Отчёт по лабораторной работе №5

Дисциплина: Информационная безопасность

Андрианова Марина Георгиевна

Содержание

[Цель работы 1](#_Toc178975842)

[Выполнение лабораторной работы 1](#_Toc178975843)

[Выводы 7](#_Toc178975844)

# 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

Для лабораторной работы необходимо проверить, установлен ли компилятор gcc, делаем это с помощью команды gcc -v. Также осуществляется отключение системы запретом с помощью setenforce 0 (рис. 1).

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

*Подготовка к лабораторной работе*

Проверяем местоположение gcc и g++ (рис. 2).



*Местоположение gcc и g++*

Осуществляется вход от имени пользователя guest (рис. 3).

Изображение выглядит как текст, Шрифт, снимок экрана, черный

Автоматически созданное описание

*Вход от имени пользователя guest*

Создание файла simpled.c и запись в файл кода (рис. 4)



*Создание файла simpled.c*

C++ Листинг 1 #include <sys/types.h> #include <unistd.h> #include <stdio.h> int main () { uid\_t uid = geteuid (); gid\_t gid = getegid (); printf ("uid=%d, gid=%d\n", uid, gid); return 0; }

Cодержимое файла выглядит следующим образом (рис. 5)

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

*Содержимое файла*

Компилирую файл, проверяю, что он скомпилировался (рис. 6)

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

*Компиляция файла*

Запускаю исполняемый файл. В выводе файла выписаны номера пользователя и групп; от вывода при вводе if они отличаются только тем, что информации меньше (рис. 7)

Изображение выглядит как текст, снимок экрана, Шрифт, черный

Автоматически созданное описание

*Сравнение команд*

Создание файла simpled2.c и запись в файл кода (рис. 8-9)



*Создание файла simpled2.c*

*Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание*

*Содержимое файла*

C++ Листинг 2 #include <sys/types.h> #include <unistd.h> #include <stdio.h> int main () { uid\_t real\_uid = getuid (); uid\_t e\_uid = geteuid (); gid\_t real\_gid = getgid (); gid\_t e\_gid = getegid () ; printf ("e\_uid=%d, e\_gid=%d\n", e\_uid, e\_gid); printf ("real\_uid=%d, real\_gid=%d\n", real\_uid, real\_gid); return 0; }

Компиляция файла simpled2.c. Запуск программы (рис. 10)

Изображение выглядит как текст, Шрифт, снимок экрана, типография

Автоматически созданное описание

*Компиляция файла simpled2.c.*

С помощью chown изменяю владельца файла на суперпользователя, с помощью chmod изменяю права доступа (рис. 11)

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

*Смена владельца файла и прав доступа к файлу*

Создание файла readfile.c и запись кода в файл (рис. 12-13)



*Создание и компиляция файла*

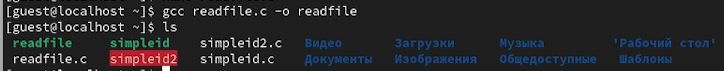
*Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание*

*Содержимое файла readfile.c*

C++ Листинг 3 #include <fcntl.h> #include <stdio.h> #include <sys/stat.h> #include <sys/types.h> #include <unistd.h> int main (int argc, char\* argv[]) { unsigned char buffer[16]; size\_t bytes\_read; int i; int fd = open (argv[1], O\_RDONLY); do { bytes\_read = read (fd, buffer, sizeof (buffer)); for (i =0; i < bytes\_read; ++i) printf("%c", buffer[i]); } while (bytes\_read == sizeof (buffer)); close (fd); return 0; }

Компиляция файла readfile.c. Запуск программы (рис. 14)



*Компиляция файла readfile.c*

Снова от имени суперпользователи меняю владельца файла readfile. Далее меняю права доступа так, чтобы пользователь guest не смог прочесть содержимое файла (рис. 15)

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

*Смена владельца файла и прав доступа к файлу*

Проверка прочесть файл от имени пользователя guest. Прочесть файл не удается (рис. 16)



*Попытка прочесть содержимое файла*

Попытка прочесть тот же файл с помощью программы readfile, в ответ получаем непонятный текст (рис. 17)

Изображение выглядит как винт, резьбы, монохромный, черно-белый

Автоматически созданное описание со средним доверительным уровнем

*Попытка прочесть содержимое файла программой*

Попытка прочесть файл \etc\shadow с помощью программы, все еще получаем непонятный текст (рис. 18)

Изображение выглядит как снимок экрана, черно-белый, линия, черный

Автоматически созданное описание

*Попытка прочесть содержимое файла программой*

Пробуем прочесть эти же файлы от имени суперпользователя и чтение файлов проходит успешно (рис. 19)

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

*Чтение файла от имени суперпользователя*

Проверяем папку tmp на наличие атрибута Sticky, т.к. в выводе есть буква t, то атрибут установлен (рис. 20)



*Проверка атрибутов директории tmp*

От имени пользователя guest создаю файл с текстом, добавляю права на чтение и запись для других пользователей (рис. 21)

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

*Создание файла, изменение прав доступа*

Вхожу в систему от имени пользователя guest2, от его имени могу прочитать файл file01.txt (рис. 22), могу перезаписать в нём информацию (рис. 23)

Изображение выглядит как текст, Шрифт, снимок экрана, черный

Автоматически созданное описание

*Попытка чтения файла*

*Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание*

*Перезапись файла*

Далее пробуем удалить файл, снова получаем отказ (рис. 24)

**

*Попытка удалить файл*

От имени суперпользователя снимаем с директории атрибут Sticky (рис. 25)

Изображение выглядит как текст, Шрифт, снимок экрана, типография

Автоматически созданное описание

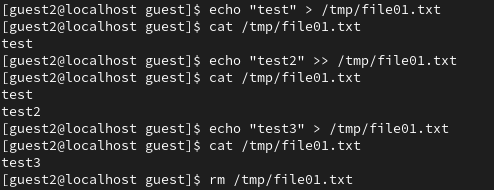
*Смена атрибутов файла*

Проверяем, что атрибут действительно снят (рис. 26)



*Проверка атрибутов директории*

Далее был выполнен повтор предыдущих действий. По результатам без Sticky-бита запись в файл и дозапись в файл осталась невозможной, зато удаление файла прошло успешно (рис. 27)



*Повтор предыдущих действий*

Возвращение директории tmp атрибута t от имени суперпользователя (рис. 28)

Изображение выглядит как текст, Шрифт, снимок экрана

Автоматически созданное описание

*Изменение атрибутов*

# Выводы

Изучила механизм изменения идентификаторов, применила SetUID- и Sticky-биты. 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.