

Лабораторная работа №6

Мандатное разграничение прав в Linux

Ильин Андрей Владимирович

Содержание

1	Цель работы	4
2	Задачи	5
3	Теоретическое введение	6
3.1	Термины	6
3.2	Окружение	6
4	Выполнение лабораторной работы	8
5	Анализ результатов	19
6	Выводы	20
	Список литературы	21

Список иллюстраций

4.1	Проверка SELinux	8
4.2	Статус веб-сервера	9
4.3	Контекст безопасности Apache	9
4.4	Переключатели SELinux для Apache	10
4.5	Статистика по политике	11
4.6	Тип файлов и поддиректорий Apache	11
4.7	Создание html.test	12
4.8	Просмотр http://127.0.0.1/test.html (1)	13
4.9	Изменение контекст файла test.html	14
4.10	Просмотр http://127.0.0.1/test.html (2)	14
4.11	Просмотр audit.log	15
4.12	Изменение httpd.conf на прослушивание TCP-порта 81	16
4.13	Перезапуск Apache	16
4.14	Добавление порта 81 в список портов	17
4.15	Просмотр http://127.0.0.1/test.html (3)	18

1 Цель работы

Развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux. Проверить работу SELinux на практике совместно с веб-сервером Apache.

2 Задачи

1. Настроить и запустить сервер Apache.
2. Исследовать влияние различных параметров на работу сервера.

3 Теоретическое введение

3.1 Термины

- Терминал (или «Bash», сокращение от «Bourne-Again shell») — это программа, которая используется для взаимодействия с командной оболочкой. Терминал применяется для выполнения административных задач, например: установку пакетов, действия с файлами и управление пользователями. [1]
- Права доступа определяют, какие действия конкретный пользователь может или не может совершать с определенными файлами и каталогами. [2]
- Расширенные атрибуты файловых объектов (далее - расширенные атрибуты) - поддерживаемая некоторыми файловыми системами возможность ассоциировать с файловыми объектами произвольные метаданные. [3]
- Security Enhanced Linux (SELinux) – это система контроля доступа, которая в настоящее время встраивается в большинство Linux-дистрибутивов. [4]

3.2 Окружение

- Rocky Linux - это корпоративная операционная система с открытым исходным кодом, разработанная таким образом, чтобы быть на 100% совместимой с Red Hat Enterprise Linux. Он находится в стадии интенсивной разработки сообществом. [5]

- Git - это распределенное программное обеспечение для контроля версиями. [6]
- VirtualBox - это кросс-платформенное ПО для виртуализации x86 и AMD64/Intel64 с открытым кодом для корпоративного и домашнего использования. [7]

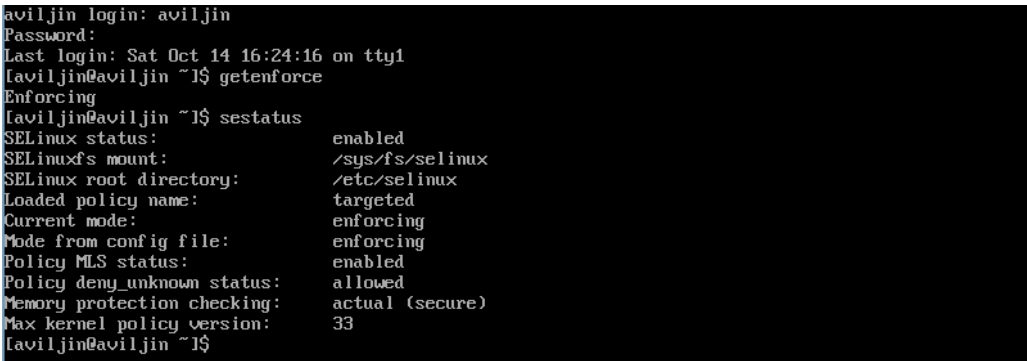
4 Выполнение лабораторной работы

1. Убедимся, что SELinux работает в режиме enforcing политики targeted.

(рис. 4.1)

```
getenforce
```

```
sestatus
```

A terminal window with a black background and white text. The text shows the output of the 'sestatus' command. It starts with a login prompt 'aviljin login: aviljin', followed by a password prompt 'Password:', then a last login message 'Last login: Sat Oct 14 16:24:16 on tty1'. The user 'aviljin@aviljin' enters '~' and runs 'getenforce', which returns 'Enforcing'. Then the user runs 'sestatus', which displays a list of SELinux parameters and their values: SELinux status: enabled, SELinuxfs mount: /sys/fs/selinux, SELinux root directory: /etc/selinux, Loaded policy name: targeted, Current mode: enforcing, Mode from config file: enforcing, Policy MLS status: enabled, Policy deny_unknown status: allowed, Memory protection checking: actual (secure), and Max kernel policy version: 33.

```
aviljin login: aviljin
Password:
Last login: Sat Oct 14 16:24:16 on tty1
aviljin@aviljin ~]$ getenforce
Enforcing
aviljin@aviljin ~]$ sestatus
SELinux status:                enabled
SELinuxfs mount:                /sys/fs/selinux
SELinux root directory:         /etc/selinux
Loaded policy name:              targeted
Current mode:                    enforcing
Mode from config file:           enforcing
Policy MLS status:               enabled
Policy deny_unknown status:      allowed
Memory protection checking:      actual (secure)
Max kernel policy version:       33
aviljin@aviljin ~]$
```

Рис. 4.1: Проверка SELinux

2. Запустим веб-сервер. Убедимся, что он работает. (рис. 4.2)

```
service httpd start
```

```
service httpd status
```



```

[aviljin@aviljin ~]$ sudo service httpd start
[sudo] password for aviljin:
Redirecting to /bin/systemctl start httpd.service
[aviljin@aviljin ~]$ service httpd status
Redirecting to /bin/systemctl status httpd.service
■ httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: active (running) since Sat 2023-10-14 16:39:13 MSK; 11s ago
     Docs: man:httpd.service(8)
   Main PID: 1390 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0 B/s"
      Tasks: 213 (limit: 17484)
    Memory: 37.0M
       CPU: 55ms
    CGroup: /system.slice/httpd.service
            └─1390 /usr/sbin/httpd -DFOREGROUND
              └─1391 /usr/sbin/httpd -DFOREGROUND
                └─1392 /usr/sbin/httpd -DFOREGROUND
                  └─1393 /usr/sbin/httpd -DFOREGROUND
                    └─1395 /usr/sbin/httpd -DFOREGROUND

Oct 14 16:39:13 aviljin systemd[1]: Starting The Apache HTTP Server...
Oct 14 16:39:13 aviljin systemd[1]: Started The Apache HTTP Server.
Oct 14 16:39:13 aviljin httpd[1390]: Server configured, listening on: port 80

```

Рис. 4.2: Статус веб-сервера

3. Найдем веб-сервер Apache в списке процессов, определим его контекст безопасности. (рис. 4.3)

`ps auxZ | grep httpd`

```

[aviljin@aviljin ~]$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root      1390  0.0  0.4 20168 11416 ?        Ss   16:39   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  1391  0.0  0.2 21600  7120 ?        S    16:39   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  1392  0.0  0.3 1538124 10864 ?        S1   16:39   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  1393  0.0  0.5 1669260 17000 ?        S1   16:39   0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache  1395  0.0  0.5 1538124 17000 ?        S1   16:39   0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-c0:c0.c1023 aviljin 1612 0.0  0.0 3876 2024 tty1 S+  16:40   0:00 grep --color=auto httpd
[aviljin@aviljin ~]$ _

```

Рис. 4.3: Контекст безопасности Apache

4. Посмотрим текущее состояние переключателей SELinux для Apache. (рис. 4.4)

`sestatus -b | grep httpd`

```

httpd_can_network_connect      off
httpd_can_network_connect_cobbler  off
httpd_can_network_connect_db    off
httpd_can_network_memcache     off
httpd_can_network_relay        off
httpd_can_sendmail              off
httpd_dbus_avahi                off
httpd_dbus_sssd                 off
httpd_dontaudit_search_dirs    off
httpd_enable_cgi                on
httpd_enable_ftp_server        off
httpd_enable_homedirs          off
httpd_execmem                   off
httpd_graceful_shutdown        off
httpd_manage_ipa                off
httpd_mod_auth_ntlm_winbind    off
httpd_mod_auth_pam              off
httpd_read_user_content        off
httpd_run_ipa                   off
httpd_run_preupgrade            off
httpd_run_stickshift            off
httpd_serve_cobbler_files       off
httpd_setrlimit                 off
httpd_ssi_exec                  off
httpd_sys_script_anon_write    off
httpd_tmp_exec                  off
httpd_tty_comm                  off
httpd_unified                   off
httpd_use_cifs                  off
httpd_use_fusefs                off
httpd_use_gpg                   off
httpd_use_nfs                   off
httpd_use_openscryptoki         off
httpd_use_openstack             off
httpd_use_sasl                  off
httpd_verify_dns                off
laviljin@aviljin ~1$ _

```

Рис. 4.4: Переключатели SELinux для Apache

5. Посмотрим статистику по политике. (рис. 4.5)

seinfo

```

[aviljin@aviljin ~]$ seinfo
Statistics for policy file: /sys/fs/selinux/policy
Policy Version:          33 (MLS enabled)
Target Policy:           selinux
Handle unknown classes:  allow
Classes:                 135
Permissions:             457
Sensitivities:           1
Categories:              1024
Types:                   5057
Attributes:              253
Users:                   8
Roles:                   14
Booleans:                348
Cond. Expr.:             378
Allow:                   63159
Neverallow:              0
Auditallow:              163
Dontaudit:               8417
Type_trans:              249506
Type_change:             87
Type_member:             35
Range_trans:             6161
Role allow:              37
Role_trans:              418
Constraints:             70
Validatetrans:           0
MLS Constrain:          72
MLS Val. Tran:           0
Permissives:             2
Polcap:                  6
Defaults:                7
Typebounds:              0
Allowxperm:              0
Neverallowxperm:        0
Auditallowxperm:        0
Dontauditxperm:         0
Ibendportcon:            0
Ibpkeycon:               0
Initial SIDs:            27
Fs_use:                  35
Genfscon:                109
Portcon:                 660
Netifcon:                0
Nodecon:                 0
[aviljin@aviljin ~]$ _

```

Рис. 4.5: Статистика по политике

- Определив тип файлов и поддиректорий, находящихся в директориях `/var/www`, `/var/www/html`. Определив круг пользователей, которым разрешено создание файлов в директории `/var/www/html`. (рис. 4.6)

```

ls -lZ /var/www
ls -lZ /var/www/html
ls -alF /var/www

```

```

[aviljin@aviljin ~]$ ls -lZ /var/www
total 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 May 16 23:21 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 6 May 16 23:21 html
[aviljin@aviljin ~]$ ls -lZ /var/www/html
total 0
[aviljin@aviljin ~]$ ls -alF /var/www
total 4
drwxr-xr-x. 4 root root 33 Oct 14 16:28 ./
drwxr-xr-x. 20 root root 4096 Oct 14 16:28 ../
drwxr-xr-x. 2 root root 6 May 16 23:21 cgi-bin/
drwxr-xr-x. 2 root root 6 May 16 23:21 html/
[aviljin@aviljin ~]$

```

Рис. 4.6: Тип файлов и поддиректорий Apache

- Создайте от имени суперпользователя html-файл в `/var/www/html/test.html`. Проверим контекст созданного файла. Обратимся к файлу через веб-сервер. (рис. 4.7, 4.8)

```
<html>
  <body>
    test
  </body>
</html>
```

```
touch /var/www/html/test.html
vim /var/www/html/test.html
ls -Z /var/www/html/test.html
lynx http://127.0.0.1/test.html
```



```
[root@aviljin aviljin]# cat /var/www/html/test.html
<html>
  <body>
    test
  </body>
</html>
[root@aviljin aviljin]# ls -Z /var/www/html/test.html
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test.html
[root@aviljin aviljin]# lynx http://127.0.0.1:80/test.html
```

Рис. 4.7: Создание html.test



Рис. 4.8: Просмотр <http://127.0.0.1/test.html> (1)

8. Изменим контекст файла `/var/www/html/test.html` с `httpd_sys_content_t` на любой другой, к которому процесс `httpd` не должен иметь доступа, например, на `samba_share_t`. Попробуем ещё раз получить доступ к файлу через веб-сервер. Просмотрим log-файлы веб-сервера Apache. (рис. 4.9, 4.10, 4.11)

```
chcon -t samba_share_t /var/www/html/test.html
```

```
ls -Z /var/www/html/test.html
```

```
lynx http://127.0.0.1/test.html
```

```
tail /var/log/audit/audit.log
```

```
root@aviljin aviljinl# chcon -t samba_share_t /var/www/html/test.html
root@aviljin aviljinl# ls -Z /var/www/html/test.html
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html
root@aviljin aviljinl# exit
exit
```

Рис. 4.9: Изменение контекст файла test.html



Рис. 4.10: Просмотр <http://127.0.0.1/test.html> (2)

```

type=AVC msg=audit(1697292622.829:152): avc: denied { getattr } for pid=1392 comm="httpd" path="/
var/www/html/test.html" dev="dm-0" ino=33656787 scontext=system_u:system_r:httpd_t:s0 tcontext=uncon
fined_u:object_r:samba_share_t:s0 tclass=file permissive=0
type=SYSCALL msg=audit(1697292622.829:152): arch=c000003e syscall=262 success=no exit=-13 a0=ffffff9
c a1=7fac2c03b9a8 a2=7fac297f98b0 a3=100 items=0 ppid=1390 pid=1392 auid=4294967295 uid=48 gid=48 eu
id=48 suid=48 fsuid=48 egid=48 sgid=48 fsgid=48 tty=(none) ses=4294967295 comm="httpd" exe="/usr/sbi
n/httpd" subj=system_u:system_r:httpd_t:s0 key=(null)ARCH=x86_64 SYSCALL=newfstatat AUID="unset" UID
="apache" GID="apache" EUID="apache" SUID="apache" FSUID="apache" EGID="apache" SGID="apache" FSGID=
"apache"
type=PROCTITLE msg=audit(1697292622.829:152): proctitle=2F7573722F7362696E2F6874747064002D44464F5245
47524F554E44
type=USER_END msg=audit(1697292719.470:153): pid=1815 uid=1000 auid=1000 ses=1 subj=unconfined_u:unc
onfined_r:unconfined_t:s0-s0:c0.c1023 msg='op=PAM:session_close grantors=pam_keyinit,pam_limits,pam
_systemd,pam_unix,pam_umask,pam_xauth acct="root" exe="/usr/bin/su" hostname=aviljin addr=? terminal=
/dev/tty1 res=success'UID="aviljin" AUID="aviljin"
type=CRED_DISP msg=audit(1697292719.470:154): pid=1815 uid=1000 auid=1000 ses=1 subj=unconfined_u:un
confined_r:unconfined_t:s0-s0:c0.c1023 msg='op=PAM:setcred grantors=pam_unix acct="root" exe="/usr/b
in/su" hostname=aviljin addr=? terminal=/dev/tty1 res=success'UID="aviljin" AUID="aviljin"
type=USER_AUTH msg=audit(1697292824.207:155): pid=2019 uid=1000 auid=1000 ses=1 subj=unconfined_u:un
confined_r:unconfined_t:s0-s0:c0.c1023 msg='op=PAM:authentication grantors=pam_unix acct="aviljin" e
xe="/usr/bin/sudo" hostname=aviljin addr=? terminal=/dev/tty1 res=success'UID="aviljin" AUID="avilji
n"
type=USER_ACCT msg=audit(1697292824.214:156): pid=2019 uid=1000 auid=1000 ses=1 subj=unconfined_u:un
confined_r:unconfined_t:s0-s0:c0.c1023 msg='op=PAM:accounting grantors=pam_unix acct="aviljin" exe="
/usr/bin/sudo" hostname=aviljin addr=? terminal=/dev/tty1 res=success'UID="aviljin" AUID="aviljin"
type=USER_CMD msg=audit(1697292824.215:157): pid=2019 uid=1000 auid=1000 ses=1 subj=unconfined_u:unc
onfined_r:unconfined_t:s0-s0:c0.c1023 msg='cwd="/home/aviljin" cmd=7461696C202F7661722F6C6F672F61756
469742F61756469742E6C6F67 exe="/usr/bin/sudo" terminal=tty1 res=success'UID="aviljin" AUID="aviljin"
type=CRED_REFR msg=audit(1697292824.215:158): pid=2019 uid=1000 auid=1000 ses=1 subj=unconfined_u:un
confined_r:unconfined_t:s0-s0:c0.c1023 msg='op=PAM:setcred grantors=pam_unix acct="root" exe="/usr/b
in/sudo" hostname=aviljin addr=? terminal=/dev/tty1 res=success'UID="aviljin" AUID="aviljin"
type=USER_START msg=audit(1697292824.216:159): pid=2019 uid=1000 auid=1000 ses=1 subj=unconfined_u:u
nconfined_r:unconfined_t:s0-s0:c0.c1023 msg='op=PAM:session_open grantors=pam_keyinit,pam_limits,pam
_systemd,pam_unix acct="root" exe="/usr/bin/sudo" hostname=aviljin addr=? terminal=/dev/tty1 res=suc
cess'UID="aviljin" AUID="aviljin"
[aviljin@aviljin ~]$ _

```

Рис. 4.11: Просмотр audit.log

9. Попробуйте запустить веб-сервер Apache на прослушивание TCP-порта 81.
Перезапустим веб-сервер Apache. (рис. 4.12, 4.13)

```

vim /etc/httpd/conf/httpd.conf
service httpd restart
service httpd status

```

```
# configuration, error, and log files are kept.
#
# Do not add a slash at the end of the directory path. If you point
# ServerRoot at a non-local disk, be sure to specify a local disk on the
# Mutex directive, if file-based mutexes are used. If you wish to share the
# same ServerRoot for multiple httpd daemons, you will need to change at
# least PidFile.
#
ServerRoot "/etc/httpd"

#
# Listen: Allows you to bind Apache to specific IP addresses and/or
# ports, instead of the default. See also the <VirtualHost>
# directive.
#
# Change this to Listen on a specific IP address, but note that if
# httpd.service is enabled to run at boot time, the address may not be
# available when the service starts. See the httpd.service(8) man
# page for more information.
#
#Listen 12.34.56.78:80
Listen 81

#
# Dynamic Shared Object (DSO) Support
#
# To be able to use the functionality of a module which was built as a DSO you
# have to place corresponding 'LoadModule' lines at this location so the
# directives contained in it are actually available _before_ they are used.
# Statically compiled modules (those listed by 'httpd -l') do not need
# to be loaded here.
#
# Example:
# LoadModule foo_module modules/mod_foo.so
#
Include conf.modules.d/*.conf
```

Рис. 4.12: Изменение httpd.conf на прослушивание TCP-порта 81

```
[root@aviljin aviljin]# service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[root@aviljin aviljin]# service httpd status
Redirecting to /bin/systemctl status httpd.service
■ httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; disabled; preset: disabled)
   Active: active (running) since Sat 2023-10-14 17:15:53 MSK; 2min 32s ago
     Docs: man:httpd.service(8)
  Main PID: 2056 (httpd)
    Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served/sec: 0 B/s"
     Tasks: 213 (limit: 17484)
    Memory: 28.8M
       CPU: 86ms
    CGroup: /system.slice/httpd.service
           └─2056 /usr/sbin/httpd -DFOREGROUND
             └─2057 /usr/sbin/httpd -DFOREGROUND
               └─2058 /usr/sbin/httpd -DFOREGROUND
                 └─2059 /usr/sbin/httpd -DFOREGROUND
                   └─2060 /usr/sbin/httpd -DFOREGROUND

Oct 14 17:15:53 aviljin systemd[1]: Starting The Apache HTTP Server...
Oct 14 17:15:53 aviljin httpd[2056]: Server configured, listening on: port 81
Oct 14 17:15:53 aviljin systemd[1]: Started The Apache HTTP Server.
[root@aviljin aviljin]#
```

Рис. 4.13: Перезапуск Apache

10. Проанализируем лог-файлы. Добавим порт 81 в список портов. (данный порт, как оказалось, уже был в списке, из-за этого не произошло сбоя).

Перезапустим сервер. Вернем контекст httpd_sys_content_t. Попробуем получить доступ к файлу через веб-сервер (рис. 4.14, 4.15)

```
tail -n5 /var/log/messages
```

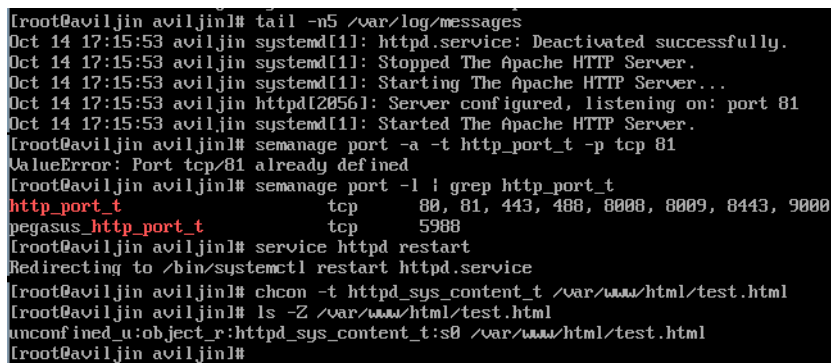
```
semanage port -a -t http_port_t -p tcp 81
```

```
semanage port -l | grep http_port_t
```

```
service httpd restart
```

```
chcon -t httpd_sys_content_t /var/www/html/test.html
```

```
lynx http://127.0.0.1/test.html
```



```
[root@aviljin aviljin]# tail -n5 /var/log/messages
Oct 14 17:15:53 aviljin systemd[1]: httpd.service: Deactivated successfully.
Oct 14 17:15:53 aviljin systemd[1]: Stopped The Apache HTTP Server.
Oct 14 17:15:53 aviljin systemd[1]: Starting The Apache HTTP Server...
Oct 14 17:15:53 aviljin httpd[2056]: Server configured, listening on: port 81
Oct 14 17:15:53 aviljin systemd[1]: Started The Apache HTTP Server.
[root@aviljin aviljin]# semanage port -a -t http_port_t -p tcp 81
ValueError: Port tcp/81 already defined
[root@aviljin aviljin]# semanage port -l | grep http_port_t
http_port_t                tcp      80, 81, 443, 488, 8008, 8009, 8443, 9000
pegasus_http_port_t        tcp      5988
[root@aviljin aviljin]# service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[root@aviljin aviljin]# chcon -t httpd_sys_content_t /var/www/html/test.html
[root@aviljin aviljin]# ls -Z /var/www/html/test.html
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test.html
[root@aviljin aviljin]#
```

Рис. 4.14: Добавление порта 81 в список портов



Рис. 4.15: Просмотр <http://127.0.0.1/test.html> (3)

5 Анализ результатов

Работа выполнена без непредвиденных проблем в соответствии с руководством. Ошибок и сбоев не произошло.

6 Выводы

Нам удалось развить навыки администрирования ОС Linux. Получить первое практическое знакомство с технологией SELinux. Проверить работу SELinux на практике совместно с веб-сервером Apache. Также мы настроили и запустили сервер Apache. Исследовали влияние различных параметров на работу сервера.

Список литературы

1. Терминал Linux [Электронный ресурс]. URL: [https://www.reg.ru/blog/linux-shpargalka-komandy-terminala-dlya-novichkov/#:~:text=%D0%A2%D0%B5%D1%80%D0%BC%D0%B8%D0%BD%D0%B0%D0%BB%20\(%D0%B8%D0%BB%D0%B8%20%D0%BC%D0%B8%20%D0%B8%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F%D0%BC%D0%B8%7D](https://www.reg.ru/blog/linux-shpargalka-komandy-terminala-dlya-novichkov/#:~:text=%D0%A2%D0%B5%D1%80%D0%BC%D0%B8%D0%BD%D0%B0%D0%BB%20(%D0%B8%D0%BB%D0%B8%20%D0%BC%D0%B8%20%D0%B8%20%D1%83%D0%BF%D1%80%D0%B0%D0%B2%D0%BB%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%BF%D0%BE%D0%BB%D1%8C%D0%B7%D0%BE%D0%B2%D0%B0%D1%82%D0%B5%D0%BB%D1%8F%D0%BC%D0%B8%7D).

2. Права доступа [Электронный ресурс]. URL: <https://codechick.io/tutorials/unix-linux/unix-linux-permissions>.

3. Расширенные атрибуты [Электронный ресурс]. URL: <https://wiki.astralinux.ru/pages/viewpage.action?pageId=149063848#:~:text=Common%20Edition%202.12-,%D0%A7%D1%82%D0%BE%20%D1%82%D0%B0%D0%BA%D0%BE%D0%B5%20%D1%80%D0%B0%D1%81%D1%88%D0%B8%D1%80%D0%B5%D0%BD%D0%BD%D1%8B%D0%B5%20%D0%B0%D1%82%D1%80%D0%B8%D0%B1%D1%83%D1%82%D1%8B,%D1%81%20%D1%84%D0%B0%D0%B9%D0%BB%D0%BE%D0%B2%D1%8B%D0%BC%D0%B8%20%D0%BE%D0%B1%D1%8A%D0%B5%D0%BA%D1%82%D0%B0%D0%BC%D0%B8%20%D0%BF%D1%80%D0%BE%D0%B8%D0%B7%D0%B2%D0%BE%D0%BB%D1%8C%D0%BD%D1%8B%D0%B5%20%D0%BC%D0%B5%D1%82%D0%B0%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D0%B5%7D>.

4. Введение в SELinux под CentOS Stream [Электронный ресурс]. URL: <https://ruvds.com/ru/helpcenter/vvedenie-v-selinux-pod-centos-stream/>.

5. Документация Rocky Linux [Электронный ресурс]. URL: <https://docs.rockylinux.org/>.

6. Git-Guides [Электронный ресурс]. URL: <https://github.com/git-guides>.

7. VirtualBox [Электронный ресурс]. URL: <https://www.virtualbox.org/>.