /tmp

1. Повторили предыдущие шаги. В данном случае получилось выполнить удаление файла. (@fig:019)



Чтение, дозапись, запись, удаление файл /tmp/file01.txt без атрибута t

1. Повысили свои права до суперпользователя и вернули атрибут t на директорию /tmp. (@fig:020)



Чтение, дозапись, запись, удаление файл /tmp/file01.txt без атрибута t

**Выводы:**

В ходе выполнения лабораторной работы мы приобрели изучили механизмы изменения идентификаторов, применения SetUID- и Sticky-битов. Получили практические навыки работы в консоли с дополнительными атрибутами. Рассмотрели работу механизма смены идентификатора процессов пользователей, а также влияние бита Sticky на запись и удаление файлов.

# Список литературы

1. [Официальный сайт VirtualBox](https://www.virtualbox.org/)
2. [Материал для выполнения лабораторной](https://esystem.rudn.ru/pluginfile.php/2090279/mod_resource/content/2/005-lab_discret_sticky.pdf)
3. [Официальный сайт CentOS](https://www.centos.org/)