Отчет по лабораторной работе №5

Основы информационной безопасности

Разанацуа Сара Естэлл НКАбд-02-23

Содержание

# 1 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.

# 2 Выполнение лабораторной работы

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

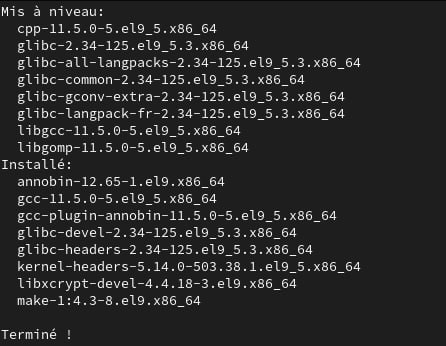


Рис. 1: Подготовка к лабораторной работе

* Создание файла simpled.c и запись в файл кода (рис. 2)

Создание файла

Рис. 2: Создание файла

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 следующти образом (рис. 3)

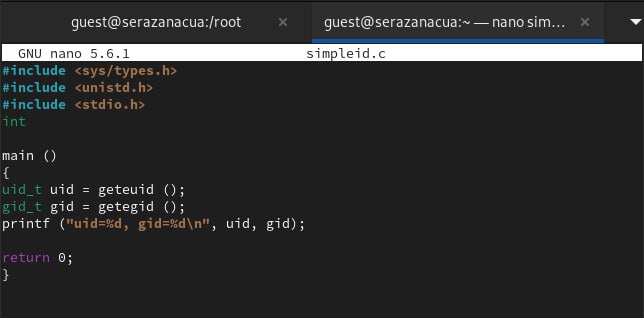


Рис. 3: Содержимое файла

* Компилирую файл, проверяю, что он скомпилировался (рис. 4)



Рис. 4: Компиляция файла

* Запускаю исполняемый файл. В выводе файла выписыны номера пользоватея и групп, от вывода при вводе if, они отличаются только тем, что информации меньше (рис. 5)

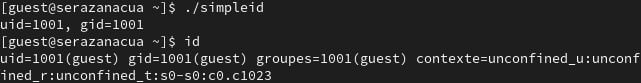


Рис. 5: Сравнение команд

* Создание, запись в файл и компиляция файла simpled2.c. Запуск программы (рис. 6)

Создание и компиляция файла

Рис. 6: Создание и компиляция 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; }

(рис. 7)

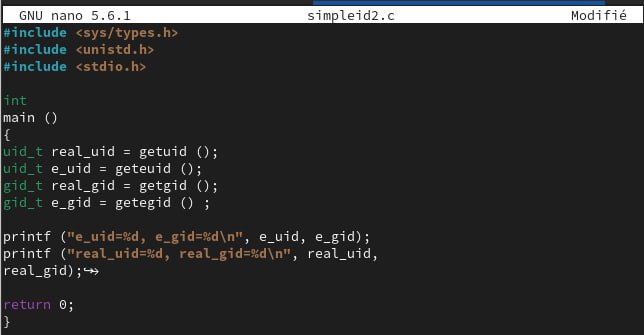


Рис. 7: Содержимое файла

* С помощью chown изменяю владельца файла на суперпользователя, с помощью chmod изменяю права доступа (рис. 9)

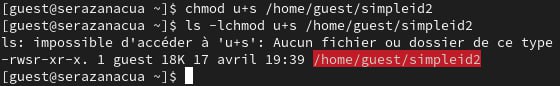


Рис. 8: Смена владельца файла и прав доступа к файлу

* Создание и компиляция файла readfile.c (рис. 10)

Создание и компиляция файла

Рис. 9: Создание и компиляция файла

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; }

(рис. 12)

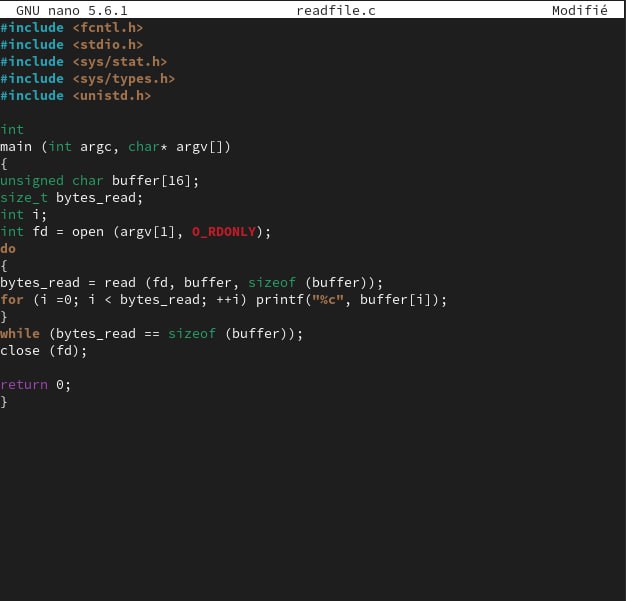


Рис. 10: Содержимое файла

* Снова от имени суперпользователи меняю владельца файла readfile. Далее меняю права доступа так, чтобы пользователь guest не смог прочесть содержимое файла (рис. 13)

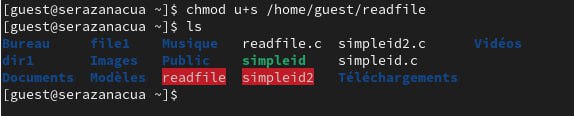


Рис. 11: Смена владельца файла и прав доступа к файлу

* Проверка прочесть файл от имени пользователя guest.Прочесть файл не удается (рис. 14)

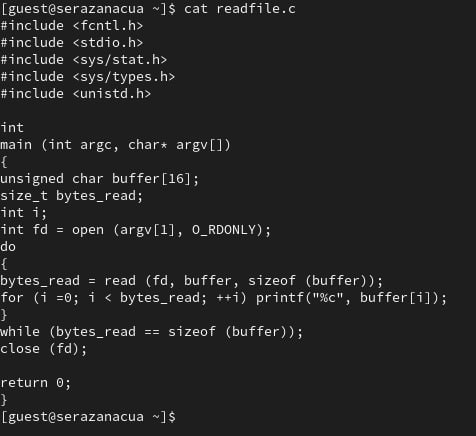


Рис. 12: Попытка прочесть содержимое файла

* Попытка прочесть тот же файл с помощью программы readfile, в ответ получаем “отказано в доступе” (рис. 15)

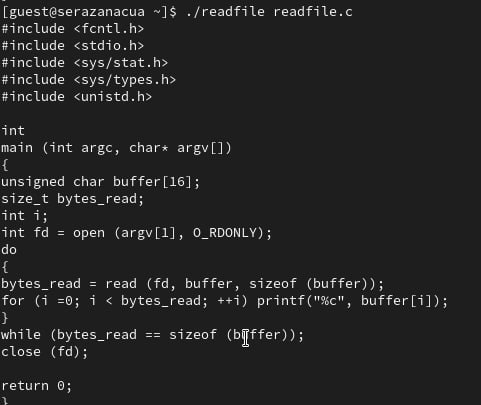


Рис. 13: Попытка прочесть содержимое файла программой

* Попытка прочесть файл \etc\shadow с помощью программы, все еще получаем отказ в доступе (рис. 16)

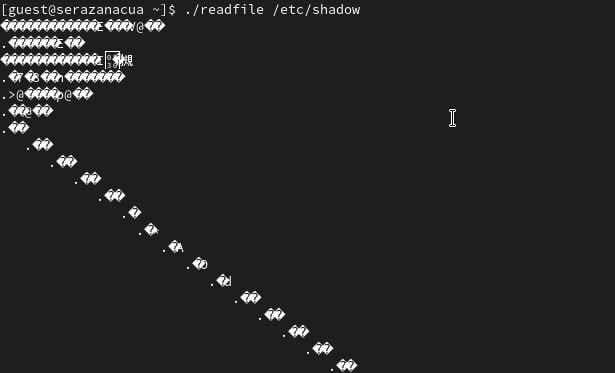


Рис. 14: Попытка прочесть содержимое файла программой

* Проверяем папку tmp на наличие атрибута Sticky, т.к. в выводе есть буква t, то атрибут установлен (рис. 17)

Проверка атрибутов директории tmp

Рис. 15: Проверка атрибутов директории tmp

# 3 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.

# Список литературы