**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ**

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ**

**ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ» (ДГТУ)

Факультет «Информатика и вычислительная техника»

Кафедра «Кибербезопасность информационных систем»

**Лабораторная работа № 8**

на тему «Дискреционное (избирательное) разграничение доступа (Discretionary Access Control, DAC) в ОССН Astra Linux.»

|  |
| --- |
| Выполнил: студент группы ВКБ43 |
| Ковалев Данил Петрович |
| (Фамилия, имя, отчество) |
| Проверил: |
| Скляров Алексей Викторович |
| (Фамилия, имя, отчество) |

# **Цель работы:** изучить возможности ОССН Astra Linux при конфигурировании дискреционной модели разграничения доступа.

**Задание 1.** Авторизоваться в ОССН в графическом режиме с учётной записью пользователя user1 (уровень доступа — 0, неиерархические категории — нет, уровень целостности — «Высокий»).

Войдем в систему за пользователя user1, которого создали в прошлой лабораторной работе. Рабочий стол от лица user1 представлен на рисунке 1.

Изображение выглядит как снимок экрана, текст, графический дизайн, круг

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 1 – интерфейс рабочего стола для user1

**Задание 2.** Создать пакетный файл в домашней директории пользователя user1 (если пользователя с таким именем нет – создайте его, в качестве содержимого пакетного файла можно использовать задание из лабораторной работы № 8).

Откроем терминал и от лица пользователя user1 создадим файл, как сказано в условии. Для этого будет использоваться редактор nano. Для создания файла нужно использовать команду: “nano some\_file.txt”. В данный файл было записано просто выражение, чтобы он сохранился. Результат представлен на рисунке 2.

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, Значок на компьютере

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 2 – новый текстовый файл

**Задание 3.** Запустить терминал Fly.

Чтобы запустить терминал открываем “Пуск”, а потом выбираем приложение терминал.

**Задание 4.** Проверить текущие права доступа к созданному файлу.

Для выполнения задания можно воспользоваться командой “stat some\_file.txt”. В результате получится то, что представлено на рисунке 3.

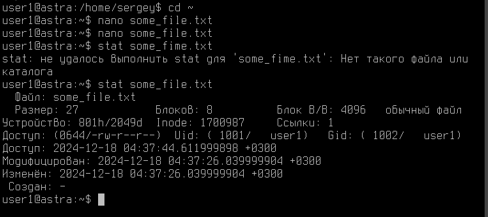


Рисунок 3 – подробная информация о созданном файле

**Задание 5.** Авторизоваться в ОССН в графическом режиме с учётной записью пользователя user3.

Для выполнения задания использовался user2 с прошлой лабораторной работы. Чтобы перейти за него, нужно в терминале ввести команду su user2. Результат выполнения представлен на рисунке 4.

Изображение выглядит как текст, Шрифт, снимок экрана, черный

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 4 – смена пользователя для задания

**Задание 6.** Попытайтесь запустить указанный файл от имени пользователя user3.

Попробуем просто открыть файл через nano. Видим, что его содержимое пусто и написано снизу, что отказано в доступе. От лица пользователя user2 мы не имеем право редактировать данный файл. Результат представлен на рисунке 5.

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

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 5 – попытка открытия файла some\_file.txt от имени user2

**Задание 7.** От имени пользователя user1 назначьте полный доступ к пакетному файлу.

Вернемся теперь за пользователя user1, используя команду “su user1”. Выдадим теперь полные права, используя команду chmod a+rw some\_file.txt. Команда представлена на рисунке 6.

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 6 – смена прав для файла

**Задание 8.** Попытайтесь запустить указанный файл от имени пользователя user3. Объясните полученный результат.

Если попробуем теперь запустить файл от имени другого пользователя, то также получим ошибку. Это происходит по той причине, потому что мы не назначили доступ другим субъектам в системе к данному объекту. Результат ошибки доступа представлен на рисунке 7.

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, компьютер

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 7 – ошибка доступа к файлу от лица user2

**Задание 9.** Измените пользователя user1, переместив его в ту же группу, к которой принадлежит user3 (по умолчанию эта группа также называется user3).

В нашем случае группа будет называться user2, добавим к ней пользователя user1, как представлено на рисунке 8.

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, веб-страница

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 8 – добавление пользователя user1 в группу user2

**Задание 10.** Завершите сеанс пользователя user3 (команда logout) и снова войдите в систему от его имени.

Команда logout не работает, в нашем случае использовался exit, как представлено на рисунке 9.

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 9 – выход пользователя из сеанса

**Задание 11.** Попытайтесь снова запустить пакетный файл, созданный в пункте 3. Объясните полученный результат.

Если попробовать заново запустить, то выдает ошибку доступа для чтения, такое происходит по той причине, так как пользователь не указывал, что все участники группы имеют права читать файл. Результат представлен на рисунке 10.

Изображение выглядит как текст, программное обеспечение, Мультимедийное программное обеспечение, компьютер

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 10 – попытка просмотреть файл от лица user2

**Задание 12**. Удалите пользователя user1.

Для удаления пользователя воспользуемся командой deluser user1 от имени администратора. После этого удалим сразу и группу. Результат выполнения задания представлено на рисунке 11.

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

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 11 – удаление пользователя и группы

**Задание 13.** Для одного и тоже файла создайте мягкую и жесткую ссылку в домашнем каталоге. Попробуйте создать ссылки одновременно для нескольких файлов.

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

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 12 – создание пустого текстового файла

Для создания жесткой ссылкой нужно воспользоваться командной: “ ln original\_file.txt hard\_link.txt”. Теперь hard\_link.txt — это просто другое имя для тех же данных на диске, что и original\_file.txt. У них один и тот же inode (идентификатор файла в файловой системе).

Для создания символической ссылки используется команда ln с ключом -s. Файл soft\_link.txt — это небольшая запись, которая просто содержит путь к файлу original\_file.txt. Он указывает на имя файла, а не на данные напрямую.

**Задание 14.** Проверьте работу файлового менеджера mc (в случае отсутствия файлового менеджера установите его).

В нашем случае по умолчанию он был, поэтому в терминал просто введем “mc”, в результате получилось то, что представлено на рисунке 13.

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

Содержимое, созданное искусственным интеллектом, может быть неверным.

Рисунок 13 – результат выполнения задания 14

**Задание 15.** Запишите абсолютный и символьный варианты команды, устанавливающей права доступа к файлу в соответствии с вариантом задания (номер варианта задания соответствует номеру студента в журнале группы).

**Вариант 11.** Полный доступ для пользователя, доступ на чтение и выполнение для группы, доступ на чтение и запись для остальных.

Требуется установить следующие права:

Пользователь (владелец): rwx (чтение, запись, выполнение = полный доступ = 7). Группа: r-x (чтение и выполнение = 5). Остальные (others): rw- (чтение и запись = 6).

Получается нам нужно выдать такие права на владельца: 4 (r) + 2 (w) + 1 (x) = 7. На группу: 4 (r) + 0 (—) + 1 (x) = 5. Остальные: 4 (r) + 2 (w) + 0 (—) = 6.

То есть в итоге, нам подходит данная команда: chmod 756 имя\_файла.

**Контрольные вопросы**

1. Как организуется разграничение доступа к файлам в Linux?

Разграничение доступа в Linux построено на трех ключевых механизмах:

Владелец и группа: Каждому файлу и каталогу назначается один владелец (user) и одна группа (group).

Права доступа (Permissions): для трех категорий субъектов — владельца (u), члена группы (g) и всех остальных (o) — отдельно задаются права:

* + Чтение (r) — просмотр содержимого файла или списка файлов в каталоге.
  + Запись (w) — изменение файла; для каталога — создание, переименование и удаление файлов в нём.
  + Выполнение (x) — запуск файла как программы/скрипта; для каталога — право войти в него (cd).

Дополнительные атрибуты (Special Bits): Расширенные механизмы для особых случаев: SUID, SGID и Sticky Bit.

Эти механизмы управляются командами chown (смена владельца/группы), chmod (смена прав) и chattr (смена атрибутов).

В ОССН Astra Linux этот базовый механизм дополнен и усилен:

Мандатное управление доступом (МУД): это основное отличие. Каждому субъекту (пользователю) и объекту (файлу) присваиваются метки безопасности (уровень классификации и неиерархические категории). Пользователь может получить доступ к файлу только если его уровень доверия не ниже уровня файла, а его категории полностью покрывают категории файла. Это предотвращает утечку данных "сверху вниз".

Дискреционное управление доступом (ДУД): это классическая модель прав доступа (rwx), описанная выше. Она работает совместно с МУД.

2. Как изменить права доступа к файлу? Как сделать это с помощью битовой строки?

Права изменяются командой chmod (change mode). С помощью битовой (восьмеричной)строки.Это самый распространенный способ. Права кодируются тремя цифрами (от 0 до 7), где каждая цифра — сумма прав для одной категории:

* 4 — чтение (r)
* 2 — запись (w)
* 1 — выполнение (x)

Первая цифра — права для владельца (user), вторая — для группы (group), третья — для остальных (others).

3. Что такое stisky-bit, SUID и SGID, для чего предназначены эти особые признаки?

Это специальные биты прав доступа, которые меняют стандартное поведение системы.

* Sticky Bit (t\*\* на месте x у others)\*\*:
  + Назначение**:** применяется к каталогу. Разрешает удаление или переименование файла в этом каталоге только его владельцу, владельцу каталога или root. Даже если у пользователя есть права на запись (w) в каталог, он не сможет удалить чужой файл.
  + Пример**:** Каталог /tmp. Все пользователи могут создавать файлы, но не могут удалить файлы, созданные другими.
  + Установка: chmod +t /directory или chmod 1755 /directory
* SUID (Set User ID, s на месте x у user):
  + Назначение: применяется к исполняемому файлу. Когда пользователь запускает такой файл, процесс выполняется с правами владельца файла, а не запустившего его пользователя.
  + Пример: Утилита passwd (/usr/bin/passwd). Она имеет SUID-бит и принадлежит root. Это позволяет обычному пользователю изменять свой пароль, который записывается в защищённый файл /etc/shadow, доступный только root.
  + Установка: chmod u+s /path/to/binary
* SGID (Set Group ID, s на месте x у group):
  + Назначение:
    1. На исполняемый файл: аналогично SUID, но процесс выполняется с правами группы файла.
    2. На каталог: Все новые файлы и подкаталоги, созданные в этом каталоге, будут наследовать группу-владельца каталога, а не первичную группу пользователя, который их создал. Это полезно для общих папок.
  + Установка: chmod g+s /directory

4. Какие могут быть файловые системы в Linux?

Linux поддерживает огромное количество файловых систем. Вот основные типы:

* Нативные (журналируемые) для дисков: ext4 (стандартная для многих дистрибутивов), XFS (высокая производительность с большими файлами), Btrfs (с поддержкой снапшотов, сжатия, избыточности), JFS.
* Специализированные: ZFS (продвинутые функции по управлению данными, но лицензионные сложности).
* Сетевые: NFS (сетевой файловый доступ от Sun), CIFS/SMB (для работы с shares Windows).
* Виртуальные (в оперативной памяти):
  + tmpfs — размещается в ОЗУ. Идеально для временных файлов (например, /tmp, /run). Быстрая, но данные стираются после перезагрузки.
  + proc — виртуальная ФС, представляющая информацию о процессах и ядре в виде файлов (/proc).
  + sysfs — предоставляет информацию об устройствах и драйверах (/sys).
* Для совместимости: vfat/fat16/fat32 (флешки), ntfs (диски Windows, с поддержкой чтения/записи через драйвер ntfs-3g).

В ОССН Astra Linux по умолчанию используется ext4.

5. Какие типы файлов существуют в Linux?

Первый символ в выводе ls -l указывает на тип файла:

* **-** — Обычный файл (regular file). Текст, картинка, бинарник, архив и т.д.
* **d** — Каталог (directory).
* **l** — Символическая ссылка (symbolic link). Файл-указатель на другой файл.
* **c** — Символьное устройство (character device). Оперирует данными как потоком байтов (например, клавиатура, терминал).
* **b** — Блочное устройство (block device). Оперирует данными блоками (например, жёсткие диски, SSD, USB-флешки).
* **s** — Сокет (socket). Файл для сетевого и межпроцессного взаимодействия (IPC).
* **p** — Именованный канал (FIFO — First In, First Out). Также для IPC.

6. Что такое символические и жесткие ссылки в Linux, приведите примеры их использования.

**Жёсткая ссылка (hard link):**

* **Что это:** дополнительное имя для существующих данных на диске (inode). Все жёсткие ссылки равнозначны. Данные удаляются только когда удалена последняя жёсткая ссылка на них.
* **Ограничения:** нельзя ссылаться на каталоги и на файлы в других разделах/файловых системах.
* **Пример использования:** Создание "короткого" имени для длинного файла в той же файловой системе без копирования.

**Символическая ссылка (symbolic link, symlink):**

* **Что это:** файл-ярлык, который содержит путь к целевому файлу. Если целевой файл удалён, ссылка становится "битой" (dangling).
* **Преимущества:** может ссылаться на каталоги и файлы в других файловых системах.
* **Пример использования:**
  1. Организация программ: несколько версий ПО лежат в /opt/program-1.2/, а симлинк /opt/program всегда указывает на текущую версию.
  2. Создание ярлыка в домашнем каталоге для быстрого доступа к глубоко вложенной папке.

7. Какие права могут быть назначены файлу? Назовите способы конфигурирования прав доступа в ОССН.

* **Базовые права:** Чтение (r), запись (w), выполнение (x) для трех категорий: владелец, группа, остальные.
* **Особые биты:** SUID, SGID, Sticky Bit (см. выше).

**Способы конфигурирования в ОССН Astra Linux:**

1. **Командная строка (терминал):** Стандартные команды chmod, chown, chgrp.
2. **Графический файловый менеджер:** обычно в свойствах файла/папки есть вкладка "Права" или "Разрешения", где можно выставить права через выпадающие списки.
3. **Центр управления Astra Linux:** позволяет задавать **политики безопасности**, которые могут, среди прочего, регулировать правила назначения прав доступа для пользователей и групп в масштабах системы.

8. По отношению к каким категориям применяются права доступа к файлу?

**Базовые права:** Чтение (r), запись (w), выполнение (x) для трех категорий: владелец, группа, остальные.

**Особые биты:** SUID, SGID, Sticky Bit (см. выше).

**Способы конфигурирования в ОССН Astra Linux:**

1. **Командная строка (терминал):** стандартные команды chmod, chown, chgrp.
2. **Графический файловый менеджер:** обычно в свойствах файла/папки есть вкладка "Права" или "Разрешения", где можно выставить права через выпадающие списки.
3. **Центр управления Astra Linux:** позволяет задавать **политики безопасности**, которые могут, среди прочего, регулировать правила назначения прав доступа для пользователей и групп в масштабах системы.