

Лабораторная работа №6

Информационная безопасность

Маметкадыров Ынтымак | НПМбд-02-20

Содержание

1	Цель работы	4
2	Теоретическое введение	5
3	Выполнение лабораторной работы	7
4	Выводы	17
	Список литературы	18

Список иллюстраций

3.1	Проверка режима enforcing политики targeted	7
3.2	Проверка работы веб-сервера	8
3.3	Контекст безопасности веб-сервера Apache	8
3.4	Текущее состояние переключателей SELinux	9
3.5	Статистика по политике	10
3.6	Просмотр файлов и поддиректорий в директории /var/www . . .	11
3.7	Создание файла /var/www/html/test.html	11
3.8	Обращение к файлу через веб-сервер	11
3.9	Изменение контекста	12
3.10	Обращение к файлу через веб-сервер	12
3.11	Просмотр log-файла	13
3.12	Установка веб-сервера Apache на прослушивание TCP-порта 81 . .	13
3.13	Перезапуск веб-сервера и анализ лог-файлов	14
3.14	Проверка установки порта 81	14
3.15	Возвращение исходного контекста файлу и обращение к файлу через веб-сервер	15
3.16	Возвращение Listen 80 и попытка удалить порт 81	15
3.17	Удаление файла test.html	16

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 Теоретическое введение

SELinux (Security-Enhanced Linux) обеспечивает усиление защиты путем внесения изменений как на уровне ядра, так и на уровне пространства пользователя, что превращает ее в действительно «непробиваемую» операционную систему. Впервые эта система появилась в четвертой версии CentOS, а в 5 и 6 версии реализация была существенно дополнена и улучшена.

SELinux имеет три основных режим работы:

- Enforcing: Режим по-умолчанию. При выборе этого режима все действия, которые каким-то образом нарушают текущую политику безопасности, будут блокироваться, а попытка нарушения будет зафиксирована в журнале.
- Permissive: В случае использования этого режима, информация о всех действиях, которые нарушают текущую политику безопасности, будут зафиксированы в журнале, но сами действия не будут заблокированы.
- Disabled: Полное отключение системы принудительного контроля доступа.

Политика SELinux определяет доступ пользователей к ролям, доступ ролей к доменам и доступ доменов к типам.

Контекст безопасности — все атрибуты SELinux — роли, типы и домены.

Более подробно см. в [1].

Apache — это свободное программное обеспечение, с помощью которого можно создать веб-сервер. Данный продукт возник как доработанная версия другого HTTP-клиента от национального центра суперкомпьютерных приложений (NCSA).

Для чего нужен Apache сервер:

- чтобы открывать динамические PHP-страницы,
- для распределения поступающей на сервер нагрузки,
- для обеспечения отказоустойчивости сервера,
- чтобы потренироваться в настройке сервера и запуске PHP-скриптов.

Apache является кроссплатформенным ПО и поддерживает такие операционные системы, как Linux, BSD, MacOS, Microsoft, BeOS и другие.

Более подробно см. в [2].

3 Выполнение лабораторной работы

Вошли в систему под своей учетной записью и убедились, что SELinux работает в режиме enforcing политики targeted с помощью команд “getenforce” и “sestatus” (рис. 3.1).

```
[itmametskadihrov@itmametskadihrov ~]$ getenforce
Enforcing
[itmametskadihrov@itmametskadihrov ~]$ 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
[itmametskadihrov@itmametskadihrov ~]$
```

Рис. 3.1: Проверка режима enforcing политики targeted

Обратились с помощью браузера к веб-серверу, запущенному на нашем компьютере, и убедились, что последний работает с помощью команды “service httpd status” (рис. 3.2).

```
[itmametskadihrov@itmametskadihrov ~]$ service httpd status
Redirecting to /bin/systemctl status httpd.service
• httpd.service - The Apache HTTP Server
  Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; preset: disabled)
  Active: active (running) since Thu 2023-10-12 15:47:10 +06; 33s ago
    Docs: man:httpd.service(8)
 Main PID: 2954 (httpd)
  Status: "Total requests: 0; Idle/Busy workers 100/0; Requests/sec: 0; Bytes served: 0"
    Tasks: 213 (limit: 25469)
  Memory: 41.4M
    CPU: 239ms
  CGroup: /system.slice/httpd.service
          └─2954 /usr/sbin/httpd -DFOREGROUND
            └─2980 /usr/sbin/httpd -DFOREGROUND
              └─2984 /usr/sbin/httpd -DFOREGROUND
                └─2985 /usr/sbin/httpd -DFOREGROUND
                  └─2986 /usr/sbin/httpd -DFOREGROUND

окт 12 15:45:45 itmametskadihrov systemd[1]: Starting The Apache HTTP Server...
окт 12 15:46:26 itmametskadihrov httpd[2954]: AH00558: httpd: Could not reliably determine the server's fully qualified domain name, setting AllowOverride None
окт 12 15:47:10 itmametskadihrov systemd[1]: Started The Apache HTTP Server.
окт 12 15:47:10 itmametskadihrov httpd[2954]: Server configured, listening on: port 80
```

Рис. 3.2: Проверка работы веб-сервера

С помощью команды “ps auxZ | grep httpd” определили контекст безопасности веб-сервера Apache - httpd_t (рис. 3.3).

```
[itmametskadihrov@itmametskadihrov ~]$ ps auxZ | grep httpd
system_u:system_r:httpd_t:s0 root 2954 0.0 0.2 20328 11708 ? Ss 15:45 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 2980 0.0 0.1 21664 7576 ? S 15:47 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 2984 0.0 0.2 1079476 11160 ? Sl 15:47 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 2985 0.0 0.4 1210612 19340 ? Sl 15:47 0:00 /usr/sbin/httpd -DFOREGROUND
system_u:system_r:httpd_t:s0 apache 2986 0.0 0.4 1079476 19332 ? Sl 15:47 0:00 /usr/sbin/httpd -DFOREGROUND
unconfined_u:unconfined_r:unconfined_t:s0-s0:c0.c1023 itmamet+ 3223 0.0 0.0 221688 2460 pts/0 S+ 15:48 0:00 grep --color=auto httpd
```

Рис. 3.3: Контекст безопасности веб-сервера Apache

Посмотрели текущее состояние переключателей SELinux для Apache с помощью команды “sestatus -bigrep httpd”, многие из переключателей находятся

в положении “off” (рис. 3.4).

```
[itmametkadihrov@itmametkadihrov ~]$ sestatus -bigrep httpd
sestatus: invalid option -- 'i'

Usage: sestatus [OPTION]

  -v  Verbose check of process and file contexts.
  -b  Display current state of booleans.

Without options, show SELinux status.
[itmametkadihrov@itmametkadihrov ~]$ sestatus -b httpd
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

Policy booleans:
abrt_anon_write                off
abrt_handle_event              off
abrt_upload_watch_anon_write   on
antivirus_can_scan_system      off
antivirus_use_jit              off
auditadm_exec_content          on
authlogin_nsswitch_use_ldap     off
authlogin_radius               off
authlogin_ubuntu               off
```

Рис. 3.4: Текущее состояние переключателей SELinux

Посмотрели статистику по политике с помощью команды “seinfo”. Множество

пользователей - 8, ролей - 14, типов 4995 (рис. 3.5).

```
[itmametskadihrov@itmametskadihrov ~]$ 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:                    5100     Attributes:              258
Users:                    8         Roles:                   14
Booleans:                 353      Cond. Expr.:             384
Allow:                    65008     Neverallow:              0
Auditallow:              170       Dontaudit:               8572
Type_trans:              265344    Type_change:             87
Type_member:              35        Range_trans:             6164
Role allow:              38         Role_trans:              420
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         Ibkeycon:                0
Initial SIDs:            27        Fs_use:                  35
Genfscon:                109       Portcon:                 660
Netifcon:                0         Nodecon:                 0
[itmametskadihrov@itmametskadihrov ~]$
```

Рис. 3.5: Статистика по политике

С помощью команды “ls -lZ /var/www” посмотрели файлы и поддиректории, находящиеся в директории /var/www. Используя команду “ls -lZ /var/www/html”, определили, что в данной директории файлов нет. Только владелец/суперпользователь

может создавать файлы в директории /var/www/html (рис. 3.6).

```
[itmametskadihrov@itmametskadihrov ~]$ ls -lZ /var/www
итого 0
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_script_exec_t:s0 6 мая 17 02:21 cgi-bin
drwxr-xr-x. 2 root root system_u:object_r:httpd_sys_content_t:s0 6 мая 17 02:21 html
[itmametskadihrov@itmametskadihrov ~]$ ls -lZ /var/www/html
итого 0
[itmametskadihrov@itmametskadihrov ~]$
```

Рис. 3.6: Просмотр файлов и поддиректорий в директории /var/www

От имени суперпользователя создали html-файл /var/www/html/test.html. Контекст созданного файла - httpd_sys_content_t (рис. 3.7).

```
[itmametskadihrov@itmametskadihrov ~]$ su -
Пароль:
[root@itmametskadihrov ~]# touch /var/www/html/test.html
[root@itmametskadihrov ~]# nano /var/www/html/test.html
[root@itmametskadihrov ~]# cat /var/www/html/test.html
<html>
<body>test</body>
</html>
[root@itmametskadihrov ~]# su - itmametskadihrov
[itmametskadihrov@itmametskadihrov ~]$ ls -lZ /var/www/html
итого 4
-rw-r--r--. 1 root root unconfined_u:object_r:httpd_sys_content_t:s0 33 окт 13 18:11 test.html
[itmametskadihrov@itmametskadihrov ~]$
```

Рис. 3.7: Создание файла /var/www/html/test.html

Обратились к файлу через веб-сервер, введя в браузере адрес “http://127.0.0.1/test.html”. Файл был успешно отображен (рис. 3.8).

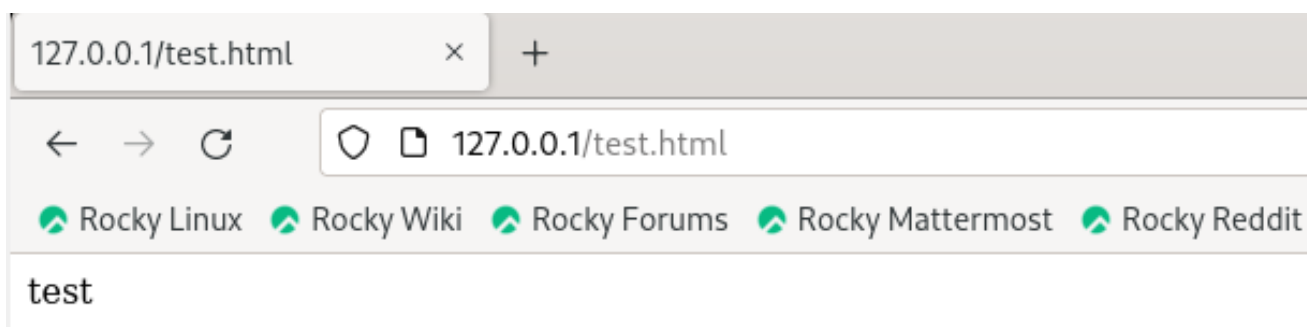


Рис. 3.8: Обращение к файлу через веб-сервер

Изучив справку `man httpd_selinux`, выяснили, что для `httpd` определены следующие контексты файлов: `httpd_sys_content_t`, `httpd_sys_script_exec_t`, `httpd_sys_script_ro_t`, `httpd_sys_script_rw_t`, `httpd_sys_script_ra_t`, `httpd_unconfined_script_exec_t`. Контекст моего файла - `httpd_sys_content_t` (в таком случае содержимое должно быть доступно для всех скриптов `httpd` и для самого демона). Изменили контекст файла на `samba_share_t` командой “`sudo chcon -t samba_share_t /var/www/html/test.html`” и проверили, что контекст поменялся (рис. 3.9).

```
[itmametskadihrov@itmametskadihrov ~]$ ls -Z /var/www/html/test.html
unconfined_u:object_r:httpd_sys_content_t:s0 /var/www/html/test.html
[itmametskadihrov@itmametskadihrov ~]$ chcon -t samba_share_t /var/www/html/test.html
chcon: не удалось изменить контекст безопасности '/var/www/html/test.html' на «unconfined_u:object_r:samba_share_t:s0»: Операция не позволена
[itmametskadihrov@itmametskadihrov ~]$ sudo chcon -t samba_share_t /var/www/html/test.html
[sudo] пароль для itmametskadihrov:
[itmametskadihrov@itmametskadihrov ~]$ ls -Z /var/www/html/test.html
unconfined_u:object_r:samba_share_t:s0 /var/www/html/test.html
[itmametskadihrov@itmametskadihrov ~]$
```

Рис. 3.9: Изменение контекста

Попробовали еще раз получить доступ к файлу через веб-сервер, введя в браузере адрес “`http://127.0.0.1/test.html`” и получили сообщение об ошибке (т.к. к установленному ранее контексту процесс `httpd` не имеет доступа) (рис. 3.10).

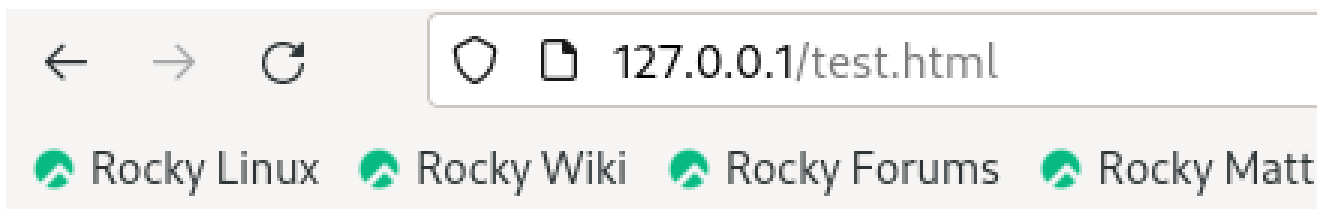


Рис. 3.10: Обращение к файлу через веб-сервер

Командой “`ls -l /var/www/html/test.html`” убедились, что читать данный файл

может любой пользователь. Просмотрели системный лог-файл веб-сервера Apache командой “sudo tail /var/log/messages”, отображающий ошибки (рис. 3.11).

```
[itmametkadihrov@itmametkadihrov ~]$ ls -l /var/www/html/test.html
-rw-r--r--. 1 root root 33 окт 13 18:11 /var/www/html/test.html
[itmametkadihrov@itmametkadihrov ~]$ sudo tail /var/log/messages
sudo: tail /var/log/messages: command not found
[itmametkadihrov@itmametkadihrov ~]$ sudo tail /var/log/messages
Oct 13 18:19:37 itmametkadihrov setroubleshoot[3825]: failed to retrieve rpm info for path '/var/w
ww/html/test.html':
Oct 13 18:19:37 itmametkadihrov systemd[1]: Created slice Slice /system/dbus-:1.1-org.fedoraprojec
t.SetroubleshootPrivileged.
Oct 13 18:19:37 itmametkadihrov systemd[1]: Started dbus-:1.1-org.fedoraproject.SetroubleshootPriv
ileged@0.service.
Oct 13 18:19:39 itmametkadihrov setroubleshoot[3825]: SELinux запрещает /usr/sbin/httpd доступ get
attr к файл /var/www/html/test.html. Для выполнения всех сообщений SELinux: sealert -l 79131f1a-0d
b9-4270-88e0-914585b5c902
Oct 13 18:19:39 itmametkadihrov setroubleshoot[3825]: SELinux запрещает /usr/sbin/httpd доступ get
attr к файл /var/www/html/test.html.#012#012***** Модуль restorecon предлагает (точность 92.2) *
*****#012#012Если вы хотите исправить метку.$TARGETЗнак _PATH по умолчанию долж
ен быть httpd_sys_content_t#012То вы можете запустить restorecon. Возможно, попытка доступа была о
становлена из-за недостаточных разрешений для доступа к родительскому каталогу, и в этом случае по
пытайтесь соответствующим образом изменить следующую команду.#012Сделать#012# /sbin/restorecon -v
/var/www/html/test.html#012#012***** Модуль public_content предлагает (точность 7.83) *****
*****#012#012Если вы хотите лечить test.html как общедоступный контент#012То необходимо изме
нить метку test.html с public_content_t на public_content_rw_t.#012Сделать#012# semanage fcontext
-a -t public_content_t '/var/www/html/test.html'#012# restorecon -v '/var/www/html/test.html'#012#
012***** Модуль catchall предлагает (точность 1.41) *****#012#012Если вы с
читаете, что httpd должно быть разрешено getattr доступ к test.html file по умолчанию.#012То реком
ендуется создать отчет об ошибке.#012Чтобы разрешить доступ, можно создать локальный модуль полити
ки.#012Сделать#012разрешить этот доступ сейчас, выполнив:#012# ausearch -c 'httpd' --raw | audit2a
llow -M my-httpd#012# semodule -X 300 -i my-httpd.pp#012
Oct 13 18:19:39 itmametkadihrov setroubleshoot[3825]: SELinux запрещает /usr/sbin/httpd доступ get
attr к файл /var/www/html/test.html. Для выполнения всех сообщений SELinux: sealert -l 79131f1a-0d
```

Рис. 3.11: Просмотр log-файла

В файле /etc/httpd/conf/httpd.conf заменили строчку “Listen 80” на “Listen 81”, чтобы установить веб-сервер Apache на прослушивание TCP-порта 81 (рис. 3.12).

```
# 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
```

Рис. 3.12: Установка веб-сервера Apache на прослушивание TCP-порта 81

Перезапускаем веб-сервер Apache и анализирует лог-файлы командой “tail -n1 /var/log/messages” (рис. 3.13).

```
[root@itmametskadihrov ~]# systemctl restart httpd
[root@itmametskadihrov ~]# tail -n1 /var/log/messages
Oct 13 18:34:13 itmametskadihrov httpd[4422]: Server configured, listening on: port 81
[root@itmametskadihrov ~]#
```

Рис. 3.13: Перезапуск веб-сервера и анализ лог-файлов

Выполнили команду “semanage port -a -t http_port_t -p tcp 81” и убедились, что порт TCP-81 установлен. Проверили список портов командой “semanage port -l | grep http_port_t”, убедились, что порт 81 есть в списке и запускаем веб-сервер Apache снова (рис. 3.14).

```
[root@itmametskadihrov ~]# semanage port -a -t http_port_t -p tcp 81
ValueError: Порт tcp/81 уже определен
[root@itmametskadihrov ~]# 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@itmametskadihrov ~]# systemctl restart httpd
[root@itmametskadihrov ~]#
```

Рис. 3.14: Проверка установки порта 81

Вернули контекст “httpd_sys_content_t” файлу “/var/www/html/test.html” командой “chcon -t httpd_sys_content_t /var/www/html/test.html” (рис. 3.15) и после этого попробовали получить доступ к файлу через веб-сервер, введя адрес “http://127.0.0.1:81/test.html”, в результате чего увидели содержимое файла - слово “test” (рис. 3.15).

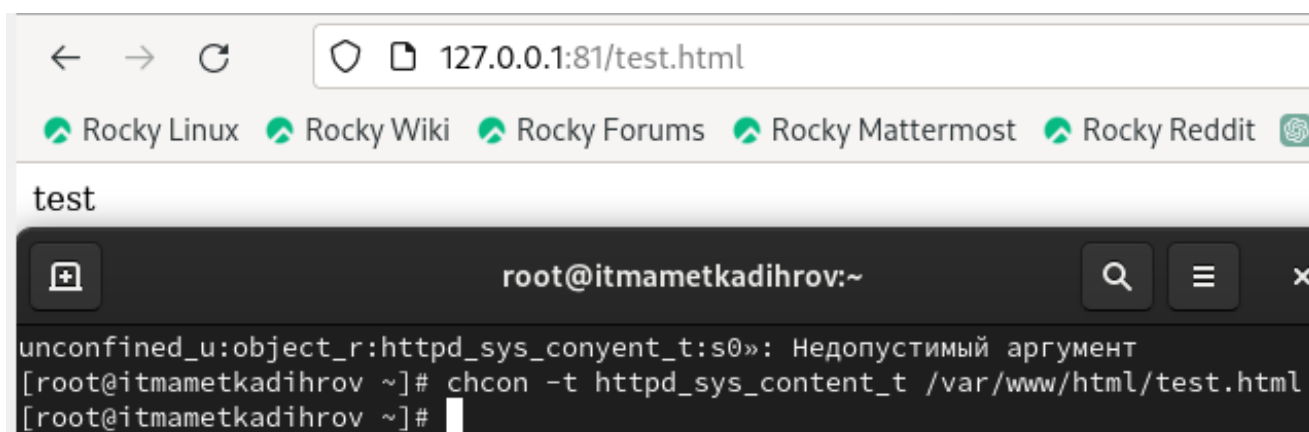


Рис. 3.15: Возвращение исходного контекста файлу и обращение к файлу через веб-сервер

Исправили обратно конфигурационный файл `apache`, вернув “Listen 80”. Попытались удалить привязку `http_port` к 81 порту командой “`semanage port -d -t http_port_t -p tcp 81`”, но этот порт определен на уровне политики, поэтому его нельзя удалить (рис. 3.16).

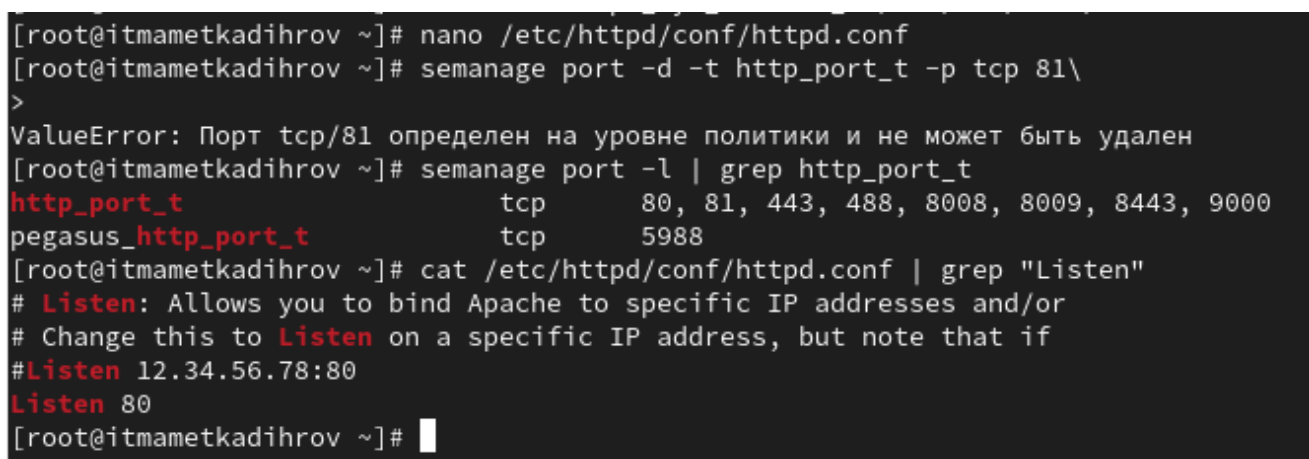


Рис. 3.16: Возвращение Listen 80 и попытка удалить порт 81

Удалили файл “`/var/www/html/test.html`” командой “`rm /var/www/html/test.html`” (рис. 3.17).

```
[root@itmametskadihrov ~]# rm /var/www/html/test.html
rm: удалить обычный файл '/var/www/html/test.html'? y
[root@itmametskadihrov ~]# ls /var/www/html
[root@itmametskadihrov ~]#
```

Рис. 3.17: Удаление файла test.html

4 Выводы

В ходе выполнения данной лабораторной работы мы развили навыки администрирования ОС Linux, получили первое практическое знакомство с технологией SELinux и проверили работу SELinux на практике совместно с веб-сервером Apache.

Список литературы

1. SELinux – описание и особенности работы с системой [Электронный ресурс]. URL: <https://habr.com/ru/company/kingservers/blog/209644/>.
2. Что такое Apache и зачем он нужен? [Электронный ресурс]. URL: <https://2domains.ru/support/vps-i-servery/shto-takoye-apache>.