Лабораторная работа №4.
Управление привилегиями и настройка безопасности в ОС “Альт”

Цель работы: Изучение методов управления привилегиями, особенностей ОС “Альт” с точки зрения безопасности, а также базовых аспектов настройки системы для повышения уровня безопасности.

1. Функционал control

Механизм *control* используется для переключения между неким набором фиксированных состояний для задач, допускающих такой набор.

Команда *control* доступна только для суперпользователя.

Запустив команду *control* без параметров можно увидеть полный список компонентов, управляемых командой вместе с их текущим состоянием и набором допустимых состояний.

```
# control
```

Описание вывода:

1-й столбец — компонент системы;

2-й столбец — текущее состояние;

3-й столбец — набор допустимых состояний;

Для того, чтобы посмотреть разрешения выполнения конкретным компонентов, надо запустить команду с ключом *help*.

Справка управления командой *su*:

```
# control su help
```

```
[root@alt ~]# control su help
public: Any user can execute /bin/su
wheel: Any user can execute /bin/su, but only "wheel" group members can switch to superuser
wheelonly: Only "wheel" group members can execute /bin/su
restricted: Only root can execute /bin/su
```

Команда *su* используется для смены пользователя в текущей сессии терминала.

Для управления командой *su* есть следующие политики:

public — любой пользователь может выполнить команду */bin/su*;

wheel — любой пользователь может выполнить команду *su*, но только пользователи, входящие в группу *wheel*, могут получить через нее права суперпользователя;

wheelonly — только пользователи, входящие в группу *wheel*, могут выполнить команду *su*;

restricted — только суперпользователь может выполнять команду *su*.

Теперь выведем текущую политику команды *su* и проверим права доступа к исполняемому файлу */bin/su*:

```
# control su
# ls -l /bin/su
```

```
[root@alt ~]# control su
wheelonly
[root@alt ~]# ls -l /bin/su
-rws--x--- 1 root wheel 31072 июл  3  2020 /bin/su
```

Создадим пользователя *testuser* с паролем *P@ssw0rd*:

useradd testuser

passwd testuser

Зайдем под учетной записью *testuser* и попробуем с помощью команды *su* войти в учетную запись *user*:

su - testuser

$ su - user

```
[root@alt ~]# su - testuser
[testuser@alt ~]$ su - user
-bash: /bin/su: Отказано в доступе
```

Поскольку пользователь *testuser* не входит в группу *wheel*, использовать команду *su* ему запрещено!

Изменим текущую политику команды *su* на *public* и проверим права доступа к исполняемому файлу */bin/su*:

control su public

ls -l /bin/su

```
[root@alt ~]# control su public
[root@alt ~]# ls -l /bin/su
-rws--x--x 1 root root 31072 июл  3  2020 /bin/su
```

Важно! Устанавливать разрешение *public* следует только в рамках учебной работы.

Еще раз зайдем под учетной записью *testuser* и попробуем с помощью команды *su* войти в учетную запись *user*:

su - testuser

$ su - user

```
[root@alt ~]# su - testuser
[testuser@alt ~]$ su - user
Password:
[user@alt ~]$
```

Для переключения состояния *control* вызывает соответствующий скрипт из каталога */etc/control.d/facilities/*:

ls /etc/control.d/facilities

Зайдем под учетной записью *testuser* и проверим работу команды *mount*:

su - testuser

$ mount

Команда *mount* предназначена для подключения файловых систем и переносных накопителей к конкретным точкам монтирования в дереве каталогов. При запуске без аргументов команда показывает все подключенные в данный момент файловые системы.

По умолчанию для команды *mount* установлена политика *public*:

control mount

```
[root@alt ~]# control mount
public
```

Изменим текущую политику команды *mount* на *restricted*:

control mount restricted

Еще раз зайдем под учетной записью *testuser* и проверим работу команды *mount*:

su - testuser

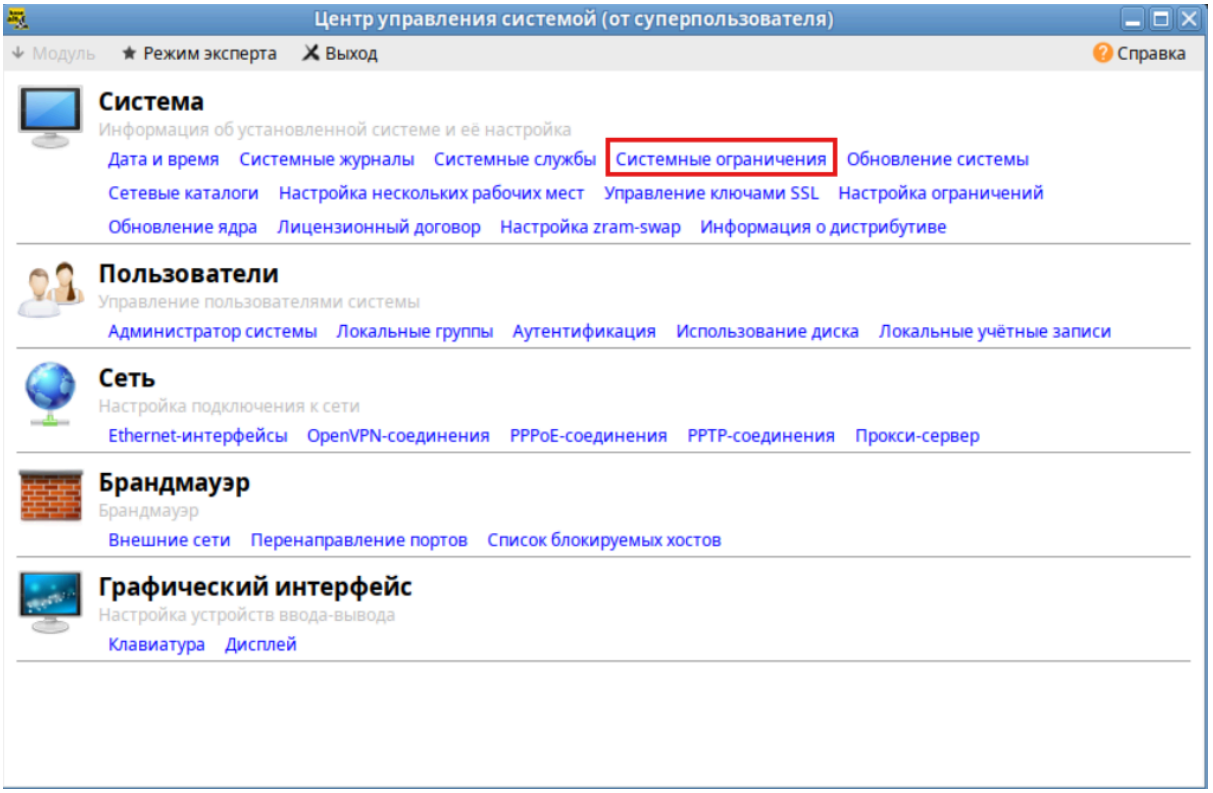
$ mount

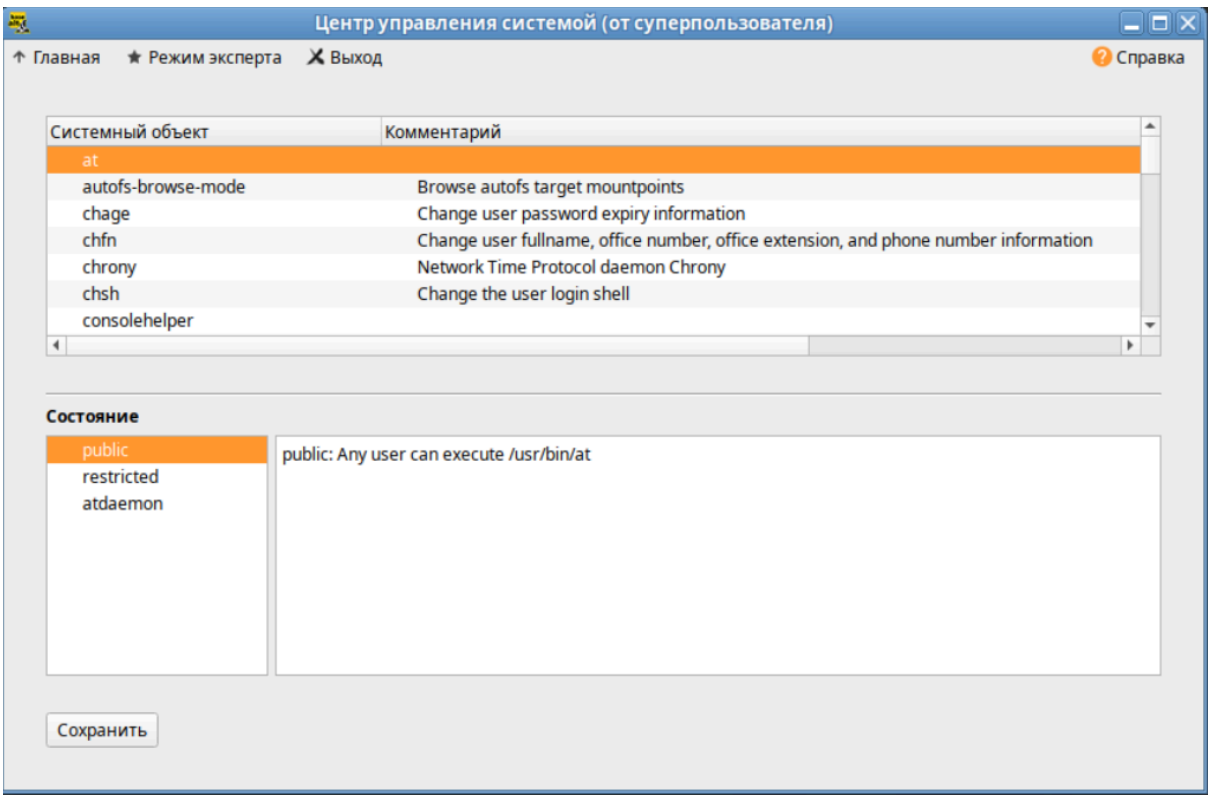
```
[root@alt ~]# su - testuser
[testuser@alt ~]$ mount
-bash: /bin/mount: Отказано в доступе
```

Управлять политиками *control* можно через ЦУС.

Для этого должен быть установлен пакет *alterator-control*.

Зайдем в ЦУС и в разделе “Система” откроем пункт “Системные ограничения”:





2. Концепция sudo

Команда sudo (сокращение от “superuser do”) используется для выполнения команд с привилегиями суперпользователя.

Перед выполнением команды *sudo* запрашивает пароль пользователя, а не пароль суперпользователя.

После выполнения *sudo* существует временной отрезок, в течение которого повторное выполнение команды *sudo* не требует пароль (удобно для взлома системы со стороны *rootkits* и хакерских атак).

С другой стороны, команда *sudo* удобна для распределения прав между несколькими администраторами системы, не предоставляя прав суперпользователя на все другие действия и не выдавая пользователю пароля суперпользователя.

В ОС “Альт Рабочая станция 10.2” команды *sudo* по умолчанию не установлен.

Предварительно установим пакет *sudo*:

```
# apt-get update
# apt-get install sudo
```

Команда *sudo* требует предварительной настройки, так как в */etc/sudoers* не описан ни один пользователь, включая суперпользователя.

```
# cat /etc/sudoers
```

В дополнение к */etc/sudoers* могут использоваться отдельные файлы из каталога */etc/sudoers.d/*.

```
# ls /etc/sudoers.d
```

Для ограничения прав на выполнение самой команды *sudo* используется механизм *control*.

Справка управления командой *sudo*:

```
# control sudo help
```

```
[root@alt ~]# control sudo help
public: Any user can execute /usr/bin/sudo
wheelonly: Only "wheel" group members can execute /usr/bin/sudo
restricted: Only root can execute /usr/bin/sudo
```

На текущий момент существуют следующие политики у команды sudo:

public — любой пользователь может получить доступ к команде */usr/bin/sudo*;

wheelonly — только пользователи из группы wheel имеют право получить доступ к команде */usr/bin/sudo*;

restricted — только суперпользователь имеет право выполнять команду */usr/bin/sudo*.

Теперь выведем текущую политику команды *su* и проверим права доступа к исполняемому файлу */bin/su*:

```
# control sudo
# ls -l /usr/bin/sudo
```

```
[root@alt ~]# control sudo
wheelonly
[root@alt ~]# ls -l /usr/bin/sudo
-rws--x--- 1 root wheel 1012528 ноя  8  2023 /usr/bin/sudo
```

Пользователь из группы *wheel* имеет право запускать саму команду *sudo*, но это не означает, что он через sudo может выполнить какую-то команду с правами суперпользователя.

Для разрешения получения прав на выполнение конкретных команд с правами суперпользователя надо отредактировать настройки правил */etc/sudoers* при помощи специальной команды *visudo* (которая не портит права на файлы):

```
# visudo
```

Если раскомментировать (убрать # в начале строки) в */etc/sudoers* следующую строку, то это даст права выполнять через *sudo* любую команду с любого хоста (например, через ssh), пользователям, входящим в группу *wheel*, запрашивая их пароль:

```
WHEEL_USERS ALL=(ALL) ALL
```

С точки зрения безопасности правильнее давать права на выполнение *sudo* не всей группе *wheel*, а конкретному пользователю, и не на все команды, а на те, которые ему необходимы для быстрого получения права суперпользователя.

Дадим пользователю *testuser* права на использование команды */usr/bin/apt-get*:

```
testuser ALL=(ALL) /usr/bin/apt-get
```

Также не забудем добавить пользователя *testuser* в группу *wheel*:

```
# usermod -aG wheel testuser
```

Зайдем под учетной записью и попробуем обновить систему:

```
# su - testuser
$ sudo apt-get update
```

```
[root@alt ~]# su - testuser
[testuser@alt ~]$ sudo apt-get update

Мы полагаем, что ваш системный администратор изложил вам основы
безопасности. Как правило, всё сводится к трём следующим правилам:

    №1) Уважайте частную жизнь других.
    №2) Думайте, прежде чем что-то вводить.
    №3) С большой властью приходит большая ответственность.

По соображениям безопасности пароль, который вы введёте, не будет виден.

[sudo] password for testuser:
Получено: 1 http://ftp.altlinux.org p10/branch/x86_64 release [4223B]
Получено: 2 http://ftp.altlinux.org p10/branch/x86_64-i586 release [1665B]
Получено: 3 http://ftp.altlinux.org p10/branch/noarch release [2844B]
Получено 8732B за 5s (1714B/s).
Найдено http://ftp.altlinux.org p10/branch/x86_64/classic pkglist
Найдено http://ftp.altlinux.org p10/branch/x86_64/classic release
Найдено http://ftp.altlinux.org p10/branch/x86_64-i586/classic pkglist
Найдено http://ftp.altlinux.org p10/branch/x86_64-i586/classic release
Найдено http://ftp.altlinux.org p10/branch/noarch/classic pkglist
Найдено http://ftp.altlinux.org p10/branch/noarch/classic release
Чтение списков пакетов... Завершено
Построение дерева зависимостей... Завершено
```

Для просмотра и анализа логов *sudo* можно использовать команду *journalctl*.

Найдем в журнале информацию о выполнении предыдущей команды.

Отфильтруем логи *sudo* и сохраним в файл *sudo.log*:

```
# journalctl _EXE=/usr/bin/sudo > sudo.log
# cat sudo.log
```

```
[root@alt ~]# journalctl _EXE=/usr/bin/sudo > sudo.log
[root@alt ~]# cat sudo.log
                alt sudo[8268]: UNSPECIFIED (__progname="sudo" uid=501 euid=0): pam_tcb(sudo:auth):
 Authentication passed for testuser from user(uid=501)
                alt sudo[8268]: testuser : HOST=alt ; TTY=pts/0 ; PWD=/home/testuser ; USER=root ;
COMMAND=/usr/bin/apt-get update
                alt sudo[8268]: pam_tcb(sudo:session): Session opened for root by user(uid=501)
```

3. Схема Tcb

Trusted Computing Base (tcb) — механизм управления теневыми паролями, выступающего в роли альтернативы традиционной схемы */etc/shadow*.

Ключевым отличием *tcb* от */etc/shadow* является уход от использования общего файла со всеми хэшами паролей в пользу разнесения хэшей паролей по отдельным каталогам и файлам. При подобной организации хранения операции с паролями можно выполнять без повышения прав, а процесс, осуществляющий обработку учетных данных, ограничен учетной записью отдельного пользователя.

Обработчик */etc/shadow* всегда получает доступ сразу ко всем хэшам паролей, т.е. уязвимость в утилите *passwd* позволяет изменить любой пароль. В *tcb* каждый файл включает только хэш одного пользователя и размещается в

каталоге, принадлежащем этому пользователю, что позволяет обойтись без повышения привилегий при запуске утилиты *passwd*.

В ОС “Альт Рабочая станция” все теневые файлы пользователей располагаются в каталоге */etc/tcb*.

ls /etc/tcb

Для совместимости с другими схемами входное имя может содержать только латинские буквы, цифры и символ подчеркивания.

Переключение между схемой хранения паролей *tcb*, классической схемой (с единым файлом */etc/shadow*) и строгой схемой (классическая, при которой команду *passwd* имеет право запускать только суперпользователь) управляется командой *control passwd* с параметрами *tcb*, *traditional* и *restricted* соответственно.

Справка управления командой *passwd*:

control passwd help

```
[root@alt ~]# control passwd help
tcb: Any user can change his own password using /usr/bin/passwd when tcb scheme is enabled
traditional: Any user can change his own password using /usr/bin/passwd when tcb scheme is disabled
restricted: Only root may change users passwords using /usr/bin/passwd
```

По умолчанию в ОС “Альт Рабочая Станция” установлена схема хранения паролей *tcb*:

control passwd

```
[root@alt ~]# control passwd
tcb
```

4. Основы Linux Login

Аутентификация пользователя выполняется с помощью файла теневого пароля. Файл теневого пароля настраивается с помощью конфигурационного файла */etc/login.defs*.

Команды *useradd*, *usermod*, *userdel* и *groupadd*, а также другие утилиты для пользователей и групп берут значения по умолчанию из этого файла. Каждая строка состоит из имени директивы и связанного с ней значения.

cat /etc/login.defs

Приведем список основных директив */etc/login.defs*:

MAIL_DIR — расположение почтовых ящиков пользователей;

UID_MIN, UID_MAX — минимальное и максимальное значения для автоматического выбора UID (от 1000 до 60000);

GID_MIN, GID_MAX — минимальное и максимальное значения для автоматического выбора GID (от 1000 до 60000);

CREATE_HOME — создание домашних директорий при добавлении нового пользователя;

UMASK — пользовательская маска *umask*.

Изменим директиву PASS_MAX_DAYS, которая отвечает за максимальное количество дней, в течение которых пароль может оставаться действительным. Для этого отредактируем следующую строку в файле */etc/login.defs* (также необходимо раскомментировать строку):

PASS_MAX_DAYS 90

Теперь создадим пользователя *newuser* с паролем *P@ssw0rd* и проверим информацию о его пароле:

useradd newuser
passwd newuser
chage -l newuser

```
[root@alt ~]# chage -l newuser
Последний раз пароль был изменён                        : сен 30, 2024
Срок действия пароля истекает                           : дек 29, 2024
Пароль будет деактивирован через                        : никогда
Срок действия учётной записи истекает                   : никогда
Минимальное количество дней между сменой пароля         : -1
Максимальное количество дней между сменой пароля        : 90
Количество дней с предупреждением перед деактивацией пароля     : -1
```

5. Управление паролями

Команда *chage* используется для управления сроком действия паролей пользователей.

Синтаксис команды *chage*:

chage [options] [username]

Опция -l отображает информацию о пароле указанного пользователя.

Выведем информацию о пароле пользователя *testuser*:

chage -l testuser

```
[root@alt ~]# chage -l testuser
Последний раз пароль был изменён                        : сен 29, 2024
Срок действия пароля истекает                           : никогда
Пароль будет деактивирован через                        : никогда
Срок действия учётной записи истекает                   : никогда
Минимальное количество дней между сменой пароля         : -1
Максимальное количество дней между сменой пароля        : -1
Количество дней с предупреждением перед деактивацией пароля     : -1
```

Справка команды *chage*:

chage --help

Установим для пароля пользователя *testuser* следующие параметры:

Минимальный срок действия пароля — 7 дней;

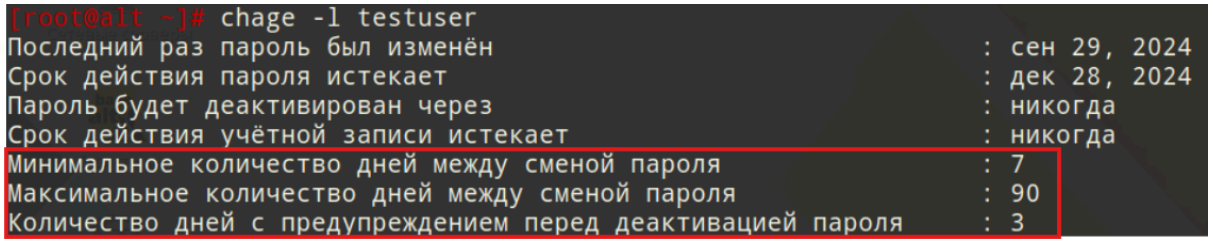
Максимальный срок действия пароля — 90 дней;

Предупреждение до истечения срока действия пароля — 3 дня.

chage -m 7 -M 90 -W 3 testuser

Теперь еще раз выведем информацию о пароле пользователя *testuser*:

```
# chage -l testuser
```



Также установим дату истечения срока действия учетной записи *testuser*:

```
# chage -E ‘2025-01-01’ testuser
```

Проверим применение настроек:

```
# chage -l testuser
```