Лабораторная работа №3.
Защита загрузчика и управление доступом ОС Linux

Цель работы: Освоение методов защиты загрузчика GRUB, изучение концепций PAM и внедрение двухфакторной аутентификации для повышения безопасности системы.

1. Обеспечение безопасности GRUB

GRUB (GRand Unified Bootloader) — загрузчик ОС от проекта GNU. GRUB предоставляет пользователю интерактивное меню, в котором можно выбрать нужную ОС или ядро Linux для загрузки. Загрузчик может обнаруживать другие установленные ОС на компьютере и добавлять их в меню загрузки. Он также позволяет пользователю изменять параметры загрузки ОС, такие как передача параметров ядра, загрузка в безопасном режиме или изменение графического разрешения экрана.

Загрузчик GRUB установлен по умолчанию в ОС “Альт Рабочая станция”.

Рабочая конфигурация загрузчика GRUB содержится в файле */boot/grub/grub.cfg*.

```
# vim /boot/grub/grub.cfg
```

Важно! Редактировать файл */boot/grub/grub.cfg* нельзя, об этом есть предупреждение в самом файле.

После редактирования файлов необходимо переконфигурировать GRUB с помощью утилиты *grub-mkconfig* или *update-grub* на основании системы скриптов в каталоге */etc/grub.d* и редактируемой конфигурации GRUB */etc/default/grub*.

Каталог */etc/grub.d* содержит скрипты, которые используются при создании */boot/grub/grub.cfg*. При выполнении команды update-grub они находят все установленные на компьютере операционный системы и linux-ядра и формируют меню загрузки. Два основных из них — *10_linux* и *30_os-prober* — отвечают за поиск linux-ядер и остальных ОС на других разделах. Файл *40_custom* позволяет добавлять свои пункты в меню загрузки.

После редактирование файлов необходимо обновить их с помощью следующей команды:

```
# grub-mkconfig -o /boot/grub/grub.cfg
```

или

```
# update-grub
```

По умолчанию при загрузке любой пользователь может добавить/изменить параметры. Этого можно избежать, если установить пароль на загрузчик.

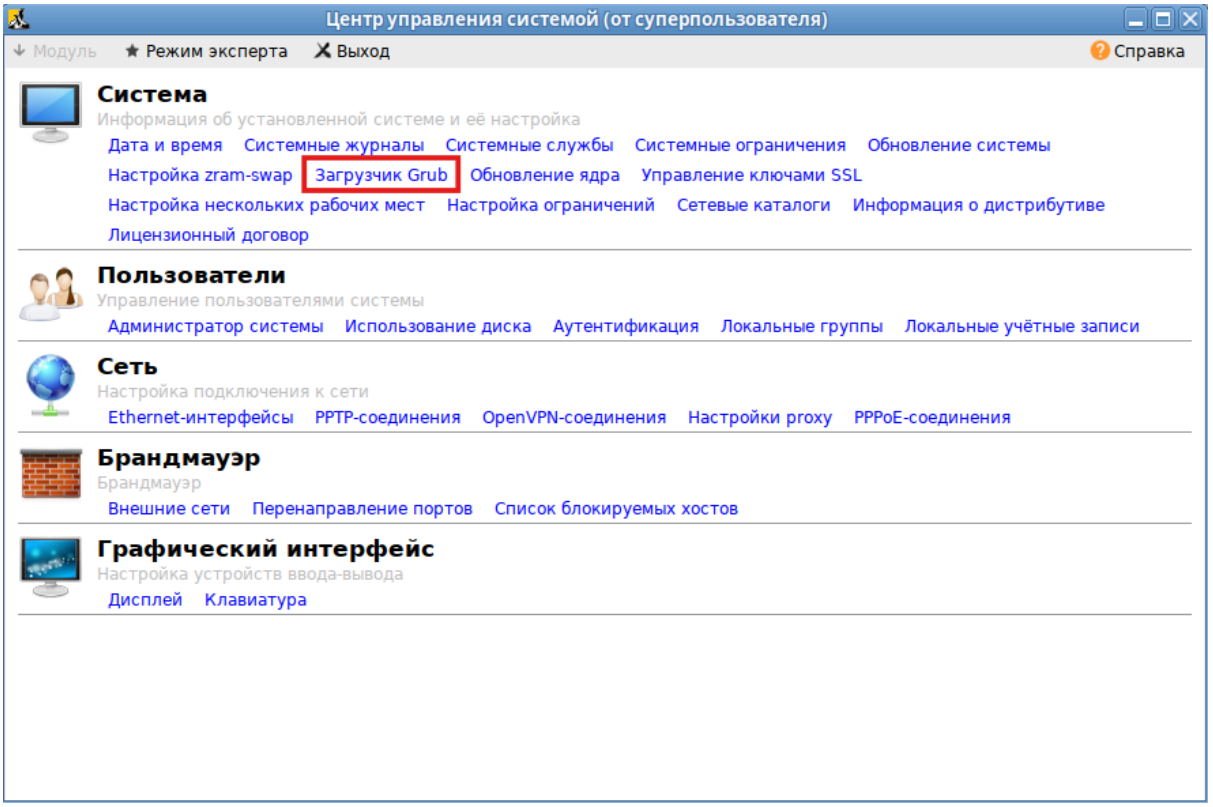
Установить пароль загрузчика можно следующими способами (выполняем **2-й способ**):

1. Через ЦУС (Центр управления системой).

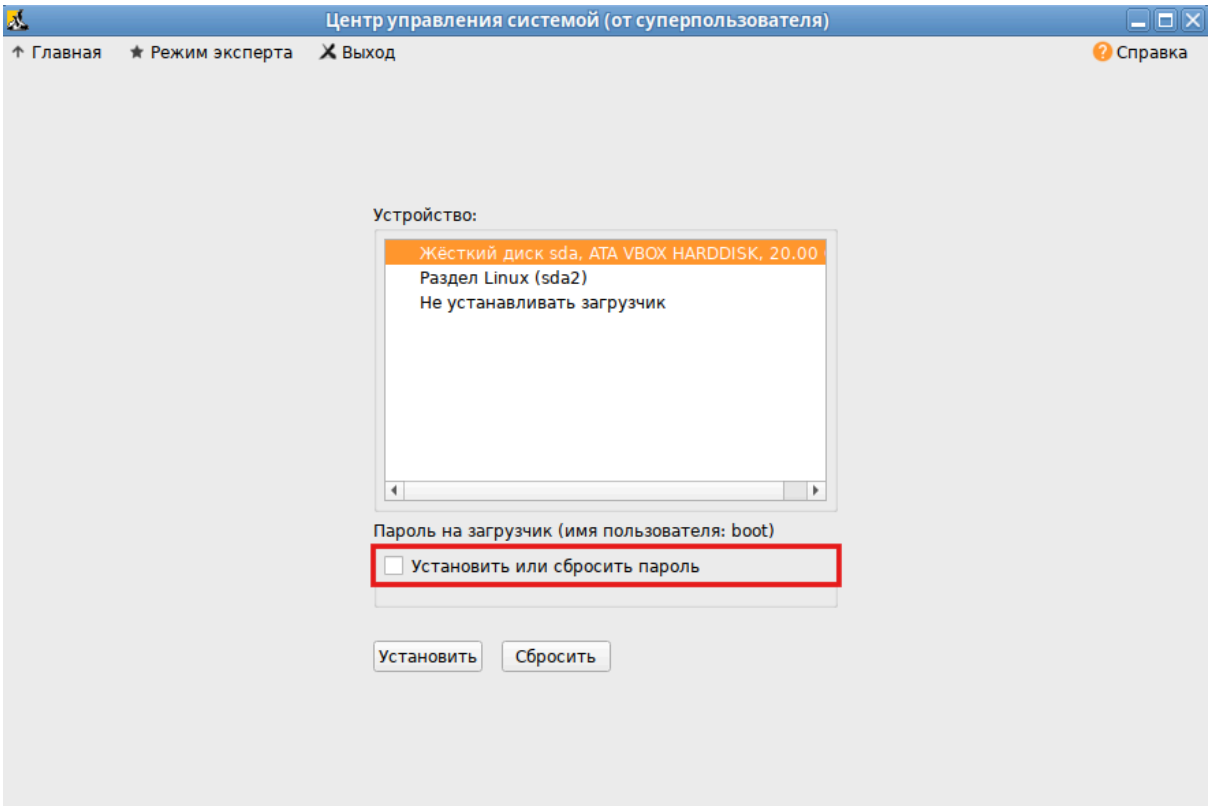
Зайдем под учетной записью суперпользователя и устанавливаем пакет *alterator-grub*:

```
# apt-get update
# apt-get install alterator-grub
```

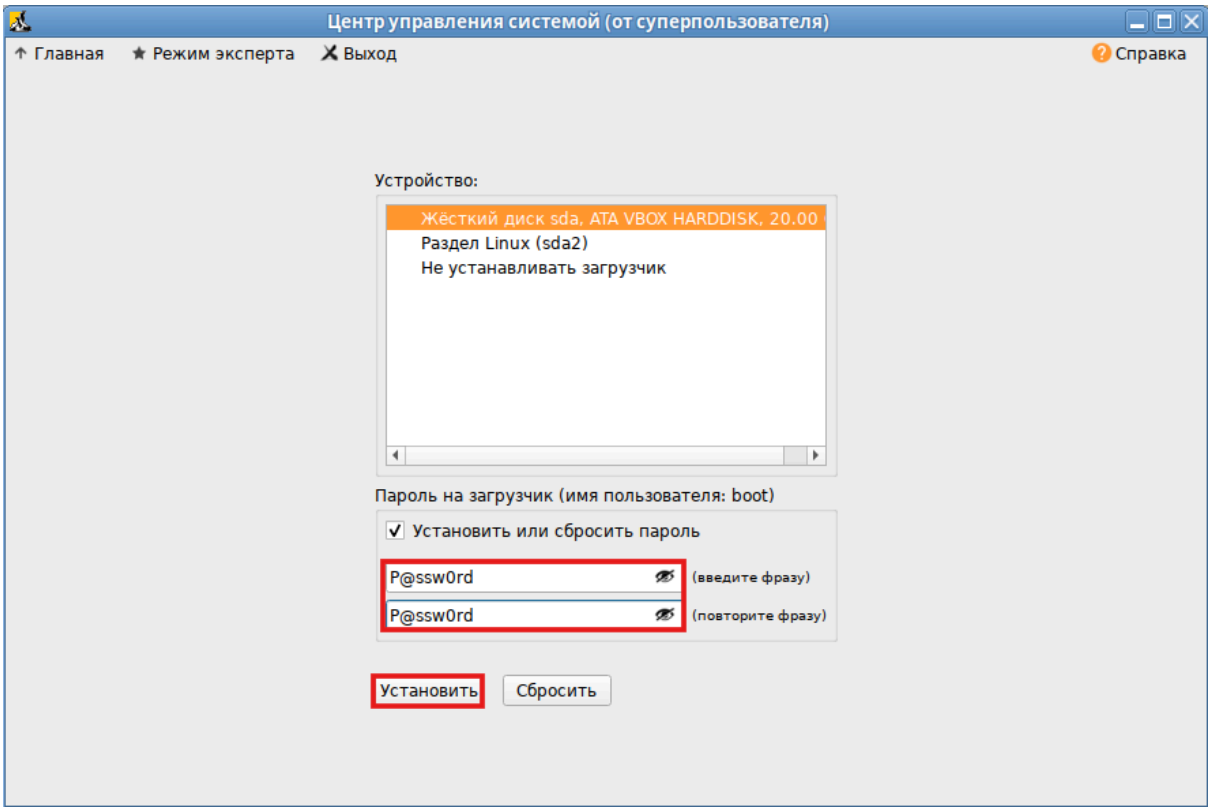
Зайдем в ЦУС и в разделе “Система” откроем “Загрузчик GRUB”:



Установим флажок “Установить или сбросить пароль”:



По умолчанию логин задан “boot”. Введите пароль “P@ssw0rd” и нажмите кнопку “Установить” (переконфигурирование GRUB происходит автоматически).



Теперь при изменении параметров загрузки (клавиша *e*) потребуется ввести имя пользователя “boot” и заданный пароль “P@ssw0rd”.

2. Через редактирование конфигураций.

Зайдем под учетной записью суперпользователя и с помощью утилиты **grub-mkpasswd-pbkdf2** сгенерируйте хеш пароля:

```
# grub-mkpasswd-pbkdf2
```

```
alt ~ # grub-mkpasswd-pbkdf2
Введите пароль:
Повторно введите пароль:
Хэш PBKDF2 вашего пароля: grub.pbkdf2.sha512.10000.6D255B4B6BF4EE6C20AA5DED1B45CC4703CB1
5EA85260882B3DD9ED10C4E126DE09E122B0AF7B6AF0C8DC18CE5C8F4E6B3FC6F2DE24EF2E9D42B3F0214B1E
0F9.E6A278A17D4B03D6BA3E87EAF5102B12985FE14C46C7D5B2E66681142F9A0D01FEC0303070AB9EE1625D
C66E859FAE053272C7A593BE8B99B150CC6871470317
```

Из выведенных данных нужно взять всю строку “grub.pbkdf2.sha512.1000…”.

Откройте файл /etc/grub.d/40_custom:

```
# vim /etc/grub.d/40_custom
```

Добавьте в него следующие строки:

```
set superusers=”boot”
password_pbkdf2 boot grub.pbkdf2.sha512.10000.6D255B4B6BF4EE6C20AA5DED1B45CC4703CB15EA85260882B3DD9ED10C4E126DE09E122B0AF7B6AF0C8DC18CE5C8F4E6B3FC6F2DE24EF2E9D42B3F0214B1E0F9.E6A278A17D4B03D6BA3E87EAF5102B12985FE14C46C7D5B2E66681142F9A0D01FEC0303070AB9EE1625DC66E859FAE053272C7A593BE8B99B150CC6871470317
```

Важно! Необходимо вставить хэш PBKDF2 **вашего** пароля.

Сохраняем файл *40_custom*. Поскольку файл /etc/grub.d/40_custom содержит хэш пароля, то рекомендуется запретить его чтение и изменение всеми, кроме суперпользователя:

```
# chmod 711 /etc/grub.d/40_custom
```

Теперь переконфигурируем загрузчик GRUB:

```
# grub-mkconfig -o /boot/grub/grub.cfg
```

```
alt ~ # grub-mkconfig -o /boot/grub/grub.cfg
Generating grub configuration file ...
Found theme: /boot/grub/themes/workstation/theme.txt
Found linux image: /boot/vmlinuz-std-def
skipping symlink: /boot/vmlinuz-std-def
Found linux image: /boot/vmlinuz
Found initrd image: /boot/initrd.img
Found linux image: /boot/vmlinuz-un-def
skipping symlink: /boot/vmlinuz-un-def
Found linux image: /boot/vmlinuz-5.15.89-un-def-alt1
Found initrd image: /boot/initrd-5.15.89-un-def-alt1.img
Found linux image: /boot/vmlinuz-5.10.164-std-def-alt1
Found initrd image: /boot/initrd-5.10.164-std-def-alt1.img
Found memtest image: /boot/memtest-5.31b.bin
done
```

После перезагрузки при попытке изменить пункт меню вам будет предложено ввести имя пользователя и пароль. Введите *boot* и пароль, который вы вводили в команде *grub-mkpasswd-pbkdf2*. Если учетные данные верны, система продолжит загрузку.

2. Параметры загрузки системы

Перед тем, как изменять параметры загрузки, выведем текущие параметры:

cat /proc/cmdline

```
BOOT_IMAGE=/boot/vmlinuz-un-def root=UUID=6aa03142-0031-4775-96a7-112d3147b228
ro resume=/dev/disk/by-uuid/2e12f157-1568-4552-afe6-7b5fef894125 panic=30 quiet
 loglevel=3 splash
```

BOOT_IMAGE=/boot/vmlinuz-un-def указывает путь к загружаемому образу ядра Linux;

root=UUID=<...> указывает, какой раздел будет использоваться в качестве корневой файловой системы;

ro означает, что корневая файловая система будет монтироваться в режиме "только для чтения" во время загрузки. Позже, после завершения загрузки, эта файловая система может быть смонтирована в режиме “чтения-записи”;

resume=/dev/disk/by-uuid/<...> указывает раздел для режима гибернации (resume), где хранятся данные о состоянии системы;

panic=30 устанавливает время (в секундах), через которое система перезагрузится после возникновения ошибки ядра (kernel panic);

quiet уменьшает количество сообщений, выводимых на экран во время загрузки, помогая сделать процесс загрузки более “тихим”;

loglevel=3 определяет уровень важности (уровень логирования) сообщений, которые будут выводиться на консоль во время загрузки системы.

splash указывает на использование графического загрузчика (Splash Screen), который замещает текстовые сообщения загрузки графическим интерфейсом.

Для *однократного* изменения параметров загрузки ядра при загрузке GRUB необходимо:

1. Нажать клавишу *e* при курсоре на нужном пункте загрузочного меню.
2. После ввода логина и пароля в открывшемся редакторе отыскать строку, начинающуюся с *linux /boot/vmlinuz*…
3. В конец этой строки дописать через пробел требуемые параметры.
4. Нажать клавишу *F10* или сочетание клавиш *Ctrl+X*.

Важно! Добавленные параметры будут актуальны до следующей перезагрузки системы.

Добавим параметр *net.ifnames=0* для использования традиционных имен сетевых интерфейсов. Т.е. вместо *ens19, ens20* будет *eth0, eth1* и т.д.

Перезагрузим систему:

```
# reboot
```

После перезагрузки нажмем клавишу *e*, чтобы изменить пункт меню *“ALT Workstation 10.2”*.

После ввода логина и пароля найдем строку, начинающуюся с *linux /boot/vmlinuz*, и добавим в конец параметр *net.ifnames=0*.

```
        linux        /boot/vmlinuz-un-def root=UUID=6aa03142-0031-4775-96a7\
-112d3147b228 ro  resume=/dev/disk/by-uuid/2e12f157-1568-4552-afe6-7b5fef89\
4125 panic=30 quiet loglevel=3 splash net.ifnames=0_
```

Нажмем клавишу *F10* или сочетание клавиш *Ctrl+X*.

Ждем окончания загрузки системы.

Затем зайдем под учетной записью пользователя *user* и в терминале выполним следующую команду:

```
$ ip a
```

```
[user@alt-1 ~]$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether fa:fe:8a:03:3d:51 brd ff:ff:ff:ff:ff:ff
    altname enp0s19
    altname ens19
    inet 10.40.28.233/22 brd 10.40.31.255 scope global dynamic noprefixroute eth0
       valid_lft 43180sec preferred_lft 43180sec
    inet6 fe80::6bf5:193c:bce9:6d14/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Убедимся, что в выводе имя сетевого интерфейса соответствует *eth0*.

Для *постоянного* изменения параметров загрузки необходимо изменить редактируемую конфигурацию загрузчика */etc/default/grub*.

Основные параметры загрузки:

GRUB_DEFAULT указывает какой пункт нужно загружать по умолчанию. Может быть указан номер, полное название или же строка saved, которая значит, что нужно загрузить пункт, указанный с помощью grub-reboot;

GRUB_SAVEDEFAULT загружает последнюю использованную запись по умолчанию;

GRUB_TIMEOUT показывает, сколько секунд будет показано загрузочное меню;

GRUB_CMDLINE_LINUX добавляет опции ядра для всех ядер, как обычных, так и режима восстановления;

GRUB_CMDLINE_LINUX_DEFAULT добавляет опции ядра только для обычных ядер;

GRUB_TERMINAL_OUTPUT указывает на модуль терминала для GRUB. Можно использовать console, только для текстового режима или gfxterm с поддержкой графики;

Установим время загрузки меню GRUB на 20 секунд.

Для этого отредактируем файл */etc/default/grub*:

```
# vim /etc/default/grub
```

Изменим следующую строку:

```
GRUB_DEFAULT=20
```

После редактирование переконфигурируем загрузчик GRUB:

```
# grub-mkconfig -o /boot/grub/grub.cfg
```

Перезагружаем систему и проверяем применение настроек:

```
# reboot
```

3. Изучение концепции PAM

PAM (Pluggable Authentication Modules — подключаемые модули аутентификации) — система библиотек, которые выполняют задачи аутентификации приложений (служб) в системе. PAM предоставляет гибкий и централизованный способ переключения методов аутентификации для защищенных приложений с помощью конфигурационных файлов вместо изменения кода приложения.

PAM используется везде, где требуется аутентификация пользователя или проверка его прав.

Компоненты, входящие в состав PAM:

1. Библиотеки PAM. Каждый метод аутентификации представлен динамически загружаемой библиотекой. Библиотеки PAM находятся в каталоге */lib64/security*.

Выведем список библиотек PAM:

```
# ls /lib64/security
```

```
[root@alt ~]# ls /lib64/security/
pam_access.so              pam_keyinit.so        pam_pwdb.so          pam_time.so
pam_canonicalize_user.so   pam_krb5.so           pam_pwhistory.so     pam_timestamp.so
pam_ccreds.so              pam_lastlog.so        pam_pwquality.so     pam_tty_audit.so
pam_console.so             pam_limits.so         pam_rhosts.so        pam_umask.so
pam_debug.so               pam_listfile.so       pam_rootok.so        pam_unix_acct.so
pam_deny.so                pam_localuser.so      pam_sameuid.so       pam_unix_auth.so
pam_echo.so                pam_loginuid.so       pam_securetty.so     pam_unix_passwd.so
pam_env.so                 pam_mail.so           pam_selinux.so       pam_unix_session.so
pam_exec.so                pam_mkhomedir.so      pam_sepermit.so      pam_unix.so
pam_faildelay.so           pam_mktemp.so         pam_setquota.so      pam_userdb.so
pam_faillock.so            pam_motd.so           pam_shells.so        pam_userpass.so
pam_filter                 pam_mount.so          pam_sss_gss.so       pam_usertype.so
pam_filter.so              pam_namespace.so      pam_sss.so           pam_vbox.so
pam_ftp.so                 pam_nologin.so        pam_stress.so        pam_warn.so
pam_gnome_keyring.so       pam_passwdqc.so       pam_succeed_if.so    pam_wheel.so
pam_group.so               pam_permit.so         pam_systemd.so       pam_winbind.so
pam_issue.so               pam_propperpwnam.so   pam_tcb.so           pam_xauth.so
```

2. Конфигурационные файлы PAM. Настройки для различных служб и приложений определены в конфигурационных файлах, которые находятся в каталоге */etc/pam.d*. Каждый файл соответствует определенной службе (например, для входа в систему, sudo и т.д.) и содержит информацию о том, какие модули PAM должны использоваться для аутентификации.

Выведем конфигурационные файлы PAM:

ls /etc/pam.d

```
[root@alt ~]# ls /etc/pam.d
acc                          packageinstall              system-auth-use_first_pass
alterator-chkpwd             passwd                      system-auth-use_first_pass-krb5
alterator-standalone         polkit-1                    system-auth-use_first_pass-krb5_ccreds
appinstall                   ppp                         system-auth-use_first_pass-krb5_ccreds-only
beesu                        roleadd                     system-auth-use_first_pass-krb5-only
chage                        roledel                     system-auth-use_first_pass-ldap
chage-chfn-chsh              runuser                     system-auth-use_first_pass-ldap-only
chfn                         runuser-l                   system-auth-use_first_pass-local
chpasswd                     screen                      system-auth-use_first_pass-local-only
chpasswd-newusers            sshd                        system-auth-use_first_pass-multi
chsh                         sssd-shadowutils            system-auth-use_first_pass-pkcs11
common-login                 su                          system-auth-use_first_pass-sss
common-login-use_first_pass  synaptic                    system-auth-use_first_pass-sss-only
config-util                  system-auth                 system-auth-use_first_pass-winbind
crond                        system-auth-common          system-auth-use_first_pass-winbind-only
cups                         system-auth-krb5            system-auth-winbind
groupadd                     system-auth-krb5_ccreds     system-auth-winbind-only
groupdel                     system-auth-krb5_ccreds-only systemd-user
groupmems                    system-auth-krb5-only       system-policy
groupmod                     system-auth-ldap            system-policy-local
lightdm                      system-auth-ldap-only       system-policy-remote
lightdm-autologin            system-auth-local           useradd
lightdm-greeter              system-auth-local-only      userdel
login                        system-auth-multi           user-group-mod
mate-screensaver             system-auth-pkcs11          usermod
newusers                     system-auth-sss             vmtoolsd
other                        system-auth-sss-only        xserver
```

3. Дополнительные конфигурационные файлы PAM. Каждый дополнительный файл конфигурации определяет конкретные настройки для определенной группы методов аутентификации. Такой подход позволяет гибко настраивать систему в зависимости от требований и нужд различных приложений и служб.

Выведем дополнительные конфигурационные файлы PAM:

ls /etc/security

```
[root@alt ~]# ls /etc/security
access.conf       console.perms.d  limits.d        pam_env.conf        pwquality.conf
console.apps      faillock.conf    namespace.conf  pam_mount.conf.xml  sepermit.conf
console.handlers  group.conf       namespace.d     pam_winbind.conf    time.conf
console.perms     limits.conf      namespace.init  pwhistory.conf
```

Жизненный цикл ПО:

1. Запуск приложения. Процесс начинается, когда пользователь пытается войти в систему или выполнить команду, требующую аутентификации.

2. Запуск PAM. Когда приложение запускается, оно вызывает библиотеку PAM для аутентификации пользователя.

3. Выбор стека модулей. PAM выбирает набор модулей аутентификации, которые будут использоваться для проверки подлинности пользователя.

4. Выполнение модулей аутентификации. PAM последовательно вызывает выбранные модули аутентификации. Каждый модуль выполняет определенную проверку, например, проверку пароля, проверку биометрических данных и т.д. Если какой-либо модуль не проходит проверку, аутентификация считается неуспешной, и управление возвращается приложению.

5. Управление потоком выполнения. PAM имеет возможность настроить стратегию аутентификации, например, требовать успешного прохождения всех модулей или разрешать аутентификацию после первого успешного прохождения модуля.

6. Успешная аутентификация. Если все модули аутентификации успешно завершены, PAM возвращает управление приложению с сообщением об успешной аутентификации.

7. Завершение работы PAM. После завершения аутентификации PAM освобождает ресурсы и завершает свою работу.

В системе PAM модули классифицируются по 4-м категориям, каждая из которых выполняет определенную роль в процессе аутентификации и управления пользователями.

Таблица 1 — Модули PAM

Модуль	Описание
auth	Аутентификация пользователя
account	Управление учетной записью
password	Управление паролем
session	Управление сеансом

В системе PAM также используются различные флаги, которые определяют, как модули будут обрабатываться в процессе аутентификации и управления пользователями.

Таблица 2 — Флаги PAM

Флаг	Описание
required	Указанный модуль должен успешно отработать. Остальные модули будут запущены при неудаче, исполнение продолжится по конфигурационному файлу.
requisite	Указанный модуль должен успешно отработать. Остальные модули будут запущены при неудаче, исполнение по конфигурационному файлу тут же прекратится.
sufficient	Если указанный модель отработает успешно, весь сервис будет считаться доступным. При неудаче этого модуля будут выполняться следующие, стоящие в конфигурационном файле после него.
optional	Результат модуля не имеет значения, если этот модуль единственный.
include	Подключить содержимое другого конфигурационного файла PAM.

4. Анализ текущих настроек PAM

Перед внесением изменений в конфигурацию, следует создать резервную копию, чтобы затем при необходимости можно было вернуться к прежним настройкам.

Создадим резервную копию всех файлов конфигурации:

```
# cd /etc
# mkdir pam.backup
# cp pam.d/* pam.backup
# ls -ld pam.backup
```

Изучим содержимое конфигурационных файлов каталога */etc/pam.d*.

Каждая строка файла состоит из нескольких полей:

тип обязательность модуль параметры

Первое поле определяет, какой тип запроса к PAM надо выполнить. Существует четыре различных типа запроса: *auth, account, password* и *session*.

Второе поле определяет, как нужно интерпретировать результат, возвращенный модулем PAM. Здесь используют флаги PAM.

Третье и четвертое поле — название библиотеки модуля и её параметры.

Изучим конфигурационный файл sshd:

cat /etc/pam.d/sshd

```
[root@alt ~]# cat /etc/pam.d/sshd
#%PAM-1.0
auth            required        pam_userpass.so
auth            include         common-login-use_first_pass
account         include         common-login
password        include         common-login
session         include         common-login
```

Модуль pam_userpass.so отвечает за аутентификацию пользователя. Требуется для успешного входа в систему и проверяет введенные имя пользователя и пароль.

Строка “auth include common-login-use_first_pass” включает другие модули аутентификации, определенные в файле *common-login*. Параметр *use_first_pass* указывает PAM использовать пароль, введенный пользователем ранее в том же процессе аутентификации.

Строка “account include common-login” включает модули управления учетными записями из файла *common-login*. Эти модули выполняют различные проверки, связанные с учетной записью пользователя, такие как блокировка учетной записи или проверка срока действия пароля.

Строка “password include common-login” включает функции управления паролями из файла *common-login*. Она отвечает за изменение пароля пользователя и за проверку его правильности.

Строка “session include common-login” включает модули, отвечающие за управление сеансом пользователя после успешной аутентификации. Здесь могут выполняться настройки окружения и регистрации сеансов, чтобы обеспечить корректную работу системы для пользователя.

Теперь посмотрим конфигурационный файл *common-login*:

cat /etc/pam.d/common-login

```
[root@alt ~]# cat /etc/pam.d/common-login
#%PAM-1.0
auth            substack        system-auth
auth            substack        system-policy
auth            required        pam_nologin.so
account         substack        system-auth
account         substack        system-policy
account         required        pam_nologin.so
password        include         system-auth
password        include         system-policy
session         substack        system-auth
session         required        pam_loginuid.so
-session        optional        pam_systemd.so
session         substack        system-policy
```

Файл *common-login* определяет набор модулей для аутентификации, управления учетными записями и сеансами, а также условные проверки для управления доступом пользователей.

4. Настройка PAM для повышения уровня безопасности

4.1. Ограничение доступа к системе для определенных групп пользователей или с определенных IP-адресов с помощью модуля pam_access.so и файла /etc/security/access.conf.

Модуль *pam_access.so* обеспечивает управление входом в систему. Этот модуль может использоваться для принятия решения о том, каким пользователям разрешен вход в систему. Т.к. PAM имеет средства аутентификации по сети, то контролируется не только кто может или не может зайти, но и откуда.

По умолчанию правила управления доступом берутся из файла конфигурации */etc/security/access.conf*.

Формат файла /etc/security/access.conf:

permission:users:origins

permission – знак «+» (плюс) – предоставление доступа или знак «-» (минус) – отказ в доступе.

users – список пользователей или групп пользователей или ключевое слово *ALL*.

origins – список TTY (для локального доступа), имен хостов, доменных имен, IP-адресов, ключевого слова *ALL* или *LOCAL*.

Создадим группу *test*:

```
# groupadd test
```

Создадим пользователя *testuser* с паролем *P@ssw0rd* и добавим его в группе *test*:

```
# useradd testuser
# passwd testuser
# usermod -aG test testuser
```

Отобразим информацию о пользователе *testuser*:

```
# id testuser
```

```
[root@alt ~]# id testuser
uid=501(testuser) gid=502(testuser) группы=502(testuser),501(test)
```

Добавим в файл */etc/security/access.conf* ограничение доступа к системе для группы пользователей *test* с локального хоста:

```
-:test:127.0.0.0/24
```

Далее необходимо сконфигурировать стек PAM для использования модуля *pam_access.so* для ограничения доступа на основе ограничений, определенных в файле */etc/security/access.conf*. Для этого допишем в файл */etc/pam.d/system-auth-local-only* следующую строку:

```
account required pam_access.so
```

```
#%PAM-1.0
auth            required        pam_tcb.so shadow fork nullok
account         required        pam_tcb.so shadow fork
account         required        pam_access.so
password        required        pam_passwdqc.so config=/etc/passwdqc.conf
password        required        pam_tcb.so use_authtok shadow fork nullok write_to=tcb
session         required        pam_tcb.so
```

Важно! Обратите внимание на порядок строк в конфигурационных файлах PAM.

Теперь при попытке зайти под учетной записью пользователя *testuser* видим следующую ошибку:

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

Также попробуем подключиться к пользователю *testuser* по SSH:

ssh testuser@127.0.0.1

```
[root@alt ~]# ssh testuser@127.0.0.1
testuser@127.0.0.1's password:
Connection closed by 127.0.0.1 port 22
```

Важно! Предварительно необходимо проверить статус службы sshd:

systemctl status sshd

Если служба не запущена, необходимо запустить ее:

systemctl enable --now sshd

После попытки подключения посмотрим логи в журнале службы:

journalctl -xeu sshd.service

```
alt sshd[3744]: pam_tcb(sshd:auth): Authentication passed for testuser from (uid=0)
alt sshd[3744]: pam_access(sshd:account): access denied for user `testuser' from `127.0.0.1'
alt sshd[3744]: Failed password for testuser from 127.0.0.1 port 60420 ssh2
alt sshd[3744]: fatal: Access denied for user testuser by PAM account configuration [preauth]
```

4.2. Сложность пароля с использованием модуля pam_passwdqc.so.

Файл */etc/passwdqc.conf* состоит из 0 или более строк следующего формата:

опция=значение

Символы пробела между опцией и значением не допускаются.

Используемые типы паролей по классам символов (алфавит, число, спецсимвол, верхний и нижний регистр) определяются следующим образом:

1. Тип *N0* используется для паролей, состоящих из символов только одного класса.

2. Тип *N1* используется для паролей, состоящих из символов двух классов.

3. Тип *N2* используется для парольных фраз, кроме этого требования длины, парольная фраза должна также состоять из достаточного количества слов.

4. Типы *N3* и *N4* используются для паролей, состоящих из символов трех и четырех классов, соответственно.

Ключевое слово *disabled* используется для запрета паролей выбранного типа N0-N4 независимо от их длины.

Каждое следующее число в настройке “min” должно быть не больше, чем предыдущее.

Основные опции файла */etc/passwdqc.conf*:

1. max=N — максимально допустимая длина пароля. Эта опция может быть использована для того, чтобы запретить пользователям устанавливать пароли, которые могут быть слишком длинными для некоторых системных служб. Значение 8 обрабатывается особым образом: пароли длиннее 8 символов, не отклоняются, а обрезаются до 8 символов для проверки надежности (пользователь при этом предупреждается).

2. passphrase=N — число слов, необходимых для ключевой фразы (значение 0 отключает поддержку парольных фраз).

3. match=N — длина общей подстроки, необходимой для вывода, что пароль хотя бы частично основан на информации, найденной в символьной строке (значение 0 отключает поиск подстроки). Если найдена слабая подстрока пароль не будет отклонен; вместо этого он будет подвергаться обычным требованиям к прочности при удалении слабой подстроки. Поиск подстроки нечувствителен к регистру и может обнаружить и удалить общую подстроку, написанную в обратном направлении.

4. similar=permit|deny — параметр similar=permit разрешает задать новый пароль, если он похож на старый (параметр similar=deny — запрещает). Пароли считаются похожими, если есть достаточно длинная общая подстрока, и при этом новый пароль с частично удаленной подстрокой будет слабым.

5. random=N[,only] — размер случайно сгенерированных парольных фраз в битах (от 26 до 81) или 0, чтобы отключить эту функцию. Любая парольная фраза, которая содержит предложенную случайно сгенерированную строку, будет разрешена вне зависимости от других возможных ограничений. Значение only используется для запрета выбранных пользователем паролей.

6. enforce=none|users|everyone — параметр enforce=users задает ограничение задания паролей в passwd на пользователей без полномочий root. Параметр enforce=everyone задает ограничение задания паролей в passwd и на пользователей, и на суперпользователя root. При значении none модуль PAM будет только предупреждать о слабых паролях.

7. retry=N — количество запросов нового пароля, если пользователь с первого раза не сможет ввести достаточно надежный пароль и повторить его ввод.

Установим следующие требования к паролю:

1. Пароль должен содержать не менее 8 символов.
2. Пароль должен содержать символы из 4-х разных классов.
3. Ограничение задания паролей с помощью утилиты *passwd* распространялось и на пользователей, и на суперпользователя root.
4. Количество запросов нового пароля не больше 2-х.

Отредактируем следующие строки в файле */etc/passwdqc.conf*:

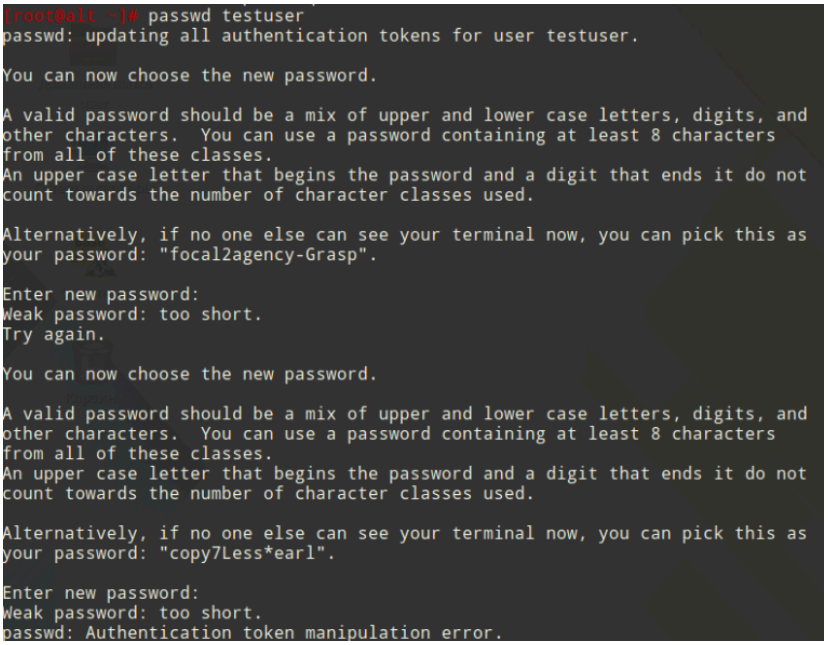
```
min=disabled,disabled,disabled,disabled,8
enforce=everyone
retry=2
```

Теперь попробуем сменить пароль пользователя *testuser*.

Текущий пароль: *P@ssw0rd*

Новый пароль: *resu123*

```
# passwd testuser
```



Мы не сможем установить пароль *resu123* для пользователя *testuser*, поскольку он не соответствует критериям.

4.3. Настройка модуля pam_failock.so

Модуль *pam_faillock* блокирует возможность аутентификации пользователя, на основании заранее определенного количества неудачных попыток входа. Настройка модуля может производиться через редактирование файла */etc/security/faillock.conf*.

Добавим в конец файла */etc/security/faillock.conf* следующие строки:

```
deny=3
unlock_time=600
```

deny – количество неудачных попыток входа, после которых возможность аутентификации будет заблокирована (по умолчанию 3).

unlock_time – интервал времени (в секундах), в течении которого возможность аутентификации для пользователя, превысившего количество попыток входа, будет заблокирована (по умолчанию 600 – 10 минут).

Далее включаем данный модуль в */etc/pam.d/system-auth-local-only*:

```
#%PAM-1.0
auth            requisite               pam_faillock.so preauth deny=3 unlock_time=600
auth            [success=1 default=bad] pam_tcb.so shadow fork nullok
auth            [default=die]           pam_faillock.so authfail deny=3 unlock_time=600
auth            sufficient              pam_faillock.so aurhfail deny=3 unlock_time=600
account         required                pam_tcb.so shadow fork
account         required                pam_access.so
password        required                pam_passwdqc.so config=/etc/passwdqc.conf
password        required                pam_tcb.so use_authtok shadow fork nullok write_to=tcb
session         required                pam_tcb.so
```

В качестве проверки делаем 3 ошибки при вводе пароля пользователя *testuser*:

```
user@alt ~ $ su - testuser
Password:
su: Authentication failure
user@alt ~ $ su - testuser
Password:
su: Authentication failure
user@alt ~ $ su - testuser
Password:
su: Authentication failure
user@alt ~ $ su - testuser
The account is locked due to 3 failed logins.
(10 minutes left to unlock)
su: Authentication failure
```

С помощью утилиты **faillock** проверим содержимое файлов записей об ошибках аутентификации пользователя testuser:

```
# faillock --user testuser
```

```
alt ~ # faillock --user testuser
testuser:
When                Type  Source                                           Valid
2024-09-24 00:20:50 RHOST localhost                                            V
2024-09-24 00:20:57 RHOST localhost                                            V
2024-09-24 00:21:02 RHOST localhost                                            V
```

4.4. Двухфакторная аутентификация

Настроим двухфакторную аутентификацию (2FA) для SSH входа на Linux сервер с помощью Google PAM (Pluggable Authentication Module) и мобильного приложения Google Authenticator. 2FA позволяет добавить дополнительный слой безопасности при аутентификации на Linux хосте по SSH. Теперь для входа на сервер кроме имени и пароля пользователя, необходимо ввести одноразовый цифровой пароль (Time-based One-time Password — TOTP), который будет генерироваться.

Установим пакет *Google PAM Authenticator*:

```
# apt-get update
# apt-get install libpam-google-authenticator
```

Далее выполним следующую команду:

```
# google-authenticator
```

Важно! Для отображения QR-кода необходимо установить библиотеку *libqrencode*:

```
# apt-get install libqrencode*
```

Утилита *google-authenticator* сгенерирует и отобразит в консоли QR-код.

Запустим приложение *Google Authenticator* на смартфоне. Выберите “Сканировать QR-код”. В результате в приложении появится новая запись для пользователя и сервера. В этой записи получаем одноразовый пароль для подключения к хосту.

По умолчанию одноразовый токен меняется раз в 30 секунд.

Теперь отредактируем файл */etc/pam.d/sshd*:

```
#%PAM-1.0
auth            include         system-auth-local
auth            [success=1 default=ignore]      pam_google_authenticator.so echo_verification_code
auth            include         pam_userpass.so
auth            include         common-login-use_first_pass
account         include         common-login
password        include         common-login
session         include         common-login
```

Также изменяем следующие параметры в файле */etc/openssh/sshd_config*:

```
PermitRootLogin yes
ChallengeResponseAuthentication yes
```

Перезапускаем службу *sshd*:

```
# systemctl restart sshd
```

Зайдем под учетной записью *user*.

Далее попробуем подключиться к пользователю *root* на локальном хосте:

$ ssh root@127.0.0.1