

Отчёта по лабораторной работе № 3

Основы Информационной безопасности

Кинсиклунон Доря Флора

Содержание

0.1	Цель работы	4
0.2	Теоретическое введение	4
0.3	Теоретическое введение	5
0.4	Выполнение лабораторной работы	7
0.5	Выводы	9
0.6	Список литературы	10

Список иллюстраций

1	Рис. 3.4: Изменение атрибутов	9
---	---	---

Список таблиц

1	Установление права и разрешённых действий	10
---	---	----

0.1 Цель работы

Получение практических навыков работы в консоли с атрибутами файлов для групп пользователей.

0.2 Теоретическое введение

В операционной системе Linux есть много отличных функций безопасности, но одна из самых важных - это система прав доступа к файлам. Изначально каждый файл имел три параметра доступа. Вот они: • Чтение - разрешает получать содержимое файла, но на запись нет. Для каталога позволяет получить список файлов и каталогов, расположенных в нем • Запись - разрешает записывать новые данные в файл или изменять существующие, а также позволяет создавать и изменять файлы и каталоги • Выполнение - невозможно выполнить программу, если у нее нет флага выполнения. Этот атрибут устанавливается для всех программ и скриптов, именно с помощью него система может понять, что этот файл нужно запускать как программу Каждый файл имеет три категории пользователей, для которых можно устанавливать различные сочетания прав доступа: • Владелец - набор прав для владельца файла, пользователя, который его создал или сейчас установлен его владельцем. Обычно владелец имеет все права, чтение, запись и выполнение • Группа - любая группа пользователей, существующая в системе и привязанная к файлу. Но это может быть только одна группа и обычно это группа

владельца, хотя для файла можно назначить и другую группу

- Остальные - все пользователи, кроме владельца и пользователей, входящих в группу файла Команды, которые могут понадобиться при работе с правами доступа:
- “ls -l” - для просмотра прав доступа к файлам и каталогам
- “chmod категория действие флаг файл или каталог” - для изменения прав доступа к файлам и каталогам (категорию действие и флаг можно заменить на набор из трех цифр от 0 до 7)

Значения флагов прав:

- — - нет никаких прав
- -x - разрешено только выполнение файла, как программы, но не изменение и не чтение
- -w - разрешена только запись и изменение файла
- -wx - разрешено изменение и выполнение, но в случае с каталогом, невозможно посмотреть его содержимое
- r- - права только на чтение
- r-x - только чтение и выполнение, без права на запись
- rw- - права на чтение и запись, но без выполнения
- rwx - все права

0.3 Теоретическое введение

Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется.

В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним

можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов, а производить так называемую дельтакомпрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объём хранимых данных.

Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить (слить) изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом.

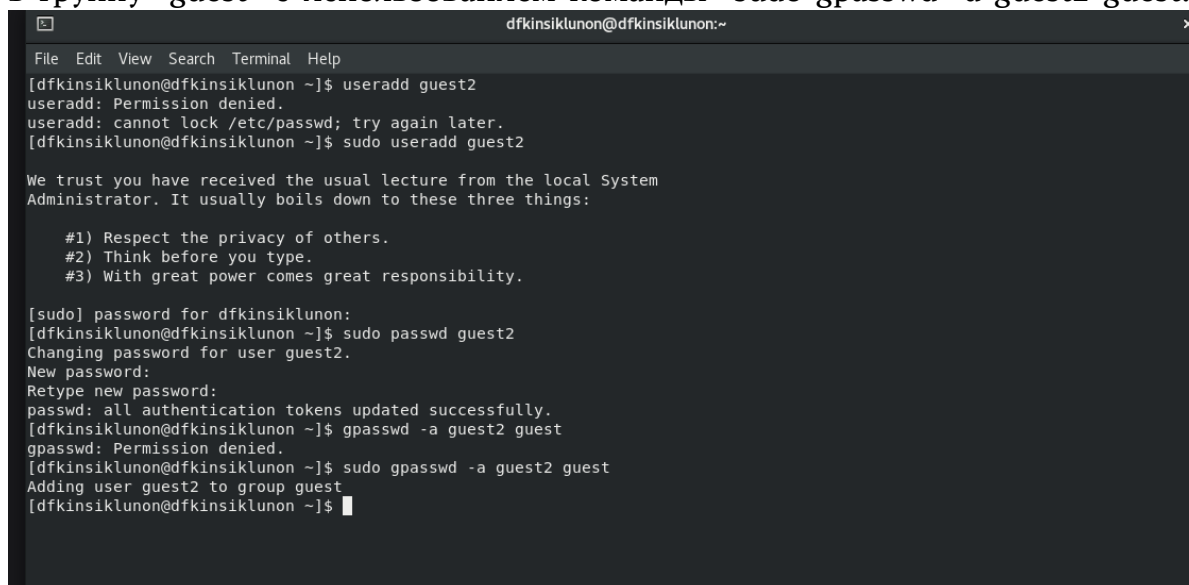
Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить.

В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным.

Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

0.4 Выполнение лабораторной работы

В предыдущей лабораторной работе, операционная система создала учетную запись пользователя с именем “guest2” (поскольку пользователь “guest” уже был создан в предыдущей лабораторной работе), используя команду “sudo useradd guest2” и установила пароль для этого пользователя с помощью команды “sudo passwd guest2.” Затем пользователь “guest2” был добавлен в группу “guest” с использованием команды “sudo gpasswd -a guest2 guest.”



```
dfkinsiklunon@dfkinsiklunon:~  
File Edit View Search Terminal Help  
[dfkinsiklunon@dfkinsiklunon ~]$ useradd guest2  
useradd: Permission denied.  
useradd: cannot lock /etc/passwd; try again later.  
[dfkinsiklunon@dfkinsiklunon ~]$ sudo useradd guest2  
  
We trust you have received the usual lecture from the local System  
Administrator. It usually boils down to these three things:  
  
#1) Respect the privacy of others.  
#2) Think before you type.  
#3) With great power comes great responsibility.  
  
[sudo] password for dfkinsiklunon:  
[dfkinsiklunon@dfkinsiklunon ~]$ sudo passwd guest2  
Changing password for user guest2.  
New password:  
Retype new password:  
passwd: all authentication tokens updated successfully.  
[dfkinsiklunon@dfkinsiklunon ~]$ gpasswd -a guest2 guest  
gpasswd: Permission denied.  
[dfkinsiklunon@dfkinsiklunon ~]$ sudo gpasswd -a guest2 guest  
Adding user guest2 to group guest  
[dfkinsiklunon@dfkinsiklunon ~]$
```

Затем я вошла в систему с двумя разными пользователями на двух разных консолях с помощью команд “su - guest” и “su - guest2.” С помощью команды “pwd” я определила, что оба пользователя находятся в своих домашних каталогах, что соответствует строке приглашения командной строки. Я подтвердила имена пользователей с помощью команды “whoami” и получила соответственно “guest” и “guest2.” С использованием команд “groups guest” и “groups guest2” я определила, что пользователь “guest” является участником группы “guest,” а пользователь “guest2” является участником как группы “guest,” так и группы “guest2.” Я сравнила эту информацию с выводом команд “id -Gn guest,” “id -Gn guest2,” “id -G guest” и “id -G guest2.” Данные совпали, за исключением второй команды “id -G,” которая отобразила номера групп 1001 и 1002, что также является верным.

```
guest@dfkinsiklunon:~  
[dfkinsiklunon@dfkinsiklunon ~]$ su - guest  
su: group uest does not exist  
[dfkinsiklunon@dfkinsiklunon ~]$ su - guest  
Password:  
su: Authentication failure  
[dfkinsiklunon@dfkinsiklunon ~]$ su - guest  
Password:  
[guest@dfkinsiklunon ~]$ pwd  
/home/guest  
[guest@dfkinsiklunon ~]$ whoami  
guest  
[guest@dfkinsiklunon ~]$ groups guest  
guest : guest  
[guest@dfkinsiklunon ~]$ id -Gn guest  
guest  
[guest@dfkinsiklunon ~]$ id -G guest  
1001  
[guest@dfkinsiklunon ~]$  
  
guest2@dfkinsiklunon:~  
[dfkinsiklunon@dfkinsiklunon ~]$ su - guest2  
su: group uest2 does not exist  
[dfkinsiklunon@dfkinsiklunon ~]$ su - guest2  
Password:  
[guest2@dfkinsiklunon ~]$ pwd  
/home/guest2  
[guest2@dfkinsiklunon ~]$ whoami  
guest2  
[guest2@dfkinsiklunon ~]$ groups guest2  
guest2 : guest2 guest  
[guest2@dfkinsiklunon ~]$ id -Gn guest2  
guest2 guest  
[guest2@dfkinsiklunon ~]$ id -G guest2  
1002 1001  
[guest2@dfkinsiklunon ~]$
```

Просмотрела файл /etc/group командой “cat /etc/group”

```
guest@dfkinsiklunon:~  
ssh_keys:x:993:  
printadmin:x:992:  
rtkit:x:172:  
pipewire:x:991:  
pulse-access:x:990:  
pulse-rt:x:989:  
pulse:x:171:  
qemu:x:107:  
clevis:x:988:  
usbmuxd:x:113:  
gluster:x:987:  
rpc:x:32:  
chrony:x:986:  
avahi:x:70:  
brlapi:x:985:  
setroubleshoot:x:984:  
saslauth:x:76:  
libstoragegmt:x:983:  
dnsmasq:x:982:  
sssd:x:981:  
libvirt:x:980:  
cockpit-ws:x:979:  
cockpit-wsinstance:x:978:  
flatpak:x:977:  
stapusr:x:156:  
stapsys:x:157:  
stapdev:x:158:  
colord:x:976:  
rpcuser:x:29:  
gdm:x:42:  
gnome-initial-setup:x:975:  
pesign:x:974:  
sshd:x:74:  
slocate:x:21:  
tcpdump:x:72:  
dfkinsiklunon:x:1000:  
guest:x:1001:guest2  
guest2:x:1002:  
[guest@dfkinsiklunon ~]$
```

От имени пользовате-

ля “guest2” я зарегистрировала этого пользователя в группе “guest” с помощью

команды “newgrp guest.” Затем, от имени пользователя “guest,” я изменила права на директорию “/home/guest,” разрешив все действия для пользователей в этой группе с помощью команды “chmod g+rxw /home/guest.” Также, используя пользователя “guest,” я сняла все атрибуты с директории “/home/guest/dir1” с помощью команды “chmod 000 dir1” и проверила правильность снятия атрибутов с помощью команды “ls -l.”

```

guest@dfkinsiklunon:~$ chmod g+rxw /home/guest
guest@dfkinsiklunon:~$ chmod 000 /home/guest/dir1
guest@dfkinsiklunon:~$ ls -l
total 0
drwxr-xr-x. 2 guest guest 6 Sep 16 18:47 Desktop
d-----r-x. 2 guest guest 6 Sep 16 18:57 dir1
drwxr-xr-x. 2 guest guest 6 Sep 16 18:47 Documents
drwxr-xr-x. 2 guest guest 6 Sep 16 18:47 Downloads
drwxr-xr-x. 2 guest guest 6 Sep 16 18:47 Music
drwxr-xr-x. 2 guest guest 241 Sep 16 19:02 Pictures
drwxr-xr-x. 2 guest guest 6 Sep 16 18:47 Public
drwxr-xr-x. 2 guest guest 6 Sep 16 18:47 Templates
drwxr-xr-x. 2 guest guest 6 Sep 16 18:47 Videos
guest@dfkinsiklunon:~$

[guest2@dfkinsiklunon:~]$ su -guest2
su: group uest2 does not exist
[guest2@dfkinsiklunon:~]$ su - guest2
Password:
[guest2@dfkinsiklunon:~]$ pwd
/home/guest2
[guest2@dfkinsiklunon:~]$ whoami
guest2
[guest2@dfkinsiklunon:~]$ groups guest2
guest2 : guest2 guest
[guest2@dfkinsiklunon:~]$ id -Gn guest2
guest2 guest
[guest2@dfkinsiklunon:~]$ id -G guest2
1002 1001
[guest2@dfkinsiklunon:~]$ newgrp guest
bash: newgrp: command not found...
[guest2@dfkinsiklunon:~]$ newgrp guest
[guest2@dfkinsiklunon:~]$

```

Рис. 1: Рис. 3.4: Изменение атрибутов

Заполним таблицу «Установленные права и разрешённые действия» 3.1.

Создание файла: “echo”text” > /home/guest/dir1/file2” Удаление файла: “rm -r /home/guest/dir1/file1” Запись в файл: “echo”textnew” > /home/guest/dir1/file1” Чтение файла: “cat /home/guest/dir1/file1” Смена директории: “cd dir1” Просмотр файлов в директории: “ls dir1” Переименование файла: “mv /home/guest/dir1/file1 filenew” Смена атрибутов файла: “chattr -a /home/guest/dir1/file1”

0.5 Выводы

В итоге выполнения данной лабораторной работы я приобрела практические навыки работы с атрибутами файлов для групп пользователей.

Интересно отметить, что работа с атрибутами файлов и правами доступа в консоли Linux является важной частью администрирования системы. Эти навы-

Таблица 1: Установление права и разрешённых действий

Права директории	000	100	200	300	400	500	600	700
Права файла	000	100	200	300	400	500	600	700
Создание файла	-	-	-	+	-	-	-	+
Удаление файла	-	-	-	+	-	-	-	+
Запись в файл	-	+	-	+	-	+	-	+
Чтение файла	-	+	-	+	-	+	-	+
Смена директории	-	-	-	+	-	+	-	+
Просмотр файлов в директории	-	-	-	-	+	+	+	+
Переименование файла	-	-	-	+	-	-	-	+
Смена атрибутов файла	-	-	-	+	-	-	-	+

ки позволяют управлять безопасностью данных и ресурсов, а также определить, какие пользователи и группы имеют доступ к определенным файлам и каталогам. Это особенно важно в корпоративных и многопользовательских средах, где необходимо строго контролировать доступ к информации. Работа в командной строке также позволяет более гибко управлять этими настройками и быстро вносить изменения при необходимости.

0.6 Список литературы

Права доступа к файлам в Linux [Электронный ресурс]. 2019. URL: <https://losst.ru/prava-dostupa-k-fajlam-v-linux>.