

Министерство образования и науки РФ
ФГБОУ ВО «Томский государственный университет
систем управления и радиоэлектроники»
Кафедра комплексной информационной безопасности
электронно-вычислительных систем (КИБЭВС)

А.А. Конев, А.Ю. Якимук

БЕЗОПАСНОСТЬ ОПЕРАЦИОННЫХ СИСТЕМ

Практикум для самостоятельной работы

для студентов специальностей и направлений

10.03.01 – «Информационная безопасность»,

10.05.02 – «Информационная безопасность
телекоммуникационных систем»,

10.05.03 – «Информационная безопасность
автоматизированных систем»,

10.05.04 – «Информационно-аналитические системы безопасности»

В-Спектр
Томск, 2017

Требования к аппаратному и программному обеспечению

- 1) ПК со следующими минимальными системными требованиями:
 - процессор семейства Pentium с тактовой частотой от 1GHz или аналогичный ему AMD, желательно с поддержкой аппаратной виртуализации (VT-x/AMD-V);
 - ОЗУ от 1Гб;
 - видеоадаптер SVGA (1024x768) или выше;
 - свободное место на HDD 100гб;
 - устройства взаимодействия с пользователем: клавиатура, мышь;
 - опционально оптический накопитель.
- 2) Программный продукт для виртуализации VirtualBox (www.virtualbox.org).
 - Вне зависимости от того, какая ОС выступает в качестве хост системы, необходимо установить VirtualBox Extension Pack (www.virtualbox.org/wiki/Downloads). Инструкция по установке (<https://www.virtualbox.org/manual/ch08.html#vboxmanage-extpack>).
- 3) Дистрибутив Debian GNU/Linux.
 - «Чистый» дистрибутив можно скачать с FTP сервера Яндекса:
`ftp://mirror.yandex.ru/debian-cd/current/i386/iso-cd/debian-ВЕРСИЯ ВЫПУСКА-i386-xfce+lxde-CD-1.iso`
 - Скачать с <ftp://mirror.yandex.ru/debian-cd/current/i386/iso-cd/> 2 файла SHA1SUMS и MD5SUMS
 - Проверить образ на целостность нужно с помощью:
\$ sha1sum -c SHA1SUMS
\$ md5sum -c MD5SUMS

Debvm – основная виртуальная машина (название в VirtualBox Менеджер).

В каталоге с виртуальной машиной необходимо поместить файл «Пароли.txt» со следующим содержанием:

```
user – 12345  
root - qwerty
```

Лабораторная работа №2. Аутентификация в прикладных приложениях при помощи физического объекта – наличие персонального eToken.

Лабораторная работа №4. Аудит – наличие персонального eToken.

Установка базовой виртуальной операционной системы

При установке виртуальной операционной системы необходимо следовать следующим указаниям:

- 1) Создайте в VirtualBox новую виртуальную машину.
 - Тип ОС: Linux.

- Версия: Debian.
 - Размер основной памяти 1024 Мб.
 - Создать новый жесткий диск ⇒ VDI (VirtualBox Disk Image). Этот тип лучше подойдет для совместимости между версиями VirtualBox ⇒ Динамически расширяющийся образ 20 Гб или больше, если позволяют объемы жесткого диска.
- 2) Далее в свойствах созданной виртуальной машины установите следующее:
- Носители. Подключите скаченный ранее образ к IDE контроллеру USB. Установите галочку напротив «Включить контроллер USB 2.0» и добавьте пустой фильтр (первый значок справа).
- 3) Установка Debian GNU/Linux.
- При установке следует применить некоторые параметры.
- Имя пользователя: user ; Пароль:12345.
 - Суперпользователь: root; Пароль:qwerty.
 - Разметка диска:

Раздел диска	ФС	Точка монтирования
/dev/sda1 (Primary)	Ext4	/boot
/dev/sda2 (Primary)	Swap	Раздел подкачки
/dev/sda3 (Primary)	Ext4	/
/dev/sda5 (Logical)	Ext4	/home

- Сеть. Если есть особые настройки в сети, то, дойдя до соответствующего пункта, указать.
 - Сервера обновлений лучше выбирать или Яндекс или ftp.ru.debian.org/debian/ .
- 4) До установки каких-либо дополнительных программ, создайте резервную копию системы (о восстановлении из резервной копии см. п. восстановление системы). Выполните в терминале:

```
sudo tar cvpzf backup.tgz --exclude=/proc
--exclude=/lost+found\
--exclude=/backup.tgz
--exclude=/mnt --exclude=/sys /
```

Восстановление системы

Рекомендуется использовать LiveCD любого дистрибутива Linux.

- 1) Загрузитесь с LiveCD. После загрузки перейдите в терминал.
- 2) Подключите USB-диск или другое устройство, где хранится файл backup.tgz.
- 3) Проверьте контрольные суммы файла backup.tgz. В каталоге также

должен находиться SHA1SUM файл. Запустите проверку:

```
$ sha1sum -c backup.tgz
```

Эталонный Хеш (5b19b049590acfe0ce18fec63fb74e6c3178e07e)

4) На жестком диске, выбранном для восстановления системы, выполните команду, которая сотрет все содержимое и заменит на то, что в архиве:

```
# tar -cxvzpf backup.tgz -C /
```

5) Если не изменялась схема разбиения дисков, то потребуются скорректировать файлы /etc/fstab в соответствии с новой схемой.

6) Перезагрузитесь.

Необязательно восстанавливать всю систему. Из данного архива можно только взять ядро или какие-то другие эталонные конфигурационные файлы.

Подготовка виртуальной операционной системы к лабораторным работам

Создайте отдельную группу, для которой будет разрешен доступ к работе с утилитой sudo. Выполните:

1) \$ su

2) введите пароль root

3) # groupadd -g 999 done

4) # usermod -G done user

5) # nano /etc/sudoers

6) добавьте после строки **%sudo ALL=(ALL) ALL** строку

```
%done ALL=(ALL) ALL
```

7) Ctrl+O, Enter, Ctrl+X.

8) # exit

Установка eToken PKI Client

1) Скачайте PKI Client <http://www.aladdin-rd.ru/support/download/464/>

2) Распакуйте архив, прочитайте инструкцию

eToken_PKI_Client_Версия_Linux_Admin_Guide_Rev_B.pdf

3) Перед установкой клиента необходимо установить PSCS-Lite Service.

Скачивать клиент лучше через официальный сайт
<http://packages.debian.org/search?suite=stable§ion=all&arch=any&searchon=sourcenames&keywords=pcsc-lite>.

4) Скачайте архив вида pcsc-lite_НОМЕР_ВЕРСИИ.orig.tar.gz.

5) Выполните ./configure с ключами:

```
sudo ./configure --enable-libusb --disable-libudev --disable-libhal
```

6) Выполните make && make install.

7) Далее установите pcscd, sudo apt-get install pcscd.

8) Установите sudo apt-get libqt4-core && libqt4-gui (конкретные версии пакетов можно узнать из официальной инструкции).

9) В каталоге, куда распакован архив eToken, перейдите в:

./Ubuntu/Deb installation/pkiclient-НОМЕР_ВЕРСИИ_i386.deb

- 10) Выполните `sudo dpkg -i pkiclient-НОМЕР_ВЕРСИИ_i386.deb`
- 11) Если в ошибках требуется `libccid`, то `sudo apt-get libccid` и перейдите к шагу 10.
- 12) В разделе меню «Прочее» запустите Start eToken PKI Client, затем eToken Properties.
- 13) Если нет ошибок, то клиент запущен, и можно работать.

Лабораторная работа №3. Дискреционное разграничение доступа

- 1) Создайте второго пользователя `user1`:
`$ sudo useradd -m -g done -s /bin/bash user1`
- 2) Задайте пароль 54321:
`$ sudo passwd user1`
- 3) Требуется наличие виртуальной машины Linux(DAC).

Лабораторная работа №4. Аудит

- 1) Установите демон аудита `auditd`:
`$ sudo apt-get install auditd`
- 2) Перезагрузитесь.
- 3) Выполните в терминале:
`$ /etc/init.d/auditd status`

Ответ должен быть примерно таким:

```
user@work:/boot$ /etc/init.d/auditd status
auditd is running..
```

Рисунок 1 – статус демона `auditd`

Если ответ другой, то попробуйте вручную запустить демон:

```
$ sudo /etc/init.d/auditd start
```

Иначе выполните:

```
$ sudo /etc/init.d/auditd restart
```

Обновления

Систему необходимо поддерживать в актуальном состоянии. Для этого периодически необходимо выполнять обновления:

- 1) `$ sudo apt-get update` – обновить список пакетов
- 2) `$ sudo apt-get upgrade` – установить новые пакеты

Следите за новыми выпусками Debian GNU/Linux на сайте www.debian.org/index.ru.html. Если вышла новая версия Debian, то рекомендуется обновиться:

- 1) `$ sudo apt-get dist-upgrade`

ПРИМЕЧАНИЕ!!! У проекта `debian` есть 3 ветки:

- 1) `Stable` – только проверенный и оттестированный софт
- 2) `Testing` – экспериментальная ветка, которая спустя некоторое время

станет stable. Содержит более свежий софт и не всегда стабильный.

3) Sid – самые свежие программы попадают сюда. Команда Debian не выпускает для данной ветки обновлений безопасности.

Настоятельно рекомендуется работать только со стабильной веткой!

Лабораторная работа №1. Учетные записи пользователей

Целью данной лабораторной работы является изучение основ для управления учетными записями пользователей.

“Правильное управление учетными записями является залогом безопасности системы. Редко используемые учетные записи становятся главными атаками злоумышленников, как и те записи, пароли к которым легко подобрать. Даже если для подключения учетных записей используются автоматизированные утилиты, важно понимать, какие при этом происходят изменения в системе”

Немет Э., Снайдер Г., Хейн Т. «Руководство администратора Linux, 2-е издание»

I. Основные файлы управления учетными записями

1) Система учета пользователей опирается на следующие конфигурационные файлы:

/etc/passwd – информация о пользователе

/etc/shadow – скрытая информация о пользователе

/etc/group – информация о группах

/etc/gshadow – скрытая информация о группах

/etc/default/useradd – свойства, назначаемые по умолчанию новым учетным записям

/etc/login.defs – настройки безопасности пароля

/etc/skel – каталог, содержащий личные файлы настроек по умолчанию

Все учетные записи хранятся в файле /etc/passwd. Выведите содержание этого файла на экран, выполнив:

```
$ cat /etc/passwd
```

(ПРИМЕЧАНИЕ!!! Приглашение \$ - это работа от учетной записи пользователя

- работа от учетной записи суперпользователя)

```

user@work:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
Debian-exim:x:101:103::/var/spool/exim4:/bin/false
statd:x:102:65534::/var/lib/nfs:/bin/false
messagebus:x:103:106::/var/run/dbus:/bin/false
avahi:x:104:107:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
gdm:x:105:113:Gnome Display Manager:/var/lib/gdm:/bin/false
haldaemon:x:106:114:Hardware abstraction layer,,,:/var/run/hald:/bin/false
usbmux:x:107:46:usbmux daemon,,,:/home/usbmux:/bin/false
saned:x:108:117::/home/saned:/bin/false
hplip:x:109:7:HPLIP system user,,,:/var/run/hplip:/bin/false
user:x:1000:1000:user,,,:/home/user:/bin/bash
user@work:~$

```

Рисунок 1 – результат вывода команды cat

Из рисунка 1 видны существующие учетные записи, созданные в системе. Общий формат записи в файле passwd:

username : password: UID : GID : GECOS : home_dir : shell

username – регистрационное имя (не более 32 символов).

password - пароль, который указывается при регистрации. В открытом виде не хранится, так как используется теневой пароль и на месте пароля стоит знак “x”.

UID – идентификатор пользователя (32-х битное целое число). Назначается ОС по умолчанию, обычно для нового созданного пользователя имеет значение 1000 и далее по нарастающей.

GID – идентификатор группы (32-х битное целое число). Совпадает с идентификатором группы, в которую входит пользователь, из файла /etc/group.

GECOS – используется для хранения более подробной информации о пользователе (дата рождения, телефонный номер, e-mail и т.д.)

home_dir – домашний каталог пользователя.

shell – командный интерпретатор, например /bin/bash или /bin/zsh.

Выполните:

```
$ cat /etc/passwd | grep user
```

(| – это конвейер, он принимает **STDOUT** от предыдущей команды **cat** и фильтрует его на наличие записи «user»)

Попытайтесь объяснить полученный результат команды.

2) Файл /etc/shadow

Данный файл содержит хеши паролей и другие настройки, связанные с учетными записями. Поэтому важно, чтобы данный файл, а тем более хеши, хранящиеся в нем, не попали под действие программ типа John The Ripper.

Выведите содержимое файла /etc/shadow:

```
$ cat /etc/shadow
```

```
user@work:~$ cat /etc/shadow
cat: /etc/shadow: Отказано в доступе
user@work:~$ █
```

Рисунок 2 – результат вывода команды cat

Данный файл доступен для чтения только суперпользователю, поэтому выполните:

```
$ sudo cat /etc/shadow
```

```
user@work:~$ sudo cat /etc/shadow
[sudo] password for user:
root:$6$WHkiFAqD$NBIKntsnpKZdEiT8Nq04s7FxD3P0$FMRm89ctEMLCe5G4uPQ10beIg5dwQ4z7Vnl9I6Nfc
7naSrzsEnT8vpRE1:15412:0:99999:7:::
daemon*:15412:0:99999:7:::
bin*:15412:0:99999:7:::
sys*:15412:0:99999:7:::
sync*:15412:0:99999:7:::
games*:15412:0:99999:7:::
man*:15412:0:99999:7:::
lp*:15412:0:99999:7:::
mail*:15412:0:99999:7:::
news*:15412:0:99999:7:::
uucp*:15412:0:99999:7:::
proxy*:15412:0:99999:7:::
www-data*:15412:0:99999:7:::
backup*:15412:0:99999:7:::
list*:15412:0:99999:7:::
irc*:15412:0:99999:7:::
gnats*:15412:0:99999:7:::
nobody*:15412:0:99999:7:::
libuuid!:15412:0:99999:7:::
Debian-exim!:15412:0:99999:7:::
statd*:15412:0:99999:7:::
messagebus*:15412:0:99999:7:::
avahi*:15412:0:99999:7:::
gdm*:15412:0:99999:7:::
haldaemon*:15412:0:99999:7:::
usbmux*:15412:0:99999:7:::
saned*:15412:0:99999:7:::
hplip*:15412:0:99999:7:::
user:$6$bYfKRGBX$SwZ2tQqzpwPv0e2txWGX6Lc6VqHxJ.3S1MAMfdKlc.S90LKHIAGauCTYmZ6J8FgkEo3mR
ArMKtouNp0YU4WQ/:15412:0:99999:7:::
user@work:~$ █
```

Рисунок 3 – результат вывода команды cat

Формат этого файла:

```
1 : 2 : 3 : 4 : 5 : 6 : 7 : 8 : 9
```

- 1) Регистрационное имя пользователя. Оно берется из файла /etc/passwd.
- 2) Хеш пароля.
- 3) Дата последнего изменения пароля.
- 4) Минимальное число дней между изменениями пароля.
- 5) Максимальное число дней между изменениями паролей.
- 6) Число дней, оставшихся до истечения срока действия пароля.
- 7) Количество дней по истечении срока действия пароля.
- 8) Срок действия учетной записи.
- 9) Резервное поле.

ПРИМЕЧАНИЕ!!! Файлы /etc/passwd и /etc/shadow не синхронизируются автоматически при «ручном» редактировании.

3) Файл /etc/group

Выполните в терминале:

```
$ cat /etc/group
user@work:~$ cat /etc/group
root:x:0:
daemon:x:1:
bin:x:2:
sys:x:3:
adm:x:4:
tty:x:5:
disk:x:6:
lp:x:7:
mail:x:8:
news:x:9:
uucp:x:10:
man:x:12:
proxy:x:13:
kmem:x:15:
dialout:x:20:
fax:x:21:
voice:x:22:
cdrom:x:24:user
floppy:x:25:user
tape:x:26:
sudo:x:27:
audio:x:29:user
dip:x:30:user
www-data:x:33:
backup:x:34:
operator:x:37:
list:x:38:
irc:x:39:
src:x:40:
gnats:x:41:
shadow:x:42:
utmp:x:43:
video:x:44:user
sasl:x:45:
plugdev:x:46:user
staff:x:50:
games:x:60:qwe
users:x:100:
nogroup:x:65534:
libuuid:x:101:
crontab:x:102:
Debian-exim:x:103:
mlocate:x:104:
```

Рисунок 4 – результат вывода команды cat

Формат файла:

group_name : password : GID : Members of group

group_name – имя группы.

password – хеш пароля. Вместо реального хеша может стоять знак “x” говорящий о том, что пароли групп хранятся в файле /etc/gshadow.

GID – уникальный идентификатор группы.

Members of group – список членов группы через запятую.

II. Создание пользователя

4) Для создания нового пользователя существует команда useradd.

Формат команды:

useradd параметры имя

Параметры могут быть следующими (рассмотрим только основные, более подробно `man useradd`):

- d каталог – домашний каталог пользователя
- g группа – основная группа, к которой может принадлежать пользователь
- e дата – дата отключения учетной записи
- f число – количество дней до отключения учетной записи навсегда.
- G Группа [,..] – дополнительные группы, в которых будет пользователь

5) Создайте нового пользователя в системе:

```
$ sudo useradd -m -g user -G games,audio -s /bin/bash kibevs
```

```
user@work:~$ sudo useradd -m -g users -G games,audio -s /bin/bash kibevs  
[sudo] password for user:  
user@work:~$ █
```

Рисунок 5 – создание нового пользователя

Проанализируйте введенную выше команду:

-m – создает домашнюю директорию пользователя в каталоге `/home/<имя пользователя>`

-g – данный ключ указывает основную группу, куда будет входить пользователь (в данном случае это группа `user`)

-G – с помощью этого ключа созданный пользователь стал также членом групп `games` и `audio`.

-s – командный интерпретатор, который будет доступен пользователю.

`kibevs` – имя созданного пользователя.

ПРИМЕЧАНИЕ!!! Вполне логичен вопрос, откуда узнать, где, какие группы и для чего нужны, в какие группы включать пользователя, а в какие – нет. Группа, задаваемая в команде `useradd` с ключом `-g` (основная группа) должна отражать категорию (например, `user`, `students` и т.д.). Однако не всегда такое полезно. Например, явное название группы может отражать привилегии ее участников в группе. Если есть возможность, то лучше для своего пользователя создать новую группу. Членом каких групп будет являться ваш пользователь зависит от того, к чему вам нужен доступ. Группа `cdrom` позволяет работать с оптическими дисками, `mail` работать с МТА и т.д. В документации по вашему дистрибутиву должно быть расписано какие группы и для чего нужны.

6) Пользователь создан, проверьте, так ли это. Выполните:

```
$ cat /etc/passwd
```

```

user@work:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
Debian-exim:x:101:103::/var/spool/exim4:/bin/false
statd:x:102:65534::/var/lib/nfs:/bin/false
messagebus:x:103:106::/var/run/dbus:/bin/false
avahi:x:104:107:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
gdm:x:105:113:Gnome Display Manager:/var/lib/gdm:/bin/false
haldaemon:x:106:114:Hardware abstraction layer,,,:/var/run/hald:/bin/false
usbmux:x:107:46:usbmux daemon,,,:/home/usbmux:/bin/false
saned:x:108:117::/home/saned:/bin/false
hplip:x:109:7:HPLIP system user,,,:/var/run/hplip:/bin/false
user:x:1000:1000:user,,,:/home/user:/bin/bash
qwe:x:1001:100::/home/qwe:/bin/bash
kibevs:x:1002:100::/home/kibevs:/bin/bash
user@work:~$

```

Рисунок 6 – результат работы команды cat

На рисунке 6 в предпоследней строчке виден созданный пользователь.

7) Проверьте состояние файла /etc/shadow:

\$ sudo cat /etc/shadow

```

user@work:~$ sudo cat /etc/shadow
root:$6$WHkiFAqD$NBiKntsnpKZdEiT8Nq04s7FxD3P0$FMRm89ctEMLCe5G4uPQ10beIg5dwQ4z7VnL9I6Nfc
7naSrzsEnT8vpRE1:15412:0:99999:7:::
daemon*:15412:0:99999:7:::
bin*:15412:0:99999:7:::
sys*:15412:0:99999:7:::
sync*:15412:0:99999:7:::
games*:15412:0:99999:7:::
man*:15412:0:99999:7:::
lp*:15412:0:99999:7:::
mail*:15412:0:99999:7:::
news*:15412:0:99999:7:::
uucp*:15412:0:99999:7:::
proxy*:15412:0:99999:7:::
www-data*:15412:0:99999:7:::
backup*:15412:0:99999:7:::
list*:15412:0:99999:7:::
irc*:15412:0:99999:7:::
gnats*:15412:0:99999:7:::
nobody*:15412:0:99999:7:::
libuuid!:15412:0:99999:7:::
Debian-exim!:15412:0:99999:7:::
statd*:15412:0:99999:7:::
messagebus*:15412:0:99999:7:::
avahi*:15412:0:99999:7:::
gdm*:15412:0:99999:7:::
haldaemon*:15412:0:99999:7:::
usbmux*:15412:0:99999:7:::
saned*:15412:0:99999:7:::
hplip*:15412:0:99999:7:::
user:$6$bYFKRGBX$SwZ2tQqzpwepV0e2txWG6Lc6VqHxJ.3S1MAMfdKlC.S90LkHIAGauCTYmZ6J8FgkEo3mR
ArMKtounP0YU4WQ/:15412:0:99999:7:::
qwe!:15433:0:99999:7:::
kibevs!:15433:0:99999:7:::
user@work:~$

```

Рисунок 7 – результат работы команды cat

Предпоследняя строчка говорит о том, что пользователь создан и существует. Однако обратите внимание на знак ! (фиктивный пользователь). Он находится в поле пароля, а там должен находиться хеш.

8) Новый пользователь не может остаться без пароля. Выполните:

```
$ sudo passwd kibevs
```

```
user@work:~$ sudo passwd kibevs
Введите новый пароль UNIX:
Повторите ввод нового пароля UNIX:
passwd: пароль успешно обновлён
user@work:~$ █
```

Рисунок 8 – задание нового пароля

У созданного пользователя есть свой домашний каталог (ключ `-m` позволяет создать его). Выполните:

```
$ ls /home
```

```
user@work:~$ ls /home
kibevs  lost+found  user
user@work:~$ █
```

Рисунок 9 – результат работы команды `ls`

Также созданный пользователь входит в группу `user` как основную и в `games, audio` как в дополнительные. Выполните:

```
$ grep kibevs /etc/group
```

```
user@work:~$ grep kibevs /etc/group
audio:x:29:user,kibevs
games:x:60:kibevs
user:x:1000:kibevs
user@work:~$ █
```

Рисунок 10 – результат работы команды `grep`

Из рисунка 10 видно, что помимо пользователя `kibevs` в группу `audio` входит еще пользователь `user`.

III. Редактирование учетной записи

Возникают ситуации, когда необходимо изменить какие-либо параметры учетной записи. Возможны два варианта:

9) Вручную путем редактирования `/etc/passwd` и `/etc/shadow`

Итак, рассмотрим ситуацию, когда пользователю нужно сменить командную оболочку (см. п.1.1 и п.2) с `bash` на другую.

ПРИМЕЧАНИЕ!!! Список доступных в системе интерпретаторов можно посмотреть так:

```
$ cat /etc/shells
```

```

user@work:~$ cat /etc/shells
# /etc/shells: valid login shells
/bin/csh
/bin/sh
/usr/bin/es
/usr/bin/ksh
/bin/ksh
/usr/bin/rc
/usr/bin/tcsh
/bin/tcsh
/usr/bin/esh
/bin/dash
/bin/bash
/bin/rbash
user@work:~$ █

```

Рисунок 11 – список интерпретаторов в Debian

Откройте файл /etc/passwd в текстовом редакторе (например, GNU Nano):

```
$ sudo nano /etc/passwd
```

В последней строчке присутствует пользователь kibevs.

```

GNU nano 2.2.4                               Файл: /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
Debian-exim:x:101:103::/var/spool/exim4:/bin/false
statd:x:102:65534::/var/lib/nfs:/bin/false
messagebus:x:103:106::/var/run/dbus:/bin/false
avahi:x:104:107:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
gdm:x:105:113:Gnome Display Manager:/var/lib/gdm:/bin/false
haldaemon:x:106:114:Hardware abstraction layer,,,:/var/run/hald:/bin/false
usbmux:x:107:46:usbmux daemon,,,:/home/usbmux:/bin/false
saned:x:108:117:./home/saned:/bin/false
hplip:x:109:7:HPLIP system user,,,:/var/run/hplip:/bin/false
user:x:1000:1000:user,,,:/home/user:/bin/bash
kibevs:x:1002:100:./home/kibevs:/bin/bash

```

^{^G} Помощь ^{^O} Записать ^{^R} ЧитФайл ^{^Y} ПредСтр ^{^K} Вырезать ^{^C} ТекПозиц
^{^X} Выход ^{^J} Выворнять ^{^W} Поиск ^{^V} СледСтр ^{^U} ОтмВырезк ^{^T} Словарь

Рисунок 12 – список пользователей

Далее измените интерпретатор с /bin/bash на /bin/sh.

```

GNU nano 2.2.4                               Файл: /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
Debian-exim:x:101:103::/var/spool/exim4:/bin/false
statd:x:102:65534::/var/lib/nfs:/bin/false
messagebus:x:103:106::/var/run/dbus:/bin/false
avahi:x:104:107:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
gdm:x:105:113:Gnome Display Manager:/var/lib/gdm:/bin/false
haldaemon:x:106:114:Hardware abstraction layer,,,:/var/run/hald:/bin/false
usbmux:x:107:46:usbmux daemon,,,:/home/usbmux:/bin/false
saned:x:108:117::/home/saned:/bin/false
hplip:x:109:7:HPLIP system user,,,:/var/run/hplip:/bin/false
user:x:1000:1000:user,,,:/home/user:/bin/bash
kibevs:x:1002:100::/home/kibevs:/bin/sh

```

[Прочитано 30 строк]

^G Помощь	^O Записать	^R ЧитФайл	^Y ПредСтр
^X Выход	^J Выровнять	^W Поиск	^K Вырезать
		^V СледСтр	^C ТекПозиц
			^T Словарь

Рисунок 13 – список пользователей

Сохраните полученный файл **Ctrl+O** затем **Enter**, потом **Ctrl+X**.

С помощью таких же действий можно изменить и другие поля в файле `/etc/passwd`. Однако данный способ хорош для одного редактирования нескольких пользователей. Когда же необходимо как-либо изменить учетные записи 100,1000 и более пользователей, то более разумным кажется использование специальных команд.

10) Команда `usermod`.

Команда `usermod` предназначена для редактирования различных полей в следующих файлах:

- `/etc/group`
- `/etc/gshadow`
- `/etc/passwd`
- `/etc/shadow`

Формат команды:

`usermod [парметры] LOGIN`

То есть в зависимости от переданного параметра она изменит тот или иной файл (`man usermod` содержит много увлекательных подробностей). Вам понадобится ключ `-s`, который задет имя командной оболочки. Для начала просмотрите содержимое файла `/etc/passwd`.

\$ cat /etc/passwd

```
user@work:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
Debian-exim:x:101:103::/var/spool/exim4:/bin/false
statd:x:102:65534::/var/lib/nfs:/bin/false
messagebus:x:103:106::/var/run/dbus:/bin/false
avahi:x:104:107:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
gdm:x:105:113:Gnome Display Manager:/var/lib/gdm:/bin/false
haldaemon:x:106:114:Hardware abstraction layer,,,:/var/run/hald:/bin/false
usbmux:x:107:46:usbmux daemon,,,:/home/usbmux:/bin/false
saned:x:108:117::/home/saned:/bin/false
hplip:x:109:7:HPLIP system user,,,:/var/run/hplip:/bin/false
user:x:1000:1000:user,,,:/home/user:/bin/bash
kibevs:x:1002:100::/home/kibevs:/bin/sh
user@work:~$
```

Рисунок 14 – список пользователей

Теперь выполните:

\$ sudo usermod -s /bin/bash kibevs

Если все прошло успешно, то выведите содержимое /etc/passwd на экран. Должно получиться как на рисунке 15 (последняя строка).

```
user@work:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
Debian-exim:x:101:103::/var/spool/exim4:/bin/false
statd:x:102:65534::/var/lib/nfs:/bin/false
messagebus:x:103:106::/var/run/dbus:/bin/false
avahi:x:104:107:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/bin/false
gdm:x:105:113:Gnome Display Manager:/var/lib/gdm:/bin/false
haldaemon:x:106:114:Hardware abstraction layer,,,:/var/run/hald:/bin/false
usbmux:x:107:46:usbmux daemon,,,:/home/usbmux:/bin/false
saned:x:108:117::/home/saned:/bin/false
hplip:x:109:7:HPLIP system user,,,:/var/run/hplip:/bin/false
user:x:1000:1000:user,,,:/home/user:/bin/bash
kibevs:x:1002:100::/home/kibevs:/bin/bash
```

Рисунок 15 – список пользователей

IV. Создание\Редактирование\Удаление группы

Для **создания новой группы** существует команда `groupadd`. Формат команды:

`groupadd` параметры имя

Параметры могут быть следующими (рассмотрим только основные, более подробно `man groupadd`):

-g gid – идентификатор группы.

-r – необходимость создания системной группы.

-f – блокирует создание групп с одинаковыми GID.

11) Создайте новую группу. Для этого выполните:

```
$ sudo groupadd -g 1001 fvs
```

В данной команде создает новая группа `fvs` с идентификатором 1001.

ПРИМЕЧАНИЕ!!! Идентификатор групп, в зависимости от дистрибутива, меньше 1000 относится к системным группам, соответственно больше 1000 - к группам пользователя.

Выведите содержимое `/etc/group` на экран. Результат должен быть таким:

```
voice:x:22:
cdrom:x:24:user
floppy:x:25:user
tape:x:26:
sudo:x:27:
audio:x:29:user,kibevs
dip:x:30:user
www-data:x:33:
backup:x:34:
operator:x:37:
list:x:38:
irc:x:39:
src:x:40:
gnats:x:41:
shadow:x:42:
utmp:x:43:
video:x:44:user
sasl:x:45:
plugdev:x:46:user
staff:x:50:
games:x:60:kibevs
users:x:100:
nogroup:x:65534:
libuid:x:101:
crontab:x:102:
Debian-exim:x:103:
mlocate:x:104:
ssh:x:105:
messagebus:x:106:
avahi:x:107:
netdev:x:108:user
lpadmin:x:109:
ssl-cert:x:110:
fuse:x:111:
utempter:x:112:
gdm:x:113:
haldaemon:x:114:
powerdev:x:115:user
scanner:x:116:saned,user
saned:x:117:
user:x:1000:kibevs
done:x:998:user
fvs:x:1001:
user@work:~$ █
```

Рисунок 16 – список групп

12) Для редактирования группы существует команда `groupmod`. У неё такие же ключи, как и у `groupadd`. Также отредактировать группу можно, изменив вручную файл `/etc/group` (необходимо наличие прав суперпользователя).

13) Добавьте нового пользователя в созданную группу `fvs`. Выполните:

```
$ sudo usermod -G fvs kibevs
```

Здесь `-G` в качестве дополнительной группы для пользователя `kibevs` указывает `fvs`.

Проверьте результат. Выполните:

```
$ cat /etc/group
```

Результат должен быть таким:

```
cdrom:x:24:user
floppy:x:25:user
tape:x:26:
sudo:x:27:
audio:x:29:user
dip:x:30:user
www-data:x:33:
backup:x:34:
operator:x:37:
list:x:38:
irc:x:39:
src:x:40:
gnats:x:41:
shadow:x:42:
utmp:x:43:
video:x:44:user
sasl:x:45:
plugdev:x:46:user
staff:x:50:
games:x:60:
users:x:100:
nogroup:x:65534:
libuuid:x:101:
crontab:x:102:
Debian-exim:x:103:
mlocate:x:104:
ssh:x:105:
messagebus:x:106:
avahi:x:107:
netdev:x:108:user
lpadmin:x:109:
ssl-cert:x:110:
fuse:x:111:
utempter:x:112:
gdm:x:113:
haldaemon:x:114:
powerdev:x:115:user
scanner:x:116:saned,user
saned:x:117:
user:x:1000:
done:x:998:user
fvs:x:1001:kibevs
```

Рисунок 17 – список групп

Абсолютно то же самое можно сделать вручную, как в пункте 9.

V. Удаление пользователя и группы

14) Старые учетные записи и группы необходимо удалять полностью, чтобы не оставлять злоумышленникам возможности для попыток взлома.

Для удаления учетной записи пользователя есть команда `userdel` (в `man userdel` как всегда за подробностями). Формат команды:

```
userdel [параметры] LOGIN
```

Интерес представляет параметр `-r`, который удаляет файлы в домашнем каталоге с самим домашним каталогом. Выполните:

```
$ sudo userdel -r kibevs
```

Проверьте основные файлы, относящиеся к учетным записям на предмет того, что учетная запись действительно удалена.

15) Выполните:

```
$ grep kibevs /etc/passwd
user@work:~$ grep kibevs /etc/passwd
user@work:~$ █
```

Рисунок 18 – результат поиска в /etc/passwd

Команда grep не нашла пользователя kibevs в файле /etc/passwd.

16) Выполните:

```
$ sudo grep kibevs /etc/shadow
```

Результат работы grep также должен быть отрицательным.

17) Выполните:

```
$ ls /home
user@work:~$ ls /home/
lost+found user
user@work:~$ █
```

Рисунок 19 – результат работы ls

Домашний каталог пользователя kibevs отсутствует.

18) Для **удаления** группы есть команда groupdel. Формат команды

```
groupdel имя_группы
```

Следует заметить, что группу нельзя удалить, если в ней есть пользователи. Сначала их нужно вывести из группы.

Так как пользователь kibevs был удален, то группа fvs не имеет участников. Выполните:

```
$ cat /etc/group | grep fvs
user@work:~$ cat /etc/group | grep fvs
fvs:x:1001:
user@work:~$ █
```

Рисунок 20 – результат работы cat и grep

Теперь спокойно можно удалять группу из системы. Выполните:

```
$ sudo groupdel fvs
```

А затем ещё раз выполните:

```
$ cat /etc/group | grep fvs
user@work:~$ cat /etc/group | grep fvs
user@work:~$ █
```

Рисунок 21 – результат работы cat и grep

Группы fvs больше не существует.

Лабораторная работа №2. Аутентификация в прикладных приложениях при помощи физического объекта.

I. Утилита «Свойства eToken»

1) Подключите eToken к USB-порту. Запустите сначала PKI Client: Меню – Прочие – Start eToken PKI Client. Вид основного окна представлен на рис. 1.

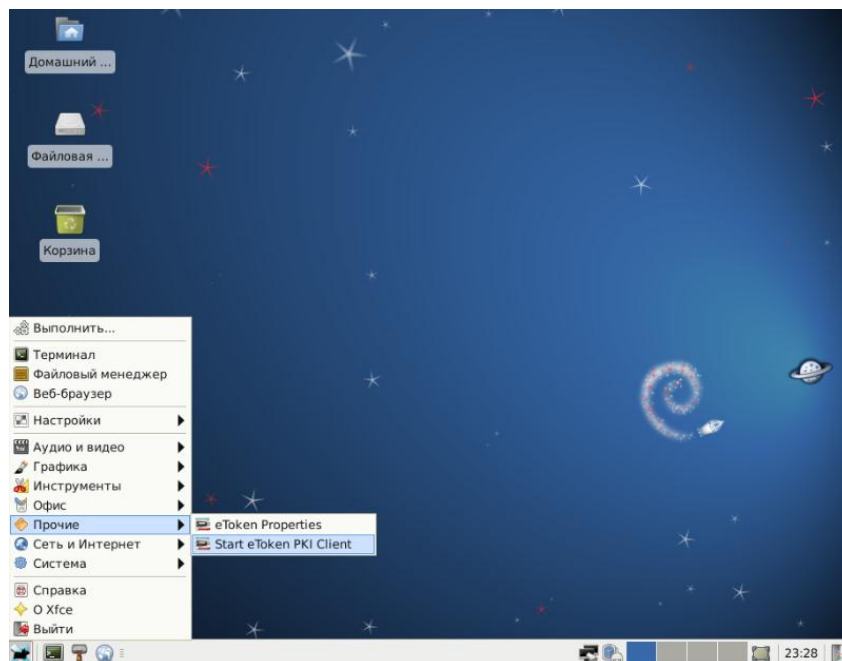


Рис. 1 – Запуск PKI Client

2) Теперь можно запустить утилиту eToken Properties: Меню – Прочие – eToken Properties. Вид основного окна представлен на рис. 2.

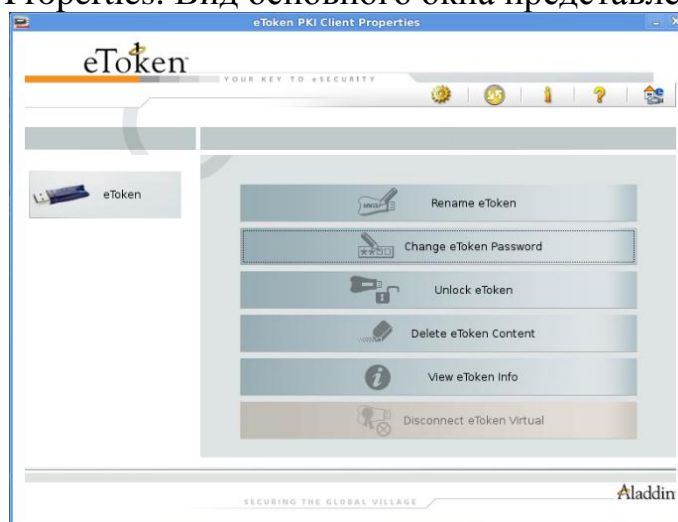


Рис. 2 – Вид основного окна утилиты «Свойства eToken»

3) Смените PIN-код. Используемый PIN-код по умолчанию: «1234567890». При смене PIN-кода необходимо соблюдать требования, предъявляемые к его качеству. Достижение отметки 100% означает, что введённый PIN-код отвечает установленным требованиям (рис. 3).



Рис. 3 – Смена PIN-кода

4) Переименуйте eToken (рис. 4). Для возможности простого определения принадлежности eToken необходимо присвоить ему уникальный в системе идентификатор пользователя (login), которому выдаётся eToken. При первой операции с eToken необходимо ввести PIN-код.



Рис. 4 – Переименование eToken

Измените режим интерфейса на «Дополнительно» (значок «Advanced View» на панели инструментов). В данном режиме предоставляется доступ к дополнительным настройкам и функциям по работе с подключенными eToken (рис. 5). В основном окне режима «Дополнительно» предоставляется информация о выбранном eToken.

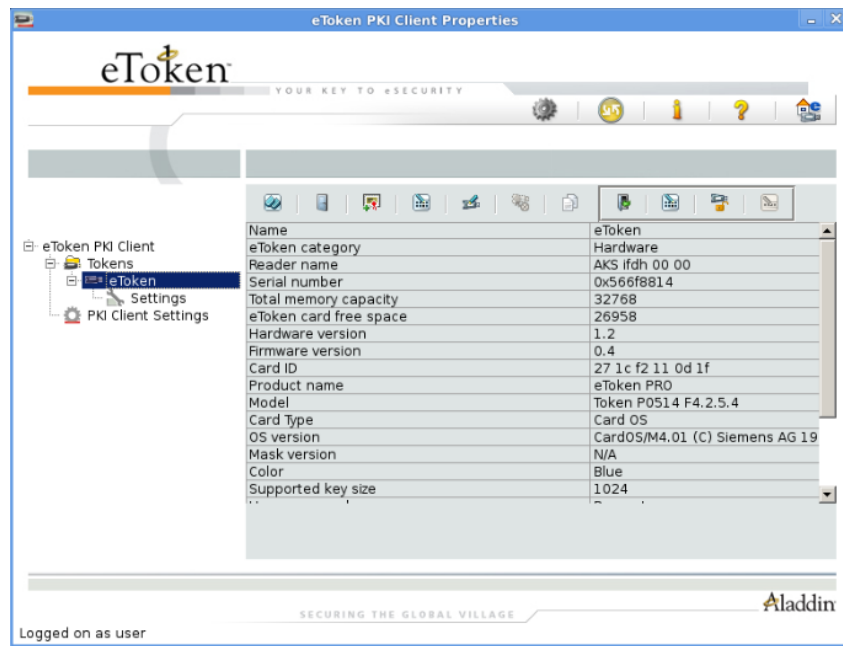


Рис. 5 – Вид основного окна для eToken в режиме «Дополнительно».

5) В разделе «Настройки eToken PKI Client» («eToken PKI Client Settings») возможна установка требований к качеству PIN-кода eToken, которые будут записаны на него при форматировании (рис. 6). Просмотр требований, сохранённых на eToken, возможен в разделе «Настройки» («Settings»), выбранного eToken.

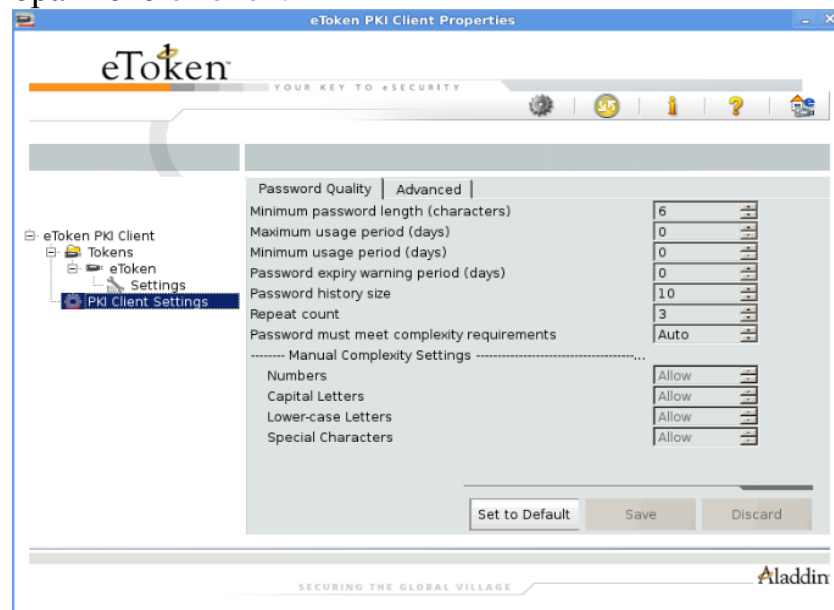


Рис. 6 - Настройка параметров качества PIN-кода eToken.

6) В режиме «Дополнительно» выберите подключенный eToken и на панели инструментов выберите «Форматирование» («Initialize eToken»). В окне «Настройки форматирования eToken» (рис. 7) установите PIN-код eToken или требование к обязательной смене пароля при первом использовании (если оставите PIN-код по умолчанию), и PIN-код администратора eToken. Также можно установить максимальное количество ошибок ввода соответствующих PIN-кодов. Отформатируйте eToken. Внимание! При форматировании есть возможность указать ключ форматирования («Advanced» - «Change Initialization Key»). Не изменяйте

настройки этой вкладки, так как при незнании ключа форматирования нельзя восстановить его в первоначальном состоянии, что приводит к неработоспособности eToken

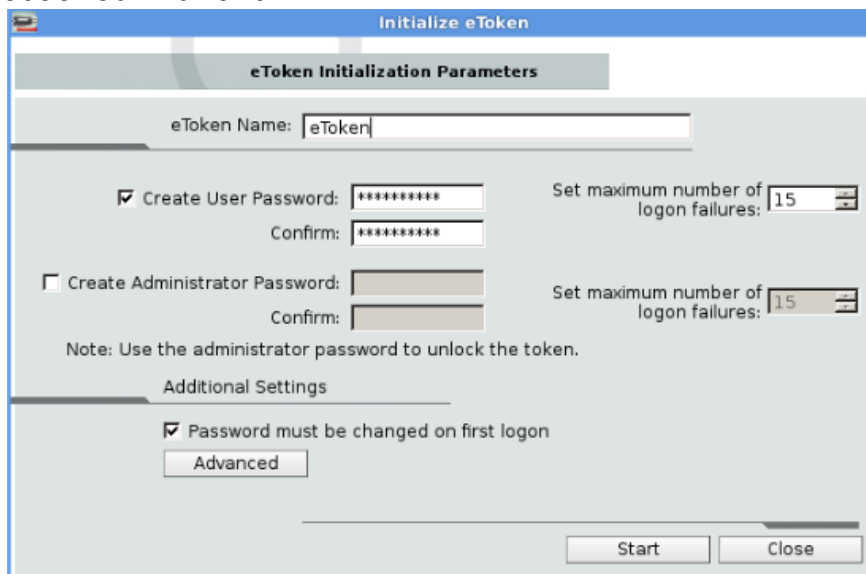


Рис. 7 - Параметры форматирования eToken.

7) Выберите подключенный eToken. На панели инструментов выберите значок «Log on as Administrator». Введите PIN-код администратора (рис. 8). Администратору предоставляются дополнительные функции. На панели инструментов выберите значок «Set user password». Эта функция позволяет администратору задать новый PIN-код eToken, если пользователь забыл свой текущий PIN-код.

Лабораторная работа №3. Дискреционное разграничение доступа

Целью данной работы является практическое изучение дискреционного механизма разграничения доступа на основе встроенных средств ядра Linux, позволяющих управлять доступом к файлам и папкам.

I. Основы дискретной модели

Ход работы

- 1) Войдите в операционную систему под учётной записью user.
- 2) Создайте файл:

```
$ touch file1
```

- 3) Посмотрите атрибуты созданного файла file1. Выполните:

```
$ ls -l file1
```

```
user@work:~$ ls -l file1
-rw-r--r-- 1 user user 0 Map 16 11:25 file1
user@work:~$
```

Рисунок 1 – результат выполнения команды

Рассмотрим полученную строку более подробно:

-	rw-	r--	r--
1	2	3	4

Таблица 1 – биты доступа

Это классическая 9-ти битовая система контроля доступа, характерная для UNIX и UNIX подобных операционных систем. Определим, что означают приведенные в таблице 1 биты доступа:

- 1) Определяет тип файла. Может иметь следующие значения:
 - - это простой файл,
 - d – каталог,
 - s – сокет,
 - p – именованный канал,
 - l – символическая ссылка
- 2) Биты доступа для владельца файла. В нашем случае владелец может читать (r-read), изменять (w-write), выполнять (x-execute) файл.
- 3) Биты доступа для членов группы. В нашем случае члены группы могут читать (r-read) файл. – означает отсутствие каких-либо прав доступа.
- 4) Биты доступа для “остальных” пользователей, то есть те, кто не вошли в группу. В нашем случае члены группы могут читать (r-read) файл. – означает отсутствие каких-либо прав доступа.

Анализируя строку на рисунке 1 далее после 9 битов доступа, видим цифру 1. Данная цифра означает число жестких ссылок на созданный нами файл. Число ссылок равно 1, так как единственная жесткая ссылка на файл – это его родительский каталог.

user	user
1	2

Таблица 2 – владелец и группа

У каждого файла и папки в Linux есть владелец. Для созданного файла владельцем является user (1). У владельца файла, как принято, полный доступ к нему (r,w,x), хотя в нашем случае (пункт 2) владелец может только читать и изменять файл.

Возникают ситуации, когда находясь под другой учетной записью, пользователь хочет получить доступ к файлу\папке другого пользователя. Для этого есть группы. Группа позволяет какому-либо демону\сервису объединить пользователей для доступа. Для нас пользователь user (1) состоит в группе user(2)(Таблица 2). Для группы (пункт 3) тоже определены биты доступа. В нашем случае члены группы user могут только читать файл. Чтобы посмотреть, в какие группы входит пользователь, выполните:

\$ grep user /etc/group

```
user@work:~$ grep user /etc/group
cdrom:x:24:user
floppy:x:25:user
audio:x:29:user
dip:x:30:user
video:x:44:user
plugdev:x:46:user
users:x:100:
netdev:x:108:user
powerdev:x:115:user
scanner:x:116:saned,user
user:x:1000:
done:x:998:user
user@work:~$
```

Рисунок 2 – результат выполнения команды

То есть, являясь, к примеру, членом группы audio, пользователь может работать с утилитами, отвечающими за звук.

0 – это размер файла на диске.

Mar 16 11:25 – дата последней модификации файла.

4) Подводя итоги рассмотренному выше, можно составить матрицу доступа для созданного нами файла.

5)

Категория пользователей (субъекты)	<i>file1</i> (объект)
Владелец	rw
Группа	r
Остальные	r

Таблица 3 – матрица доступа

II. Изменение прав доступа

Для изменения прав доступа к файлам и папкам необходимо воспользоваться командой `chmod`.

Данная команда позволяет работать в двух режимах:

- 1) По символьным обозначениям (мнемоническая спецификация).
- 2) Использование восьмеричной формы записи.

Для каждого файла или каталога Linux различает три категории пользователей: владелец, группа и остальные. Эти категории перечислены в таблице 4:

Категория пользователей	Обозначение
Владелец	u (user)
Группа	g (group)
Остальные	o (others)

Таблица 4 – Категории пользователей и их обозначения

Есть еще одно обозначение – a (all). Это совокупность u+g+o.

Просто так обозначения никакой роли не играют. Команда `chmod` поддерживает установку битов доступа:

	Значение	Биты доступа
=	Установка прав доступа	rwX
-	Отобрать существующие права доступа	rwX
+	Добавить существующие права доступа	rwX

Таблица 5 – Категории пользователей и их обозначения

Для созданного файла в пункте 2 выполните:

```
$ chmod g+wx file1
```

а затем

```
$ ls -l file1
```

```
user@work:~$ chmod g+wx file1
user@work:~$ ls -l file1
-rw-rwxr-- 1 user user 0 Mar 16 11:25 file1
user@work:~$ █
```

Рисунок 3 – результат выполнения команды

В этом примере вы добавили (+) для членов группы (g) новые права (wx) для доступа к файлу file1.

Отберите у “остальных” право на чтение. Выполните:

```
$ chmod o-r file1
```

а затем
\$ ls -l file1

```
user@work:~$ chmod o-r file1
user@work:~$ ls -l file1
-rw-rwx-- 1 user user 0 Map 16 11:25 file1
user@work:~$ █
```

Рисунок 4 – результат выполнения команды
Теперь “остальные” не могут вообще ничего делать с файлом.

2) Использование восьмеричной формы записи.

При использовании восьмеричной формы записи первая цифра относится к владельцу, вторая к группе, а третья – к другим пользователям.

Восьмеричное число	Двоичное число	Режим доступа
0	000	---
1	001	--x
2	010	-w-
4	100	r--

Таблица 6 – восьмеричная и двоичная форма записи

Как считаются суммарные права доступа:

Владелец	Группа	Остальные
$4 + 2 + 1 = 7$	$4 + 1 = 5$	2
$r + w + x = rwx$	$r + x = rx$	w

Таблица 7 – подсчет прав доступа

В результате получается следующая запись. Выполните:

\$ chmod 752 file1 а затем
\$ ls -l file1

```
user@work:~$ chmod 752 file1
user@work:~$ ls -l file1
-rwxr-x-w- 1 user user 0 Map 16 11:25 file1
user@work:~$ █
```

Рисунок 5 – биты доступа

Отберите у группы и остальных право доступа к файлу. Выполните:

\$ chmod 700 file1 && ls -l file1

```
user@work:~$ chmod 700 file1 && ls -l file1
-rwx----- 1 user user 0 Map 16 11:25 file1
user@work:~$ █
```

Рисунок 6 – биты доступа

Как видно биты доступа установились верно.

III. Примеры работы с дискреционной моделью

Создайте в своей домашней директории папку TEMP:

\$ mkdir TEMP

Проверьте свойства для созданной папки. Выполните:

```
$ ls -ld TEMP/
```

```
user@work:~$ ls -ld TEMP/  
drwxr-xr-x 2 user user 4096 Mar 17 00:02 TEMP/  
user@work:~$ █
```

Рисунок 7 – биты доступа

Перейдите в созданную папку и создайте в ней файл. Для этого выполните:

```
$ cd TEMP && touch file
```

Посмотрите для созданного файла права доступа. Выполните:

```
$ ls -l file
```

```
user@work:~$ cd TEMP && touch file  
user@work:~/TEMP$ ls -l  
итого 0  
-rw-r--r-- 1 user user 0 Mar 17 00:15 file  
user@work:~/TEMP$ █
```

Рисунок 8 - биты доступа

Построим матрицу доступа:

Субъект		Объекты	
Текущий пользователь (user)	Категория пользователей	TEMP	file
	Владелец (user)	rwX	rw
	Группа	rX	r
	Другие	rX	r

Таблица 8 – матрица доступов

Таблица 8 демонстрирует наглядно, что **текущий пользователь** user (субъект) имеет полный доступ к папке TEMP (объект) и право на чтение и запись file (объект). Можно проверить это. Выполните:

```
$ cd .. && ls -aliS > TEMP/file
```

Это команда демонстрирует наглядно, что субъект (user) записывает результат вывода команды в файл file (объект). Субъект может записывать в файл, так как:

- 1) Сначала проверяются права доступа для папки TEMP. Они позволяют владельцу, в нашем случае, просматривать содержимое папки (read), удалять\переименовывать файл (write), читать файлы и каталоги и запускать файлы (execute).
- 2) Так как доступ у нашего субъекта для работы с данными в папке есть, следующий объект - это файл (file). Для него у нас тоже есть право на чтение и запись (rw).

Сменим пользователя, для того чтобы посмотреть, как другой субъект сможет прочитать файл. Выполните:

1) \$ su user1

2) Введите пароль 54321. Должно получиться следующее:

```
user@work:~$ su user1
Пароль:
user1@work:/home/user$
```

Рисунок 9 – результат работы команды

3) Сейчас вы сменили пользователя и оказались относительно файла в категории “другие”. Попробуйте вывести содержимое файла (рисунок 10):

```
$ cat TEMP/file
```

Категория “другие” имеет права (rx) на объект (TEMP) и r на объект file. Поэтому мы смогли вывести содержимое файла на экран.

4) Выполните `$ exit && sudo chmod 750 TEMP/file`.

5) Повторите пункты 1,2,3.

6) В результате должна получиться ошибка (рисунок 11):

```
user1@work:/home/users$ cat TEMP/file
итого 124
130867 -rw----- 1 user user 4375 Mar 16 13:32 .bash_history
130817 drwxr-xr-x 19 user user 4096 Mar 17 00:02 .
  2 drwxr-xr-x  4 root root 4096 Mar 13 23:45 ..
130843 drwx----- 3 user user 4096 Mar 15 00:04 .cache
130832 drwxr-xr-x  5 user user 4096 Mar 13 23:48 .config
130839 drwx----- 3 user user 4096 Mar 13 23:48 .dbus
130865 drwx----- 2 user user 4096 Mar 14 20:18 .gconf
130866 drwx----- 2 user user 4096 Mar 14 20:19 .gconfd
130860 drwx----- 2 user user 4096 Mar 13 23:48 .gvfs
130844 drwxr-xr-x  3 user user 4096 Mar 13 23:48 .local
130870 drwxr-xr-x  2 user user 4096 Mar 17 00:15 TEMP
130853 drwxr-xr-x  2 user user 4096 Mar 13 23:48 .wicd
130831 drwxr-xr-x  2 user user 4096 Mar 13 23:48 Видео
130828 drwxr-xr-x  2 user user 4096 Mar 13 23:48 Документы
130825 drwxr-xr-x  2 user user 4096 Mar 13 23:48 Загрузки
130830 drwxr-xr-x  2 user user 4096 Mar 13 23:48 Изображения
130829 drwxr-xr-x  2 user user 4096 Mar 13 23:48 Музыка
130827 drwxr-xr-x  2 user user 4096 Mar 13 23:48 Общедоступные
130823 drwxr-xr-x  2 user user 4096 Mar 13 23:48 Рабочий стол
130826 drwxr-xr-x  2 user user 4096 Mar 13 23:48 Шаблоны
130820 -rw-r--r--  1 user user 3184 Mar 13 23:45 .bashrc
130819 -rw-r--r--  1 user user  675 Mar 13 23:45 .profile
130893 -rw-----  1 user user  620 Mar 17 00:01 .ICEauthority
130822 -rw-r--r--  1 user user  317 Mar 17 00:01 .xsession-errors
130818 -rw-r--r--  1 user user  220 Mar 13 23:45 .bash_logout
130857 -rw-----  1 user user  115 Mar 17 00:01 .Xauthority
130868 -rw-----  1 user user   28 Mar 17 00:01 .dmrc
130861 -rw-r-----  1 user user    5 Mar 17 00:01 .vboxclient-clipboard.pid
130887 -rw-r-----  1 user user    5 Mar 17 00:01 .vboxclient-display.pid
130889 -rw-r-----  1 user user    5 Mar 17 00:01 .vboxclient-seamless.pid
user1@work:/home/users$
```

Рисунок 10 – результат работы команды

```
user1@work:/home/user$ cat TEMP/file
cat: TEMP/file: Отказано в доступе
user1@work:/home/user$
```

Рисунок 11 – результат работы команды

Субъект user1 является для объекта file членом категории “другие”, которая не имеет прав доступа (см. пункт 4). Владельцем для данного объекта пользователь user1 не является.

Доступ к файлу имеют члены группы user (rx). Добавьте в неё туда вашего пользователя. Выполните:

```
$ sudo nano /etc/group
```

Найдите строку `user:x:1000:` и после последнего двоеточия впишите туда user1 (рисунок 12):

```
GNU nano 2.2.4                               Файл: /etc/group
nogroup:x:65534:
libuuid:x:101:
crontab:x:102:
Debian-exim:x:103:
mlocate:x:104:
ssh:x:105:
messagebus:x:106:
avahi:x:107:
netdev:x:108:user
lpadmin:x:109:
ssl-cert:x:110:
fuse:x:111:
utempter:x:112:
gdm:x:113:
haldaemon:x:114:
powerdev:x:115:user
scanner:x:116:saned,user
saned:x:117:
user:x:1000:user1
done:x:998:user,user1
```

Рисунок 12 – результат работы команды

Далее нажмите Ctrl+O, а затем Enter. Далее наберите в терминале exit и повторите пункты 1-3. Содержимое файла должно вывестись на экран.

Лабораторная работа №4. Аудит

I. Создание, удаление и модификация правил аудита

`auditctl` – это утилита, позволяющая управлять подсистемой аудита ядра Linux.

Для выполнения работы интересны 4 опции этой утилиты:

- a – добавить новое правило в список;
- d – удалить последнее введенное правило из списка;
- D – очистить список правил;
- l – вывести список текущих правил.

1) Выведите список текущих правил. Выполните в терминале:

```
$ sudo auditctl -l
user@work:~$ sudo auditctl -l
No rules
user@work:~$
```

Рисунок 1 – результат вывода команды

“No rules” – это значит, что в списке правил ничего нет.

Для добавления правил используется следующая форма команды `auditctl`:

`auditctl [OPTIONS]`

Для выполнения лабораторной работы нам понадобятся 4 вышеперечисленные опции (man `auditctl` более подробно).

2) Опция `-a`.

Общий вид выглядит так: `auditctl -a list,action`

list – это имя события, которое собственно добавится в список.

Определены 5 основных событий:

task – события, связанные с созданием процессов;

entry – события, происходящие при входе в системный вызов;

user – события, использующие параметры пользовательского пространства, такие как `uid`, `pid` и `gid`;

exclude – используется для игнорирования событий. Иначе говоря, это событие работает как фильтр для тех событий, которые вы не хотите видеть.

action - это действие, которое должно произойти в ответ на возникшее событие (обязательно указывать в команде). Их всего два: `never` и `always`. В первом случае события не записываются в журнал событий, во втором – записываются.

В итоге правило может выглядеть так:

`auditctl -a user,always`

То есть события, использующие параметры пользовательского пространства, такие как `uid`, `pid` и `gid`, будут заноситься в журнал.

Опция `-a` сама по себе несколько обширна и охватывает события всей системы. Если необходимо отфильтровать вывод в журнал, то пригодятся опции `-S` и `-F`.

3) Опция '-S' задает имя системного вызова или номер (например open,fork,exec).

```
auditctl -a entry,always -S fork
```

Здесь в журнал запишутся события, связанные при входе в системный вызов, в частности системного вызова fork().

4) Опция '-F' - еще один дополнительный фильтр, имеющий множество полей. Нам все не понадобятся, а нужно будет только одно:

```
-F n=v
```

Здесь n – имя (имеет много значений, как всегда man auditctl в помощь).

= - оператор (помимо =, есть 7 других значений).

v – значение, зависящие напрямую от имени.

С учетом оговоренных ключей можно расширить опцию -a. Выполните в терминале данную команду:

```
$ sudo auditctl -a exit,always -S close -F gid=1
```

```
user@work:~$ sudo auditctl -a exit,always -S close -F gid=1
user@work:~$ █
```

Рисунок 2 - результат выполнения команды

При выходе из системного вызова в журнал будут записаны только события, связанные с вызовом close для пользователя с GID = 1 (группа пользователя root).

Теперь можно вывести список текущих правил. Выполните:

```
$ sudo auditctl -l
```

```
user@work:~$ sudo auditctl -a exit,always -S close -F gid=1
user@work:~$ sudo auditctl -l
LIST_RULES: exit,always gid=1 (0x1) syscall=close
user@work:~$ █
```

Рисунок 3 - результат вывода команды

Заданное правило было успешно применено.

II. Дополнительные опции.

Опция **-w** позволяет добавить правило для наблюдения за **объектом файловой системы**. Форма опции:

```
-w path
```

Где path – абсолютный путь к файлу. Например, так:

```
auditctl -w /etc/inittab
```

Здесь в журнал будут помещаться все события, относящиеся к файлу inittab.

Опция **-p** обычно работает в связке с **-w**, так как позволяет задать некий фильтр для отслеживания изменения следующих битов доступа:

```
a,r,w,x
```

Применим опцию **-p** к предыдущей команде:

```
auditctl -w /etc/inittab -p rw
```

Здесь в журнал будут помещаться все события, относящиеся к файлу inittab при попытке его чтения (r-read) или модификации (w-write).

III. Опции -d и -D.

Для начала занесем еще одно правило. Выполните:

```
$ sudo auditctl -a entry,always -S fork (1)
```

```
user@work:~$ sudo auditctl -a entry,always -S fork
```

Рисунок 4 - результат выполнения команды

Посмотрим список текущих правил. Выполните:

```
user@work:~$ sudo auditctl -l
LIST_RULES: entry,always syscall=fork
LIST_RULES: exit,always gid=1 (0x1) syscall=close
user@work:~$ █
```

Рисунок 5 - результат вывода команды

Удалим из списка правил команду (1). Для этого применим ключ -d, который удаляет последнее введенное правило по известным list и action, а также системному вызову (syscall). То есть, фактически вам необходимо переписать команду (1) только с ключом -d. Выполните:

```
$ sudo auditctl -d entry,always -S fork
```

А затем выполните:

```
$ sudo auditctl -l
```

```
user@work:~$ sudo auditctl -d entry,always -S fork
user@work:~$ sudo auditctl -l
LIST_RULES: exit,always gid=1 (0x1) syscall=close
user@work:~$ █
```

Рисунок 6 - результат вывода команды

Как видно - правило (1) удалилось.

Чтобы полностью очистить список правил, нужно применить ключ -D. Выполните:

```
$ sudo auditctl -D
```

```
user@work:~$ sudo auditctl -D
No rules
user@work:~$ █
```

Рисунок 7 - результат выполнения команды

IV. Конфигурационные файлы

Главный конфигурационный файл располагается в каталоге:

```
/etc/audit/audit.conf
```

В рамках этой лабораторной работы ничего изменять в нем не нужно !!!

Файл, в который вы и записывали предыдущие правила, находится в каталоге /etc/audit/audit.rules. Этот файл содержит правила аудита в формате auditctl. При своей работе демон auditd использует именно эти файлы.

При перезагрузке ОС в файле /etc/audit/audit.rules все правила удаляются.

V. Пример правил аудита

Рассмотрим пример простейшего сценария для демона auditd, который позволяет записывать в журнал все попытки как-либо изменить файл.

Для начала создайте пустой файл. Выполните:

```
$ touch 123
```

Перенаправьте вывод команды ls в созданный вами файл. Выполните:

```
$ ls -al > 123
```

Убедитесь, что файл 123 не пуст:

```
$ cat 123
```

```
user@work:~$ cat 123
итого 116
130817 drwxr-xr-x 18 user user 4096 Map 15 22:00 .
  2 drwxr-xr-x  4 root root 4096 Map 13 23:45 ..
130843 drwx----- 3 user user 4096 Map 15 00:04 .cache
130832 drwxr-xr-x  5 user user 4096 Map 13 23:48 .config
130839 drwx----- 3 user user 4096 Map 13 23:48 .dbus
130865 drwx----- 2 user user 4096 Map 14 20:18 .gconf
130866 drwx----- 2 user user 4096 Map 14 20:19 .gconfd
130860 drwx----- 2 user user 4096 Map 13 23:48 .gvfs
130844 drwxr-xr-x  3 user user 4096 Map 13 23:48 .local
130853 drwxr-xr-x  2 user user 4096 Map 13 23:48 .wcid
130831 drwxr-xr-x  2 user user 4096 Map 13 23:48 Видео
130828 drwxr-xr-x  2 user user 4096 Map 13 23:48 Документы
130825 drwxr-xr-x  2 user user 4096 Map 13 23:48 Загрузки
130830 drwxr-xr-x  2 user user 4096 Map 13 23:48 Изображения
130829 drwxr-xr-x  2 user user 4096 Map 13 23:48 Музыка
130827 drwxr-xr-x  2 user user 4096 Map 13 23:48 Общедоступные
130823 drwxr-xr-x  2 user user 4096 Map 13 23:48 Рабочий стол
130826 drwxr-xr-x  2 user user 4096 Map 13 23:48 Шаблоны
130867 -rw-----  1 user user 3228 Map 15 21:53 .bash_history
130820 -rw-r--r--  1 user user 3184 Map 13 23:45 .bashrc
130819 -rw-r--r--  1 user user  675 Map 13 23:45 .profile
130892 -rw-----  1 user user  620 Map 15 21:53 .ICEauthority
130822 -rw-r--r--  1 user user  317 Map 15 21:54 .xsession-errors
130818 -rw-r--r--  1 user user  220 Map 13 23:45 .bash_logout
130857 -rw-----  1 user user  115 Map 15 21:53 .Xauthority
130861 -rw-----  1 user user   28 Map 15 21:53 .dmrc
130868 -rw-r-----  1 user user   5 Map 15 21:53 .vboxclient-clipboard.pid
130887 -rw-r-----  1 user user   5 Map 15 21:53 .vboxclient-display.pid
130889 -rw-r-----  1 user user   5 Map 15 21:53 .vboxclient-seamless.pid
130870 -rw-r--r--  1 user user   0 Map 15 22:01 123
```

Рисунок 8 - результат вывода команды

Очистите журнал, выполнив команду:

```
$ sudo auditctl -D
```

Добавьте в журнал новое правило по контролю за созданным вами файл 123:

```
$ sudo auditctl -w /home/user/123 -p rwx
```

В журнал будут заноситься события, связанные с файлом 123 при попытке доступа к нему (биты r,w,x).

Произведите анализ журнала.

VI. Анализ журнала

Демон auditd создает свой журнал в файле /var/log/audit/audit.log. Данный файл можно анализировать как вручную, так и с помощью утилиты aureport.

Вызовите команду aureport с флагом '-f':

```
$ sudo aureport -f
```

Вы увидите примерно следующее:

```
user@work:~$ sudo aureport -f
/sbin/audispd permissions should be 0750

File Report
=====
# date time file syscall success exe auid event
=====
1. 15.03.2012 22:05:27 123 5 yes /bin/cat -1 4
user@work:~$
```

Рисунок 9 - результат выполнения команды

Рассмотрите поля (значение полей, а также количество записанных событий, могут отличаться у вас от тех, что приведены на рисунке 9):

- # - порядковый номер события
- date\time** – дд.мм.гг и время доступа
- file** – собственно файл, к которому был доступ
- syscall** – номер системного вызова
- success** – завершилась ли успешно попытка
- exe** – программа производившая доступ
- auid** – lognuid

Получив список всех попыток доступа и номера событий, каждое из них можно проанализировать индивидуально с помощью утилиты ausearch.

VII. Утилита ausearch

Применяется как дополнительный фильтр при анализе журналов аудита.

ausearch можно использовать для поиска событий по именам системных вызовов. Выполните:

```
$ sudo ausearch -sc open
```

Так как в пункте 3.3 мы выводили содержимое файла, то использовался системный вызов open:

```
user@work:~$ sudo ausearch -sc open
/sbin/audispd permissions should be 0750
----
time->Thu Mar 15 22:05:27 2012
type=PATH msg=audit(1331823927.888:4): item=0 name="123" inode=130870 dev=08:06 mode=01
00644 ouid=1000 ogid=1000 rdev=00:00
type=CWD msg=audit(1331823927.888:4): cwd="/home/user"
type=SYSCALL msg=audit(1331823927.888:4): arch=40000003 syscall=5 success=yes exit=3 a0
=bfcc57ba a1=8000 a2=0 a3=2 items=1 ppid=1715 pid=1761 auid=4294967295 uid=1000 gid=100
0 euid=1000 suid=1000 fsuid=1000 egid=1000 sgid=1000 fsgid=1000 tty=pts0 ses=4294967295
comm="cat" exe="/bin/cat" key=(null)
```

Рисунок 10 - результат вывода команды

Должно получиться примерно так, как на рисунке 10. Отметьте пару вида msg=audit(1331823927.888:4) (5,7 строка). Число, стоящее после знака : , обозначает id события. В данном случае оно равно 4 (у вас оно может отличаться).

Используя этот id можно проанализировать журнал с помощью ключа – а:

```
$ sudo ausearch -a 4
```

```
user@work:~$ sudo ausearch -a 4
/sbin/audispd permissions should be 0750
----
time->Thu Mar 15 22:05:27 2012
type=PATH msg=audit(1331823927.888:4): item=0 name="123" inode=130870 dev=08:06 mode=01
00644 ouid=1000 ogid=1000 rdev=00:00
type=CWD msg=audit(1331823927.888:4): cwd="/home/user"
type=SYSCALL msg=audit(1331823927.888:4): arch=400000003 syscall=5 success=yes exit=3 a0
=bfcc57ba a1=8000 a2=0 a3=2 items=1 ppid=1715 pid=1761 auid=4294967295 uid=1000 gid=100
0 euid=1000 suid=1000 fsuid=1000 egid=1000 sgid=1000 fsgid=1000 tty=pts0 ses=4294967295
comm="cat" exe="/bin/cat" key=(null)
```

Рисунок 11 - результат вывода команды

Зная имя исполняемого файла (именно того с помощью которого мы выводили значение файла на экран) можно также отфильтровать журнал:

```
$ sudo ausearch -x /bin/cat
```

```
user@work:~$ sudo ausearch -x /bin/cat
/sbin/audispd permissions should be 0750
----
time->Thu Mar 15 22:05:27 2012
type=PATH msg=audit(1331823927.888:4): item=0 name="123" inode=130870 dev=08:06 mode=01
00644 ouid=1000 ogid=1000 rdev=00:00
type=CWD msg=audit(1331823927.888:4): cwd="/home/user"
type=SYSCALL msg=audit(1331823927.888:4): arch=400000003 syscall=5 success=yes exit=3 a0
=bfcc57ba a1=8000 a2=0 a3=2 items=1 ppid=1715 pid=1761 auid=4294967295 uid=1000 gid=100
0 euid=1000 suid=1000 fsuid=1000 egid=1000 sgid=1000 fsgid=1000 tty=pts0 ses=4294967295
comm="cat" exe="/bin/cat" key=(null)
```

Рисунок 12 - результат вывода команды

Команда `ausearch` имеет много полезных опций, как всегда в `man ausearch` за подробностями.

Лабораторная работа №5. PAM (Pluggable Authentication Modules) – Модули аутентификации

Классическая *nix модель предполагает хранение данных относящихся к учетной записи в файлах /etc/passwd и /etc/shadow. Все это хорошо для стандартной процедуры аутентификации, но если необходимо внедрить более «продвинутые» методы, например: биометрия, смарт-карты, сложные системы аутентификации и т.д. Сразу же встает вопрос о перекомпиляции ядра, поиске драйверов для работы программно-аппаратных средств (биометрия). Технология PAM решает эту задачу, так как создает прослойку между низкоуровневыми модулями и приложением осуществляющим аутентификацию.



Рисунок 1 – Библиотека PAM обрабатывает файл pam.d и загружает соответствующие модули

I. Основы

Механизм PAM объединяет множество низкоуровневых схем аутентификации в API высокого уровня, позволяющий создавать приложения, использующие аутентификацию независимо от применяемой схемы аутентификации. Принципиальным свойством PAM является динамическая настройка аутентификации при помощи файла /etc/pam.d или /etc/pam.conf.

1) Основной конфигурационный файл для PAM называется /etc/pam.conf. Просмотрите его содержимое, выполнив:

```
$ cat /etc/pam.conf
```

```
user@work:~$ cat /etc/pam.conf
# -----#
# /etc/pam.conf #
# -----#
#
# NOTE
# ----
#
# NOTE: Most program use a file under the /etc/pam.d/ directory to setup their
# PAM service modules. This file is used only if that directory does not exist.
# -----#
# Format:
# serv. module      ctrl      module [path]      ...[args..]      #
# name type         flag
user@work:~$ █
```

Рисунок 2 – Вывод команды cat

Файл пуст и не содержит никаких конфигурационных данных, а в абзаце NOTE явно написано, что программы будут использовать директорию

/etc/pam.d/ для доступа к своим конфигурационным файлам, в противном случае конфигурация будет читаться отсюда.

2) Проверьте состав каталога /etc/pam.d. Выполните:

```
$ ls /etc/pam.d/
```

```
user@work:~$ ls /etc/pam.d/
atd          common-auth          cups             other          xscreensaver
chfn         common-password     gdm             passwd
chpasswd    common-session      gdm-autologin  polkit-1
chsh        common-session-noninteractive login            su
common-account cron                 newusers        sudo
user@work:~$ █
```

Рисунок 3 – Вывод команды ls

На рисунке 3 вы видите директорию pam.d содержащую конфигурационные файлы для каждого сервиса: cron,su,sudo (например).

3) Низкоуровневые модули присутствуют в системе. Администратор может использовать эти модули для построения цепочек. Выполните (команда может выполняться долго):

```
$ sudo find / -name pam_*.so
```

```
/lib/security/pam_ck_connector.so
/lib/security/pam_selinux.so
/lib/security/pam_mkhome.so
/lib/security/pam_smack.so
/lib/security/pam_warn.so
/lib/security/pam_listfile.so
/lib/security/pam_group.so
/lib/security/pam_exec.so
/lib/security/pam_access.so
/lib/security/pam_rhosts.so
/lib/security/pam_issue.so
/lib/security/pam_mail.so
/lib/security/pam_vbox.so
/lib/security/pam_lastlog.so
/lib/security/pam_tally2.so
/lib/security/pam_tally.so
/lib/security/pam_filter.so
/lib/security/pam_securetty.so
/lib/security/pam_rootok.so
/lib/security/pam_deny.so
/lib/security/pam_ftp.so
/lib/security/pam_pwhistory.so
/lib/security/pam_xauth.so
/lib/security/pam_localuser.so
/lib/security/pam_succeed_if.so
/lib/security/pam_time.so
/lib/security/pam_timestamp.so
/lib/security/pam_permit.so
/lib/security/pam_faildelay.so
/lib/security/pam_sepermit.so
/lib/security/pam_keyinit.so
/lib/security/pam_shells.so
/lib/security/pam_userdb.so
/lib/security/pam_namespace.so
/lib/security/pam_debug.so
/lib/security/pam_motd.so
/lib/security/pam_gnome_keyring.so
/lib/security/pam_env.so
/lib/security/pam_nologin.so
/lib/security/pam_wheel.so
/lib/security/pam_loginuid.so
/lib/security/pam_cracklib.so
/lib/security/pam_umask.so
user@work:~$ █
```

Рисунок 4 – Результат работы find

4) Перейдите в каталог /etc/pam.d. Выполните:

```
$ cd /etc/pam.d/
```

5) Выведите на экран конфигурационный файл для сервиса cron (системный планировщик). На его примере рассмотрим структуру конфигурационного файла. Выполните:

```
$ cat cron
```

```
user@work:/etc/pam.d$ cat cron
#
# The PAM configuration file for the cron daemon
#
@include common-auth

# Read environment variables from pam_env's default files, /etc/environment
# and /etc/security/pam_env.conf.
session      required pam_env.so

# In addition, read system locale information
session      required pam_env.so envfile=/etc/default/locale

@include common-account
@include common-session-noninteractive
# Sets up user limits, please define limits for cron tasks
# through /etc/security/limits.conf
session      required pam_limits.so
```

Рисунок 5 – Результат вывода cat

Поле `@include` **имя сервиса**. Если данному сервису требуется часть функционала другого, то используется конструкция `@include`. В данном случае требуется сервис `common-auth` (рисунок 3).

Присутствует несколько строк, начинающихся с `session`. Рассмотрим строку вида

```
session      required pam_env.so envfile=/etc/default/locale (1)
```

Это основное PAM правило, состоящее из 4 столбцов:

Для приведенной выше строки (1) имеем:

`session` - модуль поддержки сессии и регистрации действий пользователя.

`required` – модуль обязателен.

`pam_env.so` – имя модуля.

`envfile=/etc/default/locale` – аргумент.

б) Все правила записываются в логическом порядке и следуют один за другим (рисунок 5).

ПРИМЕЧАНИЕ!!! Создание собственных PAM модулей не всегда нужно для устанавливаемых программ, так как в зависимости от дистрибутива (RPM-based, DEB –based, Source-based) они уже включены разработчиками. А вот модификация существующих модулей вполне возможна для требуемых целей.

Тип модуля	Флаг, определяющий параметры модуля	Полный путь к файлу модуля	Аргументы модуля
auth – используется для аутентификации и проверки привилегий пользователя	required – модуль обязателен	/lib/security/имя_модуля.so	Специфичные для модуля аргументы
account – модуль распределения ресурсов системы между пользователями	optional – необязателен		
session – модуль поддержки сессии и регистрации действий пользователя	sufficient – достаточный		
password – модуль проверки пароля	requisite – модуль обязателен, а в случае ошибки управление передается приложению		

Таблица 1 – основные элементы правил

II. Практика

- 7) При выполнении некоторых лабораторных работ вы будете использовать команду su, для того чтобы сменить свой идентификатор на идентификатор другого пользователя. При нормальной работе эта утилита требует пароль, а также она имеет модуль PAM (Рисунок 3).

ПРИМЕЧАНИЕ!!! Использует ли та или иная утилита систему PAM можно с помощью команды ldd. Например, для утилиты su:

```
$ ldd /bin/su
```

```

user@work:~$ ldd /bin/su
linux-gate.so.1 => (0xb7706000)
libpam.so.0 => /lib/libpam.so.0 (0xb76ea000)
libpam_misc.so.0 => /lib/libpam_misc.so.0 (0xb76e7000)
libc.so.6 => /lib/i686/cmov/libc.so.6 (0xb759f000)
libdl.so.2 => /lib/i686/cmov/libdl.so.2 (0xb759b000)
libcrypt.so.1 => /lib/i686/cmov/libcrypt.so.1 (0xb7569000)
/lib/ld-linux.so.2 (0xb7707000)
user@work:~$ █

```

Рисунок 6 – Результат работы ldd

Как видно из рисунка 6 утилита su использует библиотеки для работы с PAM.

8) Отредактируйте модуль su для утилиты su. Выполните:

```
$ sudo nano /etc/pam.d/su
```

Перед строкой:

```
auth sufficient pam_rootok.so
```

Добавить строку вида:

```
auth sufficient pam_permit.so
```

Данная строка означает, что аутентификации при использовании утилиты su достаточно просто получить к ней доступ (man pam_permit).

9) Сохраните получившееся, нажав Ctrl+O,Enter,Ctrl+X.

10) В результате вы должны без ввода пароля получить доступ к любой учетной записи в ОС. Выполните:

```
$ su
```

```

user@work:~$ su
root@work:/home/user# █

```

Рисунок 7 – root доступ

ПРИМЕЧАНИЕ!!! su без аргументов осуществляет доступ к учетной записи суперпользователя.

Теперь вы можете делать **АБСОЛЮТНО ЧТО УГОДНО В СИСТЕМЕ**. Изменив одну лишь строчку, вы получили доступ ко всей системе, что еще раз подтверждает большую сложность и ответственность написания PAM модулей.

11)Выполните:

```
$ exit
```

12) Удалите добавленную строку в пункте 8, выполнив шаги 8-9.

Хорошая вводная статья для новичков по разработке собственных PAM модулей <http://www.xakep.ru/magazine/xa/086/112/1.asp>.

Лабораторная работа №6. Виртуализация на уровне операционной системы

Метод виртуализации, при котором ядро операционной системы поддерживает несколько изолированных экземпляров пространства пользователя, вместо одного.

1) Введение

2) Для того чтобы увеличить безопасность служб, необходимо создать директорию, которая будет являться для программы корневой. В Linux для этого существует команда `chroot`, которая собственно создает окружение `chroot`:

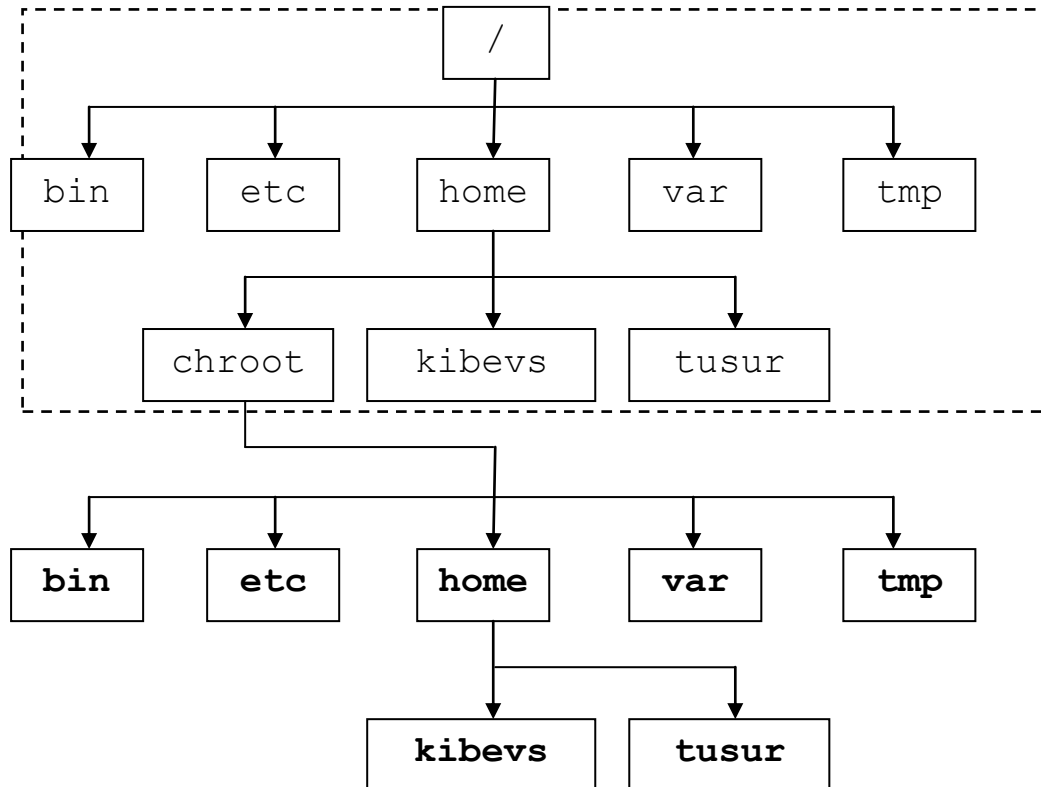


Рисунок 1- пример окружения `chroot`

Посмотрите на рисунок 1. На нем показана часть файловой системы Linux. В домашнем каталоге пользователей (`home`) присутствуют два пользователя `kibevs` и `tusur`, а также специальная директория `chroot` (название не обязательно должно носить имя `chroot`).

Внутри этой директории присутствуют свои каталоги `bin`, `etc`, `home`, `var`, `tmp`. Содержание этих каталогов может быть эквивалентно реальным их «аналогам», а может содержать только те системные утилиты, библиотеки, которые необходимо только для работы отдельного сервиса. Таким образом, именно в этом пространстве будет работать сервис, считая, что это и есть реальная файловая система сервера.

3) Более подробную справку можно получить из `info` документации. Выполните:

```
$ info coreutils 'chroot invocation'
```

II. Практика

4) При использовании chroot обычно «запирают» отдельную программу либо из соображений безопасности, либо для тестирования, а иногда для создания так называемых honeypot-ов.

5) Чтобы изучить работу механизма chroot, будем экспериментировать с bash (командный интерпретатор).

6) При работе с chroot придется отойти от схемы, приведенной на рисунке 1. Для начала выведите основные разделы, присутствующие в ОС, чтобы понять, как будет строиться будущий chroot каталог. Выполните:

```
$ ls /
```

```
user@work:~$ ls /
bin  dev  home  lib      media  opt   root  selinux  sys  usr  vmlinuz
boot etc  initrd.img  lost+found  mnt   proc  sbin  srv      tmp  var
```

Рисунок 2- существующие разделы

Структура каталога home:

```
$ ls /home
```

```
user@work:~$ ls /home
lost+found  user
```

Рисунок 3- структура home

В результате получим следующую структуру (для экономии места не все разделы обозначены):

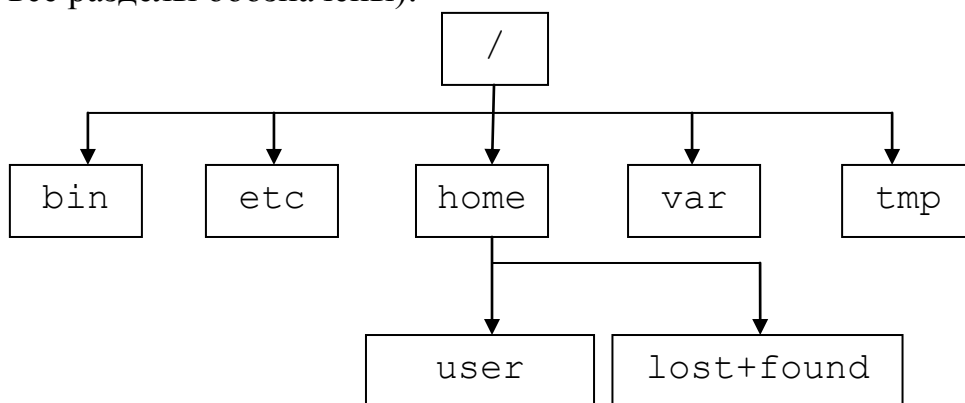


Рисунок 4- текущая структура каталогов

7) Каталог chroot вы будете создавать в домашней директории уже существующего в системе пользователя user, как показано на рисунке:

Выполните:

```
$ mkdir chroot а затем
```

```
$ ls
```

```
user@work:~$ mkdir chroot
user@work:~$ ls
chroot  Видео  Загрузки  Музыка  Рабочий стол
TEMP    Документы  Изображения  Общедоступные  Шаблоны
user@work:~$
```

Рисунок 5- созданный каталог chroot

В результате у вас получится следующая структура:

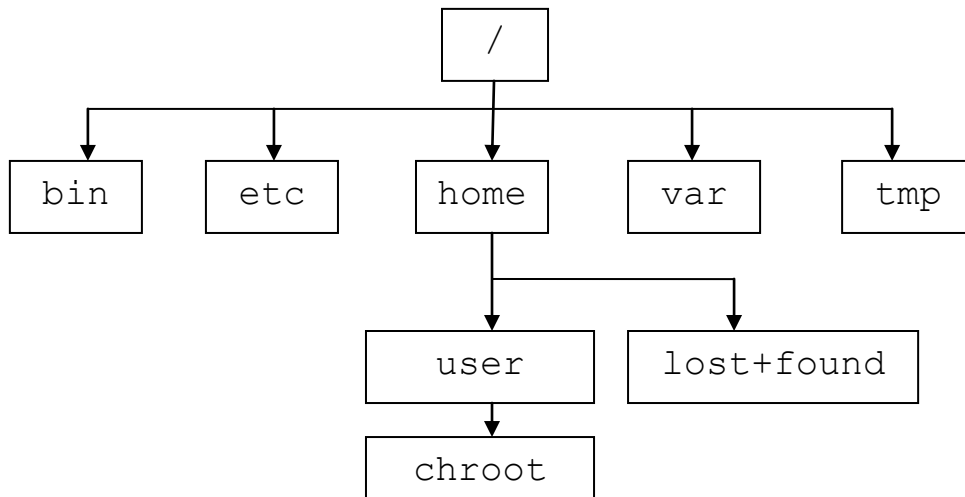


Рисунок 6- текущая структура каталогов

На рисунке 7 созданный каталог chroot будет корнем для нового окружения.

8) Прежде чем попасть в новую корневую директорию, необходимо определить, какие конфигурационные файлы, библиотеки необходимы для работы того или иного сервиса.

Определите, какие динамические библиотеки необходимы для работы bash. Для этого необходимо выполнить:

```
$ sudo ldd /bin/bash
```

```

user@work:~$ sudo ldd /bin/bash
[sudo] password for user:
linux-gate.so.1 => (0xb7739000)
libncurses.so.5 => /lib/libncurses.so.5 (0xb76ef000)
libdl.so.2 => /lib/i686/cmov/libdl.so.2 (0xb76eb000)
libc.so.6 => /lib/i686/cmov/libc.so.6 (0xb75a3000)
/lib/ld-linux.so.2 (0xb773a000)
user@work:~$ █
  
```

Рисунок 7 – список используемых библиотек

9) Теперь создайте изолированную среду для bash. Как видно из рисунка 8 для работы bash требуются библиотеки из каталога /lib. Также требуется каталог /bin для работы bash.

10) Создайте директории bash,bin,lib:

```
$ mkdir chroot/bash
$ mkdir chroot/bash/bin
$ mkdir chroot/bash/lib
```

а затем выполните

```
$ ls chroot/bash
```

```

user@work:~$ mkdir chroot/bash
user@work:~$ mkdir chroot/bash/bin
user@work:~$ mkdir chroot/bash/lib
  
```

Рисунок 8 – создание директорий

```
user@work:~$ ls chroot/bash/
bin lib
user@work:~$
```

Рисунок 9 – структура chroot

11) В результате схема на рисунке 6 изменится следующим образом:

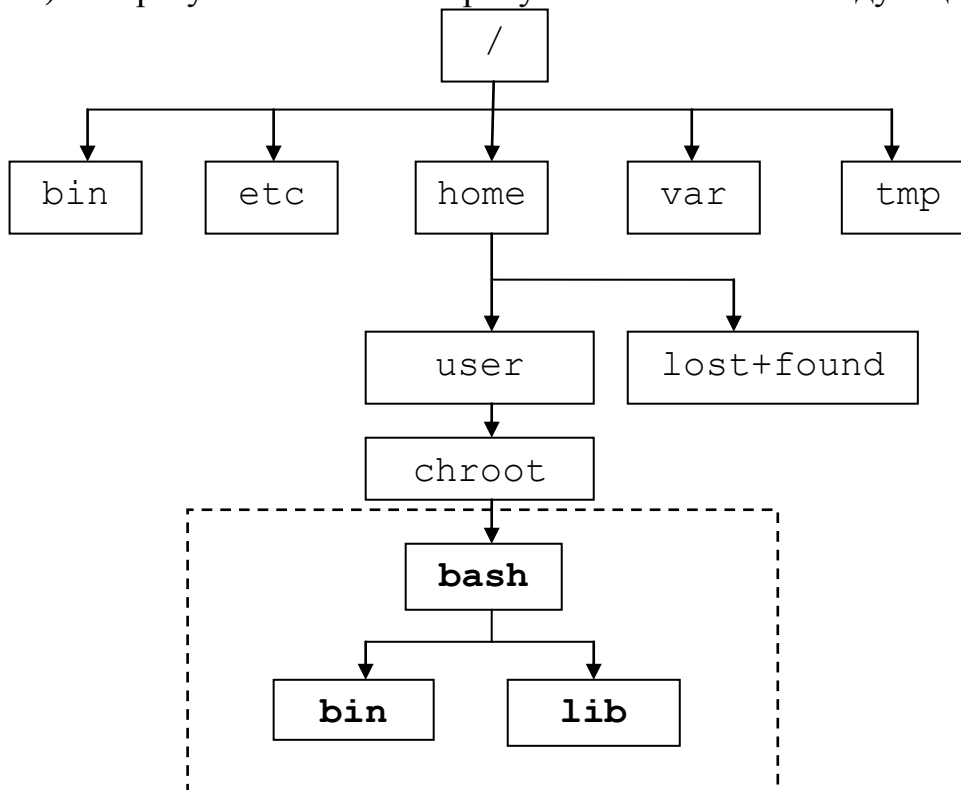


Рисунок 10 – структура chroot

12) Каталоги созданы, теперь вы можете скопировать файлы, используемые bash (/lib) в изолированный lib, и скопировать файл bash в изолированный каталог bin. Этим вы создадите ту же самую среду, но с ограниченными возможностями.

На этом этапе меняется структура. Ресурсы, необходимые для работы, перемещаются из реального каталога в директорию окружения.

Собственно произведите действие, показанное на рисунке 11. Скопируйте библиотеки, полученные в пункте 8 (рисунок 7). Для этого выполните:

```
$ sudo cp /lib/libncurses.so.5 /home/user/chroot/bash/lib/
$ sudo cp /lib/i686/cmov/libdl.so.2 /home/user/chroot/bash/lib
$ sudo cp /lib/i686/cmov/libc.so.6 /home/user/chroot/bash/lib
$ sudo cp /lib/ld-linux.so.2 /home/user/chroot/bash/lib
$ sudo cp /bin/bash /home/user/chroot/bash/bin/
```

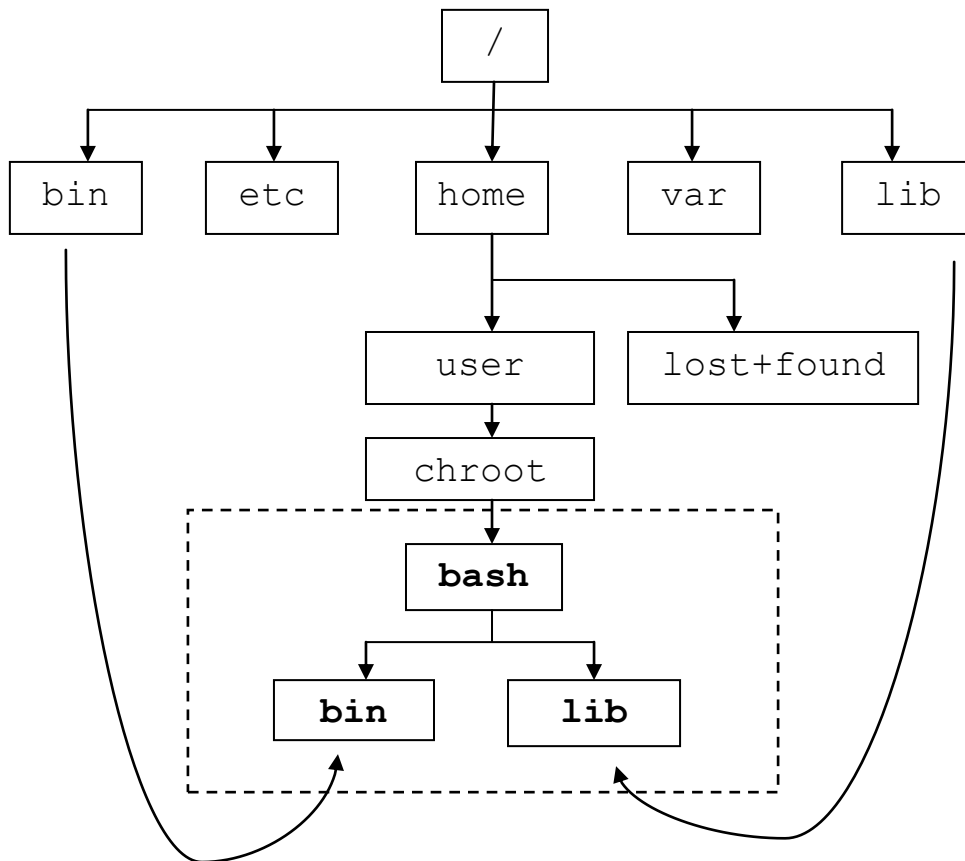


Рисунок 11 – структура chroot

Если все сделано верно, то вы увидите приглашение `bash-4.1#`. Это свидетельствует, что интерпретатор `bash` запущен успешно. Приглашение `#` означает, что вы работаете с правами суперпользователя.

14) Попробуйте вводить какие-либо команды: `cp`, `ls`. Вы увидите ошибки:

```

bash-4.1# ls
bash: ls: command not found
bash-4.1# cp
bash: cp: command not found
bash-4.1# █
  
```

Рисунок 14 – ошибки при работе

Зато работают команды `pwd` и `echo`. Выполните:

```

bash-4.1# pwd
bash-4.1# echo "KIBEVS"
bash-4.1# echo "KIBEVS"
KIBEVS
bash-4.1# pwd
/
bash-4.1# █
  
```

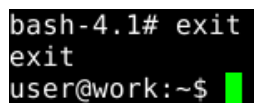
Рисунок 15 – команды `pwd` и `echo` работают

Внутри изолированной среды ничего, кроме `echo` и `pwd`, не работает. Это потому, что внутри среды `chroot` кроме `bash` нет никаких других команд, а `echo` является встроенной командой.

15) Обратите внимание на вывод команды `pwd` (рисунок 15). / показывает, что текущим является корневой каталог.

Выйдите из `chroot` окружения набрав `exit`:

```
bash-4.1# exit
```



```
bash-4.1# exit
exit
user@work:~$
```

Рисунок 16 – выход из `chroot` окружения

III. Недостатки

16) Механизм `chroot` не является краеугольным оплотом по запуску «недоверенных» программ. `Chroot` был скомпрометирован <http://www.bpfh.net/simes/computing/chroot-break.html>.

17) Отсутствие тонкой настройки `chroot`. Например, отсутствуют: дисковые квоты, ограничения памяти, изоляция сети, квоты ЦПУ.

18) Разобранные выше пример является классическим для демонстрации работы `chroot`. Если потребуется запуск в отдельном окружении более «сложной» программы (например `firefox`), то потребуется больше усилий, чем команда `ldd`. Возможно, необходимо осуществлять трассировку системных вызовов (`strace`), вызовов библиотек (`ltrace`) и т.д.