Отчёт по лабораторной работе № 5

Дисциплина: Основы информационной безопасности

Перелыгин Сергей Викторович

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

* Сделать отчёт по лабораторной работе в формате Markdown.
* В качестве отчёта предоставить отчёты в 3 форматах: pdf, docx и md.

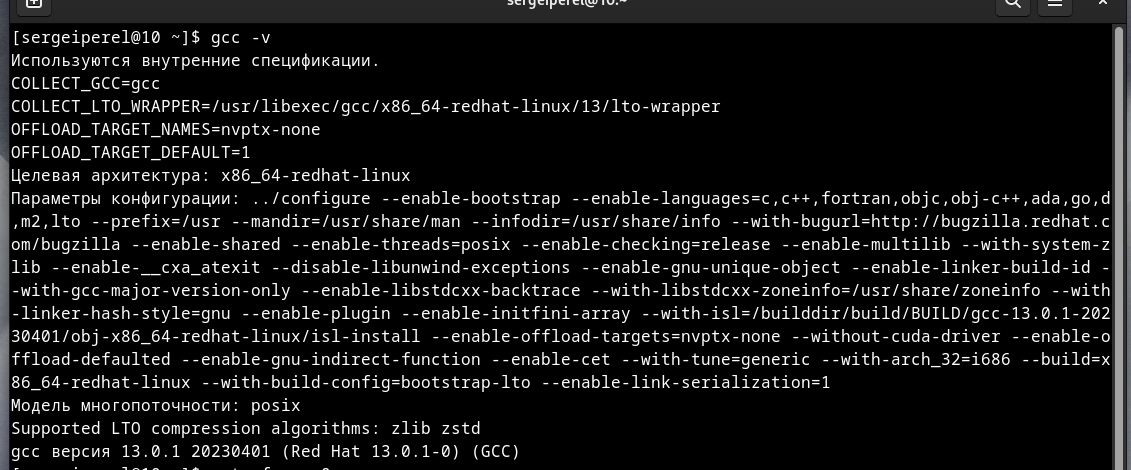
# Теоретическое введение

SetUID, SetGID и Sticky - это специальные типы разрешений позволяют задавать расширенные права доступа на файлы или каталоги.

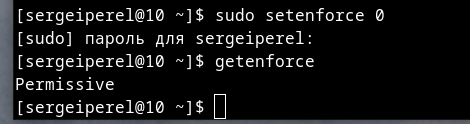
• SetUID (set user ID upon execution — «установка ID пользователя во время выполнения) являются флагами прав доступа в Unix, которые разрешают пользователям запускать исполняемые файлы с правами владельца исполняемого файла. • SetGID (set group ID upon execution — «установка ID группы во время выполнения») являются флагами прав доступа в Unix, которые разрешают пользователям запускать исполняемые файлы с правами группы исполняемого файла. • Sticky bit в основном используется в общих каталогах, таких как /var или /tmp, поскольку пользователи могут создавать файлы, читать и выполнять их, принадлежащие другим пользователям, но не могут удалять файлы, принадлежащие другим пользователям.

# Выполнение лабораторной работы

1. Для начала я убедился, что компилятор gcc установлен, используя команду “gcc -v”. Затем отключил систему запретов до очередной перезагрузки системы командой “sudo setenforce 0”, после чего команда “getenforce” вывела “Permissive” (рис. 4.1 и 4.2).

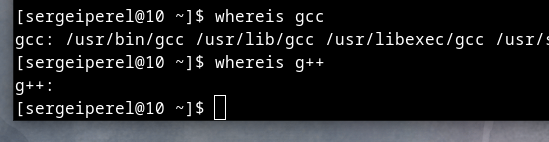


Предварительная подготовка 1



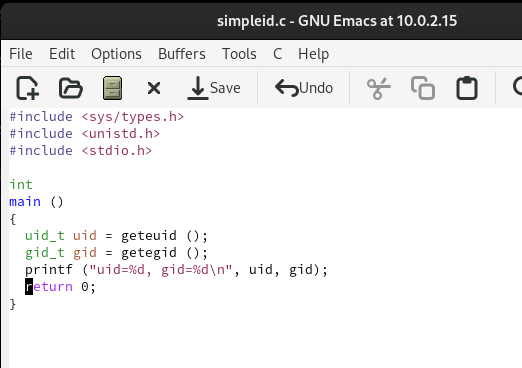
Предварительная подготовка 2

1. Проверил успешное выполнение команд “whereis gcc” и “whereis g++” (их расположение) (рис. 4.3).



Команда “whereis”

1. Вошел в систему от имени пользователя guest командой “su - guest”. Создал программу simpleid.c командой “touch simpleid.c” и открыл её в редакторе emacs.
2. Код программы выглядит следующим образом (рис. 4.4).



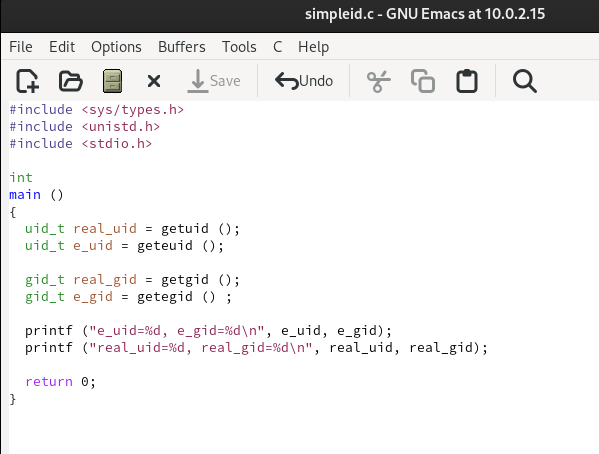
Предварительная подготовка 3

1. Скомпилировал программу и убедился, что файл программы был создан командой “gcc simpleid.c -o simpleid”. Выполнил программу simpleid командой “./simpleid”, а затем выполнил системную программу id командой “id”. Результаты, полученные в результате выполнения обеих команд, совпадают (uid=1001 и gid=1001) (рис. 4.5).



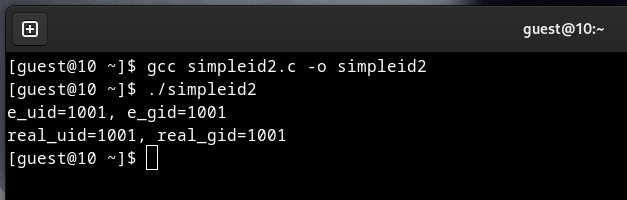
Предварительная подготовка 4

1. Усложнил программу, добавив вывод действительных идентификаторов (рис. 4.6).



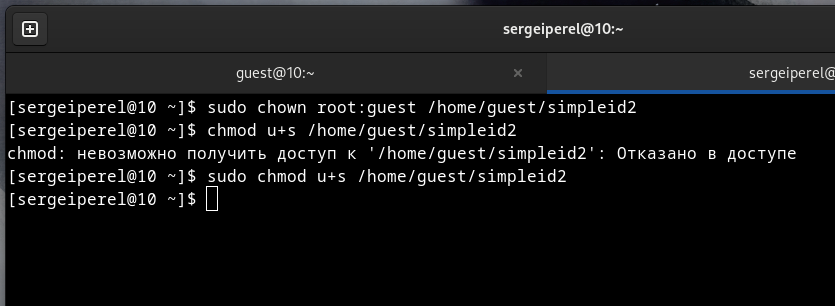
Предварительная подготовка 5

1. Получившуюся программу назвал simpleid2.c.
2. Скомпилировал и запустил simpleid2.c командами “gcc simpleid2.c -o sipleid2” и “./simpleid2” (рис. 4.7).

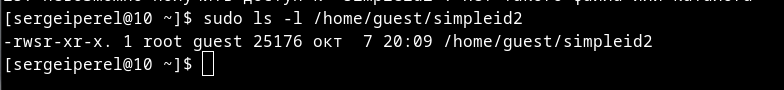


Компиляция и выполнение программы simpleid2

1. От имени суперпользователя выполнил команды “sudo chown root:guest /home/guest/simpleid2” и “sudo chmod u+s /home/guest/simpleid2” (рис. 4.8), затем выполнил проверку правильности установки новых атрибутов и смены владельца файла simpleid2 командой “sudo ls -l /home/guest/simpleid2” (рис. 4.9). Этими командами была произведена смена пользователя файла на root и установлен SetUID-бит.

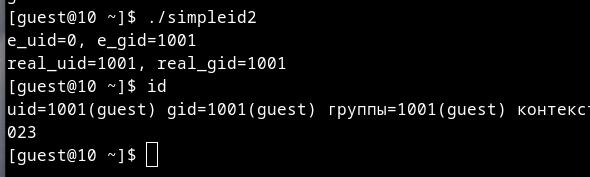


Установка новых атрибутов (SetUID)



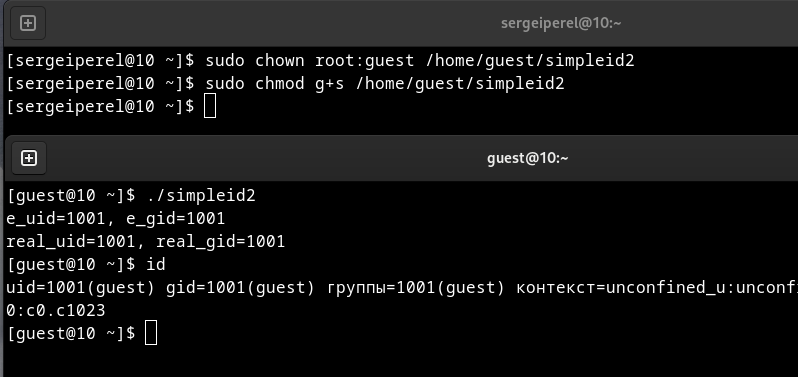
Проверка правильности установки новых атрибутов

1. Запустил программы simpleid2 и id. Теперь появились различия в uid (рис. 4.10).



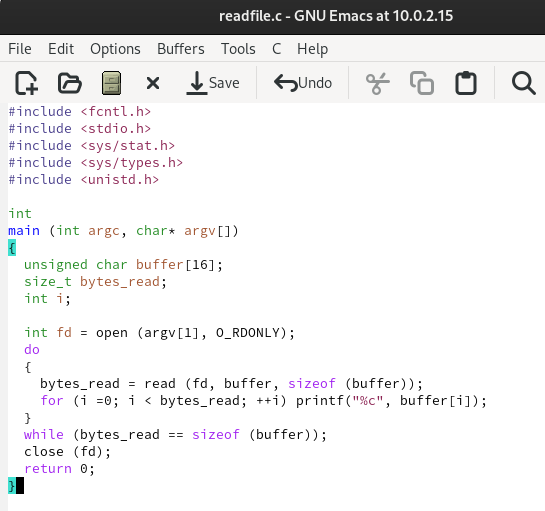
Запуск simpleid2 после установки SetUID

1. Проделал тоже самое относительно SetGID-бита. Также можем заметить различия с предыдущим пунктом (рис. 4.11).



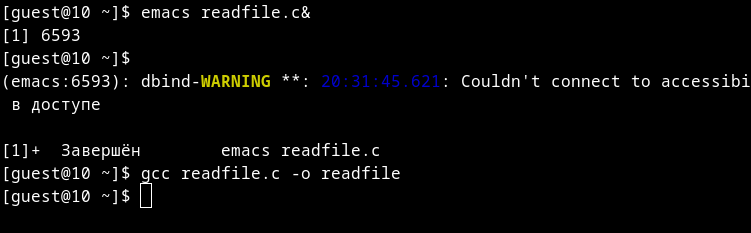
Запуск simpleid2 после установки SetGID

1. Создаем программу readfile.c (рис. 4.12).

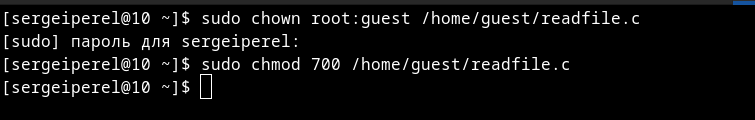


Программа readfile.c

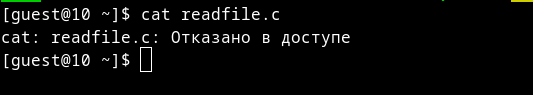
1. Скомпилировал созданную программу командой “gcc readfile.c -o readfile” (рис. 4.13). Сменил владельца у файла readfile.c командой “sudo chown root:guest /home/guest/readfile.c” и поменял права так, чтобы только суперпользователь мог прочитать его, а guest не мог, с помощью команды “sudo chmod 700 /home/guest/readfile.c” (рис. 4.14). Теперь убедился, что пользователь guest не может прочитать файл readfile.c командой “cat readfile.c”, получив отказ в доступе (рис. 4.15).



Компиляция readfile.c

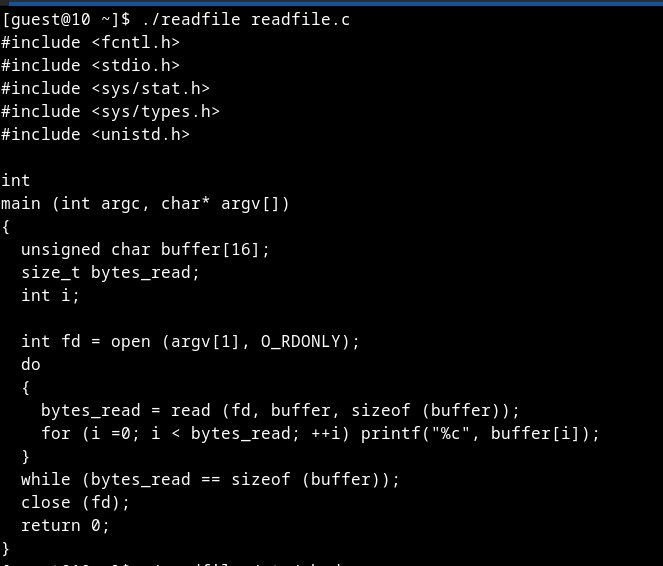


Смена владельца и прав доступа у файла readfile.c

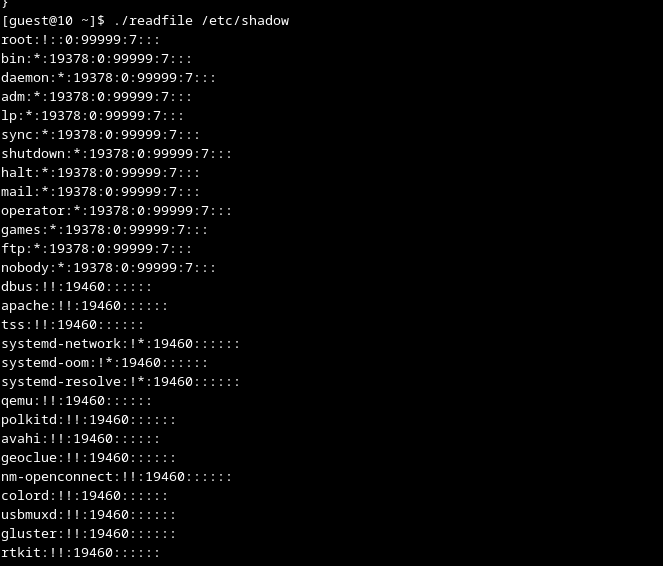


Попытка прочитать файл

1. Поменял владельца у программы readfile и устанавила SetUID. Проверил, может ли программа readfile прочитать файл readfile.c командой “./readfile readfile.c”. Прочитать удалось (рис. 4.16). Аналогично проверил, можно ли прочитать файл /etc/shadow. Прочитать не удалось (рис. 4.17).

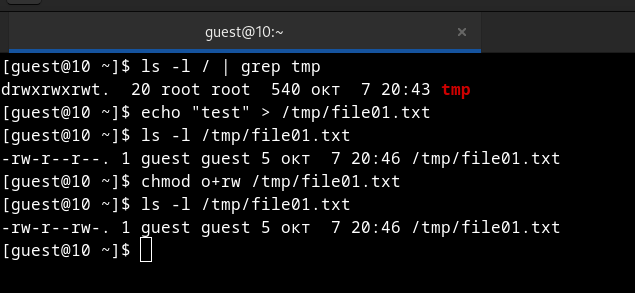


Запуск программы readfile



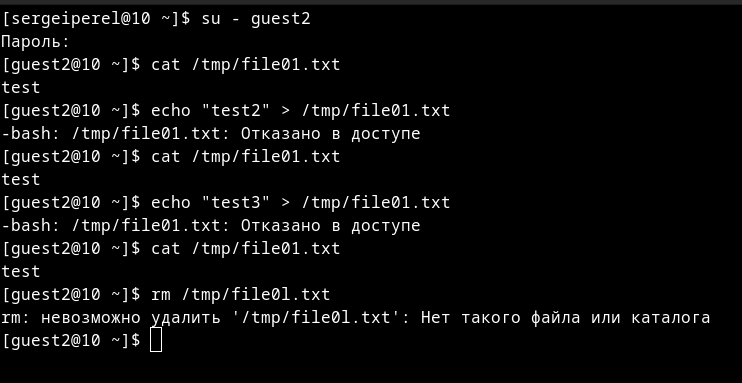
Файл /etc/shadow

1. Командой “ls -l / | grep tmp” убедился, что атрибут Sticky на директории /tmp установлен. От имени пользователя guest создал файл file01.txt в директории /tmp со словом test командой “echo”test” > /tmp/file01.txt”. Просмотрел атрибуты у только что созданного файла и разрешаем чтение и запись для категории пользователей “все остальные” командами “ls -l /tmp/file01.txt” и “chmod o+rw /tmp/file01.txt” (рис. 4.18).



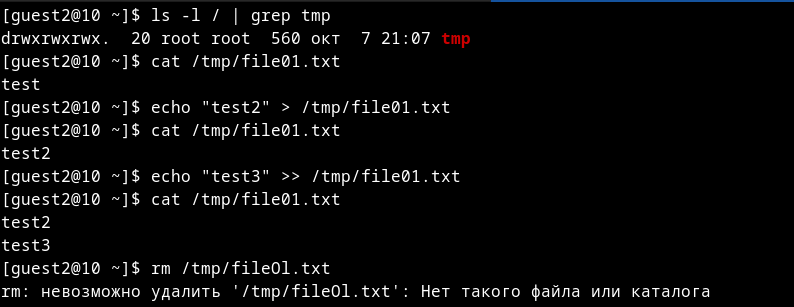
Создание файла file01.txt

1. От имени пользователя guest2 попробовал прочитать файл командой “cat /tmp/file01.txt” - это удалось. Далее попытался дозаписать в файл слово test2, проверить содержимое файла и записать в файл слово test3, стерев при этом всю имеющуюся в файле информацию - эти операции выполнить не удалось. От имени пользователя guest2 попробовал удалить файл - это не удается, возникает ошибка (рис. 4.19).



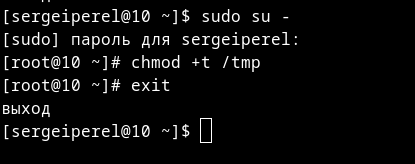
Попытка выполнить действия над файлом file01.txt от имени пользователя guest2

1. Повысил права до суперпользователя командой “sudo su -” и выполнил команду, снимающую атрибут t с директории /tmp “chmod -t /tmp”. После чего покинул режим суперпользователя командой “exit”. Повторил предыдущие шаги. Теперь мне удалось выполнить все, кроме команды удалить файл file01.txt от имени пользователя, не являющегося его владельцем (рис. 4.20).



Удаление атрибута t (Sticky-бита) и повторение действий

1. Повысил свои права до суперпользователя и вернул атрибут t на директорию /tmp (рис. 4.21).



Возвращение атрибута t (Sticky-бита)

# Выводы

**Вывод:** В ходе выполнения данной лабораторной работы я изучил механизмы изменения идентификаторов, применение 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.

# Библиография

* Медведовский И.Д., Семьянов П.В., Платонов В.В. Атака через Internet. — НПО “Мир и семья-95”, 1997. — URL: http://bugtraq.ru/library/books/attack1/index.html
* Медведовский И.Д., Семьянов П.В., Леонов Д.Г. Атака на Internet. — Издательство ДМК, 1999. — URL: http://bugtraq.ru/library/books/attack/index.html
* Запечников С. В. и др. Информационн~пасность открытых систем. Том 1. — М.: Горячаая линия -Телеком, 2006.
* Введение в информационную безопасность. Типы уязвимостей. (Д.Гамаюнов, МГУ)
* Практические аспекты сетевой безопасности. Вводная лекция. Сетевая безопасность. Стек протоколов TCP/IP. (Д. Гамаюнов, МГУ)
* Практические аспекты сетевой безопасности. Сетевая безопасность. Межсетевые экраны. (В. Иванов, МГУ)
* Практические аспекты сетевой безопасности. Сетевая безопасность. Системы обнаружения и фильтрации компьютерных атак (IDS/IPS). (Д. Гамаюнов, МГУ)
* Практические аспекты сетевой безопасности. Контроль нормального поведения приложений. Security Enhanced Linux (SELinux) (В. Сахаров, МГУ)