Отчёта по лабораторной работе № 3

Основы Информационной безопасности

Кинсиклунон Доря Флора

Содержание

## 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 несколь ких 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.

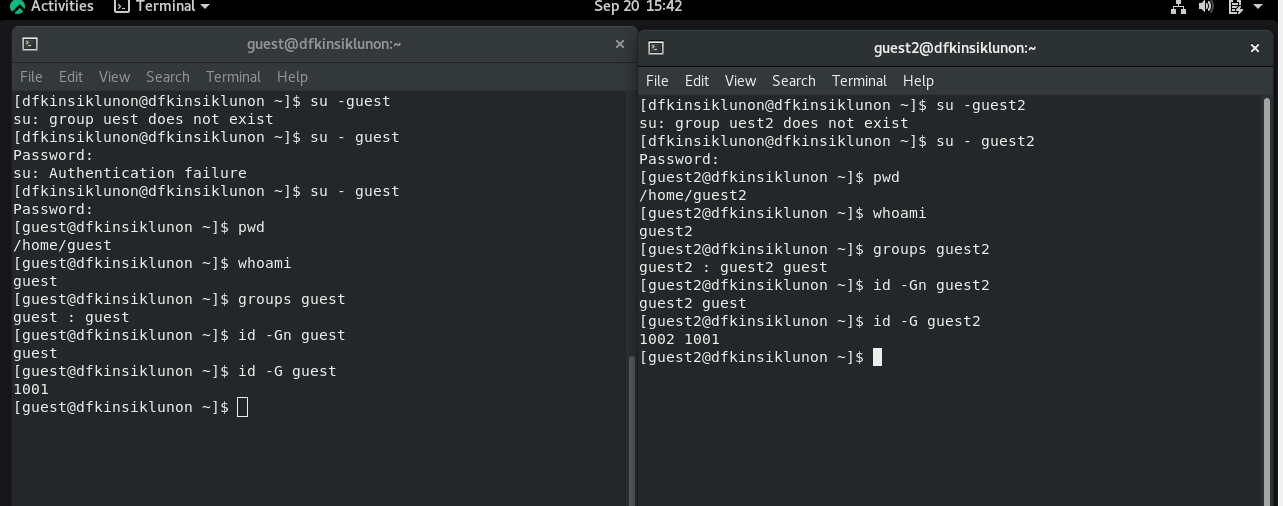
Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно 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.

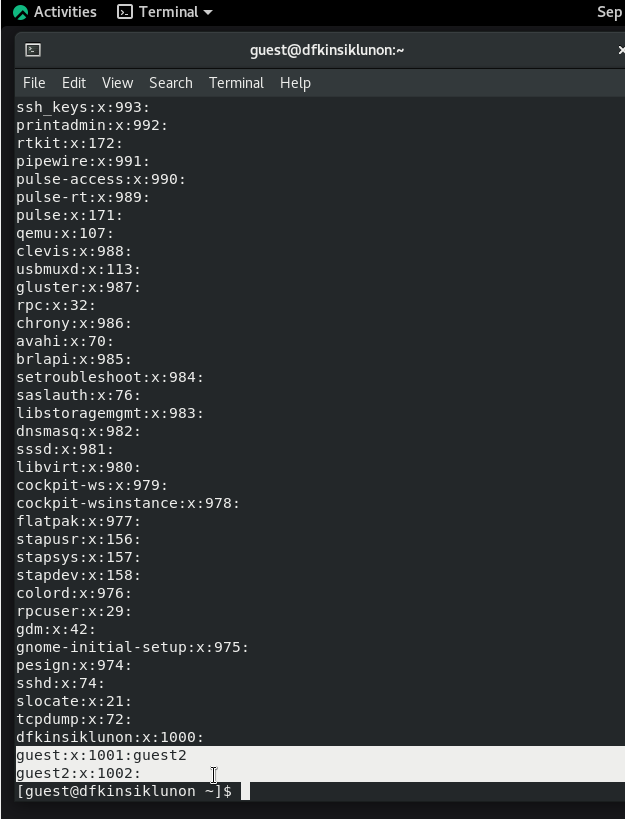
Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

## 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.” 

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 пользователей 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. 

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

 От имени пользователя “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+rwx /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.”

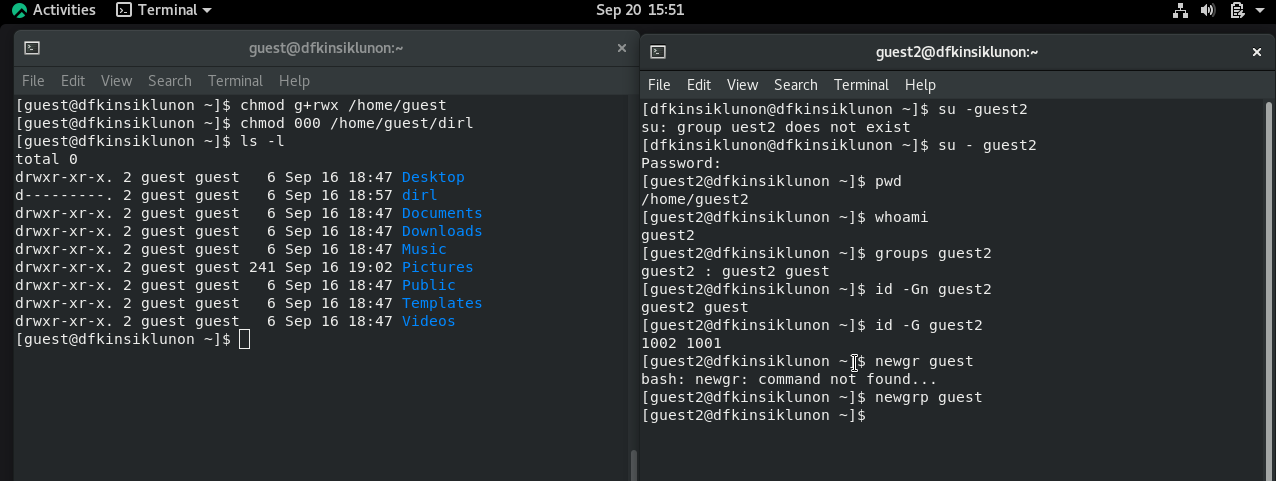


Рис. 3.4: Изменение атрибутов

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 навыки 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.