

## **Лабораторная работа №15. Обеспечение безопасности путем ограничения доступа. Создание системы резервного копирования данных в операционной системе семейства GNU Linux. Конфигурирование, компиляция и установка собственного ядра Linux**

**Цель работы:** А) Изучить принципы и процесс создания пользователей, групп и разграничения прав доступа.

В) Изучить принципы и процесс создания системы резервного копирования данных с использованием rsync.

С) Изучить принципы и процесс конфигурации, компиляции и установки ядра Linux из исходных кодов.

### **Теоретические сведения**

А) su – смена пользователя

adduser – создание пользователя

addgroup – создание группы

usermod – редактирование пользователя

chown – смена владельца

chgrp – смена группы-владельца

chmod – изменение прав доступа

Абсолютная запись устанавливает права на прописанные (например, chmod 755)

Относительная запись дополняет текущие права файла или директории (например, chmod o+rx)

Права записываются в порядке доп.-владелец-группа-остальные и могут быть как в виде чисел, так и в виде букв. Доп = SUID,SGID, sticky bit.

Права	Применимо к файлам	Применимо к каталогам
Read	Прочитать содержимое файла	Отобразить список содержимого каталога
Write	Изменить содержимое файла	Разрешить создавать, удалять или задавать права на файлы
Execute	Запустить файл как программу	Переход в каталог

Права	Цифровое представление
Read (Чтение)	4
Write (Запись)	2
Execute (Выполнение)	1

Права	Числ. значение	Относит. режим	Применение к файлам	Применение к каталогам
SUID	4	u+s	Пользователь выполняет файл с разрешениями владельца файла.	Нет смысла применять
SGID	2	g+s	Пользователь выполняет файл с разрешениями владельца группы.	Файлы, созданные в каталоге, получают одного и того же владельца группы.
sticky bit	1	+t	Нет смысла применять	Запрещает пользователям удалять файлы от других пользователей.

Если необходима более сложная система разграничения прав доступа, то для этого подойдут списки контроля доступа ACL, если они поддерживаются системой.

`getfacl` – получение информации о текущих настройках ACL

`setfacl` – установка настроек ACL

Например, `setfacl -m g:mygrp:rwx /home/alex` даст полные права к директории пользователям группы `mygrp`.

**В) Rsync** является утилитой командной строки в Linux, предназначенной для синхронизации и копирования файлов и директорий локально или удаленно. Rsync обладает мощными возможностями сжатия, а также выполняет инкрементное копирование, что позволяет экономить пропускную способность сети и время при повторных запусках резервного копирования.

Синтаксис утилиты rsync опции source destination.

Пример опций:

- a – режим архивирования
- v – вывод информации о синхронизации
- progress – вывод процесса выполнения
- z – сжатие файлов перед передачей
- delete – удаление файлов, которых нет в источнике
- include – включить файлы по шаблону
- exclude – исключить файлы по шаблону

Пример путей

/home/kali

admin@192.168.0.30:/home/share

rsync://192.168.0.30:/home/share

Пример конфигурационного файла

```
pid file = /var/run/rsyncd.pid
lock file = /var/run/rsync.lock
log file = /var/log/rsync.log
[share]
path = /home/kali/syncdir
hosts allow = *
hosts deny = *
list = true
uid = root
gid = root
read only = false
```

Подробные сведения о синтаксисе и ключах утилиты можно получить командой `man rsync`.

С)

/usr/src/src – директория, в которой обычно хранятся исходные коды ПО  
Скопировать текущий конфигурационный файл можно в /boot, назвать его нужно .config. Для его актуализации достаточно выполнить `make oldconfig`

`make menuconfig` – удобный метод конфигурации ядра. При конфигурации:

\* - включает опцию в ядро

M – загружает опцию как модуль

[ ] – отключает опцию

Для компиляции ядра необходимо выполнить следующие команды, находясь в директории с исходными кодами:

```
make
```

```
make bzImage
```

```
make modules
```

```
make modules_install
```

Причем рекомендуется выполнять команду `make` с ключом `-jX`, где `X` = число ядер процессора, которое вы желаете использовать при компиляции. Процесс компиляции не быстрый, поэтому желательно выставить значение максимально возможное. Если возникают проблемы с выполнением команды `make`, установите запрашиваемые утилитой пакеты.

Для установки ядра и обновления загрузчика необходимо выполнить следующие действия, находясь в директории с исходными кодами:

```
cp arch/i386/boot/bzImage /boot/vmlinuz-ВерсияЯдра
```

```
cp System.map /boot/System.map-ВерсияЯдра
```

```
update-grub2
```

Альтернативный вариант: `make install`

Если при загрузке возникает ошибка, что отсутствует `initrd/initramfs`, воспользуйтесь, например, `mkinitramfs`, поместите созданный файл в загрузочную директорию, затем обновите загрузчик. Убедитесь, что в конфигурационном файле `grub(/boot/grub/grub.cfg)` указан путь к вашему `initrd/initramfs`. Если не указан, то укажите, предварительно сохранив бэкап конфигурационного файла.

## **Порядок выполнения работы**

- А) 1.** Создайте 3 пользователей: alex, brian, john.
- 2.** Создайте 2 группы: teacher, student
- 3.** Поместите alex в группу teacher, а brian в группу student

4. Сделайте так, чтобы домашние директории всех трех пользователей принадлежали группе `teacher`.

5. Также дайте всем пользователям возможность использовать команду `sudo`(для удобства).

6. Дайте возможность пользователям группы `teacher` переходить и читать содержимое директорий всех трех пользователей.

6.1 Для домашней директории `alex` сделайте это абсолютным путем.

6.2. Для домашней директории `brian` сделайте это относительным путем.

7. Для домашней директории `john` дайте следующие права: владелец может все, группа `teacher` читать и переходить, остальные читать и переходить.

8. Создайте в домашней директории `john` директорию `share` с такими правами, что владелец может все, группа-владелец `teacher` может все, остальные переходить и читать, при этом использовать `SGID` и `sticky bit`.

9. Создайте в директории `share` несколько файлов от лица `john` и от лица `alex`.

10. Попробуйте удалить файлы `alex` от лица `john` или наоборот.

11. В домашней директории `john` создайте скрипт, выводящий произвольное сообщение в терминал, а в комментарий поместите какую-нибудь информацию. Установите такие права, чтобы владелец мог все, группа-владелец `teacher` читать и выполнять, а остальные ничего не могли.

12. Проверьте, что `alex` может и прочесть, и выполнить скрипт, а `brian` не может ни того, ни другого.

13. Расскажите, что означают следующие права доступа:

13.1. `drwxrwSrwx`

13.2. `drwxr—r-t`

13.3. `-rwsr-s---`

14. Если в системе есть поддержка `ACL`, получите информацию о разрешениях файла скрипта.

15. При помощи ACL добавьте полные права доступа brian и группе student.

16. Убедитесь, что теперь brian обладает правами работать с файлом скрипта.

Бонус: выполните команду `sl`

**В) 1.** Установите утилиту rsync

2. Создайте директорию с произвольным названием(пример в работе syncdir). Если есть второе устройство в локальной сети, то можно производить синхронизацию с ним, достаточно до пути директории указать ssh или rsync адрес. Далее в работе будет локальная синхронизация.

3. Создайте конфигурационный файл и задайте необходимые параметры: директорию, разрешить доступ всем подключениям, пользователь, от лица которого будет выполняться синхронизация

4. Запустите сервер rsync и добавьте его в автозагрузку

5. Создайте в любой директории, кроме ранее созданной, файл, и при помощи rsync синхронизируйте его с директорией syncdir(в режиме архивирования, с отображением информации о процессе копирования и с использованием сжатия файлов). Проверьте, что синхронизация выполнена.

6. Создайте директорию, затем поместите (или создайте в ней) в нее несколько файлов.

7. Синхронизируйте эту директорию с syncdir(в режиме архивирования, с отображением информации о процессе копирования и с использованием сжатия файлов). Проверьте, что синхронизация выполнена.

8. Внесите изменения в директорию(например, измените содержимое файла или добавьте новые файлы) и снова синхронизируйте с теми же параметрами. Убедитесь, что передавались только «изменения».

9. Напишите скрипт по выполнению синхронизации любой директории с syncdir с логированием в журнал. Не забудьте дать файлу права на выполнение. Проверьте скрипт на работоспособность.

10. Сделайте так, чтобы скрипт выполнялся каждый день в 18:00 при помощи cron.

Бонус: выполните команду `oneko`

- С) 1. Загрузите исходные коды с сайта [kernel.org](http://kernel.org) (stable или longterm).
2. Распакуйте архив в директорию, предназначенную для исходных кодов ПО в Linux.
3. Скопируйте текущий конфигурационный файл в директорию с исходными кодами и произведите его «актуализацию».
4. Находясь в директории с исходными кодами, запустите процесс конфигурации.
5. Выполните конфигурацию так, чтобы:
- 5.1. Была возможность виртуализации KVM, XEN.
  - 5.2. Поддержка всех видов драйверов virtio
  - 5.3. Поддержка cgroups
  - 5.4. Поддержка iptables
  - 5.5. Поддержка initrd
  - 5.6. Поддержка загрузки и выгрузки модулей
  - 5.7. Поддержка SMP с максимальным количеством ядер равным 4
  - 5.8. Поддержка дополнительных файловых систем NFS, MSDOS
  - 5.9. Поддержка UTF-8 и всех кириллических кодировок.
6. Запустите процесс сборки ядра.
7. Установите ядро.
8. Выполните конфигурацию загрузчика для загрузки созданного ядра
9. Перезагрузите устройство с выбором созданного ядра и убедитесь, что загрузка прошла успешно

Бонус: выполните команду `smatrix`