Лабораторная работа № 5

Дискреционное разграничение прав в Linux. Исследование влияния дополнительных атрибутов

Алибаева Данагуль НБибд-01-18

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

1. Войдите в систему от имени пользователя guest.
2. Создайте программу simpleid.c
3. Скомплилируйте программу и убедитесь, что файл программы создан: gcc simpleid.c -o simpleid
4. Выполните программу simpleid: ./simpleid
5. Выполните системную программу id: id и сравните полученный вами результат с данными предыдущего пункта задания.
6. Усложните программу, добавив вывод действительных идентификаторов
7. Скомпилируйте и запустите simpleid2.c: gcc simpleid2.c -o simpleid2 ./simpleid2
8. От имени суперпользователя выполните команды
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.
13. Создайте программу readfile.c
14. Откомпилируйте её. gcc readfile.c -o readfile
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?
19. Проверьте, может ли программа readfile прочитать файл /etc/shadow? Отразите полученный результат и ваши объяснения в отчёте.
20. Выясните, установлен ли атрибут Sticky на директории /tmp, для чего выполните команду ls -l / | grep tmp
21. От имени пользователя guest создайте файл file01.txt в директории /tmp со словом test: echo “test” > /tmp/file01.txt
22. Просмотрите атрибуты у только что созданного файла и разрешите чтение и запись для категории пользователей «все остальные»: ls -l /tmp/file01.txt chmod o+rw /tmp/file01.txt ls -l /tmp/file01.txt
23. От пользователя guest2 (не являющегося владельцем) попробуйте прочитать файл /tmp/file01.txt: cat /tmp/file01.txt
24. От пользователя guest2 попробуйте дозаписать в файл /tmp/file01.txt слово test2 командой echo “test2” > /tmp/file01.txt Удалось ли вам выполнить операцию?
25. Проверьте содержимое файла командой cat /tmp/file01.txt
26. От пользователя guest2 попробуйте записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой echo “test3” > /tmp/file01.txt Удалось ли вам выполнить операцию?
27. Проверьте содержимое файла командой cat /tmp/file01.txt
28. От пользователя guest2 попробуйте удалить файл /tmp/file01.txt командой rm /tmp/fileOl.txt Удалось ли вам удалить файл?
29. Повысьте свои права до суперпользователя следующей командой su - и выполните после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp: chmod -t /tmp
30. Покиньте режим суперпользователя командой exit
31. От пользователя guest2 проверьте, что атрибута t у директории /tmp нет: ls -l / | grep tmp
32. Повторите предыдущие шаги. Какие наблюдаются изменения?
33. Удалось ли вам удалить файл от имени пользователя, не являющегося его владельцем? Ваши наблюдения занесите в отчёт.
34. Повысьте свои права до суперпользователя и верните атрибут t на директорию /tmp: su - chmod +t /tmp exit

# Теоретическое введение

Setuid и Setgid (сокращения от англ. set user ID upon execution — «установка ID пользователя во время выполнения» и англ. 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 владельца или группы исполняемого файла. [1]

В Unix-подобных системах приложение запускается с правами пользователя, вызвавшего указанное приложение. Это обеспечивает дополнительную безопасность, так как процесс с правами пользователя не сможет получить доступ на запись к важным системным файлам, например /etc/passwd, который принадлежит суперпользователю root. [1]

Если на исполняемый файл установлен бит suid, то при выполнении эта программа автоматически меняет «эффективный userID» на идентификатор того пользователя, который является владельцем этого файла. То есть, независимо от того — кто запускает эту программу, она при выполнении имеет права хозяина этого файла. [1]

Sticky bit — дополнительный атрибут файлов или каталогов в операционных системах семейства UNIX. [2]

Сегодня когда sticky bit используется в основном для каталогов, чтобы защитить в них файлы. Из такого каталога пользователь может удалить только те файлы, владельцем которых он является. Примером может служить каталог /tmp, в который запись открыта для всех пользователей, но нежелательно удаление чужих файлов. Установка атрибута производится утилитой chmod. [2]

В операционной системе Solaris для файлов, не являющихся программами, имеет строго противоположное действие — запрещает сохранение данных этого файла в системном кэше. [2]

# Выполнение лабораторной работы

1.Лабораторная работа выполнялась дома со следующими характеристиками техники:

– Intel(R) Core(TM) i5-8300H CPU @ 2.30GHz, 2304 МГц, ядер: 4, логических процессоров: 8

– ОС Майкрософт Windows 10 Pro

– VirtualBox верс. 6.1.26

1. Убедилась, что в системе установлен компилятор gcc с помощью команды gcc –v (рис 1.1).

Figure 1: 1.1. Проверка компилятора gcc

Figure 1: 1.1. Проверка компилятора gcc

1. Отключила систему запретов до очередной перезагрузки системы командой setenforce 0 (рис 1.2).

Figure 2: 1.2. Отключение системы запретов

Figure 2: 1.2. Отключение системы запретов

1. Ввела команду getenforce, которая должна выводить Permissive (рис 1.3).

Figure 3: 1.3. Ввод команды getenforce

Figure 3: 1.3. Ввод команды getenforce

1. Проверила компилятор языка С с помощью команд whereis gcc и whereis g++ (рис 1.4).

Figure 4: 1.4. Проверка компилятора языка С

Figure 4: 1.4. Проверка компилятора языка С

1. Создала программу simpleid.c (рис 1.5).

Figure 5: 1.5. Создание программы simpleid.c

Figure 5: 1.5. Создание программы simpleid.c

1. Скомплилировала программу и убедилась, что файл программы создан: gcc simpleid.c -o simpleid (рис 1.6).

Figure 6: 1.6. Компилирование программы simpleid.c

Figure 6: 1.6. Компилирование программы simpleid.c

1. Выполнила программу simpleid: ./simpleid (рис 1.7).

Figure 7: 1.7. Выполнение программы simpleid.c

Figure 7: 1.7. Выполнение программы simpleid.c

1. Выполнила системную программу id: id и сравнила полученный результат с данными предыдущего пункта задания (рис 1.8). Uid и gid совпадают.

Figure 8: 1.8. Выполнение системной программы id

Figure 8: 1.8. Выполнение системной программы id

1. Усложнила программу, добавив вывод действительных идентификаторов (рис 1.9). Получившуюся программу назвала simpleid2.c.

Figure 9: 1.9. Создание программы simpleid2.c

Figure 9: 1.9. Создание программы simpleid2.c

1. Скомпилировала и запустила simpleid2.c: gcc simpleid2.c -o simpleid2 ./simpleid2 (рис 1.10).

Figure 10: 1.10. Компилирование программы simpleid2.c

Figure 10: 1.10. Компилирование программы simpleid2.c

1. От имени суперпользователя выполнила команды chown root:guest /home/guest/simpleid2 chmod u+s /home/guest/simpleid2 (рис 1.11).

Figure 11: 1.11. Выполнение команды chown

Figure 11: 1.11. Выполнение команды chown

1. Повысила временно свои права с помощью su (рис 1.12).

Figure 12: 1.12. Повышение временно прав

Figure 12: 1.12. Повышение временно прав

1. Выполнила проверку правильности установки новых атрибутов и смены владельца файла simpleid2: ls -l simpleid2 (рис 1.13).

Figure 13: 1.13. Проверка правильности установки новых атрибутов

Figure 13: 1.13. Проверка правильности установки новых атрибутов

1. Запустила simpleid2 и id: ./simpleid2 id Сравнила результаты (рис 1.14). Программа запустилась от root с uid = 0, но реальный его id = 1002.

Figure 14: 1.14. Запуск команд simpleid2 и id

Figure 14: 1.14. Запуск команд simpleid2 и id

1. Проделала тоже самое относительно SetGID-бита (рис 1.15).

Figure 15: 1.15. Повторный запуск команд с SetGID-бита

Figure 15: 1.15. Повторный запуск команд с SetGID-бита

1. Создала программу readfile.c (рис 1.16).

Figure 16: 1.16. Создание программы readfile.c

Figure 16: 1.16. Создание программы readfile.c

1. Откомпилировала её с помощью команды gcc readfile.c -o readfile (рис 1.17).

Figure 17: 1.17. Компилирование программы readfile.c

Figure 17: 1.17. Компилирование программы readfile.c

1. Сменила владельца у файла readfile.c и изменила права так, чтобы только суперпользователь (root) мог прочитать его, a guest не мог (рис 1.18).

Figure 18: 1.18. Смена владельца у файла readfile.c

Figure 18: 1.18. Смена владельца у файла readfile.c

1. Проверила, что пользователь guest не может прочитать файл readfile.c (рис 1.19).

Figure 19: 1.19. Проверка пользователя guest

Figure 19: 1.19. Проверка пользователя guest

1. Сменила у программы readfile владельца и установила SetU’D-бит (рис 1.20).

Figure 20: 1.20. Смена у программы readfile владельца

Figure 20: 1.20. Смена у программы readfile владельца

1. Проверила, может ли программа readfile прочитать файл readfile.c (рис 1.21). Да, может.

Figure 21: 1.21. Проверка чтения файла readfile.c

Figure 21: 1.21. Проверка чтения файла readfile.c

1. Проверила, может ли программа readfile прочитать файл /etc/shadow (рис 1.22). Да, может.

Figure 22: 1.22. Проверка чтения файла /etc/shadow

Figure 22: 1.22. Проверка чтения файла /etc/shadow

1. Выяснила, установлен ли атрибут Sticky на директории /tmp, с помощью команды ls -l / | grep tmp (рис 1.23).

Figure 23: 1.23. Установление атрибута Sticky

Figure 23: 1.23. Установление атрибута Sticky

1. От имени пользователя guest создала файл file01.txt в директории /tmp со словом test: echo “test” > /tmp/file01.txt (рис 1.24).

Figure 24: 1.24. Создание файла file01.txt в директории /tmp

Figure 24: 1.24. Создание файла file01.txt в директории /tmp

1. Просмотрела атрибуты у только что созданного файла и разрешила чтение и запись для категории пользователей «все остальные»: ls -l /tmp/file01.txt chmod o+rw /tmp/file01.txt ls -l /tmp/file01.txt (рис 1.25).

Figure 25: 1.25. Разрешение чтения и записи

Figure 25: 1.25. Разрешение чтения и записи

1. От пользователя guest2 (не являющегося владельцем) попробовала прочитать файл /tmp/file01.txt: cat /tmp/file01.txt (рис 1.26).

Figure 26: 1.26. Попытка прочитать файл file01.txt

Figure 26: 1.26. Попытка прочитать файл file01.txt

1. От пользователя guest2 попробовала дозаписать в файл /tmp/file01.txt слово test2 командой echo “test2” > /tmp/file01.txt (рис 1.27). Операцию выполнить удалось.

Figure 27: 1.27. Попытка дозаписать в файл file01.txt

Figure 27: 1.27. Попытка дозаписать в файл file01.txt

1. Проверила содержимое файла командой cat /tmp/file01.txt (рис 1.28).

Figure 28: 1.28. Проверка содержимого файла file01.txt

Figure 28: 1.28. Проверка содержимого файла file01.txt

1. От пользователя guest2 попробовала записать в файл /tmp/file01.txt слово test3, стерев при этом всю имеющуюся в файле информацию командой echo “test3” > /tmp/file01.txt (рис 1.29). Операцию выполнить удалось.

Figure 29: 1.29. Попытка записать в файл file01.txt

Figure 29: 1.29. Попытка записать в файл file01.txt

1. Проверила содержимое файла командой cat /tmp/file01.txt (рис 1.30).

Figure 30: 1.30. Проверка содержимого файла file01.txt

Figure 30: 1.30. Проверка содержимого файла file01.txt

1. От пользователя guest2 попробовала удалить файл /tmp/file01.txt командой rm /tmp/fileOl.txt (рис 1.31). Команду выполнить не удалось.

Figure 31: 1.31. Попытка удалить файл file01.txt

Figure 31: 1.31. Попытка удалить файл file01.txt

1. Повысила свои права до суперпользователя следующей командой su - и выполнила после этого команду, снимающую атрибут t (Sticky-бит) с директории /tmp: chmod -t /tmp (рис 1.32).

Figure 32: 1.32. Повышение своих прав до суперпользователя

Figure 32: 1.32. Повышение своих прав до суперпользователя

1. Покинула режим суперпользователя командой exit (рис 1.33).

Figure 33: 1.33. Ввод команды exit

Figure 33: 1.33. Ввод команды exit

1. От пользователя guest2 проверила, что атрибута t у директории /tmp нет: ls -l / | grep tmp (рис 1.34).

Figure 34: 1.34. Проверка атрибута t

Figure 34: 1.34. Проверка атрибута t

1. Повторила предыдущие шаги (рис 1.35).

Figure 35: 1.35. Повтор предыдущих шагов

Figure 35: 1.35. Повтор предыдущих шагов

1. Файл удалить удалось (рис 1.36). Получается, что если есть атрибут Sticky t, то удалять файл может только владелец файла. Но если атрибута нет, то удалить этот файл может любой пользователь.

Figure 36: 1.36. Удаление файла file01.txt

Figure 36: 1.36. Удаление файла file01.txt

1. Повысила свои права до суперпользователя и вернула атрибут t на директорию /tmp: su - chmod +t /tmp exit (рис 1.37).

Figure 37: 1.37. Возвращение атрибута t

Figure 37: 1.37. Возвращение атрибута t

# Выводы

В результате выполнения работы я изучила механизмы изменения идентификаторов, применение SetUID- и Sticky-битов. 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

1. SetUID // URL: https://ru.wikipedia.org/wiki/Suid#:~:text=setuid%20%D0%B8%20setgid%20%D1%8F%D0%B2%D0%BB%D1%8F%D1%8E%D1%82%D1%81%D1%8F%20%D1%84%D0%BB%D0%B0%D0%B3%D0%B0%D0%BC%D0%B8,%2Fetc%2Fpasswd%2C%20%D0%BA%D0%BE%D1%82%D0%BE%D1%80%D1%8B%D0%B9%20%D0%BF%D1%80%D0%B8%D0%BD%D0%B0%D0%B4%D0%BB%D0%B5%D0%B6%D0%B8%D1%82%20%D1%81%D1%83%D0%BF%D0%B5%D1%80%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8E%20root (дата обращения: 12.11.2021).
2. Sticky bit // URL: https://ru.wikipedia.org/wiki/Sticky\_bit (дата обращения: 12.11.2021).